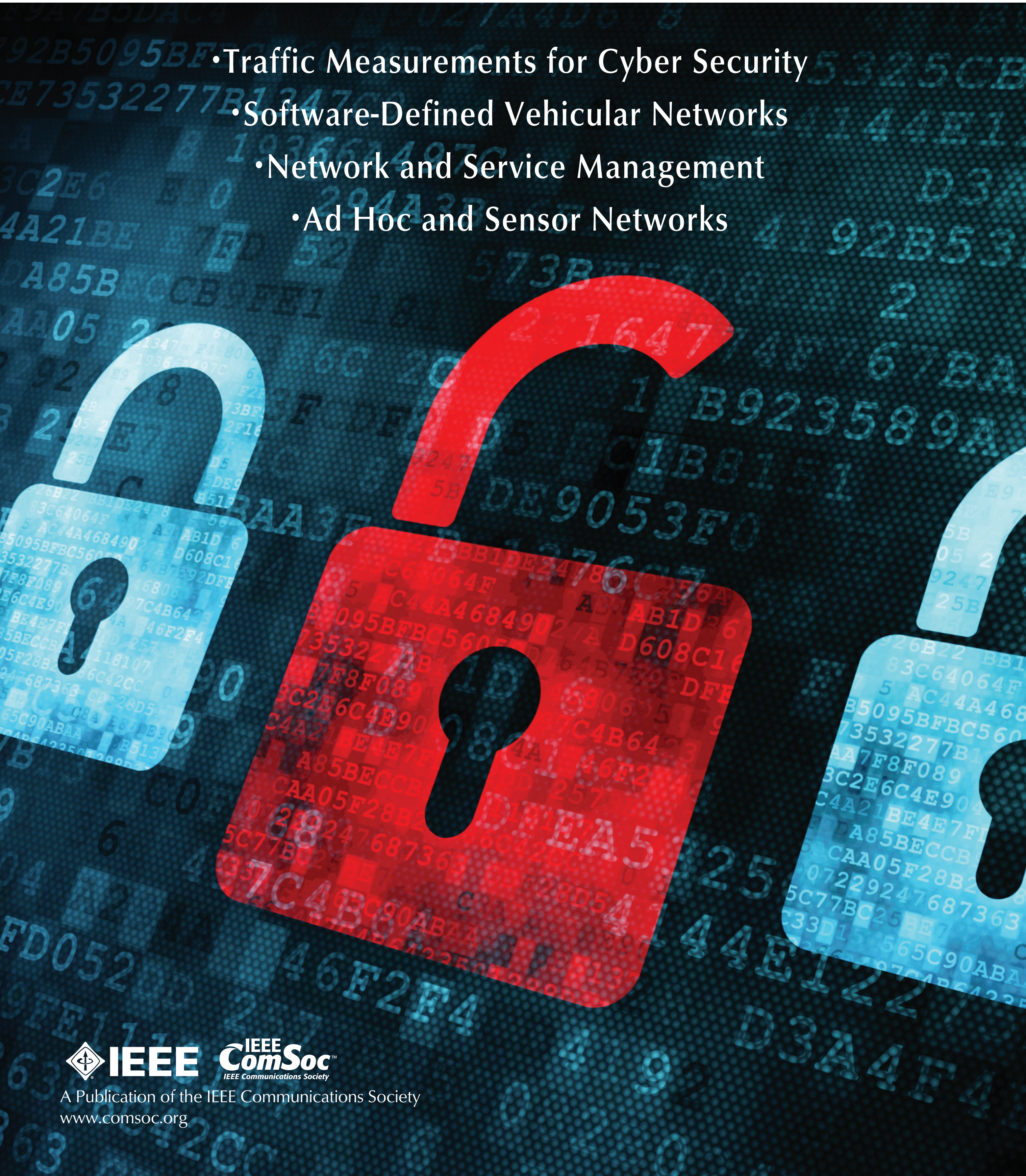
IEEE COMMUNICATIONS MAGAZINE

July 2017, Vol. 55, No. 7



- Traffic Measurements for Cyber Security
- Software-Defined Vehicular Networks
- Network and Service Management
- Ad Hoc and Sensor Networks

IEEE ComSoc IEEE Communications Society
A Publication of the IEEE Communications Society
www.comsoc.org

NATIONAL INSTRUMENTS
Designed by Ettus Research
JTAG
RF 0

IEEE COMMUNICATIONS MAGAZINE

JULY 2017, vol. 55, no. 7
www.comsoc.org/commag

Traffic Measurements for Cyber Security

Guest Editors: Wojciech Mazurczyk, Maciej Korczyn´ski, Koji Nakao, Engin Kirda, Cristian Hesselman, and Katsunari Yoshioka

Software-Defined Vehicular Networks: Architecture, Algorithms, and Applications: Part 1

Guest Editors: Guangjie Han, Mohsen Guizani, Yuanguo Bi, Tom H. Luan, Kaoru Ota, Haibo Zhou, Wael Guibene, and Ammar Rayes

IEEE COMMUNICATIONS MAGAZINE (ISSN 0163-6804) is published monthly by The Institute of Electrical and Electronics Engineers, Inc. Headquarters address: IEEE, 3 Park Avenue, 17th Floor, New York, NY 10016-5997, USA; tel: +1 (212) 705-8900; http://www.comsoc.org/commag. Responsibility for the contents rests upon authors of signed articles and not the IEEE or its members. Unless otherwise specified, the IEEE neither endorses nor sanctions any positions or actions espoused in *IEEE Communications Magazine.*

ANNUAL SUBSCRIPTION: $71: print, digital, and electronic. $33: digital and electronic. $1001: non-member print.

EDITORIAL CORRESPONDENCE: Address to: Editor-in-Chief, Osman S. Gebizlioglu, Huawei Technologies, 400 Crossing Blvd., 2nd Floor, Bridgewater, NJ 08807, USA; tel: +1 (908) 541-3591, e-mail: Osman.Gebizlioglu@huawei.com.

POSTMASTER: Send address changes to *IEEE Communications Magazine*, IEEE, 445 Hoes Lane, Piscataway, NJ 08855-1331. GST Registration No. 125634188. Printed in USA. Periodicals postage paid at New York, NY and at additional mailing offices. Canadian Post International Publications Mail (Canadian Distribution) Sales Agreement No. 40030962. Return undeliverable Canadian addresses to: Frontier, PO Box 1051, 1031 Helena Street, Fort Eire, ON L2A 6C7.

SUBSCRIPTIONS: Orders, address changes — IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08855-1331, USA; tel: +1 (732) 981-0060; e-mail: address.change@ieee.org.

ADVERTISING: Advertising is accepted at the discretion of the publisher. Address correspondence to: Advertising Manager, *IEEE Communications Magazine*, IEEE, 445 Hoes Lane, Piscataway, NJ 08855-1331.

SUBMISSIONS: The magazine welcomes tutorial or survey articles that span the breadth of communications. Submissions will normally be approximately 4500 words, with few mathematical formulas, accompanied by up to six figures and/or tables, with up to 10 carefully selected references. Electronic submissions are preferred, and should be submitted through Manuscript Central: http://mc.manuscriptcentral.com/commag-ieee. Submission instructions can be found at the following: http://www.comsoc.org/commag/paper-submission-guidelines. All submissions will be peer reviewed. For further information contact Tarek El-Bawab, Associate Editor-in-Chief (telbawab@ieee.org).

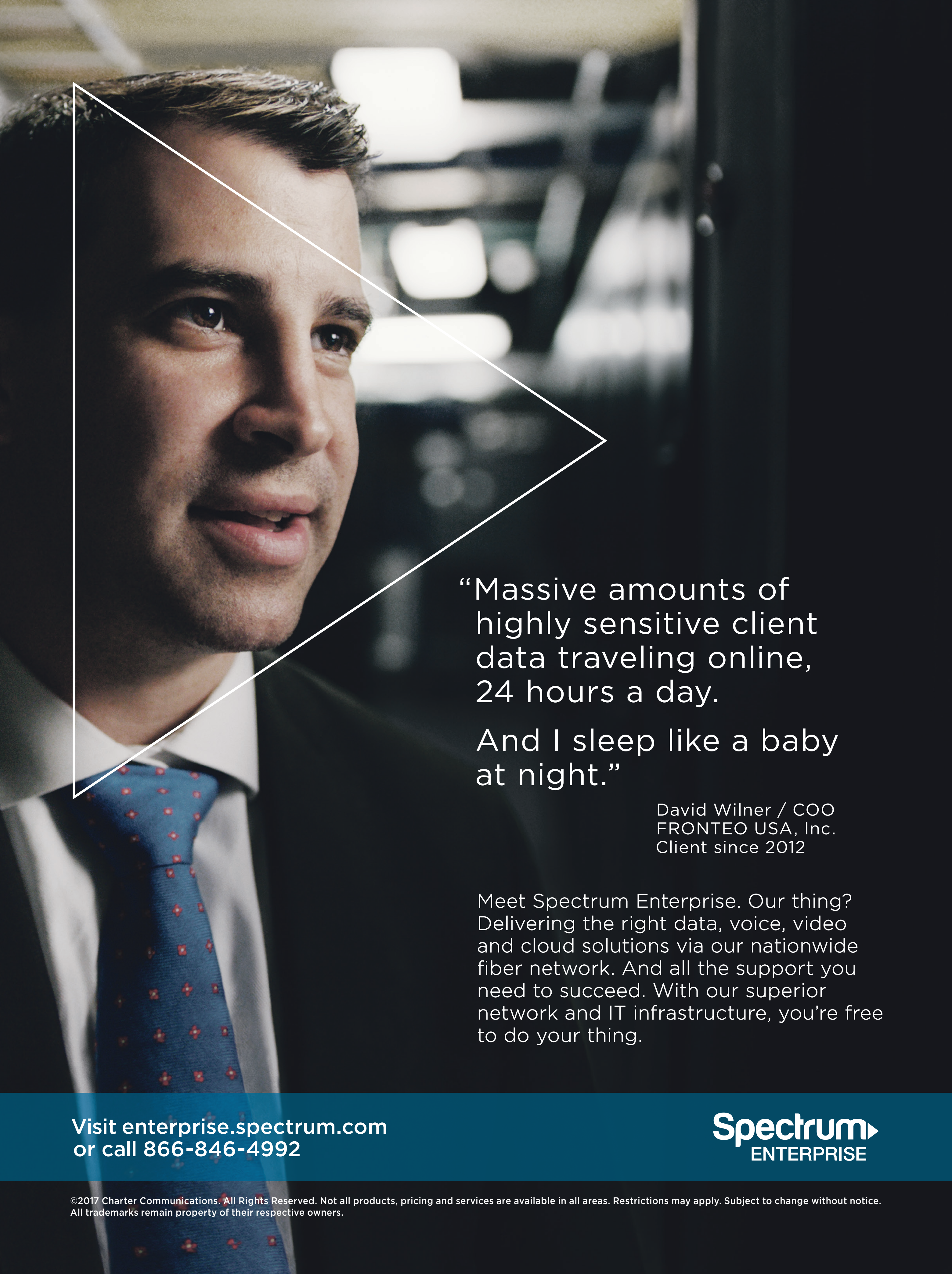
Network and Service Management

Series Editors: George Pavlou and Jürgen Schönwälder

Ad Hoc and Sensor Networks

Series Editors: Edoardo Biagioni, Silvia Giordano, and Ciprian Dobre

Accepted from Open Call



"Massive amounts of highly sensitive client data traveling online, 24 hours a day.

And I sleep like a baby at night."

David Wilner / COO
FRONTEO USA, Inc.
Client since 2012

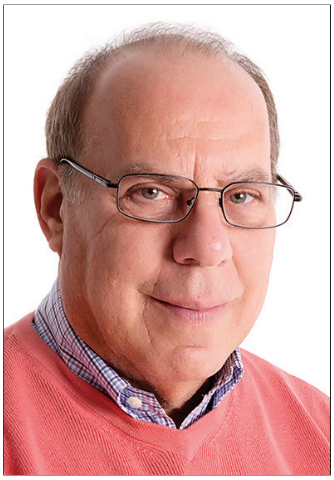
Meet Spectrum Enterprise. Our thing? Delivering the right data, voice, video and cloud solutions via our nationwide fiber network. And all the support you need to succeed. With our superior network and IT infrastructure, you're free to do your thing.

Visit enterprise.spectrum.com
or call 866-846-4992

Spectrum
ENTERPRISE

IEEE COMSOC MARKETING

In the past few years we have seen a significant decrease in ComSoc membership. As you have read in previous President's Pages, we have developed and are in the process of implementing many ideas to reverse this trend. However, we still need the proper notifications and advertisements of our efforts to be displayed or sent to our prospective members. This is where our new marketing efforts come in. Last year we added a new Staff Director of Marketing, and this year a Volunteer Chief Marketing Officer. As you read this President's Page by Stan Moyer and November's by Daphne Bartlett, you will learn of our work to date to address the membership issue.



Harvey Freeman

Stan Moyer

People use the word "marketing" to mean different things. To many, marketing is simply advertising. However, I believe marketing is more than that. In fact, I like the American Marketing Association's definition:

"Marketing is the activity, set of institutions, and processes for creating, communicating, delivering, and exchanging offerings that have value for customers, clients, partners, and society at large."

The main reason I like that definition is that it contains the phrase "offerings that have value for customers." One of the main functions of marketing is determining exactly what the customer values — what they need or want. So for the IEEE Communications Society, what does marketing mean? That is what I'd like to share in this column this month.

Before I delve into that, let me first explain who I am. My name is Stan Moyer, and I have been a volunteer member of the Communications Society (ComSoc) since 1990. I have served on the ComSoc Board of Governors in a variety of roles for the past 14 years, including chairing an ad hoc committee on marketing that was created by the ComSoc President, Harvey Freeman. Most recently I was appointed the Chief Marketing Officer (CMO) of ComSoc, a position that was approved by the ComSoc Board of Governors at its meeting in Paris in May 2017. The creation of this role is strong evidence of the Board's support for marketing and the importance it has for ComSoc. Additionally, about a year ago, ComSoc hired a staff Marketing Director, Daphne Lee Bartlett, to fill a position that had been vacant for quite some time. The two of us have been collaborating this past year to assess ComSoc's marketing needs and opportunities.

So now, let's get back to what marketing means for ComSoc. If we want marketing to help ensure that our offerings have value for customers, then we need to understand more about ComSoc's customers. Essentially, ComSoc's customers are anybody that ComSoc provides with a service and/or product, so that list includes ComSoc members, publication subscribers, conference attendees, and recipients of other ComSoc services. Most current ComSoc members have advanced educations, postgraduate degrees in EE, physics, mathematics, computer sciences, business or related fields, and we can assume that ComSoc's non-member customers have similar backgrounds.

The offerings that ComSoc provides can be grouped into several major areas: membership, publications, conferences, standards, and education and training. It should not come as a surprise that these areas align fairly well with the five ComSoc Vice Presidential (VP) areas:

- •Technical and Educational Activities
- •Conferences
- •Member and Global Activities
- •Publications
- •Industry and Standards Activities

The fact that products and services come out of all the VP areas is the major reason that the volunteer marketing position was elevated to the level of CMO, so that the CMO could report directly to the President and not favor any one of the VP areas in particular.

As ComSoc products and services relate to the five VP areas, the people doing the marketing are the volunteers and the ComSoc staff who serve each of those VP areas. Additionally, the ComSoc marketing department, led by Daphne Bartlett, supports the VP areas by facilitating marketing activities; and the ComSoc marketing committee, chaired by the CMO, coordinates and communicates with the VP areas.

The increased emphasis on marketing in the past year has resulted in several major marketing efforts, which will be explained in more detail in the remainder of this article. These efforts encompass:

- •Social media audit and consolidation
- •Website redesign
- •Brand identity toolkit
- •Marketing collateral
- •Marketing strategy

A social media audit was completed early this year which consisted of an analysis of all ComSoc social media channels and accounts. The channels included Facebook, Instagram, LinkedIn, Twitter, blogs, YouTube, and more. The different accounts included those managed by ComSoc staff, local chapters, and different conferences. The result of the audit was a recommendation for consolidation that will "clean-up" the social media channels and accounts that are outdated and/or have low activity, and a plan to improve effectiveness. For example, before the audit a large percentage of ComSoc's conferences had their own Twitter account, so ComSoc's Twitter followers were "following" different

Twitter accounts, which could not all be easily reached. Basically, we had too many accounts, which left our audience confused and not knowing where to go for official ComSoc information. ComSoc will consolidate all of the different social media accounts into a few ComSoc core accounts and use hashtags to promote conferences, events, and other topical areas. By implementing these recommendations, ComSoc should see a more effective and efficient use of social media, with a better return on investment for its efforts, and an ability to promote ComSoc and its products and services to a wider audience.

The website redesign should have the largest impact on ComSoc "customers" as the website is typically the main "window" for viewing, finding, and/or utilizing ComSoc products and services. The main goal of the website redesign is to make the website user-friendly, easy to navigate, and provide a consistent voice and messaging. We want the website to be the "go-to resource" for ComSoc members, potential members, and other customers of ComSoc products and services. ComSoc has engaged with an outside web design agency to assist with the website redesign and has already completed several phases of this work, including discovery (user interviews and needs assessment), definition (content and technical audit), and delivery of a Needs Assessment document that outlines their findings from this research. Currently a brand voice and content strategy is being developed, which will lead into the site build and launch phase. The targeted launch of the redesigned site is the fall of 2017.

A brand identity toolkit will be developed to ensure a more consistent brand identity for all ComSoc events and chapters. This toolkit will explain the basic usage rules for the different corporate identity elements, such as the ComSoc name and logo, and how to best utilize them. The toolkit will include brand guidelines for the logo, typography, and imagery. In addition, the toolkit will have templates and tools such as a PowerPoint template, banner stands, posters, and flyers. The result should help foster ComSoc brand recognition and integrity across local chapters, ComSoc conferences, and other ComSoc sponsored activities.

ComSoc's reserves of marketing collateral has been depleted, so an effort is underway to create new collateral such as brochures, flyers, and ComSoc-branded items such as pins and pens. This collateral is utilized by ComSoc staff and volunteers at various ComSoc events, and other events where ComSoc has a presence. This collateral is a good mechanism for communicating with existing and potential ComSoc customers.

Having a ComSoc marketing strategy is important, as it serves as a guide in framing marketing decisions toward a common set of goals and objectives that correspond to all products and services. The efforts described above were determined to be high priority items, but ComSoc is also developing an overall strategy, in addition to marketing plans for specific events and programs, that will allow ComSoc to be more proactive in its planning and execution of marketing programs. A better understanding of the resources required and available for implementing these programs, and anticipating potential issues, will enable ComSoc to more effectively provide offerings of value to its customers. Finding the time to do this long-range thinking and planning is often challenging in the face of near-term activities and issues, but creating this plan is one of the most important efforts the marketing committee and staff are undertaking this year.

Thanks for taking the time to learn more about ComSoc marketing. We welcome anyone who would like to become more involved and/or provide input. If you are interested in either (or both), please feel free to contact Stan Moyer, the Chief Marketing Officer, at smoyer@comsoc.org, or Daphne Bartlett, the Marketing Director, at daphne.bartlett@comsoc.org.

ComSoc 2017 Election
Take Time to Vote

Ballots were e-mailed and/or postal mailed 15 May 2017 to all ComSoc members (excluding Student Members, Associate Members, and Affiliates) whose memberships were effective prior to 1 May 2017. You must have an e-ballot or paper ballot before you can vote.

VOTE NOW using the URL below. You will need your IEEE Account username/password to access the ballot. If you do not remember your password, you may retrieve it on the voter login page.

https://eballot4.votenet.com/IEEE

If you have questions about the IEEE ComSoc voting process or would like to request a paper ballot, please contact ieee-comsocvote@ieee.org or +1 732 562 3904.

If you do not receive a ballot by 30 June, but you feel your membership was valid before 1 May 2017, you may e-mail ieee-comsocvote@ieee.org or call +1 732 562 3904 to check your member status. (Provide your member number, full name, and address.)

Please note IEEE Policy (Section 14.1) that IEEE mailing lists should not be used for "electioneering" in connection with any office within the IEEE.

Voting for this election closes 21 July 2017 at 4:00 p.m. EDT! Please vote!

INTERCONNECTIONS FOR COMPUTER COMMUNICATIONS AND PACKET NETWORKS

By Roberto Rojas-Cessa, CRC Press, 2017, ISBN 978-1-4822-2696-6, hardcover, 275 pages

Reviewer: Grzegorz Danilewicz

An interconnection network is a major, yet often underestimated, part of telecommunication and computer network nodes. The scope of applications of this type of network is very wide. They can be used in small systems to connect processors or, on the other hand, in very large data centers to connect servers and other equipment mounted in a number of racks.

The book, authored by Rojas-Cessa, is organized into 12 essential chapters. Each part of the book presents problems of interconnecting in slightly different environments. The first part is devoted to interconnection networks for multiprocessors and consists of two chapters. The second part is the largest portion of the book, and it is dedicated to interconnection networks for packet switching in data networks. This part consists of chapters 3-11. The last part contains a single chapter related to data-center networks (DCNs).

The organization of the book is very convenient. It is easy to read each part separately. Moving around the book is easy with a table of contents at the beginning of the book and separate tables of contents starting each chapter. In addition, every chapter ends with sample exercises. Thanks to that, the work can be used as a textbook accompanying lectures or a book for self-improvement.

At the beginning, the author presents a Preface together with an Organization of the Book and additional Suggested Coverage. In this section, Rojas-Cessa explains what should be the best order of reading the book and following the presented content. The author also indicates prerequisites required from the reader before getting acquainted with the individual chapters. These are very limited to the basics of computer networks.

In two chapters of Part I, different interconnection network topologies and the related routing algorithms are described. The three groups of routing algorithms in interconnection networks are described: static, oblivious, and adaptive.

Part II presents all aspects of packet switching, starting from the address lookup problem and ending with multistage switching fabric architectures. Address lookup and packet classification are performed in the network nodes, such as routers, and are strongly related to a switching network control. From this point of view, presentation of these functions encourages readers to follow the rest of Part II, where packet switching functions are presented in details. Basics of packet switching and details about switching fabric structures are discussed in Chapters 5-11. For example, input-queued switches and internally buffered packet switches are presented. Many aspects of switching fabric control are thoroughly discussed. For example, packet matching mechanisms in input-queued switches are presented in detail. In this part of the book, some aspects of traffic models are dealt with for the sake of switching fabric performance analyses. These problems require mathematical background and the readers' ability to analyze equations.

In Part III, the recent solutions related to DCNs are presented. It is only a short review of nine different DCN solutions, with presentation on switch-centric, server-centric and hybrid architectures. Placement of switch functions and physical connections between different elements of DCNs are illustrated. DCN solutions are evaluated, for example, against the length of connection paths and scalability.

All parts of the book are richly illustrated with numerous figures. According to the saying that one image is worth a thousand words, drawings make it much easier for a reader to follow the discussion. By the way, a mixed style of figures is the weakest aspect of the reviewed book, in my opinion. However, this does not change my overall positive rating of the whole book. Some topics have been discussed using complex mathematical tools, but undergraduate students will not have problems understanding the content.

The book ends with a very solid bibliography containing 191 positions. The largest part of the bibliography gathers positions from the 1990s and 2000s, but some positions from the three most recent years are also included. Interested readers will then be able to broaden their self-studies on interconnection networks. The bibliography is followed by the useful terms index.

In my opinion, this book is mostly aimed at undergraduate students interested in modern telecommunication and computer networks. Nevertheless, graduate students will also find this book a helpful textbook for their learning efforts.

GCN GLOBAL COMMUNICATIONS NEWSLETTER

July 2017
ISSN 2374-1082

MEMBER AND GLOBAL ACTIVITIES

Latin America Region

Interview with Carlos Lozano, Director of the LA Region

By Stefano Bregni, Vice-President for Member and Global Activities, and Carlos Lozano, Director of the LA Region

This is the eigth article in the series started in November 2016 and published monthly in the IEEE ComSoc *Global Communications Newsletter*, which covers all areas of IEEE ComSoc Member and Global Activities. In this series of articles, I introduce the six MGA Directors (Sister and Related Societies; Membership Services; AP, NA, LA, EMEA Regions) and the two Chairs of the Women in Communications Engineering (WICE) and Young Professionals (YP) Standing Committees. In each article, one by one they present their sector activities and plans.

Stefano Bregni

Carlos Lozano

In this issue, I interview Carlos Lozano, Director of the Latin America Region.

Carlos A. Lozano Garzon is Head of the postgraduate degree in Information Security at the Universidad Catolica de Colombia, Colombia. He received his Ph.D. degree from both the Universidad de los Andes, Bogota, Colombia, and the Universida de Girona, Spain, in 2017. From 2005 to 2012, he was an assistant professor at the Universidad de San Buenaventura, Colombia. As an IEEE ComSoc volunteer in the Latin-America Region, Carlos has served as DLT/DSP Coordinator, Membership Development Coordinator, ComSoc Student Chapter Advisor, ComSoc Colombia Chapter Chair, and Technical Activities Coordinator at ComSoc Colombia. He served on the organizing committee of the IEEE Colombian Conference on Communications and Computing, the IEEE Latin-American Conference on Communications, and the IEEE Conference of the Andean Council. He was the Publicity Vice-chair for Latin-America for IEEE Globecom 2015.

It is my true pleasure to interview Carlos, my friend for a very long time, and offer him the opportunity to outline his initiatives and plans in the LA Region.

Bregni: Hola Carlos! Welcome to the *Global Communications Newsletter*. Would you please introduce the IEEE ComSoc Latin America Region to our readers?

Lozano: In Latin America, ComSoc has a presence in 16 countries through 24 professional chapters and 43 student chapters distributed in Central and South America and the Caribbean. The number of ComSoc members in the LA Region is 933 in total (as of April 2017), distributed as shown in the Table.

IEEE current grade description	Count of members
Affiliate	9
Associate Member	10
Fellow	1
Graduate Student Member	70
Life Fellow	1
Life Member	17
Life Senior	17
Member	538
Senior Member	113
Student Member	157
Total	**933**

Table 1. Members by category in the Latin America Region.

Bregni: How is the LA Board organized?

Lozano: Our Latin America Board is responsible for stimulating, coordinating and promoting the activities of ComSoc members and chapters throughout the LA region. This Board is composed of the following volunteers:

- Secretary and Membership Development: Ignacio Castillo
- Technical Activities: José David Cely
- Awards: Lisandro Zambenedetti
- *Global Communications Newsletter* Reporter: Carlos Martínez
- Social Networks: Ana Maria Ospina
- Industry Relations Committee: Pedro Aguilera
- Student Activities Committee: Maytee Zambrano
- DLT and DSP Coordinator: Carlos Eduardo Velasquez
- Advisory Committee: Nelson Fonseca
- Advisory Committee: Araceli Garcia
- Advisory Committee: Ricardo Veiga

In this two years we want to improve the board's communication with chapter chairs in order to provide better support for their local activities; propose strategies to enroll industry professionals to participate in these activities; encourage the grade advancement of our members; and finally to strengthen our regional conference, the IEEE Latin-American Conference on Communications.

Bregni: Every year, Regional Boards assign the Chapter Achievement Award to the best Chapter of the year in the Region, based on successful activities reported. How do you select the CAA Winner in the Latin America Region?

Lozano: In order to select the winner of the Chapter Achievement Award, we examined questionnaires of 19 Chapters, reporting their activities in the previous year. Among those, the activities of the Panama Chapter were considered outstanding.

Through a joint effort with IEEE local chapters and other organizations such as 5G Americas, the chapter succeeded in promoting the importance of ComSoc in the development and implementation of actual and future technologies.

The Panama Chapter Chair is Dr. Maytee Zambrano. She was invited to receive their award at the Awards Luncheon Ceremony at the recent IEEE GLOBECOM 2016 Conference in Washington, DC.

Bregni: Do you have any other significant LA Region Award?

Lozano: Nowadays, we have two Awards to recognize our regional members:

(Continued on Newsletter page 4)

CHAPTER REPORT

Activities of the IEEE ComSoc Nanjing Chapter

By Professor Lianfeng Shen, Chair of IEEE ComSoc Nanjing Chapter

The IEEE ComSoc Nanjing Chapter was established in October 2009. Currently, the IEEE ComSoc Nanjing Chapter is responsible for the members and activities in three provinces of China, Jiangsu, Zhejiang, and Anhui, one of the most important and prosperous areas in China. The activities of the Chapter in 2016 were focused on the following main areas:

- •Organizing or contributing to the organization of international conferences on communications.
- •Organization of presentations, short courses, and seminars offered by world-class specialists from well known institutes.

IEEE Vehicular Technology Conference (VTC) 2016 Spring

The 83rd VTC (VTC 2016 Spring) was held on 15–18 May 2016 in Nanjing. VTC Nanjing was organized by Southeast University and co-sponsored by the IEEE ComSoc Nanjing Chapter. The general co-chair Prof. Xiaohu You, the technical program co-chair Prof. Fu-Chun Zheng, and the publicity co-chair Prof. Xiqi Gao were from Southeast University in Nanjing.

As one of the top conferences on communications and vehicular electronics, VTC has a long history of over 60 years and a well-established reputation in both IEEE ComSoc and the IEEE Vehicular Technology Society (VTS). VTC 2016 Spring in Nanjing was historic, as it was the first time VTC was held in China.

VTC Nanjing attracted more than 600 academics, engineers, and students from around the world. The conference's technical program included one plenary panel, three keynote speeches, seven tutorials, seven workshops, more than 400 oral presentations, and more than 100 poster presentations. VTS president, Prof. Javier Gozalvez, praised VTC Nanjing as "one of the best VTC in history."

International Conference on Wireless Communications and Signal Processing (WCSP) 2016

WCSP 2016 was held on 13–15 October 2016 in Yangzhou, China, organized by Nanjing University of Posts and Telecommunications and co-sponsored by the IEEE ComSoc Nanjing Chapter. WCSP 2016 brought together international researchers from academia and practitioners from industry to meet and exchange ideas and recent research advances on all aspects of wireless communications and signal processing.

Following the great success of WCSP 2009-2015, WCSP 2016 attracted more than 350 academics, engineers, and students from around the world. There were a total of 779 submissions, which were rigorously and independently peer-reviewed by 193 TPC members and many reviewers. Based on relevance, originality, technical contributions and presentation, 310 high quality papers were accepted (an acceptance ratio of 39.79 percent). Accepted papers were grouped into 49 sessions within seven technical symposia. Five keynote speeches were delivered by Prof. Nirwan Ansari, Prof. Wei Zhang, Dr. James Kimery, Prof. Rui Zhang, and Prof. Dan Schonfeld.

WCSP 2017 will be organized by Southeast University in Nanjing.

WCSP 2016: the general co-chair Prof. Zhen Yang gave the welcome speech at the opening ceremony.

IEEE VTC 2016 Spring. From left to right: Prof. Jae Hong Lee, Prof. Pingzhi Fan, Prof. Fu-Chun Zheng, Mr. Ganghua Yang, Dr. Wanshi Chen, Prof. Javier Gozalvez, Prof. Xiaohu You, Prof. Zhisheng Niu.

IEEE VTC 2016 Spring. Prof. Lajos Hanzo and Prof. Javier Gozalvez.

Photos from the Third Women's Workshop on Communications and Signal Processing at IEEE GLOBECOM 2016, Washington DC, US.

A Quick Look at Other Activities of the Nanjing Chapter in 2016

During 2016, the National Mobile Communication Research Laboratory at Southeast University has hosted a number of lectures and seminars by distinguished experts in communications, listed below.

- •Prof. Eric Klumperink provided a fresh look at cognitive radio transceiver chips on 1 March.
- •Profs. Pascal Lorenz and Abbas Jamalipour spoke about architectures of next generation wireless networks and software defined dense cellular networks on 12 April.
- •Prof. Geoffrey Ye Li gave the lecture "LTE on Unlicensed Band" on 11 May.
- •Prof. Rui Zhang shared his broad knowledge of wireless communication with unmanned aerial vehicles on 19 May.
- •Prof. Jianwei Huang gave a talk on crowd-sourced mobile video streaming on 3 June.

Amateur Radio Lectures In Poland and India

By Miroslav Skoric, IEEE Austria Section

To give some personal contribution to the IEEE ComSoc's activities 'in the field', I decided to make two conference tours in 2016, one in IEEE Region 8 and another in Region 10. The 24th International Conference of Computer Networks (CN2016) was held in Palac Brunow, a rural place in southwestern Poland, while ICRCICN 2016 was in Kolkata, India.

In Poland I performed my complete program (theoretical+practical) as a 'one-man-band' because in that area THERE were no radio amateurs. ThankS to ADI AF-16, my new portable VHF radio transceiver, I was able to reach an FM repeater located at Góra Szrenica, a mountain approximately 40 km away from the conference venue. It was also an opportunity for Piotr Gaj from Silesian University of Technology in Gliwice, the head of the CN2016 organizing committee, to get the personal touch with the Taiwanese product.

After the conference, I spent a few days in Wroclaw to visit local radio amateurs. The first correspondent was Mike SQ6WEM, who provided information on local radio clubs. The next day, Marcin SQ6POL organized my visit to the SP6PWS club location, where I exchanged experiences with local radio enthusiasts. The following day I was in touch with Robert 3Z6AET, who was active in dasr.pl (an emergency radio communication group in that part of Poland). Finally, a friendly meeting was made with Michal SQ6IYV from Politechnika Wroclawska (Wroclaw University of Technology). Polish amateur radio life seemed to be in good condition.

The India tour started with an extended lecture at ABES Engineering College in Ghaziabad, a city close to New Delhi. The lecture was split into two days because I had a plenty of slides. Each day we had a hearty lunch, provided by the school. Everything was very well organized thankfully to Pankaj Sharma, associate professor at ABES (first from right in the photo).

Lecture at ABES Engineering College in Ghaziabad.

Ambarish Nag Biswas, secretary of the West Bengal Amateur Radio Club

The lecture attracted many female students. As I noticed, Indian schooling is not worried about any decrease of interest for technical education for young women. The workshop was combined with a practical session performed by a skilled radio amateur, Sandeep Baruah, VU2MUE, scientist-E with Vigyan Prasar, an autonomous body under the Department of Science & Technology, Government of India.

My next lecture was with the SSIT Chapter at the IEEE Kerala Section. The chapter's chair, Satish Babu, and his associate, Ranjit Nair, took care of everything. The session was held at the Science and Technology Museum in Trivandrum, collocated with the local radio club's office. There were approximately 30 participants, including a few members of the 'Trivandrum Amateur Radio Society' (TARS). The museum's director opened the event.

Piotr Gaj, Silesian University of Technology in Gliwice, head of the CN2016 organizing committee.

Unfortunately, due to a tight flight schedule, I spent only 24 hours in the beautiful southern tip of the Indian peninsula. So I continued to Hyderabad to visit the National Institute of Amateur Radio (NIAR). They organized my next full-day session with the GMR Institute of Technology in Rajam, Andhra Pradesh state, where I was accommodated in a very nice guest house. Although the campus of GMR IT was almost three hours by car from the nearest airport, and a few kilometers away from the local village, it had all the needed facilities for students and employees. Back at Hyderabad, I had a nice dinnertime discussion with NIAR's director Mohan Ram, VU2MYH, and his deputy Jose Jacob, VU2JOS.

The main part of the travel was to the Kolkata Conference, organized by the Department of Information Technology, RCC Institute of Information Technology, and the IEEE Young Professionals Affinity Group. At the airport I was greeted by members of the 'West Bengal Amateur Radio Club,' led by its secretary Ambarish Nag Biswas, VU2JFA (Figure 3). He and his boys provided an ad-hoc demo of their radio skills during my tutorial session.

The conference was held at the hotel 'Stadel.' The leading person, professor Siddhartha Bhattacharyya, invited me to give another lecture for his students. The second session took place on the premises of the RCC Institute.

The final part of this journey was a full-day seminar at Thapar University in Patiala. Having a long tradition and pleasant campus environment, that was a good place to finish the educational program. Although it was yet another 24-hour travel segment, I took a chance to visit popular street-food stalls, escorted by my host, Professor A. K. Verma. Returning from Punjab state to New Delhi was via a five-hour car ride. As expected, the 'holy cows' appeared on the roads here and there, but it was all part of the fun.

Indian education has stayed wide open for the amateur radio. Having a good impressions, I can confirm that there will be more tutorials and workshop sessions in years to come. Our plans include starting an 'international conference on the amateur radio in education,' as well as events in the form of 'summer schools.' Should you want to collaborate, please get in touch with me.

LATIN AMERICA REGION/*Continued from page 1*

- Young Professional Award, created with the purpose of rewarding a member with an outstanding and promising professional record in the field of communications.
- Distinguished Service Award, given to a member who made significant contributions to the development of our Society activities in the Region.

For 2016, the awarded members were Leandro Aparecido Villas and Carlos Martinez, respectively.

Bregni: What are the main challenges of the LA Region, in your opinion?

Lozano: We have many challenges in our region, but I think right now the most relevant are the following:

- Declining membership
- Lack of interest of practitioners in the ComSoc activities
- Need for programs to directly support ComSoc Student Chapters.

Coping with these three issues is the base of our plan for this year.

Bregni: What are the best programs in the LA Region?

Lozano: In order to provide more benefits to our members, we are developing a webinar series program. In the past two years, we worked in conjunction with the LA Computer Society on a webinar series pilot program that included 16 sessions. The lectures were presented in Spanish, Portuguese or English[1].

Through this pilot we could impact many young professional and student members, as well as many people who were not yet members of IEEE or our Society.

For this year, we are involving the chapter chairs in order to include more lecturers and more attendees. We are currently working on the topic list and the schedule for the webinars that we plan to develop in the next semester.

Bregni: Can you speak about the DLT in the LA Region? Is it very important in particular for the LA Region?

Lozano: In our region, one of the most successful programs is ComSoc's Distinguished Lecturer Tour (DLT). Through this program our members have access to top-of-the-line lectures delivered by premier lecturers. Also, it is important to note that the chapters use these lectures to attract more members. All the lecturers, using their own style, act as ambassadors who encourage the attendees to join our Society.

In 2016 we organized four tours for eight different sections and 14 cities; for 2017 we have already developed three tours for eight different sections and 10 cities.

Bregni: Can you say something about the IEEE Latin-American Conference on Communications (IEEE LATINCOM)? Why is it important for the LA Region?

Lozano: IEEE LATINCOM is a series of international conferences organized by the ComSoc Latin America Region, which was created in 2009. LATINCOM has been held in Medellin (Colombia, 2009), Bogota (Colombia, 2010), Belem do Pará (Brasil, 2011), Quenca (Ecuador, 2012), Santiago do Chile (Chile, 2013), Cartagena de Indias (Colombia, 2014), Arequipa (Perú, 2015), and Medellin (Colombia, 2016). This year, the conference will be held in Guatemala City on 8–10 November 2017.

IEEE LATINCOM covers the scope of all ComSoc Technical Committees and has a broad technical program that includes renowned keynote speakers and comprehensive tutorials presenting the state of the art in communications. Its prime goal is to provide a platform for researchers and practitioners in Latin America to share research and development results, to meet and to network.

This conference is an attempt by our society to reach out to the members and to offer them a venue that meets the specific needs of our region.

Bregni: How was the last Latin America Regional Chapter Chair Congress?

Lozano: The 2016 Latin America Regional Chapter Chair Congress (LA-RCCC 2016) was held in Medellin, Colombia on 14–15 November 2016, in conjunction with IEEE LATINCOM 2016.

The LA-RCCC was a very successful meeting that had the participation of 17 Chapter Chairs, five LA ComSoc Board members, three Distinguished ComSoc Volunteers (Vice President of Member and Global Activities, Director of Membership Services, and Director of the North America Region), the ComSoc Executive Director, and the IEEE Vehicular Technology Society President.

As a result of this meeting we proposed an ambitious plan for the next few years, focused on increasing membership engagement, especially for young members and practitioners; qualification of our members; promoting member elevations and regional candidates to the Distinguished Lecturer Program; and a joint effort with other Societies in our region to promote projects with a higher social impact.

Bregni: To conclude, can you say anything else to our readers?

Lozano: We encourage our members to be involved in our committees with the aim of strengthening them and being able to develop the programs and projects proposed for the years to come.

[1] http://sites.ieee.org/r9/computer-society-webinars/

GCN GLOBAL COMMUNICATIONS NEWSLETTER

STEFANO BREGNI
Editor
Politecnico di Milano — Dept. of Electronics and Information
Piazza Leonardo da Vinci 32, 20133 MILANO MI, Italy
Tel: +39-02-2399.3503 — Fax: +39-02-2399.3413
Email: bregni@elet.polimi.it, s.bregni@ieee.org

IEEE COMMUNICATIONS SOCIETY

STEFANO BREGNI, VICE-PRESIDENT FOR MEMBER AND GLOBAL ACTIVITIES
CARLOS ANDRES LOZANO GARZON, DIRECTOR OF LA REGION
SCOTT ATKINSON, DIRECTOR OF NA REGION
ANDRZEJ JAJSZCZYK, DIRECTOR OF EMEA REGION
TAKAYA YAMAZATO, DIRECTOR OF AP REGION
CURTIS SILLER, DIRECTOR OF SISTER AND RELATED SOCIETIES

REGIONAL CORRESPONDENTS WHO CONTRIBUTED TO THIS ISSUE
EWELL TAN, SINGAPORE (ewell.tan@ieee.org)

IEEE ComSoc
IEEE Communications Society

www.comsoc.org/gcn
ISSN 2374-1082

NANJING CHAPTER/*Continued from page 2*

- Prof. Koichi Asatani introduced network science and its applications to future networking on 6 June.
- Prof. Lu Gan delivered the talk "Structured Random Matrix Theory" on 27 June.
- Prof. Ying Cui kicked off the technical talk "Joint Caching and Multicasting in Large-Scale Cache-Enabled Wireless Networks" on 30 June.
- Prof. Sheng Yang gave a talk on fading broadcast channels with channel uncertainty on 12 August.
- Prof. Xiaodai Dong shared her knowledge about hybrid processing in massive MIMO for 5G on 19 September.
- Prof. Chengshan Xiao discussed the key problems in underwater acoustic MIMO communications on 10 October.
- Prof. Liuqing Yang gave a lecture on energy-harvesting relay networks and Prof. Rui Zhang presented a paradigm shift of wireless security on 17 October.
- Prof. Tony Q. S. Quek gave an overview of fundamentals and recent advances in 5G on 27 October.
- Prof. Hua Qian shared his broad knowledge of 5G IoT on 24 November.

UPDATED ON THE COMMUNICATIONS SOCIETY'S WEB SITE

www.comsoc.org/conferences

2017

JULY

IEEE ISCC 2017 — IEEE Symposium on Computers and Communications, 3–6 July
Heraklion, Greece
http://www.ics.forth.gr/iscc2017/index.html

IEEE NETSOFT 2017 — IEEE Conference on Network Softwarization, 3–7 July
Bologna, Italy
http://sites.ieee.org/netsoft/

ICUFN 2017 — Int'l. Conference on Ubiquitous and Future Networks, 4–7 July
Milan, Italy
http://icufn.org/

IEEE ICME 2017 — IEEE Int'l. Conference on Multimedia and Expo, 10–14 July
Hong Kong, China
http://www.icme2017.org/

SPLITECH 2017 — Int'l. Multidisciplinary Conference on Computer and Energy Science, 12–14 July
Split, Croatia
http://splitech2017.fesb.unist.hr/

CITS 2017 — Int'l. Conference on Computer, Information and Telecommunication Systems, 21–23 July
Dalian, China
http://atc.udg.edu/CITS2017/

ICCCN 2017 — Int'l. Conference on Computer Communication and Networks, 31 July–3 Aug.
Vancouver, Canada
http://icccn.org/icccn17/

AUGUST

ISWCS 2017 — Int'l. Symposium on Wireless Communication Systems, 28–31 Aug.
Bologna, Italy
http://iswcs2017.org/

SEPTEMBER

ITC29 2017 — International Teletraffic Congress, 4–8 Sept.
Genoa, Italy
https://itc29.org/

IEEE CSCN 2017 — IEEE Conference on Standards for Communications & Networking, 5–7 Sept.
Helsinki, Finland
http://cscn2017.ieee-cscn.org/

ICACCI 2017 — Int'l. Conference on Advances in Computing, Communications and Informatics, 13–16 Sept.
Udupi, India
http://icacci-conference.org/2017/

IEEE Sarnoff Symposium 2017, 18–20 Sept.
Newark, NJ
https://ewh.ieee.org/conf/sarnoff/2017/

SOFTCOM 2017 — Int'l. Conference on Software, Telecommunications and Computer Networks, 21–23 Sept.
Split, Croatia
http://softcom2017.fesb.unist.hr/

IEEE CLOUDNET 2017 — IEEE Int'l. Conference on Cloud Networking, 25–27 Sept.
Prague, Czech Republic
http://cloudnet2017.ieee-cloudnet.org/

OCTOBER

I3C 2017 — IoT Int'l, Innovation Conference, 5–7 Oct.
Saodoa. Morocco
http://i3c2017.emena.org/index.html

IEEE PIMRC 2017 — IEEE Int'l. Symposium on Personal, Indoor & Mobile Radio Communications, 8–13 Oct.
Montreal, Canada
http://pimrc2017.ieee-pimrc.org/2015/08/21/sample-news-post/

IEEE CNS 2017 — IEEE Conference on Communications and Network Security, 9–11 Oct.
Las Vegas, NV
http://cns2017.ieee-cns.org/

HONET-ICT 2017 — Int'l. Conference on Smart Cities: Improving Quality of Life Using ICT & IoT, 9–11 Oct.
Irbid, Jordan
http://honet-ict.org/

WCSP 2017 — Int'l. Conference on Wireless Communications and Signal Processing, 11–13 Oct.
Nanjing, China
http://www.ic-wcsp.org/

CyberC 2017 — Int'l. Conference on Cyber-Enabled Distributed Computing and Knowledge, 12–14 Oct.
Nanjing, China

IEEE HEALTHCOM 2017 — IEEE Int'l. Conference on e-Health Networking, Application & Services, 12–15 Oct.
Dalian, China
http://healthcom2017.ieee-healthcom.org/

IEEE SmartGridComm 2017 — IEEE International Conference on Smart Grid Communications, 16–19 Oct.
Dresden, Germany
http://sgc2017.ieee-smartgridcomm.org/

CSNet 2017 — Cyber Security in Networking Conference, 18–20 Oct.
Rio de Janeiro, Brazil
http://csnet2017.dnac.org/

ICTC 2017 — Int'l. Conference on Information and Communication Technology Convergence, 18–20 Oct.
Jeju Island, Korea
http://ictc2017.org/

ATC 2017 — Int'l. Conference on Advanced Technologies for Communications, 18–20 Oct.
Quynhon, Vietnam
http://atc-conf.org/

INTEC 2017 — Int'l. Conference on Internet of Things, Embedded Systems and Communications, 20–22 Oct.
Gafsa, Tunisia
http://www.iintec.org/

–Communications Society portfolio events appear in bold colored print.
–Communications Society technically co-sponsored conferences appear in black italic print.
–Individuals with information about upcoming conferences, Calls for Papers, meeting announcements, and meeting reports should send this information to: IEEE Communications Society, 3 Park Avenue, 17th Floor, New York, NY 10016; e-mail: p.oneill@comsoc.org; fax: + (212) 705-8996. Items submitted for publication will be included on a space-available basis.

TRAFFIC MEASUREMENTS FOR CYBER SECURITY

Wojciech Mazurczyk

Maciej Korczyński

Koji Nakao

Engin Kirda

Cristian Hesselman

Katsunari Yoshioka

Computers and open communication networks have become increasingly interwoven with our daily lives and have profoundly changed our societies. While this has significantly increased people's well being, our growing dependence on an increasingly pervasive, complex, and ever evolving network infrastructure also poses a wide range of cyber security risks with potentially large socio-economic impacts. For example, the increasing number of ill-secured networked devices in combination with growing network capacities enables miscreants to launch disruptive distributed denial of service (DDoS) attacks, such as the 1.2 Tb/s botnet attack on Dyn of late 2016.

Within this context, network traffic measurements and monitoring have become a crucial line of research. It enables us to enhance our understanding of cyber security threats and use this knowledge to develop new ways to detect and mitigate them. Example applications of network measurement research include the analysis of how malicious software proliferates and operates, and how it exploits users' behavior, assessments of the effectiveness of cyber security countermeasures, of the "badness" of Internet service providers, and estimations of the revenues of cyber criminals.

The aim of this Feature Topic is to further increase the ComSoc community's understanding of the current evolutionary state of cyber threats, defenses, and intelligence. To accomplish this, we brought together nine high-quality papers that discuss the latest results of academic and industry researcher.

In the first article, "Demystifying DDoS-as-a-Service," Zand *et al.* present their analysis of 17 providers of DDoS-as-a-service (DaaS). The authors used various measurements and, for instance, discovered that the DaaS providers used a mix of traditional and application-level DDoS attacks that often existed for only a short period of time, and often executed their attacks through the DNS protocol.

Sood *et al.* analyze the properties of botnet command and control (C&C) panels in their article, "Cybercrime at a Scale: A Practical Study of Deployments of HTTP-Based Botnet Command and Control Panels." The authors analyze the URLs of over 9000 HTTP-based C&C panels and, based on their findings, make several recommendations, such as monitoring for TOR traffic and non-standard HTTP ports because they are typically used to communicate with bot clients.

The third article, "Traffic-Aware Patching for Cyber Security in Mobile IoT," proposes a novel traffic-aware scheme to patch important intermediate nodes based on the traffic volumes to prevent major security exploits in Internet of Things (IoT) devices with limited patching resources in order to limit malware propagation.

The next article, "Characterizing the HTTPS Trust Landscape: A Passive View from the Edge," discusses current shortcomings in the different trust relationships between parties involved in secure HTTPS transactions that affect the security of online users.

The fifth article, "Scalable Traffic Sampling Using Centrality Measures on Software-Defined Networks," proposes traffic measurement which can be achieved by using a packet sampling method that captures data packets at switches and steers them toward, for instance, an intrusion detection system (IDS) on software-defined networks (SDNs).

In the article "Quiet Dogs Can Bite: Which Booters Should We Go After, and What Are Our Mitigation Options?," Santanna *et al.* provide another interesting viewpoint of DDoS-as-a-service. Through a number of measurements, they show that there are several DaaS providers that are under the radar of security initiatives, even advertising high attack power with low price and very popular, and discuss their potential mitigation techniques different entities.

In the next article, "Measuring the Energy Consumption of Cyber Security," Caviglione *et al.* measure energy consumption of popular cryptographic algorithms with different parameters (i.e., key length, loads, and operation modes) and scenarios (i.e., end nodes and network devices). From the measurements, they provide insights; for example, software optimization could play a major role in energy savings.

In the article "On Understanding the Existence of a Deep Torrent," Rodriguez-Gomez *et al.* suggest a new concept of "Deep Torrent" that indicates torrents that are available in BitTorrent but cannot be found by means of public websites or search engines. They show by the measurement of their crawler that the estimated size of Deep Torrent is 67 percent of the total number of resources shared in the BitTorrent network.

Finally, the article "Toward Stream-Based IP Flow Analysis" discusses stream-based IP flow analysis, in which IP flows are processed and analyzed in data streams immediately after an IP flow is observed. The authors then explain how this approach can benefit real-time network security analysis and improve situational awareness.

We are confident that readers will enjoy this Feature Topic and will find the articles interesting. In addition, we also hope that the presented results will stimulate further research in this important area of information and network security.

We would like to express our thanks for the support and help of Osman Gebizlioglu, Editor-in-Chief of *IEEE Communications Magazine*, Joseph Milizzo of the ComSoc staff, the leading researchers contributing to the Feature Topic, and the excellent reviewers.

Biographies

Wojciech Mazurczyk [M'11, SM'13] received his M.Sc., Ph.D. (Hons.), and D.Sc. (habilitation) degrees in telecommunications from the Warsaw University of Technology (WUT), Poland, in 2004, 2009, and 2014, respectively. He is currently an associate professor with the Institute of Telecommunications at WUT and a researcher at Fern Universität, Hagen, Germany. His research interests include network security, bioinspired cybersecurity and networking, and information hiding. Since 2013, he has been an Associate Technical Editor of *IEEE Communications Magazine*.

Maciej Korczyński holds a Ph.D. (2012) in computer science from the University of Joseph Fourier, France. He is currently a post-doctoral researcher in the Economics of Cyber Security research group at Delft University of Technology. His main research interests include information and network security, large-scale passive and active Internet security measurements, economics of cyber security, incident data analysis, anomaly and attack detection, bio-inspired cyber security, encrypted traffic classification, and security of TLS and DNS protocols. His personal website can be found at http://mkorczynski.com.

Koji Nakao received his B.E. degree in mathematics from Waseda University in 1979. Since joining KDDI in 1979, he was engaged in the research on information security technology for KDDI until 2015. In 2004, he started also working in NICT to research and develop cyber security technologies. His present position is Distinguished Researcher, and he manages research activities for NICT. He has also been a visiting professor of Yokohama National University.

Engin Kirda is a professor at the College of Computer and Information Science and the Department of Electrical and Computer Engineering of Northeastern University, Boston, Massachusetts. He is also the director of the Northeastern Information Assurance Institute. He is an uthor of over 140 scientific papers. Heis interested in network security with focus on web security, binary analysis, and malware detection. He has been a TPC member of refereed conferences, including ACM CCS, USENIX Security, IEEE Security and Privacy, WWW, NDSS, RAID, ESORICS, and DSN. His personal website can be found at http://www.ccs.neu.edu/home/ek.

Cristian Hesselman directs SIDN Labs, the research team of the operator of the .nl top-level domain, SIDN. His research interests include Internet security, stability, and privacy measurements, threat detection, and collaborative security. He is a member of SIDN's leadership team and a member of the Security and Stability Advisory Committee at ICANN. He holds a Ph.D. (2005) and an M.Sc. (1996) in computer science from the University of Twente, the Netherlands. His company site can be found at www.sidnlabs.nl; his personal site can be found at www.hesselman.net.

Katsunari Yoshioka holds a Ph.D. (2005) in computer engineering from Yokohama National University, Japan. He is an associate professor at the Graduate School of Environment and Information Sciences, Yokohama National University. His research interests cover a wide range of information security, including malware analysis, network monitoring, and intrusion detection.

Traffic Measurements for Cyber Security

Demystifying DDoS as a Service

Ali Zand, Gaspar Modelo-Howard, Alok Tongaonkar, Sung-Ju Lee, Christopher Kruegel, and Giovanni Vigna

The authors present a measurement study of 17 different DaaS providers, in which they analyzed the different techniques used to launch DDoS attacks, as well as the infrastructure leveraged in order to carry out the attacks. Results show a growing market of short-lived providers, where DDoS attacks are available at low cost (tens of dollars) and capable of easily disrupting connections of over 1.4 Gb/s.

Abstract

In recent years, we have observed a resurgence of DDoS attacks. These attacks often exploit vulnerable servers (e.g., DNS and NTP) to produce large amounts of traffic with little effort. However, we have also observed the appearance of application-level DDoS attacks, which leverage corner cases in the logic of an application in order to severely reduce the availability of the provided service. In both cases, these attacks are used to extort a ransom, to hurt a target organization, or to gain some tactical advantage. As it has happened for many of the components in the underground economy, DDoS has been commoditized, and DDoS as a service (DaaS) providers allow paying customers to buy and direct attacks against specific targets. In this article, we present a measurement study of 17 different DaaS providers, in which we analyzed the different techniques used to launch DDoS attacks, as well as the infrastructure leveraged in order to carry out the attacks. Results show a growing market of short-lived providers, where DDoS attacks are available at low cost (tens of dollars) and capable of easily disrupting connections of over 1.4 Gb/s. In our study, particular attention was given to characterize application-level (HTTP) DDoS attacks, which are more difficult to study given the low volume of traffic they generate and the need to study the logic of the application providing the target service.

Introduction

Distributed denial of service (DDoS) attacks have been a problem on the Internet for more than 15 years. However, the recent increase in the number of DDoS attacks and in the amount of traffic that they generate has attracted the attention of the media, the industry, and the research community alike. This new wave of attacks exploit asymmetries in vulnerable services to generate large amounts of traffic or use large amounts of resources with relatively little effort from the attacker. For example, misconfigured Network Time Protocol (NTP) services can be leveraged to generate gigabytes of data with a simple spoofed request. This generated traffic exhausts the bandwidth available at the target. We call this type of (more traditional) attack an *extensive* DDoS.

However, there is another type of DDoS attack in which the lack of availability of a resource is due to the fact that a single interaction with the target requires an unusually high amount of resources in order to be processed. For example, on a web site, there might be a search form that, when provided with certain values, might require an extremely large database query that slows the whole website to a crawl. We call this kind of attack an asymmetric application-level or *intensive* DDoS.

While extensive DDoS attacks have been studied for quite a while [1] and some remediation has been provided (e.g., coordinated filtering managed by blacklists, rate limiting, patching of vulnerable services), intensive DDoS attacks have not received the same level of attention. The latter is more difficult to characterize because they often depend on the logic of the application providing the target service. In addition, these attacks do not rely on large volumes of data and therefore can go undetected by volumetric detection mechanisms. Finally, since the attacker communicates with the service following the service protocol, the attacker's requests are similar to a legitimate request and hence more difficult to filter out.

As both extensive and intensive DDoS attacks become an integral part of the efforts of cybercriminals to obtain financial gains (e.g., by blackmailing organizations under attack or by obtaining a tactical advantage in time-sensitive settings), the provision of DDoS service has become commoditized. We now see the rise of DDoS as a service (DaaaS) offerings, in which DDoS providers attack a target in exchange for money.

Background

In this section we introduce the different types of DDoS attacks available, as well as the basic infrastructure of the DaaS providers, which are the subject of our study.

Types of DDoS Attacks

A DDoS attack can be extensive or intensive. An extensive attack relies on high volumes of traffic that by itself is harmless. A malicious actor needs a considerable amount of resources to successfully execute an extensive attack, as it is costly to generate enough traffic volume to impact a large target. Examples of these attacks include SYN flood, UDP flood, reflected Domain Name Service (DNS), and reflected NTP.

In most extensive attacks, miscreants may use a technique called amplification. Leveraging amplification, the attacker continuously abuses a

Digital Object Identifier:
10.1109/MCOM.2017.1600980

Ali Zand, Christopher Kruegel, and Giovanni Vigna are with the University of California, Santa Barbara; Gaspar Modelo-Howard is with Symantec; Alok Tongaonkar is with RedLock; Sung-Ju Lee is with KAIST.

0163-6804/17/$25.00 © 2017 IEEE

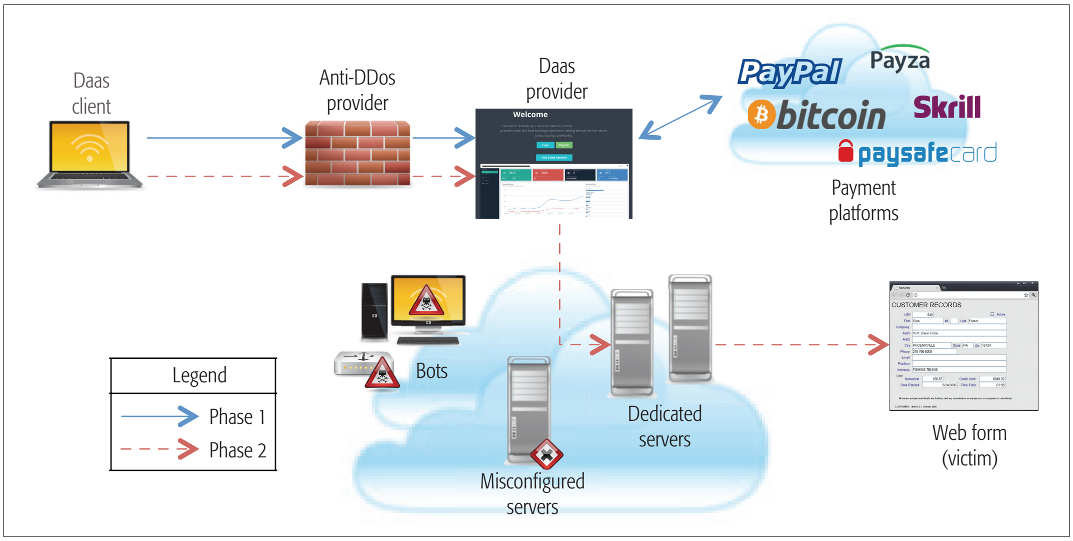


Figure 1. Infrastructure used by DaaS providers, including the payment platforms employed (phase 1) and the set of resources to launch the selected DDoS attack (phase 2). Intensive attacks predominantly utilize dedicated hosts with high bandwidth.

Given the shady nature of the business, DaaS providers are not particularly dependable services. In our study, we found them to have a short life span (compared to legitimate online services), measured in weeks to months. Of the 17 providers identified and tested, only 7 were functional at the end of our three-month evaluation.

set of hosts that responds to a request with a considerably larger response that is delivered to the destination of the attacker's choosing. Previous studies have shown that this amplification factor differs according to the used protocol and can be as high as 4670×. These types of attacks have achieved throughputs as high as 500 Gb/s and affected enterprises with large infrastructures such as Sony PlayStation Network, Cloudflare, and several U.S. banks.

Intensive attacks, on the other hand, target specific weaknesses in a target application. Any request (or request access pattern) that takes a considerably larger amount of resources on the server than the client can be leveraged to perform this attack. These vulnerabilities can be due to problems like memory leaks and long running processes that never free their resources. Most cases of intensive attacks target HTTP servers, given their popularity on the Internet. Examples include submitting data to web forms found on the victim server, at very slow rates (one byte at a time), and opening multiple connections that are kept alive by sending partial packets. These examples have been implemented by the *R-U-Dead-Yet?* (*RUDY*) and *Slowloris* tools [2], respectively. Also worth noting is that intensive attacks only send legit packets, not malformed ones, making the resulting traffic appear legitimate, complicating their detection by security systems.

Basic Scenario for a DDoS as a Service Providers

The continued rise of DDoS attacks as a way to target the online presence of organizations can be attributed to several factors. One possibility is that these attacks are often conducted through botnets, which often encompass thousands of computers. Pools of vulnerable computers are always available, given the constant discovery of software bugs.

Another possible factor for the rise of DDoS attacks is the commoditization phenomenon that these types of attacks have seen in the last few years. A large number of DaaS providers are available on the Internet, providing cheap access to both extensive and intensive DDoS attacks. Using a subscription-based model, the providers' fees range between $2 and $15 for basic packages. They support different payment mechanisms, ranging from traditional online systems like PayPal to the Bitcoin electronic currency and anonymous payment systems like Paysafecard. The basic packages allow launching attacks for 60–90 s and currently produce attack volume peaking at more than 1.4 Gb/s. More expensive packages are also available, which provide longer attack periods and subscription terms. The same sets of extensive and intensive DDoS attacks are available for all subscription packages.

Figure 1 shows a diagram of the infrastructure used by DaaS providers to offer their *pay, point, and click service*. The diagram includes the payment platform used (phase 1, *pay*), as well as the components used by the providers to launch a DDoS attack (phase 2, *point and click*). As shown in the diagram, intensive attacks are launched using dedicated servers, since only a small set of hosts is required and software needs to be installed to interact with the logic of the web application under attack. Botnets and misconfigured hosts are commonly used when launching the volumetric, extensive attacks.

A common trait found in DaaS providers is the usage of anti-DDoS service providers to protect their web platforms. As many of them claim to be only used to stress test the resources owned by a customer, the providers include DDoS protection mechanisms in their infrastructure.

Given the shady nature of the business, DaaS providers are not particularly dependable services. In our study, we found them to have a short life span (compared to legitimate online services), measured in weeks to months. Of the 17 providers identified and tested, only 7 were functional at the end of our three-month evaluation. Additionally, those providers that were functional delivered an average of only 44 percent of the offered services. We also found several systems provided intermittent service.

There are multiple risk factors associated with studying cyber-miscreants. To deal with these factors and to develop the ethical framework for our experiments, we followed the ethical guidelines for computer security research defined in The Menlo Report and consulted previous work where researchers actively interacted with systems or networks used by cyber-miscreants.

DaaS/run	1	2	3	4
APO	2	–	90	2289
BIG	90	415	61	170
DAR	4256	–	–	–
DES	38,194	11,889	20,922	10,727
DIV	–	4	8	–
GRI	20,752	–	–	–
HAZ	–	1	2	1
IDD	–	4	2	64
ION	5	4	4	14,118
IPS	2284	–	–	–
NET	1776	1854	1556	982
POW	2759	3727	3723	–
QUA	8132	–	–	–
RAG	30,505	4018	4	3
RES	8499	–	–	–
TIT	21,609	2274	3501	8238
WRA	7219	6891	11,699	95

Table 1. Traffic generated by each DaaS (MB).

The DDoS as a Service Landscape

Methodology

We identified 28 different DaaS providers for our study, from visiting multiple hacking sources: forums, blogs, mailings lists, and news sites. A user account was then created on each of the 28 providers. After reviewing the corresponding websites, 17 were determined to be operational. The other 11 failed to provide a working service interface. We later realized that this failure rate is the result of the common short and intermittent life span experienced by DaaS providers (usually weeks to months). For example, 12 out of the 17 providers were available since the start of our investigation, while the other 5 became active later in the process.

Using each of the 17 operational providers, we investigated the DaaS ecosystem from both sides of the attack.

As a DaaS Customer: After registering on the website of each provider, their services were bought for a limited time, selecting the cheapest services available on each website. The prices varied from $2 to $15. We studied the different functionalities provided on these websites to help determine how their advertisement, payment systems, and business aspects work. Additionally, our analysis also included a look at their offered attack capabilities.

As a DDoS Victim: We set up a machine to serve as a target of DDoS attacks and ordered each provider to launch the strike against it. The victim machine was an Ubuntu Linux machine with 8 GB of RAM, 1 TB of SSD disk space, dual-core Intel processor, an optical fiber network connection of 10 Gb/s to the Internet, running an Apache web server with MediaWiki software, and hosting a clone of a university's department website. The machine was connected to the Internet through a dedicated link that allowed isolation of our tests from the rest of the university campus network and prevented it from being negatively affected. We captured all the traffic aimed at our victim machine, its responses, and its internal state during the attacks.

Each DaaS was tested four times over a period of three months, from May to July 2014. In each of the four runs, we tested all the attack types offered by each of the working DaaS and captured all the resulting traffic. At all times during the testing, we ran only one type of attack from a single DaaS. Also, to prevent late packets from one attack from being mixed with the next, we waited for 100 s between consecutive attacks.

Ethical Considerations

There are multiple risk factors associated with studying cyber-miscreants. To deal with these factors and to develop the ethical framework for our experiments, we followed the ethical guidelines for computer security research defined in the Menlo Report [3] and consulted previous work where researchers actively interacted with systems or networks used by cyber-miscreants [4, 5].

To reduce the risk of financing possible cyber-miscreants during our experiments, we purchased the cheapest services from the DaaS providers. This meant a single DaaS provider received no more than $45, as we repeated the experiments three times on the most expensive ($15) service used.

Another risk factor for studies such as ours is to unwittingly and negatively affect other victims. In this case, the victims can be compromised machines used by the providers to launch the DDoS attacks or other machines and networks on the path of the attack that are affected by the amount of generated traffic. To mitigate the potential risks, our experiments included conditions to restrict the duration and intensity of the attacks, limit the path of the attack traffic, and coordinate the experiments with the system administrators of our campus networks.

As mentioned before, we ran each attack for only 60 s to limit the impact of each attack. In addition, the target machine used to receive the attacks was located on an isolated subnet of our campus network and connected to a dedicated 10 Gb/s link so that the traffic generated during the tests would not affect other subnets (and their hosts) on campus. We also ran all high traffic tests during weekend nights to further reduce impacting network bystanders.

We acquired the campus network administrators' permission to run our tests before proceeding, agreed on a schedule, and established a contingency plan in case an undesirable situation happened. We followed up with the network administrators after each round of experiments and confirmed with them that an experiment had not negatively affected other parts of the campus network before proceeding with the next round.

Finally, it should be mentioned that our research was out of scope of the institution-al review board (IRB) committee given that the experiments with DaaS providers did not include any type of direct or indirect experiments with human beings.

Results for DaaS Providers

The four test runs generated around 255 GB of traffic and more than 94.1 million packets. The top four protocols (DNS, CHARGEN, Simple Network Management Protocol [SNMP], and NTP) produced 91.3 percent of the total traffic generated. DNS was the top traffic contributor with 71.07 GB, while NTP was the top packet generator with 34.9 million packets. Attacks using HTTP only produced 0.71 GB from 4.72 million packets.

Table 1 shows the amount of traffic generated by each DaaS during a run. Those providers that were not active in a run are shown with a dash (—). Results showed that 10 to 14 DaaS were active in a single run and that traffic generated varied among the different providers. For example, the RAG[1] and DES DaaS generated 30.5 and 38.2 GB each in run 1, while APO and ION only produced 2 and 5 MB. Out of the 47 tests that produced traffic across the four different runs, 26 (55 percent) produced at least 1 GB.

The functionalities provided by different DaaS providers differ greatly in terms of their claimed and actual attack types provided. Table 2 shows the offered attack capabilities of each DaaS. In this table, each row is a type of attack, and each column represents a DaaS. A checkmark (✓) indicates that the feature was offered and indeed worked during the experiments. An (×) means the feature was offered but did not work for any test run. A blank space means that the feature was not offered.

A total of 28 different attack methods were identified across the 17 DaaS providers under evaluation. Out of these attack methods, 17 were extensive DDoS attacks, 7 were intensive, and 4 never worked. Of these seven intensive attacks, we found that some of the tools used by the providers to launch these attacks targeted different web server implementations. For example, the *Apache Remote Memory Exhaustion* (ARME) tool is only effective against Apache servers, as the name implies, while the Slowloris tool targets Apache, HTTPd, and GoAhead web servers. As observed in our experiments, both tools send partial,legitimate packets to keep connections open and do not generate large volumes of traffic compared to extensive attacks.

Table 3 present the number of completed TCP connections to the victim, the number of unique non-spoofed IP addresses, and the maximum observed throughput for the DaaS producing the largest traffic.

DaaS Infrastructure for Intensive Attacks

To characterize the machines and networks used by the DaaS providers to launch their intensive attacks, we first determined the non-spoofed IP addresses that initiated the attacks. An address was labeled non-spoofed if at least one complete TCP connection was established with our victim server during the test, which provided a lower bound of the actual situation. Among all (intensive and extensive) attack traffic observed, only 0.71 percent was associated with non-spoofed addresses, an expected result given the usual incognito nature of extensive attacks and the considerably larger traffic they produce.

Using the technique described above, a total of 26,271 non-spoofed IP addresses were identified in all the attacks launched to our victim server and across the five providers that successfully produced the attacks. As shown in Table 4, the number of IP addresses used by a DaaS varied from 35 (TIT) to 21,809 (WRA). The low number of addresses for TIT was a sign of the DaaS soon to go offline, as the service stopped after our second run. WRA, on the other hand, consisted of a large botnet, primarily composed of compromised or misconfigured WordPress web servers. WRA was also the only provider to successfully produce six different types of intensive attacks (GET and POST floods, ARME, Slowloris, RUDY, and XML-RPC pingback) and worked for all four runs.

IP2Location [6] was consulted to determine the geographical information of the IP addresses, their autonomous system number (ASN), and the type of networks to which they were connected. As IP2Location provides various degrees of geolocation accuracy, we limited our analysis to using country and region (state in the United States) information in order to determine the location of addresses. Additionally, we used their classification of subnets and ASNs to label the IP addresses as part of one of the following three types of networks: *broadband/residential*, *commercial hosting providers*, and *other*.

Results show DaaS with different geographical extensions and mixtures of types of machines. The United States and China were the largest sources of machines for the providers, with the United States providing at least 55 percent of the machines in the cases of WRA, DES, and BIG. China was the largest source for RAG and TIT, providing at least 39 percent of the attacking hosts. RAG presented a larger number of countries hosting machines and associated ASNs than BIG, even though they both had similar numbers of IP addresses. 81 percent of the addresses used by RAG were in 10 different countries, and 74.1 percent were connected to broadband networks. In comparison, BIG had 81 percent of its machines located in one country (United States) and 128 addresses (93.3 percent) are connected to networks identified for hosting. Moreover, 85 of those addresses were attributed to a single data center in Arizona. We experienced more effective (able to leave our server unresponsive) and reliable (available through all runs) attacks by using BIG than when launching attacks through RAG, which not surprisingly suggests that machines in hosting networks might be more valuable for DaaS than in those in broadband networks.

After identifying the addresses with at least a complete TCP connection in the intensive attacks, we knew that the attacker's machine either had that IP address, or went through a proxy or VPN using that address. To determine each case, we scanned the IP address actively and also fingerprinted the host passively, as both approaches complement each other. An active scan interacts with the target host by sending a predefined set of packets and determining the type of the host based on its response. As such, this approach allows identifying when a proxy is used. In con-

Our findings show that 81.5 percent of the non-spoofed IP addresses belonged to Linux machines and 12.5 percent to Windows hosts; the rest of the machines were not identified. The high occurrence of Linux hosts and non-spoofed IP addresses suggests that the DaaS providers depended on machines that use popular OSs, such as dedicated servers and Internet of Things devices, to successfully launch attacks.

[1] Throughout this article, each DaaS provider is referred to by a three-letter code in order to keep its real name anonymous and avoid publicizing its service. For example, a DaaS named GeneralTester could be referred to as GRL.

Attack/DaaS	APO	BIG	DAR	DES	DIV	GRI	HAZ	IDD	ION	IPS	NET	POW	QUA	RAG	RES	TIT	WRA	No. DaaS
Extensive attacks																		
UDP	(×)		✓	✓	(×)	✓	(×)	(×)	(×)	✓	✓			✓		✓		7/12
Home Conn.				✓													(✓)	1/2
XSYN	(×)			✓											(×)		(×)	1/4
SSYN	(×)		(×)	✓					(×)	(×)	✓		✓	✓	(×)		✓	5/10
SSDP			✓										✓		✓			1/1
ESSYN	(×)				(×)			(×)						✓		✓	✓	3/6
ZSSYN																		1/1
NUDP (Net BIOS)				✓														1/1
SUDP (SNMP)		✓		✓	(×)													2/3
Website				✓														1/1
XBOX Live				✓														1/1
DNS					(×)		(×)								✓		✓	2/4
CHARGEN	(×)				(×)			(×)				✓	(×)	✓				2/6
NTP					(✓)									✓	✓		✓	4/5
TCP Amp.																	✓	1/1
RUDP								(×)										1/2
UDPLAG	(×)		✓		(×)			(×)	(×)	✓	✓	(×)	(×)	✓		✓	✓	8/14
Intensive attacks																		
POST				(×)		(×)			(×)		(×)			✓	(×)		✓	2/7
HEAD				(×)		(×)			(×)		(×)			✓	(×)		(×)	1/7
GET				(×)		(×)			(×)		(×)			✓	(×)		✓	2/7
ARME				(×)		(×)			(×)		(×)			✓	(×)		✓	2/7
SLOWLORIS				✓		(×)			(×)		(×)			(×)	(×)	✓	✓	3/8
RUDY			(×)	(×)					(×)	(×)	(×)			(×)	(×)	✓	✓	2/9
XML-RPC		✓	(×)	✓	(×)	(×)					(×)		(×)		(×)		✓	3/9
Not working																		
Source Engine											(×)							0/1
KS												(×)						0/1
Joomla			(×)															0/1
OVH						(×)												0/1
No. Attacks	0/6	2/2	3/7	10/17	0/8	5/12	0/2	0/5	0/9	2/4	4/11	1/3	2/5	10/12	3/12	5/5	12/15	

Table 2. Attack methods offered by each DaaS provider tested.

trast, a passive fingerprinting method observes the traffic originating from the target host and determines its type by looking for patterns that identify a particular operating system or application.

Our findings show that 81.5 percent of the non-spoofed IP addresses belonged to Linux machines and 12.5 percent to Windows hosts; the rest of the machines were not identified. The high occurrence of Linux hosts and non-spoofed IP addresses suggests that DaaS providers depended on machines that use popular OSs, such as dedicated servers and Internet of Things devices, to successfully launch attacks. In terms of proxies used by the providers, we found that they

	Number of connections/number of unique IP addresses				Max. attack size (Mb/s)/run
DaaS/run	1	2	3	4	
BIG	20,408/127	7076/85	6625/39	2314/50	84.65/2
DES	–/–	–/–	76,483/9409	51/1	690.18/2
RAG	4226/168	1665/168	–/–	–/–	852.49/1
RES	7523/527	–/–	–/–	–/–	1494.05/1
WRA	55,077/459	89,728/271	71,819/278	51,564/21,573	579.84/2

Table 3. Number of connections and unique IP addresses for top traffic generating DaaS per run.

DaaS providers offered multiple subscription options for their services, at different prices. For ten providers, a higher price only means longer period of attack and longer-term subscriptions. In other words, they did not offer additional attack methods or an increase in the intensity of the attacks.

employed proxies in very small numbers, as only 0.76 percent of the non-spoofed addresses were identified as proxies, anonymizing VPN service or TOR exit node. IP2Location also provided information on addresses identified as proxies, validating 92 percent of our results.

Through the four runs of experiments launching intensive attacks, we found few cases of IP address sharing among providers. Most did not share any addresses, and in the cases were they did, it was in very low numbers (1 to 5 addresses). This suggests the appropriation or exclusive control of the machines by each DaaS. WRA was the only exception to this, sharing 5223 addresses with DES, thanks to exploiting a high-risk vulnerability [7] on WordPress servers that was publicly reported during our runs. The vulnerability did not provide a mechanism for attackers to control who could exploit these servers, thus leaving the opportunity for sharing.

Table 5 shows the number of IP addresses reused by BIG and WRA during our experimental runs, as these were the only providers that generated non-spoofed traffic in all four executions. The diagonals in the table show (in bold italic) the total number of IP addresses used by each DaaS in a single run. From our experiments, both providers had to continuously add new machines to their networks, as many of the IP addresses from an attack execution would not be found in the next. As an example, BIG showed 122 addresses in the first run, but only 66 (54 percent) of those would be present in the second run. The attacker needs to constantly find new machines, which is not always trivial. From the second to the third run, BIG went from 82 to 37 IP addresses, and only two of those were new. In the case of WRA, the 21,573 different addresses found in the fourth run correspond to web servers exhibiting the high-risk vulnerability to WordPress, as discussed above.

Operational Stability

Given the shady nature of their business, DaaS providers are not particularly dependable services. Our study found them to have a short life span (compared to legitimate online services), measured in weeks to months. This was supported by the fact that 11 of the 28 DaaSs identified failed to provide any service, while several of the other DaaSs briefly disappeared during the different executions. Only seven of the 17 DaaS were functional for all four runs, while four were successfully used in three runs and one DaaS was available in two runs. Additionally, 3 of the 11 providers that were not working when we first accessed them started working after three months.

13 out of the 17 tested providers claimed to support intensive DDoS attacks, but when we tested them, only five successfully executed one or more types of application layer DDoS attacks. Out of the 17 DaaS providers tested, only 7 were still working after we finished our study.

Payment Methods

The most popular payment methods used by the DaaS providers were the popular online payment system PayPal and the Bitcoin digital currency. Other methods found included the payment platforms Google Wallet, Paysafecard (which allows anonymous transfers), Payza (transfers using email), and Skrill (focused on low-cost transfers). During the tests, three of the providers had their Paypal accounts deactivated and could not receive money.

DaaS providers offered multiple subscription options for their services at different prices. For 10 providers, a higher price only means a longer period of attack and longer-term subscriptions. In other words, they did not offer additional attack methods or an increase in the intensity of the attacks.

We evaluated GRI, one of the four providers that claimed better throughput and additional methods of attacks, to observe the difference between the cheap and more expensive options. This DaaS was chosen as it offered the most powerful attack, and in terms of throughput, pricing was cheaper than other DaaS ($50, compared to up to $300 in the case of RAG), and offered a different class of attack. Results show that the more expensive service gives access to two VIP servers (servers that regular accounts do not have access to) at the same time (and therefore able to execute two concurrent attacks). The amount of traffic generated and the list of offered attacks by each VIP server were not different from its cheap service.

Related Work

Research on the analysis of existing DDoS attack vectors [8–11] has focused on the resources available on the Internet that can be used to launch DDoS attacks. Particularly, researchers have studied the amplification effect produced from using certain network services on the impact from using botnets to create DDoS attacks. Our work complements previous research by providing an unabridged analysis of the new vector available to attackers: application-level, intensive DaaS.

DaaS	Total No. IP addresses	No. countries	No. ASNs	Type of network			No. proxies found	Additional information
				Broadband	Hosting	Other		
BIG	165	20	40	6.7%	93.3%	0.0%	0	U.S. hosts 81.8% of all addresses, while next four countries account for 8.5%
DES	9405	88	1446	11.8%	84.8%	0.4%	11	U.S. hosts 61% of all addresses, followed by 10 countries with more than 100 addresses each
RAG	162	36	84	74.1%	6.8%	19.7%	58	China accounts for 39.5% of all addresses, while Brazil, Indonesia, Rusia, and Guatemala together host 27.16%
TIT	35	10	22	45.7%	48.6%	5.7%	0	China and U.S. host 45% and 22.9%, respectively
WRA	21,809	117	3075	20.12%	79.82%	0.06%	130	U.S. accounts for 55.1% of all addresses, while 19 other countries host at least 140 addresses

Table 4. Geographical distribution of the IP addresses for each of the DaaS providers that generated intensive attacks. The table also includes for each provider: the number of ASNs involved, the type of network to which the addresses where connected, and the number of proxy servers identified.

	Big				WRA			
Run/run	1	2	3	4	1	2	3	4
1	***122***	66	35	22	***426***	176	176	157
2	–	***82***	35	20	–	***269***	184	163
3	–	–	***37***	17	–	–	***277***	170
4	–	–	–	***49***	–	–	–	***21,573***

Table 5. Number of non-spoofed IP addresses reused, per run, for BIG and WRA. Values in the diagonal (shown in bold italic) represent the total number of IP addresses used to launch intensive attacks in each run.

Rossow [10] studied several UDP-based services available on the Internet that can be misused for amplification during a DDoS attack, showing that they are numerous and easy to find on the Internet, and providing a byte amplification factor of up to 4670. Kührer *et al.* [9] showed the possibility of using various TCP servers as reflective traffic amplifiers, and measured their possible impact. Czyz *et al.* [8] studied the temporal properties of reflectors, especially from NTP servers, while Rijwijk-Deij *et al.* [11] showed that a byte amplification factor of over 102 is possible by abusing the DNSSEC extensions.

Recent work [12, 13] has also looked at the rising threat of DaaS providers. We consider all previous studies complementary to ours, as they did not analyze the application-level, intensive DDoS attacks that can be launched from these providers, as done in our study. Karami *et al.* [12] only evaluated the infrastructure used for extensive attacks, while Santanna *et al.* [13] limited the study to extensive attacks using the DNS or CHARGEN protocols. Noroozian *et al.* [14] profiled the victims of extensive attacks launched by DaaS providers by using a network of honeypots running open services to launch amplification attacks. The study found that 88 percent of the victims were housed in broadband and hosting ISP networks, while the ICT development and GDP per capita of the host countries also help explain the victimization rate.

Conclusions

With the goal of demystifying the newly prevalent class of DaaS providers, we identified and studied 28 of these online systems. Given the short life of many of the providers found, we analyzed the behavior of 17 over a period of three months. Results show DaaS providers commonly offer both extensive and intensive DDoS attacks, and over different protocols. Customers only have to spend tens of dollars to have access to the attacks, which we were able to use to launch 1-minute attacks that generated 255 GB of traffic and were able to achieve throughput of 1.4 Gb/s, at a cost of tens of dollars.

In our study, we showed that many of these publicly accessible providers allow users to launch intensive attacks, hence the need to also study this increasingly popular threat. Results show that these providers pose a real threat to web servers on the Internet as they have access to networks of up to tens of thousands of machines to generate traffic that looks inconspicuous but leaves the servers unresponsive.

References

[1] R. Chang, "Defending against Flooding-Based Distributed Denial-Of-Service Attacks: A Tutorial," *IEEE Commun. Mag.*, vol. 40, no. 10, Oct. 2000, pp. 42–51.

[2] E. Cambiaso *et al.*, "Slow DoS Attacks: Definition and Categorisation," *Int'l. J. Trust Management in Comp. and Commun.*, vol. 1, no. 3-4, Jan. 2013, pp. 300–19.

[3] D. Dittrich and E. Kenneally, "The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research," U.S. Dept. Homeland Sec., Aug. 2012.

[4] C. Kanich *et al.*, "Spamalytics: An Empirical Analysis of Spam Marketing Conversion," *Proc. 15th ACM Conf. Comp. Commun. Sec.*, Oct. 2008, pp. 3–14.

[5] B. Stone-Gross *et al.*, "Your Botnet Is My Botnet: Analysis of a Botnet Takeover," *Proc. 16th ACM Conf. Comp. Commun. Sec.*, Nov. 2009, pp. 635–47.

[6] IP2Location, commercial IP geolocation databases, Jan. 2015; http://www.ip2location.com/databases/, accessed Jan. 5, 2015.

[7] Symantec, "Security Focus: WordPress Slider Revolution Responsive Plugin 'img' Parameter Arbitrary File Download Vulnerability," July 2014; http://www.securityfocus.com/bid/68942, accessed Sept. 13, 2014.

[8] J. Czyz *et al.*, "Taming the 800 Pound Gorilla: The Rise and Decline of NTP DDoS Attacks," *Proc. ACM SIGCOMM Conf. Internet Measurement*, Nov. 2014, pp. 435–48.

[9] M. Kührer *et al.*, "Hell of a Handshake: Abusing TCP for Reflective Amplification DDoS Attacks," *Proc. 8th USENIX Wksp. Offensive Technologies*, Aug. 2014.

[10] C. Rossow, "Amplification Hell: Revisiting Network Protocols DDoS Abuse," *Proc. Network Distrib. Sys. Sec. Symp.*, Feb. 2014.

[11] R. van Rijswijk-Deij, A. Sperotto, and A. Pras, "DNSSEC and Its Potential for DDoS Attacks," *Proc. ACM SIGCOMM Conf. Internet Measurement*, Nov. 2014, pp. 449–60.

[12] M. Karami, Y. Park, and D. McCoy, "Stress Testing the Booters: Understanding and Undermining the Business of DDoS Services," *Proc. 25th Int'l. World Wide Web Conf.*, Apr. 2016, pp. 1033–43.

[13] J. Santanna *et al.*, "Booters: An Analysis of DDoS-as-a-Service Attacks," *Proc. IFIP/IEEE Int'l. Symp. Integrated Network Mgmt.*, May 2015, pp. 243–51.

[14] A. Noroozian *et al.*, "Who Gets the Boot? Analyzing Victimization by DDoS-as-a-Service," *Proc. Int'l. Symp. Research Attacks, Intrusions, Defenses*, Sept. 2016, pp. 368–89.

Biographies

Ali Zand (zand@cs.ucsb.edu) received his Ph.D. in 2015 from the University of California Santa Barbara, working on system security research with a focus on cyber situation awareness. His research interests include automatic service dependency detection, automatic asset protection prioritization, botnet C&C signature generation, cyber situation awareness measurement, DDoS attack studies, and social media spam detection.

Gaspar Modelo-Howard [SM] (gaspar@acm.org) is a senior principal data scientist in the Center for Advanced Machine Learning at Symantec. His research interest are computer and network security, with a focus on web security, intrusion detection and response, and malware detection. He is also an adjunct professor in computer security at Universidad Tecnológica de Panamá. He is a member of ACM and Usenix.

Alok Tongaonkar (alok@redlock.io) is head of Data Science at RedLock. Previously, he was a data scientist director leading the Center for Advanced Data Analytics at Symantec. He has a Ph.D. in computer science from Stony Brook University, New York. His research focuses on application of machine learning and big data technologies for developing innovative security, networking, and mobile app analytic products. He has been granted multiple patents by USPTO. He is a Senior Member of ACM.

Sung-Ju Lee [F] (sjlee@cs.kaist.ac.kr) is an associate professor and an Endowed Chair Professor at the Korea Advanced Institute of Science and Technology (KAIST). He received his Ph.D. in computer science from the University of California, Los Angeles and spent 15 years in the industry in Silicon Valley before joining KAIST. His research interests include computer networks, mobile computing, network security, and HCI. He is a recipient of multiple awards, including the HP CEO Innovation Award and the Test-of-Time Paper Award at ACM WINTECH 2016. He is an ACM Distinguished Scientist.

Christopher Kruegel (chris@cs.ucsb.edu) is a professor in the Computer Science Department at the University of California, Santa Barbara and one of the co-founders of Lastline, Inc., where he serves as the chief scientist. His research interests include most aspects of computer security, with an emphasis on malware analysis, web security, and intrusion detection. He is a recipient of the NSF CAREER Award, MIT Technology Review TR35 Award for young innovators, and IBM Faculty Award.

Giovanni Vigna [SM] (vigna@cs.ucsb.edu) is a professor in the Department of Computer Science at the University of California, Santa Barbara and the CTO at Lastline, Inc. His research interests include malware analysis, vulnerability assessment, the underground economy, binary analysis, web security, and mobile phone security. He leads the Shellphish hacking group, which has participated in more DEF CON CTF competitions than any other group in history. He is a Senior Member of ACM.

Cybercrime at a Scale: A Practical Study of Deployments of HTTP-Based Botnet Command and Control Panels

Aditya K Sood, Sherali Zeadally, and Rohit Bansal

Cybercriminals deploy botnets for conducting nefarious operations on the Internet. Botnets are managed on a large scale and harness the power of compromised machines, which are controlled through centralized portals known as C&C panels. In this empirical study, the authors analyzed many over 9000 C&C web URLs to better understand the deployment and the operational characteristics of HTTP-based botnets.

ABSTRACT

Cybercriminals deploy botnets for conducting nefarious operations on the Internet. Botnets are managed on a large scale and harness the power of compromised machines, which are controlled through centralized portals known as C&C panels. C&C panels are considered as attackers' primary operating environment through which bots are controlled and updated at regular intervals of time. C&C panels also store information stolen from the compromised machines as a part of the data exfiltration activity. In this empirical study, we analyzed many over 9000 C&C web URLs to better understand the deployment and the operational characteristics of HTTP-based botnets.

INTRODUCTION

Cybercriminals are using crimeware-as-a-service (CaaS) [1] by designing automated software for spreading infections across the Internet. CaaS represents an underground market business model in which illegal and unauthorized software and services are distributed by cybercriminals for illicit use on the Internet. CaaS provides automated software such as browser exploit packs (BEPs) [2–4], which are used heavily in drive-by download attacks. BEPs are well designed web-based software kits that are bundled with a number of browser exploits which are used to exploit vulnerabilities in browsers to download malware to the end-user system using drive-by download attacks [5, 6]. In this attack, the attacker coerces the end user to visit a malicious domain through a phishing attack by using social engineering tricks [7]. For example, sending a phishing email [8] with embedded links (URLs) with an important message related to banking operations. A URL consists of three primary components: the protocol name, which in the case of the web is http or https, the remote server address in the form of domain name or IP address, and the path to the resource on that server. Once the user clicks the embedded link, the associated URL is opened in the browser. The URL points to the BEP, which fingerprints the end user's browser environment against installed plugins and vulnerable components to serve the required exploit. Once the exploit is successfully executed, the end-user system is compromised, and the malware is downloaded onto the system. In this way, the end-user system is compromised through drive-by download attacks with the use of BEPs.

A bot [9] is sophisticated malware that is distributed through drive-by download attacks. When a number of bots work together, they form a botnet, which works in accordance with attackers' instructions. Botnets are managed through command and control (C&C) panels [10]. In practice, C&C panels are centralized automated software, and are used to operate and manage botnets. C&C panels are used for multiple functions, including: (1) updating the bot binary for new updates; (2) sending commands for performing targeted activities in the compromised system; (3) receiving exfiltrated data from the end-user systems; (4) installing an additional set of malware as a part of CaaS; and so on. Without C&C panels, botnets are hard to manage and operate. Hence, C&C panels play a vital role in making botnets design robust and effective.

RELATED WORKS

Kotov and Massacci [2] conducted a preliminary analysis of the source codes of approximately 30 BEPs to understand the drive-by download mechanism used to infect end-user systems. The study revealed that exploits embedded in BEPs are used in a naive way. Hue and Wang [5] also conducted a study on insecure practices that are used to deploy JavaScripts on the web by analyzing the severity and nature of approximately 6805 unique websites. The study found that websites use JavaScript in an insecure fashion that can result in security risks. Niels *et al.* [6] conducted a study to determine how drive-by download attacks are being triggered on a large scale and the relationship of user surfing habits including distribution of malware. The study revealed that 3 million malicious URLs were initiating drive-by download attacks, and the malicious URLs were also found to be listed in the Google search engine.

All these previous studies mentioned above discussed the *pre-infection* scenario to highlight how end-user systems can be infected. In this study, we focus on the *post-infection* scenario in which end-user systems have already been compromised and how the installed malware communicates with C&C panels. We analyze the different characteristics and features of the deployed C&C

Digital Object Identifier:
10.1109/MCOM.2017.1600969

Aditya K Sood is with Blue Coat Systems; Sherali Zeadally is with the University of Kentucky; Rohit Bansal is with SecNiche Security

0163-6804/17/$25.00 © 2017 IEEE

panels by examining the URLs to determine the tactics followed by the cybercriminals to evade detection and achieve successful C&C communication.

Contributions of This Work

The main research contributions of this work can be summarized as follows:

- We conducted an analytical study of over 9000 C&C URLs pertaining to HTTP-based botnets to better understand the various techniques, including communication channels and data exfiltration strategies that are used by attackers to deploy botnets.
- Our empirical study demonstrates how the HTTP-based botnets have been deployed in the last few years, including characteristics related to design and communication. The C&C URLs provide ample information about the state of botnets, which is discussed in the rest of the article.
- Finally, we highlight strategies that can be deployed for detecting and preventing botnet communications over HTTP.

Understanding HTTP-Based C&C Design and Communication Model

HTTP C&C Panel Design

We first analyze the simple design of a C&C communication channel and associated primary components as shown in Fig. 1. Generally, the complete C&C design for HTTP-based communication involves a bot (malware), a gate, and the admin panel. The gate component and admin panel together constitute the full C&C panel.

We describe the components of the C&C panel below.

Gate Component: The gate acts as a filtering component for all the incoming requests originating from the bot to the C&C. The gate component ensures that the information passed to the C&C is free from all types of anomalies. The gate implements several security checks not limited to:

- Verifying that incoming requests are coming from the registered bots; the gate performs verification using checksum value, tokens, or decryption of encrypted payload. This is part of the bot authentication process.
- Allowing downloading of configuration files by validating requests that are sent by registered bots that are authorized to do so.
- Verifying that requests are coming from an expected geographical location which has been targeted during drive-by download infections. It can be broad depending on the design. For example, if the gate is expected to receive requests from the European region, no requests from any location in Asia should be accepted by the gate.
- Implementing filtering of HTTP requests and trigger redirection if an anomaly is detected. For example, gates can redirect anomalous requests to some different location on the Internet.

Admin Component: It is the main administration panel, which is deployed to manage and control the botnets across the Internet. It is considered as an attacker's launchpad for managing all types of operations related to bots. The admin panel is used for several functions not limited to:

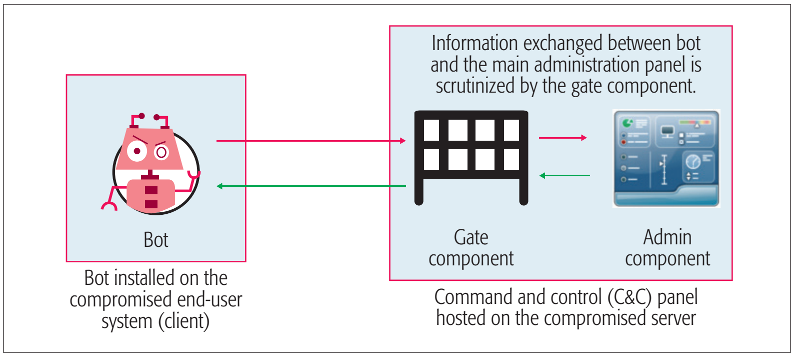

Information exchanged between bot and the main administration panel is scrutinized by the gate component.
Bot
Bot installed on the compromised end-user system (client)
Gate component
Admin component
Command and control (C&C) panel hosted on the compromised server

Figure 1. Basic design of a C&C panel.

- Managing bots installed on compromised machines, which includes operations such as sending updates, removing bots, command execution, and so on
- Storing and processing all types of exfiltrated data whether it is system information, user credentials, certificates, and so on
- Providing an interface to perform database related operations through a web panel
- Managing all the different modules [11] that are designed to enhance the functionality of bots
- Generating reports and dashboards for providing a granular view of compromised machines running the bots

HTTP-Based C&C Communication

Next, we describe how the C&C communication channel is set up between the bot installed on the end-user system and the C&C panel using HTTP. After a successful drive-by download, the end-user system is infected with a bot. Once the bot is installed in the end-user system, the following actions are performed to complete the initial communication channel.

Connecting the Bot to the C&C Panel: The bot sends back the information, such as medium access control (MAC) address, operating system details, installed browsers, and more, about the system to the C&C panel. The idea is to notify the attacker managing the C&C panel that the bot has been successfully installed on the target system. The HTTP-based C&C panel can be contacted in the following ways.

Hardcoded C&C Information: The C&C domain name or IP address is hardcoded in the bot binary. The domain or IP address is queried, and the bot sends the information back to the C&C panel.

Generating C&C Information Algorithmically: The C&C's domain names can be generated in a pseudo-random manner using domain-generation algorithms (DGAs) [12], and output domains are called algorithm generated domains (AGDs). In this technique, it is hard to determine the C&C domain name up front as it is not hardcoded; rather, a DGA is embedded in the bot binary and becomes active when the bot is run in the system. The seed value is known to the bot and the attacker for the DGA, the bot generates a number of DNS requests, and the attacker registers one of the domains. Eventually, the domain is resolved and C&C communication starts.

Cybercriminals use non-standard HTTP ports to access the C&C panel on the hosting servers. Generally, TCP port 80 is reserved for HTTP and TCP port 443 is reserved for HTTP over Secure Socket Layer (SSL). This tactic is followed to avoid detection by generic signatures that dissect the web-based communication using standard TCP ports.

Hybrid Approach to Generating C&C Information: In this approach, both hardcoded and DGA techniques are used. The idea is to implement a safe backup failover strategy in which if one technique fails, another can be used to set up a C&C communication channel. For example, if the hardcoded domain is not active, the bot can use the DGA to initiate the C&C channel.

Registering the Bot with the C&C Panel: The C&C receives the information from the bot and registers the bot as legitimate. The C&C can perform the following checks to ensure that the request has been received from a legitimate bot:

- The bot can send the hardcoded checksum value embedded in the binary itself with the first HTTP request that is verified by the C&C panel.
- The bot can also send an encrypted payload containing information that validates the legitimate nature of the bot. When this encrypted payload is decrypted by the C&C panel, it scans the information and makes sure that the bot is authorized and can be registered.
- The bot can also use a token or pre-shared key to validate itself to the C&C panel.

Updating the Bot by C&C Panel: Once the bot is registered, the C&C panel performs the next set of operations:

- The modules related to data exfiltration, infections, and so on are enabled in the C&C panel by the attacker, and the associated database tables are made active.
- A new configuration file is generated in the C&C panel based on the information sent by the bot about the environment of the compromised system.
- The configuration file is sent to the bot to update the functionalities of various components present in the bot so that associated operations can be performed.

After executing the steps above, the communication channel between the bot and the C&C panel becomes active.

C&C Panel Design and Communication Characteristics

In this section, we discuss a number of features related to the design and communication model to characterize the deployment of C&C panels for botnet operations.

IP-Address-Based C&C Communication:

- Attackers can opt to host C&C panels directly by using an IP address without registering any domain names. This strategy helps them to avoid generating any DNS traffic because no domain name resolution is queried by the system infected with the malware (bot). As a result of this, detection solutions relying on DNS traffic do not trigger any alert because the malware does not generate any DNS traffic.
- To perform direct C&C communication without domain names, the attackers need to embed (hardcode) the IP address in the malware binary itself. Generally, the IP addresses are passed as strings and can be recovered from the binary by performing reverse engineering.

Non-Standard HTTP Ports: Cybercriminals use non-standard HTTP ports to access the C&C panel on the hosting servers. Generally, TCP port 80 is reserved for HTTP, and TCP port 443 is reserved for HTTP over Secure Socket Layer (SSL). This tactic is followed to avoid detection by generic signatures that dissect web-based communication using standard TCP ports. Additionally, cybercriminals can use the shared hosting network by not altering the communication over reserved HTTP ports rather creating more web-based services on non-standard ports. Using this approach, the cybercriminals can prevent generic connections to the C&C servers.

The Onion Router Web Communication Using Reserved Top Level Domains: Cybercriminals also use The Onion Router (TOR) anonymity network, which is a hidden service that allows them to host C&C panels in a stealthy fashion. Basically, TOR is a legitimate service that allows end users to surf the Internet in an anonymous manner. For cybercriminals, it is a good choice to deploy web C&C panels that are accessible only through the TOR communication channel, which is completely encrypted and anonymous in nature. TOR makes C&C communication network traffic analysis harder for researchers. As per RFC 7686 [13], the top-level domain (TLD) reserved for TOR-based web communication is ”.onion.” A random domain name is constructed using RSA 1025 Key with SHA-1 hash algorithm [14] and “.onion” TLD which the bot needs to connect back over the TOR channel.

Encryption: SSL/Transport Layer Security for C&C Communication: Cybercriminals also prefer to deploy C&C panels using SSL certificates in order to ensure end-to-end encryption. The bot has to first perform a handshake with the C&C server before any actual data is transmitted. After the handshake, the complete communication channel is encrypted. In this scenario, it does not matter whether the certificates are obtained from the authorized parties or not because these are the bots, not the browsers, that initiate the connections to the server. Most of the time self-signed certificates are used for C&C communication as they are easy to manage.

Server-Side Deployments: Cybercriminals can use different server side programming languages to design a C&C panel. C&C panels can be authored in Hypertext Preprocessor (PHP), Active Server Pages (ASP), ASP .NET (ASPX), a Common Gateway Interface (CGI) module, Java Server Pages (JSP), and so on. The most important factor that plays a significant part in selecting the server side's programming language is ease of deployment and management. It has been found that PHP is the preferred choice of cybercriminals to build C&C panels as it is free and can be used for general-purpose programming in addition to server-side scripting.

C&C Panels Deployment on Compromised WordPress CMS: It has been found that cybercriminals are exploiting vulnerabilities in content management systems (CMSs) such as WordPress, Joomla, Drupal, Magento, and others. WordPress is the leading CMS and has been extensively abused by cybercriminals to conduct multiple sets of attacks [15], including drive-by download, phishing, and so on. Compromised WordPress deployments are used to host C&C panels for managing botnets.

Experimental Methodology and Procedures

Data Collection Methodology

To conduct this study, we require C&C URLs of HTTP-based botnets. The URLs should belong to the real-time deployment of C&C servers where actual working botnet C&C panels have been found. This makes the study effective because every single C&C URL references the actual botnet deployment, so the insights gathered during the course of this experiment will reveal the practical nature of web-based C&C panels.

We obtained the data from multiple sources as follows:

- We fetched the botnet C&C panels list from cybercrime tracker (cybercrime-tracker.net), zeus tracker (zeustracker.abuse.ch), pony tracker (tracker.h3x.eu), and others. These trackers are managed by a group of security researchers who deploy techniques such as sandboxing, web crawling, and URL analysis to determine whether specific URLs belong to botnets or not.
- We developed URL extracting scripts to retrieve the required data from a third-party online storage platform such as pastebin. This step helps us to obtain the botnet URLs from the third-party storage repositories used by researchers to share the results of their analysis. Sometimes, cybercriminals also share botnet URLs through these storage platforms.
- We collected the data by manual surfing through websites providing information related to botnets. This process helped us to collect URLs of the botnets that are indexed through search engines, thereby increasing the population in our data sets.
- We developed multiple scripts to calculate the Shannon entropy of the URLs to determine the randomness of characters. This is done to verify the randomness of C&C URLs to determine the predictability.

Using all the techniques, we collected close to 19,786 botnet URLs.

Data Selection

Data Processing and Cleaning: We collected approximately 19,786 URLs during the data collection phase, as mentioned above. For this experiment, we selected and verified the collected URLs as follows:

- For the active URLs, we crawled the C&C domains using tools such as wget and curl by camouflaging HTTP headers to mimic browser identity to ensure that the URL is not downloading any malware (executable) to the end-user system. This is very important because this study aims at highlighting post-infection scenarios after the malware has been downloaded onto the system. Thus, URLs used for drive-by download attacks are not useful for this study.
- For the non-active URLs, we performed the analysis on the structure of the URLs based on the intelligence highlighted in the earlier research [10] which includes passive fingerprinting of the web resources including web pages, file names, and directory structures. All the URLs containing fingerprints of BEPs used for drive-by downloads were removed from the data set in addition to the URLs that contain direct executable filenames with extensions inlcuding exe, jpg, and others.

After the data selection phase, we obtained 10,436 C&C URLs out of 19,786 URLs. Once the C&C URLs were extracted from the dataset, we performed an additional step of selecting the unique URLs from the list.

Selecting Unique C&C URLs: We consider the deployment of C&C panel as unique; that is, C&C URLs are treated as unique if:

- The domain name or IP address is different in the C&C URL.
- Different web directories are present on the same server represented by the C&C URL.
- Different web pages are present in the same web directory on the same server represented by the C&C URL.
- The domain name or IP address is the same, but the web resources (directory, web pages) are different.

The URLs' uniqueness is obtained by dissecting the layout of the URL, that is, analyzing how exactly the URL is generated. The uniqueness of URLs is not based on the dedicated and shared hosting mechanisms. This is because a shared hosting server can have multiple C&C panels hosted using different hosts. Hence, out of 10,436 C&C URLs, we got 9437 unique C&C URLs.

The URLs' uniqueness is obtained by dissecting the layout of the URL, that is, analyzing how exactly the URL is generated. The uniqueness of URLs is not based on the dedicated and shared hosting mechanisms. This is because a shared hosting server can have multiple C&C panels hosted using different hosts. Hence, out of 10,436 C&C URLs, we got 9437 unique C&C URLs.

We performed an analysis of the characteristics of approximately 9437 unique URLs specific to different C&C panels found in real time. The C&C characteristics include encrypted and anonymous communication using SSL/TLS and TOR, respectively, the use of IP addresses instead of domain names, deployment of C&C panels over non-HTTP ports and compromised servers, and others. As a part of the experiment, we also conducted entropy tests using Shannon entropy on the 9437 C&C URLs. The idea is to understand the randomness of the characters or numbers in the URLs. It is interesting to understand the randomness of the URLs because it shows how hard or easy it is to predict the URL. If the entropy value is low, it is easy to predict the URL or brute force the web directory on the server as it maps back to the use of simple dictionary words or combinations of numbers. For example, if the entropy value of the URL or domain name is low, it is easy to brute-force the complete URL or additional web resources such as the web page or the directory on the server. If the entropy value is high, the URL or domain name is more random, and thus cannot be predicted easily.

Results and Discussions

Results on HTTP-Based Botnet C&C Panels Characteristics

We make the following observations from the results shown in Table 1:

- We found that out of 9437 botnet C&C URLs, 1702 URLs used IP addresses instead of DNS names. This means that attackers did not regis-

Serial number	HTTP-based botnet C&C panel characteristics	Total number of URLs	Percentage of URLs
1	IP-address-based C&C communication	1702	18.029%
2	DNS-based C&C communication	7738	81.97%
3	Non-standard HTTP ports	119	1.260%
4	Encryption (SSL/TLS) for C&C communication	53	.561%
5	TOR web communication using reserved TLDs	41	.434%
6	Server-side deployment: PHP ASP ASP.NET CGI Direct directory referencing	 9101 59 21 1 255	 96.43% .625% .222% .0105% 2.702%
7	C&C panels deployment on compromised WordPress	457	4.832%

Table 1. Analysis of HTTP-based botnet C&C panels characteristics.

ter any domain names that resolved to specific IP addresses. The attackers prefer to use direct IP addresses. The remaining 7738 URLs were found to be using registered domain names. Out of 7735 URLs, 234 unique TLDs have been used for a fully qualified domain name (FQDN), which specifies the exact location in the DNS tree hierarchy covering the root zone and the TLD. This number comprises the domains that have been registered or compromised by the cybercriminals for deploying C&C panels. Table 2 shows the top 20 TLDs used in FQDNs employed for hosting C&C panels. The top 20 TLDs comprise 6521 domains used for unique URLs for C&C panels out of the 7735 domains, representing approximately 84.27 percent of the C&C panels that use DNS for resolving C&C panel address.

•Out of 9437 botnet C&C URLs, 119 URLs were found to be using non-standard HTTP ports. As mentioned earlier, web-based communication uses TCP port 80 for all HTTP and TCP port 443 for all HTTPS traffic. We found that attackers also prefer to deploy C&C panels on web servers that use non-standard HTTP ports.

•Out of 9437 botnet C&C URLs, only 58 C&C URLs were found to be using HTTPS. The encryption we are highlighting here is SSL/TLS full channel encryption. The bots also use custom-level encryption, and they send encrypted payloads (HTTP POST body) over a non-HTTPS (TCP port 80) channel. However, in that case HTTP headers are visible.

•Out of 9437 botnet C&C URLs, 9101 C&C panels were found to be written in PHP, which is close to 96.43 percent. This reveals that PHP is used on a large scale in the design of HTTP-based C&C panels. We found that ASP and ASP.NET were also used, but together were only close to .85 percent of the URLs and are not that often used for designing HTTP-based C&C panels. 2.702 percent of the URLs were directly referencing web directories instead of web pages, so it was hard to determine whether the default pages belong to PHP, JSP, ASP, ASP.NET, CGI, or other. We did not find any web C&C panel written in JSP in the data sample set used for the analysis.

•We also observed that C&C panels have also been deployed using compromised WordPress websites; 4.84 percent of C&C panels were found to be hosted on WordPress.

Entropy Results

Table 3 shows the entropy results we obtained. We observed that 5994 domain names out of 7735 have entropy values less than of .40, which shows that domain names are predictable in nature. However, when these domain names are used to build complete URLs (including the path to the web resources), the entropy increases: we found that 4618 URLs out of 7735 have entropy values greater than .50. Similarly the entropy increases when IP addresses are used in URLs.

The results presented in this empirical analysis highlight the state of HTTP-based botnets. The results for non-HTTP based botnets such as peer-to-peer (P2P) and Internet Relay Chat (IRC) will not be entirely same as those of HTTP-based botnets. This is because: (1) the protocols used for IRC and P2P work differently compared to HTTP; (2) the botnet architecture is different (e.g, P2P is based on distributed communication, whereas IRC is based on the text-based client-server conferencing model, Internet messaging); and (3) HTTP is heavily used for web operations, whereas protocols supporting P2P and IRC communication models are not that widely used for normal user operations. Consequently, HTTP-based botnets can hide their communications in normal HTTP flows in the network.

Strategies for Detecting and Preventing Botnet Communications over HTTP

Based on the results obtained from our empirical measurements, we recommend that the following strategies and mechanisms should be taken into consideration when building security solutions for detecting botnet communications.

•It is false to assume that bots always generate DNS traffic before setting up a communication channel with the C&C server. It depends on the choice of the cybercriminals on how they want to set up communication channels. For detecting C&C communication or data exfiltration, in addition to DNS traffic, direct connections using IP addresses followed by HTTP traffic to external servers should be monitored and analyzed for better monitoring.

•The security solutions should analyze TOR anonymous communication that is happening in the enterprise networks. As a matter of fact, whether the encrypted TOR traffic can be decrypted or not, the TOR communication should not be allowed and must be restricted once detected in the enterprise network environment.

•HTTP traffic analysis should not be performed only on standard TCP ports such 80 and 443. HTTP communication occurring on all non-standard ports to external servers should be dissected to observe anomalies so that unauthorized HTTP communications can be restricted.

•It is not safe to assume that bots do not use SSL/TLS for C&C communication. We have noticed traces of C&C panels that have been deployed over SSL/TLS, thereby resulting in complete end-to-end encryption. This demonstrates that it has become essential to also analyze the encrypted traffic. For example, SSL/TLS communication initiated using a self-signed certificate with anomalies and a handshake performed using weak ciphers should be checked. Robust algorithms are required to detect anomalous encrypted traffic to detect potential HTTP-based C&C communication channels.

Finally, we highlight some prevention measures that can be taken to avoid the infections in the first place.

•Users should follow safe surfing habits and avoid clicking unauthorized URLs embedded in the phishing emails or shared using other communication channels on the Internet, including online social networks (OSNs) and so on, as a part of social engineering attacks. Users should update their systems at regular intervals of time to ensure no vulnerable software is running in the systems. Users' systems should be equipped with advanced anti-virus (AV) software to provide assurance against infections that occur from known set of malware.

•Inline traffic analysis should be conducted to dissect the HTTP traffic, and behavior modeling should be performed to detect anomalies in the traffic. Large scale traffic analysis results in building robust detection models using machine learning, data mining, natural language processing, contextual analysis, and other methods. Malware signatures and heuristics can also be used in conjunction with machine learning models to build hybrid solutions to detect and prevent botnet communications. The data obtained from AV engines running on the end-user systems should be correlated with the network traffic obtained from the HTTP Proxies and malware sandbox solutions to obtain the granular details to detect and prevent suspicious network communication mapping to botnet activities in the network environment. Correlation of logs from multiple resources in the network security devices such as firewalls, intrusion detection systems (IDSs), intrusion prevention systems (IPSs), proxies, and so on, should provide a single window pane to better understand the communication happening in the enterprise networks and anomaly detection so that appropriate actions can be taken in a timely manner.

Serial number	Top 20 TLDs used in domains for C&C panel deployment	Count of TLDs
1	com	3213
2	ru	804
3	net	537
4	org	281
5	biz	242
6	info	228
7	in	209
8	com.br	138
9	co.uk	131
10	tk	94
11	su	85
12	es	77
13	pw	75
14	com.au	68
15	me	64
16	de	61
17	eu	60
18	xyz	54
19	nl	51
20	fr	49

Table 2. Top 20 TLDs used in domains used to host HTTP-based C&C panels.

The algorithms designed for HTTP-based botnets cannot be made directly applicable to non-HTTP-based botnets in a similar layout because of differences in the communication models and configured protocols in the deployed environments. However, the generic strategies described above can be applied to the detection and prevention of non-HTTP-based botnets provided that the algorithms are made communication-protocol-specific. It means algorithms using techniques such as machine learning, heuristics, and signatures must use features from the protocols used by the non-HTTP based botnets. For example, P2P botnets are based on the decentralized architecture, and the communication protocol is different from HTTP, so the algorithm needs to be tuned accordingly. The overall idea is that detection and protection mechanisms can be similar, but they need to be designed in accordance with the botnet communication models.

Serial number	Data layout	Total count	Entropy < 0.40	Entropy >= .40 and < .45	Entropy >= .45 and < .50	Entropy >= .50 and <.55	Entropy >= .55
1	Extracted domain names from URLs	7735	5994	1553	186	2	0
2	URLs with domain names	7735	5	197	2485	4618	430
3	Extracted IP addresses as hosts for URLs	1702	1699	3	0	0	0
4	URLs with IP addresses	1702	10	34	830	519	309

Table 3. Shannon entropy layout of URLs containing domain names and IP addresses.

Users should follow safe surfing habits and avoid clicking unauthorized URLs embedded in phishing emails or shared using other communication channels on the Internet such as online social networks as a part of social engineering attacks.

Conclusion

We have conducted an empirical study of the deployment of HTTP-based botnet C&C panels. The study has been conducted over more than 9000 botnet C&C URLs to understand the design and communication model used by HTTP-based botnets. The URLs specific to C&C panels pertain to financial botnets, including Zeus, Citadel, Pony, and so on, ransomware, POS malware, and others that utilize HTTP as a primary communication protocol. This study has highlighted very interesting characteristics of C&C panels. Based on the results obtained, we have also elaborated on the strategies that need to be deployed to detect botnet communications occurring over HTTP.

Acknowledgments

We thank the anonymous reviewers for their valuable comments, which helped us to improve the quality and presentation of this article.

References

[1] A. Sood and R. Enbody, "Crimeware-as-a-Service, A Survey of Commoditized Crimeware in the Underground Market," *Int'l. J. Critical Infrastructure Protection*, vol. 6, no. 1, 2013, pp. 28–38.

[2] V. Kotov and F. Massacci, "Anatomy of Exploit Kits: Preliminary Analysis of Exploit Kits as Software Artefacts," *Proc 5th Int'l. Conf. Engineering Secure Software and Systems*, 2013, Springer-Verlag, , pp. 181–96; DOI=http://dx.doi.org/10.1007/978-3-642-36563-8_13

[3] A. Sood and S. Zeadally, "Drive-by Download Attacks: A Comparative Study of Browser Exploit Packs Features and Attack Techniques," *IEEE IT Professional*, vol. 18, no. 5, 2016, pp. 18–25.

[4] T. Taylor *et al.*, "Detecting Malicious Exploit Kits Using Tree-Based Similarity Searches," *Proc. Sixth ACM Conf. Data Application Security Privacy*, 2016, pp. 255–66; DOI: http://dx.doi.org/10.1145/2857705.2857718.

[5] C. Yue and H. Wang, "A Measurement Study of Insecure Javascript Practices on the Web," *ACM Trans. Web*, vol. 7, no. 2, May 2013.

[6] N. Provos *et al.*, "All Your iFRAMEs Point to Us," *Proc. 17th USENIX Conf. Security Symp.*, pp. 1–15.

[7] F. Mouton, L. Leenen, and H. S. Venter, "Social Engineering Attack Examples, Templates and Scenarios," *Computers & Security*, vol. 59, June 2016, pp. 186–209.

[8] E. Kirda and C. Kruegel, "Protecting Users against Phishing Attacks," *Computer J.*, vol. 49, no. 5, 2006, pp. 554–61.

[9] A. Sood, S. Zeadally, and R. Enbody, "An Empirical Study of HTTP-Based Financial Botnets," *IEEE Trans. Dependable and Secure Computing*, vol. 13, no. 2, 2013, pp. 236–51.

[10] A. Sood, "Exploiting Fundamental Weaknesses in Botnet C&C Panels," *Proc. BlackHat Security Conf.*, Aug. 2014; https://www.blackhat.com/docs/us-14/materials/us-14-Sood-What-Goes-Around-Comes-Back-Around-Exploiting-Fundamental-Weaknesses-In-Botnet-C&C-Panels.pdf, accessed 20 Sept. 2016.

[11] A. Sood, R. Enbody, and R. Bansal, "Dissecting Spy Eye — Understanding the Design of Third Generation Botnets," *Computer Networks*, vol. 57, no. 2, Feb. 2013, pp. 436–50.

[12] A. Sood and S. Zeadally, "A Taxonomy of Domain Generation Algorithms," *IEEE Security and Privacy Mag.*, vol. 18, no. 5, 2016, pp. 46–53.

[13] J. Applebaum and A. Muffet, "The '.onion' Special-Use Domain Name," IETF RFC 7686 Oct. 2015, <https://tools.ietf.org/rfc/rfc7686.txt>; accessed 23 Sept. 2016.

[14] Trac, "Generating TOR Onion Domains," Apr. 2010; https://trac.torproject.org/projects/tor/wiki/doc/HiddenServiceNames, accessed 22 Sept. 2016.

[15] Sucuri, "Website Hacked Trend Report — Q1 Report on Post-Hack Actions by Attackers," Jan. 2016; https://sucuri.net/website-security/Reports/Sucuri-Website-Hacked-Report-2016Q1.pdf, accessed 26 Sept. 2016.

Biographies

Aditya K. Sood (soodadit.msu@gmail.com) is a director of the Security and Elastica Cloud Threat Labs, Blue Coat Systems, a Symantec Company. His research interests include cloud and web security, malware analysis, mobile security, and penetration testing. He received a Ph.D. in computer science from Michigan State University.

Sherali Zeadally (szeadally@uky.edu) received his Bachelor's degree in computer science from the University of Cambridge, United Kingdom, and his doctoral degree in computer science from the University of Buckingham, United Kingdom. He is an associate professor in the College of Communication and Information at the University of Kentucky. He is a Fellow of the British Computer Society and a Fellow of the Institution of Engineering Technology, United Kingdom.

Rohit Bansal (rb@secniche.org) is a senior security researcher at SecNiche Security Labs. His research interests include malware analysis, reverse engineering, and web application security. He has a B.E. in computer science from Uttar Pradesh Technical University.

Traffic-Aware Patching for Cyber Security in Mobile IoT

Shin-Ming Cheng, Pin-Yu Chen, Ching-Chao Lin, and Hsu-Chun Hsiao

Abstract

The various types of communication technologies and mobility features in IoT on one hand enable fruitful and attractive applications, but on the other hand facilitate malware propagation, thereby raising new challenges in handling IoT-empowered malware for cyber security. Compared to the malware propagation control scheme in traditional wireless networks, where nodes can be directly repaired and secured, in IoT, compromised end devices are difficult to patch. Alternatively, blocking malware via patching intermediate nodes turns out to be a more feasible and practical solution. Specifically, patching intermediate nodes can effectively prevent the proliferation of malware propagation by securing infrastructure links and limiting malware propagation to local device-to-device dissemination. This article proposes a novel traffic-aware patching scheme to select important intermediate nodes to patch, which applies to the IoT system with limited patching resources and response time constraint. Experiments on real-world trace datasets in IoT networks are conducted to demonstrate the advantage of the proposed traffic-aware patching scheme in alleviating malware propagation.

Introduction

By integrating the ability to sense the physical world and the privilege of availing communication capabilities, the Internet of Things (IoT) enables close interactions between humans and machines. IoT generally consists of numerous IoT end devices for sensing and action, intermediate nodes with wired connectivity for data relaying, and application servers in the cloud for data control and analysis. Typically, IoT devices can communicate with each other with minimal human intervention and build an autonomous and complex network. As the boundary between machines and humans gets blurry, adversaries in cyberspace can threaten human users' safety and privacy in the physical world. Obviously, the growing popularity of devices with rich wireless communication capabilities has made IoT attractive to digital viruses and malicious contents. Consequently, in recent years the security issues in IoT have been an ever increasing concern [1–3].

From an adversary's perspective, the unique features of IoT facilitate the exploitation of devices as well as the propagation of IoT malware. These features include constrained resources, heterogeneous links, and vulnerable usability, which are discussed as follows.

Resource-Constrained IoT Devices: Compared to the intermediate nodes located at the end side of the infrastructure with wired connectivity, IoT devices designed to perform simple sensing and actuation operations have limited computation and communication capabilities. In this case, the algorithm and mechanism applied on IoT devices are relatively simple. As a result, the attacker can spend much less resources to break into IoT devices, rendering them the targets of malicious users. For example, due to the overhead of certificate management and public key cryptography, many existing IoT devices fail to support state-of-the-art secure communication protocols (e.g., SSL/TLS). Therefore, the adversary can eavesdrop on sensitive sensor data and even manipulate data without being detected. Another example is that IoT devices often have limited entropy sources, which results in weak cryptographic keys that can be predicted by the attacker. Moreover, since most IoT devices run on embedded Linux operating systems (OSs), the attacker can easily create IoT malware by recompiling existing Linux malware for other instruction set architectures.

Heterogeneity: In order to support different kinds of IoT applications, IoT devices are often equipped with heterogeneous communication and computation capabilities for the purpose of seamless operations. However, the heterogeneity and potentially vast amount of IoT devices facilitate the fabrication of identity and hiding of malware. Moreover, as shown in Fig. 1, compromised IoT devices might disseminate malware via heterogeneous communication links as described below.

Infrastructure Links: IoT malware can propagate using infrastructure-based communication technologies, such as GSM/GPRS/UMTS/LTE and WLAN, via intermediate nodes, such as access point (AP), base station (BS), or gateway. In particular, IoT malware inherits the threats caused by computer malware. Similar to computer malware, most IoT malware families today scan the IP address space for vulnerable victims and spread via the Internet. Due to the widespread use of weak login credentials and the fact that many IoT devices are Internet-accessible, some botnets have allegedly harvested more than one million infected IoT devices (http://thehackernews.com/2016/10/iot-dyn-ddos-attack.html).

The authors propose a novel traffic-aware patching scheme to select important intermediate nodes to patch, which applies to the IoT system with limited patching resources and response time constraint. Experiments on real-world trace datasets in IoT networks are conducted to demonstrate the advantage of the proposed traffic-aware patching scheme in alleviating malware propagation.

Shin-Ming Cheng and Ching-Chao Lin are with National Taiwan University of Science and Technology; Pin-Yu Chen is with IBM Thomas J. Watson Research Center; Hsu-Chun Hsiao is with National Taiwan University.

Digital Object Identifier: 10.1109/MCOM.2017.1600993

0163-6804/17/$25.00 © 2017 IEEE

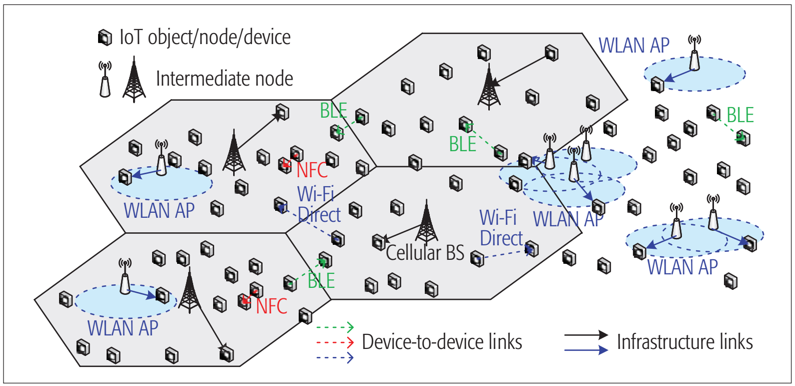


Figure 1. IoT platform with infrastructure and device-to-device links.

Device-to-Device Links: IoT malware could exploit proximity-based wireless media such as Bluetooth Low Energy (BLE), WiFi Direct, and near field communication (NFC) to infect the devices in the vicinity [4]. In this case, IoT malware is stored and forwarded by taking advantage of mobility and ubiquity. For example, Colin O'Flynn in Black Hat USA 2016 as well as Ronen and Shamir [5] discussed the possibility of a light bulb worm, which allows a reprogrammed bulb to re-flash nearby bulbs.

Usability: Security is only as strong as its weakest link, and the weakest link, in many cases, is the humans who implement, operate, and use the system. For example, a proven secure cryptographic primitive, if implemented or used incorrectly, can still be circumvented. Moreover, users may choose to ignore or even bypass a security mechanism if it prevents (e.g., due to slow performance, badly designed user interface, and unclear instructions) the users from doing what they mean to do. Since IoT devices often lack convenient input and output interfaces, the original security features might be bypassed by non-professional IT users, thereby increasing the possibility and risk of human errors and facilitating the spreading of malware [2].

Obviously, software updates and patching are necessary to prevent IoT devices from being compromised. A single software flaw will make a tremendous range of IoT devices vulnerable to attacks since software components are reused in different devices (http://blog.senr.io/blog/400000-publicly-available-iot-devices-vulnerable-to-single-flaw). However, without a friendly user interface (UI) through which to be alerted about security updates, most users forget to update software installed in IoT devices and leave them out of date. In addition, without basic programming knowledge and security awareness, users might be unwilling to take a manual-download-and-install approach for software updating. As a result, it is critical to design a reasonable solution to prevent the occurrence of large-scale malware propagation among trillions of unpatched, insecure, and even compromised IoT devices.

Instead of patching resource-constrained and UI-unfriendly compromised IoT devices directly, this article introduces a more feasible solution, where operators could only patch or recover IoT devices via infrastructure (i.e., securing the intermediate nodes). In this case, a patched AP, BS, or gateway could stop the malware propagation by patching via infrastructure links. The concept of leveraging intermediate nodes to improve IoT security has appeared in the recent commercial product F-Secure SENSE (https://community.f-secure.com/t5/F-Secure-SENSE/What-are-the-current-protection/ta-p/82972). However, its main purpose is to block malicious websites and IoT botnet masters instead of considering securing important infrastructure links between IoT devices and intermediate nodes. On the other hand, the idea behind IoT Sentinel [6] is similar to our solution, where the types of IoT devices are identified by intermediate nodes, and the communications of vulnerable IoT devices are constrained by enabling enforcement of rules. Different from our solution, software-defined networking (SDN) is exploited in IoT Sentinel for network flow isolation and prevention of malware propagation.

With limited efforts and resources, an operator might not be able to patch all intermediate nodes but only a portion of them. One naive method is to simply patch those intermediate nodes in a random order. However, a smarter approach is to protect the most important node first, as suggested by the framework of network robustness analysis [7, 8]. This article proposes a traffic-aware patching scheme, where the operator patches the intermediate nodes sequentially in descending importance order. In particular, an intermediate node that could have contact with a large number of IoT devices will be protected first. Moreover, such a volume-based patching approach is effective against the current infamous distributed denial of service (DDoS) attacks launched by IoT bots.

By leveraging a real-world trace dataset containing communication history over device-to-device and infrastructure links, we conduct an extensive experiment to demonstrate the effect of constraining malware propagation via infrastructure links. To the best of the authors' knowledge, this article is the first work discussing the control of malware propagation from the perspective of infection paths, which could avail the damage estimation caused by the malware and improve the development of attack detection methods for IoT networks.

How to Compromise IoT Devices

IoT devices are an attractive attack target for cybercriminals: IoT devices often employ weak security measures, and their compromise can lead to privacy breaches and safety threats in the real world. The insecurity of existing IoT devices has been highlighted repeatedly by security researchers and practitioners. Recently, several malware families were found to target vulnerable IoT devices (e.g., routers, IP cameras, and CCTVs) and form botnets for DDoS. It is estimated that some IoT botnets comprise more than one million infected devices, and thus can generate high-volume DDoS traffic even without amplification. For example, in September 2016, an IoT botnet called Mirai crippled a website with 620 Gb/s of attack traffic, which is almost twice as much as the biggest DDoS attack witnessed in 2015. Later, in October 2016, the same botnet attacked the Dyn DNS service provider, taking down a large portion of websites in North America, including GitHub, Twitter, Netflix, and so on (https://www.us-cert.gov/ncas/alerts/TA16-288A). At DEF CON 2016, security researchers showed a proof-of-concept IoT ransomware that demands ransom for a hacked smart thermostat, which will be set to a high temperature without timely payment (https://www.pentestpartners.com/blog/thermostat-ransomware-a-lesson-in-iot-security/). As attackers are finding creative ways to monetize infected IoT devices, it is inevitable to see an increase of new IoT malware families that are more destructive and contagious than ever.

IoT malware can propagate via infrastructure links and/or device-to-device links. We discuss both cases in this section.

Compromising IoT Devices via Infrastructure Links

Many of the IoT malware families today propagate via infrastructure links, particularly the Internet. Moreover, they share a common infection and spreading pattern: The attacker harvests new vulnerable IoT devices through address space scanning. This scanning can be performed by external servers, such as command and control (C&C) servers, or by the compromised devices. The attacker targets Telnet- or SSH-accessible devices that use default or weak login credentials and thus can easily obtain root access permission by brute-force password cracking. Once the attacker gets the shell of the hacked device, the malware payload is downloaded and installed. IoTPOT [2], an IoT honeypot project, observed at least four IoT malware families that can propagate via Telnet. In addition to cracking weak passwords, some malware also exploits software vulnerabilities. For example, CCTV-targeting RADIATION malware exploits ShellShock and some known CCTV vulnerabilities to spread from device to device.

Compromising IoT Devices via Device-to-Device Links

Malware can also propagate in proximity via device-to-device links in addition to infrastructure links. Cabir and Commwarror are examples of mobile worms that spread via Bluetooth and infect mobile phones running the Symbian OS.

Although we have not witnessed device-to-device IoT malware in the wild, it is theoretically possible. For example, researchers pointed out the possibility of light bulb worms that spread to nearby bulbs via Zigbee [5] and worms that infect wearable trackers and then spread to others by Bluetooth (http://www.theregister.co.uk/2015/10/21/fitbithack/). Moreover, since proximity-based wireless interfaces are often always on, and users have no control to disable them, it would be difficult to contain malware propagation given the large attack surface.

Regardless of how malware propagates, the risk of self-replicating IoT malware is amplified by unpatched IoTs. Patching vulnerable IoT devices nevertheless remains extremely expensive and far from successful in practice. In 2015, Charlie Miller and Chris Valasek demonstrated remote exploitation of a Jeep, which forced Chrysler to recall and patch 1.4 million vehicles (https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/). Cui and Stolfo [9] discovered more than 540,000 publicly accessible devices using default root passwords — an old but persistent vulnerability since the invention of password-based authentication. Worse yet, the problems encountered when patching computers and mobile phones (e.g., privacy, legacy devices, and lack of incentives) will linger and even exacerbate when attempting to patch IoT devices.

Modeling of IoT Malware

The topic of modeling malware/virus spreading has been investigated in a traditional scenario where computers or laptops are not connected to the Internet. Since the spread of epidemics among people is similar to the spread of malware over networks, the current literature adopts the idea from epidemiological models to build the models for malware on the assumption of a homogeneous infection path [10]. In the mobile environment, malware can propagate via intermittently connected networks by taking advantage of opportunistic encounters [11]. Wang *et al*. [12] study spreading patterns of mobile phone viruses, which may traverse through multimedia messaging services (MMS) or Bluetooth by simulations. Cheng and Chen [13] further model malware propagation in generalized social networks consisting of delocalized and localized links.

From the discussion of the previous section, we understand that in practice patching compromised IoT devices is difficult to achieve. Consequently, the current formulation of malware propagation and the control model [14] cannot be applied directly in the IoT field. Typically, in one of the most famous susceptible-infection-recover (SIR) models, the malware is assumed to be detected and repaired at each node, which reflects the transition from "infected" state to "recovered" state. Regarding the IoT device that detects the malware, instead of directly patching it, it is more feasible to patch on the infrastructure side to prevent further spreading of malware. In this case, compromised IoT devices located in the coverage area of the patched intermediate nodes are controlled, that is, malware cannot be propagated via patched intermediate nodes. As a result, the infrastructure links can be regarded as "recovered" while the compromised IoT device remains "infected," using the terminology of the SIR model. The observation that malware control in IoT environment can be cast as a "link recovery" problem instead of a "node recovery" problem motivates a different development of modeling and formulation.

Feasible Patching Schemes in the IoT Environment

This section proposes patching schemes for the IoT environment, where we can only control infrastructure links but not the compromised nodes themselves. The patching scheme consists of several phases. In the *detecting phase*, infrastructure leverages a traditional intrusion detection system (IDS) or firewall to identify the existence of malware or a compromised node. Once malicious code is found to be propagated from the compromised IoT devices, the *patching phase* starts to analyze the malware and patches the intermediate nodes according to a patching sequence to prevent the large-scale propagation of malware. In practice, intermediate nodes are capable of performing resource-intensive tasks and thus can support over-the-air (OTA) update mechanisms. In the *patching phase*, such OTA mechanisms allow

IoT devices are an attractive attack target for cybercriminals: IoT devices often employ weak security measures, and their compromise can lead to privacy breaches and safety threats in the real world. The insecurity of existing IoT devices has been highlighted repeatedly by security researchers and practitioners.

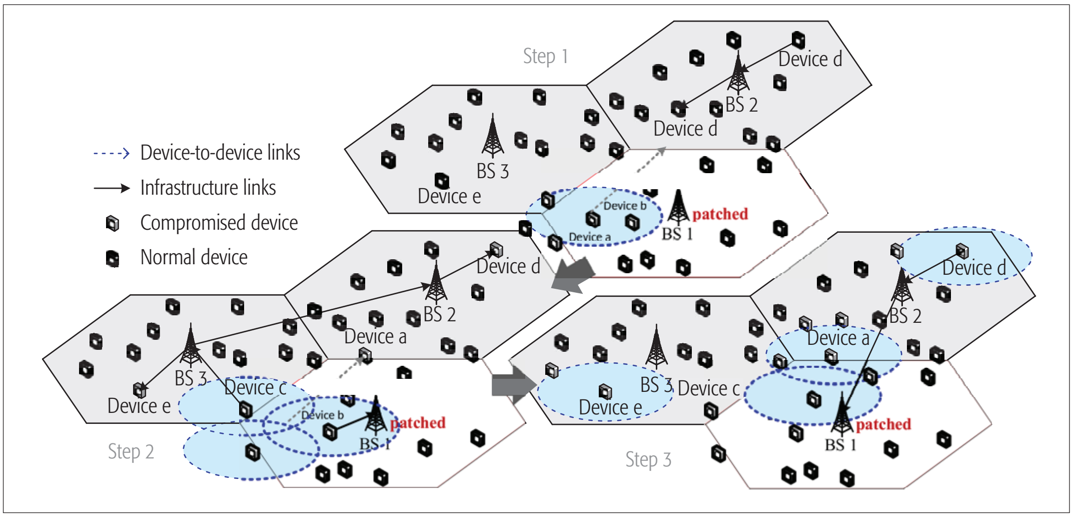


Figure 2. Illustration of malware propagation under the infrastructure patch scheme.

```
Input: The set of intermediate nodes, $S_{AP}$; The time to start patching, $t_p$; The percentage of patched intermediate nodes, p
1: If currentTime < $t_p$ then
2:     Collect traffic information for each intermediate node
3: else
4:     If currentTime ≥ $t_p$ then
5:         Sort intermediate nodes according to the
6:             importance metric in descending traffic order
7:         Patch top p % $S_{AP}$
8:     end if
9: end if
```

Algorithm 1. Traffic-aware patching.

the administrator to remotely install required update on the intermediate nodes, thereby ensuring timely mitigation of compromised nodes. In addition, since intermediate nodes are significantly fewer than IoT devices, the administrator can also manually patch legacy intermediate nodes that do not support OTA update.

Figure 2 describes an example of how a compromised device propagates malware in an IoT environment with patched and unpatched intermediate nodes. For the devices located in the coverage area of the patched intermediate nodes, two possible operations are executed.

Compromised devices can distribute malware via device-to-device links but not infrastructure links. As shown in step 1 of Fig. 2, the compromised device propagates malware to devices b and c in the vicinity. However, in step 2 of Fig. 2, device b cannot propagate malware via infrastructure link since the malware is blocked at the patched BS.

Normal devices can only be compromised via device-to-device links since the malware propagated from infrastructure will be identified and blocked by the patched intermediate nodes. For example, in step 3 of Fig. 2, device d propagates malware from BS 2 to BS 1; however, the patched BS 1 will not relay the malware to any device in its coverage area.

For the devices located in the coverage area of the unpatched intermediate nodes, there are no means to prevent malware propagation. For example, in step 2 of Fig. 2, device c under unpatched BS 3 could infect device d controlled by unpatched BS 2 via infrastructure links. Moreover, device a moving from patched BS 1 to unpatched BS 2 could propagate malware via device-to-device links freely.

Algorithm 1 describes the detailed steps in the *patching phase*. With limited resources and efforts, the operator could provide a fixed amount of patches on the intermediate nodes (e.g., *p* percentage). To alleviate the propagation from the infrastructure links, the *p* percent most important intermediate nodes will be chosen for patching. It is similar to the idea of protecting the most important node to maintain network robustness [7]. As a result, we introduce the traffic monitoring duration (lines 1 and 2, Algorithm 1) for evaluating the importance of intermediate nodes. From the monitored results, the proposed traffic-aware patching scheme sorts the intermediate nodes in descending order according to the traffic volumes (lines 5 and 6, Algorithm 1), and the top *p* percent intermediate nodes are patched (line 7, Algorithm 1).

Obviously, the proposed volume-based patching is effective against attacks that generate a large number of traffic volume (e.g., DDoS attacks). The patched intermediate nodes could prevent the redirection of malicious traffic introduced by the DDoS attack launched by the IoT botnets.

Performance Evaluation

In this section, we implement the proposed traffic-aware patching scheme and compare its performance with a randomized patching scheme on real-life traffic traces collected from a mobile social network consisting of 59 users (devices) and 1751 APs [15]. In this network, each user can communicate with other users through two types of links: an infrastructure link via (possibly multiple) APs and a direct device-to-device link to users within transmission range. These two types of links among users are similar to the illustration of mobile IoT in Fig. 1. As mentioned previously, in this experiment infrastructure links can be made secure via patching, whereas direct device-to-device links are vulnerable to potential security threats.

Following the vulnerability analysis of transmission attacks in [3], we simulate the propagation dynamics of self-replication malicious codes by first randomly selecting a user in the network as the initially compromised device. Then, using the actual traces of communication patterns provided by the dataset [15], each infected device can compromise its contact through an infrastructure link with probability λ_{inf}, and can compromise its contact through a direct device-to-device link with probability λ_{dir}. Specifically, if one of the APs in the communication path between one infected device and its contact has been successfully patched, malware propagation is in vain due to enhanced security.

For traffic-aware patching, we are interested in investigating the trade-offs between the time spent on analyzing traffic volume (i.e., the traffic monitoring duration) and the time instance to patch APs (i.e., the patch time). As described in the previous section, given a fixed amount of patches, the proposed traffic-aware patching scheme sorts the APs in descending order according to the traffic volume in the traffic monitoring duration, and provides patches to the top APs. Intuitively, longer traffic monitoring duration better specifies the important APs in communicating devices. However, longer traffic monitoring duration also leads to more exploits in security vulnerabilities due to later patch time. As a result, given a fixed amount of patches, we aim to study the nontrivial optimal patch time that collects sufficient traffic information for patching while minimizing the security risks.

Figure 3 shows the fraction of compromised users with respect to different patch time and patched APs under the traffic-aware patching scheme. To demonstrate the effectiveness of the proposed traffic-aware patching scheme, Fig. 4 further compares the difference of compromised users between the no-patching scheme and the traffic-aware scheme. It can be observed that the best patching strategy that leads to a maximal decrease in the number of compromised users compared to the no-patching scheme is to monitor the traffics for 40 seconds and then provide patches to all APs. Note that 100 percent patched APs (i.e., securing all infrastructure links) with patch time 0 may not be the optimal patch strategy since the malicious codes are still able to propagate through direct device-to-device links. To further understand the effect of traffic-aware patching, for a given fraction of patched APs, Fig. 5 shows the optimal patch time that leads to the lowest total number of compromised users. We observe that if one is able to patch more APs, late patch time can have better performance, which suggests that traffic volumes are indeed important information for patching.

For fair comparison, we also compare the performance of traffic-aware patching with random patching. Random patching provides immediate patches (i.e., has patch time 0) and randomly selects a fraction of APs to patch. Figure 6 shows the difference between the fraction of compromised users under random patching to that of traffic-aware patching, where larger positive values imply that traffic-aware patching is more effective in securing the network and vice versa. We observe that traffic-aware patching is significantly better than random patching in the regime of a few patched APs (e.g., below 30 percent). Moreover, given a fixed fraction of patched APs, for traffic-aware patching, there is at least one patch time that leads to either better or identical performance compared to random patching, which suggests the robustness and reliability of the proposed patching scheme. Even in the regime of many patched APs (e.g., above 90 percent), the performance of traffic-aware patching is still superior to random patching, which suggests the importance of patching APs with high traffic volume for enhanced security.

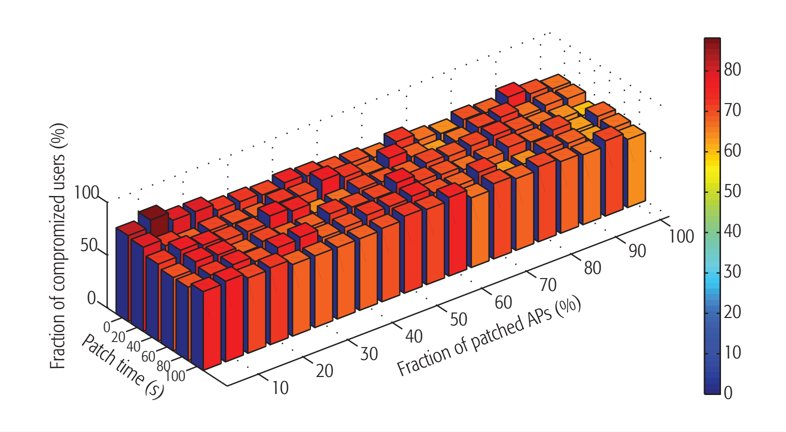


Figure 3. Fraction of compromised users with respect to different patch time and patched APs under the traffic-aware patching scheme. λ_{inf} = 0.00004 and λ_{dir} = 0.00001.

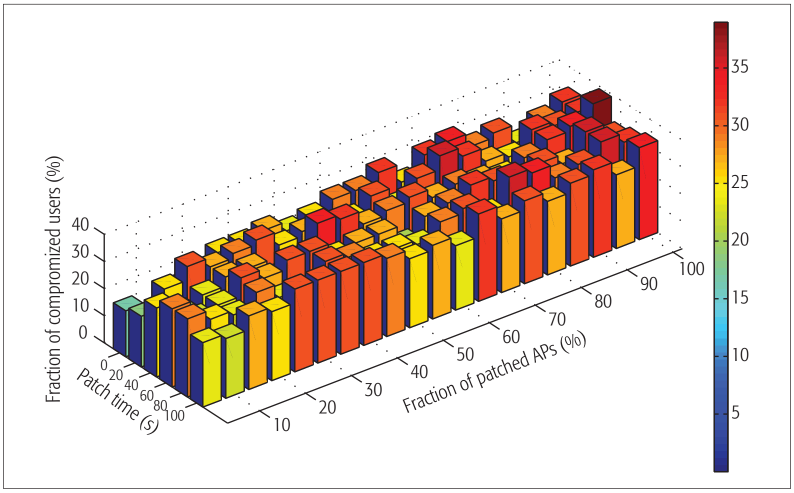


Figure 4. Performance comparison of traffic-aware patching scheme vs. the no-patching scheme. This figures shows the difference of compromised users between the no-patching scheme and the traffic-aware scheme. λ_{inf} = 0.00004 and λ_{dir} = 0.00001. The results are averaged over 500 trials.

Some Ongoing Challenges and Open Research Questions

Here we discuss several ongoing challenges and open research questions related to IoT malware propagation and patching.

Transfer Learning for Optimal Patch Time: In the experiments, we find that the patch time is crucial to preventing malware propagation. How to design and simulate realistic testbeds to assist in determining the optimal patch time and

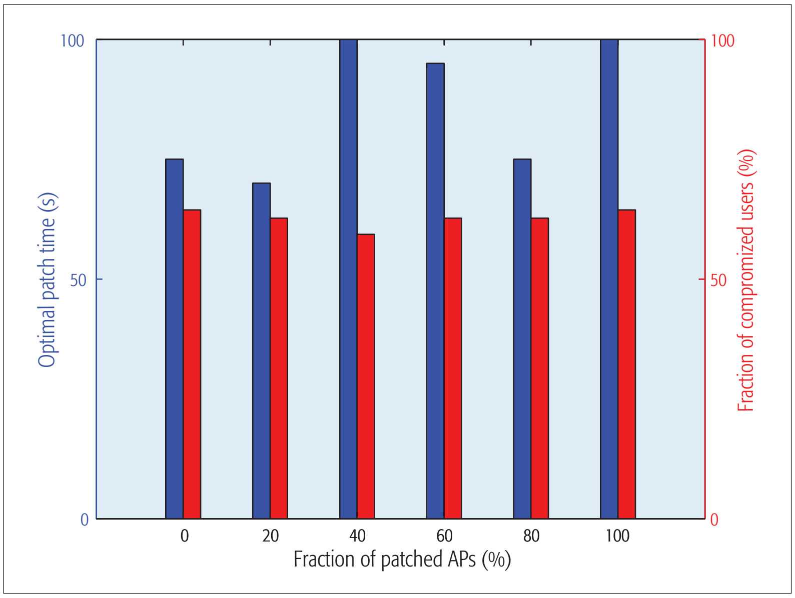


Figure 5. Optimal patch time and the corresponding number of compromised users given patched APs. λ_{inf} = 0.00004 and λ_{dir} = 0.00001. The results are averaged over 500 trials.

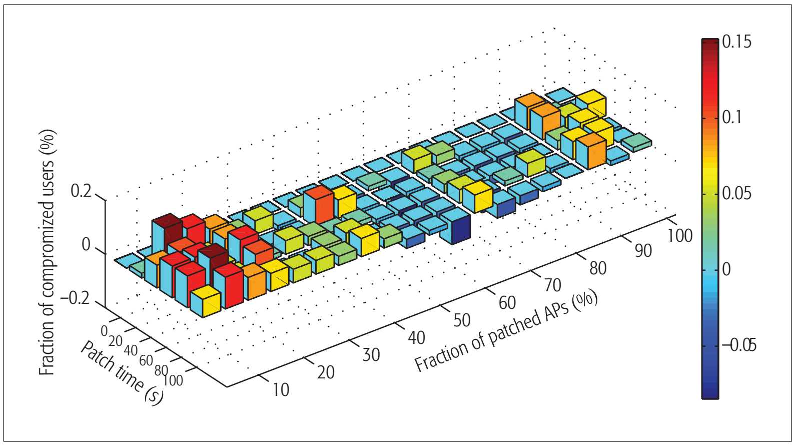


Figure 6. Performance comparison between random patching and traffic-aware patching in terms of the difference between the fraction of compromised users under random patching to that of traffic-aware patching. λ_{inf} = 0.00004 and λ_{dir} = 0.00001. The results are averaged over 500 trials.

to enable transfer learning for defending real-life unknown security threats are ongoing challenges.

Predictive Malware Propagation Models for Mobile IoT: In this article, we have addressed patching issues in mobile IoT as link recovery instead of node recovery, where the latter has been extensively studied in traditional wireless networking scenarios. How to establish effective mathematical models for predicting malware propagation dynamics in mobile IoT that take into account the traffic-aware and random patching schemes are new research challenges.

Various Importance Metrics for Intermediate Nodes: The proposed scheme simply applies traffic volume as the metric to determine the importance of intermediate nodes and the patching sequence. It can be regarded as protecting the entire network by patching a relatively small fraction of intermediate nodes with the highest degree metric. The operator could consider more information about intermediate nodes, such as the topology of intermediate nodes, in order to design a more effective importance metric for determining the patching sequence. For example, the betweenness metric could be leveraged, which is defined as the fraction of all shortest paths passing through the node among all shortest paths between each node pair in the network.

Patching via Path-Based Traffic Patterns: The proposed traffic-aware patching scheme only considers the one-hop traffic information in terms of the traffic volume from IoT devices to intermediate nodes. The patching scheme could benefit from the knowledge beyond one-hop information, such as the path-based end-to-end traffic patterns. However, path-based traffic patterns are relatively difficult to collect or acquire compared to one-hop traffic information.

How to Achieve (Virtual) Patching: IoT devices often lack user-friendly interfaces and are left unattended after installation. As a consequence, users have trouble knowing whether a device is hacked, and even if they do, they may find it challenging to *manually patch* the device: they need to retrieve updated firmware online, access the hacked device, install the firmware, and so on. Thus, *automatic patching* is needed to secure IoT at scale.

One promising direction is for IoT devices to support firmware OTA (FOTA), as most PCs and mobile phones do nowadays. However, an efficient and secure FOTA for IoT remains an open challenge due to the heterogeneity of IoT networks. For example, transport security and code signing are required to ensure the authenticity of the updated firmware. The IoT gateway might help reduce the overhead by caching and offloading the security check. Moreover, the human factors need to be taken into consideration as well. As in the PC and mobile phone worlds, forcing software update without explicit user consent can be disastrous. It can even be life-threatening if the update happens at the wrong time (e.g., updating a vehicle while driving).

Concluding Remarks

This article considers the security threats incurred by the heterogeneous links of IoT and designs a novel patching scheme to alleviate malware propagation. Instead of the impractical solution of directly patching compromised IoT devices, we propose to patch important intermediate nodes based on the traffic volumes to prevent major security exploits and to avoid catastrophic malware propagation. With the proposed traffic-aware patching scheme, malware propagation is restricted to direct device-to-device connection, and therefore the damage of malware propagation can be significantly reduced. We conduct experiments in an IoT environment to demonstrate the effectiveness of the proposed traffic-aware patching scheme, and we also discuss some ongoing research challenges and open research questions related to IoT patching.

The proposed traffic-aware patching scheme and the experimental results bring new insights to IoT security. For instance, the infeasibility of direct patching on IoT devices calls for new IoT malware models and security assessment approaches. The experimental results can assist in developing new attack detection techniques and patching strategies for preventing malware propagation.

Obviously, the resource-constrained, user-unfriendly, and heterogeneous features of IoT devices hinder the security design and development for IoT. However, the experimental results indicate a promising method to secure the entire IoT system by patching intermediate nodes. In summary, we provide the following two guidelines for how to consider cyber security when designing IoT systems accordingly

The resource-constrained, user-unfriendly, and heterogeneous features of IoT devices hinder the security design and development for IoT. However, the experimental results indicate a promising method to secure the entire IoT system by patching intermediate nodes.

•The consideration of intermediate nodes that bridge the gap between resource-constrained IoT devices and powerful IoT application servers is necessary when designing cyber security for IoT. By shifting computation-consuming, security related functionalities (e.g., flow identification, filtering, and isolation) to intermediate nodes, they can play the role of onsite guards. In particular, the flexibility and reconfigurability of intermediate nodes could easily introduce patches and updates to mitigate the IoT malware propagation or attacks in a timely manner.

•The future cyber security solution for IoT should take into consideration that adversaries might leverage IoT devices with unpatched vulnerabilities to propagate malware via device-to-device links. In other words, the security mechanisms developed for IoT shall coexist with insecure, unpatched legacy IoT devices with uncontrolled device-to-device channels. A notification mechanism is suggested to help users identify the IoT devices at risk and further deny possible device-to-device connections.

Acknowledgment

This work was supported in part by the Taiwan Information Security Center (TWISC), Academia Sinica, and the Ministry of Science and Technology, Taiwan, under grants MOST 104-2923-E-011-006-MY2, 105-2221-E-002-146-MY2, and 105-2218-E-001-001.

References

[1] J. Granjal, E. Monteiro, and J. Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," *IEEE Commun. Surveys & Tutorials*, vol. 17, July 2015, pp. 1294–1312.

[2] Y. Minn *et al.*, "IoTPOT: Analysing the Rise of IoT Compromises," *Proc. USENIX Wksp. 2015*, Aug. 2015.

[3] P.-Y. Chen *et al.*, "Decapitation via Digital Epidemics: A Bio-Inspired Transmissive Attack," *IEEE Commun. Mag.*, vol. 54, no. 6, June 2016, pp. 75–81.

[4] G. Zyba *et al.*, "Defending Mobile Phones from Proximity Malware," *Proc. IEEE INFOCOM 2009*, Apr. 2009, pp. 1503–11.

[5] E. Ronen and A. Shamir, "Extended Functionality Attacks on IoT Devices: The Case Of Smart Lights," *Proc. IEEE S&P Europe 2016*, Mar. 2016.

[6] M. Miettinen *et al.*, "IoT Sentinel: Automated Device-Type Identification for Security Enforcement in IoT," *CoRR*, vol. abs/1611.04880v2, 2016.

[7] P.-Y. Chen and S.-M. Cheng, "Sequential Defense against Random and Intentional Attacks in Complex Networks," *Phys. Rev. E*, vol. 91, Feb. 2015, p. 022805.

[8] P.-Y. Chen and A. O. Hero, "Assessing and Safeguarding Network Resilience to Nodal Attacks," *IEEE Commun. Mag.*, vol. 52, no. 11, Nov. 2014, pp. 138–43.

[9] A. Cui and S. J. Stolfo, "A Quantitative Analysis of the Insecurity of Embedded Network Devices: Results of a Wide-Area Scan," *Proc. ACSAC 2010*, Dec. 2010, pp. 97–106.

[10] S. Peng, S. Yu, and A. Yang, "Smartphone Malware and Its Propagation Modeling: A Survey," *IEEE Commun. Surveys & Tutorials*, vol. 16, no. 2, Apr. 2014, pp. 952–41.

[11] S. Tanachaiwiwat and A. Helmy, "Encounter-Based Worms: Analysis and Defense," *Ad Hoc Net.*, vol. 7, no. 7, Sept. 2009, pp. 1414–30.

[12] P. Wang *et al.*, "Understanding the Spreading Patterns of Mobile Phone Viruses," *Science*, vol. 324, no. 5930, May 2009, pp. 1071–75.

[13] S.-M. Cheng *et al.*, "On Modeling Malware Propagation in Generalized Social Networks," *IEEE Commun. Lett.*, vol. 15, no. 1, Jan. 2011, pp. 25–27.

[14] P.-Y. Chen, S.-M. Cheng, and K.-C. Chen, "Optimal Control of Epidemic Information Dissemination over Networks," *IEEE Trans. Cybernetics*, vol. 44, no. 12, Dec. 2014, pp. 2316–28.

[15] W. Dong, B. Lepri, and A. Pentland, "Modeling the Co-Evolution of Behaviors And Social Relationships Using Mobile Phone Data," *Proc. MUM 2011*, Dec. 2011, pp. 134–43.

Biographies

Shin-Ming Cheng [S'05, M'07] received his B.S. and Ph.D. degrees in computer science and information engineering from National Taiwan University, Taipei, in 2000 and 2007, respectively. He was a postdoctoral research fellow at the Graduate Institute of Communication Engineering, National Taiwan University, from 2007 to 2012. Since 2012, he has been with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, as an assistant professor. His current research interests include mobile networks, wireless communication, cyber security, and complex networks. He was a recipient of the IEEE PIMRC 2013 Best Paper Award and the 2014 ACM Taipei/Taiwan Chapter K. T. Li Young Researcher Award.

Pin-Yu Chen [S'10, M'16] received his B.S. degree in electrical engineering and computer science (undergraduate honors program) from National Chiao Tung University, Taiwan, in 2009, his M.S. degree in communication engineering from National Taiwan University in 2011, and his Ph.D. degree in electrical engineering and computer science and M.A. degree in statistics from the University of Michigan Ann Arbor in 2016. He is currently a research scientist of AI Foundations Group at IBM Thomas J. Watson Research Center. His research interest is graph data analytics and their applications to data mining, machine learning, and cyber security. He is a member of the Tau Beta Pi Honor Society and the Phi Kappa Phi Honor Society, and was the recipient of the Chia-Lun Lo Fellowship from the University of Michigan Ann Arbor. He was also the recipient of the IEEE GLOBECOM 2010 GOLD Best Paper Award and several travel grants, including IEEE ICASSP 2014 (NSF), IEEE ICASSP 2015 (SPS), IEEE Security and Privacy Symposium 2016, Graph Signal Processing Workshop 2016, and ACM KDD 2016.

Ching-Chao Lin received his B.S. degree in computer science and information engineering from National Taiwan University of Science and Technology in 2015. He is currently working toward an M.S. degree in computer science and information engineering at the National Taiwan University of Science and Technology. His research interests include cyber security and wireless networks.

Hsu-Chun Hsiao is an assistant professor in the Department of Computer Science and Information Engineering, and the Graduate Institute of Networking and Multimedia at National Taiwan University. She also holds an adjunct assistant researcher position in the Center of Information Technology and Innovation at Academia Sinica. She completed her B.S. (2006) and M.S. (2008) at National Taiwan University, and her Ph.D. at Carnegie Mellon University (2014). Her research interests include network security, anonymity and privacy, and applied cryptography.

Characterizing the HTTPS Trust Landscape: A Passive View from the Edge

Gustaf Ouvrier, Michel Laterman, Martin Arlitt, and Niklas Carlsson

The authors present an overview of the current trust landscape and provide statistics to illustrate and quantify some of the risks facing typical users. Using measurement results obtained through passive monitoring of the HTTPS traffic between a campus network and the Internet, they provide concrete examples and characterize the certificate usage and trust relationships in this complex landscape.

Abstract

Our society increasingly relies on web-based services like online banking, shopping, and socializing. Many of these services heavily depend on secure end-to-end transactions to transfer personal, financial, and other sensitive information. At the core of ensuring secure transactions are the HTTPS protocol and the “trust” relationships between many involved parties, including users, browsers, servers, domain owners, and the third-party CAs that issue certificates binding ownership of public keys with servers and domains. This article presents an overview of the current trust landscape and provides statistics to illustrate and quantify some of the risks facing typical users. Using measurement results obtained through passive monitoring of the HTTPS traffic between a campus network and the Internet, we provide concrete examples and characterize the certificate usage and trust relationships in this complex landscape. By comparing our observations against known vulnerabilities and problems, we highlight and discuss the actual security that typical Internet users (e.g., the people on campus) experience. Our measurements cover both mobile and stationary users, consider the involved trust relationships, and provide insights into how the HTTPS protocol is used and the weaknesses observed in practice. While the security properties vary significantly between sessions, out of the 232 million HTTPS sessions we observed, more than 25 percent had weak security properties.

Introduction

We are living in an information society in which organizations and in dividual users frequently must rely on the security and privacy offered by the HTTPS protocol.

With HTTPS, any type of data that can be exchanged between a client and a server using regular HTTP requests and responses are transferred over an end-to-end connection encrypted using Transport Layer Security (TLS) or its predecessor, Secure Sockets Layer (SSL). With increased value of the information exchanged over the Internet, it is perhaps not surprising that HTTPS usage is increasing [1]. HTTPS can provide secure end-to-end transfers of money and other sensitive information, and is often used by authentication-based services such as online banking, shopping sites, and social networking services.

With increased awareness of wiretapping and manipulation of network traffic, HTTPS has also become common among services that have not traditionally used secure end-to-end connections, including streaming services such as YouTube and Netflix that account for a majority of current Internet traffic volume. This trend has been further augmented by free solutions from the Let’s Encrypt project [1].

In addition to the TLS/SSL protocol suite, the security of HTTPS relies on a set of complex “trust” relationships. For example, consider a simple scenario (Fig. 1a) in which a user accesses a single-server website using HTTPS. In this scenario, the user must trust his/her browser, the server with which the browser will exchange information, a certificate presented by the server to the client, the third-party certification authority (CA) who vouches that the certificate belongs to the corresponding server/domain, as well as the strength of the keys, hashes, and cryptographic algorithms (determined by a pairwise negotiated cipher suite) used for connection establishment and data exchange.

Naturally, not all relationships are equally trustworthy. In particular, typical web sessions involve a diversity of servers, CAs, certificates, cryptographic algorithms, keys, and hashes. Measurements are therefore important to understand the practical vulnerabilities to which a user is exposed. In this article, we first untangle the trust relationships in this landscape and then present a data-driven characterization of the trust landscape and the security risks observed in practice for the different relationship types. All results and findings are discussed in the context of known vulnerabilities and previous measurement studies.

For this study, we passively collected and analyzed all HTTPS usage between a university campus network and the Internet for a week-long period, capturing more than 232 million HTTPS sessions. We develop a novel session-based labeling methodology that allows us to perform per-device and per-OS type analysis without requiring IP addresses to be recorded. The dataset includes traffic from both mobile and stationary users, and provides an overview of the actual security that typical Internet users (on campus) experience when browsing the web, as well as the trust relationships that are often invisible to the end user.

Our characterization shows that many per-session properties of mobile and stationary clients are almost identical, and highlights current weaknesses in each of the relationships in the land-

Digital Object Identifier: 10.1109/MCOM.2017.1600981

Gustaf Ouvrier and Niklas Carlsson are with Linköping University; Michel Laterman is with SAP; Martin Arlitt is with the University of Calgary.

0163-6804/17/$25.00 © 2017 IEEE

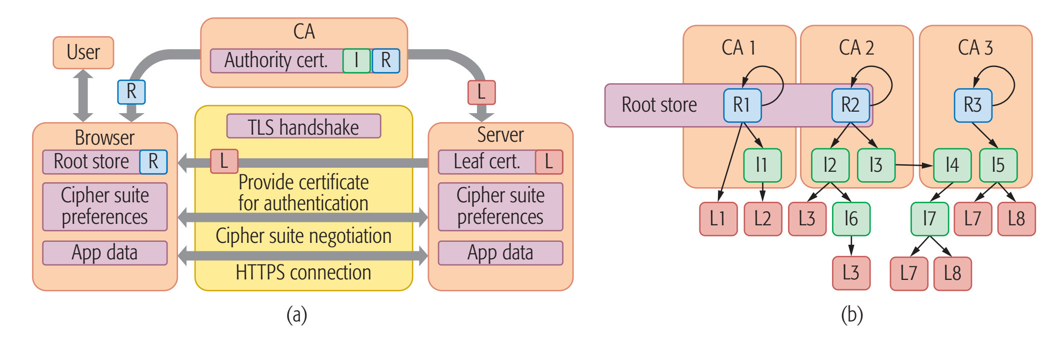


Figure 1. Example trust relationships in an HTTPS scenario: a) relationships and involved parties; b) certificate landscape example.

When the CA issues a certificate, it vouches for the identity of the website by digitally signing it using one of its authority root certificates. This creates a cryptographic binding between the website's identity and the public key contained inside the certificate, which can be validated using the CA's root certificate.

scape, including that clients often use older browser versions, that there is a skewed popularity among the CAs, and that there is a lack of adherence to best practices by all involved parties. For example, ciphers are frequently used by both clients and servers long after their weaknesses have become public knowledge, certificates often have long validation periods (providing more time to crack them), and CAs often issue certificates that allow their signing ability to be further delegated. Furthermore, both multi-domain and wildcard certificates (which allow many domains and sub-domains to share the same certificate) are very popular, suggesting that website owners often prioritize convenience and economy over security. Overall, our per-session quality evaluation shows that a significant portion of the sessions have relatively weak security.

Background and Related Work

We begin with an overview of a functionally successful HTTPS session and describe the role of the involved parties in each trust relationship.

Certificate Authorities

At the center of the HTTPS landscape are the CAs. The CAs are "trusted" third-party organizations that are responsible for issuing digital certificates and administering their validity. Before issuing a certificate, CAs are required to verify the identity of the website and make sure that the requester is the owner. When the CA issues a certificate, it vouches for the identity of the website by digitally signing it using one of its authority root certificates. This creates a cryptographic binding between the website's identity and the public key contained inside the certificate, which can be validated using the CA's root certificate.

A web browser contains a root store, which is a selection of root certificates that it trusts. Furthermore, a browser generally trusts all certificates L that can be validated using a root certificate R from its root store. The reason for the root store is to relieve users, who often do not even know about certificates, from having to specify which CAs they trust themselves.

Transport Layer Security Handshake

When establishing a new HTTPS connection, the client (i.e., the browser) and server must complete a TLS handshake in which they agree upon details for the session, including which TLS/SSL version to use and which cryptographic algorithms (i.e., cipher suite) to employ. During the handshake process (Fig. 1a) the server presents an X.509 digital certificate (L), which is used to authenticate the server to the client. This certificate contains information about the identity of the server (e.g., a website domain name), a validity period, and a public key, and is signed by a CA with a root certificate (R). Given a certificate, the client checks that the identity matches the target server, that the certificate is in its validity period, and that the CA's digital signature is valid. If validation is successful, a session key is determined, and an end-to-end encrypted communications channel is opened.

Certificate Landscape Example

Each CA controls a handful of authority certificates, called *root certificates*, which are validated by them directly. The root certificates are used to issue further certificates, which can be either another authority certificate, called an *intermediate certificate*, or an end-entity certificate, called a *leaf certificate*. Every new intermediate certificate can issue further intermediate certificates resulting in a chain of trust, referred to as a certificate chain. Figure 1b shows a simple certificate landscape example. Here, R1, R2, and R3 are root certificates to their respective CAs, while I1, I2, I3, and I5 are intermediate certificates directly signed by the root certificates. L1–L8 are leaf certificates for individual example websites, vouched for by the corresponding CAs. The case of I3 signing I4 is called a cross signing and can be useful when one CA resides in the root store and the other does not. For example, the Let's Encrypt project (https://letsencrypt.org/, last accessed March 2017) uses cross signing by IdenTrust (large CA) so that they can reach more users, long before they themselves are included in all root stores.

In our example only R1 and R2 reside in the root store, so a website with certificates L1–L6 will be considered trusted by the browser, while L7–L8 are untrusted. The use of intermediate certificates helps spread the workload of identity checking and signing procedures, as globally operating CAs can delegate such tasks to local intermediate CAs. Most importantly, using easily removable/changeable intermediate certificates for online business signing purposes also allows the private keys of root certificates to be kept offline. However, since every intermediate certif-

Browser	Browser's share of all sessions	Breakdown of browser version usage			
		Up-to-date	One behind	Two behind	Older
Chrome	51.48%	22.40%	64.68%	1.82%	11.12%
Safari	22.55%	0.62%	41.52%	10.36%	47.50%
Firefox	18.98%	0.00%	0.00%	66.90%	33.10%
Internet Explorer	6.72%	2.20%	45.76%	13.48%	38.56%

Table 1. Browser usage and version distribution (October 11–17, 2015).

icate can be used to issue a valid certificate for any domain, each intermediate certificate comes with its own risks and attack surface [2].

Trust Relationship Breakdown

To untangle the trust relationships, consider our original single-server scenario (Fig. 1a). First, the user must trust the browser and its implementation of HTTPS, and that the browser is up-to-date against the latest known security vulnerabilities. Second, the browser (and implicitly the user) needs to trust all CAs in the browser's root store. If a single trusted CA is compromised and starts generating certificates for non-trusted servers, this significantly compromises the security that a browser provides. With many available CAs, each with their own strengths and weaknesses, there are significant differences in which CAs distinct browsers select to trust. Disparate web services may also select different CAs for their certificates.

Third, the browser needs to trust the server with which it communicates. This trust is often built around X.509 certificates signed by the CAs in the root store or by other trusted entities to which the CAs have delegated part of this responsibility. These chained trust relationships are further complicated by (chained) certificates often valid for different time periods and difficult to invalidate when trust relationships are broken.

Finally, the browser must trust the cipher suite negotiated between the browser and server during the TLS handshake, where the cipher suites determine a combination of a key exchange, encryption, and message authentication code (MAC) algorithm. Ciphers have different cryptographic strengths, and many ciphers in use today have known vulnerabilities and can therefore pose a significant risk to the confidentiality of the information transferred between the two parties.

Related Work

Both active and passive measurements have been used to analyze particular aspects that go into a secure HTTPS connection.

Many of these works have examined attacks targeting particular aspects of the connection establishment, including the key exchange [3], targeted ciphers [4], and MACs [5], to compromise the security of the HTTPS connections. This article characterizes and discusses the security experienced by regular users within the context of some of these attacks.

We also use passive measurements by Holz *et al.* [6] (published in 2011) and by Durumeric *et al.* [7] (published in 2013), together with our own measurements, as reference points for a longitudinal discussion. In addition to complementing these studies with more recent data points and a tutorial-style overview of the landscape, we also present complementary new analyses based on our novel session-based labeling, for example, which allows us to compare and contrast the heterogeneous security offered to both mobile and stationary devices. The remaining references are used to support claims.

Methodology and Data Collection

Our dataset was collected by passively monitoring the Internet traffic to/from the University of Calgary, Canada, at the university's multi-gigabit-per-second ingress/egress link. We used the Bro (https://www.bro.org/, last accessed March 2017) network security monitor to log specific information about the non-encrypted part of the TLS/SSL handshake, including all digital certificates sent.

We filter sessions based on IP prefix and focus on the traffic with servers located outside the campus. To distinguish between mobile and stationary users, we map the user-agent strings observed in HTTP sessions to IP addresses for five-minute rolling windows. Assuming that HTTP and HTTPS sessions from the same IP address within a window use the same user agent, this allows us to create a mapping between user agents (seen in the HTTP data) and HTTPS sessions. After making the mapping, the IP addresses are removed. The methodology leverages the fact that typical web sessions, even when only visiting a single website, involve requests to many servers; some accessed with HTTP and some with HTTPS. By extracting OS and browser related information from the user agent, this novel methodology allows us to perform per-device and per-OS type analysis, while preserving user privacy. Due to limited observed differences in certificate usage, in this article, we primarily focus on the distinction of mobile and stationary users.

Our dataset was collected during a week-long period (October 11–17, 2015). In total, we observed 232,640,189 sessions using TLS/SSL, 67,664 unique certificates, and 552,387,188 certificate exchanges. Of the sessions, 67.6 percent contained (one or more) certificates, while the rest were session resumptions. Using known user-agent strings, we identified 46,913,633 mobile HTTPS sessions (53.5 percent iPhone/iPad/iPod and 45.6 percent Android) and 109,549,848 sessions from stationary clients. Both subsets have similar resumption ratios (29.9 and 32.1 percent, respectively). Of the 67,664 unique certificates, 750 were authority certificates, and the remaining 98.9 percent were leaf certificates.

Trust Relationship Analysis

Older Browser Versions: The browser plays a key role in the HTTPS landscape. Despite the popular browsers using automatic updates and/or frequent reminders to upgrade to the latest versions, many clients still use outdated browsers. To illustrate this, Table 1 shows the fraction of sessions associated with different browser versions. Here, the latest officially released stable version is considered "up-to-date," and versions that do not have the latest security update are considered behind. We do not take into account Beta or developer versions. With the exception of Chrome, where 64.7 percent of the sessions are (only) a single version behind, the majority of sessions are using browsers that are at least two versions behind. These results highlight a significant delay in the rollout of new security updates.

Skewed Usage toward a Few CAs: With a few exceptions, any organization with control of an authority certificate can issue certificates for any domain. If a single authority certificate is compromised, the whole system becomes vulnerable [8]. Although our dataset includes 750 authority certificates, we find that the vast majority of non-self-signed leaf certificates are issued by only a handful of organizations. The top five organizations signing leaf certificates are Comodo CA Limited (with 22.9 percent of the sessions), Go Daddy (18.1 percent), GeoTrust (16.3 percent), DigiCert (9.4 percent), and GlobalSign (6.8 percent). Part of this skew is due to rich-get-richer effects as buyers often select popular CAs as these may be less likely to be removed from root stores [8].

To meet increasing market demands, many identity checks have become less stringent over time. Extended validation (EV) certificates were introduced to help restore the resulting waning user trust in certificates. EV certificates are intended to follow stricter issuing criteria needed by organizations where secure communication is essential. Interestingly, when considering EV certificates, the skew is both higher and more toward different CAs than for regular certificates, with Symantec Corporation (56.2 percent) and DigiCert (27.6 percent) making up the majority of the observed EV sessions (and 37.9 percent of the certificates). While the top issuers are somewhat different than for leaf certificates in general, we note that EVs only are observed in 4.94 percent of the leaf certificates and 6.27 percent of all sessions.

Some trust in Symantec may be inherited through the acquisition of Verisign's authentication business unit in 2010. Unfortunately, even the most highly used (and trusted) CAs can be compromised or make mistakes that degrade the overall security. For example, recently it was discovered that Symantec had issued test certificates for 76 domains they did not own (including Google domains) and another 2458 unregistered domains [9] (published October 28, 2015; last accessed March 2017.) Google has since demanded that Symantec logs its certificates in publicly auditable certificate transparency (CT) logs [10].

Weak Cryptography in Certificates: Despite being susceptible to known attacks and CAs no longer signing new certificates with SHA1, SHA1 (with RSA encryption) is the second most used signature algorithm in our dataset. SHA1 is responsible for signing 24.9 percent of the leaf certificates and 50.3 percent of the authority certificates. The recommended SHA256 (with RSA encryption) signing algorithm has replaced SHA1 more so far for leaf certificates (72.3 percent SHA256) than authority certificates (42.8 percent SHA256). Although we observed improvements compared to the 98.7 percent share of SHA1 that Durumeric *et al.*, [7] observed in 2013, there is still a long way to go. While decisions by Mozilla, Microsoft, and Google, for example, to phase out SHA1 (e.g., not showing a padlock symbol or to various degrees blocking SHA1 usage) may speed up this process, there have been setbacks in the outphasing as some of the browser companies have softened their decisions, including Mozilla re-enabling support for SHA1 in Firefox [11]. Service providers such as Facebook and Twitter have also suggested a delay in the phaseout of SHA1, due to concern that millions of users with older devices would lose access to their services. In addition, we observed 10 authority certificates (1.33 percent) and 97 leaf certificates (0.14 percent) still using MD5, almost seven years after Sotirov *et al.* [5] successfully created a rogue CA certificate. Also, we observed a non-negligible number of authority (1.33 percent) and leaf (5.61 percent) certificates using weak 1024-bit RSA keys, which the National Institute of Standards and Technology (NIST) recommended to stop using in 2013 [12].

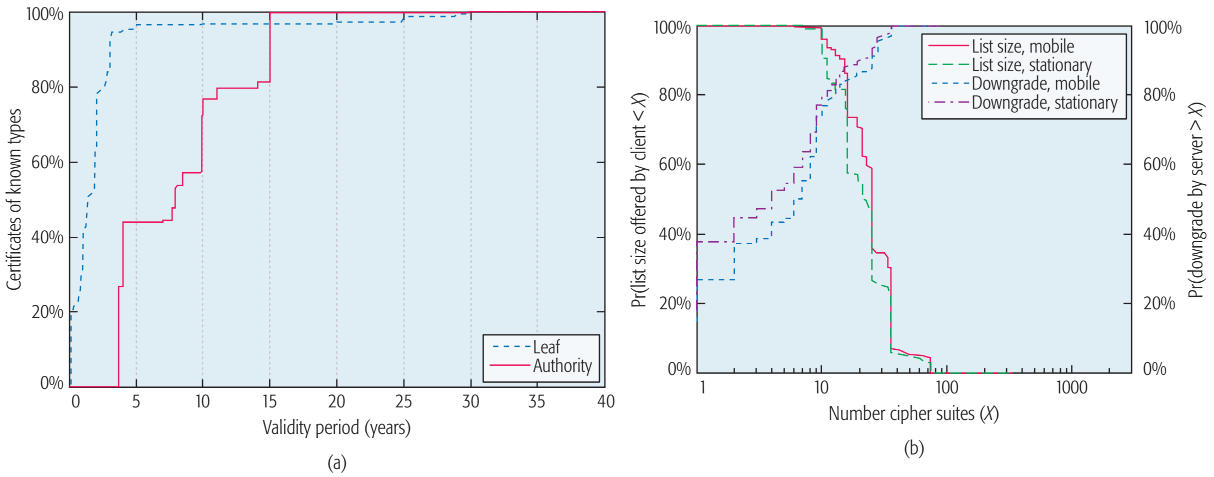
When inspecting EV certificates further, we did not find any certificates using weak keys (length less than 2048-bit), but did not discover any significantly stronger keys either. Most EV certificates were signed using SHA256 (84.6 percent) or SHA1 (15.2 percent), with SHA1 being observed in 25.3 percent of the sessions.

Lack of Path-Length Constraints: The maximimum path length of a certificate is decremented for each non-self-issued certificate in the path. It limits the length of a potential certificate chain and the trust delegation that is possible. Despite providing an extra measure of protection from misuse and helping to mitigate mistakes like issuing authority certificates instead of leaf certificates, we observed that 26.7 percent of all authority certificates did not specify any path length constraints.

Wildcard and Multi-Domain Certificates: Wildcard certificates (valid for all subdomains; e.g., **.domain.com*) were used in over 70 percent of sessions. While the wildcard feature is convenient for administrators, it also poses the risk of validating rogue or buggy hosts [13]. Since the private key for a certificate must be stored on each machine, the attack surface increases with the number of machines and domains covered by a certificate. We also observed significant usage of the Subject Alternative Name (SAN) extension, which allows multiple domain names to be specified for a certificate. For example, 68 percent of unique certificates have at least two domain names, 20 percent of all certificate observations are for the 9 percent of certificates with at least 10 domain names, and 134 certificates (0.2 percent) had more than 100 subjects. Of these, the top five belong to GlobalSign (514 and 510 subjects), Google (two different certificates with 503 subjects), and Technische Universität München (429 subjects). In the top 10 we observed another four universities and another Google certificate.

Long Validity Period Durations: Due to continuous advances in computer and cryptographic technologies, certificates valid for an extended time period can quickly become viable targets for

To meet increasing market demands, many identity checks have become less stringent over time. Extended validation certificates were introduced to help restore the resulting waning user trust in certificates. EV certificates are intended to follow stricter issuing criteria needed by organizations where secure communication is essential.



(a)

(b)

Figure 2. Selected statistics for certificates and cipher suite downgrades: a) CDF of certificate validity periods; b) cipher suite list size and downgrades.

attack. Using shorter validity periods is therefore a good practice. Figure 2a shows the cumulative distribution function (CDF) of the validity period lengths of observed certificates. Typically, authority certificates have longer lifespans (e.g., 4, 10, and 15 years) than leaf certificates (e.g., 1, 2, or 3 years), but we also identified lifespans of up to 37 years. This is far beyond the predicted security lifespan for many certificates, according to NIST's current prediction that 112-bit security will be acceptable through 2030 [12].

We also validated the certificate chains during each non-resumption HTTPS session using the Mozilla root store. While most sessions were validated successfully (94.8 percent), a non-negligible share (4.2 percent) were not. About 1 percent of sessions contained self-signed certificates, and in a few cases the certificate was outside its validity period (21,819 expired and 4537 not yet valid).

Server Downgrades of TLS/SSL Version: The security of HTTPS heavily depends on which TLS/SSL version and cipher suite is used, where the cipher suite is a combination of a key exchange, encryption, and MAC algorithm. These details are negotiated during the TLS handshake. The browser first informs the server about the highest TLS/SSL protocol it supports and sends a list of supported cipher suites, ordered by preference. The server determines the final protocol and selects a cipher suite from this list. While the server typically chooses the version offered by the browser, we find that in 4 percent of the sessions the server downgraded to a lower version. Overall, the usage is almost completely divided between TLSv1.2 (80 percent) and TLSv1.0 (19 percent). Less than 0.1 percent of the sessions used SSL (v. 3 and v. 2).

Known Weaknesses in the Key Exchange: The key exchange algorithm in the protocol is used to derive the shared session key. In 2015 alone, two attacks were discovered targeting the key exchange algorithm. The FREAK and Logjam attacks exploit bugs in the TLS/SSL implementation to downgrade sessions of servers that still support RSA-EXPORT and DHE-EXPORT grade ciphers, respectively [3]. Such downgrades allow an attacker to passively eavesdrop on the session. In total, we identified 428 instances of TLS_RSA_EXPORT and 27 instances of TLS_DHE_EXPORT being used. In general, however, we find that a majority of sessions use elliptic curve Diffie-Hellman (e.g., 62.11 percent use TLS_ECDHE_RSA and 20.32 percent use TLS_ECDHE_ECDSA), and 83.65 percent of all sessions have "perfect forward secrecy" (i.e., previously recorded sessions would not be compromised by long-term keys being compromised in the future). While elliptic curves have long been recommended by security experts (and still are), it should be noted that ECDHE-based solutions in particular have recently come under scrutiny due to the influence that the National Security Agency (NSA) of the United States has in their design [3].

Weak encryption and MACs: While AES-128 and AES-256 are currently the most commonly used encryption algorithms, we discovered many cases where ciphers with weaker encryption are both offered and selected. For example, despite being prohibited from use in TLS [14] and viable attacks against RC4 being published in 2013 [4], the RC4 cipher is still offered (54.7 percent) and used (7.6 percent) in many sessions. On a positive note, the overall numbers are down from what Holz *et al.* [6] observed in 2011. Fewer concerns have been raised regarding MACs for data integrity and authenticity. This is consistent with the relatively shorter lifetime of session MACs (compared to certificate MACs) and the relatively strong (session) MACs that we observed, including SHA256 (48.4 percent), SHA1 (31.0 percent), and SHA384 (14.8 percent). While MD5 (5.9 percent) is no longer acceptable in situations where collision resistance is required, such as for digital signatures, it is not urgent to stop using MD5 in HMAC-MD5 schemes [15]. MD5 is almost exclusively used together with the RC4 cipher.

Downgrade of Ciphers: Referring to the CDFs in Fig. 2b, in more than 99 percent of all sessions the clients offer at least nine cipher suites and often substantially more. As these lists often include weak ciphers, these results suggest that clients often prioritize compatibility (by giving servers many options) over security. Unfortunately, servers typically do not pick the clients' top candidate and sometimes perform substantial downgrades. For example, the most preferred option is only chosen in 36 percent of all sessions, and in 20 percent of the cases an option outside the top 10 is selected.

To gain additional insights into the downgrades, we looked closer at in what position the RC4 cipher was when chosen by the server. In more than 80 percent of the cases it is outside the top 10, and in 60 percent of the cases it is outside the top 25, suggesting that the choice to use RC4 may be the result of poorly configured servers. We have also noticed cases where the servers appear to prioritize user experience over security by not turning away visitors not supporting strong security options.

Mobile vs. Stationary Users: In general, we have observed very small differences in per-session statistics and corresponding distribution when comparing mobile and stationary users. For example, consider the distributions shown in Fig. 2b. Also, the cipher suites (offered and used) and the certificate chain length distributions are almost identical for mobile and non-mobile users, with both types typically seeing a chain length of two (44.8 and 45.2 percent) or three (47.4 and 47.8 percent). In both cases we observed chain lengths of up to 21. This can perhaps partially be explained by the client classes (as an aggregate within the class) producing highly correlated communication patterns. For example, a more detailed analysis of the visited websites shows that the relative popularities of the visited domains (as measured by their popularity ranks) are very similar for the mobile and non-mobile users on campus. Perhaps the largest difference between mobile and stationary sessions were observed for RC4 usage. Here, we observe a slightly higher usage of RC4 among mobile sessions (9.23 percent) compared to the stationary sessions (6.92 percent). However, in general our observations are consistent and very similar across the two classes.

Session Quality Evaluation

We conclude our analysis by evaluating the observed HTTPS session qualities. Figure 3 summarizes our results using a four-level classification. Here, a session is classified as *Acceptable* if it:

- Uses protocol version TLSv1.0 or better
- Uses a NIST approved encryption cipher with at least 112-bit security strength
- Has a version-3 leaf certificate with a validity period of at most 25 months (2 years and a grace period)
- Uses a signature algorithm SHA1 or better
- Uses public keys with at least 112-bit security strength

Sessions that do not satisfy these minimum requirements are classified as *Weak*. *Good* sessions further require that the certificate uses a stronger signature algorithm than SHA1 and either a key exchange algorithm with "perfect forward secrecy" or encryption with at least 128-bit security strength. Finally, for *Strong* sessions, we further restrict the protocol version to TLS v. 1.1 or better, the certificate validity period to at most 13 months, and the public key to use at least 128-bit security strength. This last class is included to illustrate roughly where we have observed the upper bound region of what is used in practice (hence the much smaller usage observed here).

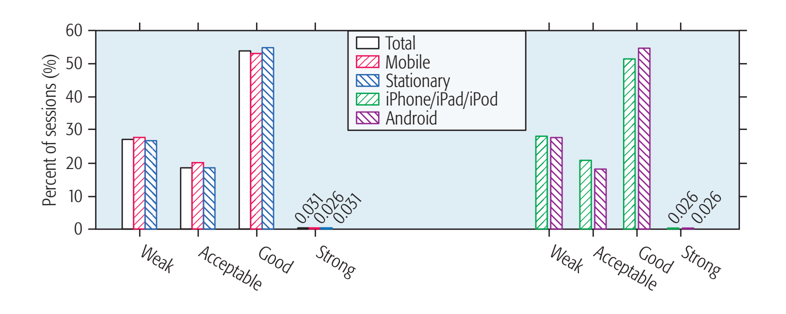


Figure 3. Session quality evaluation based on four-level security classification.

During our measurements, a majority of the sessions were classified as *Good* (53.8 percent), an additional 18.9 percent as *Acceptable*, but a very insignificant portion as *Strong* (0.03 percent). We again did not find any significant differences between the mobile and stationary client devices, or when comparing Apple and Android devices within the mobile category. While the bulk of sessions have at least *Acceptable* quality, the many *Weak* (27.3 percent) sessions cause concern.

Concluding Remarks

Using passive measurements, we have characterized and discussed current shortcomings in the different trust relationships that affect the security of online users. We have highlighted risks associated with the lack of adherence to best practices, including the slow outphasing of weak protocols and similarly slow adoption of new versions. For example, while modern browsers may be quick with security updates and patches, users typically use far from the latest versions. There is a significant skew in the organizations signing certificates, with differences between regular leaf certificates and EV certificates suggesting substantial differences in websites' trust in different CAs. Wildcard and multi-domain certificates are very popular, suggesting that websites often prioritize convenience or cost over security. Many certificates are valid for extended durations and do not limit path lengths to prevent authority certificates from further delegating signing ability. We have also seen that some clients offer broken ciphers (e.g., RC4), and servers sometimes choose them over better options. In general, we have observed limited differences between mobile and stationary clients, suggesting that our characterization is invariant to which of these two types of users is selected. Finally, our per-session quality evaluation showed that a significant portion (over 25 percent) have weak security quality. These results highlight the fact that many browsers and servers prioritize user experience over security.

References

[1] M. Aertsen *et al.*, "No Domain Left Behind: Is Let's Encrypt Democratizing Encryption?" *CoRR*, vol. arXiv:1612.03005v1 [cs.CR], Dec. 2016.

[2] B. Amann *et al.*, "No Attack Necessary: The Surprising Dynamics of SSL Trust Relationships," *Proc. ACSAC*, Dec. 2013, pp. 179–88.

[3] D. Adrian *et al.*, "Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice," *Proc. ACM CCS*, Oct. 2015, pp. 5–17.

[4] N. AlFardan *et al.*, "On the Security of RC4 in TLS," *Proc. USENIX Security*, Aug. 2013, pp. 305–20.

[5] A. Sotirov *et al.*, "MD5 Considered Harmful Today," *Proc. Annual Chaos Commun. Congress*, Dec. 2008.

[6] R. Holz *et al.*, "The SSL Landscape: A Thorough Analysis of the X.509 PKI Using Active and Passive Measurements," *Proc. IMC*, Nov. 2011, pp. 427–44.

[7] Z. Durumeric et al., "Analysis of the HTTPS Certificate Ecosystem," *Proc. IMC*, Oct. 2013, pp. 291–304.

[8] H. Asghari *et al.*, "Security Economics in the HTTPS Value Chain," *Proc. WEIS*, June 2013.

[9] R. Sleevi, *Sustaining Digital Certificate Security* (Google Security Blog); https://security.googleblog.com/2015/10/sustaining-digital-certificate-security.html.

[10] J. Gustafsson *et al.*, "A First Look at the CT Landscape: Certificate Transparency Logs in Practice," *Proc. PAM*, Mar. 2017.

[11] "Mozilla Re-Enables Support for SHA-1 in Firefox," *SecurityWeek News*; http://www.securityweek.com/mozilla-re-enables-support-sha-1-firefox: published Jan. 7, 2016; accessed Mar. 2017.

[12] E. Barker *et al.*, "Recommendation for Key Management, Part 1: General (Revision 3)," NIST Special Publication 800-57, July 2012.

[13] P. Saint-Andre, S. Santesson, and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)," IETF RFC 6125, Mar. 2011.

[14] A. Popov, "Prohibiting RC4 Cipher Suites," IETF RFC 7465, Feb. 2015.

[15] S. Turner and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms," IETF RFC 6151, Mar. 2011.

Biographies

Gustaf Ouvrier is an M.Sc. student at Linköping University, Sweden. His research interests include security, Internet measurements, and traffic characterization.

Michel Laterman received his M.Sc. in computer science from the University of Calgary, where he worked on network traffic monitoring. He currently works as a software developer at SAP.

Martin Arlitt [SM] is a principal research scientist at Hewlett Packard Enterprise, where he has worked since 1997. His general research interests are workload characterization and performance evaluation of distributed computer systems. His 80+ research papers have been cited over 9600 times (according to Google Scholar). He has 31 granted patents and more pending. He is an ACM Distinguished Scientist and an adjunct assistant professor at the University of Calgary.

Niklas Carlsson is an associate professor at Linköping University, Sweden. He received his M.Sc. degree in engineering physics from Umeå University, Sweden, and his Ph.D. degree in computer science from the University of Saskatchewan, Saskatoon, SK, Canada. His research interests include design, modeling, characterization, and performance evaluation of distributed systems and networks.

Scalable Traffic Sampling Using Centrality Measure on Software-Defined Networks

Seunghyun Yoon, Taejin Ha, Sunghwan Kim, and Hyuk Lim

ABSTRACT

With regard to cyber security, pervasive traffic visibility is one of the most essential functionalities for complex network systems. A traditional network system has limited access to core and edge switches on the network; on the other hand, SDN technology can provide flexible and programmable network management operations. In this article, we consider the practical problem concerning how to achieve scalable traffic measurement using SDN functionalities. Less intrusive traffic monitoring can be achieved by using a packet sampling technique that probabilistically captures data packets at switches, and the sampled traffic is steered toward a traffic analyzer such as an IDS on SDN. We propose the use of a centrality measure in graph theory for deciding the traffic sampling points among the switches. In addition, we discuss how to decide the traffic sampling rates at the selected switches. The results of the simulation and SDN testbed experiments indicate that the proposed sampling point and rate decision methods enhance the intrusion detection performance of an IDS in terms of malicious traffic flows in large-scale networks.

INTRODUCTION

As the population of Internet users continues to grow, the number of devices connected to the Internet is increasing rapidly. Because of this explosive expansion of the network scale, there is huge demand for flexible and scalable network management. Moreover, cyber security for home and enterprise networks has become more important because our daily data applications and access to services such as banking, shopping, business, education, and transportation are provided via the Internet. One of the rapidly increasing threats is ransomware, which is computer malware that executes a cryptovirology attack and demands a ransom payment to restore the damage [1, 2]. These malware propagations via the Internet can be prevented by capturing suspicious packets on the network and inspecting them by using security application solutions such as an intrusion detection system (IDS). Usually, IDSs are deployed on networks in data centers. These systems are connected to edge or core switches, and are used to monitor network traffic patterns and inspect data packets in order to detect malicious activities.

A software-defined network (SDN — http://www.opennetworking.org/sdn-resources/sdn-definition) is a promising technology that can provide flexibility, robustness, and programmability. Briefly, the principal concept of SDN is to decouple the network control plane from the data forwarding plane. Forwarding decisions in traditional networks are made by a routing algorithm in each switch; on the other hand, the controller on an SDN is responsible for controlling the forwarding operations of the switches in a centralized manner. In this regard, the OpenFlow (http://www.openflow.org) protocol is one of the most popular protocols used for communication between the SDN controller (for the control plane) and SDN-enabled switches (for the data plane). Due to this flexibility and programmability, SDN technology can be applied to cyber security network applications as well as high-performance data center networks. For example, the SDN controller can enable a switch to duplicate the traffic flow of interest and steer the traffic toward a traffic collector by simply updating the forwarding table of the switch via an OpenFlow protocol. Network programmability such as traffic duplication and rerouting can facilitate the implementation of traffic monitoring operations for cyber security. In [1], a ransomware mitigation method that exploits SDN functionalities was proposed. It uses the SDN functionality for forwarding/steering traffic in order to inspect all the Domain Name Service (DNS) traffic packets (which may include the DNS query for ransomware proxy servers) and the traffic duplication functionality to implement the DNS packet inspection without incurring delays in the DNS response time.

Another important problem associated with network traffic monitoring is the acquisition of network traffic packets in a less intrusive manner. If every packet belonging to a traffic flow is captured and forwarded to a traffic collector, the amount of traffic that is newly generated for traffic monitoring purposes would be the same as that of the original traffic, and may cause significant congestion on the network. For less intrusive monitoring, it would be desirable to selectively capture traffic packets rather than capturing every packet at the switches. This method is known as *packet sampling* and can be implemented in several ways. Given information about the network topology and traffic flows, it is essential to decide where to sample traffic and how much traffic is sampled at each of the selected switches when the total amount of sampled traffic is bound for

The authors consider the practical problem concerning how to achieve scalable traffic measurement using SDN functionalities. Less intrusive traffic monitoring can be achieved by using a packet sampling technique that probabilistically captures data packets at switches, and the sampled traffic is steered toward a traffic analyzer such as an IDS on SDN. The authors propose the use of a centrality measure in graph theory for deciding the traffic sampling points among the switches.

The authors are with the Gwangju Institute of Science and Technology.

Digital Object Identifier: 10.1109/MCOM.2017.1600990

0163-6804/17/$25.00 © 2017 IEEE

For network traffic monitoring, there is a trade-off relationship between the network resource overhead and the amount of information acquired for traffic inspection. As cyber threats and attacks become more complicated, it is desirable that network traffic monitoring systems are able to support the FPC functionality for cases when traffic monitoring is required for malware inspection.

less intrusive traffic monitoring. In terms of deciding the sampling points, we propose the use of a centrality measure in graph theory for exploiting the network topology and flow information. The use of a centrality measure enables us to select switches with relatively higher importance. Once a subset of switches is chosen, packets passing through each switch can be sampled at a certain rate in a fair manner. The results of the simulation and SDN testbed experiments show that the proposed approach can significantly reduce the sampling points on the network while retaining the traffic inspection performance for malicious traffic such as that containing viruses and ransomware in a large network.

Network Traffic Monitoring

Network traffic monitoring includes various administrative operations to acquire network traffic information such as the routing paths of traffic flows, the traffic volume, the network/transport layer protocol types of the traffic flows, and the payload size distribution of the packets in each flow. The use of these various types of traffic information makes it possible to infer the network topology and the network resource status including the packet loss rates and congestion levels at the switches and routers in the network. Once information about the network traffic is gathered, it can be analyzed using statistical and information-theoretical methodologies for network operation and management. The analysis of the dynamic patterns resulting from changes in the traffic statistics allows a network administrator to detect abnormal operations of network links and nodes and to precisely identify faulty links and nodes. Subsequently, an appropriate management operation can be conducted to resolve the networking problems. For example, OpenNetMon was proposed to acquire the flow-level network status such as throughput, delay, and packet loss rate using the OpenFlow protocol for fine-grained traffic engineering [3]. In addition, network traffic monitoring is also an essential function of cyber security because malicious network activities such as distributed denial of service (DDoS) and port scanning generate their own unique traffic patterns. Thus, security applications can use the traffic patterns to defend the network system against malicious threats. In [4], it was proposed to combine the OpenFlow and sFlow (http://www.sflow.org) protocols to gather flow-level traffic statistics in order to detect and identify network anomalies such as DDoS, worms, and port scanning attacks.

The use of network traffic monitoring requires the data packets belonging to each traffic flow to be captured in order to acquire information about these traffic flows. We consider two approaches: *partial-packet capturing* (PPC) and *full-packet capturing* (FPC). The former approach involves capturing and storing the header of a packet because it contains the most useful information about the flows. Optionally, PPC may capture the first several bytes of the packet payload as well as the header to obtain additional information about the contents of the packet payload. sFlow and NetFlow (http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html) are the most popular monitoring solutions for PPC. Because PPC captures at most the first 200 bytes of each packet, the size of each report for the captured packets is much smaller than that of the original packets. If the reports of multiple packets are compressed before delivery to the traffic collector, the amount of additional traffic generated for network monitoring is further reduced. However, PPC can be limitedly used for malware inspection such as the detection of a computer virus, malicious code injection, and ransomware detection. On the other hand, FPC captures both the header and payload of a packet. Because it is possible to analyze the payload as well as the information specified in packet headers, a variety of network threats can be inspected and prevented. For example, an IDS can be deployed to inspect the payloads of data packets to detect potential malware using a signature-based matching method, which compares the payload of input packets with a number of known malicious binary patterns [5]. However, FPC causes the amount of traffic on the network to double, and the resulting network congestion may disrupt the traffic if every packet is captured, and the header and payload of each packet are duplicated.

For network traffic monitoring, there is a trade-off relationship between the network resource overhead and the amount of information acquired for traffic inspection. As cyber threats and attacks become more complicated, it is desirable that network traffic monitoring systems are able to support the FPC functionality for cases when traffic monitoring is required for malware inspection.

Network Traffic Monitoring Location

The location at which network traffic monitoring is performed by capturing traffic packets is an important factor that has a significant impact on the monitoring performance. Ideally, any point on a network (i.e., all the switches and routers) would need to be accessible to capture the network flows for traffic inspection. In traditional networks, the number of traffic monitoring devices (e.g., network tap and port mirror) is quite limited, and feasible locations at which the devices are deployed are also restricted.

Figure 1 shows two possible locations for traffic monitoring, edge switches, and core switches. Since the edge switches are directly connected to client devices, the number of flows at each switch is relatively small in comparison to that in switches on the network backbone. As a result, it is easy to achieve fine-grained traffic monitoring for the individual flows that pass through each switch. However, since one capturing device must be deployed at each edge switch, the number of capturing devices required for network-wide traffic monitoring would be considerable. This increase in the number of required capturing devices would incur management overhead, and may not be a feasible solution for a large network. As shown in Fig. 1, the core switches constitute the backbone of the network, and can be connected to the network of an Internet service provider (ISP). Since each core switch serves a number of flows at a high rate, high-performance capturing devices should be used for full-rate traffic capturing. However, it would not be possible to identify each traffic flow at the core switches without missing packets in real time. Note that the number of devices used for traffic capturing is usually smaller than that required for traffic capturing at edge switches, and they can be more easily managed in a centralized manner. In addition, as shown in Fig. 1, at the core switches, it would not be possible to capture those traffic flows that only pass through edge switches. Moreover, some traffic flows may be unnecessarily captured more than once at intermediate switches.

Figure 2 shows SDN-based traffic capturing. SDN technology provides more flexible traffic monitoring by mirroring and rerouting the traffic flows of interest. SDNs do not necessitate discrimination between edge switches and core switches. SDN-based traffic capturing obviates the need to exploit hardware-based capturing devices; this capturing method relies on OpenFlow to capture traffic packets. In practice, this simply involves updating the flow table of each switch using OpenFlow. In [6], a fast traffic monitoring architecture (named Planck), which leverages the port mirroring feature of switches, was proposed. A high sampling rate is achieved by dividing N ports in a switch into m monitor ports and $(N - m)$ data ports, and data traffic is forwarded to one of the monitoring ports. However, instead of using the mirroring feature of the switch, if it is necessary to monitor a specific traffic flow and capture data packets belonging to the particular flow on SDN, the controller can simply duplicate the traffic flow and forward the flow to a specific port of the switch by using the "OUTPUT" action of OpenFlow. Once the duplicated traffic is forwarded to the port, it can be steered to a traffic collector by the SDN controller as for a normal traffic flow.

Probabilistic Packet Sampling

Probabilistic packet sampling is widely used in both PPC and FPC for traffic monitoring to avoid excessive traffic overhead on the network due to traffic duplication. Instead of capturing every packet passing through a switch, it selectively captures a certain number of packets from the traffic flows according to a sampling policy. In [7], several sampling policies were proposed and discussed in detail. Among them, the systematic packet sampling (SPS) method captures every packet for a sampling duration from a starting point in time, and probabilistic packet sampling (PPS) is a method to selectively capture packets from traffic flows with a uniform or non-uniform probability p. For example, if p is 10 percent for PPS, only 10 percent of packets are captured, and the remaining 90 percent are simply discarded. In [8], an OpenFlow extension (named FleXam) was proposed to support both SPS and PPS methods with a variety of sampling options. In [9], OpenSample was proposed to exploit the probabilistic packet sampling of the sFlow protocol to achieve a fast flow-level traffic statistics measurement on SDNs. Figure 2 shows an example of probabilistic packet sampling on an SDN. For a given set of sampling rates, the SDN controller updates the flow table of each switch using OpenFlow. Then the sampled traffic is steered toward the traffic collector. As mentioned, since the sampled traffic flows are newly generated flows, the SDN controller has to compute less congested flow paths to the traffic collector and update the flow table of switches using the computed routing paths.

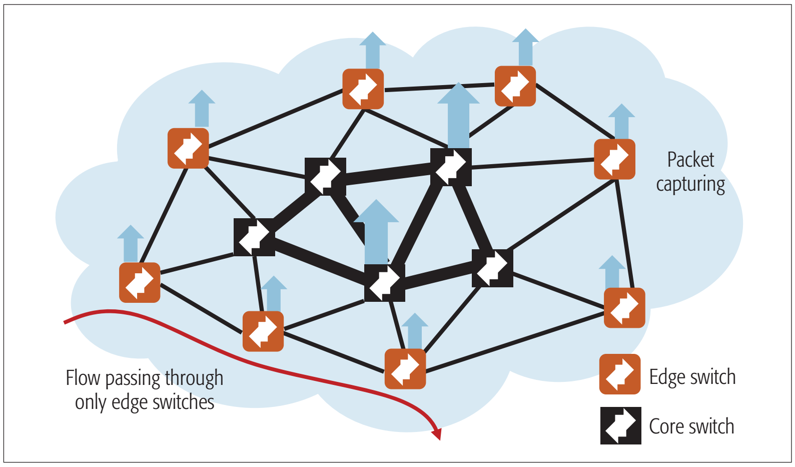


Figure 1. Packet capturing at edge switches vs. core switches.

The use of traffic sampling techniques can lead to a significant reduction in network overhead because sampled traffic duplication can be significantly decreased. However, there is the risk of useful information not being acquired from the discarded packets. Therefore, it is crucial to appropriately decide the sampling rates for traffic monitoring on the network. Two different approaches can be considered to decide the sampling rate: *per-flow (PF) sampling* and *per-switch (PS) sampling*. First, a different sampling rate can be applied to each flow. It is possible to adjust the sampling rate for each flow; therefore, fine-grained packet sampling is possible by increasing and decreasing the sampling rate. However, per-flow sampling of this nature may not be scalable with respect to the number of flows on the network because frequent updating of the flow table consumes substantial network resources, and the memory space reserved for the flow table in a switch is limited. In addition, the implementation and running complexities for per-flow operations are also significant [10]. Alternatively, each switch could use the same sampling rate for every flow that passes through the switch. Since the number of switches on a network does not change dynamically, the operational complexity is much lower than that of per-flow sampling. However, if the sampling rates are not properly determined, some flows are either excessively sampled at multiple switches or not sampled at all. The sampling rate decision for the per-switch sampling rates that approximate per-flow sampling rates is a challenging problem with high computational complexity.

For malicious traffic inspection, we consider a probabilistic full packet sampling on SDNs. If a larger number of traffic packets on the network were to be sampled with large sampling rates, it would be possible to acquire a higher level of traffic information for cyber security. However, it is more desirable to restrict the value of sampling rates for the following reasons.

Network Overhead

As the sampling rates increase, the amount of sampled traffic increases proportionally. Because the sampled traffic is forwarded to the traffic collector, it consumes network resources and may incur network congestion, which interferes with the data delivery of normal traffic flows. Therefore, the number of duplicated packets for traffic

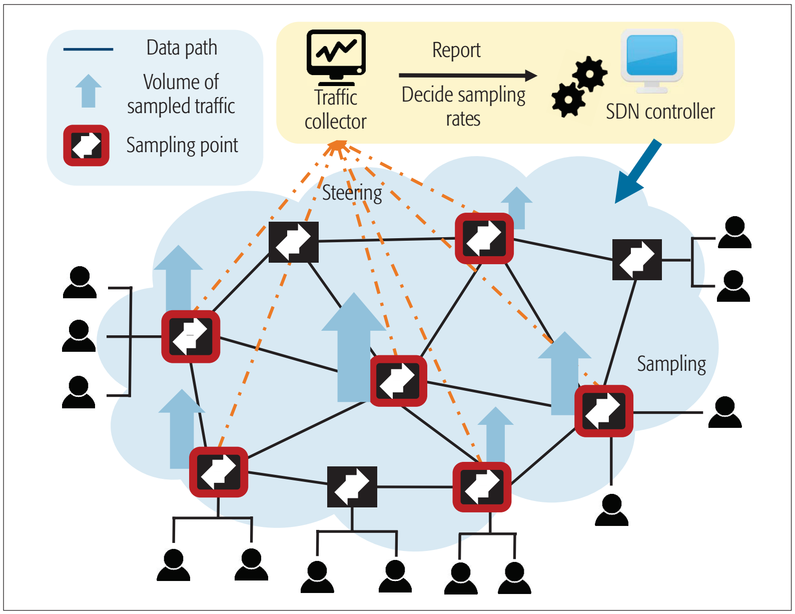


Figure 2. Packet capturing on SDN with OpenFlow-enabled switches.

sampling should be kept as small as possible for less intrusive traffic monitoring.

Limited Analysis Capability

Traffic inspection is conducted by analyzing the sampled traffic by security applications such as an IDS, which usually has limited processing capability. If the rate of incoming traffic to the IDS exceeds its capability, the IDS cannot process the incoming traffic without dropping packets. Therefore, the amount of sampled traffic should preferably not exceed the processing capacity of the IDS.

Sampling Point and Rate Decision

Sampling Point Decision

We propose a scalable packet sampling point decision scheme using a graph theoretic centrality measure, which qualifies the relative importance of switches on the network. Although packet samplings in every OpenFlow-enabled switch are possible using OpenFlow on SDN, it is not desirable in a large-scale network, because it may incur overhead caused by flow-table processing at the SDN controller. Instead, a subset of switches can be selected to perform packet sampling. In graph theory, centrality measures such as degree, closeness, and betweenness centrality indicate the relative importance of vertices in the graph. The importance of a particular vertex can be quantified based on its topological relationship among the other vertices such as the number of neighboring vertices and the number of edges required to reach each of the other vertices. The concept of centrality measures has been applied to various areas such as social networking to find the most influential person.

When computing the centrality measures, one may use the same network topology as for physical links. However, that approach may reflect the physical network topology itself rather than the characteristics of flows. For traffic monitoring, it would be more effective to use only active links that currently serve the traffic flows rather than all the physical links in the entire network. On SDNs, information about the flow paths is readily available by using SDN northbound application programming interfaces (APIs). Furthermore, it is more computationally efficient to calculate the centrality measures using the number of active links rather than all the physical links.

We propose the use of the betweenness centrality for sampling point decision, which is defined as the number of shortest paths that pass through the switch for all the node pairs [11]. If any pair of nodes has k possible shortest paths, each possible path is counted by $1/k$ rather than 1 in the computation of the betweenness centrality. The betweenness centrality of a switch is simply obtained by the number of flows that pass through the switch. Figure 3 shows a simple network topology with six switches and six flows and its corresponding flow information matrix. The betweenness centrality can easily be computed by the flow information provided by the SDN controller. Let $\mathbf{M}$ denote the flow information matrix $\mathbf{M} = [m_{ij}]$, where m_{ij} is a binary number. If the ith flow passes through the jth switch, $m_{ij} = 1$; otherwise, $m_{ij} = 0$. Note that the dimension of $\mathbf{M}$ is the number of flows by the number of switches. The betweenness centrality for the jth switch $c_j = \Sigma_i m_{ij}$, (i.e., the summation of the elements at the jth column). The betweenness centralities for the switches are given by 1, 3, 2, 2, 3, and 2.

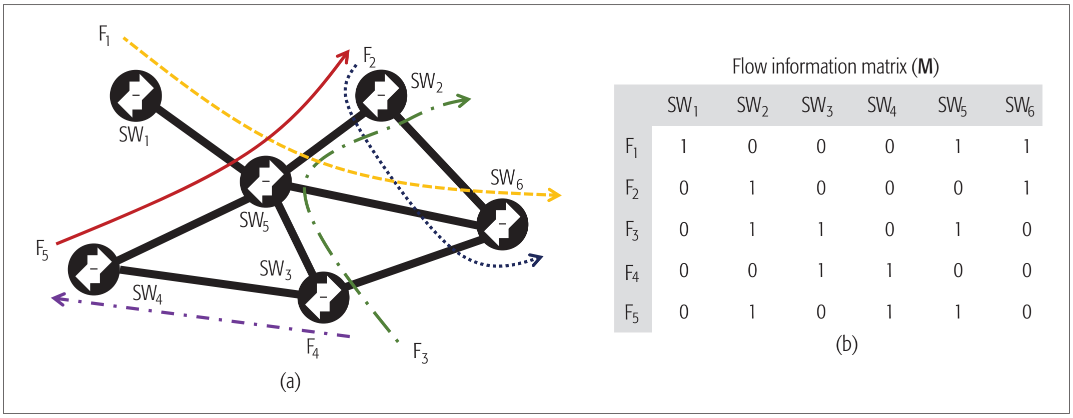
In addition to the original betweenness centrality, we propose a new extended betweenness centrality to avoid excessive sampling of traffic flows passing through a few bottleneck switches with high betweenness centrality. For example, if a number of flows pass through multiple bottleneck switches, the switches on the path would have a high betweenness centrality, and the flows would be unnecessarily sampled multiple times at the switches. Note that, in general, switches located near the core of the network have a high betweenness centrality. The extended betweenness centrality is iteratively computed. Given $\mathbf{M}$, a single switch with the highest betweenness centrality is selected. Then all the flows that go through the selected switch are excluded from $\mathbf{M}$, and this procedure is repeated with the updated $\mathbf{M}'$ until there are no remaining flows on the flow information matrix.

Sampling Rate Decision

Once the sampling points are selected, each switch has to be allocated a sampling probability. It is also worth noting that the aggregated volume of sampled packets is retained below the maximum IDS capability of C in bits per second. Let $\mathbf{x}$ denote the switch sampling probability vector, where the kth element of $\mathbf{x}$ is the sampling probability of the kth switch. In addition, let $\mathbf{r}$ and $\mathbf{A}$ denote the flow rate vector $\mathbf{r} = [r_i]$, where r_i is the data rate of the ith flow and the flow rate information matrix $\mathbf{A} = [a_{ij}]$, where $a_{ij} = r_i \cdot m_{ij}$, respectively. We consider two possible choices for per-switch sampling. First, $\mathbf{x}$ can be set by

$$x_k = \frac{1}{\sum_i a_{ij}} \cdot \frac{C}{\#\text{ of switches}}$$

such that each switch may sample traffic packets evenly at the same rate, which is equal to C divided by the number of switches. Second, every



	SW1	SW2	SW3	SW4	SW5	SW6
F1	1	0	0	0	1	1
F2	0	1	0	0	0	1
F3	0	1	1	0	1	0
F4	0	0	1	1	0	0
F5	0	1	0	1	1	0

Figure 3. Simple network topology and the corresponding flow information matrix.

switch may have the same sampling probability, that is,

$$x_k = \frac{C}{\sum_i \sum_j a_{ij}}.$$

In this case, the rate of sampled traffic at each switch is proportional to the aggregated traffic rate passing through the switch.

In addition, it is also possible to arbitrarily control the sampling of each individual flow by assigning a different sampling probability to each switch. This *flow-level sampling* is considered as a per-switch sampling technique that can approximate per-flow sampling. Let **b** denote the target sampling rate vector for the flows for the flow-level sampling. Then the sampling rate vector **x** is obtained by a solution that satisfies **A** · **x** = **b**. As in the above per-switch sampling, each flow is sampled either evenly at the same rate by setting

$$\mathbf{b} = \frac{C}{\#\text{ of flows}} \cdot \vec{1}$$

or at a rate that is proportional to its traffic rate by setting

$$\mathbf{b} = \frac{C}{\sum_j r_j} \cdot \mathbf{r}$$

For the flow-level sampling, the solution **x** can be obtained by using the pseudo-inverse of **A** (i.e. $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$).

Simulation

To evaluate the traffic sampling performance, we conducted simulations using the Boston University Representative Internet Topology Generator (BRITE — https://www.cs.bu.edu/brite) for generating a large-scale topology, and Network Simulator 2 (ns-2) to simulate the network flows for malicious packet detection scenarios. We created two types of topologies: clustered transit-stub and grid-type mesh topologies. Each network has 200 switches, and the number of flows varies from 200 to 1000. Each flow has a random data rate from 1 to 100 Mb/s, and it is assumed that 3 percent of total flows are malicious. The IDS capacity for malicious packet inspection is fixed at 1 Gb/s. The reported values are the averages of 1000 simulation runs. Regarding the sampling point decision, we compare the extended betweenness-centrality-based sampling point decision method with the original betweenness-centrality-based sampling point decision and random point decision methods. For each sampling rate decision, four sampling rate decision methods are evaluated: even per-switch (PS) sampling, rate-proportional PS sampling, even flow-level (FL) sampling, and rate-proportional FL sampling.

Figure 4 shows the average capture failure rates of malicious flows on transit-stub and mesh topologies. The capture failure rate is computed under the assumption that the traffic rates of normal and malicious flows are given [12]. The number of sampling points is given by the proposed extended betweenness-centrality-based method, and the other methods select the same number of sampling points. As the number of flows increases, the capture failure rate increases because the number of captured packets belonging to each flow decreases. Note that the aggregate rate of sampled traffic is fixed at the inspection capacity of 1 Gb/s. The number of sampling points only includes 15 switches in the transit-stub topology and varies from 35 to 60 switches in the mesh topology. The results show that the proposed sampling point selection algorithm provides the lowest capture failure rates among the three sampling point decision methods. Regarding the sampling rate decision method, the flow-level sampling methods perform more effectively than the per-switch sampling methods in most cases. However, when the sampling points are determined by the proposed extended betweenness-centrality-based method, the three sampling rate decision methods show almost the same capture failure rate.

Figure 5 shows the average capture failure rate for even FL sampling when the number of sampling points varies for the random selection and original betweenness-centrality-based methods. The number of sampling points for the proposed method was 15 in the transit-stub topology and 55 out of 200 switches in mesh topology. The number of flows is 700 in each topology. Since the probability that a flow will pass through core switches connecting mesh topologies in the transit-stub topology is high, this topology has fewer sampling points. The result indicates that the proposed algorithm can achieve almost the same performance with a much smaller number of sampling points compared to the other methods.

Since the probability that a flow will pass through core switches connecting mesh topologies in the transit-stub topology is high, this topology has fewer sampling points. The result indicates that the proposed algorithm can achieve almost the same performance with a much smaller number of sampling points compared to the other methods.

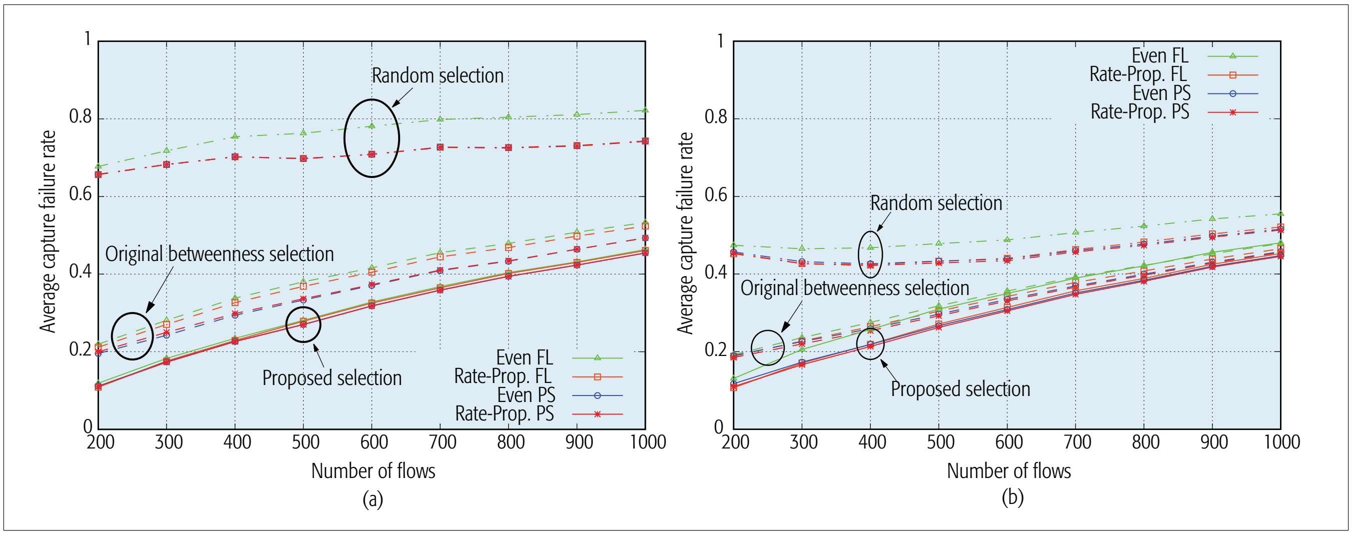


Figure 4. Average capture failure rate on transit-stub and mesh topologies: a) transit-stub topology; b) mesh topology.

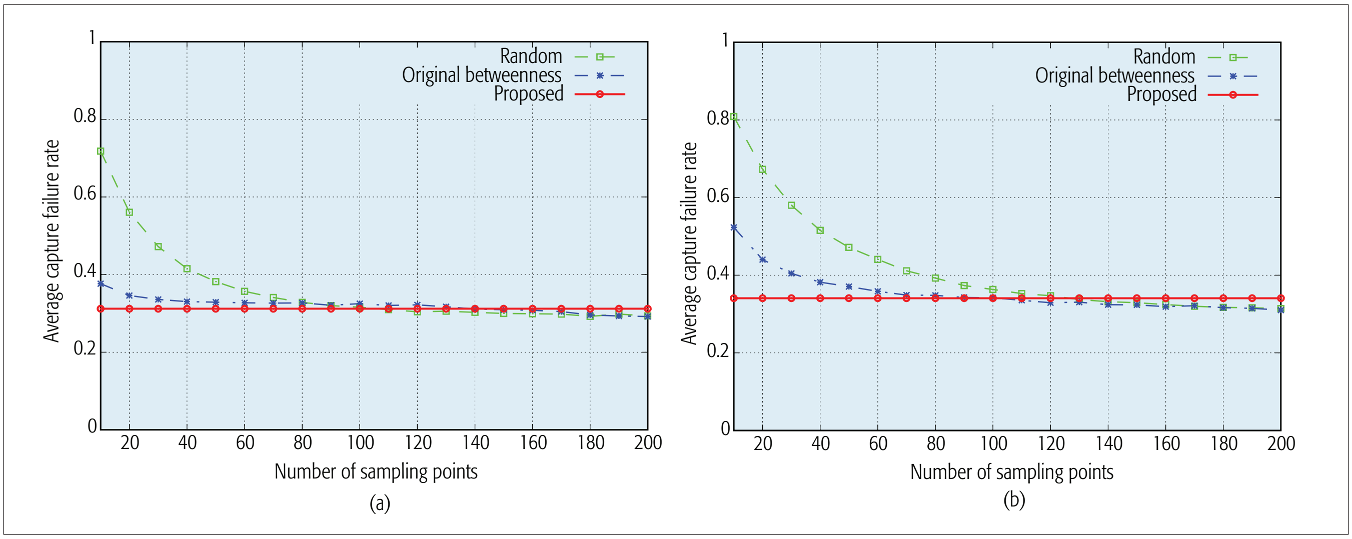


Figure 5. Performance comparison with respect to the number of sampling points: a) transit-stub topology; b) mesh topology.

Experiment

We constructed an SDN-enabled testbed to evaluate the traffic sampling performance. We consider signature-based ransomware propagation detection using IDS. Because some ransomware attacks use malicious toolkits such as Angler, Neutrino, and RIG [13], their propagation can be detected by inspecting whether the captured packets include the malicious toolkit.

Figure 6a illustrates the topology of our SDN-enabled testbed. It consists of six HP 2920 OpenFlow-enabled switches (OpenFlow 1.3 supported), four Open vSwitches (OVSs) running on Linux embedded boxes, two HP workstations (one HP workstation for the SDN controller and the other for IDS), and 15 host PCs. The SDN controller is configured with the helium version of OpenDaylight. Snort IDS is used to inspect the traffic for detecting malicious attacks. Snort is an open source IDS that inspects network traffic using a variety of rulesets and generates security alarms when suspicious network activities are detected. In the experiment, the number of flows is 15, and their data rate varies from 5 to 50 Mb/s. Two malicious flows are added with a rate of 30 and 40 Mb/s, respectively. The IDS detects the ransomware propagation using Snort rulesets for malicious toolkit signatures. The SDN controller updates the sampling probabilities of OpenFlow-enabled switches when it either receives an "OFPT_PACKET_IN" message for newly added flows or detects changes in the data rate of current data flows in service.

Figure 6b shows the rates of malicious traffic forwarded to the IDS. Initially, there are 15 flows. After 10 s, two malicious flows are generated by attackers. The SDN controller detects the two new flows and calculates the sampling points and rates for the switches. The traffic sampling is stabilized in 2 s, as shown in Fig. 6b. The proposed sampling point decision method captures malicious flows at higher data rates than the method that samples from every switch. It is also observed that rate-proportional FL sampling achieves higher data rates than even FL sampling in this network configuration.

Conclusion

Network traffic monitoring plays an increasingly important role in cyber security. Unlike traditional networking systems, the SDN technology

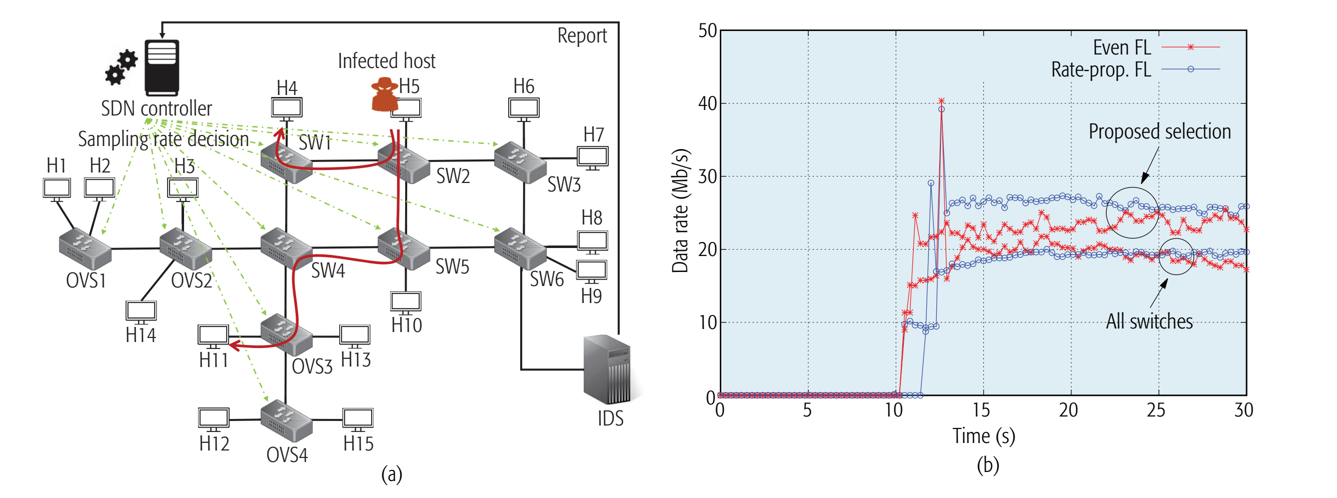


Figure 6. SDN testbed topology and the rates of captured malicious traffic: a) SDN testbed topology; b) captured malicious traffic rate.

provides programmable functionalities that enable OpenFlow-enabled switches to perform probabilistic traffic sampling and to steer the sampled traffic toward a traffic collector for network traffic inspection. In this article, we focus on a scalable traffic sampling point and rate decision scheme that uses a centrality measure to allow the network traffic to be secured with low monitoring overhead. The simulation and experimental results demonstrate that the proposed method achieves less intrusive monitoring and decreases the malicious flow capture failure rate in a scalable manner.

Acknowledgment

This work was supported by an Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (no. 2017-0-00421, Cyber Security Defense Cycle Mechanism for New Security Threats)

References

[1] K. Cabaj and W. Mazurczyk, "Using Software-Defined Networking for Ransomware Mitigation: The Case of CryptoWall," *IEEE Network*, Nov./Dec. 2016, pp. 12–19.
[2] N. Scaife *et al.*, "CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data," *Proc. IEEE Int'l Conf. Distrib. Comp. Sys.*, June 2016, pp. 303–12.
[3] N. L. M. van Adrichem *et al.*, "OpenNetMon: Network Monitoring in Openflow Software-Defined Networks," *Network Operations and Management Symp.*, May 2014.
[4] K. Gioties *et al.*, "Combining OpenFlow and sFlow for an Effective and Scalable Anomaly Detection and Mitigation Mechanism on SDN Environments," *Elsevier Computer Networks*, vol. 62, Dec. 2013, pp. 122–36.
[5] C. Xu *et al.*, "A Survey on Regular Expression Matching for Deep Packet Inspection: Applications, Algorithms, and Hardware Platforms," *IEEE Commun. Surveys & Tutorials*, vol. 18, May 2016, pp. 2991–3029.
[6] J. Rasely *et al.*, "Planck: Millisecond-Scale Monitoring and Control for Commodity Networks," *Proc. ACM Conf. SIGCOMM*, Aug. 2014, pp. 407–18.
[7] M. Korczynski *et al.*, "An Accurate Sampling Scheme for Detecting SYN Flooding Attacks and Portscans," *IEEE ICC*, June 2011, pp. 1–5.
[8] S. Shirali-Shahreza and Y. Ganjali, "FleXam: Flexible Sampling Extension for Monitoring and Security Applications in OpenFlow," *Proc. ACM SIGCOMM Wksp. Hot Topics in Software Defined Networking*, Aug. 2013, pp. 167–68.
[9] J. Suh *et al.*, "OpenSample: A Low-Latency, Sampling-Based Measurement Platform for Commodity SDN," *Proc. IEEE Int'l Conf. Distrib. Comp. Sys.*, July 2014, pp. 228–37.
[10] Y. Liu *et al.*, "On the Resource Trade-off of Flow Update in Software-Defined Networks," *IEEE Commun. Mag.*, vol. 54, no. 6, June 2016, pp. 88–93.
[11] L. C. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40, Mar. 1977, pp. 35–41.
[12] T. Ha *et al.*, "Suspicious Traffic Sampling for Intrusion Detection in Software-Defined Networks," *Elsevier Computer Networks*, vol. 109, Nov. 2016, pp 172–82.
[13] R. Brewer, "Ransomware Attacks: Detection, Prevention and Cure," *Elsevier Network Security*, vol. 2016, Sept. 2016, pp. 5–9.

Biographies

Seunghyun Yoon [S'16] received his B.S. degree from the School of Computer Science and Electrical Engineering, Handong Global University, Pohang, Korea, in 2016. He is currently pursuing a Ph.D. degree at the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST), Gwangju, Korea. His research interests include cyber-physical systems (CPS) and security issues in software defined networking.

Taejin Ha received his B.S. degree from the School of Computer Science and Electrical Engineering, Handong Global University in 2011. He is currently pursuing a Ph.D. degree at the School of Electrical Engineering and Computer Science, GIST. His research interests include security issues in software defined networking.

Sunghwan Kim received his B.S. degree from the School of Computer Science and Engineering, Dongguk University, Seoul, Korea, in 2015, and his M.S. degree from the School of Information and Communications, GIST in 2017. He is currently pursuing a Ph.D. degree at GIST. His research interests include cloud computing and security in software defined networking.

Hyuk Lim [S'97, M'03] (hlim@gist.ac.kr) received his B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from the School of Electrical Engineering and Computer Science, Seoul National University, Korea, in 1996, 1998, and 2003, respectively. From 2003 to 2006, he was a postdoctoral research associate with the Department of Computer Science, University of Illinois at Urbana-Champaign. He is currently a full professor with the School of Electrical Engineering and Computer Science, GIST. His research interests include analytical modeling and empirical evaluation of computer networking systems, network protocol design and performance analysis of wireless networks, and industrial Internet/cyber-physical systems.

Quiet Dogs Can Bite: Which Booters Should We Go After, and What Are Our Mitigation Options?

José Jair Santanna, Ricardo de O. Schmidt, Daphne Tuncer, Anna Sperotto, Lisandro Z. Granville, and Aiko Pras

Large network security companies often report websites, called Booters, that offer DDoS attacks as a paid service as the primary reason for the increase in occurrence and power of attacks. Although hundreds of active Booters exist today, only a handful of those that promoted massive attacks faced mitigation and prosecution actions. The authors focus on Booters that are "under the radar" of security initiatives.

Abstract

Large network security companies often report websites, called Booters, that offer DDoS attacks as a paid service as the primary reason for the increase in occurrence and power of attacks. Although hundreds of active Booters exist today, only a handful of those that promoted massive attacks faced mitigation and prosecution actions. In this tutorial article we focus our attention on Booters that are "under the radar" of security initiatives, by advertising high attack power and being very popular on the Internet. We discuss and provide grounds for critical thinking on what should be further done toward Booter mitigation.

Introduction

Booters can easily be found on the public web through search engines (e.g., Google). Distributed denial of service (DDoS) attacks performed by Booters can be hired for a couple of U.S. dollars. Booters also present multiple ways of paying for their "service" (e.g., Paypal, Bitcoin, and credit card), while offering various types of attacks (e.g., SYN flood, DNS-based reflection, and application layer attacks). Karami *et al*. [1] showed that the large number of active Booters and the ease with which these can be found and their service hired contribute to the increasing occurrence of DDoS attacks. This observation proved to be correct given that the majority of attacks, including the most powerful DDoSs, have been launched by Booters (at a data rate higher than 100 Gb/s), as reported by Akamai (https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/akamai-q2-2016-state-of-the-internet-security-report.pdf, accessed 21 March 2017).

Although hundreds of active Booters exist, few of those involved in massive attacks underwent mitigation actions. Booters that to date have been the target of investigations, mitigations, or prosecutions are the ones that successfully disrupted the operation of popular services, such as Xbox Network, PlayStation Network, Instagram, and Tinder (http://krebsonsecurity.com/?s=booter, accessed 21 March 2017). In 2016, the vDos Booter [2] was reported to have launched more than 170,000 DDoS attacks in less than four months; as a consequence, vDos owners were arrested. In 2016, a sustained 540 Gb/s attack, launched by the LizardStresser Booter (https://www.arbornetworks.com/blog/asert/rio-olympics-take-gold-540gbsec-sustained-ddos-attacks, accessed 21 March 2017), was also witnessed during the Olympic Games in Brazil, as well as a staggering terabit-per-second attack using the Mirai botnet (also related to Booters — https://krebsonsecurity.com/2016/10/hackforums-shutters-booter-service-bazaar, accessed 21 March 2017) targeting OVH (https://arstechnica.com/security/2016/09/botnet-of-145k-cameras-reportedly-deliver-internets-biggest-ddos-ever/, accessed 21 March 2017) and Dyn (http://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack, accessed 21 March 2017). These are only a few examples of Booter attacks, which were eventually reported in the news and caught the public's attention. However, in only some cases did the people responsible for the Booters behind these attacks face legal consequences. The goal of this tutorial article is to raise awareness about Booters that stay "under the radar" of security initiatives by advertising high attack power and being extremely popular.

The research on Booter mitigation is still at an early stage. Most of the existing work has been focused on looking at the technical characteristics of the attacks performed by Booters [3–5] and profiling their targets [6]. Other initiatives [3, 7] have used leaked Booter databases to, for example, enumerate the characteristics of hired attacks. Other research efforts have been exploring issues associated with identifying Booter websites [8], discovering and mitigating the infrastructures used by Booters to perform attacks [9, 10], and describing Booters' financial operations [1]. In this tutorial article, we extend the contribution from those previous efforts by providing extra ground for discussions and critical thinking on what one can further do and how to mitigate Booters.

In the first part of this article, we focus on answering the question *which Booters should we go after?* Using a combination of measurement datasets that we collected ourselves and also

Digital Object Identifier:
10.1109/MCOM.2017.1600992

José Jair Santanna, Ricardo de O. Schmidt, Anna Sperotto, and Aiko Pras are with the University of Twente; Daphne Tuncer is with University College London; Lisandro Z. Granville is with Federal University of Rio Grande do Sul.

0163-6804/17/$25.00 © 2017 IEEE

retrieved from public sources, we highlight which of those “under the radar” Booters are very popular and advertise high attack power for low prices, but have not yet undergone any meaningful mitigation action. Our ground-truth is a list of 435 Booter domain names from the Booter Blacklist initiative (http://booterblacklist.com, accessed 21 March 2017) [8]. Our dataset and associated scripts for data analysis are publicly available at http://jairsantanna.com/booter_ecosystem_analysis (accessed 21 March 2017). In the second part of this article, we provide a thorough discussion of mitigation options to address the problem of Booters. Our methodology is based on identifying organizations (in)directly involved with Booters that could take part in mitigation actions to inhibit or even dismantle Booter operations. We finally conclude the article by discussing the lessons learned.

Which Booters Should We Go After?

Mitigation and prosecution actions performed against the Booter ecosystem (i.e., websites, owners, clients, and infrastructure) have mostly targeted those Booters that launched powerful attacks toward large organizations. However, there are still hundreds of Booters, such as those revealed by the Booter Blacklist initiative, that are somehow “under the radar” of security initiatives and therefore rarely the target of mitigation actions. Obviously, not all Booters could be mitigated at once, but a priority order would be welcome. The first Booters to be mitigated should preferably be the ones that perform the most powerful attacks. Identifying these Booters is a task mostly restricted to large network security companies that have the ability to classify the most dangerous attacks of those targeting their clients. In this section, we describe a heuristic to prioritize the mitigation of a (second) set of Booters. Our heuristic relies on the following three premises.

Booters’ Services Are Not Likely to Be Ethical or Legal: It is debatable whether an illegal Booter can be a legitimate stress tester. However, as presented by Douglas *et al.* in [11], the attack infrastructure used by Booters mostly consists of compromised/misused machines (e.g., botnets and amplifier services). Others have attested to this argument by hiring attacks from Booters and testing them against controlled environments [3–5].

The Ratio between the Number of Accesses to Their Websites and the Number of Launched Attacks Is Similar between All Booters: This premise leads us to conclude that the most accessed Booters are those likely to launch more attacks.

The Attack Power Advertised by Booters Can Be Factual: It has been observed that Booters, in general, deliver far less attack power than they promise to their clients [5]. However, Booters that caught the attention of the media performed attacks stronger than they actually advertised on their respective websites. For example, `lizardstresser.su` attacked the PlayStation and Xbox networks during Christmas 2013 with 300 Gb/s attack power, while on their website (as of 2013) attacks up to 125 Gb/s were advertised. To further support our premise, we argue that it is quite easy to find amplifiers for reflection attacks and/or to compromise a large number of systems (e.g., Internet of Things devices). Therefore, skilled hackers and owners of Booters can easily scale up their attack power [12].

Based on our premises, our heuristic to highlight Booters consists of four steps. First, we identify the most accessed Booters using the website ranking provided by Alexa (http://alexa.com, accessed 21 March 2017). For each of the 435 Booter domain names in http://booterblacklist.com, we scrape the Alexa rankings from 1 November 2016 to 1 February 2017. Our analysis only considers those Booter domain names that ranked up to 3 millionth in Alexa, which represents around 1 percent of the total number of registered domain names in the entire Internet (http://verisign.com/innovation/dnib, accessed 21 March 2017). We then scrape these top-ranked Booter domain names to reveal their highest advertised attack rate (i.e., the most powerful attack) and their price range. Finally, we investigate the dates of creation and expiration of their domain names based on Whois information. This last step shows how long the top-ranked Booters are offering (and likely delivering) attacks without facing any type of mitigation action.

Figure 1 summarizes our findings. From the ground-truth list of 435 Booter domain names, 33 ranked among the top-1 percent of all most accessed domain names on the Internet (Fig. 1a). In addition to their position in Alexa’s ranking, we observed that 8 Booters offer attacks with a rate of 100 Gb/s or higher (Fig. 1c); these are Booters ranked in the following positions: 4, 7, 8, 13, 14, 18, 25, and 32. Attacks of 100 Gb/s or more are powerful enough to bring most systems offline on the Internet, especially those that are not protected by large security companies. Figure 1b shows that some of these 8 Booters (i.e., Booters ranked at 4, 14, 18, and 32) charge at maximum US$100, while their cheapest service plan is US$10 or less. Such a range of prices is surprising when considering that the cost of recovering from a DDoS attack is on average US$53,000 for small and medium companies, and US$417,000 for large companies (https://press.kaspersky.com/files/2015/09/IT_Risks_Survey_Report_Threat_of_DDoS_Attacks.pdf, accessed 21 March 2017). Based on our premises and findings, it can therefore be deduced that these last four Booters (ranked at 4, 14, 18, and 32) are the ones that, at the lowest cost to their clients, can do the most damage to the target of an attack. Furthermore, we observed that four other Booters offer attacks for free (Booters ranked 1, 5, 9, and 10). However, upon closer examination of these Booters, we discovered that, except for the Booter ranked 9th, they all promote services from other (paid) Booters. We believe that these “free-service” Booters are used to increase the popularity of actual paid Booters.

From those Booters listed in Fig. 1, three domain names are currently for sale, ranked 19, 29, and 31. These domains pointed to actual Booter websites that were active in the past, as confirmed by the Internet Archive initiative (https://archive.org, accessed 21 March 2017; The Internet Archive has dozens of historical snap-

Mitigation and prosecution actions performed against the Booter ecosystem have mostly targeted those Booters that launched powerful attacks towards large organizations. However, there still exist hundreds of Booters, such as those revealed by the Booter Blacklist initiative, that are somehow “under the radar.”

Infected machines can be part of a botnet able to perform various types of attacks. Misused public services are in turn only used for reflection and amplification attacks. The last element in the Booter ecosystem is the Booter operational database, where all information about clients and hired attacks is stored.

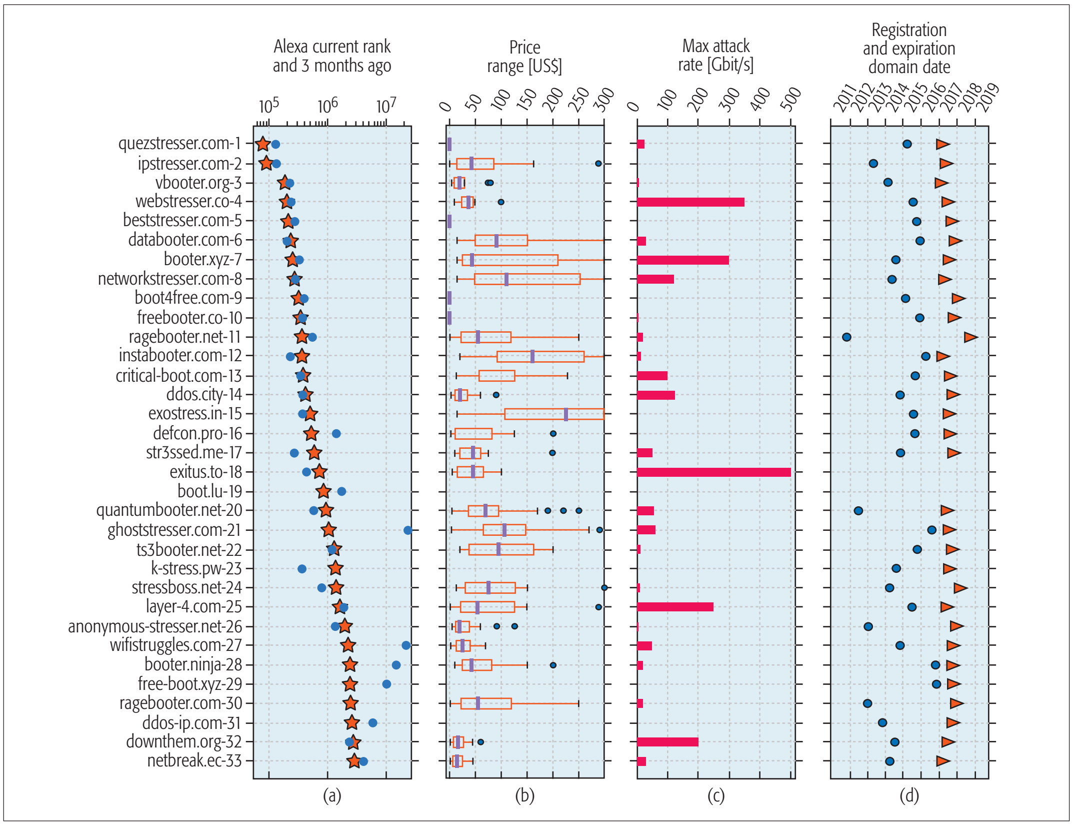


Figure 1. a) Top ranked Booter domain names, up to the 3 millionth position in Alexa ("star" is the current rank, while "dot" is the rank 3 months ago); b) price range (minimum, gigabits per second); d) registration and expiration dates of domain names.

shots of these specific domain names.). These are still highly ranked domains in Alexa because users still try to reach them. This assumption is supported by the DNSDB initiative (https://www.dnsdb.info, accessed 21 March 2017), one of the largest collections of DNS records worldwide. We found in DNSDB records that each of these three domains have received thousands of DNS requests (likely interpreted as web access) in the last two years.

Finally, we observed that two Booter domain names (ranked 11 and 30) point to the same Booter website. This Booter has, among all the top ranked ones, the oldest domain creation date: it was registered in 2011. Although it was reported in 2013 by a security specialist (https://krebsonsecurity.com/2013/05/ragebooter-legit-ddos-service-or-fed-backdoor, accessed 21 March 2017), we are unaware of any mitigation or prosecution action against it. A possible explanation is that this Booter is actually an "FBI backdoor," as described by its owner. A speech by the CEO of CloudFlare mentioned that "sometimes we have court orders to not take (web)sites down" (https://www.youtube.com/watch?v=Wr-PrSqI16A&t=2929s, accessed 21 March 2017). Whether true or not, the concrete fact is that this Booter is still online. We further discuss CloudFlare and other DDoS protection services in the following section.

Our heuristic clearly provides means to highlight Booters "under the radar" of security companies that should be the first to undergo mitigation actions. In the next section, we discuss how third-party organizations can enroll on mitigation actions against Booters.

Who Can Help Perform Mitigation Actions?

Figure 2 shows the ecosystem of Booters, that is, all elements involved in Booter activities. To identify organizations that can engage in mitigation actions against Booters, we first look at those that are (in)directly involved in the Booter ecosystem. To hire an attack, a client must first access the Booter website and create an account. The access to a Booter website usually happens via a third-party cloud-based security provider (CBSP) transparent to the client. The payment for a hired attack (or an attack plan — sets of attacks that can be performed within a given period of time) is done via a third-party payment system. After selecting a "service" and paying for the plan, clients can launch attacks at any time and against any target on the Internet.

To perform DDoS attacks, Booters use a back-end infrastructure that consists of three types of machines: command and control (C&C) machines, infected machines (computers with bugs in Fig. 2), and misused public services (computers with exclamation marks in Fig. 2). Booters are unlikely to send attack traffic directly from their C&C machines. Instead, infected machines can be part of a botnet able to perform various types of attacks. Misused public services are in turn only used for reflection and amplification attacks (e.g., DNS-based and NTP-based attacks). The last element in the Booter ecosystem is the Booter operational database, where all information about clients and hired attacks is stored.

In addition to CBSPs and a payment system, five other organizations are also (in)directly involved in the Booter ecosystem:

- Web hosting companies that host the content of Booter websites
- Top-level domain (TLD) operators
- Domain registrars that provide means for the registration of Booter domain names
- Web indexing and search companies that facilitate finding Booter websites
- Local DNS resolvers that resolve Booter domain names to IP addresses

We next show to what extent these third-parties are involved with Booters and discuss potential actions they could perform to support the mitigation of the Booter phenomenon. The starting point of the analysis presented in this section is the same list of 435 Booter domain names described and analyzed in the previous section.

TLDs Operators, Domain Registrars, Web Hosting Companies, and CBSPs

These four types of organizations are analyzed together for the following two reasons. First, they are linked to Booters mainly via domain names. Second, the same company may provide different types of services. Examples of such organizations include SIDN (https://sidn.nl, accessed 21 March 2017), which is both a TLD operator (of .nl) and a domain registrar; GoDaddy (http://godaddy.com, accessed 21 March 2017), which is both a domain registrar and a web hosting company; and CloudFlare (https://cloudflare.com, accessed 21 March 2017), which is both a web hosting company and a CBSP.

We use distinct methodologies to analyze each of these four types of organization: for TLDs, we look into the composition of Booter domain names; for domain registrar, we rely on Whois information; and for web hosting and CBSPs, we use their IP address and autonomous system (AS) information (http://www.team-cymru.org/IP-ASN-mapping.html, accessed 21 March 2017).

As shown in Fig. 3, by looking at the composition of domain names, we observe that more than 68 percent of all 435 Booters are registered within the .com and .net TLDs. Other TLDs account for less than 5 percent of registrations each. For example, .nl accounts for around 1 percent of registrations. We also see that 74 percent of Booter domain names contain the terms "stresser," "booter," or "ddos." Information on the composition of domain names could be used by TLD operators and registrars to, for example, take down existing domains or prevent the registration of new (suspect) ones. An enabler to check Booter domain names was proposed in [8]. Preventing the registration of new domains could, however, impact the registration of valid domain names that could eventually be classified as suspect.

We analyzed the impact of domain names that have the terms "stresser," "booter," or "ddos" in their composition, and are registered within .com, using a large-scale active DNS measurement dataset [13]. We found out that from all 2721 domains names in .com containing one of the three terms, only 61 domain names (less than 3 percent) are *not* related to Booters.

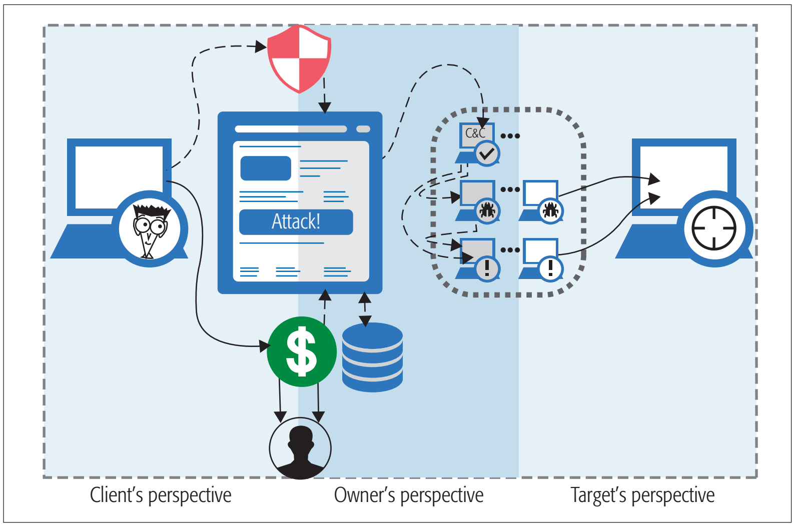


Figure 2. Booter ecosystem.

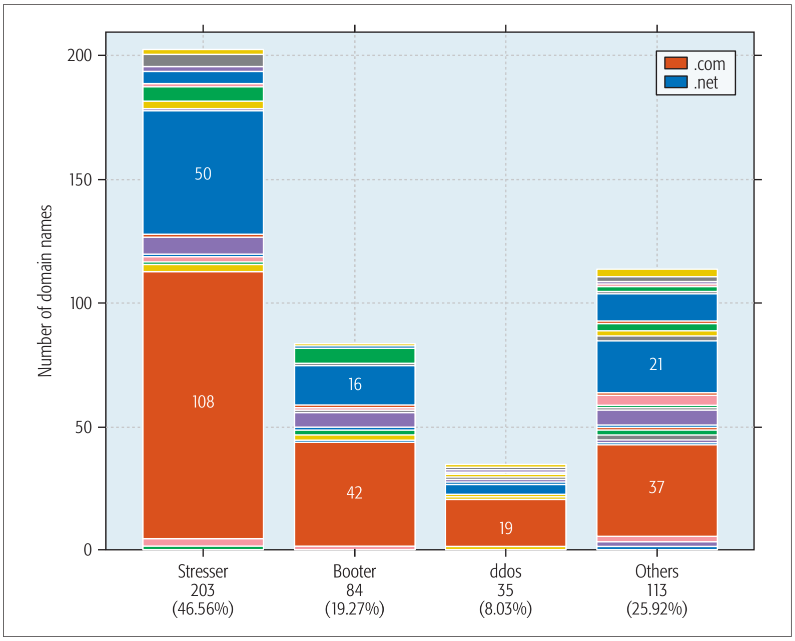


Figure 3. Domain word composition and TLDs distribution (.com and .net highlighted).

That is, by filtering registrations based on these three terms, a very small percentage of legitimate registrations would be affected. However, Booter owners could overcome these actions by adopting alternative terminologies. By analyzing Booters' Whois information, we observe that almost 70 percent of all Booters are within the top 10 registrars, as can be seen in Fig. 4, if Enom, GoDaddy, and Namecheap decided to act against Booters, around 50 percent of all Booters would be affected.

When looking at the IP addresses and ASs related to 202 (online) Booter domain names, we also noticed that some companies would have a higher impact if they got involved in mitigation actions. For example, CloudFlare is involved with at least 76 Booters (37 percent). The fraction of Booters behind CloudFlare dropped significantly compared to a previous study [5]: 88 percent — 52 out of 59 Booters (in that study, 49 Booters

It is extremely easy to find Booter websites through public web search engines, such as Google, Bing, and Yahoo. To prevent users from interacting with Booters, search engines could notify them that hiring or even accessing Booter "services" would potentially have legal implications.

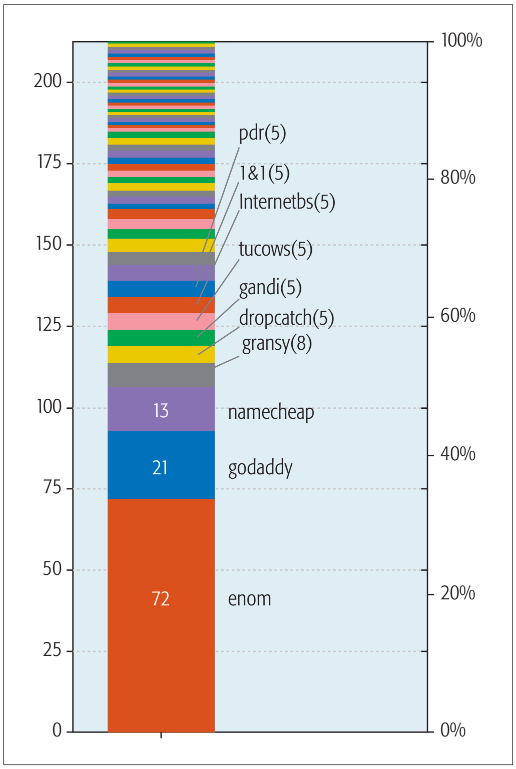


Figure 4. Registrars analysis based on Whois information.

are part of the 37 percent seen in the current analysis). Given that Booters typically attack each other [7], competing for market shares or even to simply show off their attack power, we believe that if CloudFlare (and other CBSPs) stopped protecting Booters, these would eventually take each other offline — or at least have their reachability compromised. However, this action would only have an impact if all CBSPs decide to get involved, leaving no options to Booters but being out of a DDoS protection service.

Booters behind CBSPs require a more refined investigation in order to determine the web hosting company where their websites are actually hosted (ASs). To determine the web hosting companies specifically obscured by CloudFlare, we used the CloudPiercer initiative (http://cloudpiercer.org, accessed 21 March 2017) [14], which applies several metrics to look up actual (or historical) IP addresses. Using this methodology, we found out that 24 web hosting companies host 47 Booters (out of 76 in CloudFlare), as depicted in the middle (zoom-in) graph of Fig. 5. The other 29 Booters are also likely to be protected and hosted by CloudFlare. Merging web hosting companies in Fig. 5 (ASs) with the discovered hidden ASs, we observe that the top 10 web hosting companies do not change (their ranks do, however). For example, comparing the left and right graphs in Fig. 5, it can be observed that OVH and GoDaddy gain 6 and 2 positions, respectively. The main takeaway message from this analysis is that if the top web hosting companies enroll in effective mitigation actions (e.g., simply stop hosting alleged Booters), a high percentage of Booters would go offline. However, Booters could, again, adapt to such an action by moving to other hosting companies.

Payment Systems

Payment systems are one of the main elements of the Booter ecosystem. In 2015, Karami *et al.* [1] reported a joint effort made with PayPal by which Paypal accounts allegedly belonging to Booter owners were deactivated. This operation was very successful, momentarily reducing the number of payments and attacks by Booters. However, Booters have partially overcome this mitigation action. For example, only one Booter among those listed in Fig. 1 still offer PayPal as a payment option. Other Booters now use various crypto-currencies, such as Bitcoin, Litecoin, and Dogecoin. This change in the payment system makes it harder to trace Booter owners by following the money they earn. The action by PayPal had a positive impact on the Booter ecosystem, given that only a small number of Booter clients have Bitcoin wallets. In addition, based on the profile of a typical Booter client, we believe that not many of them would be willing to create a Bitcoin wallet to simply perform attacks.

Web Searching Companies

It is extremely easy to find Booter websites through public web search engines, such as Google, Bing, and Yahoo. To prevent users from interacting with Booters, search engines could notify them that hiring or even accessing Booter "services" would potentially have legal implications. This action is similar to one currently done for "unsafe sites" in Google Chrome (https://support.google.com/chrome/answer/99020/, accessed 21 March 2017), and could reduce the number of accesses to Booters and, ultimately, the number of attacks launched by Booters.

DNS Resolver Operators

A straightforward way to prevent users from accessing Booters is by blacklisting Booter domain names at DNS resolvers. In this way, when an IP address resolution is needed for a blacklisted domain name, the resolution is refused. Booter websites would still be reachable via alternative DNS resolvers that do not block the resolution, or via VPNs or the Tor browser. However, considering that Booters under CBSPs can block access from VPNs and Tor nodes, this action by DNS resolver operators could ultimately result in a significant reduction of the number of attacks from Booters.

It is very important to highlight that the mitigation actions described in this section might require a court order before they are put in place. For example, CloudFlare's CEO stated that "it is tricky when private organizations act as law enforcement" and that "they comply with any court order" (https://www.youtube.com/watch?v=Wr-PrSqI16A, accessed 21 March 2017). Although the legitimacy of services offered by Booters is still debatable, Douglas *et al.* [11] state that it is unlikely that Booters provide legal and ethical services, because their back-end infrastructures are composed of compromised machines or misused systems (e.g., botnets, DNS resolvers, NTP servers, and Webshells). Determining the back-end infrastructure (or parts of it) of a Booter can be done by hiring an attack against a controlled environment, as was done in previous works [3–5].

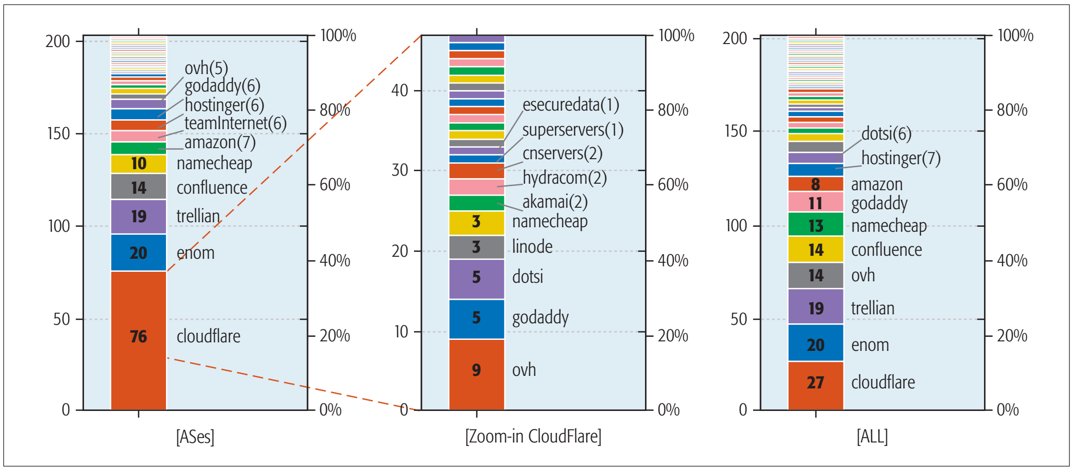


Figure 5. Web hosting analysis based on ASs (left), with zoom-in on the ASs hidden by CloudFlare (middle), and the overall merged results (right).

In this article, we raised awareness about the hundreds of silent Booters, safely operating "under the radar" of security actions, that could at any point in time cause substantial damage to any system in the Internet. We hope that our findings will foster further discussions and effective actions against Booters.

Lessons Learned

In this article, we have two goals. Our first objective is to identify Booters "under the radar" of security actions that should face mitigation in a higher priority order. Our second objective is to determine organizations that (in) directly interact with Booters and could act to mitigate Booters.

To achieve the first goal, we propose a heuristic based on website popularity, maximum attack rate, price range, and domain creation and expiration. Using this heuristic and a set of premises, we identified 33 Booter domain names that should face mitigation with higher priority, and provided arguments to justify the need for such mitigation actions. We showed that Booters "under the radar" pose a potential risk and, as such, we consider proactive mitigation to be the best course of action.

Concerning the second goal, we learned that dismantling the entire Booter ecosystem is very challenging. None of the mitigation actions mentioned above could eliminate, on a standalone basis, the Booter phenomenon. However, if some of them were actually deployed, we would certainly see a decrease in Booter operations, similar to what happened after PayPal's operation against Booters in 2015. This decrease would be mostly caused by lay users (i.e., Booter clients) that would not be able to overcome challenges imposed by the mitigation actions. While technically skilled users would still find a way to use Booter services, they remain a minority.

Booter owners are likely to find ways to overcome any mitigation action. Booters can profit from relatively safe business when not calling too much attention from society and security specialists. To date, legal actions against both Booter owners and clients have been taken only in cases where large corporations were targeted by DDoS attacks. In this article, we raise awareness about the hundreds of silent Booters, safely operating "under the radar" of security actions, that could at any point in time cause substantial damage to any system in the Internet. We hope that our findings will foster further discussions and effective actions against Booters.

Acknowledgments

This work is funded by the FLAMINGO (EU FP7 ICT- 318488) and D3 (NWO 628001018) projects. Ricardo de O. Schmidt's research is also funded by the SAND (http://www.sand-project.nl) and NWO DAS (http://www.das-project.nl) projects. Special thanks go to the teams of Farsight Security, Cloudpiercer, and Mattijs Jonker for assisting us in the experiments and providing valuable information.

References

[1] M. Karami, P. Youngsam, and D. McCoy, "Stress Testing the Booters: Understanding and Undermining the Business of DDoS Services," *Proc. Int'l. Conf. World Wide Web*, 2016.

[2] B. Krebs, "Alleged vDOS Proprietors Arrested in Israel," http://krebsonsecurity:com/2016/09/alleged-vdos-proprietors-arrested-in-israel/#more-36288, 2016, accessed 21 Mar. 2017.

[3] M. Karami and D. McCoy, "Understanding the Emerging Threat of DDoS-as-a-Service," *Proc. USENIX Wksp. Large-Scale Exploits and Emergent Threats*, 2013.

[4] V. Bukac *et al.*, "Service in Denial — Clouds Going with the Winds," *Network and System Security*, 2015.

[5] J. J. Santanna *et al.*, "Booters-An Analysis of DDoS-as-a-Service Attacks," *Proc. IFIP/IEEE Symp. Integrated Network and Service Management*, 2015.

[6] A. Noroozian *et al.*, "No Who Gets the Boot? Analysing Victimization by DDoS-as-a-Service," *Proc. Int'l. Symp. Research in Attacks, Intrusions, and Defenses*, 2016.

[7] J. J. Santanna *et al.*, "Inside Booters: An Analysis on Operational Databases," *Proc. IFIP/IEEE Int'l. Symp. Integrated Network Management*, 2015.

[8] J. J. Santanna *et al.*, "Booter Blacklist: Unveiling DDoS-for-Hire Websites," *Proc. Intl. Conf. Network and Service Management*, 2016.

[9] L. Krämer *et al.*, "AmpPot: Monitoring and Defending Against Amplification DDoS Attacks," *Research in Attacks, Intrusions, and Defenses*, 2015.

[10] J. Krupp, M. Backes, and C. Rossow, "Identifying the Scan and Attack Infrastructures Behind Amplification DDoS Attacks," *Comp. and Commun. Security*, ser. CCS '16. ACM, 2016.

[11] D. Douglas *et al.*, "Booters: Can Anything Justify Distributed Denial-of-Service (DDoS) Attacks for Hire?" *J. Info., Commun. and Ethics in Society*, vol. 15, no. 1, 2017.

[12] A. Pras *et al.*, "DDoS 3.0 — How Terrorists Bring Down the Internet," *Proc. Int'l. GI/ITG Conf.*, MMB and DFT, 2016.

[13] R. van Rijswijk-Deij *et al.*, "A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements," *IEEE JSAC*, vol. 34, no. 7, 2016.

[14] T. Vissers *et al.*, "Maneuvering Around Clouds: Bypassing Cloud-Based Security Prwoviders," *Proc. Conf. Comp. and Commun. Security*, 2015.

Biography

José Jair Santanna is a Ph.D. candidate in the Design and Analysis of Communication Systems Group at the University of Twente, the Netherlands. His research interests are in the areas of Internet security, management and measurements, and (big) data analysis.

Ricardo de Oliveira Schmidt is a postdoctoral researcher within the chair of Design and Analysis of Communication Systems, University of Twente, and a research engineer at SIDN Labs, the Netherlands. He obtained his Ph.D. from the University of Twente in 2014. His research interests are in Internet security, management, and measurements.

Daphne Tuncer is a research associate in the Communications and Information Systems Group, Department of Electronic and Electrical Engineering, University College London (UCL), United Kingdom. She obtained her Ph.D. in electronic and electrical engineering from UCL in November 2013. Her research interests are in the areas of software-defined networks (in particular applied to network resource management), cache/content management, and adaptive network resource management.

Anna Sperotto is an assistant professor at the Design and Analysis of Communication Systems Group of the University of Twente. She received a Ph.D. degree from the University of Twente in 2010, with the thesis *Flow-Based Intrusion Detection*. Her research interests include network security, network measurements, and traffic monitoring and modeling.

Lisandro Zambenedetti Granville is an associate professor at the Federal University of Rio Grande do Sul. He served as TPC Co-Chair of IFIP/IEEE DSOM 2007 and IFIP/IEEE NOMS 2010, and as General Co-Chair of IFIP/IEEE CNSM 2014. He is Chair of IEEE ComSoc's Committee on Network Operations and Management, Co-Chair of the IRTF's Network Management Research Group, and President of the Brazilian Computer Society. His interests include network management, software-defined networking, and network functions virtualization.

Aiko Pras is a professor Internet security at the University of Twente, where he is a member of the Design and Analysis of Communication Systems (DACS) group. His research interests include Internet security, measurements, and management. He is chairing the IFIP Technical Committee on Communications Systems (IFIP-TC6), and has been Chair of the EU Future Internet cluster and coordinator of the European Network of Excellence on Management of the Future Internet. He serves on many steering committees and editorial boards.

Measuring the Energy Consumption of Cyber Security

Luca Caviglione, Mauro Gaggero, Enrico Cambiaso, Maurizio Aiello

The authors propose to investigate the energy required by the most popular cryptographic algorithms. The collected measures are used to model relationships between power drains and size of the key or offered load via a black box approach. Results can also be used to prevent classical traffic analysis campaigns.

ABSTRACT

The Internet is a core tool for developing commercial and social relationships. As a consequence, cyber security must be properly assessed, for instance, to face new and sophisticated threats. To deliver large-scale services, proper countermeasures characterized by a non-negligible energetic impact have to be pursued. From this perspective, this article proposes to investigate the energy required by the most popular cryptographic algorithms. The collected measures are used to model relationships between power drains and size of the key or offered load via a black box approach. Results can also be used to prevent classical traffic analysis campaigns.

INTRODUCTION

Nowadays, the Internet connects a huge amount of entities coordinating and exchanging personal and sensitive data. For instance, the Internet of Things (IoT) approach is used to perform field measurements, cloud platforms offer computing resources as a commodity, and personal mobile devices enable connectivity while on the road. As a consequence, cyber security is definitely a core requirement for mobile, pervasive, and complex network architectures [1]. Unfortunately, delivering such a rich set of services, especially in a trusted and secure manner, does not come for free. In fact, Internet service providers (ISPs) or entities operating a data center usually face high expenditures, mainly in terms of energy bills. Therefore, understanding and optimizing the energy consumption of computing and network appliances have become relevant research topics [2–4]. However, aspects related to energy consumption of cyber security mechanisms have often been neglected, even if the emerging trend is to explicitly consider their impact as well [5]. In this perspective, a relevant portion of the literature focuses on mobile devices (e.g., [6]), mainly due to their battery-operated flavor, limited amount of computing/storage resources, and intrinsic difficulties to measure power drains in a non-invasive manner [7]. This article tries to fill this gap and proposes to characterize the energy consumption of different standard cryptographic algorithms deployed in modern ISPs, data centers, and end nodes. In fact, providing security is often an integrated process involving entities placed both in the core and at the border of the network [1]. In more detail, enlightening relations among energy requirements and cyber security allows:

- Estimating the energy requirements of security mechanisms to assess their economic impact and perform optimizations
- Demonstrating how traffic related to security aspects could be measured through an higher-level indicator, such as energy drain, in order to ensure scalability by preventing the need to capture packets and process big-data-like information
- Supporting the idea of using energy consumption as a marker to develop novel cyber security mechanisms, for example, to early detect attacks.

To this aim, we performed a measurement campaign on several production-quality cryptographic algorithms. Specifically, particular attention was focused on selecting implementations that can be deployed both within end nodes and machinery used in an ISP and in a data center. To model data collected in different use cases, we introduced a black box approach relying on statistical tools. In particular, our goal is to find a qualitative relationship among cyber security aspects and power requirements, for instance, to offer guidelines to engineer and optimize large-scale deployments or to design novel configurations. To this aim, we used a least squares approach to obtain a polynomial model of the energy costs of cyber security. Results indicate that there is room for developing a more green and secure Internet.

The rest of this article is structured as follows. The following section presents the reference scenario and the considered security technologies, while the third section briefly discusses the theoretical background used for modeling the energy consumption. Then we deal with the adopted testbed and showcase numerical results. The final section concludes the work by reviewing the most important lessons learned.

REFERENCE SCENARIO AND CONSIDERED SECURITY TECHNOLOGIES

As previously pointed out, networks are a relevant part of our lives. As a possible example, smartphones are used by about 65 percent of the global population to perform financial activities, share data over online social networks, and communicate in real time. This leads to infrastructures characterized by a high degree of heterogeneity; for instance, wireless loops have different security requirements with respect to wired trunks. As a

Digital Object Identifier:
10.1109/MCOM.2017.1600955

The authors are with the National Research Council of Italy.

consequence, networks should provide a proper degree of cyber security since the volume and type of data are of interest to cyber criminals, for example, to profile users or collect information for large-scale social engineering attacks. Nevertheless, providing a secure environment requires acting on different entities ranging from user devices to remote machineries.

In modern scenarios, cyber security is provided through a vast set of techniques that can interact in a very complex manner. In fact, guaranteeing trusted and secure features encompasses a rich variety of protocols (e.g., to deliver the information via Transport Layer Security) and machineries (e.g., to distribute and manage credentials or certificates). For instance, a server devoted to implement authentication, authorization, and accounting can be complex, especially if scalability is needed. Unfortunately, achieving precise understanding of how the different hardware and software components contribute to energy drains is still an open research problem, especially due to heterogeneity of implementations [3, 5, 7]. Assessments of the energy used by tools like antivirus, spam filters, and firewalls have already been partially addressed (see, e.g., [8, references therein]). However, a clear understanding of what and how energy is depleted is still missing. Therefore, we decided to solely focus on basic security services, and in particular to evaluate the energy requirements of cryptographic algorithms. In fact, functionalities of tools used to enforce cyber security can be decomposed into simple mechanisms to guarantee communication integrity, non-repudiation of a message, authentication, as well as consistence of a generic fragment of information. Specifically, we considered the following classes of algorithms.

Encryption Algorithms: They take plaintext and a key as inputs and provide ciphered text as output. This operation can be performed by many different methods, for example, by means of text block expansion and reduction, data permutation, or substitution boxes [8]. The algorithms considered in this article are Advanced Encryption Standard Cipher Block Chaining (AES-CBC), AES-Electronic Code Book (AES-ECB), Blowfish, Data Encryption Standard Electronic Code Book (DES-ECB), 3DES, and Rivest Cipher 4 (RC4).

Hashing Algorithms: They are primarily used to check the data integrity by computing a fixed-length string against a text provided as input. Hash functions are usually engineered to provide a unique output difficult to invert [9]. The considered algorithms are Message Digest 2, 4, and 5 (MD2, MD4, MD5), RACE Integrity Primitives Evaluation Message Digest (RIPEMD)-160, Secure Hash Algorithms 1 and 2 (SHA-1, SHA-2), Tiger, and Whirlpool.

Keyed-Hash Message Authentication Codes (HMACs): Mainly used to avoid message and hash tampering, they offer a mechanism for message authentication by using cryptographic hash functions in combination with a secret shared key [10]. The considered algorithms are the same tested for the hashing case.

Table 1 briefly describes the different cyber security algorithms taken into account. We point out that some of them are outdated (e.g., SHA-1 has been considered insecure since 2010, and its support was dropped in 2016 from Google Chrome) or "flawed," that is, known vulnerabilities have been disclosed (e.g., DES-ECB and 3DES) [11]. However, since such methods are still widely adopted, especially in legacy devices, they have been considered for the sake of completeness.

Modeling

The crucial issue of constructing a qualitative model of the energy required by cyber security mechanisms is the poor granularity of the available data. This is a direct consequence of the adopted standard solutions. For instance, the length of the key used for encrypting information in a real scenario does not vary in a "continuous" way since only well defined, discrete values are considered. Accordingly, we decided to use a dataset containing only the energy consumption of "feasible" configurations adopted in production-quality environments.

A direct consequence of the "quantization" of the available configurations is the need to create a model of energy consumption that is simple but at the same time robust to noises characterizing the collected data. Toward this end, the relationship between the power consumption and the different cyber security methods was modeled through polynomials. In particular, we used a least squares technique to tune the coefficients by minimizing the mean square error between the available measurements and the output of the models (i.e., the so-called residuals). In order to limit the impact of noises, the degree of the polynomials had to be chosen to be much smaller than the number of available measures. This avoids the overfitting phenomenon, that is, the obtained models interpolate a random error (the noise) instead of approximating the real underlying relationship between the considered quantities.

More specifically, our goal is to approximate the relationships between the size of the key for data encryption and the energy consumption as well as between the amount of processed data and the power drain. Using least squares, the resulting models are very robust to noises, and the unknown parameters can be obtained by using simple algebraic equations; thus, there is no need to apply complex optimization procedures.

Measurement Methodology

To build the dataset used for modeling the energy consumption of cryptographic algorithms running in both end nodes and network devices, we conducted an extensive set of trials. In more detail, tests were performed to capture a mixed set of use cases, especially to understand the impact of the "strength" of security algorithms on the energy footprint. Thus, we performed experiments by varying the following parameters:

- *Algorithm:* For each algorithm, we evaluated its energetic impact to understand whether the complexity and the implementation play a role.
- *Size of the key:* We tested how varying the size of the key influences the required energy. As said, the sizes of the key have been selected to reflect production-quality requirements.
- *Load (or volume):* All the permutations of the aforementioned configurations were stressed

The crucial issue of constructing a qualitative model of the energy required by cyber security mechanisms is the poor granularity of the available data. This is a direct consequence of the adopted standard solutions. For instance, the length of the key used for encrypting information in a real scenario does not vary in a "continuous" way since only well-defined, discrete values are considered.

Algorithm	Acronym	Brief Description
AES-CBC	Advanced Encryption Standard — Cipher Block Chaining	The encryption is based on a substitution-permutation network. In this case, each block of plaintext is XORed with the previous ciphered block before being encrypted.
AES-ECB	Advanced Encryption Standard — Electronic Code Book	Simpler variation of the AES. In this case, the original message is divided into blocks, and each one is encrypted separately.
Blowfish	—	A Feistel-network-based block cipher.
DES-ECB	Data Encryption Standard — Electronic Code Book	It takes a fixed-length string of plaintext and transforms it through a Feistel network.
3DES	Triple DES	3DES applies the DES three times to increase robustness.
RC4	Rivest Cipher 4	It generates a pseudorandom stream of bits via permutation and pointers.
MD2, MD4, and MD5	Message Digests 2, 4, and 5	Hashing algorithms using different functions (4 in the case of MD5).
RIPEMD-160	RACE[1] Integrity Primitives Evaluation Message Digest	It is similar to MD, but it is considered more secure.
SHA-1 and SHA-2	Secure Hash Algorithms 1 and 2	A family of hashing functions using different architectures over the years to increase robustness.
Tiger	—	A collision-resistant hashing function based on the Merkle-Damgård principle.
Whirlpool	—	A more secure modification of the AES.

[1] RIPEMD was developed within the European Union Project RACE Integrity Primitives Evaluation (RIPE) project, 1988–1992, supported by the EU RACE Program — Research and Development in Advanced Communications Technologies.

Table 1. Considered cyber security algorithms.

Algorithm	Key size (bits)								
	64	128	160	192	198	256	384	512	1024
AES-CBC		•		•		•			
AES-ECB		•		•		•			
Blowfish	•	•		•		•		•	•
DES-ECB	•								
3DES		•		•					
RC4	•	•		•		•		•	•
MD2	•			•	•	•			
MD4	•			•	•	•			
MD5	•			•	•	•			
RIPEMD-160		•	•	•	•	•	•		
SHA-1			•	•	•	•	•	•	
SHA-2			•	•	•	•	•	•	
Tiger				•		•			
Whirlpool					•	•	•		

Table 2. Algorithms and key sizes considered for our tests.

with different traffic. In this way, we tried to assess the energetic scalability of the algorithms, also to enlighten some critical aspects of the implementation. The offered load was considered of increasing sizes, that is, 1 kB, 10 kB, 100 kB, 1 MB, 10 MB, 100 MB, and 1 GB, to account for usages ranging from a user device sending a small amount of data to a core/border appliance processing traffic groomed from a high-speed link.

To effectively measure the energetic footprint of cyber security, it is mandatory to remove possible overheads introduced by the hosting machinery. For instance, many algorithms need access to layer 2 (L2) interfaces or to interact with the buffering architectures of devices. This is a critical issue, as precisely identifying and quantifying the energy requirements of network components, protocols, and specific hardware subsystems are still mostly open problems [3]. Therefore, after preliminary evaluations of different configurations (i.e., emulated devices, virtual machine-based nodes, and real components), we decided to use a controlled framework built from scratch. The testbed was created using Linux (Ubuntu 14.04.2 LTS, GNU/Linux 3.16.0-20-generic ×86/64 kernel) running on an Intel Dual Core E2160 CPU at 1.80 GHz with 8 GB of RAM. To collect and process data, we implemented ad hoc Java and Python modules together with bash scripts. Regarding the cyber security algorithms, we used Java/Linux implementations, which is the choice commonly used in production-quality settings and Android-based mobile devices. In particular, we adopted the Bouncy Castle Cryptographic application programming interface (API) Libraries (release 1.55, August 2016 — https://www.bouncycastle.org/java.html, accessed November 2016).

As is commonly done in the literature, we estimated the required energy by exploiting the tight relation between the CPU usage and the consumed power [3, 6]. In fact, many works show that the power used for the computation is the predominant part of the energy consumed within a device (see, e.g., [8]). Therefore, along the lines of [12], we measured the used computing resources without considering overheads due to test conditions or other competing processes. In other words, we assumed that the consumed energy is proportional to the amount of CPU used for the processing, where the proportionality coefficient depends on the specific hardware/software technology. The amount of used CPU was measured for each configuration (i.e., type of algorithm, length of the key, and offered load), and samples were stored in a database for further processing. To model data, we used Matlab on a PC equipped with an Intel Core2 Duo CPU at 1.8 GHz and 2 GB of RAM.

Numerical Results

In this section, we present the results obtained through an extensive measurement campaign. Specifically, Table 2 showcases the considered algorithms jointly with all the different keys used for our investigation. Each configuration was tested with different loads, ranging from 1 kB to 1 GB. Regarding the polynomials used for modeling consumption, we fixed the degree to 2 to limit the impact of coarse-grained measurements. We point out that this limit is due to the fact that the length of the keys cannot vary "continuously"; rather, it must adhere to standard values (see the discussion above). As previously pointed out, trials focused on modeling a "qualitative" behavior. In fact, the precise understanding of how the technologies used for networking or computing contributes to energy drains has been an important topic for at least a decade [3, 6], and it is still part of ongoing research (see, e.g., [13] for the case of IoT-based scenarios). As an example, consumptions are highly influenced by the hosting hardware, such as commodity hardware vs. ad hoc field programmable gate array (FPGA)-based implementations. Therefore, identifying the proper value for the proportionality coefficient between energy consumption and amount of used CPU is challenging and outside the scope of this work. Thus, we just focused on the CPU used by a given algorithm to complete a task, which is a general abstraction of the energy consumed by an appliance to run the functionalities implementing the security layer. The reported results are "relative" values, that is, they are not absolute quantities, but scaled against the maximum measured CPU usages (taken equal to 1). With a little abuse of terminology, in the following we refer to "CPU usage," "energy," and "power" consumption interchangeably, as they are proportional.

Figure 1 depicts the CPU used by different encryption algorithms to process various traffic volumes using a 128-bit key. Similar results were obtained for other key lengths, but they are not reported for the sake of brevity. The main finding is that all the considered algorithms exhibit two different consumption profiles: an almost constant consumption trend for loads smaller than 10^7 bytes, and a linear increasing one for higher volumes of traffic (notice the logarithmic scale on the x-axis). The major exceptions are the RC4 and AES-ECB methods, which appear to be almost insensible to the amount of data to be processed. This also suggests the presence of major optimizations within the software implementation. In general, the ISP, data center engineers, and software developers should prefer more robust solutions having the same energy footprint. For instance, it turns out that using load-insensitive mechanisms is preferable if some form of load distribution is not possible (e.g., having distributed architectures processing in parallel smaller fractions of the overall traffic). As another example, results indicate that, when in the presence of a mobile population with limited power sources, it would be possible to trade energy for security, for instance, by using simpler or more efficient cyber security solutions.

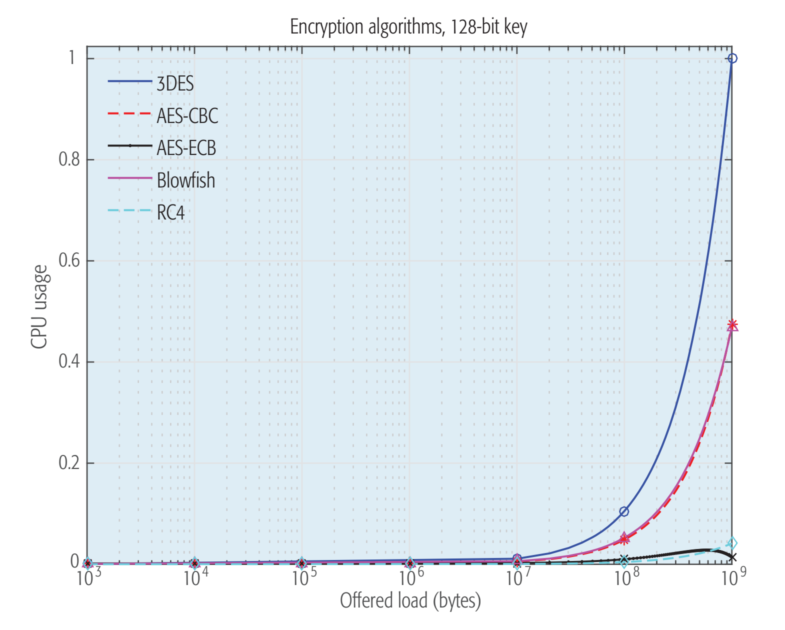


Figure 1. Relationship between CPU usage and offered load for different encryption algorithms with a 128-bit key.

Figure 2 shows the results of encryption algorithms when varying the length of the key. For the sake of compactness, we report only the results related to Blowfish and RC4. Such methods appear to be insensitive to the used key, thus making it preferable to adopt more robust solutions since they do not account for additional energy requirements or battery drains. Similar considerations could also be made for the remaining encryption algorithms, which have trends similar to the Blowfish one. The presence of some energy-insensitive techniques suggests that reducing the length of the key to pursue economic and energy savings could be useless.

Figure 3 displays the results obtained by investigating different hashing algorithms. Also in this case, two different zones characterize consumption, that is, the required CPU is almost constant for loads less than 10^7 bytes, while it increases linearly for larger loads. The Whirlpool algorithm is revealed to be quite power-hungry; hence, it is not suitable for mobile devices or to pursue energy efficiency. Instead, the remaining algorithms have similar consumptions, and therefore the ISP/data center operator can select the most suitable techniques without paying too much attention to energy.

Figure 4 presents the results for the case of HMAC algorithms with key length of 256 bits. The other key lengths provide comparable results but have been omitted for the sake of brevity. The reported trends are similar to the ones of Fig. 3. However, this is not surprising since HMAC offers message authentication by means of cryptographic hash functions. Therefore, the considerations regarding the energy requirements are the same as in the case of the hashing algorithms. It is worth noting that, for the MD2 case, the low degree of sophistication does not match with its high energy requirements. By performing additional investigations, we found that this is due to a poor software implementation of the algorithm. As a consequence, this showcases that code optimization can make a relevant difference in terms of economic expenditure for the energy bill and the quality of experience of end users, for example, by avoiding reduced lifetime of mobile devices due to excessive power depletion.

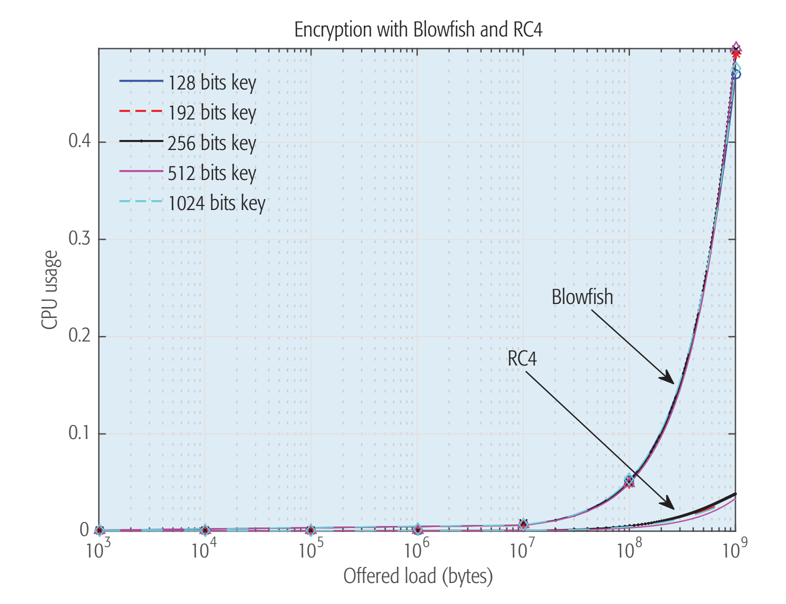


Figure 2. Relationship between CPU usage and offered load for Blowfish and RC4 by varying the length of the key.

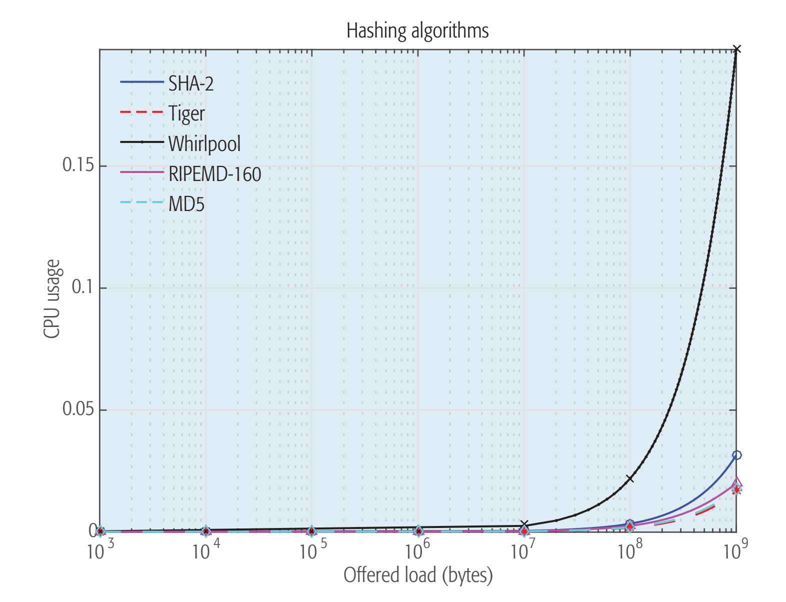


Figure 3. Relationship between CPU usage and offered load for hashing algorithms.

Lessons Learned and Conclusions

As discussed, understanding the energy requirements of cyber security techniques is fundamental for the development of green and secure network environments. The main lessons learned and possible future research directions are the following.

Refrain from Pursuing Economic and Energy Savings at the Price of Cyber Security: Our results indicate that algorithms like MD4, SHA-1, and SHA-2 have small energy footprints. However, as they are considered highly insecure, their adoption should be avoided even if favorable in energy.

Optimize the Code: The comparison of different cryptographic algorithms hints that software optimization could play a major role in terms of economic savings. For instance, the excessive consumption of MD2 reported in Fig. 4 is not fully justified by its computational requirements. In this vein, the next generation of mobile, trusted, and secure networks should not only be secure by-design, but also energy-efficient. Code optimization could also be an early and effective countermeasure to prevent energy-draining attacks [14].

Offload and Fragment if Needed: The obtained models of power consumption show two different behaviors characterizing cyber security techniques, that is, constant vs. linear for low vs. high loads. Therefore, load fragmentation may be favorable due to simpler and more efficient entities working in parallel. This could also be a benefit for nodes with limited capabilities, for instance, by offloading some security operations via a cloud-based paradigm. For the specific case of mobility provided by cellular networks, some security features could be implemented through a cloud radio access network model. However, the delegation "outside" the device makes the access to the cloud an additional point of fragility, which should be carefully assessed.

There Is Room for Runtime Optimizations: Since algorithms with similar security degrees have different consumption, some optimizations could be performed within the ISP or the data center, such as switching the security mechanisms to more energy-efficient ones if there are no foreseen risks. In other words, a choral coordination among firewalls, network probes, and anticipatory security systems could allow to trade between security and energy efficiency, if needed.

Precisely Knowing the Energy Required by Security Mechanisms Can Be Used as a Novel Marker to Perform Anomaly Detection and to Prevent Non-Scalable or Computationally Intensive Traffic Analysis: For instance, a growth in the power consumption could reveal the presence of some form of denial of service (DoS) or distributed DoS (DDoS) attacks (see, e.g., [5, references therein]). However, the detailed investigation of these topics, including long-lasting DDoS attacks, is left to future works.

Lastly, as a part of future developments, the approach proposed in this article could be used to improve high-level models such as the one reported in [15] to provide an online estimation of the used power. This can lead to additional benefits:

- Refine estimation of the energy drained in mobile nodes to help the optimization of architectures.
- Enhance models used to quantify the energy used by Internet-scale service providers by explicitly considering the contribution of security-related algorithms.
- Develop novel traffic analysis techniques able to correlate consumption with loads, for instance, for early attack detection.

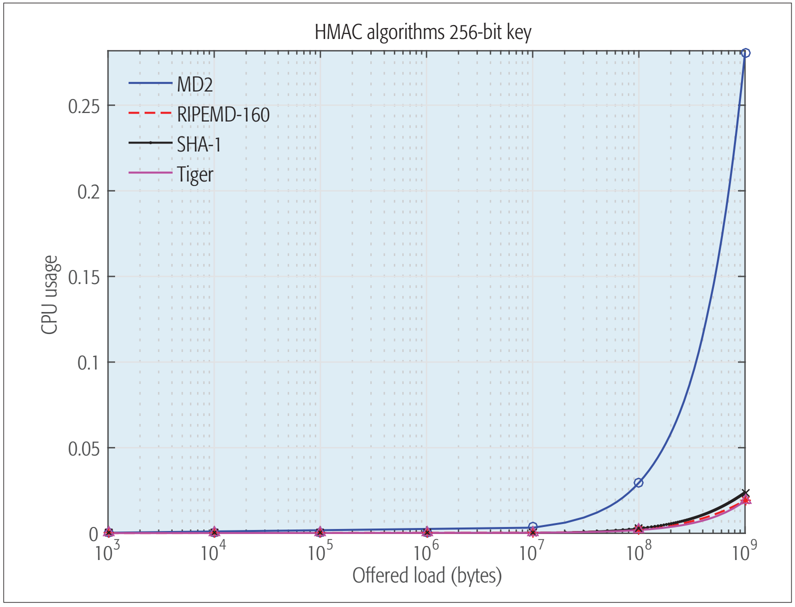


Figure 4. Relationship between CPU usage and offered load for HMAC algorithms with a 256 bit key.

Acknowledgment

This work has been partially supported by the research project STARTECO, funded by the Italian Ministry of University and Research (MIUR), project id DM62554.

References

[1] J. Jang-Jaccard and S. Nepal, "A Survey of Emerging Threats in Cybersecurity," *J. Computer and System Sciences*, vol. 80, no. 5, Aug. 2014, pp. 973–93.

[2] S. Subramanya *et al.*, "Beyond Energy-Efficiency: Evaluating Green Datacenter Applications for Energy-Agility," *Proc. 7th ACM/SPEC Int'l. Conf. Performance Engineering*, Mar. 2106, pp. 185–96.

[3] A. Bianzino *et al.*, "A Survey of Green Networking Research," *IEEE Commun. Surveys & Tutorials*, vol. 14, no. 1, 1st qtr. 2012, pp. 3–20.

[4] M. Gaggero and L. Caviglione, "Predictive Control for Energy-Aware Consolidation in Cloud Datacenters," *IEEE Trans. Control Systems Technology*, vol. 24, no. 2, Mar. 2016, pp. 461–74.

[5] A. Merlo, M. Migliardi, and L. Caviglione, "A Survey on Energy-Aware Security Mechanisms," *Pervasive and Mobile Computing*, vol. 24, Dec. 2015, pp. 77–90.

[6] N. R. Potlapally *et al.*, "A Study of the Energy Consumption Characteristics of Cryptographic Algorithms and Security Protocols," *IEEE Trans. Mobile Computing*, vol. 5, no. 2, Feb. 2006, pp. 128–43.

[7] N. Vallina-Rodriguez and J. Crowcroft, "Energy Management Techniques in Modern Mobile Handsets," *IEEE Commun. Surveys Tutorials*, vol. 15, no. 1, First Qtr. 2013, pp. 179–98.

[8] X. Li and F. T. Chong, "A Case for Energy-Aware Security Mechanisms," *Proc. 27th Int'l. Conf. Advanced Information Networking and Applications*, Mar. 2013, pp. 1541–46.

[9] T. Eisenbarth *et al.*, "A Survey of Lightweight-Cryptography Implementations," *IEEE Design & Test of Computers*, vol. 24, no. 6, 2006, pp. 522–33.

[10] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," IETF RFC 2104, Network Working Group, Feb. 1997.

[11] H. Alanazi *et al.*, "New Comparative Study between DES, 3DES and AES within Nine Factors," *J. Computing*, vol. 2, no. 3, Mar. 2010, pp. 152–57.

[12] L. Caviglione *et al.*, "Seeing the Unseen: Revealing Mobile Malware Hidden Communication via Energy Consumption and Artificial Intelligence," *IEEE Trans. Info. Forensics and Security*, vol. 11, no. 4, April 2016, pp. 799–810.

[13] W. Trappe, R. Howard, and R. S. Moore, "Low-Energy Security: Limits and Opportunities in the Internet of Things," *IEEE Security Privacy*, vol. 13, no. 1, Jan.-Feb. 2015, pp. 14–21.

[14] F. Palmieri, S. Ricciardi, and U. Fiore, "Evaluating Network-Based DoS Attacks under the Energy Consumption Perspective: New Security Issues in the Coming Green ICT Area," *Proc. Int'l. Conf. Broadband and Wireless Computing, Communication and Applications*, Oct. 2011, pp. 374–79.

[15] L. Zhang *et al.*, "Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones," *Proc. IEEE/ACM/IFIP Int'l. Conf. Hardware/Software Codesign and System Synthesis*, Oct. 2010, pp. 105–14.

Biographies

Luca Caviglione received his Ph.D. degree in electronics and computer engineering from the University of Genoa, Italy. Since 2007 he has been a researcher with the Institute of Intelligent Systems for Automation, National Research Council of Italy. His research interests include P2P systems, wireless communications, cloud architectures, and network security. He is an Associate Editor of *Transactions on Emerging Telecommunications Technologies*.

Mauro Gaggero received his B.Sc. and M.Sc. degrees in electronics engineering and his Ph.D. degree in mathematical engineering from the University of Genoa in 2003, 2005, and 2010, respectively. Since 2011, he has been a researcher with the Institute of Intelligent Systems for Automation, National Research Council of Italy. His research interests include control, optimization, and learning from data. He is an Associate Editor of the IEEE Control Systems Society Conference Editorial Board.

Enrico Cambiaso received his Ph.D. in 2016 in computer science from the University of Genoa with a thesis titled *Design and Development of Slow DoS Attacks*. His scientific interests are related to computer and network security, communication protocols, cyber-attacks, intrusion detection systems, covert channels, and cloud computing.

Maurizio Aiello graduated in 1994, and worked as a freelance consultant for universities, research centers, and private industries. Since August 2001, he has been responsible for the network infrastructure of the National Research Council of Italy. He is a teacher at the University of Genoa and University College of Dublin. He is a student coordinator, and manages fellowships and EU projects in the computer security field. His research interests are in network security and protocols.

On Understanding the Existence of a Deep Torrent

Rafael A. Rodríguez-Gómez, Gabriel Maciá-Fernández, and Alberto Casares-Andrés

The authors present an implementation of a complete system to crawl the Deep Torrent and evaluate its existence and size. We describe a basic experiment crawling the Deep Torrent for 39 days, in which an initial estimation of its size is 67 percent of the total number of resources shared in BitTorrent network.

ABSTRACT

Nowadays, a great part of Internet content is not reachable from search engines. Studying the nature of these contents from a cyber security perspective is of high interest, as they could be part of many malware distribution processes, child pornography or copyrighted material exchange, botnet command and control messages, and so on. Although the research community has put a lot of effort into this challenge, most of the existing works are focused on contents that are hidden in websites. However, there are other relevant services that are used to keep and transmit hidden resources, such as P2P protocols. In the present work, we suggest the concept of Deep Torrent to refer to those torrents available in BitTorrent that cannot be found by means of public websites or search engines. We present an implementation of a complete system to crawl the Deep Torrent and evaluate its existence and size. We describe a basic experiment crawling the Deep Torrent for 39 days, in which an initial estimation of its size is 67 percent of the total number of resources shared in the BitTorrent network.

INTRODUCTION

In the early days of the Internet, crawling the web was a relatively easy task. Search engines were able to index almost all the contents in the web. However, after some time, web contents have considerably evolved to a more "dynamic" behavior; for example, web servers often use databases to build and serve dynamic web pages. As pointed out in 1994 by Jill Ellsworth, this evolution leveraged the apparition of the *invisible* Web. In 2001, Bergman [1] divided the web contents into a *Surface Web*, that is, the indexed content crawled by search engines, and a *Deep Web*, containing all the dynamically generated content as a response to query forms. Nowadays, the Deep Web is more generally defined as informational content on the Internet that presents any of the following characteristics:

- It is not accessible through direct queries made by conventional search engines.
- It is only accessed through specific and targeted queries or keywords.
- It is either not indexed or cannot be indexed by conventional search engines.
- It is somehow protected by security mechanisms, including login IDs and passwords, certificates, membership registrations, codes, and so on.

Crawling the Deep Web is a challenging task, not only due to the hidden nature of its contents, but also because of its large scale. In 2001, Bergman [1] estimated that the Deep Web was 400–550 times greater than the Surface Web, and other authors [2] gave some insights about its size, including more than 307,000 sites, 450,000 databases, and 1,258,000 interfaces, describing an increase in size by 3–7 times during the period 2000–2004. In 2007, some Deep Web directory services started to index databases in the web, although their coverage was still small, ranging from 0.2 to 15.6 percent [2]. In 2015, the data stored on just the 60 largest Deep Web sites was estimated to be 40 times larger than the size of the entire Surface Web [3].

From a cyber security perspective, discovering and analyzing the structure and dynamics of this huge amount of hidden content is of paramount importance. Many illegal activities in cyber space are based on the existence of this hidden content. Common examples of this fact are the presence of malware propagation mechanisms, botnets communications, exchange of child pornography or copyrighted contents, and so on.

The use of services like peer-to-peer (P2P) is relevant to some of these illegal activities. As an example, some of the botnets studied in previous research works use *command and control* mechanisms that are based on existing P2P networks (parasite P2P botnets) [4]. Despite this fact, the crawling and analysis of the resources shared using these protocols has received little attention so far. Again, we can conclude that, from a cyber security perspective, studying the contents in P2P networks is essential.

In this article, we focus on the Mainline implementation of BitTorrent, the most used P2P network nowadays. The publication of a resource in BitTorrent is done by somehow sharing a `torrent` file containing metadata related to the content description and location of the shared resource. These `torrent` files are either published in the public web (specific websites for torrent files that are referred to as *torrent-discovery sites* herein) or simply sent to interested users in an out-of-band channel (email, deep web, IRC, etc.). This mechanism for publication led us to make an analogy with the classification of contents in the web, and divide the BitTorrent resources into two parts: those that are publicly announced in the web, which we denote as the *Surface Torrent*, and those that remain hidden

Digital Object Identifier: 10.1109/MCOM.2017.1600959

Rafael A. Rodríguez-Gómez and Gabriel Maciá-Fernández are with CITIC-UGR; Alberto Casares-Andrés is with 4IQ.

to the general public and are shared in private communities, that is, the *Deep Torrent*. Note that the concept of Deep Torrent does not include only *private torrents*. In fact, in this work we do not consider private torrents, as they have been analyzed in other works [5]. We specifically focus on resources that are being announced by out-of-band mechanisms instead of public websites, while still being shared in public trackers or DHT.

In the present work we focus on demonstrating the existence of such a Deep Torrent and evaluating its size. We propose the use of a combined crawler for both the Surface and Deep Torrent based on a Mainline monitoring module and a web crawler for torrent-discovery sites. The system combines the output of these two modules to obtain a list of Deep Torrent resources. We make an evaluation of the system for a 39-day period, extracting experimental results about the Deep Torrent. To our knowledge, there is no previous research published on analyzing or describing these phenomena.

The rest of the article is organized as follows. In the following section, some related work is presented. Following that, some fundamentals of BitTorrent-based networks are given. After that, the overall functional architecture for the proposed Deep Torrent crawling system is detailed. Then we describe the preliminary results obtained from our proposed system. The final section draws the main conclusions and points out directions for future work.

Related Work

Since the work by Bergman in 2001 [1], there have been some efforts in the research community to investigate the magnitude and features of the Deep Web. These efforts have been concentrated in two directions.

The first one is related to the understanding of the nature of hidden contents and the methodologies to automate data extraction from Deep Web sites [6]. The second direction of research is related to optimizing the number of queries used to dig the web in order to obtain the maximum percentage of hidden contents [7]. The first prototype of our system is aligned with the first direction of research, as we are really interested in getting an overview of the features of the Deep Torrent without caring too much about efficiency.

Additionally, there are works specifically focused on crawling torrent-discovery sites [8, 9]. These works are not really focused on extracting information about the Deep Torrent, as they are only able to get information about the torrent files publicly published in torrent-discovery sites (Surface Torrent). Among torrent-discovery sites, it is worth mentioning the existence of the so-called distributed hash table (DHT) search engines, which publish information (magnet links instead of torrent files) about resources being shared in the BitTorrent DHT. The first engine capable of searching the BitTorrent DHT was btdigg. This engine was active during our research period and closed in June 2016 for several months. It is currently active again under a different domain (https://btdig.com/). In December 2016, a new DHT search engine called `Alphareign` (https://alphareign.se/) appeared. Up to our knowledge, these are the only DHT search engines up to date.

Regarding the research efforts related to monitoring activity in BitTorrent, in our previous work [4] we developed a monitoring system to detect files belonging to P2P parasite botnets. There are other similar approaches in the literature, like [10–12]. Unluckily, all of them are aimed to analyze some features of BitTorrent files without paying attention to the Deep Torrent phenomena.

Finally, it is important to highlight that the combination of both modules, that is, a Surface Torrent crawler in the web and a BitTorrent monitor, in a complete system to crawl the Deep Torrent is not present in the literature, and it represents a contribution of this work.

BitTorrent General Concepts

The BitTorrent protocol is used to share resources among peers in a large network. For every resource shared in BitTorrent, the nodes of the network can play different roles: *seeders* are nodes that contain a complete copy of a shared resource; *leechers* are those that have partially downloaded the considered resource — note that leechers really download the parts of a resource not only from seeders, but also from other leechers — and finally, *trackers* are special nodes in the network that keep track of the leechers and seeders for every shared resource.

To locate the resources shared in the network, the BitTorrent protocol uses `torrent` files, which contain metainformation about resources and, when necessary, about their corresponding trackers. The 20-byte SHA-1 hash of the `info` section of a `torrent` file is called `infohash`, and it uniquely identifies a resource in the network.

`Torrent` files are stored in torrent-discovery servers (normally web-based) that allow users to search contents and then get the corresponding `torrent` file to start the corresponding download. Some examples of these torrent-discovery sites are https://thepiratebay.org/, https://torrentdownloads.me/ or http://extratorrent.cc/, among others.

Since 2005, the BitTorrent protocol implements a distributed operation mode that does not require the participation of trackers. It was first implemented in the Azureus torrent client (currently known as Vuze). In this operation mode, a DHT is used to store the correspondence among the resources and the peers that share them. Here, we could say that each peer plays the role of a tracker. Currently, there are two different incompatible implementations of DHT: Vuze and Mainline. Both are specific implementations of Kademlia [13]. In this article we focus on Mainline [14], as its use is more widespread [8].

Mainline uses 20-byte unique identifiers for both nodes and resources (infohash) in the DHT network. In the case of nodes, they are known as nodeIDs, and are randomly generated the first time a BitTorrent client is initiated. These identifiers will not change unless a user manually uninstalls the BitTorrent application or changes its configuration file. Even if a user changes its IP address, its nodeID will remain, and for this reason, we can assume that nodeID is a unique identifier per user.

In Mainline, a metric for the closeness between a DHT node and a resource is defined as the XOR operation between their corresponding identifiers:

From a cyber security perspective, discovering and analyzing the structure and dynamics of this huge amount of hidden content is of paramount importance. Many illegal activities in the cyber space are based on the existence of this hidden content.

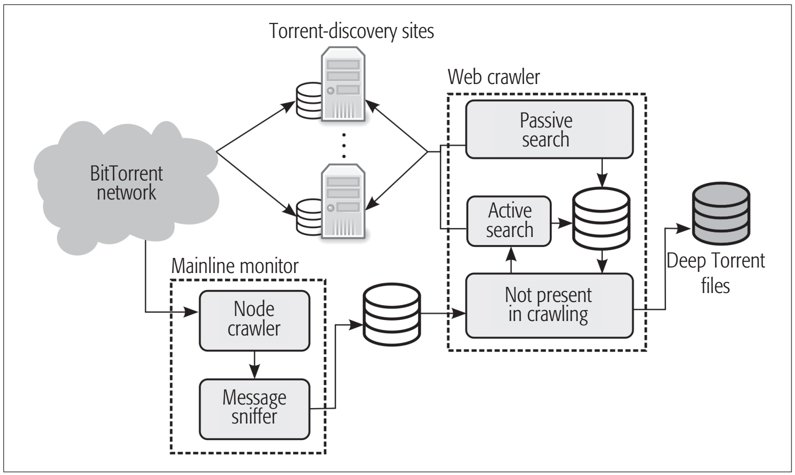


Figure 1. Functional architecture of the Deep Torrent crawler.

`nodeID` and `infohash`. The DHT nodes that are closer to a resource are in charge of keeping track of the list of peers sharing it.

There are four queries in the Mainline DHT protocol:

- `ping:` verifies if a peer is alive and responsive.
- `find_node:` requests a node for the list of closest nodes to a given nodeID in its routing table. A response message is issued with the IP address, port, and `nodeID` of every node in this list.
- `announce_peer:` announces that a peer holds the resource (or a part of it identified by its `infohash`.
- `get_peers:` get a list of peers associated with a infohash. If the queried DHT node does not have this information, it returns the eight nodes in its routing table closest to the `infohash` supplied in the query.

Then, if a peer wants to announce that it has a copy of a given resource `infohash`$_i$, it has to first find the list of peers that are closest to `infohash`$_i$. For this purpose, it sends `get_peers` messages that iteratively reach the nodes in the DHT containing this information, thus getting the response. After that, an `announce_peer` message is sent to the nodes in the list of peers. As this information expires after a timeout that depends on the client implementation (around 30 minutes), the announcing peer is responsible for re-announcing the tuple `<IP:port,infohash`$_i$> over time.

Note that announcing the resources is a necessary condition to allow other nodes in the Mainline network to download them. Based on this fact, we reduce the problem of monitoring the shared resources to that of monitoring `announce_peers` messages in the network. In what follows, we describe how we manage to achieve this.

Deep Torrent Crawler

The proposed Deep Torrent crawler is based on two modules (Fig. 1): a Mainline monitor and a web crawler for torrent-discovery sites. The Mainline monitor module is in charge of obtaining the resources that are being actively announced in the BitTorrent network. In parallel, the web crawler extracts resources that can be found in torrent-discovery sites (Surface Torrent). Finally, both forms of data are combined to find the resources that are really being announced in the BitTorrent network but cannot be found in the torrent-discovery sites. Following our own definition, these would be the resources that belong to the Deep Torrent.

Mainline Monitor Module

The monitoring module for the Mainline network is based on our previous work [4] and is composed of two submodules: a node crawler and a message sniffer.

Node Crawler: The purpose of the node crawler is to obtain all the active peers in a specific zone of the Mainline network and to maintain the updating of this list. A zone of the network is defined as all the identifiers with a common prefix. For example, the crawler can monitor an 8-bit prefix zone by extracting all the active nodes in the network whose nodeID begins with the same eight bits.

The crawling process starts getting a list of nodes in the monitored zone, `known_list`, by recursively sending `find_nodes` messages to some hardcoded bootstrap nodes. Once it has a minimum number of known nodes, two threads are launched. One periodically asks the nodes in `known_list` about new ones, and the other thread receives their answers and registers the new nodes into `known_list`.

Message Sniffer: Its aim is to include our monitor node in the routing tables of the DHT nodes previously collected in `known_list`. To accomplish it, this module periodically sends ping messages to the DHT nodes, indicating that it is alive and responsive. In this process, we forge the source nodeID so that many different sybil nodes are included in routing tables. As we are interested in receiving the same messages as the nodes in `known_list`, the fake nodeIDs are chosen so that they are close to them.

In summary, the Mainline monitoring procedure works as follows. First, we obtain the active nodes of a specific zone by using the node crawler module. After that, we try to be inserted into the routing tables of these nodes by including our sybils as neighbors. As a result, legitimate nodes send `announce_peer` messages to our sybils when they are sharing a resource with `infohash` in the monitored zone. We log all these `announce_peer` messages into a database, registering the `infohash` of the announced resource, IP address, port, `nodeID` of the announcer node, and the message arrival timestamp.

Note that this module does not alter in any way the proper operation of the monitored zone. The only effect is that real nodes in the monitored zone will send some extra messages to our sybils.

Web Crawler Module

Recall that torrent-discovery sites publish `torrent` files that are used to start the download of a specific resource. These sites usually have a query interface that allows users to obtain information related to the searched torrent resources. Based on this information, a user is able to decide which is the best `torrent` file for downloading a given resource.

In order to make our crawler capable of extracting this knowledge, we use two methods.

- Passive search: Information announced in the torrent-discovery sites is obtained by using Rich Site Summary (RSS) feeds.
- Active search: We also query special websites for the resources already identified in the monitoring of the Mainline network and focus on those that have not been previously identified in the rich site summary (RSS) data source.

Regarding passive search, RSS feeds of the monitored sites are periodically queried by our crawler, and all the announced resources are stored in a database of known resources. The information stored in this database is:

- Unique identifier of the resource (`infohash`)
- Name of the resource
- Size in bytes
- Number of seeders and leechers
- Timestamp at which this resource started to be shared
- Website from which this information was obtained
- Timestamp of the instant at which the crawler got the information

The idea of the active search is leveraging the information already extracted from the Mainline monitor module to make a deeper search of indexed resources in the torrent-discovery sites. Here, all the resources identified in the Mainline monitor module that have not been found in the queried RSS are first identified. For each of them, using the `infohash` announced in Mainline, a new specific query is launched to certain websites that allow searching a torrent by its `infohash`. Only when a resource is not found at this point is it labeled as a hidden resource and stored in the Deep Torrent resources database.

Measurement Results

We monitored a part of the Mainline network for 39 days, from March 16, 2016 until April 24, 2016. During this period, two Mainline monitors were launched to monitor the zones with an 8-bit prefix equal to `0x09` and `0x10`, respectively. This represents 128 of the complete Mainline network (2 zones of a size of 1 out of 256 each). As `nodeID`s are randomly assigned, we consider that this sample is representative of the behavior of the whole Mainline network. The main reason to use two different sensors is to check if the obtained results are biased for a specific zone.

We have conducted a preliminary experiment to check the accuracy of our crawler when sniffing the `infohash`es announced in a given zone. Three different instances of the Mainline monitor have been launched in the same zone (`0x09`). In this setup, our estimation for the percentage of resources monitored by the Mainline crawler is the percentage of resources observed by the three sensors. Thus, any resource monitored by only one or two sensors is considered to be a non-observed resource in the monitoring (worst case). In Fig. 2 we can see that our estimation is that more than 90 percent of the resources are being monitored. Note that this number is in line with the performance already indicated in other works [11].

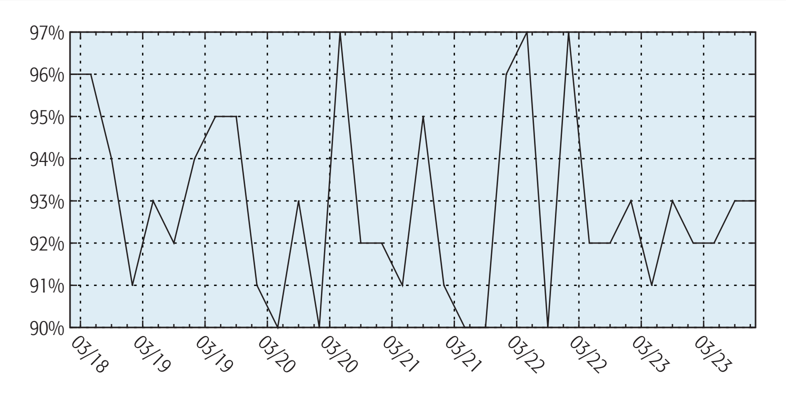


Figure 2. New `infohash`es discovered every hour during the first week.

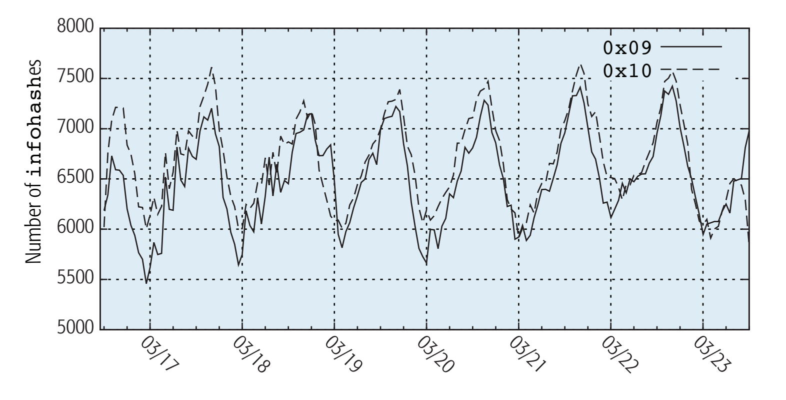


Figure 3. Percentage of coincidence between `infohash`es monitored by three different sensors in zone `0x09`.

A total of 321,962 different resources have been monitored during this period, 166,035 in the `0x09` zone and 155,927 in the `0x10` zone. We can see in Fig. 3 the evolution of the new `infohash`es discovered every hour during the first week in both zones. The similar behavior of both monitors lead us to the conclusion that these results could be generalized to other zones.

For each of the monitored resources we have stored every `announce_peer` message received, logging the `infohash`, origin `nodeID`, IP address and port, and timestamp. In Fig. 4 we can see the evolution of the number of peers communicating within the `0x09` zone for the first week of our monitoring period. The number of peers exhibit a periodic behavior with an increasing mean value that stabilizes after some days. Depending on the time of day, around 95,000 peers are actively sending/receiving messages to/from our system.

As reported in the Sandvine 2015 report [15], Asia is the continent with the highest percentage of BitTorrent usage. This fact is reflected in Fig. 5, where we show the monitored IP addresses grouped by its continent geolocation. Note that, due to this greater percentage of users from Asia, we obtain a periodic wave showing the typical evolution in day/night traffic in Fig. 4. In fact, using the time UTC+8 (China), the maximum number of peers is reached at 9 p.m. and the minimum at 4 a.m.

These resources have been shared by 86,915,611 different nodes (different `nodeID`s) with 23,417,933 different IP addresses from all

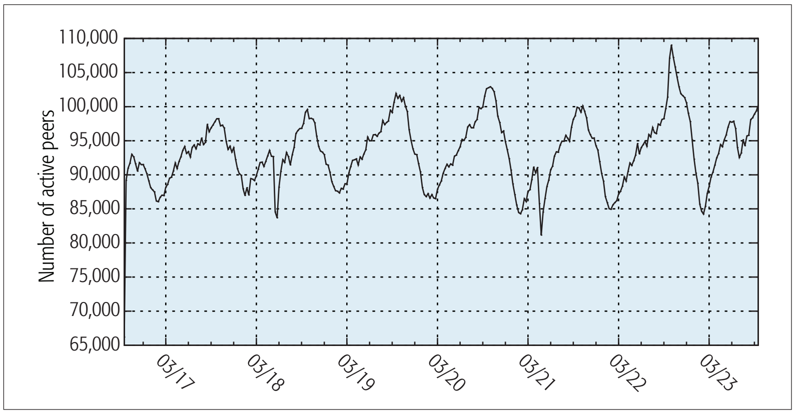


Figure 4. Evolution of the number of connected peers (`0x09` zone) during the first week.

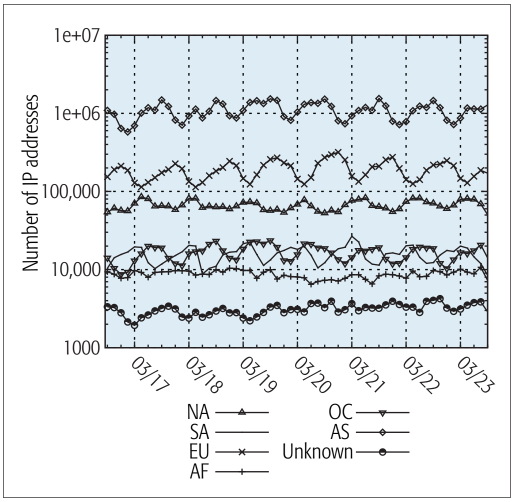


Figure 5. Evolution of the number of different IP addresses grouped by continent during the first week (`0x09` zone).

the continents. Note the huge difference between the number of nodes and IP addresses. This could be due to either the existence of network address translation (NAT) boxes or the use of sybil mechanisms. For example, DHT search engines like `bitdigg` make use of sybil procedures to collect information from a network, in a similar way as we are doing in our Mainline crawler. A prior inspection of these data showed that there are certain IP ranges that comprise a huge number of nodes. As an example, two IP ranges from Russia and Kazajstán contain 8 million and 6 million `nodeIDs`, respectively, presenting a mean value of 14,000 `nodeIDs` per IP. Due to this size, we consider it more likely that they have a sybil behavior than that they are NAT boxes.

Estimation of the Deep Torrent Size: Using the web crawler module, we have conducted our *passive search* since December 20th 2015, receiving information from some of the most common torrent-discovery sites. First, we chose a meta-search engine, `torrentz` (https://torrentz.eu/), due to the fact that it allows searching information in a large list of other torrent-discovery sites. During the monitoring period, `torrentz` comprised a list of 29 torrent sites.[1] In addition, we also directly checked some of the more relevant torrent-discovery sites, including http://bitsnoop.com/ and https://piratebay.to/, among others. Finally, we also decided to collect information from https://btdigg.org/, the only DHT search engine at the time of the experiment.

Each of these sites generates a periodic report with the newest torrent resources, which are subsequently downloaded and stored by our crawler. Depending on the torrent-discovery site, the frequency of the crawling varies between 24 and 48 hours. As a result, we have stored in our database a total of 22,174,122 resources. Out of the 321,962 resources collected in the Mainline monitor module, we found 80,869 (25.12 percent) within the 22 million resources obtained by the web crawler.

For the rest of the resources (a total of 241,093), we conducted an *active search* using some of the most common torrent-discovery sites that allow finding torrent resources by using their `infohash`. After this, we only found information about 23,878 additional resources of our set. In the end, we have 217,215 unidentified resources, which supposes 67.47 percent of the monitored resources. This is our estimated size of the Deep Torrent. Note that these results are only a proof of concept, as more exhaustive search methodologies for the Surface Torrent could be followed. Anyway, the obtained percentage points out that the size of Deep Torrent is not negligible at all.

Exploring Features of Shared Resources: One application of the web crawler is to explore the meta-data included in `torrent` files to draw conclusions about the contents and the sharing mechanisms.

For example, we wanted to inspect the *active duration* of the sharing of resources in order to find out possible differences between the Deep and Surface Torrent resources. This duration is defined as the number of hours during which the monitor receives messages announcing a specific resource. The results can be seen in Fig. 6. First, note that Deep Torrent resources are shared for less time. This is an expected result, as these resources are not publicly published in torrent-discovery sites and therefore are not expected to be very popular. Indeed, there are a total of 159,195 Deep Torrent resources with less than 5 hours of active duration, which supposes 73.29 percent of the total amount of Deep Torrent resources, while the number of Surface Torrent resources with less than 5 hours of active duration is 25,015 (23.88 percent of the total amount of Surdace Torrent resources). However, it is notable that many of the resources in the Deep Torrent are still being shared for a long time.

Conclusions and Future Work

This article explores the Deep Torrent, that is, torrents available in BitTorrent that cannot be found by means of public websites or search engines. We discuss the necessity of studying its properties, proposing a system to crawl Deep Torrent resources that combines a Surface Torrent crawler for the web and a BitTorrent (Mainline) monitor.

For demonstrating the usefulness of the crawler, we have collected information from part of the Mainline network over 39 days, identifying a total of 321,962 resources. Among them, 32.53 percent belong to the Surface Torrent, that is, they can be found in torrent-discovery sites; and the remaining

[1] See complete list at http://web.archive.org/web/20160323031455/http://torrentz.eu/help

67.47 percent are part of the Deep Torrent. We have shown how the information obtained from the crawler is proven to be useful to extract interesting characteristics of the Deep Torrent.

Despite these results, we consider that there are many interesting details and questions to be solved as part of future work. Specifically, we plan to work on:

- Extending the monitoring period and the number of monitored zones to derive more general results
- Thoroughly studying the features of the resources in the Deep Torrent and comparing them to those of the Surface Torrent
- Including a new module in our system to automatically download Deep Torrent resources in order to study them in a posterior phase
- Trying other techniques for the crawling of Surface Torrent by our web crawler

Acknowledgments

This work has been partially supported by Spanish Government-MINECO (Ministerio de Economía y Competitividad) and FEDER funds, through project TIN2014-60346-R and the Corporacion Tecnológica de Andalucía through project CTA 15/795.

Refrences

[1] M. K. Bergman, “White Paper: The Deep Web: Surfacing Hidden Value,” *J. Electronic Publishing*, vol. 7, Aug. 2001.
[2] B. He *et al.*, “Accessing the Deep Web,” *Commun. ACM*, vol. 50, May 2007, pp. 94–101.
[3] D. Sui, J. Caverlee, and D. Rudesill, “The Deep Web and the Darknet: A Look Inside the Internet’s Massive Black Box,” tech. rep., Oct. 2015.
[4] R. A. Rodríguez-Gómez *et al.*, “Resource Monitoring for the Detection of Parasite P2P Botnets,” *Computer Networks*, vol. 70, no. 0, 2014, pp. 302–11.
[5] X. Chen, Y. Jiang, and X. Chu, “Measurements, Analysis and Modeling of Private Trackers,” *Proc. 2010 IEEE 10th Int’l. Conf. Peer-to-Peer Computing*, 2010.
[6] M. Balduzzi and V. Ciancaglini, “Cybercrime in the Deep-Web,” *Proc. Black Hat 2015 EU*, Amsterdam, The Netherlands, 2015.
[7] Y. He *et al.*, “Crawling Deep Web Entity Pages,” *Web Search and Data Mining*, 2013, pp. 355–64.
[8] C. Zhang *et al.*, “Unraveling the BitTorrent Ecosystem,” *IEEE Trans. Parallel Distrib. Sys.*, vol. 22, July 2011, pp. 1164–77.
[9] H. Jin *et al.*, “Inaccuracy in Private Bit- Torrent Measurements,” *Int’l. J. Parallel Programming*, vol. 43, Oct. 2013, pp. 528–47.
[10] M. Steiner, T. En-Najjary, and E. W. Biersack, “A Global View of KAD,” *Internet Measurement Conf.*, 2007.
[11] G. Memon *et al.*, “Montra: A Large-Scale DHT Traffic Monitor,” *Computer Networks*, vol. 56, Feb. 2012, pp. 1080–91.

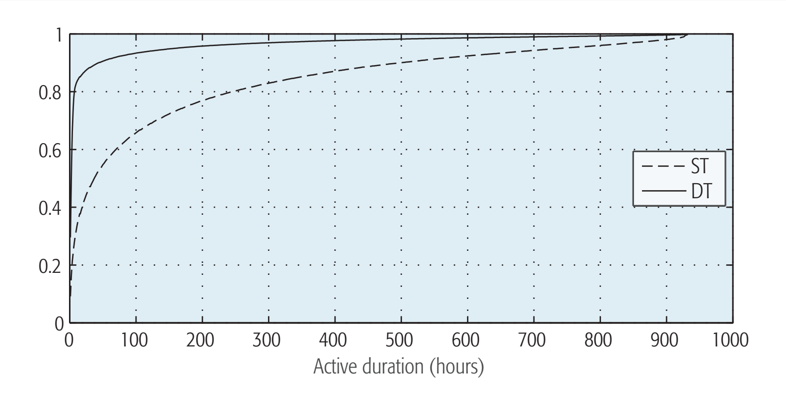


Figure 6. Normalized cumulative histogram of the active duration of Deep Torrent (DT) and Surface Torrent (ST) resources.

[12] M. Varvello and M. Steiner, “DHT-Based Traffic Localization in the Wild,” *Proc. 2013 IEEE INFOCOM*, Apr. 2013, pp. 3141–46.
[13] P. Maymounkov and D. Mazières, “Kademlia: A Peer-to-Peer Information System Based on the XOR Metric,” *Revised Papers from the 1st Int’l. Wksp. Peer-to-Peer Systems*, IPTPS ’01, 2002, pp. 53–65.
[14] “Mainline DHT Implementation,” http://bittorrent.org/beps/bep 0005.html, accessed 12 May 2016.
[15] Sandvine, “Global Internet Phenomena Asia-Pacific & Europe,” 2015.

Biographies

Rafael A. Rodríguez Gómez is a postdoctoral researcher in the Department of Signal Theory, Telematics, and Communications at the University of Granada, Spain. At the same university, he studied telecommunication engineering (2003–2008). After that, he received his Ph.D.; his thesis was *P2P Networks Protection through Traffic Analysis*. Nowadays, his research interest is focused on network security, more specifically on P2P networks, P2P traffic analysis, as well as detection and defense against attacks related to these networks.

Gabriel Maciá-Fernández is an associate professor in the Department of Signal Theory, Telematics, and Communications of the University of Granada. He received an M.S. in telecommunications engineering from the University of Seville, Spain, and a Ph.D. in telecommunications engineering from the University of Granada. His research interests are focused on computer and network security, with special focus on intrusion detection, ethical hacking, reliable protocol design, network information leakage, and denial of service.

Alberto Casares-Andrés is working for 4iQ in the CTO’s office researching new security threats and finding the differential value in that difficult market. He was responsible for engineering management from 2011 to 2016, and is an expert in data mining and project management. He has led several I+D+i projects supported by the Spanish Ministry of Industry. He holds a Master’s degree in soft computing and intelligent systems and a Computer Engineering degree from the University of Granada.

Toward Stream-Based IP Flow Analysis

Tomas Jirsik, Milan Cermak, Daniel Tovarnak, and Pavel Celeda

The authors discuss the transformation of present IP flow analysis into a stream-based approach to face current challenges in IP flow analysis. They examine the possible positive and negative impacts of the transformation and present examples of real-world applications, along with their recommendations.

ABSTRACT

Analyzing IP flows is an essential part of traffic measurement for cyber security. Based on information from IP flows, it is possible to discover the majority of concurrent cyber threats in high-speed, large-scale networks. Some major prevailing challenges for IP flow analysis include, but are not limited to, analysis over a large volume of IP flows, scalability issues, and detecting cyber threats in real time. In this article, we discuss the transformation of present IP flow analysis into a stream-based approach to face current challenges in IP flow analysis. We examine the possible positive and negative impacts of the transformation and present examples of real-world applications, along with our recommendations. Our ongoing results show that stream-based IP flow analysis successfully meets the above-mentioned challenges and is suitable for achieving real-time network security analysis and situational awareness.

INTRODUCTION

Monitoring IP flows and their analysis play a vital role in network traffic measurements for cyber security. Currently, IP flows are broadly used for traffic measurement in large-scale, high-speed networks, cloud environments, and various enterprise networks [1]. IP flow analysis is used for detecting the majority of severe contemporary cyber threats, such as denial of service (DoS), botnets, and advanced persistent threats (APTs). Moreover, the analysis can be done on both unencrypted and encrypted traffic, as IP flows gather information only from packet headers. Such advantages have made IP flow monitoring a fundamental part of network traffic measurement for cyber security.

Nevertheless, IP flow analysis still faces several challenges raised by the rapid evolution of the threat landscape. First, network traffic measurement has become a big data problem. Due to the increasing volume and velocity of network traffic, it has become expensive and impractical to first store and then read again all IP flows from large networks for analysis. Second, it has become impossible to analyze a large volume of IP flows from networks in real time. This plays an important role in automated defense mechanisms that need to take an action as soon as possible [2]. Current approaches try to face this challenge by increasing the hardware performance of analytical machines or simple master-slave architectures. Nevertheless, the IP analysis itself stays centralized, scalability of these solutions is limited, and analysis time still remains relatively long. Last, but not least, the time needed to detect a cyber attack is a challenge for IP flow analysis. Current IP flow-based cyber security solutions exhibit a detection delay on the order of minutes. Such a delay may be fatal when we try to reduce the harm caused by an attack [3]. Therefore, demands for near real-time attack detection have risen recently.

To address the above-mentioned challenges, we present a transformation of current IP flow monitoring and analysis into a scalable stream-based approach. In the stream-based approach, the IP flows are processed and analyzed in data streams immediately after an IP flow is observed. The analysis of IP flows in data streams reduces the volume of data that needs to be stored. This is because data is kept in primary memory for the time necessary for processing, and only results are stored in the secondary memory. This represents the greatest advantage of the stream-based concept. It allows the user to perform an immediate data analysis, which makes real-time attack detection possible. Moreover, thanks to the distribution of data streams within a computing cluster, this is possible even in large-scale, high-speed networks.

In this article, we describe the transformation of current IP flow traffic measurement and analysis toward a scalable stream-based solution. We present a workflow of stream-based IP flow analysis, along with its prototype implementation. Capabilities of the approach are demonstrated on cyber security use cases followed by practical implications for its usage.

BASIC CONCEPTS

To cover the basic concepts used in this article, we provide an overview of IP flow-based network monitoring and data stream processing. This overview can be considered as a high-level abstraction of the main ideas of both areas; for a detailed description, consult [1, 4, 5].

NETWORK IP FLOW MONITORING

IP flow monitoring was principally designed to monitor high-speed network traffic in large-scale networks. Since the performance limitations do not allow processing, storing, and analyzing all information from each packet in such networks (deep packet inspection), an abstraction of single direction communication, called an IP flow, was introduced. An IP flow is defined as a set of packets passing through a point in the network during a certain time interval. All packets belonging to a particular IP flow have a set of common properties called flow keys (Internet Engineering Task

Digital Object Identifier: 10.1109/MCOM.2017.1600972

The authors are with Masaryk University.

0163-6804/17/$25.00 © 2017 IEEE

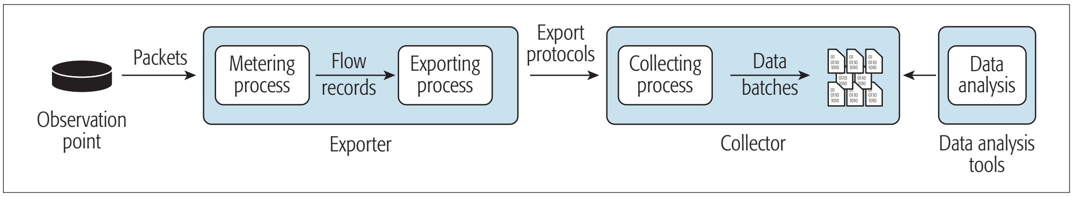


Figure 1. Traditional workflow of network IP flow monitoring and analysis.

Force [IETF] RFC 7011). The traditional 5-tuple of flow keys consists of source and destination IP address, source and destination port, and transport protocol. Apart from the traditional 5-tuple, the IP flow contains statistics about the connection (e.g., the number of packets in an IP flow), and may be enriched by information from the application layer of network traffic. IP flow information is stored in flow records.

The traditional workflow of IP flow monitoring consists of several different interconnected systems, as depicted in Fig. 1. A flow record is generated at an observation point in a network by a flow exporter. The exporter captures information from packet headers and creates flow records during the metering process. The created flow records are submitted to the exporting process to be sent to a flow collector via export protocols, such as NetFlow or IPFIX. The collector receives flow records from one or more exporters, processes them, and stores them for further analysis. The collecting process manages flow records and stores them, usually in one- to five-minute batches into binary flat files (e.g., nfdump — http://nfdump.sourceforge.net/, SiLK — https://tools.netsa.cert.org/silk/) or column-oriented databases (FastBit — https://sdm.lbl.gov/fastbit/, Vertica — https://www.vertica.com/, etc.). Row-oriented databases (e.g., MySQL, PostgreSQL) are not suitable for flow storage and querying due to their insufficient performance. Individual batches are then available for further data analysis.

There are three main application areas of IP flow analysis: flow analysis and reporting, threat detection, and performance monitoring [1]. Flow analysis and reporting covers querying and filtering flow data for relevant information (network visibility), statistics overview of the data (top *N* statistics, etc.), traffic accounting, reporting, and alerting (e.g., exceeding transfer data quota). The threat detection area focuses on the analysis of specific traffic events, most often scans, DoS, worms, and botnets [4]. Performance monitoring reports the status of running services on the network by observing application metrics, such as delay, jitter, and round-trip time.

All three application areas for IP flow analysis have one thing in common: a time aspect. As the network is monitored in time, the majority of statistics, detection methods, and performance characteristics are aggregated over a given time window (e.g., top talkers in the last hour, the number of transferred bytes in the last minute). The time window is strongly influenced by the settings of the network monitoring process, that is, the size of the batches in the collecting process. Data analysis can be performed only when a new batch occurs. The batch is set to five minutes in the majority of IP flow analysis tools. The analysis is then run every five minutes. This means that, for example, an attack or service outage may be detected with a five-minute delay. Such delay causes automatic defense mechanisms to take action too late, and protected systems are more affected. However, the demand for real-time analysis has risen recently in order to achieve shorter detection and reaction times. The analysis delay can be shortened by replacing the batch-based analysis with stream-based analysis, where each flow record is analyzed immediately as it arrives.

Traditional processing	vs.	Stream processing
Data stored as persistent sets	Data	Infinite streams of individual data tuples
Large secondary memory	Storage	Small primary memory
Ad hoc	Queries	Continuous
No real-time capabilities	Real-time	Real-time processing
Single-query	Optimization	Multi-query
Mature tools and technologies	Maturity	New tools and technologies

Table 1. Differences between traditional and stream data processing.

Data Stream Processing

Stream processing systems (historically referred to as data stream management systems [5]) emerged in response to the poor performance of traditional persistent databases, which were not designed for the rapid and continuous updates of individual data items continuously arriving at high velocities. The key differences between traditional data processing and stream processing are summarized in Table 1.

Data stream is a possibly infinite, discrete, and ordered sequence of data elements with a given schema and assigned timestamp. Stream processing systems are designed to evaluate continuous queries over many data streams in real time, while predominantly using only primary memory for storage. The existing implementations of stream processing systems differ in several aspects, including, but not limited to, query language capabilities, nature of processed data, time model, and so on. For example, Esper (http://www.espertech.com/esper/) is a full-fledged stream processing engine focused on evaluating continuous SQL-like queries over streams of events. For an extensive survey of stream processing systems, see [6].

Nowadays, a new generation of stream processing systems is emerging that is generally referred to as distributed stream processing frameworks. These systems are used to process generic data streams and provide capabilities for distributed processing. In many cases, users must

Analyzing IP flows in real time was almost impossible previously due to the poor performance of data processing systems. In recent years, however, a change has occurred, and a number of scalable systems for fast batch-based and stream-based processing of large volumes of data were progressively introduced.

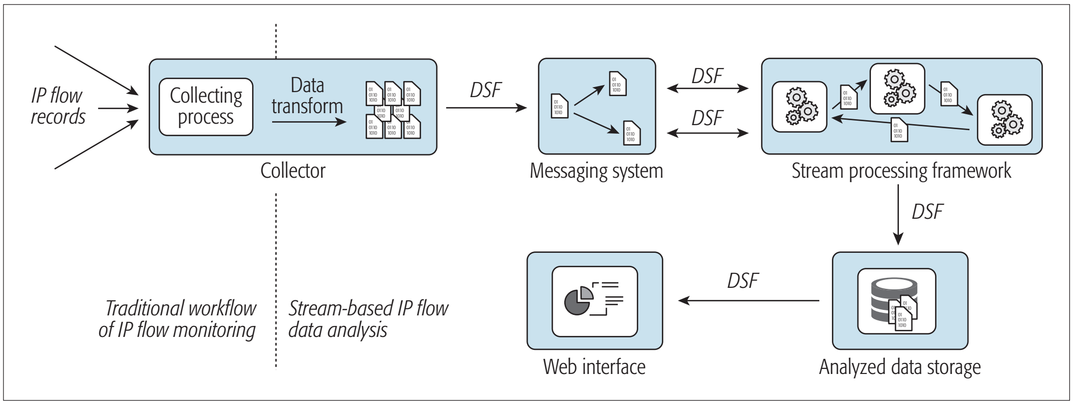


Figure 2. Stream-based workflow of network IP flow monitoring and analysis.

implement their own processing logic, but they are provided with powerful abstractions that allow them to transparently execute the implemented logic in a parallel distributed way. The most notable examples of distributed stream processing frameworks include Samza, Spark, Storm, and Flink (all maintained by Apache Software Foundation — http://{samza|spark|storm|flink}.apache.org/).

Stream-Based Workflow of IP Flow Analysis

The transformation of the traditional workflow of network IP flow monitoring into a stream-based one raises new challenges and requirements that must be addressed. We summarize the functional and nonfunctional requirements for this transformation and describe the resulting workflow together with relevant systems. To demonstrate the possibilities of this approach, we present the Stream4Flow framework that is based on modern systems for large data processing.

Design Considerations

To successfully transform the batch-based workflow of IP flow analysis into stream-based, it is necessary to meet the same requirements as the original approach and in real time. The data processing speed plays an especially important role, but so do other requirements that must be met, such as a set of available data processing operations, fault tolerance, and system durability. As regards the minimal data processing speed of the approach, it must at least correspond to the average number of flows generated by observation points inside the monitored network. For example, in a medium-sized network of 24,000 active IP addresses, we observed an average of 12,000 flows/s and 110,000 flows/s in the nation-wide research and education network. It can be expected that these numbers will grow in the future and, for that reason, the scalability possibilities of the stream-based processing should also be considered so that it will not be necessary to significantly change the data processing algorithms.

The stream-based approach of IP flow data analysis must enable the IP flow data to be precessed in a similar way as traditional batch-based approaches. This means that it should provide at least the same basic set of data processing operations. Based on the common IP flow analysis algorithms, we identified the following minimal set of operations that should be provided: *filter*, *count*, *aggregation*, *combination*, *sort*, and *Top N*. The stream-based approach should also enable applying these operations to larger units of data; thus, the window functionality is necessary to supply traditional batch-based approaches. In addition to the available operations, stream-based data processing must also ensure that each flow is processed just once to avoid skewed results. Thus, the recoverability and durability options of the data processing system should be considered too.

Workflow Design

Analyzing IP flows in real time was almost impossible previously due to the poor performance of data processing systems. In recent years, however, a change has occurred, and a number of scalable systems for fast batch-based and stream-based processing of large volumes of data were progressively introduced. In the article [7], the authors demonstrated that distributed stream processing frameworks, such as Spark, Samza, and Storm, are able to process at least 500,000 flows/s using 16 or 32 processor cores, which is sufficient for common networks. Thanks to this, it is possible to utilize these frameworks and extend the traditional workflow of IP flow monitoring and analysis. This enables IP flows to be analyzed in real time and provides other analytical methods that are not possible, or difficult to achieve, in common batch-based systems.

A generic interconnection of the typical workflow of stream-based data processing and traditional IP flow monitoring workflow is shown in Fig. 2. To allow such interconnection, it is necessary to enable the collector to transform IP flow records into a suitable data serialization format (DSF). Alternatively, the collector can be omitted from the workflow if the IP flow exporter is able to provide flow records in such a format. The typical format for distributed stream processing frameworks is the JavaScript Object Notation (JSON) format, which enables it to suitably represent any data records. However, the JSON format is not space-efficient and can cause overloading of the network if a lot of IP flows are processed. In the case of a large amount of transmitted data, it is better to utilize a more space-efficient data serialization format, such as binary JSON (BSON) or MessagePack.

The collector's ability to transform IP flow records into a suitable data serialization format is currently not widespread for common IP flow collectors, but several solutions, such as IPFIXCol (https://github.com/CESNET/ipfixcol) and Logstash (https://www.elastic.co/products/logstash), exist, and it can be assumed that new solutions will emerge in the near future. To effectively distribute the transformed IP flows, it is advisable to utilize a messaging system that serves as an input interface for the stream processing framework. There are many such systems, such as ActiveMQ, RabbitMQ, and Apache Kafka (for the full list see [8]); however, to process IP flows, it is necessary to select one providing sufficient throughput. Currently, the most suitable system is Apache Kafka, which offers sufficient message throughput and is compatible with most data stream processing frameworks.

As mentioned earlier, modern distributed stream processing frameworks, such as Samza, Storm, Spark, and Flink, provide sufficient data processing throughput. Thus, in the case of selecting an appropriate system, the decision needs to consider the deployment environment and functional requirements, such as data reliability, scalability, and operators, that suit the considered use well [7]. The intended use must also be considered during the selection of appropriate data storage for analysis results that can be stored in a common relational database, as well as in a next generation database. The storage should support advanced queries over stored data and provide an optimal interface for a web interface so that IP flow analysis results can be visualized to a user. Currently, the most common approach for connecting distributed data processing systems and data storage is the deployment of Elastic Stack (https://www.elastic.co/products), composed of Logstash, Elasticsearch, and Kibana.

As discussed above, the workflow of stream-based IP flow analysis combines several interconnected components. The choice of systems for each of the components should reflect the proposed use, deployment environment, and other mentioned requirements. A list of the most suitable systems for stream-based IP flow analysis is listed in Table 2.

Workflow component	Suggested systems
Collector	IPFIXCol, Logstash
Messaging system	Apache Kafka, NATS, RabbitMQ (The full list available in [8].)
Stream processing framework	Spark Streaming, Flink, Samza, Storm, Trident (all maintained by Apache Software Foundation)
Data storage	Elasticsearch, Druid, OrientDB (next generation databases)
User interface	Kibana, Grafana, Tableau

Table 2. Suggested systems for the workflow of stream-based IP flow analysis.

Workflow Prototype

To demonstrate the possibilities of the presented workflow for real-time analysis of IP flows, the Stream4Flow framework was introduced (https://github.com/CSIRT-MU/Stream4Flow). This framework, among others, interconnects modern systems for fast IP flow data processing, provides simple administration, enables fast application development, and demonstrates the possibilities of stream-based IP flow analysis. The basis of the framework is formed by the IPFIXCol collector, which enables incoming IP flow records to be transformed into the JSON format provided to the Kafka messaging system. The selection of Kafka was based on its scalability and partitioning possibilities, which provide sufficient data throughput. Apache Spark was selected as the data stream processing framework for its quick IP flow data throughput [7], available programming languages (Scala, Java, or Python), and MapReduce programming model [9]. The analysis results are stored in Elastic Stack containing Kibana, which enables browsing and visualizing the results. The Stream4Flow framework also contains the additional web interface (Fig. 3) in order to make administration easier and visualize complex results of the analysis.

Implications for Cyber Security

In the following paragraphs, we discuss two use cases of stream-based IP flow analysis applications. The use cases are chosen to reflect real-world issues of network traffic measurement for cyber security. These issues were identified by others in the literature [2, 10], and confirmed by our own experience, gained during day-to-day operations of the Computer Security Incident Response Team (CSIRT). The team operates a large-scale network of 22,500 hosts with average traffic rates of 6000 flows/s.

For each use case, we introduce the specific problem at hand, and discuss the possible benefits of stream-based IP flow analysis. In addition, we share our experience with the experimental deployment of the Stream4Flow framework prototype and its applications in our network. Last, but not least, some of the possible pitfalls stemming from the deployment are discussed. All of the deployed applications are publicly available within the Stream4Flow repository.

In-Depth Situational Awareness

The goal of cyber situational awareness is to provide in-depth comprehension of events in monitored environments, which is essential for the effective defense of computer networks [2]. A holistic view of the network (a macro view) is typically provided by traditional network monitoring applications. However, to develop a comprehensible overview of the network, more in-depth information (a micro view) is needed, for example, information about individual hosts and their actions. The combination of macro and micro views gives security analysts the ability to observe the overall status, as well as the status of any specific elements in a network, and thus to develop in-depth comprehension of the network.

Stream-based IP flow analysis represents a suit-

To develop a comprehensible overview of the network, more in-depth information (a micro view) is needed, for example, information about individual hosts and their actions. The combination of macro and micro views gives security analysts the ability to observe the overall status, as well as the status of any specific elements in a network, so to develop in-depth comprehension of the network.

In stream-based IP flow analysis, the data cannot be analyzed retrospectively. Since ex-post analysis is as vital as real-time analysis in the context of complex network security, we recommend extending the stream-based workflow with a suitable primary data retention store to make the optional ex-post analysis possible.

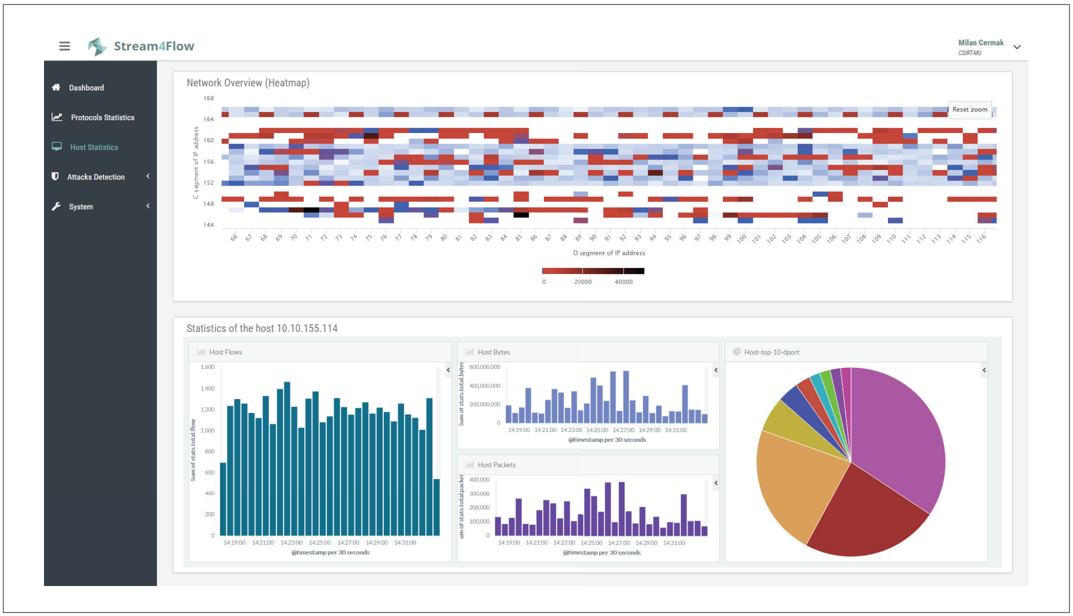


Figure 3. Stream4Flow web interface with network overview and detailed characteristics of a selected host.

able approach for creating a current micro view of the network. It allows us to compute a number of detailed characteristics simultaneously due to the support of scalable and distributed computing. Distributed stream processing systems are able to create a micro view of the network as they provide enough computational power to compute a number of live and detailed statistics. First, these systems are designed with scalability in mind. Thus, the computational resources can be instantly increased by adding additional computational nodes to the system. Second, they provide the means to distribute the IP flow analysis over multiple computational nodes, which allows analyses at a scale that is impossible on a single machine. The distribution is achieved via the MapReduce programming model [9], traditionally used for distributed batch-based processing, but now adapted to also be utilized in a streaming fashion. Third, all the data are processed on the fly in primary memory, which increases throughput as no disk I/O operations are necessary during the analysis. The unique combination of scalability, distributed processing, and on-the-fly data processing makes the creation of a micro view for situational awareness possible in real time.

Our experimental deployment provides a demonstration of an in-depth situational awareness application in a stream-based workflow prototype. We use the MapReduce programming model [9] for creating the micro view by computing host statistics for all devices in our network. A host's IP address is set as a map key for data distribution. The choice of the key and MapReduce model enables us to compute detailed characteristics that represent a host's activity in the network (the number of transferred packets, bytes, communication partners, etc.), and a host's communication profile (e.g., frequently visited IP addresses, web pages, or active hours in a network). The micro view enables us to detect malicious host activities or abrupt changes in host behavior caused by the attacker. Our production instance of the prototype is able to maintain these statistics and detections for each of the 22,500 hosts in real time.

The ability of the stream-based approach to process large volumes of IP flow data has also been shown in [7]. We benchmarked current distributed stream processing systems and their suitability for IP flow data analysis on large volumes. The benchmark measured the throughput of the systems on a set of typical analysis queries. The systems were able to process up to 2 million flows/s on a cluster with 32 vCPU in total. This result was also confirmed by an experimental deployment of the prototype in our network, where it was able to successfully process all the provided IP flows.

It is important to point out here that stream processing changes the nature of the data analysis itself, since the data are processed on the fly, and the analysis must be performed in a certain fashion. In batch-based IP flow data analysis, it is possible to perform a query over historical data, or search back through raw data for additional information after detecting a successful attack. In stream-based IP flow analysis, the data cannot be analyzed retrospectively (ex-post analysis). The start of the stream-based data analysis is marked with the creation of a particular analytical continuous query. Since ex-post analysis is as vital as real-time analysis in the context of complex network security, we recommend extending the stream-based workflow with a suitable primary data retention store to make the optional ex-post analysis possible.

Real-Time Attack Detection

A cyber attack can happen in a fraction of a second and cause serious harm [2]. Distributed DoS attack (DDoS) represents a convenient illustration of such a case. A purpose of the attack is to make a service unavailable and consequently to cause financial loss to a service provider. The cost of unavailability can reach millions of dollars per minute [3]. To reduce the costs, we need to be able to detect the cyber attack as soon as possible. The detection time can be reduced by using real-time attack detection methods instead of traditional batch-based methods with detection delays on the order of minutes. Reducing the detection delay from minutes to seconds enables us to take relevant precautions instantly and considerably reduce financial damages caused by an attack.

The stream-based approach enables immediate attack detection as it is capable of analyzing network traffic in real time. In stream-based analysis, an IP flow is analyzed on the fly as soon as it is received by the system or in micro-batches (e.g., 1 s batches). The detection method then reports an attack immediately as it receives the triggering IP flow. There is no detection delay as in the case of the batch-based approach, where the data is analyzed in several-minute batches. Besides real-time detection, the stream-based approach offers additional benefits for detecting cyber attacks. An analysis of IP flows can be done over particularly short time windows, which can reveal information such as bursts of network traffic, which would be lost in aggregation when using the usual five-minute batches (Fig. 4). Stream-based analysis also naturally implements sliding windows with slides smaller than the window size. This approach allows us to analyze the data and detect network attacks that would be split into two batches in non-stream approaches.

The operational deployment of real-time detection methods in our network highlights the advantages of the stream-based approach. We analyzed a sample of network traffic that contained 29 attacks captured from the daily operations of our CSIRT team. We compared the detection times of both the stream-based approach and the traditional batch-based approach with five-minute batches. Stream-based detection identifies an attack immediately after a triggering flow is observed, whereas the traditional approach executes the detection per batch (i.e., once every 5 min). Thanks to this immediate attack detection, the attacks were reported 181.79 s earlier on average than in the case of the batch-based approach. Due to this fact, we are able to use automated attack mitigation techniques more promptly and eventually mitigate even ongoing attacks.

Our experience shows that the majority of batch-based detection methods can be transformed into a stream-based approach. The above-mentioned reduction of the window in the stream-based approach, however, influences the IP flow analysis in several ways, and it is necessary to adapt detection methods appropriately. First, the computed statistics become more volatile, and detection techniques may report higher errors with the original settings. Therefore, the detection method settings (e.g., thresholds) need to be adapted accordingly to provide correct results. Second, the reduction of the window size raises the issue of ordering the IP flows coherently, since their misplacement to an inaccurate window may bias detection results. The IP flows may get misordered due to link latencies or data loss during their collection from the probes. Traditionally, this is managed by considering the arrival time as a baseline for ordering, but the detection methods must be adapted accordingly to reflect this necessary alleviation.

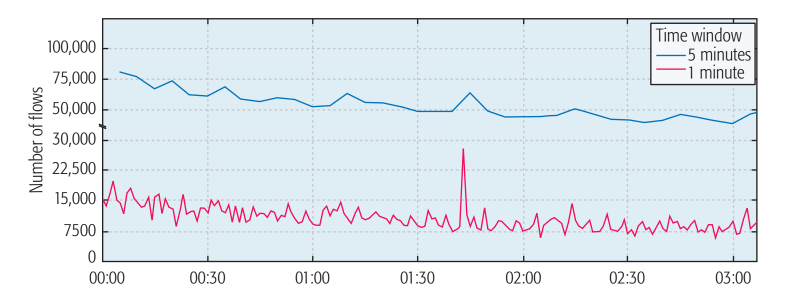


Figure 4. The discovery of a burst of traffic using short analysis windows.

Summary and Outlook

Stream-based IP flow analysis represents a natural complement to current batch-based approaches to cyber security. It aids traditional monitoring with the ability to run analytical queries that are evaluated in real time with high throughput, low latency, and good scalability, all at the same time. This allows security analysts to perform real-time analyses on network data and detect network attacks instantly, and provides them with a deep understanding of the network via in-depth situational awareness. The stream-based analysis workflow benefits from compatibility with current monitoring systems and excels in real-time attack detection, monitoring both network and individual hosts, and providing a context to network security. The presented distributed stream-based framework for IP flow analysis is able to handle streams of large volume of data at high speeds, and keeps up with the latest network monitoring trends.

Since the volume, speed, and diversity of network traffic will continue to increase in the coming years, network monitoring tools should follow this trend [11]. The tools of the future should be able to process traffic at speeds of over 100 Gb/s, gather more information from network traffic (e.g., service-specific or Internet of Things information), and should natively support a wider variety of formats for exporting IP flows (e.g., DSF). In a similar manner to the probes, stream processing systems will have to process more data at higher speeds. This challenge is partially solved by the above-described scalability of current systems. Nevertheless, we expect optimizations for managing resources more efficiently in memory allocation or query response time, for example, via the use of sketches and other probabilistic data structures that are likely to emerge. Moreover, advanced data mining and machine learning methods for intrusion detection are expected to be adapted and natively supported by future data stream systems.

We anticipate increased utilization of network IP flow monitoring for both network and host security. With the emergence of new visionary paradigms, such as the Internet of Things, host-based security will become obsolete as it will be impossible to guarantee the proper setup of host security systems for all connected devices. Network traffic measurement (i.e., IP flow analysis) will become essential for network defense. We believe that stream-based IP flow analysis is a suitable approach to reach the next generation of network security.

We anticipate increased utilization of network IP flow monitoring for both network and host security. With the emergence of new visionary paradigms, such as the Internet of Things, host-based security will become obsolete as it will be impossible to guarantee the proper setup of host security systems for all connected devices.

Acknowledgment

This research was supported by the Technology Agency of the Czech Republic under no. TA04010062, Technology for Processing and Analysis of Network Data in the Big Data Concept.

References

[1] R. Hofstede *et al.*, "Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX," *Commun. Surveys & Tutorials*, vol. 16, no. 4, 2014, pp. 2037–64.

[2] A. Kott *et al.*, *Cyber Defense and Situational Awareness*, 2014.

[3] U. Franke *et al.*, "Availability of Enterprise IT Systems: An Expert-Based Bayesian Framework," *Software Quality J.*, vol. 20, no. 2, 2012, pp. 369–94.

[4] A. Sperotto *et al.*, "An Overview of IP Flow-Based Intrusion Detection," *Commun. Surveys & Tutorials*, vol. 12, no. 3, 2010, pp. 343–56.

[5] B. Babcock *et al.*, "Models and Issues in Data Stream Systems," *Proc. 21st ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems*, 2002.

[6] G. Cugola, and A. Margara, "Processing Flows of Information: From Data Stream to Complex Event Processing," *ACM Computing Surveys*, vol. 44, no. 3, 2012, pp. 1–62.

[7] M. Cermak *et al.*, "A Performance Benchmark for NetFlow Data Analysis on Distributed Stream Processing Systems," *Proc. 2016 IEEE/IFIP Network Operations and Management Symp.*, 2016.

[8] L. Strzalkowski, "Queues," Aug. 2016; http://queues.io

[9] J. Dean, and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Commun. ACM*, vol. 51, no. 1, 2008, pp. 107–13.

[10] E. Cole, *Network Security Bible*, 2011.

[11] Cisco, "The Zettabyte Era: Trends and Analysis," Aug. 2016; http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html

Biographies

Tomas Jirsik (jirsik@ics.muni.cz) obtained his M.S. in applied mathematics from the Faculty of Science, Masaryk University, Czech Republic. He is currently a Ph.D. candidate at the Faculty of Informatics and a member of the Computer Security Incident Response Team of Masaryk University (CSIRT-MU). Besides his main research activities in network data analysis, anomaly detection, and host monitoring, he participates in several projects concerning network security monitoring, cyber education, and big data analytics.

Milan Cermak (cermak@ics.muni.cz) is a security researcher at CSIRT-MU and a Ph.D. candidate in computer systems and Technologies at the Faculty of Informatics, Masaryk University. His main research interests include large-volume network data analysis and advanced network threat detection based on similarity search. He is currently focusing on characterization of anomaly patterns and their utilization for real-time classification of ongoing network traffic.

Daniel Tovarnak (tovarnak@ics.muni.cz) obtained his M.S. in applied informatics from the Faculty of Informatics, Masaryk University. Currently, he is a Ph.D. candidate at the Faculty of Informatics and a researcher at CSIRT-MU. His focus lies in the applications of the complex event processing paradigm in the context of security monitoring. He specializes in security data processing and data normalization in distributed environments.

Pavel Celeda (celeda@ics.muni.cz) is an associate professor affiliated with CSIRT-MU. He received a Ph.D. degree in Informatics from the University of Defence, Brno, Czech Republic. His main research interests include cyber security, flow monitoring, situational awareness, and research and development of network security devices. He has been participating in a number of academic, industrial, and defense projects. He is the head of CSIRT-MU.

SOFTWARE-DEFINED VEHICULAR NETWORKS: ARCHITECTURE, ALGORITHMS, AND APPLICATIONS: PART 1

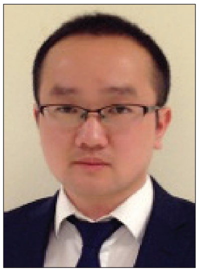
Guangjie Han

Mohsen Guizani

Yuanguo Bi

Tom H. Luan

Kaoru Ota


Haibo Zhou

Wael Guibene

Ammar Rayes

With the ever more rapid development of wireless communications and the explosive usage of mobile electronics, vehicular networks have become an attainable technology to meet the imminent demands for improving traffic safety and efficiency. In addition, there is an increasing demand from traveling users to access the Internet through IP-enabled smart devices, which enables infotainment applications to have rapidly taken on an important role in the past few years. Even though several future architectures have been proposed and investigated, it has been very challenging to coordinate vehicular networks to efficiently facilitate applications with diverse quality of service (QoS) demands. Recently, software-defined networking (SDN) has been emerging as a promising paradigm to control the network in a systematic way. The flexibility and programmability of SDN, which are lacking in today's distributed wireless substrate, not only make it attractive to satisfy the QoS requirements of vehicular multimedia services, but also simplify the resource management in vehicular networks. Consequently, there is a need to conduct research to further investigate the standardization efforts and address challenging issues in the SDN enabled vehicular networks.

In this *IEEE Communications Magazine* Feature Topic (FT), the Guest Editors invited experts from research communities to discuss the architecture, applications, challenging ideas, and standardization efforts on enabling software-defined vehicular networks (SDVNs). After a rigorous review process, 15 papers have been selected to be published in this FT, eight of which are published in Part 1; the rest will be published later, in Part 2 of this FT.

In SDVNs, travelling vehicles may lose connectivity to the central SDN controller, which undermines the benefits provided by SDN. The first article, by S. Correia *et al.*, ''An Architecture for Hierarchical Software-Defined Vehicular Networks,'' describes the design and implementation of a vehicular-based hierarchical software-defined architecture that is dedicated to improving communication performance and efficiency in case of connectivity loss between moving vehicles and the controller. Simulation results demonstrate that the proposed approach performs better than simply falling back to traditional solutions.

To meet rigorous QoS requirements of multimedia services in vehicular networks, 5G mobile communication technologies are indispensable to future SDVN. X. Ge *et al.*, in ''5G Software Defined Vehicular Networks,'' propose a new vehicular network architecture that integrates 5G mobile communication technologies and SDN, in which fog cells are employed to flexibly cover vehicles and prevent frequent handover between vehicles and roadside units (RSUs). Simulation results reveal that there is a minimum transmission delay of 5G SDVNs under different vehicle densities, and the throughput of fog cells in 5G SDVNs can be improved compared to traditional proposals.

A heterogeneous vehicular network with the support of different access technologies is indispensable to provide reliable and ubiquitous mobile access. J. Wan *et al.*, in ''Scalable and Quick-Response Software Defined Vehicular Network Assisted by Mobile Edge Computing,'' propose an SDN-enabled network architecture to guarantee low-latency and high-reliability communications by integrating IEEE 802.11p and 5G radio access technologies in vehicular networks. The practical use case validates that the proposed architecture is able to meet application-specific requirements while maintaining network scalability.

To accommodate vehicular multimedia applications with diverse QoS requirements in various practical scenarios, it is imperative to exploit specific advantages of terrestrial networks, high-altitude communication platforms, and satellite communication systems. The next article, by N. Zhang *et al.*, ''Software Defined Space-Air-Ground Integrated Vehicular Networks: Challenges and Solutions,'' addresses these issues and proposes a software defined space-air-ground integrated network architecture to support various kinds of vehicular services in a seamless, efficient, and cost-effective manner, and the research directions in the proposed integrated vehicular network architecture are identified.

A layer-based top-down approach to systematically cater for security implications is essential to SDVN. A. Akhunzada *et al.*, in ''Toward Secure Software Defined Vehicular Networks: Taxonomy, Requirements, and Open Issues,'' present a top-down approach to address the security vulnerabilities, attacks, and challenges of each layer that is tightly dependent to anticipate secure emerging SDVNs. In addition, the requirements for securing SDVNs are also presented, and open research issues are discussed.

The fast varying network topology and high complexity of network infrastructure impose great challenges on supporting dynamic vehicular communications in 5G heterogeneous networks. X. Duan *et al.*, in ''SDN Enabled 5G-VANET: Adaptive Vehicle Clustering and Beamformed Transmission for Aggregated Traffic,'' present an SDN enabled 5G-VANET by adaptive vehicle clustering and beamformed transmission to accommodate dynamic aggregated traffic. Simulation results demonstrate that the SDN coordinated vehicle clustering and beamformed transmission can efficiently support fast varying traffic.

The distinctive characteristics of SDN are anticipated to facilitate vehicular communications. To explore the domain of SDVN, I. Yaqoob *et al.*, in ''Overcoming the Key Challenges of Establishing Vehicular Communication: Is SDN the Answer?,'' investigate, highlight, and report recent research advances in the SDVN paradigm. The key requirements that need to be met in SDVNs are outlined, and several research challenges are discussed as future research directions. In addition, they conclude that although SDN can improve management capabilities and address many challenges in the traditional VANET, integrating SDN and VANET will bring new challenges.

Integrating existing Wi-Fi networks into VANET is essential to the next generation vehicular networks. T. Q. Duong *et al.*, in ''Software Defined Architecture for VANET: A Testbed Implementation with Wireless Access Management,'' propose an SD-VANET testbed architecture that utilizes already deployed WiFi networks in the Istanbul Technical University campus. The proposed architecture is dedicated to minimize the necessity of changes in current network infrastructure, and avoids any change at the control-data plane interface.

In closing, we would like to thank all the people who have made significant contributions to this FT, including the contributing authors, the anonymous reviewers, and the *IEEE Communications Magazine* publications staff. We believe that the research results presented in this FT will stimulate further research and development ideas in vehicular networks.

Biographies

Guangjie Han [S'01, M'05] is currently a professor with the Department of Information and Communication Systems, Hohai University, China. His current research interests include sensor networks, computer communications, mobile cloud computing, and multimedia communication and security. He has served on the Editorial Boards of 14 international journals, including *IEEE Access* and *Telecommunication Systems*. He has guest edited a number of Special Issues in IEEE journals and magazines. He is a member of ACM.

Mohsen Guizani [S'85, M'89, SM'99, F'09] received his B.S, M.S., and Ph.D. from Syracuse University. He is currently a professor and the ECE Department Chair at the University of Idaho. His research interests include wireless communications/mobile cloud computing, computer networks, security, and smart grid. He is the author of nine books and more 450 publications. He was the Chair of the IEEE Communications Society Wireless Technical Committee. He served as an IEEE Computer Society Distinguished Speaker.

Yuanguo Bi received his Ph.D. degree from Northeastern University, Shenyang, China, in 2010. He joined the School of Computer Science and Engineering, Northeastern University, China, as an associate professor in 2010. His current research interests focus on medium access control, QoS routing, multihop broadcast, mobility management in vehicular networks, as well as SDN enabled vehicular networks.

Tom H. Luan received his Ph.D. degree from the University of Waterloo, Ontario, Canada, in 2012. Since December 2013, he has been a lecturer in mobile and apps with the School of Information Technology, Deakin University, Melbourne, Australia. His research mainly focuses on vehicular networking, wireless content distribution, peer-to peer networking, and mobile cloud computing.

Kaoru Ota received Ph.D. degrees in computer science and engineering from the University of Aizu, Japan, in 2012. She is currently an assistant professor with the Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. She was also a research scientist with A3 Foresight Program (2011–2016) funded by the Japan Society for the Promotion of Sciences, NSFC of China, and NRF of Korea.

Haibo Zhou received his Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University, China, in 2014. From 2014 to 2016, he was a postdoctoral research fellow with the Broadband Communications Research (BBCR) Group, University of Waterloo. He is currently a research associate in the BBCR Group. His current research interests include resource management and protocol design in cognitive radio networks and vehicular networks.

Ammar Rayes [S'85, M'91, SM'15] is a Distinguished Engineer focusing on the technology strategy for Cisco Services. His research interests include IoT, network management NMS/OSS, machine learning, analytics, and security. He has authored three books, over 100 publications in refereed journals and conferences on advances in software and networking related technologies, and over 25 patents. He received B.S. and M.S. degrees from the University of Illinois at Urbana and his D.Sc. degree from Washington University, all in electrical engineering.

Wael Guibene has been a research scientist at Intel Labs since June 2015. He was awarded his Ph.D. from Telecom ParisTech in July 2013. He also holds an M.Eng. and a Master's degree in telecommunications obtained in 2009 and 2010, respectively. He worked at Eurecom as a research engineer from 2010 to November 2013, and then joined Semtech to work on LoRa systems from 2013 to June 2015. His research activities include IoT, 5G, and wireless communications.

An Architecture for Hierarchical Software-Defined Vehicular Networks

Sergio Correia, Azzedine Boukerche, and Rodolfo I. Meneguette

The authors propose a hierarchical SDN-based vehicular architecture that aims to have improved performance in the situation of loss of connection with the central SDN controller. Simulation results show that their proposal outperforms traditional routing protocols in the scenario where there is no coordination from the central SDN controller.

Abstract

With the recent advances in the telecommunications and auto industries, we have witnessed growing interest in ITS, of which VANETs are an essential component. SDN can bring advantages to ITS through its ability to provide flexibility and programmability to networks through a logically centralized controller entity that has a comprehensive view of the network. However, as the SDN paradigm initially had fixed networks in mind, adapting it to work on VANETs requires some changes to address particular characteristics of this kind of scenario, such as the high mobility of its nodes. There has been initial work on bringing SDN concepts to vehicular networks to expand its abilities to provide applications and services through the increased flexibility, but most of these studies do not directly tackle the issue of loss of connectivity with said controller entity. In this article, we propose a hierarchical SDN-based vehicular architecture that aims to have improved performance in the situation of loss of connection with the central SDN controller. Simulation results show that our proposal outperforms traditional routing protocols in the scenario where there is no coordination from the central SDN controller.

Introduction

In recent years, both vehicular ad hoc networks (VANETs) [1] and software-defined networking (SDN) [2] have gained traction and become more widespread. In the former, we have communication networks formed by vehicles exchanging data between themselves, and between devices placed at strategic points of roads and highways, in vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) fashion, respectively. The latter is an emerging network management paradigm that involves separating the control decisions from the forwarding hardware in a move aimed at simplifying the management of such networks and opening them up to innovation.

More recently, there have been efforts [3–10] toward applying SDN concepts to vehicular networks to leverage the flexibility and programmability SDN brings to the table in such dynamic scenarios. The end goal is improving the performance of the VANETs while making them better suited to provide certain services and applications. Furthermore, the ability to reconfigure the network on the fly brought by SDN allows for some interesting use cases in the vehicular scenario. An example of such a use case includes adaptive protocol deployment and multiple-tenant isolation [7]. In the former, the network can select and deploy its routing protocol based on conditions like topology density, for instance. In the latter, it is possible to create slices much like a virtual local area network (VLAN), which can be useful to mitigate a common issue in vehicular networks, the broadcast storm [11]

In software-defined vehicular networks, VANET challenges still apply, such as dealing with the highly mobile nature of the nodes forming the VANET, and at times they create new or augment existing issues that should be addressed appropriately to achieve the expected/desired enhancements by the application of SDN concepts. One such challenge relates to the connectivity between vehicles and the central SDN controller. Given the critical coordination role performed by the SDN controller, having the central controller become unreachable or unresponsive undermines the benefits provided by SDN. Most works in this area do not consider this issue, and instead fall back to the old-fashioned operational method in vehicular networks when this situation arises, reverting to using traditional routing protocols such as Ad Hoc On-Demand Distance Vector Routing (AODV), Destination-Sequenced Distance Vector Routing (DSDV) or Greedy Perimeter Stateless Routing (GPSR).

Considering the limitations in SDN fallback management, we designed and implemented a vehicular-based hierarchical software-defined architecture that aims to present improved performance and efficiency in the situation of connectivity loss with the controller. We do that by defining local SDN domains through clustering and hierarchical access to the main controller through local controllers located at the cluster heads; these SDN domains can retain some of the network intelligence when there is no connectivity to the central controller, and thus can recover from this failure more efficiently. Simulation results showed that our approach performs better than simply falling back to traditional routing protocols.

The main contributions of this work are as follows:

- The design and implementation of an architecture for hierarchical software-defined vehicular networks

The authors are with the University of Ottawa.

Digital Object Identifier:
10.1109/MCOM.2017.1601105

- The development of a new communication protocol that addresses the lack of connection with the central SDN controller
- The development of a new structure to allow creating an SDN controller dynamically
- The evaluation of this communication protocol structure in a real urban mobility scenario

The remainder of this article is divided as follows. We present some of the work related to SDN concepts applied to vehicular networks; we present the proposed SDN-based VANET architecture; we evaluate the proposed architecture in an urban scenario; and finally, we conclude this work.

Related Work

In the literature, some works have applied SDN concepts to vehicular networks and confirmed it to be a promising solution to enhance resource utilization and content distribution, as the logically centralized controller has a comprehensive view of the network topology, being able to make better decisions network-wide.

Ku *et al*. pioneered an SDN-based VANET architecture capable of supporting services such as transmission power adjustment [3]. Their design is composed of an SDN controller, SDN wireless nodes (the real vehicles), and SDN roadside units (RSUs). In their approach, the network operates in different modes, based on the level of control the SDN controller exercises over the rest of the vehicles and RSUs, ranging from total control to no control at all.

Whenever connectivity to the SDN controller is lost, the network falls back to operating as it did in regular VANETs — the vehicles are going to run traditional routing protocols (GPSR, DSDV, OLSR, and AODV) to find routes around the network. This work showed the feasibility of a software-defined VANET, but only tackled the centralized V2V communication, and did so with certain constraints, as the authors used Long Term Evolution (LTE) exclusively for the control plane communication.

He *et al.* proposed another SDN-based architecture for vehicular networks [7], taking into account vehicle-to-cloud and V2I, besides V2V communication. In this work, the authors model the network components of the VANET as SDN switches, and by doing so, their proposal can tackle infrastructure heterogeneity such as different underlying technologies like Wi-Fi, dedicated short-range communications (DSRC) [12], and WiMAX, for instance. Like the work of Ku *et al.*, we also have a fallback to well-known routing protocols when there is a loss of connection with the controller.

Huang *et al.* proposed an SDN-based architecture for vehicular sensor networks that minimizes the issues caused by the loss of connectivity with the SDN controller [10], which is achieved by the development of a service that detects connection states in real time and assists the handoff process resulting from connectivity loss with the controller.

Table 1 presents an overview of some of the recent work in SDN-based vehicular networks. As we can see there, the majority of the works build on top of a centralized architecture, which may suffer from resiliency and also scalability problems in large-scale VANETS. Although the work of Kazmi *et al.* [9] considers multiple controllers, all elements are static; thus, it is unable to deal with every aspect and scenario that can arise in a vehicular network. Also, most of these works do not deal with multiple controllers to keep the communication and the infrastructure alive when no central controller is managing the communication. In SDN, communication with the controller is critical, and it cannot become a single point of failure in the network. In the event the controller becomes unreachable/unresponsive for some reason, it should be possible to recover as quickly as possible to minimize the ill effects on the entire network. Our solution creates smaller SDN domains through clustering of vehicles to mitigate this problem. In these clusters, the cluster head acts as the SDN controller for the domain made up of its members and also takes responsibility for communicating with the primary controller on behalf of its members. Moreover, the proposed solution creates the structure to meet the quality of service (QoS) parameters that applications or services need with the network status to attempt the user's request. The following section describes the proposed solution in more detail.

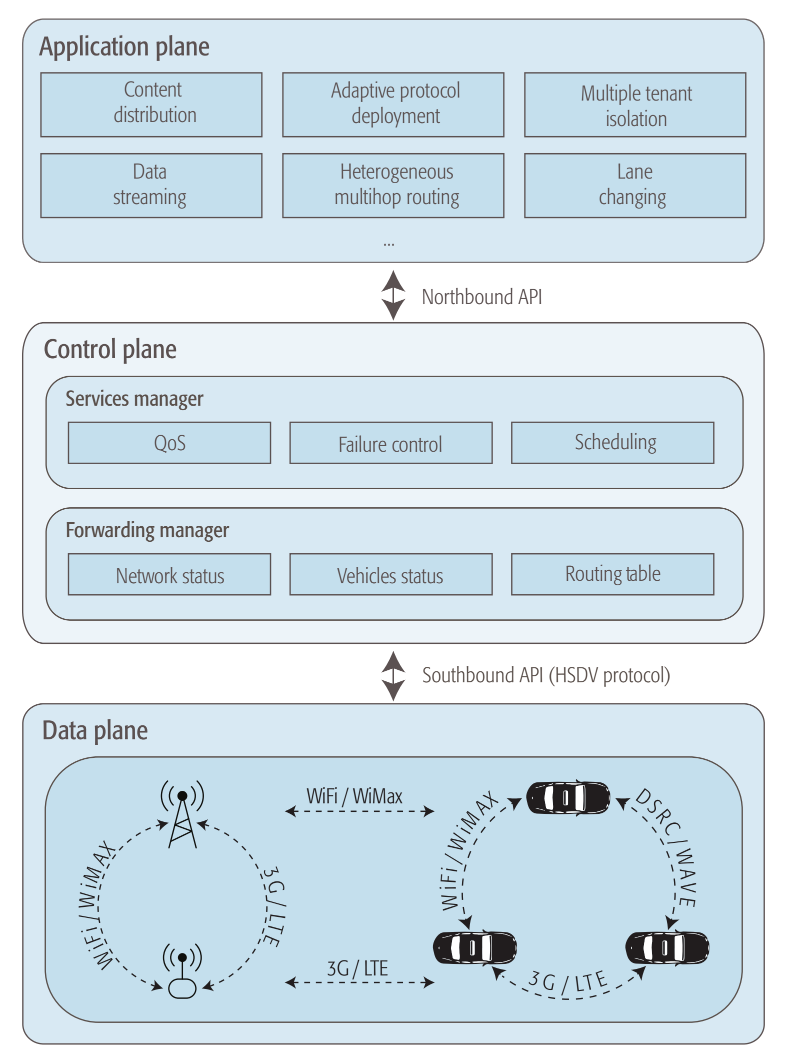


Figure 1. HSDV architecture.

Work	Architecture	Communication	Controller connectivity loss strategy	Multiple controllers	Dynamic controller creation	Resource management	Contribution
Ku *et al.* 2014 [3]	Centralized	V2V				✓	SDN-based VANET architecture
Liu *et al.*l. 2015 [4]	Centralized	V2V, V2I					Geobroadcasting for SDN-based VANETs
Zhu *et al.* 2015 [5]	Centralized	V2V					Routing for SDN-based VANETs
He *et al.* 2015 [6]	Centralized	V2V, V2I				✓	Data scheduling in SDN-based VANETs
He *et al.* 2016 [7]	Centralized	V2V, V2I					SDN-based VANET architecture
Liu *et al.* 2016 [8]	Centralized	V2V, V2I				✓	Data scheduling in SDN-based VANETs
Kazmi *et al.* 2016 [9]	Hierarchical	V2V, V2I		✓		✓	Decentralized SDN-based VANET
Huang *et al.* 2016 [10]	Centralized	V2V	✓			✓	SDN-based sensor VANET
Proposed solution	Hierarchical	V2V, V2I	✓	✓	✓	✓	Decentralized SDN-based VANET

Table 1. Some of the software-defined vehicular network efforts in the literature.

An Architecture for Hierarchical Software-Defined Vehicular Networks

In this work, we propose an SDN-based vehicular architecture called hierarchical software-defined VANET (HSDV), aimed at leveraging the flexibility and programmability brought by SDN into vehicular networks while improving overall system performance in case of connection loss between vehicles and the SDN controller. HSDV uses clustering concepts to create an infrastructure that allows the network to maintain a functional state regardless of central coordination provided by the SDN controller.

HSDV Architecture

Data Plane: responsible for the communication process, providing the data traffic flow management. In HSDV, it manages both V2V and V2I communication with different networking technologies such as WiFi and LTE. The data and control planes communicate through the *southbound API*, which in our case is the HSDV protocol.

Control Plane: responsible for managing data traffic and network intelligence. The control plane in HSDV has two main modules:

- **Services Manager:** manages service requirements and verifies that QoS requirements are met. This module is composed of three submodules: the *QoS* module, responsible for meeting the service requirements with the status of the network; the *failure control* module, responsible for keeping the service working when there is a change of network topology; and the *scheduling* module, responsible for allocating resources and managing the incoming requests.
- **Forwarding Manager:** responsible for routing the data and analyzing the network topology. It is composed of three submodules: the *network status* module, responsible for mapping the network topology and classifying it into either dense or sparse, as well as detecting disconnections from the controller; the *vehicle status* module, responsible for managing the vehicular mobility (e.g., sending messages with updates of the HSDV agents' location/velocity/direction as well as maintaining the vehicle clustering); and the *routing table* module, responsible for dealing with network routing information.

Application Plane: a set of vehicular applications and services that interact with the SDN-based system, either providing services or requesting access to the network. Content distribution and video streaming, among others, are examples of applications that belong to the application plane.

A network that implements HSDV contains two distinct network objects, classified as either an HSDV controller or an HSDV agent.

HSDV Controller: The main component in HSDV, similar to a conventional SDN, is the controller. The controller is responsible for managing the control plane, effectively dictating the behavior of the system. It is logically centralized and, as mentioned before, represents a potential single point of failure should it become unreachable or unresponsive.

HSDV Agent: The agent can communicate with the controller in the same fashion as SDN switches in a traditional SDN. It also contains the intelligence necessary to act as the controller itself in particular situations. HSDV agents represent both RSUs — which in this work include regular RSUs as well as base station cell towers — and vehicles; thus, agents abstract their underlying network communication technology, making their heterogeneity transparent.

Communication Management

The HSDV protocol uses a stable clustering technique based on the work of Rawashdeh and Mahmud [13]. This technique assists in maintaining data transmission when the central SDN controller is not reachable and aims to reduce control communication overhead, thus distributing the network control among the cluster heads.

As mentioned previously, the domain leaders in the HSDV protocol serve as local SDN controllers for their respective local domains. HSDV also has a two-layer hierarchy in which the members have their leaders as their SDN controller, and the local SDN controllers, in turn, communicate with the primary controller to receive instructions such as route updates. When a local SDN controller loses connection to the central controller, it assumes the role of the central controller, managing the network in its domain until the main controller re-establishes a connection with them.

We assume the vehicles share their location via periodic beacon messages, reporting the vehicle's current position, velocity, and direction vectors, alongside the ID of the cluster to which it belongs, its role in said domain, and a timestamp. The role reported by vehicles specifies their current cluster membership status and lists whether they are a member of a cluster or a cluster head.

In order to transmit data among different SDN domains, it is necessary to establish routes between them. HSDV does that using two messages that assist in finding a path between a given source and destination. These are the *RequestPath* and *RouteInfo* messages.

When vehicle *X* needs to transmit data to vehicle *Y*, it first verifies whether it has a route to *Y*. If so, *X* sends packets to *Y* using this route. Otherwise, the vehicle verifies if it is a controller; if not, the vehicle sends a *RequestPath* to its controller — its cluster head. If the vehicle is a local SDN controller, it selects a set of gateways to which to forward the *RequestPath* message to find a route to *Y*. However, the controller may filter this set of gateways to reduce the number of *RequestPath* messages sent, thus reducing the total control overhead. Figure 2 illustrates the route request mechanism in more detail, and Algorithm 1 describes the gateway selection procedure.

Gateway selection is based on the distance between the controller and its neighbors. The method for gateway selection is generally described by the following procedure:

- The controller checks its neighboring tables and finds out which domains are reachable from there, and through which vehicles.
- For each of these clusters, the algorithm will select the gateway closest to the controller that reaches the given target SDN domain.
- Once the controller selects the set of gateways, a filtering to reduce this set may or may not occur; it then forwards the *RequestPath* message to the selected set.

When a vehicle receiving a *RequestPath* message has a route to the target destination, it sends a *RouteInfo* message through the same path on which the *RequestPath* message arrived. Roughly, when a vehicle receives a *RequestPath* message, it starts by checking whether it has a route to the target destination. If so, the vehicle is going to check whether it is not a controller, in which case it will forward this *RequestPath* to its controller with the route update bit set. This specific bit is used to indicate that the *RequestPath* in question is not an actual route request, but an update with information about the target destination of the *RequestPath*; in either case — being or not being a controller — the vehicle is going to send back a *RouteInfo* message. If there is no route for the target destination, the vehicle is also going to check whether it is a controller or not. If it is a local SDN controller, it checks if the message has its route update bit set, in which case it updates its routing table; if it was not an update message, it selects a set of gateways to forward the message. If there is no route and the vehicle receiving the *RequestPath* is not a controller, it is going to either forward this message to its controller or select a single vehicle in a particular target domain to forward it. Figure 3 depicts the complete handling of a received *RequestPath* message.

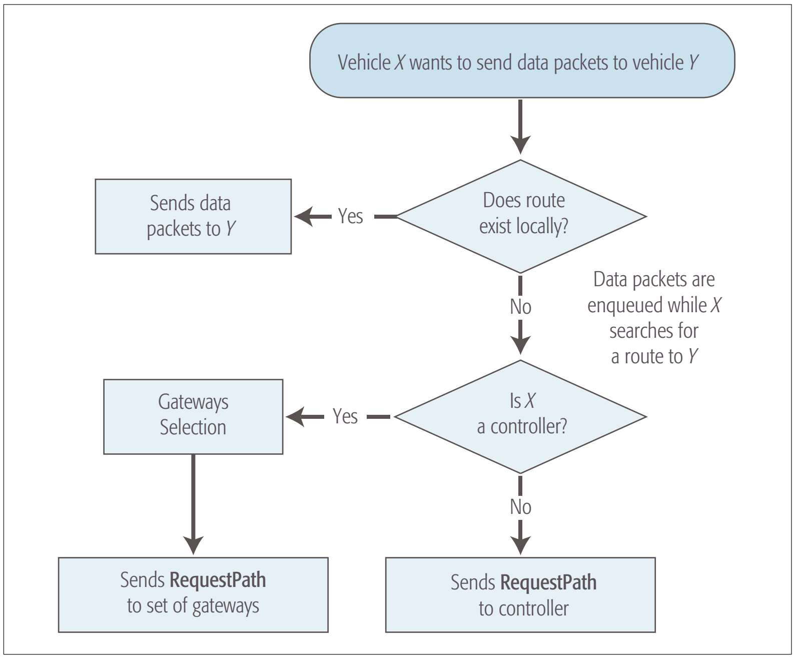

Vehicle X wants to send data packets to vehicle Y
Does route exist locally?
Yes
Sends data packets to Y
No
Data packets are enqueued while X searches for a route to Y
Is X a controller?
Yes
Gateways Selection
No
Sends RequestPath to set of gateways
Sends RequestPath to controller

Figure 2. Vehicle *X* sending data packets to vehicle *Y*.

```
gateways ← ∅
if X is a controller then
    for each different reachable domain D do
        node ← nearest vehicle that reaches D
        adds tuple (D, node) to gateways set
    end for
    if X must filter gateways then
        retains 1 in each N gateways from gateways
    end if
else
    {X was selected to reach domain D}
    gateways ← (D, nearest vehicle that reaches D)
end if
return gateways
```

Algorithm 1. Gateways selection at vehicle X — HSDV-N.

Simulation and Evaluation

In this section, we present the scenario we used to evaluate the proposed solution, which we did through experiments simulated using Network Simulator 3 (ns-3) [14] — an event-based network simulator that provides an implementation of the IEEE 802.11p protocol stack — v. 3.26. Another remarkable tool we employed in this evaluation was the Simulator of Urban Mobility (SUMO) [15] v. 0.27.1, which was used to generate vehicular mobility trace.

Scenario Description

We ran the simulations using a realistic scenario comprising a portion of the city of Ottawa, Canada, in the Sandy Hill neighborhood near the University of Ottawa. The road topology was

HSDV managed to achieve a larger data delivery ratio with lower average delay. This higher delivery resulted in a reasonable number of incoming messages, which allowed for a better throughput. Furthermore, HSDV managed to achieve a reduced control overhead when compared to AODV, DSDV, and GPS.

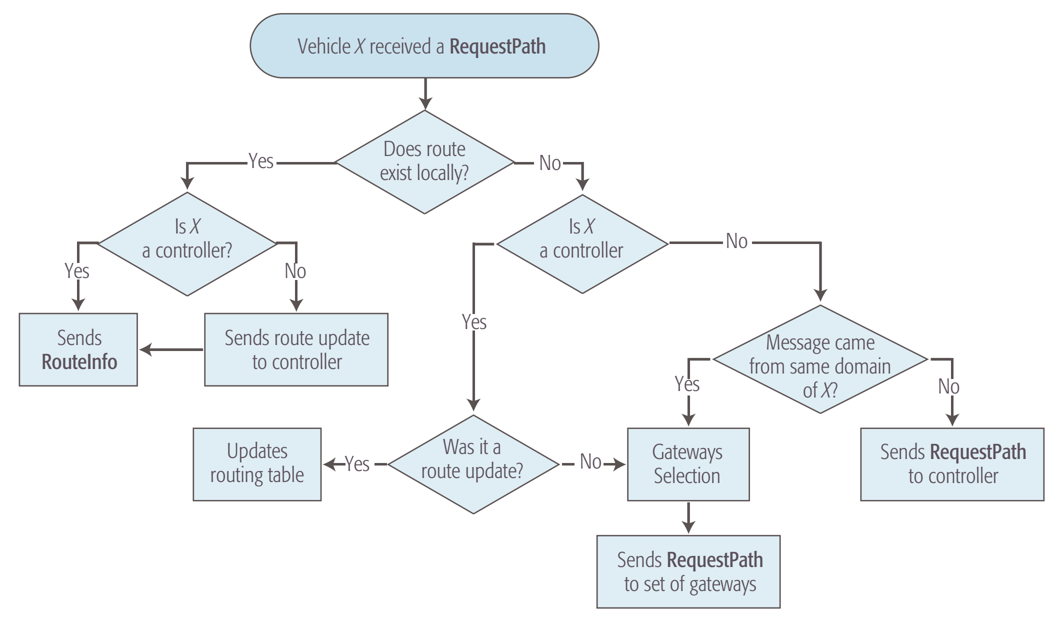

Vehicle X received a RequestPath
Does route exist locally?
Yes
No
Is X a controller?
Yes
No
Sends RouteInfo
Sends route update to controller
Is X a controller
No
Yes
Message came from same domain of X?
Yes
No
Updates routing table
Yes
Was it a route update?
No
Gateways Selection
Sends RequestPath to controller
Sends RequestPath to set of gateways

Figure 3. Vehicle *X* handling a RequestPath message received.

obtained from OpenStreetMap, filtered, formatted, and converted into a SUMO network file, which then generated the vehicular mobility traces used to populate the chosen area.

The experiments are 150 s long, which is enough to evaluate the behavior of HSDV and how it compares to AODV, DSDV, and GPSR when there is no coordination from the central SDN controller. SUMO generates the vehicular trace with the average number of vehicles varying between 25 and 150/km^2, and the cars use IEEE 802.11p as the network protocol. The transmission power is 0.98 mW, the transmission range is 200 m, and we have a beacon rate of 1 Hz.

During the complete simulation, every vehicle is sending data to another, selected randomly. There is a pause time chosen between [0, 5] s, and after that, the car transmits data for an amount of time selected between [10, 40] s. This cycle repeats until the simulation reaches its end. The data traffic is UDP, and each packet consists of 1024 bytes.

In these simulations, we aim to evaluate the performance of HSDV when the primary SDN controller does not have a connection with the vehicles, and hence the network management becomes the responsibility of the different SDN domains. The analysis of the results includes a comparison of HSDV with the traditional routing protocols AODV, DSDV, and GPSR. Furthermore, we use three different configurations of HSDV, changing the threshold among 1, 4, and 6, which in practice tells HSDV to forward the *RequestPath* message to 100, 25, and 16.7 percent of the gateways initially selected, respectively — refer to Algorithm 1.

We take into account the following four metrics in the performance evaluation:

- The *packet delivery ratio*, which considers the ratio between the data packets successfully delivered and the ones sent
- The *throughput*, which gives the rate at which the data packets are passing through the network, considering the amount of data successfully received and the times of both the first packet sent as well the last packet received
- The *average end-to-end delay*, which gives the average of the time the packets successfully transmitted took to arrive at their destinations
- The *routing overhead*, which is the ratio between the control packets sent and the data packets successfully received; this last metric gives an idea of how many control packets are needed to send to receive a data packet

We repeat the experiments 30 times, and then plot the average of the runs and error bars with 95 percent confidence interval.

Results

Figure 4a shows the average end-to-end delay. We observed that DSDV shows better performance when we consider fewer vehicles (25–75 vehicles/km^2), which is because DSDV delivers packets to vehicles that are closer (i.e., it forms short paths), which in turn implies lower delay. However, when we analyze the points with higher density (100–150 vehicles/km^2), HSDV-4 and HSDV-6 have better performance than DSDV due to the proposed technique to reduce the number of control messages in the network (Algorithm 1). We can observe that HSDV-4, HSDV-6, and GPSR have similar performance when we consider a sparse scenario due to the protocols using the nearest path. When we see the environment with more vehicles on the roads (e.g., 100, 125, and 150 vehicles/km^2), HSDV-4 and HSDV-6 achieve a reduction in their delay of about 28 percent compared to AODV and a decrease of about 2 percent in relation to GPSR. This decrease is because HSDV tries to keep only valid routes in its tables; in other words, it works to remove routes that have become invalid due to the changing mobility of vehicles. Although DSDV has better performance than HSDV-4 and HSDV-6 in the more sparse scenarios, the proposed

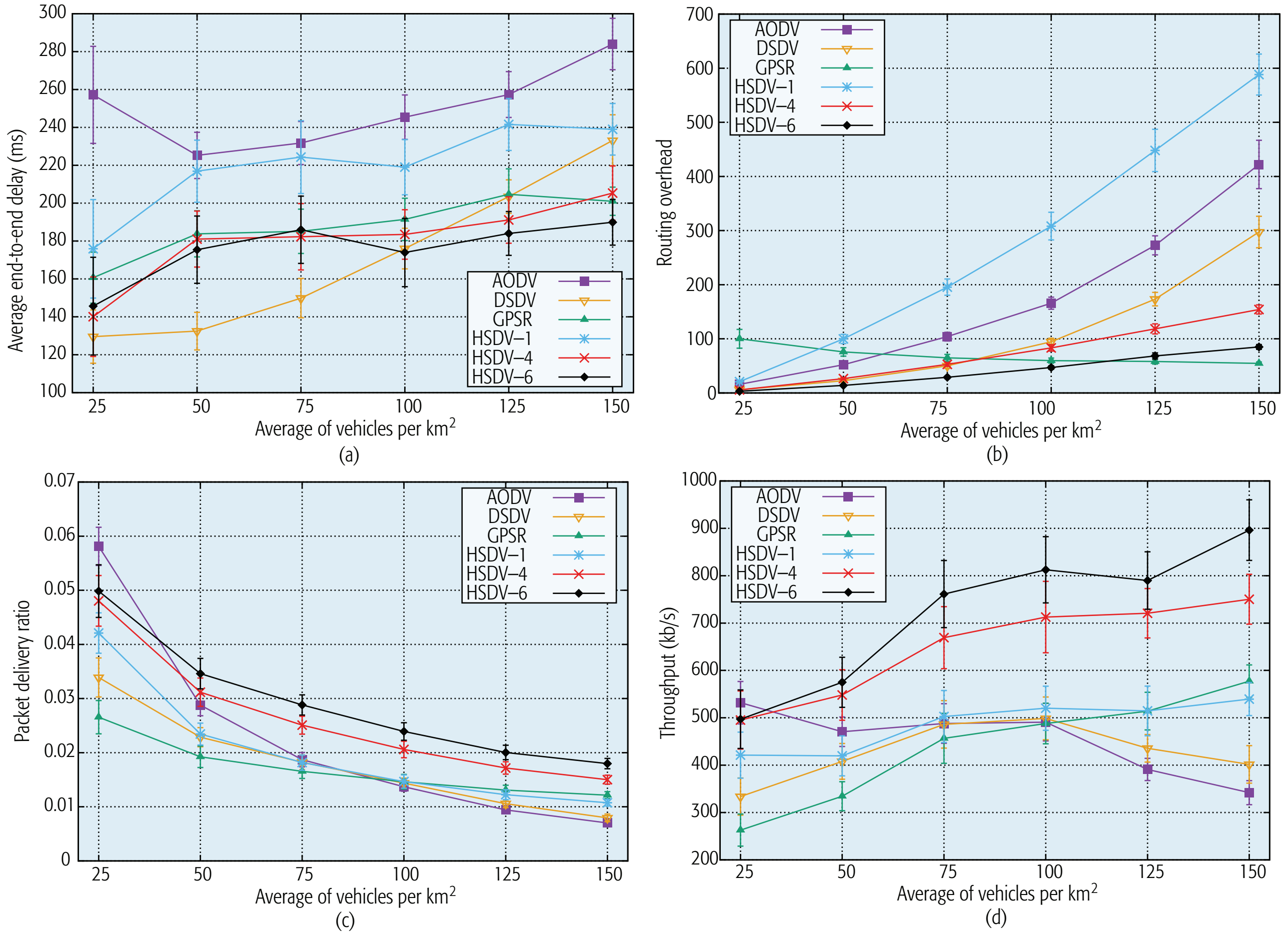


Figure 4. Performance evaluation: a) end-to-end delay; b) routing overhead; c) packet delivery ratio; d) throughput.

solution achieves a higher packet delivery ratio for this same scenario, as can be seen in Fig. 4c.

Figure 4b shows the number of control messages generated on the network. HSDV-4 and HSDV-6 have better performance than the other protocols because they select only a subset of gateways to which to forward the *RequestPath* messagesto — 25 and 16.7 percent, respectively, as mentioned when describing the scenario. On the other hand, HSDV-1 had bad performance because it tries to send *RequestPath* messages to all of its reachable clusters, which can cause a broadcast storm on the network in some circumstances. HSDV-6 has a reduction of about 20 percent when compared to AODV. When we observe the dense scenario, HSDV-4 and HSDV-6 have similar performance to GPSR because GPSR does not need to find the position of the destination; GPSR considers that the vehicle knows the location of the target. However, in a real scenario, the vehicle does not have this information when we take into account only V2V communication. Thus, HSDV has better performance even though it needs to find the destination.

Figure 4c depicts the packet delivery ratio. We can see HSDV-4 and HSDV-6 have better performance compared to other protocols because both HSDV variants employ measures to reduce the number of control messages, decreasing the collisions among the data and allowing for a higher data delivery ratio.

When we observe the point with 125 vehicles/km^2, HSDV has an increase of about 5 percent of the packet delivery ratio when compared to the other protocols. HSDV-1 again has bad performance due to the high number of control messages that end up overloading the network, hence causing a significant number of collisions and degrading its performance. If we consider a sparse scenario (25 vehicles/km^2), AODV has slightly better performance when compared to HSDV-6. However, to achieve this AODV makes use of broadcast in the network to find routes, which results in increased overhead, as seen in Fig. 4b.

Figure 4d shows the throughput. We can see that HSDV-4 and HSDV-6 have better performance when compared to other protocols, which is due to the fact that they deliver more packets. When we analyze the points from 75 to 150 vehicles/km^2, HSDV-6 has an increase of 32 percent when compared to other protocols. HSDV-1, AODV, and GPSR have almost the same performance due to high overhead, and consequently high collision rate. Therefore, due to the logic of gateway selection, forwarding the *RequestPath* messages to only a subset of gateways improves not only the throughput but also other parameters that impact on the network performance.

To summarize, HSDV managed to achieve a larger data delivery ratio with lower average delay. This higher delivery resulted in a reasonable number of incoming messages, which allowed for better throughput. Furthermore, HSDV managed to achieve reduced control overhead when compared to AODV, DSDV, and GPSR, and this was possible because the HSDV protocol offers a mechanism that selects a smaller set of gateways to forward its *RequestPath* messages.

Conclusion

In this article, we propose a solution aimed at leveraging the flexibility and programmability brought by SDN in vehicular networks toward improving overall system performance in scenarios where there is a connection loss between the vehicles and the primary SDN controller. The proposed solution uses a clustering technique to create independent local SDN domains.

Simulation results show that the proposed solution performs better than traditional routing protocols (AODV, DSDV, and GPSR). HSDV has higher delivery rates with lower delays and overhead, allowing for higher network throughput. In future works, we intend to consider information about the vehicular network to assist the main controller and local controllers in resource allocation and management. Integrating knowledge on network status and network load, content preferences is predicted to improve service delivery, thus improving user experience and QoS. Moreover, we plan to evaluate the proposed algorithm in scenarios where the primary controller is present but only partially assists in network coordination. Lastly, we plan to evaluate the performance of HSDV in scenarios with specific peculiarities, such as partial RSU deployment and highly correlated mobility patterns.

We plan to evaluate the proposed algorithm in scenarios where the primary controller is present but only partially assists in network coordination. Lastly, we plan to evaluate the performance of HSDV in scenarios with specific peculiarities, such as partial RSU deployment and highly correlated mobility patterns.

Acknowledgment

This work was partially supported by the Canada Research Chairs Program, NSERC DIVA Strategic Network, and NSERC CREATE TRANSIT Program as well as the Brazilian Council for Scientific and Technological Development (CNPq), under Grant no. 208397/2014-0/GDE, and São Paulo Research Foundation (FAPESP), under Grant no. 2015/18898-9.

References

[1] H. Hartenstein and L. Laberteaux, "A Tutorial Survey on Vehicular Ad Hoc Networks," *IEEE Commun. Mag.*, vol. 46, no. 6, 2008, pp. 164–71.

[2] B. A. A. Nunes *et al.*, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Commun. Surveys & Tutorials*, vol. 16, no. 3, 2014, pp. 1617–34.

[3] I. Ku *et al.*, "Towards Software-Defined VANET: Architecture and Services," *2014 13th Annual Mediterranean Ad Hoc Networking Wksp.*, 2014, pp. 103–10.

[4] Y.-C. Liu, C. Chen, and S. Chakraborty, "A Software Defined Network Architecture for GeoBroadcast in VANETs," *2015 IEEE ICC*, 2015, pp. 6559–64.

[5] M. Zhu *et al.*, "SDN-Based Routing for Efficient Message Propagation in VANET," *WASA 2015: 10th Int'l. Conf. Wireless Algorithms, Systems, and Applications*, Springer, 2015, pp. 788–97.

[6] Z. He, D. Zhang, and J. Liang, "Cost-Efficient Heterogeneous Data Transmission in Software Defined Vehicular Networks," *2015 IEEE 17th Int'l. Conf. High Performance Computing and Commun.*, 2015, pp. 666–71.

[7] Z. He, J. Cao, and X. Liu, "SDVN: Enabling Rapid Network Innovation for Heterogeneous Vehicular Communication," *IEEE Network*, vol. 30, no. 4, July 2016, pp. 10–15.

[8] K. Liu *et al.*, "Cooperative Data Scheduling in Hybrid Vehicular Ad Hoc Networks: VANET as a Software-Defined Network," *IEEE/ACM Trans. Net.*, vol. 24, no. 3, 2016, pp. 1759–73.

[9] A. Kazmi, M. A. Khan, and M. U. Akram, "DeVANET: Decentralized Software-Defined VANET Architecture," *2016 IEEE Int'l. Conf. Cloud Engineering Wksp.*, 2016, pp. 42–47.

[10] T. Huang *et al.*, "Building SDN-Based Agricultural Vehicular Sensor Networks Based on Extended Open vSwitch," *Sensors*, vol. 16, no. 1, 2016, p. 108.

[11] N. Wisitpongphan *et al.*, "Broadcast Storm Mitigation Techniques in Vehicular Ad Hoc Networks," *IEEE Wireless Commun.*, vol. 14, no. 6, 2007, pp. 84–94.

[12] Y. L. Morgan, "Notes on DSRC & WAVE Standards Suite: Its Architecture, Design, and Characteristics," *IEEE Commun. Surveys & Tutorials*, vol. 12, no. 4, 2010, pp. 504–18.

[13] Z. Y. Rawashdeh and S. M. Mahmud, "A Novel Algorithm to Form Stable Clusters in Vehicular Ad Hoc Networks on Highways," *EURASIP J. Wireless Commun. and Net.*, vol. 2012, no. 1, 2012, pp. 1–13.

[14] G. F. Riley and T. R. Henderson, "The ns-3 Network Simulator," *Modeling and Tools for Network Simulation*, Springer, 2010, pp. 15–34.

[15] M. Behrisch *et al.*, "SUMO — Simulation of Urban MObility: An Overview," *Int'l. Conf. Advances in System Simulation*, 2011, pp. 63–68.

Biographies

Sergio Correia (sergio.correia@uottawa.ca) is a Ph.D. candidate in the School of Electrical Engineering and Computer Science at the University of Ottawa, Canada, working under the supervision of Prof. Azzedine Boukerche. He received his Bachelor's and Master's degrees in computer science from Universidade Estadual do Ceará, Brazil, in 2010 and 2011, respectively. His current research interests lie in the areas of mobile and vehicular ad hoc networks and software-defined networking.

Azzedine Boukerche (boukerch@site.uottawa.ca) is a Distinguished Professor and holds a Senior Canada Research Chair Tier 1 position at the University of Ottawa. He was a recipient of the IEEE Canada K. Gotlieb Computer Gold Medal Award, IEEE-CS Golden Core Award, and Ontario Distinguished Research Award. His current research interests include context-interpretation-based systems for emergency class applications, sensor networks, vehicular networks, wireless multimedia, distributed and mobile computing, and mobile cloud networking.

Rodolfo I. Meneguette (rmenegue@site.uottawa.ca) is an assistant professor at the Federal Technology Institute. He received his Master's degree in 2009 from Universidade Federal de São Carlos and his doctorate from the University of Campinas (Unicamp), Brazil, in 2013. Currently, he is a postdoctoral researcher at the PARADISE Research Laboratory, University of Ottawa. His research interests are in the areas of vehicular networks, resources management, mobility flows, and vehicular clouds.

5G Software Defined Vehicular Networks

Xiaohu Ge, Zipeng Li, and Shikuan Li

ABSTRACT

With the emergence of 5G mobile communication systems and software defined networks, not only could the performance of vehicular networks be improved, but also new applications of vehicular networks are required by future vehicles (e.g., pilotless vehicles). To meet requirements of intelligent transportation systems, a new vehicular network architecture integrated with 5G mobile communication technologies and software defined networking is proposed in this article. Moreover, fog cells have been proposed to flexibly cover vehicles and avoid frequent handover between vehicles and roadside units. Based on the proposed 5G software defined vehicular networks, the transmission delay and throughput are analyzed and compared. Simulation results indicate that there is a minimum transmission delay of 5G software defined vehicular networks considering different vehicle densities. Moreover, the throughput of fog cells in 5G software defined vehicular networks is better than the throughput of traditional transportation management systems.

INTRODUCTION

Nowadays the fifth generation (5G) mobile communication systems are developed by industrial and academic researchers. With the development of millimeter-wave and massive multiple-input multiple-output (MIMO) technologies, the spectrum efficiency and energy efficiency are obviously improved for 5G wireless communications [1, 2]. With the emergence of pilotless vehicles, some rigorous requirements (e.g., the transmission delay needs to be less than 1 ms) are needed for intelligent transportation systems (ITSs) and vehicular networks [3]. To meet these rigorous requirements, 5G mobile communication technologies, cloud computing, and software defined networking (SDN) are expected to be integrated into future vehicular networks. Therefore, it is necessary to design a new network architecture for 5G vehicular networks.

Some basic issues have been investigated for vehicular networks [4–7]. Considering the drawbacks of IEEE 802.11p networks, such as poor scalability, low capacity, and intermittent connectivity, the Long Term Evolution (LTE) mobile communication technologies were proposed to support vehicular applications [4]. Moreover, the open issues of LTE vehicular networks were discussed to promote potential solutions for future vehicular networks. In [5] the basic characteristics of vehicular networks were introduced. An overview of applications and associated requirements was presented and challenges were discussed. Also, the past major ITS programs and projects in United States, Japan, and Europe were analyzed and compared. An analytical model supporting multihop relay of infrastructure-based vehicular networks was proposed to analyze uplink and downlink connectivity probabilities [6]. Simulation and experiment results revealed that there is a trade-off between the proposed performance metrics and system parameters, such as base station (BS) and vehicle densities, radio coverage, and the maximum number of hops in a path. When LTE communication technologies have been integrated into vehicular networks, the interference has cut down the performance of LTE vehicular networks [7]. To overcome this issue, the millimeter-wave transmission technology was proposed to connect users inside vehicles. On the other hand, SDN was proposed as an effective network technology, capable of supporting the dynamic nature of vehicular network functions and intelligent applications while lowering operation costs through simplified hardware, software, and management [8]. Consequently, some initial studies have been carried out to integrate SDN technology into vehicular networks [9, 10]. Utilizing SDN, an adaptive edge computing solution based on regressive admission control and fuzzy weighted queueing was proposed to monitor and react to network quality of service (QoS) changes within vehicular network scenarios [9]. Based on SDN, a cooperative data scheduling algorithm integrated at roadside units (RSUs) was developed to enhance the data dissemination performance by exploiting the synergy between infrastructure-to-vehicle (I2V) and vehicle-to-vehicle (V2V) communications [10]. However, the SDN technology in [10] is limited in RSUs. When a lot of vehicles are connected to an RSU, the frequent handover problem reduces the performance of SDN [11].

To meet the high performance requirements, such as low transmission delay and high throughput, a new architecture of 5G software defined vehicular networking is proposed in this article. The main contributions of the proposed 5G software defined vehicular network are as follows:

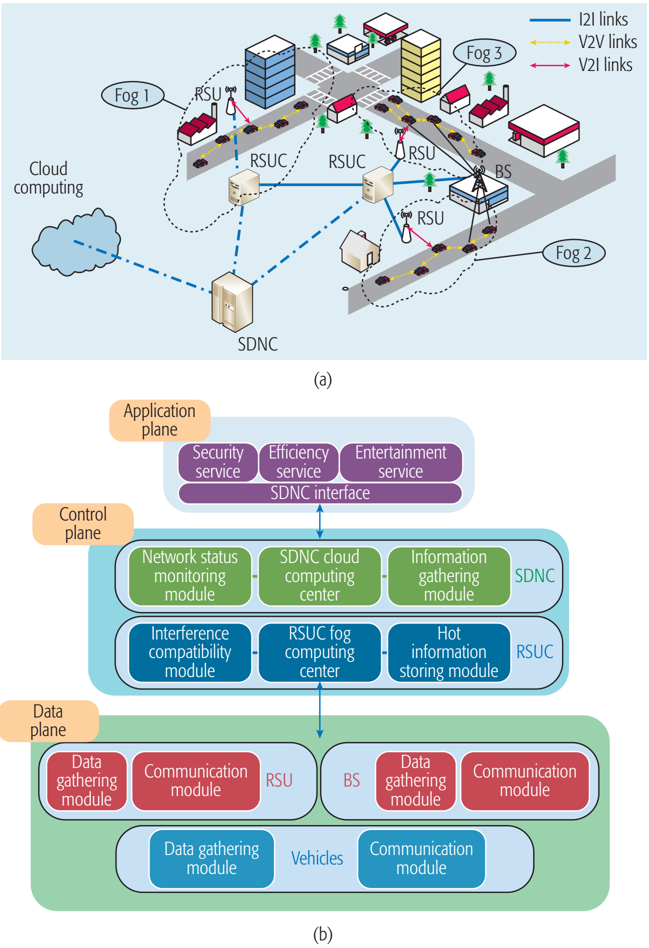
1. Based on the basic functions and requirements of vehicular networks, an architecture of a 5G software defined vehicular network integrated with SDN, cloud computing, and fog computing technologies is proposed to form three logistical planes in network architecture (i.e., the application plane, the control plane, and the data plane). Based on three logistical planes of network architecture, the control and data functions of 5G soft-

With the emergence of 5G mobile communication systems and software defined networks, not only could the performance of vehicular networks be improved, but also new applications of vehicular networks are required by future vehicles (e.g., pilotless vehicles). To meet requirements from intelligent transportation systems, a new vehicular network architecture integrated with 5G mobile communication technologies and software defined networking is proposed by the authors.

The authors would like to acknowledge the support from the NSFC Major International Joint Research Project (Grant No. 61210002) and the Fundamental Research Funds for the Central Universities under grant 2015XJGH011. This research is partially supported by the EU FP7-PEOPLE-IRSES, project acronym WiNDOW (grant no. 318992) and project acronym CROWN (grant no. 610524), and the China International Joint Research Center of Green Communications and Networking (No. 2015B01008).

The authors are with Huazhong University of Science and Technology.

Digital Object Identifier: 10.1109/MCOM.2017.1601144



(a)

(b)

Figure 1. a) Topology structure of 5G software defined vehicular networks; b) logical structure of 5G software defined vehicular networks.

ware defined vehicular networks are separated to improve the flexibility and scalability of vehicular networks.

2. The fog cell structure is proposed and performed at the edge of 5G software defined vehicular networks. Based on the fog cell structure, frequent handover between the RSU and vehicles is avoided, and an adaptive bandwidth allocation scheme is adopted for vehicles in fog cells.

3. The transmission delay and throughput of 5G software defined vehicular networks are analyzed. Simulation results indicate that there is a minimum transmission delay of 5G software defined vehicular networks considering different vehicle densities. Moreover, the throughput of fog cells in 5G software defined vehicular networks is better than the throughput of traditional transportation management systems.

In this article we propose a new architecture of 5G software defined vehicular networks adapting the cloud computing and fog computing technologies. Moreover, the control plane and data plane are separated by the SDN technology in 5G software defined vehicular networks. To avoid frequent handover between the RSU and vehicles, the fog cell is structured, and multihop relay is adopted for vehicular communications in a fog cell. Furthermore, the transmission delay and the throughput of fog cells are simulated for 5G software defined vehicular networks. Finally, the challenges of vehicular networks are discussed, and conclusions are drawn.

5G Software Defined Vehicular Networks

Topology Structure of 5G Software Defined Vehicular Networks

The cloud computing and fog computing technologies are emerging for applications of 5G vehicular networks. Moreover, SDN is becoming a flexible approach to connect wireless access networks and cloud computing centers for 5G vehicular networks. Based on cloud computing and fog computing technologies, a 5G software defined vehicular network is proposed in this article. The topology structure of 5G software defined vehicular networks is illustrated in Fig. 1a. 5G software defined vehicular networks are composed of cloud computing centers, SDN controllers (SDNCs), RSU centers (RSUCs), RSUs, BSs, fog computing clusters, vehicles, and users. Moreover, 5G software defined vehicular networks include infrastructure-to-infrastructure (I2I) links, vehicle-to-infrastructure (V2I) links, and vehicle-to-vehicle (V2V) links. Based on 5G software defined vehicular networks, the information is shared among vehicles and users under the control of the fog computing clusters. To support prompt responses from vehicles and users, fog computing clusters are configured at the edge of 5G software defined vehicular networks. The network structure of fog computing clusters is a distributed network. Most data in the edge of 5G software defined vehicular networks is saved and processed by fog computing clusters, which include the RSUC, RSUs, BSs, vehicles, and users. The SDNCs collect and forward the state information of fog computing clusters into the cloud computing centers. Moreover, the control information is sent to fog computing clusters by SDNCs. The core of 5G software defined vehicular networks, composed of SDNCs and cloud computing centers, is adopted by a centered network structure that focuses on data forwarding and resource allocation. The detailed logical structure of 5G software defined vehicular networks is described in Fig. 1b.

Logical Structure of 5G Software Defined Vehicular Networks

In Fig. 1b, the logical structure of 5G software defined vehicular networks is composed of the data plane, the control plane, and the application plane.

The data plane includes vehicles, BSs, and RSUs. Functions of the data plane are focused on data collection, quantization, and then forwarding data into the control plane [12]. In detail, the vehicle can be configured with the following function modules.

Information collection module of vehicles: The information collection module is made up of different types of sensors in a vehicle. Utilizing sensors in the vehicle, the information on the vehicle (e.g., the speed, direction, and type of vehicle) and the environment (e.g., the number of adjacent vehicles, the users in the vehicle, and the road under the vehicle) collected for 5G software defined vehicular networks.

Position information module of vehicles: The position information of vehicles includes independent position information and dependent position information. In general, the independent position information of vehicles is obtained by the GPS, which provides the detailed location of vehicles in the longitude and latitude of the Earth. The dependent position information of vehicles is obtained by sensors of vehicles, which provide the distance between adjacent vehicles. Compared to the independent position information of vehicles, the dependent position information of vehicles can provide high location precision for 5G software defined vehicular networks.

Communications module of vehicles: The communication module includes V2I and V2V communication modules. The V2I communication module provides wireless communication between vehicles and the infrastructure along the road. The V2V communication module provides wireless communication among adjacent vehicles.

BSs can provide wireless communication for vehicles and RSUCs. In 5G software defined vehicular networks, BSs transmit wireless signals by traditional LTE frequency and provide broad coverage for vehicles. In general, vehicles first access with RSUs but then access with BSs when RSUs cannot provide enough resource for wireless access in 5G software defined vehicular networks.

In 5G software defined vehicular networks, RSUs can be configured with the following function modules.

Information collection module of RSUs: Composed of different sensors (e.g., cameras and speed measurement sensors). The information collection module of RSUs can provide the speed of vehicles, traffic status, road status, and so on.

Communication module of RSUs: Including two types of links: one is the link between RSUs and the RSUC, and the other is the link between RSUs and vehicles. The links between RSUs and the RSUC are performed by fronthaul links in 5G software defined vehicular networks.

The control plane includes RSUCs and SDNC. The RSUC is the control center of a fog cell. Considering the quick mobility of vehicles and the massive wireless traffic between the RSU and vehicles, frequent handover should be avoided for wireless communications between the RSU and vehicles. To solve this issue, the fog cell is proposed for 5G software defined vehicular networks. A fog cell is composed of vehicles and an RSU. Millimeter-wave links are adopted for wireless relay communications among vehicles, and the total bandwidth of millimeter-wave is shared by all vehicles in a fog cell. Since all vehicles move in an orderly fashion on an urban road, the total vehicle group can be assumed to be an overall communication unit within millimeter-wave links in a fog cell. When one of the vehicles in a vehicle group connects with the RSU, the whole vehicle group in the fog cell could be connected with the RSU. In this case, frequent handover can be avoided for vehicles and the RSU in a fog cell. Hence, the RSUC is configured to allocate resources and improve the transmission efficiency in a fog cell. The SDNC is the total control center for 5G software defined vehicular networks and allocates resources among fog cells. Therefore, the control plane takes charge of drawing the global information map based on the data information forwarded from the data plane and then generating the control information based on rules and strategies from the application plane. To support the above functions of the control plane, RSUCs and the SDNC are configured with the following function modules:

Information collection modules of RSUC and SDNC: Drawing the global information map based on the data information from the data plane.

Networking status module: Monitoring the link status of 5G software defined vehicular networks [13].

Computing module: Deriving the control results based on the global information map and the link status of 5G software defined vehicular networks. In general, computing modules are deployed at the cloud computing center and fog computing centers [14].

Hot caching module: saving the popular data context at RSUCs to decrease the transmission delay for vehicle applications.

The application plane directly faces different application requirements from users and vehicles. Based on application requirements from users and vehicles, rules and strategies of 5G software defined vehicular networks are generated by the application plane and forwarded to the control plane. In general, the application plane includes the security service module, the service efficiency module, and the entertainment service module.

Based on the logical structure of 5G software defined vehicular networks in Fig. 1b, the data plane takes charge of collecting data, the control plane takes charge of deriving control instructions, and the application plane takes charge of generating rules and strategies.

Transmission Delay and Throughput of 5G Software Defined Vehicular Networks

Without loss of generality, the transmission delay and throughput analysis are investigated in a fog cell of 5G software defined vehicular networks. A typical fog cell is composed of an RSU and a number of vehicles in Fig. 2. To avoid frequent handover between the RSU and vehicles in the fog cell, a vehicle (i.e., the gateway vehicle) is selected to connect with the RSU, and then other vehicles are connected with the gateway vehicle by a multihop relay method. When a gateway vehicle is located in the coverage region of the RSU, the gateway vehicle directly communicates with the RSU. When other vehicles are located in the fog cell, even if these vehicles are not directly covered by the RSU in the fog cell, they will build a multihop relay route to connect with the gateway vehicle, and then the gateway vehicle will forward those requests/data to the RSU in the fog cell. When the gateway vehicle departs from the fog cell, a vehicle in the fog cell is handed off to serve as the gateway vehicle [15]. In this way, all vehicles in the fog cell can maintain wireless communications with the RSU while moving along the road. Since the fog cell is the basic composition of the proposed 5G software defined vehicular

The SDNC is the total control center for 5G software defined vehicular networks and allocates resources among fog cells. Therefore, the control plane takes charge of drawing the global information map based on the data information forwarded from the data plane and then generating the control information based on rules and strategies from the application plane.

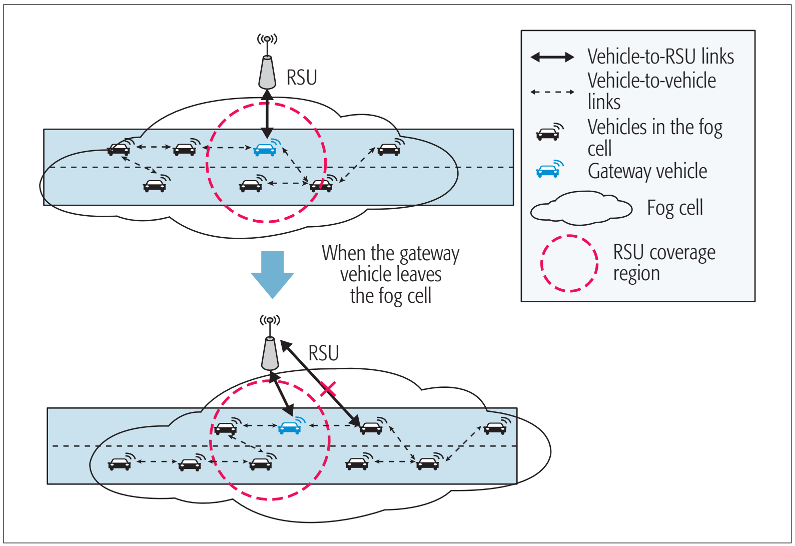


Figure 2. Vehicle communications in a typical fog cell.

networks, the transmission delay and throughput of the vehicle in a fog cell is investigated in the following sections.

Transmission Delay of 5G Software Defined Vehicular Networks

The transmission delay is one of the core metrics for 5G software defined vehicular networks. In this article, the transmission delay of the vehicle in a fog cell is analyzed for 5G software defined vehicular networks.

In Fig. 3, an RSU is located at a fog cell to serve all vehicles driving on a road of length of L. Without loss of generality, a vehicle inside a red dashed circle (i.e., VE_a) is selected to analyze the transmission delay in a fog cell of 5G software defined vehicular networks. The distance between the RSU and the vehicle VE_a is denoted as L_a. Based on the vehicle communication scheme in Fig. 2, the data packet generated from VE_a is transmitted to the RSU by a multihop vehicle relay method.

Assume that there are k hops between the RSU and the vehicle VE_a. For a data packet, the transmission delay in a fog cell of 5G software defined vehicular networks is expressed as $T = kT_{hop} + (k - 1)T_{retran}$, where T_{hop} is the average transmission delay in one hop of vehicle communications, and T_{retran} is the retransmission delay, which is the relay processing time at the relay vehicles. In a hop of vehicle communications, the wireless transmission is time slotted, and one data packet is transmitted in each time slot t_{slot}. Assume that the success transmission probability of vehicle relay communications is P_{hop}. As a consequence, the average transmission delay in one hop of vehicle communications is calculated by $T_{hop} = t_{slot}/P_{hop}$.

In this article millimeter-wave transmission is adopted for vehicle relay communications. Without loss of generality, the 60 GHz frequency spectrum is assumed to be used for vehicle relay communications. Since the wireless signals of vehicle relay communications are usually transmitted in line of sight (LOS) scenarios, the interference is ignored for the vehicle relay communications in this article. When the signal-to-noise ratio (SNR) threshold at the receiver is assumed to be θ (i.e., the data packet can be successfully received only if the SNR of receive signal is larger than the threshold θ), the success transmission probability P_{hop} is calculated by $P_{hop} = P(PL \le P_{tx}[\text{dB}] - \theta[\text{dB}] - N_0W_{mmWave}[\text{dB}])$, where $PL[\text{dB}](\delta) = 69.6 + 20.9\log(\delta) + \xi$, $\xi \sim (0, \sigma^2)$ is the path loss fading over millimeter-wave wireless channels, δ is the wireless transmission distance between the transmitter and receiver, P_{tx} is the transmission power of vehicles, N_0 is the noise power spectrum density, and W_{mmWave} is the bandwidth of millimeter-wave links.

To analyze the transmission delay in a fog cell of 5G software defined vehicular networks, the default parameters are configured as follows: the noise power spectrum density is $N_0 = -174$ dBm/Hz, the bandwidth of millimeter-wave links is $W_{mmWave} = 2$ GHz, the retransmission delay is $T_{retran} = 5$ μs, and one time slot is $t_{slot} = 5$ μs. Moreover, the transmission distance of millimeter-wave communications is limited to 50 m.

Figure 4 illustrates the transmission delay in a fog cell of 5G software defined vehicular networks with respect to the vehicle density considering different transmission distances L_a. When the vehicle density is fixed, the transmission delay increases with the increase of the transmission distances L_a. When the transmission distance L_a is fixed, the transmission delay first decreases with the increase of the vehicle density. However, numerical results indicate that there are turning points for vehicle densities (the turning points are 0.08, 0.09, and 0.105 for L_a = 300, 400, and 500, respectively). When the vehicle density is larger than or equal to the turning point, the transmission delay increases with the increase of vehicle density.

The numerical results in Fig. 4 show that there is a minimum value for the transmission delay in the fog cell of 5G software defined vehicular networks. The minimum transmission delay is 0.32, 0.46, and 0.63, corresponding to the transmission distance of 300, 400, and 500 m, respectively. When the vehicle density is low, the distance among adjacent vehicles is far, and thus the success transmission probability of millimeter-wave links is low. In this case, increasing vehicle density will decrease the distance among adjacent

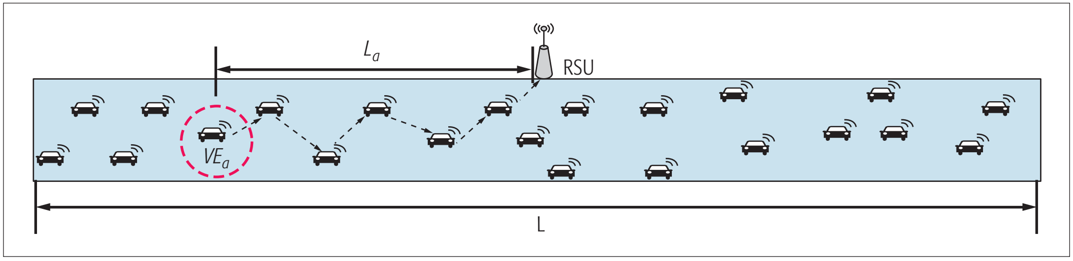


Figure 3. Transmission delay in a fog cell of 5G software defined vehicular networks..

vehicles and then increase the success transmission probability of millimeter-wave links. Hence, the transmission delay first decreases with the increase of vehicle density. When the vehicle density is larger than a threshold, the distance among adjacent vehicles is closed, and the successful transmission probability of millimeter-wave links approaches a stationary value. In this case, the transmission delay is mainly dependent on the retransmission delay in each hop of vehicle communication. In this case, increasing vehicle density will increase the number of relay hops, and then the total retransmission delay is increased. Therefore, the transmission delay increases with the increase of vehicle density.

Throughput of 5G Software Defined Vehicular Networks

In a traditional bandwidth allocation scheme, all bandwidths are averagely allocated to every vehicle in a fog cell. However, every vehicle needs different bandwidth in practical applications. Based on the control function of 5G software defined vehicular networks, which is realized at the RSUC, an adaptive bandwidth allocation scheme is proposed to optimize the throughput of fog cells in this article.

Without loss of generality, the available bandwidth in a fog cell is assumed to be B, and the maximum throughput of this fog cell is C. The average bandwidth requirement of one vehicle is B_{ave}, and the throughput of this vehicle is configured as C_{ave}. The total number of vehicles in a fog cell is assumed to be $N(N > 0)$, and the SNR at each vehicle is configured as the same. The interference is ignored in this article. Hence, in this article the throughput of a vehicle is proportional to the communication bandwidth of the vehicle. When there are N vehicles in the fog cell, the bandwidth requirement of vehicles is assumed to be governed by a uniform distribution, that is, $B_i \sim U(0,2B_{ave})$, $1 \leq i \leq N$. Considering the real bandwidth requirement from N vehicles, the bandwidth requirement $B_i \sim U(0,2B_{ave})$, $1 \leq \mathrm{i} \leq N$ from n, $n \leq B/\mathrm{B}_{ave}$ vehicles is assumed to be less than the average bandwidth requirement B_{ave} in the fog cell. The throughput of a vehicle is C_j when the bandwidth of a vehicle is allocated by B_j. For the traditional average bandwidth allocation scheme, the maximum available bandwidth for a vehicle is B_{ave} and then the total bandwidth allocated for all vehicles in fog cell is

$$B_{tra} = \sum_{j=1}^{n} B_j + (N-n) \times B_{ave}.$$

Consequently, the throughput of the fog cell is

$$C_{tra} = \sum_{j=1}^{n} C_j + (N-n) \times C_{ave}.$$

Based on the proposed adaptive bandwidth allocation scheme, the un-occupied bandwidths of n vehicles can be reused for the other $N - n$ vehicles in the fog cell. The total requirement bandwidth of the other $N - n$ vehicles is

$$\sum_{j=n+1}^{N} B_j$$

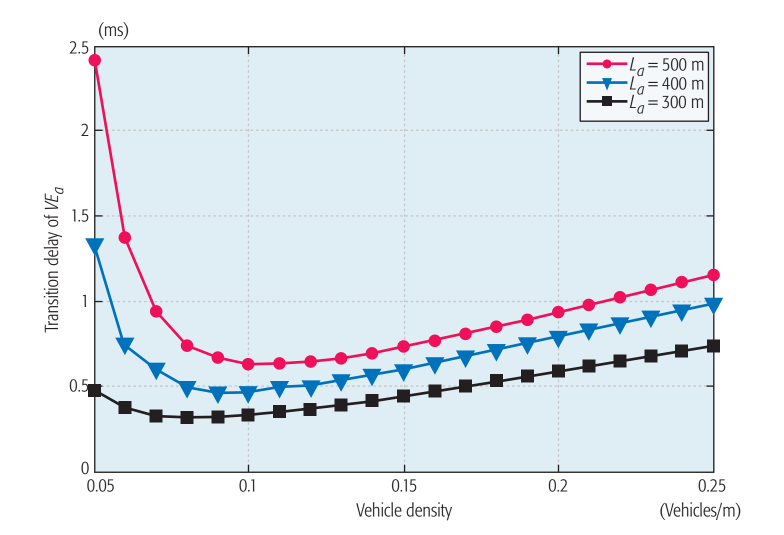


Figure 4. Transmission delay with respect to the vehicle density considering different transmission distances.

and the total available bandwidth of the other $N - n$ vehicles is

$$B - \sum_{j=1}^{n} B_j.$$

Therefore, the throughput of a fog cell adopting the adaptive bandwidth allocation scheme is

$$Min\left\{ \sum_{j=1}^{N} B_j \quad , \quad B \right\}.$$

When the parameters are configured as C = 1000 Mb/s and C_{ave} = 33 Mb/s, the throughput of a fog cell is compared to two bandwidth allocation schemes in Fig. 5. It is shown that the throughput of a fog cell with the adaptive bandwidth allocation scheme is always larger than the throughput of a fog cell with the average bandwidth allocation scheme. The reason is that the unoccupied bandwidths in the average bandwidth allocation scheme could be utilized in the adaptive bandwidth allocation scheme. When the bandwidth allocation scheme is given, the throughput of a fog cell first increases with the increase of the number of vehicles in a fog cell. When the number of vehicles is larger than 30, the throughput of a fog cell remains stationary. The reason is that all available bandwidth in a fog cell has already been allocated for vehicles. In this case, there are not any bandwidths to be allocated for additional vehicles even if the number of vehicles is larger than a specified threshold. Consequently, the throughput of a fog cell has to remain stationary when the number of vehicles is larger than a specified threshold.

Challenges of 5G Vehicular Networks

With the development of 5G mobile communication systems, high-speed wireless communications are satisfied by millimeter-wave and massive MIMO technologies. Furthermore, multimedia wireless communications are expected to be realized for 5G vehicular networks. Based on 5G high-speed wireless communications, pilot-

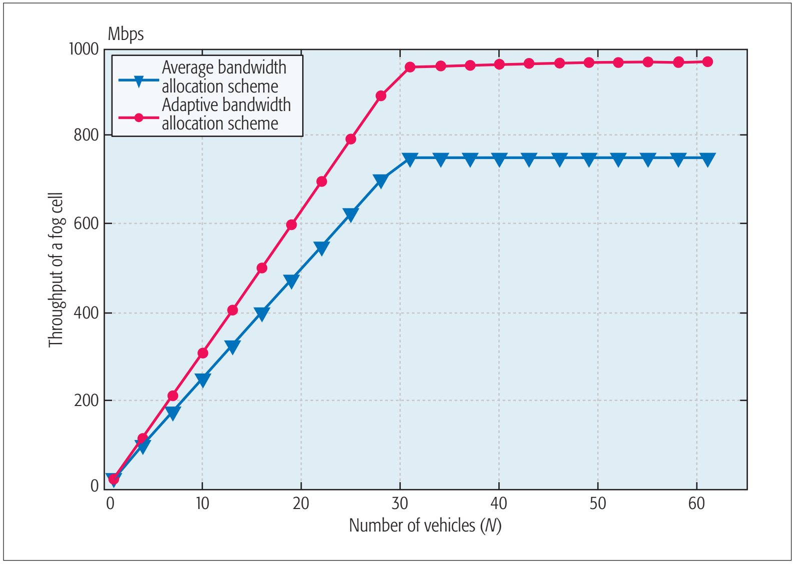


Figure 5. Throughput with respect to the number of vehicles in a fog cell.

less vehicles are emerging to change our future life. It is well known that future pilotless vehicles need to be supported by highly reliable and effective vehicular networks. However, some potential challenges and issues still need to be further investigated for 5G vehicular networks.

The low delay issues. When pilotless vehicles are deployed for city transport systems, not only traffic information but also road information should be transmitted to pilotless vehicles by vehicular networks. In general, safety message transmissions have a very low delay constraint, such as less than 1 ms. When there are many relay vehicles for a multihop relay vehicular network, the transmission delay of the warning message will be larger than a given threshold. For some extreme cases, the delay issues could cause fatal accidents. How to optimize route solution is still a key technology for 5G vehicular networks.

The frequent handover issues. In this article the fog cell is proposed to solve frequent handover between the RSU and vehicles. However, the handover among vehicles is still an issue for the multihop relay link in a fog cell. When a lot of vehicles are handed off between adjacent fog cells, the handover will be simultaneously generated for fog cells and the multihop relay links. In this case, the complexity of handover is obviously increased for 5G vehicular networks. Moreover, the propagation delay of the warning message needs to be minimized for vehicle handover in 5G vehicular networks.

The high service efficiency challenges. For future pilotless vehicles, vehicles are not only the transport tools but also entertainment centers for users. Different multimedia services need to be provided by 5G vehicular networks. Hence, massive wireless traffic is expected to increase for 5G vehicular networks. It is a great challenge to improve the service efficiency for 5G vehicular networks.

The architecture of 5G vehicular networks. To reduce the transmission delay of warning messages, a distributed network architecture is adopted for the fog cell of 5G vehicular networks. To support ITSs, the centralized network architecture is adopted for the core network of 5G vehicular networks. In this case, SDN is proposed to flexibly connect different types of network architectures. However, the scalability and compatibility of 5G vehicular networks are great challenges, especially because there are two types of network architectures in 5G vehicular networks.

Conclusions

With the development of pilotless vehicles, vehicular networks have to face rigorous performance requirements in future ITSs. 5G mobile communications, cloud computing, and SDN technologies provide potential solutions for future vehicular networks. In this article we propose a new architecture of 5G software defined vehicular networks integrating these technologies. Moreover, fog cells are established at the edge of 5G software defined vehicular networks, which utilize multihop relay networks to reduce the frequent handover between the RSU and vehicles. Simulation results indicate that there is a minimum transmission delay of 5G software defined vehicular networks considering different vehicle densities. Moreover, the throughput of fog cells in 5G software defined vehicular networks is better than the throughput of traditional transportation management systems. When the proposed challenges of 5G vehicular networks have been solved, 5G software defined vehicular networks could provide enough flexibility and compatibility to satisfy future pilotless vehicles and ITSs.

References

[1] S. Chen *et al.*, "User-Centric Ultra-Dense Networks (UUDN) for 5G: Challenges, Methodologies, and Directions," *IEEE Wireless Commun.*, vol. 23, no. 2, Apr. 2016, pp. 78–85.
[2] M. X. Gong *et al.*, "A Directional CSMA/CA Protocol for mmWave Wireless PANs," *Proc. IEEE WCNC*, Apr. 2010, pp. 1–6.
[3] X. Ge *et al.*, "Vehicular Communications for 5G Cooperative Small Cell Networks," *IEEE Trans. Vehic. Tech.*, vol. 65, no. 10, Oct. 2016, pp. 7882–94.
[4] G. Araniti *et al.*, "LTE for Vehicular Networking: A Survey," *IEEE Commun. Mag.*, vol. 51, no. 5, May 2013, pp. 148–57.
[5] G. Karagiannis *et al.*, "Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions," *IEEE Commun. Surveys & Tutorials*, vol. 13, no. 4, July 2011, pp. 584–616.
[6] W. Zhang *et al.*, "Multi-Hop Connectivity Probability in Infrastructure-Based Vehicular Networks," *IEEE JSAC*, vol. 30, no. 4, Apr. 2012, pp. 740–47.
[7] T. Taleb and A. Ksentini, "VECOS: A Vehicular Connection Steering Protocol," *IEEE Trans. Vehic. Tech.*, vol. 64, no. 3, Mar. 2015, pp. 1171–87.
[8] S. Sezer, S. Scott-Hayward, P K. Chouhan, *et al.*, "Are We Ready for SDN? Implementation Challenges for Software-Defined Networks," *IEEE Commun. Mag.*, vol. 51, no. 7, July 2013, pp. 36–43.
[9] M. Jutila, "An Adaptive Edge Router Enabling Internet of Things," *IEEE Internet of Things J.*, vol. 3, no. 6, Dec. 2016, pp. 1061–69.
[10] K. Liu *et al.*, "Cooperative Data Scheduling in Hybrid Vehicular Ad Hoc Networks: VANET as A Software Defined Network," *IEEE/ACM Trans. Net.*, vol. 24, no. 3, June 2016, pp. 1759–73.
[11] T. Taleb and K. Ben Letaief, "A Cooperative Diversity Based Handoff Management Scheme," *IEEE Trans. Wireless Commun.*, vol. 9, no. 4, Apr 2010, pp. 1462–71.
[12] M. Feng, S. Mao, and T. Jiang, "Enhancing the Performance of Future Wireless Networks with Software Defined Networking," *Front. Info. Tech. Electron. Eng.*, vol. 17, no. 7, July 2016, pp. 606–19.
[13] A. Bradai *et al.*, "Cellular Software Defined Networking: A Framework," *IEEE Commun. Mag.*, vol. 53, no. 6, June 2015, pp. 36–43.

[14] X. Ge *et al.*, "Energy Efficiency of Small Cell Backhaul Networks Based on Gauss-Markov Mobile Models," *IET Networks*, vol. 4, no. 2, Mar. 2015, pp. 158–67.

[15] I. Stojmenovic and S. Wen, "The Fog Computing Paradigm: Scenarios and Security Issues," *Proc. FedCSIS*, 2014, pp. 1–8.

Biographies

Xiaohu Ge [M'09, SM'11] (xhge@hust.edu.cn) is currently a full professor with the School of Electronic Information and Communications at Huazhong University of Science and Technology (HUST), China, and an adjunct professor with the Faculty of Engineering and Information Technology at the University of Technology Sydney, Australia. He received his Ph.D. degree in information and communication engineering from HUST in 2003. He is the director of the China International Joint Research Center of Green Communications and Networking. He has published more than 140 papers in international journals and conferences. He served as the General Chair for the 2015 IEEE International Conference on Green Computing and Communications (IEEE GreenCom). He has served as an Editor for *IEEE Transaction on Green Communications and Networking*, among others.

Zipeng Li (zipengli91@mail.hust.edu.cn) received his Bachelor's degree in telecommunication engineering and Master's degree in communication and information system from HUST in 2011 and 2014, respectively, where he is currently working toward his doctoral degree. His research interests include vehicular networks and 5G mobile communication systems.

Shikuan Li (m201671826@mail.hust.edu.cn) received his Bachelor's degree in communication and information systems from HUST in 2016, where he is currently working toward his Master's degree. His research interests include vehicular networks and 5G mobile communication systems.

A Scalable and Quick-Response Software Defined Vehicular Network Assisted by Mobile Edge Computing

Jianqi Liu, Jiafu Wan, Bi Zeng, Qinruo Wang, Houbing Song, and Meikang Qiu

Presently, IEEE 802.11p and evolving 5G are the mainstream radio access technologies in the vehicular industry, but neither of them can meet all requirements of vehicle communication. In order to provide low-latency and high-reliability communication, the authors propose an SDN-enabled network architecture assisted by MEC, which integrates different types of access technologies.

ABSTRACT

Connected vehicles provide advanced transformations and attractive business opportunities in the automotive industry. Presently, IEEE 802.11p and evolving 5G are the mainstream radio access technologies in the vehicular industry, but neither of them can meet all requirements of vehicle communication. In order to provide low-latency and high-reliability communication, an SDN-enabled network architecture assisted by MEC, which integrates different types of access technologies, is proposed. MEC technology with its on-premises feature can decrease data transmission time and enhance quality of user experience in latency-sensitive applications. Therefore, MEC plays as important a role in the proposed architecture as SDN technology. The proposed architecture was validated by a practical use case, and the obtained results have shown that it meets application-specific requirements and maintains good scalability and responsiveness.

INTRODUCTION

Developments in wireless communications, sensing, and cloud computing technologies have caused transformations in the automotive industry [1, 2]. Nowadays, a vehicle can exchange information with other vehicles (V2V), surrounding infrastructure (V2I), the Internet (V2N), roadside pedestrian (V2P), and a back-end cloud server [3]. In other words, vehicles can exchange information with everything (V2X). In the global context of road transport, vehicle connectivity is a critical enabler to support the take-off of new cases such as automated overtake, autonomous driving, cooperative collision avoidance, see-through, bird's eye view, and traffic prediction [4, 5]. Therefore, many auto manufacturers are working on development of V2X communication technologies in order to capture future business opportunities. Nevertheless, the mainstream V2X communication networks are IEEE 802.11p-based vehicular ad hoc networks (VANETs) and evolving fifth generation (5G)-based cellular networks.

Because of low latency in V2V communication, the VANET has achieved success in active safety applications such as cooperative collision avoidance. However, if the communication is performed in a rural area where vehicles are sparse, the communication link might be disconnected because there are insufficient roadside units (RSUs) for relaying. Therefore, the coverage of a VANET is constrained by the density of RSUs. Moreover, the drawback of IEEE 802.11p is the bandwidth limitation to 10 Mb/s per channel, which impacts augmented reality (AR) applications [6] on video transmission because their required data rate is 40 Mb/s. Nonetheless, the 5G cellular network with wide spectrum, multiple-input multiple-output, and ultra-dense network technologies can realize 7.5 Gb/s data rate in a stationary environment, and achieve stable connection at 1.2 Gb/s in a mobile environment, that is, in a vehicle at the speed of 100 km/h, based on 28 GHz spectrum [7]. Since the cellular network was originally designed for mobile broadband traffic, it lacks support for V2V communication. Recently, the 5G Public Private Partnership (5G-PPP) has launched initial research studies on connected vehicles. Furthermore, in Release 14, device-to-device (D2D) communication was proposed for direct data exchange among users by bypassing infrastructure within 1 ms delay, and it was anticipated that V2V direct communication will be supported by 5G in forthcoming years [8]. Hitherto, neither of the mentioned technologies can satisfy all requirements of V2X communication; thus, a new method that integrates the advantages of both access technologies and realizes high-reliability and low-latency V2X communication is needed. However, developing such a method is very challenging.

Software defined networking (SDN) represents an emerging network paradigm [9] that has the potential ability to join cellular networks and VANETs. First, SDN permits independent deployment of control, traffic forwarding, and processing entities. Resources such as RF transceivers are considered in the data plane, which allows separate optimization of platform technology and software life cycles. Second, the logically centralized control improves the service efficiency of resources. Third, the programmability makes the network more agile, so the application can select the proper radio access interface to deliver data [10].

Digital Object Identifier:
10.1109/MCOM.2017.1601150

Jianqi Liu is with Guangdong Mechanical & Electrical College; Jiafu Wan is with South China University of Technology; Bi Zeng and Qinruo Wang are with Guangdong University of Technology; Houbing Song is with West Virginia University; Meikang Qiu is with Pace University.

0163-6804/17/$25.00 © 2017 IEEE

Originally, SDN was designed for and deployed in wired network environments with high-speed switches, such as data center networks and campus networks [11]. Since then, some researchers have applied SDN in wireless sensor networks, which has provided the expected results [12, 13]. However, vehicle mobility and the rapid changes of network topology, network partitioning, data synchronization among distributed SDN controllers, seamless handover between fast-moving vehicles and controllers, and V2V communication in SDN controller-miss situations represent new research topics.

Although applying SDN in a heterogeneous vehicular network is challenging, SDN brings new insights and has high potential to improve agility, reliability, scalability, and latency performance. Besides, in order to decrease the overall delay and offload the traffic load from the backbone network, the mobile edge computing (MEC) technology is introduced into the proposed architecture [14].The main contributions of this article are as follows. A scalable SDN-enabled architecture that integrates a heterogeneous vehicular network and offers reliable communication services based on precise application-specific requirements is proposed. The scalability, responsiveness, reliability and agility of the proposed architecture are validated by a case study, named reliable communication in urban traffic management. Some of the key technical issues, including date synchronization, seamless handover, and reliable communication, are discussed in detail.

The article is organized as follows. The proposed SDN-enabled architecture is described and explained in detail. The proposed architecture was validated by a case study, reliable communication in urban traffic management, and the obtained results are provided. Afterward, three key technical issues and future work guidelines are presented. Finally, brief conclusions are given.

SDN-Enabled Architecture

In this section, the design details of the proposed architecture are addressed. There are two basic design principles. First, the proposed architecture should be protocol-agnostic and maintain scalability. The emerging technologies can easily be joined to this architecture. In addition, when one of the access technologies is defeated in the market by competition, its exit does not affect the system operation. Second, quick-response cloud service is necessary for new applications such as traffic prediction based on real-time road conditions. A vehicle, constrained by limited computing resources, has no ability to process the huge volume of uploading traffic data captured by onboard or roadside sensors; thus, this type of task should be outsourced to the infrastructure.

Application-Specific Requirements

The automotive industry has experienced advanced transformations due to innovative technologies, but the communication requirements in terms of latency, reliability, and scalability are beyond the capabilities of current 4G (or the evolving 5G) and VANETs. The application-specific requirements of five representative cases — tje cooperative collision avoidance system (CCAS), bird's eye view system (BEVS) at the road intersection, augmented reality (AR), intelligent navigation system (INS) based on real-time road conditions, and 4K live video — are presented according to end-to-end (E2E) latency, reliability (maximum tolerable packet loss rate), and data rate. The requirements on key performance indicators (KPIs) for each use case are presented in Table 1.

Use case	E2E latency (ms)	Reliability	Data rate (Mb/s)
CCAS	10	10^{-5}	Less than 5
BEVS	50	10^{-3}	40
AR	100	10^{-2}	100
INS	100	No	50
4K live video	500	No	40 (per video)

Table 1. The KPI requirements for V2X use cases [15].

Cooperative collision avoidance is a life saving use case with an ultra-high reliability requirement, and the reliability of 10^{-5} is derived from statistics of certain types of fatal accidents. The maximum steering frequency realizable by a car is about 10 Hz, while the oversampling factor of 10 is reasonable for updating a controller, so the updating cycle is 10 ms. The road safety applications, bird's eye view, and AR have higher delay tolerance than the life saving use case, but the requirements on data rate are stricter. Entertainment like a live video can improve user experience, but it is not necessary. Suppose the video's resolution is set to 4K, and the frame rate is set to 120 fps with H.266 format; then the data rate of each video needed for smooth playing is approximately 40 Mb/s. If there are five people watching different videos simultaneously, the data rate should reach 200 Mb/s. Hence, the new network architecture should meet reliability limitation of 10^{-5}, E2E communication delay less than 10 ms, and data rate equal to or greater than 200 Mb/s simultaneously.

System Architecture

Although the VANET has been studied for 10 years and authorized as a standard protocol by the U.S. Department of Transportation six years ago, it still has not scored an overwhelming victory in V2X communications. The vehicle mobility and dynamic change of network topology result in poor network robustness. In addition, high cost of infrastructure construction impacts VANET commercial deployment. The evolving 5G access technology, with perfect KPIs (peak data rate is 20 Gb/s) and the existing network infrastructure, has been applied to V2X communications. Nevertheless, D2D communication technology in 5G is immature; thus, it is impossible to determine whether it is better than IEEE 802.11p in V2V applications. Currently, the most feasible approach is to fuse the different access technologies in order to maintain the scalability and flexibility.

The proposed SDN-enabled heterogeneous network architecture is presented in Fig. 1. The VANET has big advantages in V2V communication before D2D technology is practically used, such as ultra-low latency in endpoint-to-endpoint

The automotive industry has experienced advanced transformations due to innovative technologies, but the communication requirements in terms of latency, reliability and scalability are beyond the capabilities of current 4G (or the evolving 5G), and VANET.

D2D communication technology in 5G is immature, thus it is impossible to determine whether it is better than IEEE 802.11p in V2V applications. Currently, the most feasible approach is to fuse the different access technologies in order to maintain the scalability and flexibility.

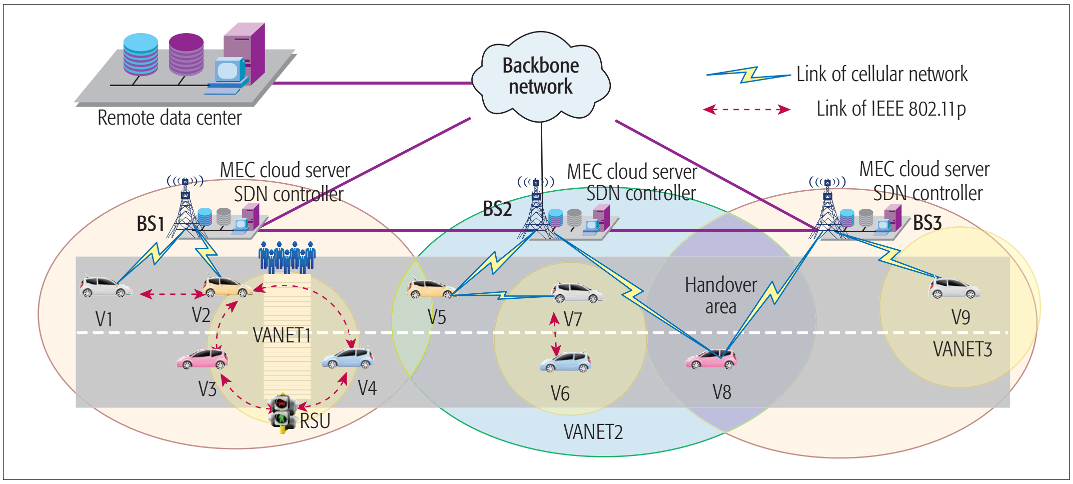


Figure 1. The SDN-enabled architecture of heterogeneous vehicular network.

communication in VANET1. However, in VANET3, the limited coverage issue is exposed because of insufficient RSUs. On the contrary, the cellular network has wider coverage, higher bandwidth, and wider spectrum. The wired network, which acts as a backbone network, is used to undertake the high-volume data exchange between the vehicle and the remote data center through a traditional network interface card (NIC). Hitherto, three types of access technologies have been used to obtain reliable V2X communication.

Unfortunately, even in an ideal network environment, the traffic data transmission is still time-consuming because of too long a distance between the vehicle and the remote data center. Due to the above, MEC technology has been introduced as communication infrastructure to tackle that issue by decreasing the overall delay. In MEC, the cloud computing resources and storage spaces are placed at the edge of the vehicular access network and are in close proximity to the mobile vehicle, which decreases the round-trip time of data packets substantially. An MEC cloud server can support delay-constrained response to client requests, and facilitates deployment of new latency-sensitive services for service providers. Another advantage of MEC is that it can offload traffic load from the backbone network. Since the applications are deployed on the MEC cloud server, and the MEC cloud server is installed on the base station (BS), the service request packets received by the BS should be redirected to itself by an inner (loopback) interface rather than delivered to the remote data center. Therefore, the inner interface is introduced as the fourth port in the proposed architecture. The existing heterogeneity, due to orchestrating four types of network access technologies, makes network management and integration challenging. Fortunately, SDN has potential to orchestrate these network entities.

The network space can be decoupled into three planes, the application plane, control plane and data plane, as shown in Fig. 2. The control plane communicates with the application plane and data plane through the application-controller plane interface (A-CPI) and the data-controller plane interface (D-CPI), respectively. As the network environment and client requirements change, the SDN controller is responsible for continuously updating network status and service requirements, and provides a policy-based optimum configuration. The key components are discussed in detail in the following.

Data Plane and Resources: The data plane abstracts the underlying network resources, such as RSUs, vehicles, BS transceivers, and Ethernet interfaces, as SDN switches. These switches comply with unified scheduling, follow the OpenFlow protocol, and forward traffic to the next hop along the path toward the selected destination network according to the control plane logic. The virtualization of a network entity decouples traffic forwarding and processing from control. That is, the data plane does not care about control policy, and it just uses what the control plane built to dispose of incoming and outgoing frames and packets. In a mobile network, radio spectrum is the most important resource. SDN-enabled architecture replaces protocol-specific radio hardware with protocol-agnostic digital transceivers, which makes the radio resources allocate or share dynamically according to the server context and client requirements. Meanwhile, the other hardware can easily be sourced and reproduced as the protocol-agnostic feature.

Control Plane and SDN controller: The control is transferred from individual switches to the central controller, and the control entity is transferred from communication devices to software that is deployed on the MEC cloud server as a software component, runs on the commodity operating system, and allows dynamic access and administration. This transformation permits a client to exchange information with the SDN controller during the service lifetime according to changes in client needs or the state of the client's virtual resources. The logically centralized control and programmability are SDN's main features. In the vehicle communication scenario, the topology changes as the vehicle moves, so the topology management in the SDN controller needs to acquire position, direction, velocity, and network connectivity in real time. The database and forwarding information base resources are also dynamically adjusted according to topology change.

D-CPI and OpenFlow: The OpenFlow logical

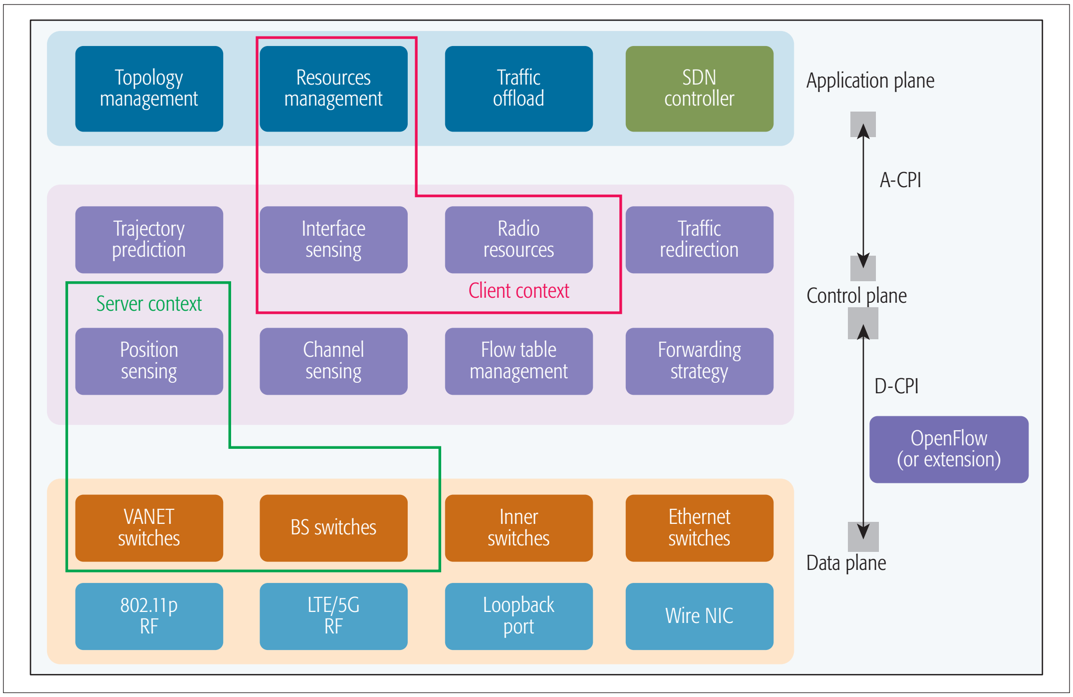


Figure 2. The core of SDN controller.

switch consists of one or more flow tables and a group table, which perform packet lookups and forwarding. The SDN switch communicates with the controller using a unified interface and dedicated channel named the control channel, as shown in Fig. 3. The controller can add, update, and delete flow entries in flow tables either reactively or proactively. When the packet is processed by the flow table, it is matched against flow entries of the flow table in order to select the corresponding flow entry. If the flow entry is found, the instruction set included in that flow entry is executed. If a packet does not match the flow entry in the flow table, a further procedure that includes packet dropping, passing to another table, or sending to the controllers depends on the table configuration.

Application Plane: The classical SDN model contains an application plane populated by instances of applications. An application represents a client entity that might request some kind of services from an SDN controller server, which usually involve data or resource plane exchanges. These visual applications make the network management more flexible. The manager can complete the network configuration by mouse and keyboard.

The SDN-enabled vehicular network has realized the integration of the VANET with the 5G cellular network. The all-purpose protocol-agnostic transceivers are employed in both VANETs and cellular networks, which facilitates replacement and maintenance in the data plane. The control, including forwarding policy, is transferred to the MEC cloud server as a software component; thus, network management becomes more flexible. The proposed architecture meets KPI requirements by combining the VANET and cellular network, while retaining scalability. Furthermore, the introduction of the MEC cloud server decreases the data transmission time and offers quickly responding service.

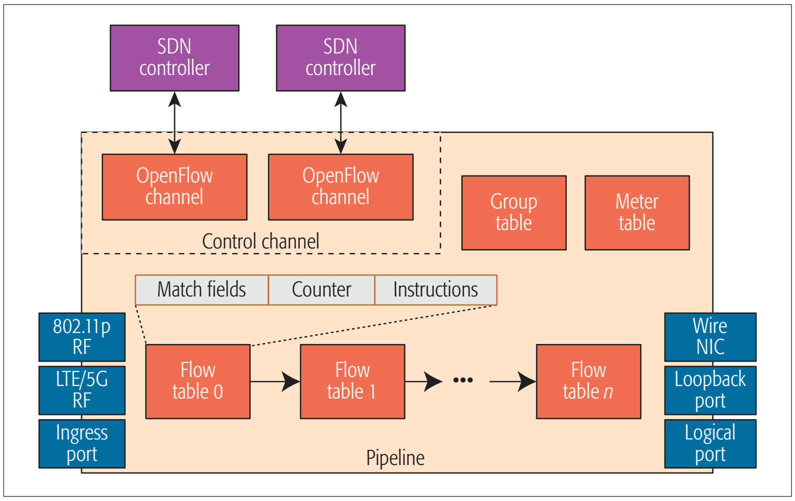


Figure 3. OpenFlow in vehicular network.

A Case Study: Reliable Communication in Urban Traffic Management

Due to increasing traffic jams, urban traffic efficiency has become a general concern. Although the construction of extra road infrastructure can reduce congestion, the effect is limited because the number of new vehicles is rising too fast. Consequently, the connected vehicle is considered as one of the most feasible solutions. High-density platooning, one connected vehicle case, can create closely spaced multiple-vehicle chains on a highway, and reduce the current distance between vehicles below 1 m, which will further increase the road utilization rate. However, high-reliability and low-latency V2V communication is needed. Vehicles within a platoon can constantly exchange their kinematic state information in real time, which allows following vehicles to implement throttle and braking control, keeping

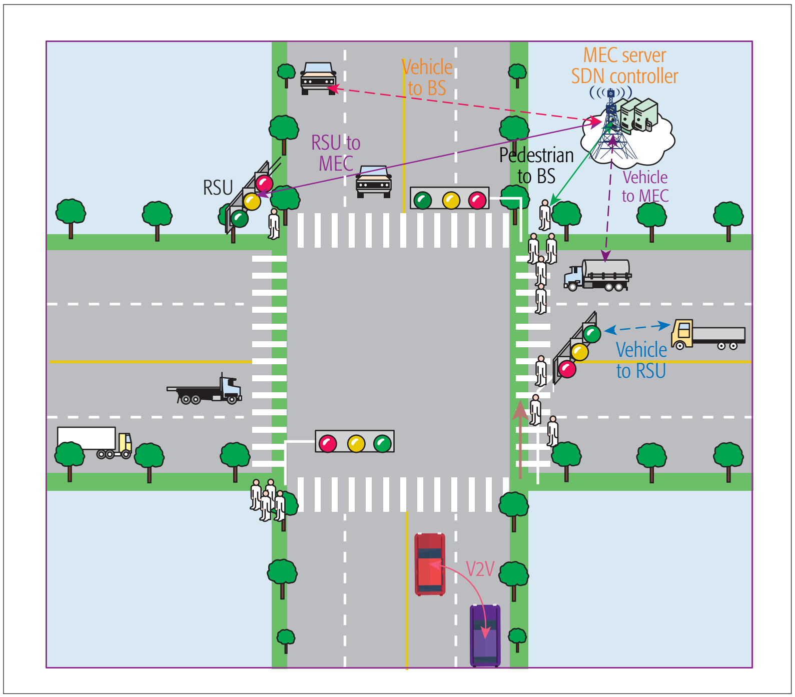


Figure 4. Reliable and low-latency communication in urban traffic management.

the distance constant. In the urban environment, the traffic congestion always happens at intersections. In highly efficient and well organized traffic entities coordination at intersections that smooth traffic and improve throughput, V2X communication plays a key role.

In order to verify the proposed SDN-enabled network architecture, it is employed to manage this complex communication issue. The data plane is simplified by the protocol-agnostic NIC and RF transceivers. The underlying devices dispose of incoming and outgoing packets according to the control plane. As shown in Fig. 4, the used control policy is discussed from six perspectives.

Vehicle to vehicle: The direct communication is performed between vehicles, which act as both users and relays for packet forwarding. The multihop communication of a VANET is an advantage, but if the number of hops is larger than three, the reliability decreases significantly and the latency cannot be guaranteed. In the proposed architecture, if the number of hops is less than three, packet forwarding takes place in a VANET or else in a cellular network.

Vehicle to RSU: The vehicle collect roads condition data from RSUs through vehicle-to-RSU communication, such as data synchronization between the vehicle and a roadside traffic light. In the proposed architecture, the RSU is not responsible for packet forwarding, and the data forwarding function is transferred to the BS.

Vehicle to BS: Since the SDN controller is deployed on the BS, the BS plays multiple roles. It is responsible for delivery of control information to OpenFlow switches and providing responses to different events such as data forwarding failure. In addition, the BS is also responsible for offloading traffic load. All listed tasks are completed by vehicle-to-BS communication.

Vehicle to MEC: The MEC cloud server, which is considered as infrastructure, is transparent to the client. The client (vehicle) requests service from or delivers packets to a remote cloud server. If the service is deployed on an MEC could server, the BS redirects it to the MEC cloud server.

RSUs to MEC: RSUs collect the real-time road conditions and deliver them to the cloud server. These raw data are redirected to the MEC cloud server by the BS.

MEC to remote cloud server: All traffic data should be preprocessed by the MEC could server and then delivered to the remote cloud server by means of data synchronization. MEC usually stores recent traffic data and responds to events based on the real-time data. On the other hand, the remote cloud server stores traffic data permanently, and makes a traffic prediction based on real-time and historical data.

In order to simplify the verification of the performance, three essential use cases — CCAS, BEVS, and INS — are used to test the KPI of the proposed architecture in the NS3 simulator. The testing site is set as a square with side length of 30 m. In order to simulate the performance in different densities of vehicles, 5 vehicles, 25 vehicles, 50 vehicles, and 100 vehicles are used to represent four levels (i.e., sparse, medium, dense, and jam traffic, respectively). Figure 5a, the delay time of IEEE 802.11p-based V2V communication in CCAS, is less than 10 ms for all density levels, and the latency performance does not deteriorate sharply for the highest vehicle density. In Fig. 5b, the reliability of CCAS is better than 10^{-5}. Further, the reliability of BEVS is low, but the obtained reliability still meets the requirement. In Fig. 5c, the cellular network could support more than 10 Gb/s data rate, which is very good. Last, in all test cases, the KPIs of the proposed architecture are better than KPI requirements.

Discussion and Future Work

In the proposed SDN-enabled architecture, the control and forwarding functions are decoupled, which has many advantages over hardware-based mobile network designs. First, the scalability performance is good. Since both wireless and wired transceivers are protocol-agnostic, the transceivers can be sourced and reproduction can be obtained easily, while the production and maintenance cost is lower than protocol-specific cost. Second, controller upgrade and management are very convenient and flexible. The SDN controller runs on general-purpose computers, so the software-only upgrade is more flexible and capacity can be enhanced easily by adding more computing power. Meanwhile, the programmability makes the network service more agile, which facilitates network management. Third, the MEC cloud server, which acts as communication infrastructure, enables the proposed architecture to meet the delay-constrained requirement. However, vehicle mobility, rapidly changing network topology, and complex radio resources allocation policy cause new issues that should be considered. Therefore, three key issues are discussed in this section.

Network Partition and Data Synchronization

A small network with a modest number of switches can be handled easily by a logical central SDN controller. However, the number of vehicles in a vehicular network might be large, and the number and size of forwarding entries might exceed the memory resources of a single SDN controller. In addition, the wide-range coverage causes high latency in data exchange between controllers and switches. Previously, in order to overcome this problem, the control logic usually partitioned the network into subnetworks and replicated switch states with the corresponding subnetwork; the distributed SDN controller was responsible for managing each subnetwork. This method operates well in stationary networks, but it does not fit the quick-changing vehicular network, because switch state needs to be updated continually as the vehicle moves. Therefore, the flood mechanism is very important for synchronization of each switch state. Fortunately, the position of a vehicle can be acquired by an onboard GNSS receiver, and its trajectory can be predicted, so the predictable and position-based flood mechanism represents an effective way to cope with the data synchronization problem.

Seamless Handover for a Fast Moving Vehicle

The term "handover" in a cellular network refers to the process of transferring an ongoing call or data session from one channel to another channel. As shown in Fig. 1, V8 is traveling from BS3's cell to BS2's cell. When V8 gets outside BS3's cell and enters BS2's cell, handover must be executed in order to avoid service termination. In an SDN-enabled vehicular network, the handover mechanism is more complex than in a traditional cellular network:

- The radio resources might need to renegotiate with a new SDN controller.
- A set of flow tables needs to be updated according to the topology change.
- With the introduction of an MEC cloud server, live migration and service redirection are necessary actions, which increase the complexity of handover.

According to the vehicle's position, direction, velocity, and destination, the trajectory prediction component can estimate vehicle position in the near future, which helps to complete service migration and flow table entries update in advance. Similar to cellular networks, there are occurrences where a handoff is unsuccessful. This occurs when many people request 4K live video simultaneously and when the required bandwidth has exceeded the maximal limiting value; if a new vehicle enters that cell, the handover will fail. Therefore, the mechanism for failure recovery and error handling also needs to be considered carefully.

Reliable Communication in the Controller-Miss Situation

The infrastructure of a cellular network has wide coverage, but there are still signal blind areas, such as wilds. Since OpenFlow switches cannot exchange control information due to the lack of a BS, vehicles will experience communication failure. As is well known, multihop VANETs and future D2D communication technologies are able to share information in an endpoint-to-endpoint way. In order to maintain link connectivity in a controller-miss situation, some communication devices need to reserve the basic mechanism to maintain V2V communication and extend OpenFlow-compatible protocol simultaneously. However, that increases hardware cost, so a unified approach needs to be addressed in future work.

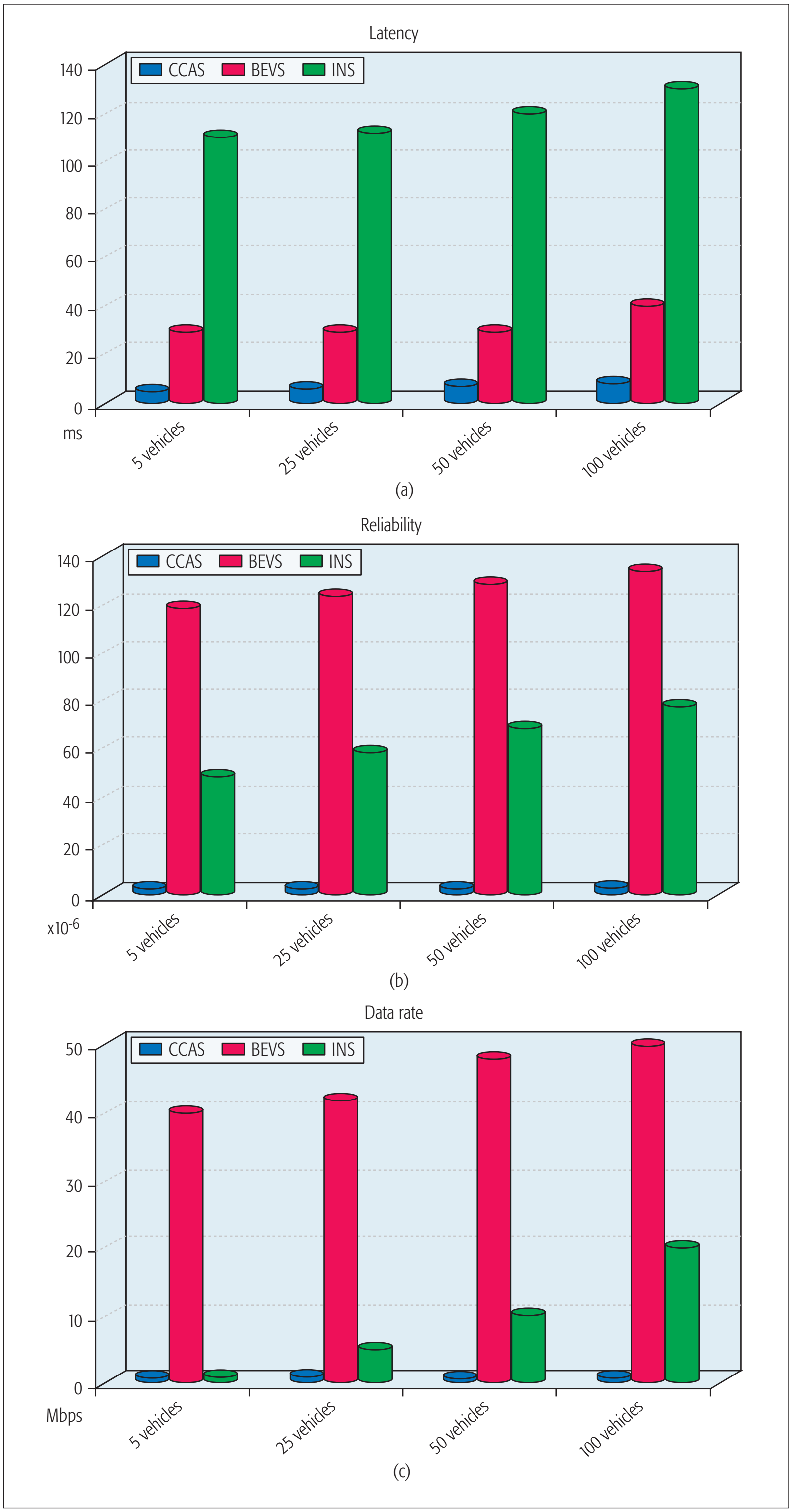


Figure 5. The KPI of the proposed architecture: a) latency performance indicator; b) reliability performance indicator; c) data rate performance indicator.

Conclusions

The V2X communication technology has provided advanced transformations in the automotive industry; hence, auto manufactures, mobile operators, and service providers have started the pro-

In order to maintain the link connectivity in a controller-miss situation, some communication devices need to reserve the basic mechanism to maintain V2V communication and extend OpenFlow-compatible protocol simultaneously. However, that increases hardware cost, so a unified approach needs to be addressed in future work.

cedure for V2X implementation in order to seize the attractive business opportunities. The proposed SDN-enabled heterogeneous vehicular network assisted by MEC can provide desired data rates and reliability in V2X communication simultaneously, which is validated by a practical use case. Meanwhile, the proposed architecture is developed to maintain scalability and rapid responsiveness. In terms of scalability, it has achieved protocol-agnostic hardware in the data plane and software-oriented control component in the control plane, which makes the hardware deployment more flexible. Moreover, since the control component runs on the commodity operation system, deployment, update, and administration can be easily implemented by software procedure. Furthermore, network configuration and forwarding policy can be updated in a timely manner according to fast changing context. On the other hand, in terms of responsiveness, the emerging MEC has facilitated the practical use of the cloud-based delay-constrained services, so the quality of user experience has been improved greatly. Therefore, the quick-response of MEC changes the traditional three-tier model of collection-delivery-processing.

Acknowledgment

The corresponding author is Dr. Jiafu Wan. This project was supported in part by grants from the Guangdong Natural Science Foundation (Nos. 2016A030313734 and 2016A030313735), the Fundamental Research Funds for the Central Universities (No. 2015ZZ079), the Guangdong Province Special Project of Industry-University-Institute Cooperation (No. 2014B090904080), and the Guangzhou science and technology plan projects (No. 201604020016).

References

[1] J. Wan, *et al.*, "VCMIA: A Novel Architecture for Integrating Vehicular Cyber-Physical Systems and Mobile Cloud Computing," *Mobile Networks and Applications*, vol. 19, no. 2, 2014, pp. 153–60.

[2] J. Jiang *et al.*, "An Efficient Distributed Trust Model for Wireless Sensor Networks," *IEEE Trans. Parallel and Distrib. Sys.*, vol. 26, no. 5, 2015, pp. 1228–37.

[3] J. Liu *et al.*, "A Survey on Position-Based Routing for Vehicular Ad Hoc Networks," *Telecommun. Sys.*, vol. 62, no. 1, 2016, pp. 15–30.

[4] J. Wan *et al.*, "Mobile Crowd Sensing for Traffic Prediction in Internet of Vehicles," *Sensors*, vol. 16, no. 1, 2016, pp. 88.

[5] G. Han *et al.*, "Green Routing Protocols for Wireless Multimedia Sensor Networks," *IEEE Wireless Commun.*, vol. 23, no. 6, Dec. 2016, pp. 140–46; DOI 10.1109/MWC.2016.1400052WC.

[6] J. Wu *et al.*, "Augmented Reality Multi-View Video Scheduling Under Vehicle-Pedestrian Situations," *Proc. ICCVE*, 2015, pp. 163–68.

[7] M. Nekovee, "Radio Technologies for Spectrum above 6 GHz — A Key Component of 5G," *Proc. 5G Radio Tech. Seminar: Exploring Technical Challenges in the Emerging 5G Ecosystem*, IET, 2015, pp. 1–46.

[8] J. Lee *et al.*, "LTE-Advanced in 3GPP Rel-13/14: An Evolution Toward 5G," *IEEE Commun. Mag.*, vol. 54, no. 3, Mar. 2016, pp. 36–42.

[9] J. Wan *et al.*, "Software-Defined Industrial Internet of Things in the Context of Industry 4.0," *IEEE Sensors J.*, vol. 16, no. 20, 2016, pp. 7373–80.

[10] S. Sezer *et al.*, "Are We Ready for SDN? Implementation Challenges for Software-Defined Networks," *IEEE Commun. Mag.*, vol. 51, no. 7, July 2013, pp. 36–43.

[11] N. McKeown *et al.*, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 38, no. 2, 2008, pp. 69–74.

[12] T. Luo *et al.*, "Sensor OpenFlow: Enabling Software-Defined Wireless Sensor Networks," *IEEE Commun. Lett.*, vol. 16, no. 11, 2012, pp. 1896–99.

[13] J. Liu *et al.*, "High-Efficiency Urban Traffic Management in Context-Aware Computing and 5G Communication," *IEEE Commun. Mag.*, vol. 55, no. 1, Jan. 2017, pp. 34–40.

[14] S. Davy *et al.*, "Challenges to Support Edge-as-A-Service," *IEEE Commun. Mag.*, vol. 52, no. 1, Jan. 2014, pp. 132–39.

[15] 5G-PPP, "5G Automotive Vision," 2015; https://5g-ppp.eu/white-papers/.

Biographies

Jianqi Liu [M] (liujianqi@ieee.org) is an associate professor with Guangdong Mechanical & Electrical College, China. He received Ph.D. and M.S. degrees from the Guangdong University of Technology (GDUT) of China. His current research interests are IoV, WSNs, and CPS.

Jiafu Wan (jiafuwan_76@163.com) [M'11] is a professor in the Guangdong Provincial Key Laboratory of Precision Equipment and Manufacturing Technology, South China University of Technology. Thus far, he has authored/co-authored more than 70 journal papers (with 60+ indexed by ISI SCIE) and 30 international conference papers. His research interests include cyber-physical systems, Industry 4.0, smart factory, industrial big data, industrial robots, and the Internet of Vehicles.

Bi Zeng (zb9215@gdut.edu.cn) is a professor in the School of Computers at GDUT. She received her Ph.D. and M.S. degrees from GDUT, and she is a member of CCF, Multi-Valued Logic and Fuzzy Logic Committee, China. Her current research interests include embedded systems, robot control techniques, and WSNs.

Qinruo Wang (wangqr2006@gdut.edu.cn) is a professor and a Ph.D. instructor in the School of Automation at GDUT. He received a B.S. degree from GDUT and an M.S. degree from Zhejiang University, China. His current research interests include automatic equipment and techniques, mechatronics, automatic network control, and wireless communication networks.

Houbing Song (h.song@ieee.org) [M'12, SM'14] received his Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, in August 2012. In August 2012, he joined the Department of Electrical and Computer Engineering, West Virginia University, Montgomery, where he is currently an assistant professor and the founding director of the Security and Optimization for Networked Globe Laboratory (SONG Lab, www.SONGLab.us). He was the very first recipient of the Golden Bear Scholar Award, the highest faculty research award at West Virginia University Institute of Technology (WVU Tech), in 2016.

Meikang Qiu [SM] (qiumeikang@yahoo.com) received his B.E. and M.E. degrees from Shanghai Jiao Tong University, and received his Ph.D. degree in computer science from the University of Texas at Dallas. Currently, he is an adjunct professor at Columbia University and an associate professor of computer science at Pace University. He is an ACM Senior Member. He has published 5 books, and 360 peer-reviewed journal and conference papers. He won the *ACM Transactions on Design Automation of Electrical Systems* 2011 Best Paper Award. Currently he is an Associate Editor of 10+ international journals, including *IEEE Transactions on Computers* and *IEEE Transactions on Cloud Computing*.

Software Defined Space-Air-Ground Integrated Vehicular Networks: Challenges and Solutions

Ning Zhang, Shan Zhang, Peng Yang, Omar Alhussein, Weihua Zhuang, and Xuemin (Sherman) Shen

ABSTRACT

This article proposes a software defined space-air-ground integrated network architecture for supporting diverse vehicular services in a seamless, efficient, and cost-effective manner. First, the motivations and challenges for integration of space-air-ground networks are reviewed. Second, a software defined network architecture with a layered structure is presented. To protect the legacy services in the satellite, aerial, and terrestrial segments, resources in each segment are sliced through network slicing to achieve service isolation. Then available resources are put into a common and dynamic space-air-ground resource pool, which is managed by hierarchical controllers to accommodate vehicular services. Finally, a case study is carried out, followed by discussion on some open research topics.

INTRODUCTION

The connected vehicle paradigm is to empower vehicles to communicate with the surrounding environment such as neighboring cars, roadside infrastructure, and traffic control centers, playing a vital role in the next generation intelligent transportation system (ITS) . With connected vehicles, a wide range of on-the-go services can be facilitated, including road safety (e.g., collision avoidance and intelligent traffic management), infotainment (e.g., social networking and online gaming), and location-dependent services (e.g., points of interest and route optimization) [1–3]. With its great potential, the connected vehicle is regarded as the next frontier for automotive revolution, and the number of connected vehicles is predicted to reach 250 million by 2020.[1] Extensive research and development efforts have been made from both industry and academia to get vehicles connected. The main enabling platforms include dedicated short-range communications (DSRC)-based 802.11p networks and cellular networks 14]. The former facilitates both vehicle-to-vehicle (V2V) and vehicle-to-roadside (V2R) communications, while the latter, such as Long-Term Evolution (LTE), can provide reliable access to the Internet. However, for 802.11p networks, it is costly and will take a long time to deploy large-scale network infrastructure such as roadside units (RSUs). Although the LTE network can exploit the existing infrastructure, it faces the issues of coverage and capacity. For instance, the LTE network has poor coverage in rural areas, while having potential network congestion in urban areas due to ever increasing wireless devices and services [5]. Furthermore, both 802.11p and cellular networks have great challenges to support high mobility, and highly mobile vehicles can suffer from frequent handovers to associate with new RSUs and base stations as the networks become even denser.

In addition to terrestrial networks, other complementary solutions, such as satellite and aerial networks, are under development. For example, the Intelsat satellite antenna (i.e., mTenna) can be embedded into the roof of a car to acquire satellite signals. Toyota's Mirai Research Vehicle equipped with mTenna can provide on-the-move services, which is demonstrated to achieve a data rate of 50 Mb/s. In addition, Tesla plans to launch 700 low-cost commercial satellites to provide Internet access, while Google will launch a fleet of 180 satellites.[2] In parallel, high-altitude platforms (HAPs) can also provide Internet access to vehicles [6]. As a matter of fact, Project Loon initiated by Google aims to leverage high-altitude balloons for broadband services in remote locations, and Facebook is attempting to deploy solar-powered drones to provide Internet access to underdeveloped areas.

To accommodate the diverse vehicular services with different quality of service (QoS) requirements in various practical scenarios (e.g., rural and urban), it is imperative to exploit specific advantages of each networking paradigm. For instance, densely deployed terrestrial networks in urban areas can support high data rate access, satellite communication systems can provide seamless connectivity to rural areas, while high altitude platforms (HAPs) can enhance the capacity for areas with high service demands. In addition, multi-dimensional real-time context-aware information regarding vehicular environments, such as in-vehicle, inter-vehicle, road conditions, and regional transport information, can be acquired to improve driving experience and facilitate intelligent traffic management. In

The authors propose a software defined space-air-ground integrated network architecture for supporting diverse vehicular services in a seamless, efficient, and cost-effective manner. The motivations and challenges for the integration of space-air-ground networks are reviewed, and then a software defined network architecture with a layered structure is presented.

[1] Gartner,"Connected Cars from a Major Element of Internet of Things," http://www.gartner.com/newsroom/id/2970017, 2015.

[2] http://www.dailymail.co.uk/sciencetech/article-2828539/Elon-Musk-s-mission-700-cheap-satellites-provide-internet-access-world.html, 2014.

Ning Zhang is with the University of Toronto; Shan Zhang, Omar Alhussein, Weihua Zhuang, and Xuemin (Sherman) Shen are with the University of Waterloo; Peng Yang is with Huazhong University of Science and Technology.

Digital Object Identifier: 10.1109/MCOM.2017.1601156

0163-6804/17/$25.00 © 2017 IEEE

From a service perspective, different QoS requirements imposed by diverse services should be satisfied in various vehicular networking scenarios in a cost-effective and flexible manner. However, the standalone territorial network has many challenges to meet such needs.

Segment	Entities	Altitude	Mobility	Round-trip	Advantages	Limitation
Space	GEO	35,786 km	Static to earth	250–280 ms	Broadcast/multicast, large coverage, rapid commercialization.	Long propagation delay; limited capacity, least flexibility, costly.
	MEO	3000 km	Medium fast	110–130 ms		
	LEO	200–3000 km	Fast	20–25 ms		
Air	HAP	17–22 km	Quasi-stationary	Medium	Large coverage, low cost, flexible movement.	Less capacity, link instability.
	UAV	Up to 30 km	Varying speeds			
Ground	DSRC	N.A.	Static	Lowest	Rich resources, high throughput.	Coverage, mobility support, vulnerable to disaster.
	Celluar	N.A.	Static			

Table 1. A comparison of different paradigms.

this article, we focus on space-air-ground integrated vehicular networks, by means of software defined networking (SDN) [7] to exploit heterogenous resources in an agile and flexible manner to support heterogenous vehicular services. The motivation and challenges for space-air-ground integrated networks are first discussed. Then a software defined space-air-ground integrated vehicular network architecture is proposed. The working relation is presented, along with the hierarchical network operation and big-data-assisted networking. The research directions are identified, followed by the conclusion of this work.

Space-Air-Ground Integrated Vehicular Network: Motivations and Challenges

Background

A satellite network is composed of a number of satellites, ground stations (GSs), and network operations control centers (NOCCs), and usually provides services for navigation, Earth observation, emergency rescue, and communication/relaying. Based on the altitude, satellites can be categorized into geostationary orbit (GSO), medium Earth orbit (MEO), and low Earth orbit (LEO) satellites. Satellite networks have the advantages of:

- Large coverage (3 GSO satellites or a constellation of LEO satellites, e.g., Iridium composed of 77 LEO satellites can provide the global coverage)
- Broadcasting/multicasting capability to support a large number of users simultaneously
- Reliable access for extreme scenarios such as in disaster relief

In addition, broadband satellite systems are expected to have capacity of 1000 Gb/s by 2020 [8]. However, the round-trip delay, especially with GSO and MEO satellites, is relatively large, which is not suited for real-time applications.

An aerial network formed by quasi-stationary HAPs (17–22 km above the earth) in the stratosphere can also be employed to provide broadband connectivity. HAPs mainly consist of unmanned airships (e.g., balloons) and aircrafts such as unmanned aerial vehicles (UAVs). The solar-powered airships can stay in the air for around five years. Compared to base stations in terrestrial communication networks, HAPs can have large coverage to offer services on a regional basis, and their movement can be partially controlled on demand to provide supplemental capacity in hotspot areas. Moreover, HAPs have the features of low cost, as well as incremental and easy deployment.

For terrestrial networks, a dedicated spectrum with bandwidth 75 MHz at 5.9 GHz has been allocated for DSRC, while cellular networks are now evolving to fifth generation (5G) wireless networks to support diverse services including vehicular communications. As for standardization, the Third Generation Partnership Project (3GPP) aims to develop a set of LTE specifications for vehicular environments (LTE-V). The terrestrial networks can provide high data rates to users, but the network coverage in rural and remote areas is poor, and the capacity in hotspots needs to be enhanced. A comparison of the different networking paradigms is given in Table 1.

Motivations for Integration

From a service perspective, different QoS requirements imposed by diverse services should be satisfied in various vehicular networking scenarios in a cost-effective and flexible manner. However, the standalone terrestrial network has many challenges to meet such needs, as detailed in the following.

•The coverage of territorial networks in rural areas (e.g., mountain areas) and highways is poor. It is very costly to deploy more network infrastructure in those sparsely populated areas. Instead, existing satellite communication systems or HAPs can efficiently provide vehicular connectivity due to their large coverage. For instance, a satellite cell diameter can be several hundred kilometers, which is equivalent to several thousands of terrestrial base stations (BSs) each having a coverage area of several kilometers in diameter [9].

•Moving vehicles suffer from frequent handover in territorial networks, degrading on-the-move service performance. The coverage diameter of an LTE macrocell BS is around 1 km, while the coverage diameter of a small cell BS (SBS) is less than 300 m [10]. Consequently, the moving vehicles on highways will experience highly frequent handovers from BS to BS.

•The diverse vehicular services cannot be served efficiently by a single technology. The LTE network can provide a high data rate for individual vehicular users, but it is not designed to efficiently broadcast similar contents to a large number of users. In such a case, satellite/HAP communication systems are more efficient thanks to their efficient broadcasting and multicasting capacity.

•With high spatial-temporal dynamics in traffic loads due to vehicle mobility, network congestion can occur even in urban/suburban areas. However, it is not cost effective to add more terrestrial network resources to meet the peak traffic demands. Exploiting available resources from other systems such as HAPs can be more effective to accommodate the bursty traffic demands.

Through interworking, the advantages from different segments can be exploited to support multifarious vehicular services and scenarios in an efficient and cost-effective manner. For instance, territorial networks can serve individual vehicular users through high-rate unicast in urban/suburban areas, while satellite networks help achieve ubiquitous coverage in rural and remote areas. Low-cost HAPs can be utilized to boost capacity at areas with poor or congested terrestrial infrastructure deployment; they can be repositioned to provide emergency communications in disaster scenarios. Moreover, both satellites and HAPs can provide road information and geolocation information to assist terrestrial networks, facilitate information dissemination as relays, and relieve the demands on terrestrial networks through data offloading.

Challenges

To integrate the space-air-ground networks for vehicular services, there are many challenges:

•Inter-operation: Different networking paradigms are supported by various communication standards and communication links, and equipped with different types of network devices. Currently, each communication system is closed due to proprietary network equipments, and dedicated gateways are required for protocol and format conversions, which significantly limits interoperability.

•Network management: Each network already comprises a large number of devices with dedicated interfaces for configuration and control. The integrated network will become more complex to manage. Moreover, due to the variety of devices with different hardware and software specifications, it is difficult to perform reconfiguration flexibly and dynamically, to either enforce high-level policies or respond to various events such as network congestion and link failures.

•Dynamic networking: The mobility of satellites and HAPs with respect to the Earth complicates integrated network operation. For instance, LEO satellites spin around the Earth in periods less than 130 minutes,[3] leading to resource availability dynamics. Moreover, vehicle mobility results in time-varying and non-uniform geographical vehicle distribution.

•QoS provisioning: Integrated networks should accommodate a wide range of vehicular services with different QoS requirements efficiently. However, the resources in the integrated network exhibit high heterogeneity. Moreover, the resource availability for vehicular services varies over time, since each segment dynamically allocates resources to support its legacy services[4] in priority.

Software Defined Space-Air-Ground Integrated Vehicular Networks

As an emerging network architecture, SDN separates the control plane and data plane, introduces logically centralized control with a global view of the network, and facilitates network programmability/reconfiguration through open interfaces. With SDN, a dynamic, manageable, cost-effective, and adaptable network can be enabled.[5] In this section, based on SDN, we propose a software defined space-air-ground integrated vehicular (SSAGV) network architecture to address the aforementioned challenges.

Network Architecture

As shown in Fig. 1, the SSAGV network comprises three main segments: space, air, and ground segments. SDN controllers can be deployed on powerful servers or in cloud computing, which regulate the network behaviors and manage network resources dynamically. Considering different segments have distinct characteristics such as communication standards and diverse network devices with various functions, the control and communication interfaces of SDN controllers for each segment should be dedicated to the corresponding segment. In fact, software defined paradigms for satellite, aerial, and terrestrial networks have been proposed separately [11–13]. To orchestrate the operation of each segment, higher-tier SDN controllers are introduced on the top of SDN controllers in each segment to support vehicular services. To facilitate the decision making at different tiers of SDN controllers, different levels of abstracted information regarding the network status are needed, such as vehicle geographical density, location-dependent content popularity, and the number of active vehicular users in a local area.

Vehicular services should not interfere with the legacy services in different segments. To this end, network slicing is performed in each segment to partition the whole network resource into various slices for different services, whereby those slices operate in an isolated manner and do not interfere with each other. To accomplish this goal, hypervisors are integrated into each segment [14]}. Specifically, the lower-level hypervisors at network components such as RSUs and SBSs schedule the local resources to different slices, where each slice can exclusively use the resources for a certain time period. The upper-level network hypervisor coordinates the lower-level hypervisors to perform network-wide resource allocation. As a result, sets of resources are allocated for legacy services. For instance, different sets of resources in satellite networks are allocated for Earth observation, navigation, and weather monitoring. Then the remaining resources from each segment are put into a space-air-ground resource pool for vehicular services. Note that network slicing is dynamically performed to achieve high resource utilization while supporting the time-varying needs of legacy and vehicular services.

Layered Structure

As shown in Fig. 2, the proposed SSAGV network is organized in three layers: infrastructure, control, and application layers. In the infra-

> As an emerging network architecture, SDN separates the control plane and data plane, introduces logically centralized control with a global view of the network, and facilitates network programmability/reconfiguration through open interfaces. With SDN, a dynamic, manageable, cost-effective, and adaptable network can be enabled.

[3] https://www.n2yo.com/satellites/?c=17, 2017.

[4] Legacy services refer to the existing services in different segments, rather than the vehicular services, e.g., earth observation and navigation in satellite networks.

[5] https://www.opennetworking.org/sdn-resources/sdn-definition, 2013

For partial control, the devices can have local intelligence. To efficiently use heterogenous network resources and conduct different levels of network functions, SDN controllers in different segments are coordinated by upper-tier controllers, through Eastbound and Westbound interfaces.

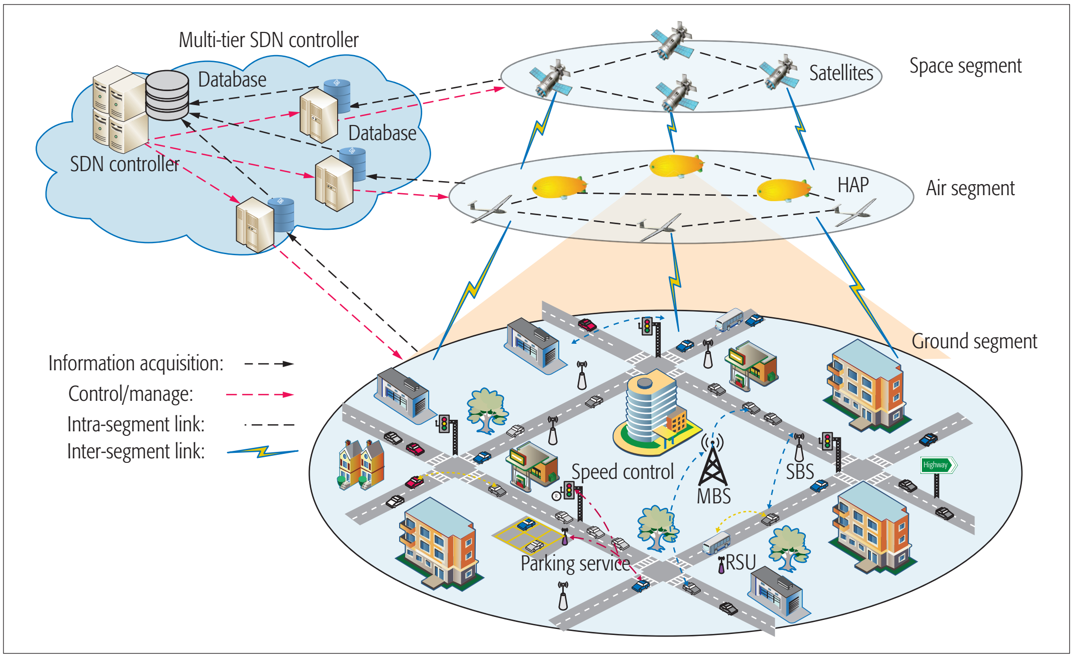


Figure 1. Software defined space-air-ground integrated vehicular (SSAGV) networks.

structure layer, computing, storage, and communication resources from different segments constitute a common resource pool. Since in each segment network slicing is performed dynamically, the space-air-ground resource pool varies with time. To deal with the dynamics in resource availability, resource virtualization can be adopted to abstract logical resource from the physical resources, and vehicular services can be deployed on a collection of virtualized resources, instead of specific physical resources. By dynamically performing virtual and physical resource mapping, disturbance to vehicular services can be alleviated.

In the control layer, for scalability, SDN controllers are organized in a hierarchical manner, targeted at network operation in different domains (i.e., local, regional, national, and global domains). Through southbound interfaces (SBIs), SDN controllers communicate and control the respective underlying physical resources. Considering the significant heterogeneity in different segments, the implementation of southbound interfaces will be specific to each segment, to facilitate various functions such as beam steering of satellites, movement control of UAV, and resource block allocation in LTE BSs. The implementation of SDN controllers should also address the scope and granularity of control (e.g., an SDN controller can fully or partially control the underlying network components. For partial control, the devices can have local intelligence. To efficiently use heterogenous network resources and conduct different levels of network functions, SDN controllers in different segments are coordinated by upper-tier controllers through eastbound and westbound interfaces.

In the application layer, a variety of vehicular services and network management functions are performed based on the functions provided by the control layer through the northbound interfaces (NBIs). The requests from an application can be translated into rules by NBIs for the SDN controllers, which are further interpreted into instructions to guide the underlaying devices through SBIs. Vehicular services correspond to the services provisioned directly to vehicular users, while network management functions facilitate efficient operation, such as mobility management, network hypervisor, data traffic classification, and data traffic engineering. Specifically, mobility management aims to provide seamless connectivity to moving vehicles by associating them with suitable access points. Data traffic classification associates data traffic flows with different vehicular services and categorizes them into different QoS classes by means of deep packet inspection (DPI) and machine learning. Based on classification, differentiated services can be enabled (e.g., through enforcing corresponding policies). Data traffic engineering can either steer data traffic to different segments and access points (e.g, RSUs and SBSs) for load balancing, or dynamically change the route to avoid congestion. Network hypervisors can help create multiple virtual networks coexisting over the common physical infrastructure by dynamically scheduling multi-dimensional resources. The virtual networks can be tailored to better support different vehicular services.

With the proposed SSAGV network architecture, the following benefits can be achieved:
- Simplified network management and cost-effective network upgrade/evolution
- Optimized network operation and resource utilization
- Flexible and agile network behavior control on the fly, along with adaptation to network dynamics such as topology changes and link failures

The SSAGV network can pave the way toward an open ecosystem with network agility and flexibility, and help achieve efficient interoperability, simplified operation and maintenance, and round-the-clock optimization of networks.

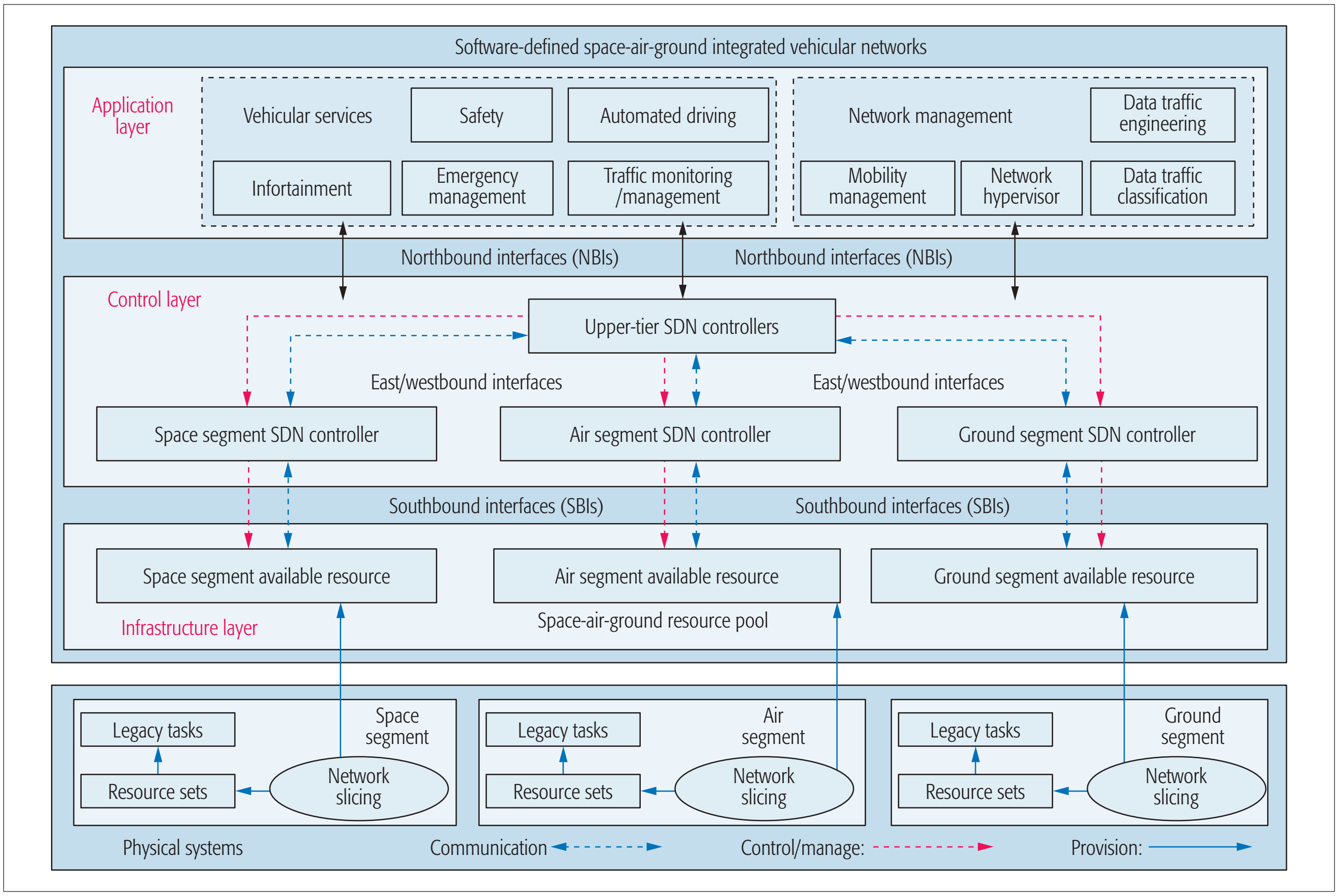

Software-defined space-air-ground integrated vehicular networks
Application layer
Vehicular services
Safety
Automated driving
Infortainment
Emergency management
Traffic monitoring /management
Network management
Data traffic engineering
Mobility management
Network hypervisor
Data traffic classification
Northbound interfaces (NBIs)
Northbound interfaces (NBIs)
Control layer
Upper-tier SDN controllers
East/westbound interfaces
East/westbound interfaces
Space segment SDN controller
Air segment SDN controller
Ground segment SDN controller
Southbound interfaces (SBIs)
Southbound interfaces (SBIs)
Space segment available resource
Air segment available resource
Ground segment available resource
Space-air-ground resource pool
Infrastructure layer
Legacy tasks
Space segment
Resource sets
Network slicing
Legacy tasks
Air segment
Resource sets
Network slicing
Legacy tasks
Ground segment
Resource sets
Network slicing
Physical systems
Communication
Control/manage:
Provision:

Figure 2. Layered structure of SSAGV networks.

Network Operation

Working Relation

In the SSAGV network, three different parties arise, including infrastructure providers (InPs), vehicular service providers (SPs), and vehicular customers, as shown in Fig. 3a. InPs provide infrastructure resources to vehicular SPs, which then provide diverse services to vehicular customers. Based on the infrastructure types, InPs mainly encompass the space segment InPs such as Intelsat, air segment InPs like Google, and ground segment InPs such as cellular network operators. Vehicular SPs request resources from different InPs to support their vehicular users, while InPs lease their resources considering the payments from vehicular SPs and the requirements of their legacy services. When an InP fails to provision the required resources, coalitions can be formed to satisfy the requirements of vehicular SPs; for example, different owners of HAPs can negotiate and cooperate to provide the required capacity for a targeted area. To facilitate the interactions between vehicular SPs and InPs, brokers can be introduced, which receive resource requirements from vehicular SPs and negotiate for resources from multiple InPs. By doing so, resources among different systems can be shared dynamically on demand.

Hierarchical Network Operation

The SSAGV network operation is complex due to the large network scale, the wide range of services, the heterogenous devices/resources, and high dynamics in network states. Therefore, network operation is performed in a hierarchical manner. Specifically, in the spatial dimension, the network is divided into different domains, which are controlled/managed by the hierarchical SDN controllers correspondingly. Lower-tier SDN controllers manage the underlying network devices in a small area. The upper-tier SDN controllers cover a large area, and can coordinate multiple lower-tier SDN controllers to perform high-level network operations. Lower-tier SDN controllers perform fine-grained control, while the upper-tier SDN controllers conduct coarse-grained control. For instance, the former can perform power control and user scheduling for the underlying devices such as BSs and RSUs, while the latter can steer different amounts of vehicular data traffic to different segments, or adjust the ON-OFF modes of BSs and RSUs for energy saving, based on the density of vehicular users. Taking content delivery as another example, the upper-tier SDN controllers update the content cached in the servers or BSs in a specific area according to the time-varying content popularity, while the lower-tier SDN controllers manipulate the content delivery based on instantaneous users' requests.

Accordingly, in the temporal domain, the network operation is performed in a multi-timescale fashion, to deal with the network dynamics, such as time-varying resource availability and burst vehicular service requests. In a large timescale, upper-tier SDN controllers adjust their strategies, according to the high-level network status (e.g., spatial traffic distribution or content popularity). In a small timescale, the lower-tier SDN control-

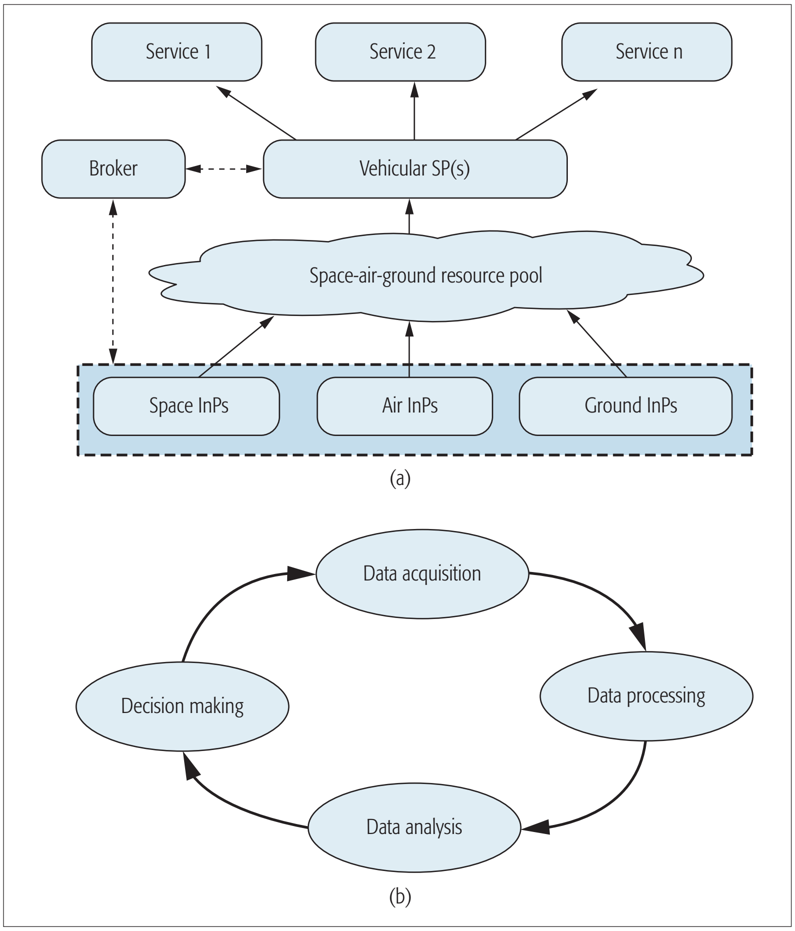


Figure 3. Working relation and big data-assisted operation: a) working relation; b) big data-assisted operation.

Number of satellites	6 in a plane
Low Earth orbit altitude	1414 km
Period	114 ~ 130 min
Inclination angle	0°
Altitude of HAP	20 km
Minimum elevation angle	10°
Radius of Earth	6371 km

Table 2. Software defined space-air-ground intergraded vehicular (SSAGV) networks.

lers update their strategies, adapting to instantaneous vehicular user requests. Within different timescales, the corresponding SDN controllers dynamically optimize their control policies based on the abstracted network information. Note that the information provided to various tiers of SDN controllers for decision making are different in terms of spatial and temporal scales. For instance, the density of vehicular users can be required for upper-tier SDN controllers, while the detailed location information of vehicular users are needed at lower-tier SDN controllers.

Big-Data-Assisted Operation

The decision making of SDN controllers relies on information collection and analysis. Considering the scale and volume of information, big data techniques can be adopted [15]. Through big data analytics, insights can be extracted to guide the decisions of SDN controllers. For instance, data traffic exhibits strong correlative and statistical features in the temporal and spatial domains. By extracting the data distribution features, network resources can be better provisioned. The big-data-assisted operation cycle is shown in Fig. 3b. Specifically, to acquire vehicle related information, in the ground segment different approaches can be employed to collect road data, including roadside cameras, electromagnetic transducers, and acoustic sensors; while in the space or air segments, aerial photography and videos of road information can be collected. After the data collection phase, raw data has to be processed before transmission, such as data compression and data fusion. For instance, the satellite with onboard processing (OBP) can extract useful information before transmission. Then data mining and machine learning techniques can be employed in the data analysis phase in order to obtain the knowledge about the network states. In addition, accurate prediction, such as for spatial-temporal traffic distribution, content popularity, and user mobility, can facilitate optimal decision making. With big data and SDN, intelligent and automated network operation can be achieved.

A Case Study

In this section, a case study is provided on SDN enabled coordination of satellite and HAPs, aimed at delivering contents to vehicular users efficiently.

Network Setting

Suppose that direct optical links from satellite to the ground is poor due to weather conditions, and HAPs can be utilized as relays. Each HAP is able to connect one satellite through the free-space optical link and connect vehicular users on the ground using microwave links, while a satellite with multiple beams can communicate with multiple HAPs. Consider a LEO satellite network of the Globalstar system, which operates at a height of 1414 km, and consists of 48 satellites in 8 planes with different inclination angles. HAPs, at the altitude of 20 km, are uniformly distributed in the air. HAPs stay relatively stable to the Earth, while the LEO satellites spin around with periods less than 130 min. The parameters are summarized in Table 2. Note that the minimum elevation angle for communications is set to 10°.

Simulation Results

We aim to maximize the average signal strength at HAPs (equivalent to maximizing the aggregate throughput) by coordinating the connections between HAPs and satellites. The signal strength mainly depends on the elevation angle. Thanks to the control channel, the SDN controller has real-time spatial information of both satellites and HAPs, which can be exploited to make round-the-clock optimal connection decisions. Specifically, the connection scheduling problem is a bipartite one-to-many matching between satellites and HAPs; and the SDN enabled coordination scheme determines one-to-many matching between the satellites and

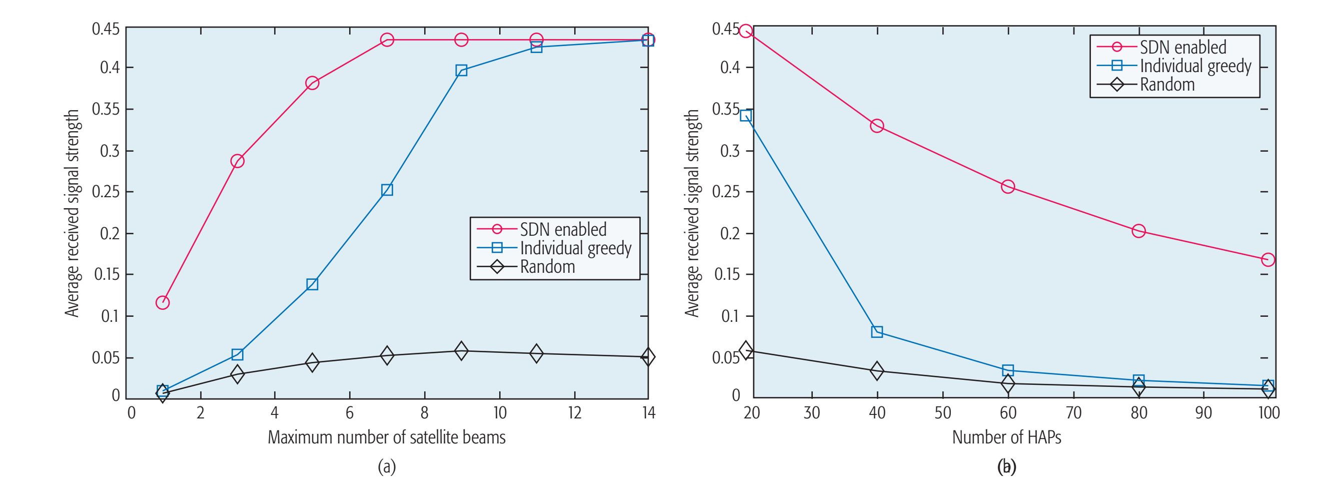


Figure 4. Performance comparison in terms of average signal strength: a) average signal strength vs. the maximum number of satellite beams; b) average signal strength vs. the number of HAPs.

HAPs. In comparison, legacy systems without spatial information can only make myopic decisions by greedily connecting HAPs to satellite beams with a high elevation angle. Results of random link establishment are also presented as a benchmark.

Figure 4a shows the average signal strength with respect to the maximum number of the satellite beams when there are 6 satellites and 50 HAPs. It can be seen that the SDN-enabled coordination scheme always outperforms individual greedy and random link establishment schemes, and quickly converges to the optimal value. In the individual greedy scheme, each HAP chooses among the available satellites to attain better signal strength. The SDN enabled coordination facilitates a centralized control, and thus helps achieve round-the-clock optimum.

Figure 4b shows the average signal strength with respect to the number of HAPs when the maximum link of a satellite is set to 3. It can be seen that the SDN-enabled coordination scheme still outperforms the individual greedy and random link establishment schemes. Moreover, with an increase in number of HAPs, the average signal strength decreases, since more HAPs compete for the limited satellite links.

Satellite networks often contain multiple planes, and this study demonstrates the considerable performance gain for one plane. The gain obtained from SDN-enabled coordination in a practical network can be even more appealing.

Research Issues

In this section, some open research issues are discussed for the SSAGV network.

Software Defined Platform Implementation

To exploit the potential benefits of the SSAGV network, a fundamental issue lies in the implementation of the SDN paradigm, including the data plane, the control plane, as well as different open interfaces. First, the scope and granularity of control for SDN controllers should be carefully devised to balance the performance and complexity. Second, since this paradigm spans various segments, the distinct features of each segment should be taken into account, such as the respective control and communication interfaces. Third, the deployment of the hierarchical SDN controllers largely affects the network performance, and it needs further investigation.

QoS-Aware Resource Allocation

With different segments integrated in the unified platform, efficient resource allocation becomes a significant challenge to support various vehicular services due to the highly dynamic network environment and multi-dimensional heterogeneity in resources and services. When performing resource allocation, the characteristics of different segments should be considered, such as the predicted trajectory of satellites and the controllable movement of HAPs. In such a case, a time-evolving resource graph (TERG) can be utilized to describe the evolution of satellite and aerial network resources. The TERG is built on a unified two-dimensional time-space basis, where vertices correspond to network elements such as satellites and HAPs, and edges denote the availability of different resources. The TERG can be further combined with available resources at the ground to dynamically support vehicular services.

Interaction among Different Parties

A vehicular SP can lease resources from different InPs for vehicular services, where InPs and vehicular SPs have different interests. For instance, the InPs aim to maximize the revenue from the vehicular SP without degrading their own legacy services, whereas the SP attempts to support vehicular services cost-effectively by exploiting different resources from InPs. The interaction between the InPs and the SP greatly affects the service performance, which can be studied through game theory. For example, the interactive procedure of price negotiation and resource dispatch can be modeled and analyzed by non-cooperative games. Furthermore, in the scenarios with multiple InPs and multiple vehicular SPs, the auction theory can be employed to analyze their interactions.

The new network operation model can pave the road for resource sharing and collaboration among different segments. To accelerate the pace of SSAGV network development, extensive research efforts are required in the outlined research directions.

Network Virtualization

To better support multifarious vehicular services/scenarios, network virtualization can be employed to create service-oriented virtual networks. Based on the service requirements, the virtual networks can be customized, including the virtual network topology and end-to-end protocols. Moreover, virtual networks can have stable virtual network topology by dynamically embedding the virtual nodes to the physical nodes, helping mitigate the adverse effects caused by the time-varying physical topology. To facilitate network virtualization, optimal virtual network topology, dynamic network embedding, and customized end-to-end protocols need extensive investigation.

Network Security

Since SDN controllers are responsible for managing resources and controlling network operation, it is imperative to protect the SDN controllers from different cyber attacks. For instance, denial-of-service (DoS) attacks can be launched to paralyze the operations of SDN controllers, or SDN controllers can be compromised through inside attacks. In addition, a variety of vehicular services require different security goals in terms of confidentiality, integrity, and authentication. Those requirements translate into various quality of protection parameters such as cryptographic variables and encryption key lengths. To better protect the network and vehicular services, the security aspect of SSAGV networks should be investigated.

Conclusion

In this article, we have proposed an SSAGV network architecture to exploit the advantages of space, air, and ground segments,to support diverse vehicular services in various scenarios efficiently and cost-effectively. The proposed open network architecture can achieve network agility and flexibility, simplify network management and maintenance, and adapt to changing user demands and network states. The new network operation model can pave the road for resource sharing and collaboration among different segments. To accelerate the pace of SSAGV network development, extensive research efforts are required in the outlined research directions.

Acknowledgments

This work is sponsored in part by the Nature Science Foundation of China No. 91638204 and the Natural Sciences and Engineering Research Council of Canada.

References

[1] N. Lu *et al.*, “Connected Vehicles: Solutions and Challenges,” *IEEE Internet of Things J.*, vol. 1, no. 4, 2014, pp. 289–99.

[2] K. Abboud, H. Omar, and W. Zhuang, “Interworking of DSRC and Cellular Network Technologies for V2X Communications: A Survey,” *IEEE Trans. Vehic. Tech.*, 2016, DOI: 10.1109/TVT.2016.2591558.

[3] N. Cheng *et al.*, “Opportunistic WiFi Offloading in Vehicular Environment: A Game-Theory Approach,” *IEEE Trans. Intelligent Transportation Systems*, vol. 17, no. 7, 2016, pp. 1944–55.

[4] S. Schwarz and M. Rupp, “Society in Motion: Challenges for LTE and Beyond Mobile Communications,” *IEEE Commun. Mag.*, vol. 54, no. 5, May 2016, pp. 76–83.

[5] M. Amadeo *et al.*, “A Satellite-LTE Network with Delay-Tolerant Capabilities: Design and Performance Evaluation,” *Proc. IEEE VTC-Fall*, 2011.

[6] Y. Zhou *et al.*, “Multi-UAV-Aided Networks: Aerial-Ground Cooperative Vehicular Networking Architecture,” *IEEE Vehic. Tech. Mag.*, vol. 10, no. 4, 2015, pp. 36–44.

[7] H. Kim and N. Feamster, “Improving Network Management with Software Defined Networking,” *IEEE Commun. Mag.*, vol. 51, no. 2, Feb. 2013, pp. 114–19.

[8] A. Gharanjik *et al.*, “Multiple Gateway Transmit Diversity in Q/V Band Feeder Links,” *IEEE Trans. Commun.*, vol. 63, no. 3, 2015, pp. 916–26.

[9] Y. Kawamoto *et al.*, “Prospects and Challenges of Context-Aware Multimedia Content Delivery in Cooperative Satellite and Terrestrial Networks,” *IEEE Commun. Mag.*, vol. 52, no. 6, June 2014, pp. 55–61.

[10] D. Lopez-Perez *et al.*, “Enhanced Intercell Interference Coordination Challenges in Heterogeneous Networks,” *IEEE Wireless Commun.*, vol. 18, no. 3, June 201?, pp. 22–301.

[11] L. Bertaux *et al.*, “Software Defined Networking and Virtualization for Broadband Satellite Networks,” *IEEE Commun. Mag.*, vol. 53, no. 3, Mar. 2015, pp. 54–60.

[12] H. Iqbal *et al.*, “A Software-Defined Networking Architecture for Aerial Network Optimization,” *Proc. IEEE NetSoft*, 2016.

[13] K. Pentikousis, Y. Wang, and W. Hu, “Mobileflow: Toward Software-Defined Mobile Networks,” *IEEE Commun. Mag.*, vol. 51, no. 7, July 2013, pp. 44–53.

[14] I. F. Akyildiz, P. Wang, and S.-C. Lin, “SoftAir: A Software Defined Networking Architecture for 5G Wireless Systems,” *Computer Networks*, vol. 85, 2015, pp. 1–18.

[15] Z. Su, Q. Xu, and Q. Qi, “Big Data in Mobile Social Networks: A QoE-Oriented Framework,” *IEEE Network*, vol. 30, no. 1, Jan./Feb. 2016, pp. 52–57.

Biographies

Ning Zhang [S’12,M’16] earned his Ph.D degree from the University of Waterloo, Ontario, Canada, in 2015. He received his B.Sc. degree from Beijing Jiaotong University and his M.Sc. degree from Beijing University of Posts and Telecommunications, China, in 2007 and 2010, respectively. From May 2015 to April 2016, he was a postdoctoral research fellow in the BBCR lab at the University of Waterloo. He is now a postdoctoral research fellow at the University of Toronto. He is now an Associate Editor of the *International Journal of Vehicle Information and Communication Systems* and a Guest Editor of *Mobile Information Systems*. He was the recipient of the Best Paper Award at IEEE GLOBECOM 2014 and IEEE WCSP 2015. His current research interests include next generation wireless networks, software defined networking, network virtualization, and physical layer security.

Shan Zhang received her Ph.D. degree from the Department of Electronic Engineering at Tsinghua University and her B.S. degree from the Department of Information from Beijing Institute of Technology, China, in 2016 and 2011, respectively. She is currently a postdoctoral fellow in the Department of Electrical and Computer Engineering, University of Waterloo. Her research interests include resource and traffic management for green communication, intelligent vehicular networking, and software defined networking. She received the Best Paper Award at the Asia-Pacific Conference on Communication in 2013.

Peng Yang received his B.Sc. degree from the Department of Electronics and Information Engineering, Huazhong University of Science and Technology (HUST), Wuhan, China, in 2013. Currently, he is pursuing his Ph.D. degree in the School of Electronic Information and Communications, HUST. Since September 2015, he is also a visiting Ph.D. student in the Department of Electrical and Computer Engineering, University of Waterloo. His current research interests include next generation wireless networking, software defined networking and fog computing.

Omar Alhussein [S’14] received his B.Sc. degree in communications engineering from Khalifa University, Abu Dhabi, United Arab Emirates, in 2013 and his M.A.Sc. degree in engineering science from Simon Fraser University, Burnaby, British Columbia, Canada, in 2015. He is currently working toward a Ph.D. degree in the Broadband Communications Research Laboratory, University of Waterloo. From January 2014 to May 2014, he was a research assistant with the Etisalat BT Innovation Centre, Khalifa University. From May 2014 to September 2015, he was with the Multimedia Communications Laboratory, Simon Fraser University. His research interests span software defined networking, network virtualization, wireless communications, and machine learning. He currently serves as a reviewer for *IEEE Communications Letters*, *IEEE Transactions on Vehicular Technology*, and other journals and conferences.

Weihua Zhuang [M'93, SM'01, F'08] has been with the Department of Electrical and Computer Engineering, University of Waterloo since 1993, where she is a professor and a Tier I Canada Research Chair in Wireless Communication Networks. Her current research focuses on resource allocation and QoS provisioning in wireless networks, and on smart grid. She is a co-recipient of several best paper awards from IEEE conferences. She was the Editor-in-Chief of *IEEE Transactions on Vehicular Technology* (2007–2013), and TPC Co-Chair of IEEE VTC-Fall 2016. She is a Fellow of the Canadian Academy of Engineering, a Fellow of the Engineering Institute of Canada, and an elected member of the Board of Governors and VP Publications of the IEEE Vehicular Technology Society.

Xuemin (Sherman) Shen [M'97, SM'02, F'09] received his B.Sc. (1982) degree from Dalian Maritime University, China, and his M.Sc. (1987) and Ph.D. (1990) degrees from Rutgers University, New Jersey, all in electrical engineering. He is a professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo. He was the associate chair for Graduate Studies from 2004 to 2008. His research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He served as the Technical Program Committee Chair/Co-Chair for IEEE INFOCOM '14 and IEEE VTC-Fall '10, Symposia Chair for IEEE ICC '10, Tutorial Chair for IEEE VTC-Spring '11 and IEEE ICC '08, Technical Program Committee Chair for IEEE GLOBECOM '07, General Co-Chair for Chinacom '07 and QShine '06, Chair of the IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He also serves/has served as Editor-in-Chief of *IEEE Network*, *Peer-to-Peer Networking and Application*, and *IET Communications*; a Founding Area Editor for *IEEE Transactions on Wireless Communications*; an Associate Editor for *IEEE Transactions on Vehicular Technology*, *Computer Networks*, and *ACM/Wireless Networks*, among others; and a Guest Editor for several publications. He received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004, 2007, and 2010 from the University of Waterloo, the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.

Toward Secure Software Defined Vehicular Networks: Taxonomy, Requirements, and Open Issues

Adnan Akhunzada and Muhammad Khurram Khan

This article contributes by presenting the security implications of emerging SDVNs to devise comprehensive thematic core layered taxonomies together with external communication APIs. Moreover, we also describe the potential requirements and key enablers toward secure SDVNs.

ABSTRACT

The emerging software defined vehicular networking (SDVN) paradigm promises to dramatically simplify network management and enable innovation through network programmability. Despite noticeable advances of SDNs in wired networks, it is also becoming an indispensable component that potentially provides flexible and well managed next-generation wireless networks, gaining massive attention from both industry and academia. In spite of all the hype surrounding emerging SDVNs, exploiting its full potential is demanding, and security is still the key concern and an equally arresting challenge. On the contrary, the complete transformation of the network into an SDN structure is still questionable, and the security and dependability of SDNs have largely been neglected topics. Moreover, the logical centralization of network intelligence and the tremendously evolving landscape of digital threats and cyber attacks that predominantly target emerging SDVNs will have even more devastating effects than they are in simple networks. Besides, the deployment of the SDVNs' novel entities and several architectural components drive new security threats and vulnerabilities. Since the SDVNs architectural layers and their corresponding APIs are heavily dependent on each other, this article aims to present a systematic top-down approach to tackle the potential security vulnerabilities, attacks, and challenges pertaining to each layer. The article contributes by presenting the security implications of the emerging SDVNs to devise comprehensive thematic core layered taxonomies together with external communication APIs. Moreover, we also describe the potential requirements and key enablers toward secure SDVNs. Finally, a plethora of open security research issues are presented that may be deemed appropriate for young researchers and professionals around the globe to tackle in anticipation of secure SDVNs.

INTRODUCTION

Vehicular networks have been envisioned to meet the imminent demands for improved transportation efficiency, road safety, reduced accidents, and avoiding and mitigating the overall impacts of heavily congested traffic. Vehicular networks are no longer a futuristic promise and can potentially provide surveillance services, safety traffic management services, and mobile vehicular cloud services. Moreover, there is increased demand for enabling access to the Internet for infotainment applications through IP-enabled smart mobile devices (SMDs), which is likely to attract a huge mass market and several key players within the networking industry. Among the many key players, the emergence of the promising software defined networking (SDN) paradigm has created great potential and hope to overcome the need for well managed, reliable, and flexible emerging software defined vehicular networks (SDVNs). Primarily, the revolutionary concept of SDNs has largely manifested in wired networks. However, SDN is now becoming an integral and indispensable component of many wireless access networks such as large-scale vehicular networks and intelligent transportation systems (ITS), and has currently gained considerable attention from both academia and industry [1]. Indeed, all the hype surrounding SDNVs is predominantly because of its centralized control, the separation of the control plane from the data forwarding plane, and enabling innovation through network programmability. Noticeably, the remarkable features of SDN provide a more vendor-agnostic, programmable, cost-effective, and innovative SDVN environment. However, academic and industrial observers are still apprehensive about the security of SDVNs. Moreover, the security and dependability of SDNs have largely been neglected topics and open issues.

Despite the promising architecture of SDVNs and centralized control intelligence having substantial managerial benefits [1, 2], unlike traditional vehicular networks, SDVNs' security and integrity remain unproven when it comes to centralized network intelligence. Moreover, the centralized placement of the management functionality represents a single point of failure, and an SDVN's controller being compromised in any way would certainly throw the entire network into chaos [3]. Moreover, the abstraction of the underlying topologies, flows, SDVN agents, and hardware resources can help significantly in harvesting core intelligence to launch thoughtful threats and diverse attacks. On the contrary, the

Digital Object Identifier:
10.1109/MCOM.2017.1601158

Adnan Akhunzada is with Comsats Institute of Information Technology; Muhammad Khurram Khan is with King Saud University.

0163-6804/17/$25.00 © 2017 IEEE

SDVN programmability aspect is also vulnerable to numerous malicious code exploits that can subsequently be used to generate massive attacks. Moreover, the application-to-control-plane and control-to-data-plane communication application programming interfaces (APIs) can also be targeted with diverse side-channel and denial of service (DoS) attacks. Besides, the various communications involved at the data plane are vulnerable to both active and passive eavesdropping [4]. Finally, the cyber attacks launched through various SDVN agents can have overwhelming effects compared to using simple vehicular networks.

To the best of our knowledge, this work is the first effort that systematically studies a way forward to secure SDVNs. Since SDVNs comprise a layered architecture, a security implication pertaining to any layer can severely affect other layers, which are heavily dependent. There is a dire need to present a layered-based top-down approach for SDVNs to systematically cater for security implications that must be double checked and attentively taken into consideration. Moreover, the establishment of trust throughout SDVNs with novel architectural components and agents is of grave concern. Dynamic and robust policy frameworks must be ensured throughout SDVNs. Keeping in view the imminent demands for improved large-scale vehicular networks and ITS, SDVNs undoubtedly necessitate a cost-effective, scalable, simple, and efficient secure system environment.

The contributions of the article are manifold. The article aims to present a systematic top-down approach to tackle the security vulnerabilities, attacks, and challenges of each layer, which is heavily dependent on anticipating secure emerging SDVNs. The article presents the critical areas of focus of each SDVN layer that must be raised on the agenda toward secure SDVNs. The article also contributes by presenting the main categories of security implications of the SDVN application layer to devise a comprehensive thematic taxonomy. The article also contributes by broadly presenting the security vulnerabilities, attacks, and challenges of the control plane to formulate a thematic taxonomy. We mainly classify the SDVN data layer into two broad categories, upper data plane and lower data plane, to discuss the potential security implications and challenges in order to devise corresponding thematic taxonomies. The comprehensive taxonomy of the external communication APIs are also presented. The potential security requirements with their key enablers of SDVNs are also highlighted and presented. Finally, we present open security issues and potential future research directions that need to be addressed to develop the emerging SDVNs.

The remainder of this article is organized as follows. We introduce a simplified overview of SDVN architecture to provide the fundamental background. We present the security vulnerabilities, attacks, and challenges of each layer using a top-down approach to anticipate secure SDVNs. We present thematic taxonomies for each layer together-with the external communication APIs. We discuss the requirements and key enablers for SDVN security. We highlight open issues in securing the SDVNs, and the article is concluded.

A Simplified View of Software Defined Vehicular Network Architecture

This section provides a brief fundamental discussion of SDNV architecture to better comprehend the security concerns with respect to SDNV architecture [1, 2]. SDVN is an emerging networking paradigm. The architecture of SDVNs mainly comprises the promising concept of SDN, which separates the control plane from the data plane and provides programming ability on the control plane [5] . Consequently, the most simplified view of the SDVN architecture mainly comprises three planes with their corresponding connected interfaces, as shown in Fig. 1. However, we further classify the data plane into two subsequent categories to give a clearer understanding of the SDVN architecture.

The application plane is also known as the application layer of the SDVN architecture. The application layer is meant to provide a set of services and applications [6]. The second most important component of the SDVN is the control plane, which is also known as the control layer of SDVNs. This is the central layer, which comprises the controller that is a software platform and represents the centralized core of network intelligence. It is a central decision point and the core of SDVN architecture. Moreover, the control plane facilitates the network's programmability as well as abstraction of the underlying resources. The overall data plane of the SDVN architecture mainly comprises the underlying network resources. The upper data plane of the SDVN comprises the forwarding hardware such as SDN-enabled switches and routers. This layer is also known as the infrastructure layer of the SDVN architecture. Since the control functionality is placed in the controller, the underlying hardware is only held responsible for forwarding. On the contrary, the lower data plane mainly comprises vehicular networks that are properly built and networked through vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communication. Moreover, the communication medium involves various types of wireless communication at the lower data plane, as shown clearly in Fig. 2. The vehicles act as end users and forwarding elements equipped with both onboard units (OBUs) and the OpenFlow protocol [7]. On the other hand, the roadsde units (RSUs) are also SDN-enabled and controlled by the SDN controller. A collection of RSUs is connected to the SDN-enabled RSU controller (RSUC) through broadband connections prior to accessing the SDN controller.

Consequently, the infrastructure layer implements the management functionality of the controller through SDN-enabled switches to forward the data and collect the network information, and sends it to the control layer. Finally, the application-to-control-plane communication is enabled through an API and is known as the northbound interface. The control-to-data-plane communication is also enabled through an API and is known as the southbound interface. The eastbound and westbound APIs are meant for back and forth communication between controllers in a distributed SDVN environment. It is also important to note that the wired extended part of the SDN structure

The application plane is also known as the application layer of the SDVNs architecture. The application layer is meant to provide a set of services and applications. The second most important component of the SDVNs is the control plane, which is also known as the control layer of SDVNs.

> Malicious SDVNs applications can manipulate the controller's internal storage shared among varied applications on the basis of certain privileges granted to access the underlying resources. Consequently, manipulating the internal database of a controller can be used for many attacks, such as manipulating network behavior.

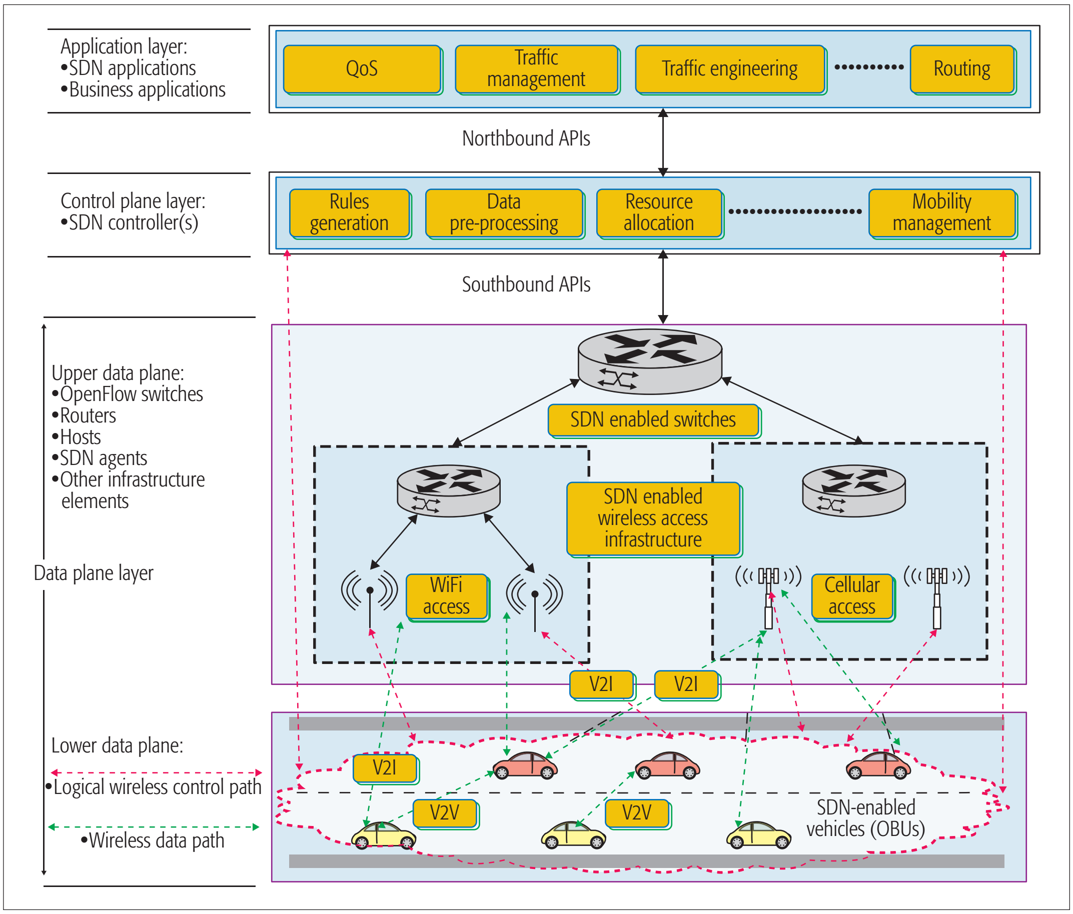


Figure 1. A simplified view of the SDVN architecture.

to SDVNs will bring almost no change, unless it largely transforms and shifts on wireless as shown in Fig. 1 and more clearly depicted in Fig. 2.

A Top-Down Approach to Addrss SDVN Security Vulnerabilities, Attacks, and Challenges

The SDVN architecture comprises three planes: the application plane, the control plane, and the data plane, with their corresponding APIs. However, the data plane is further sub-divided into two main categories, whichare the upper data plane and the lower data plane of the SDVN architecture. Figure 1 depicts a simplified view of the SDVN architecture and is used to systematically address the security vulnerabilities, attacks, and challenges of each layer using a top-down approach to anticipate secure SDVNs. Each layer is followed by their corresponding taxonomy to thoughtfully address and grasp the main areas of focus.

Software Defined Vehicular Networks: Aplication Layer

A set of SDVN applications are used in controlling the network with diverse network functions and can access the underlying network resources under certain privileges [8]. SDVN applications have many advantages, but they may cause serious security challenges. This section briefly but extensively explores the SDVN's application-plane-related security vulnerabilities, attacks, and challenges, which are mainly classified into four main categories:

1. Malign applications
2. Third-party applications
3. Application server compromise
4. Open development environments with their corresponding consequences

Malicious SDVN applications can manipulate the controller's internal storage shared among varied applications on the basis of certain privileges granted to access the underlying resources. Consequently, manipulating the internal database of a controller can be used for many attacks, such as manipulating network behavior [9] and subsequently getting authorized access, which is a very serious concern, particularly in SDVNs. Moreover, malicious applications can exclusively contribute to exhaust all the available system resources. Despite seriously affecting the performance of other applications, including the controller, it can exhaustively involve continuous consumption of a system's available memory. Moreover, by creating useless programing threads, it can cause severe CPU consumption, throwing the entire network into chaos [8]. Malign applications while dismissing a controller instance can lead to executing a system exit command. Malicious applications capable of executing system commands can simply lead to unauthorized access, which can eventually help launching diverse sophisticated attacks.

Malignant nested applications present a real challenge to deal with and are able to cause

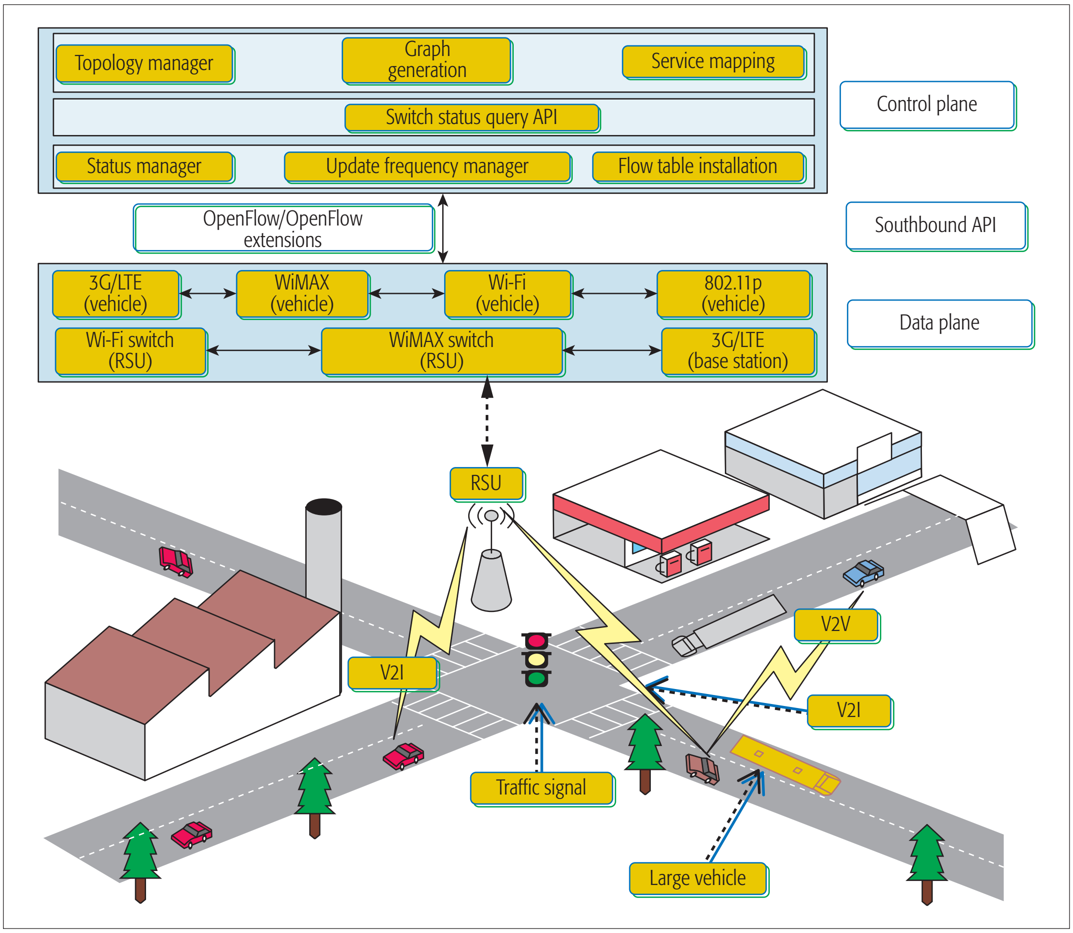


Figure 2. SDVNs with data-to-control-plane diverse communications.

Defense against diverse malicious applications will remain a challenge for SDVNs. Moreover, a compelling mechanism needs to be devised to thoroughly address the third-party or network service applications plus the accountability and access control of nested applications.

severe interference [10, 11]. Malicious applications that participate in a service chain can drop control messages before the awaited applications, thus causing extreme service chain interference. Furthermore, a malevolent nested application can sidestep the access control by issuing an instance of another class application and can be utilized as a gateway to illegal access. Moreover, authentication of nested applications is a major challenge in programmable networks, and the diversity of third-party applications makes this situation even more difficult.

An arbitrary control message issued by a malignant application may lead to flow rule modification, which is the process of overwriting an existing flow rule to transform required network behavior. Moreover, a malevolent application issuing a control message that clears the flow table entries can block the overall communication. Furthermore, manipulation of a control message by a malignant application may severely terminate the controller instances, throwing the entire network into chaos [12].

Third-party applications could also result in serious security vulnerabilities and challenges because of the lack of standard and consensus-based development environments, programming models, and paradigms, and the variety of vendors. Importantly, third-party applications could cause serious issues of interoperability and collision in security policies. Compromise of the application server, which stores sensitive user credentials, can add/remove forged but authorized flow to the network, leading to creating serious trust issues. Mechanisms to certify network devices exist, however; compelling mechanisms to establish trust to certify network applications have largely been a neglected area. Eventually, the diversity and multitude of third-party applications and non-standard open software development environments are extremely challenging.

The security vulnerabilities, attacks, and challenges of the application layer are extremely critical for the underlying SDVN layer. Defense against diverse malicious applications will remain a challenge for SDVNs. Moreover, a compelling mechanism needs to be devised to thoroughly address the third-party or network service applications plus the accountability and access control of nested applications.

Software Defined Vehicular Network Control Layer

The distinguishing property of SDVNs is centralized control network intelligence [3]. Despite significant managerial benefits, the following are some of the potential SDVN control layer security vulnerabilities, attacks, and challenges.

Unlike traditional hardware switches, the soft programmable switches running atop end host servers are easy to compromise. A compromised SDN-enabled switch may lead to poisoning the view of the controller by forging the Address Resolution Protocol (ARP) packet relayed as a packet-in message. Moreover, it can also lead to network topology poisoning by manipulating

The controller is the sole decision making entity and the core of centralized control network intelligence. A 10 Gb/s link high-speed network can lead the controller to become a bottleneck. Consequently, it can be targeted with saturation attacks, and is a more likely venue to attack as it also represents the single point of failure.

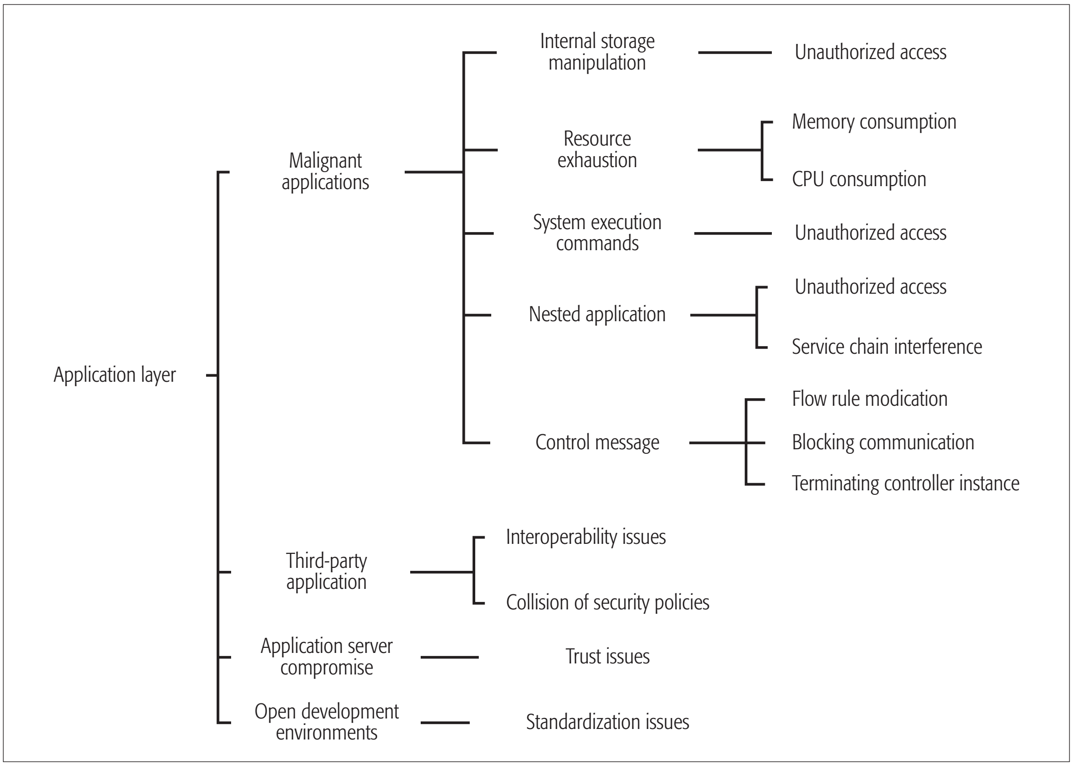


Figure 3. Taxonomy of SDVN application layer security vulnerabilities, attacks, and challenges.

the link discovery service relayed as packet-in messages and can consequently help create fake topologies. Moreover, an LLDP packet relayed as a packet-in message may also be forged to create fabricated links [12]. On the contrary, control message manipulation can affect the SDVN controller in many ways. It can leverage side-channel attacks to obtain sensitive information and may spoof a target switch to launch a switch-table flooding attack to ultimately put the controller in an unpredictable state. A real challenge is enabling the controller to properly facilitate the authentication and authorization of network resources consumed by applications implemented on top of the control plane with appropriate tracking, auditing, and isolation. Neglecting these severe issues simply leads to unauthorized access, which can eventually compromise the system or the controller to take the entire network down. A controller can potentially be targeted with flooding attacks such as denial of service (DoS), directed DoS, and distributed DoS (DDoS).

The controller is the sole decision making entity and the core of centralized control network intelligence. A 10 Gb/s link high-speed network can lead to the controller becoming a bottleneck. Consequently, it can be targeted with saturation attacks, and is a more likely venue to attack as it also represents a single point of failure. The visibility features of SDVNs of the underlying resource abstraction can better help in harvesting network intelligence that can subsequently be used for diverse exploitation. Since the control plane enforces a network-wide policy, the single-domain multiple controllers, multi-tenant SDN controllers, and multiple OpenFlow architectures may lead to serious configuration conflicts and inter-federated configuration conflicts.

Software Defined Vehicular Networks External Comunication APIs

The external communication APIs (northbound, southbound, eastbound, and westbound) are considered highly important targets for exploitation and have direct implications on the overall underlying SDVN architecture. Following are the security vulnerabilities, attacks, and challenges of diverse external communication APIs. The northbound bound SDVN APIs currently faces two mainstream challenges:

1. The lack of standardized northbound APIs for vehicular networks may pose severe threats and attacks by skilled adversaries.
2. A poorly designed northbound API for vehicular networks can easily be exploited to disrupt other ongoing application sessions or may unsubscribe an application to sensitive control messages by simply throwing an unsubscription event listener.

Consequently, it will seriously affect the underlying SDVN layer.

The control layer of SDVNs will always remain a favorable choice for attackers. Currently, the de facto standard southbound API is OpenFlow, which is enabled with optional transport layer security (TLS) for control channel security. However, TLS does not implement TCP-level protection and is not reliable [12, 13] in many cases. Consequently, OpenFlow is prone to TCP-level attacks. Moreover, the southbound API (OpenFlow) is also prone to diverse man-in-the-middle attacks. The southbound APIs can be targeted for both passive and active eavesdropping to obtain sensitive information that can subsequently cause severe attacks. The southbound APIs can be exploited with availability-related attacks that mainly refer to DoS and DDoS attacks such as communication flooding attacks. Despite Open-

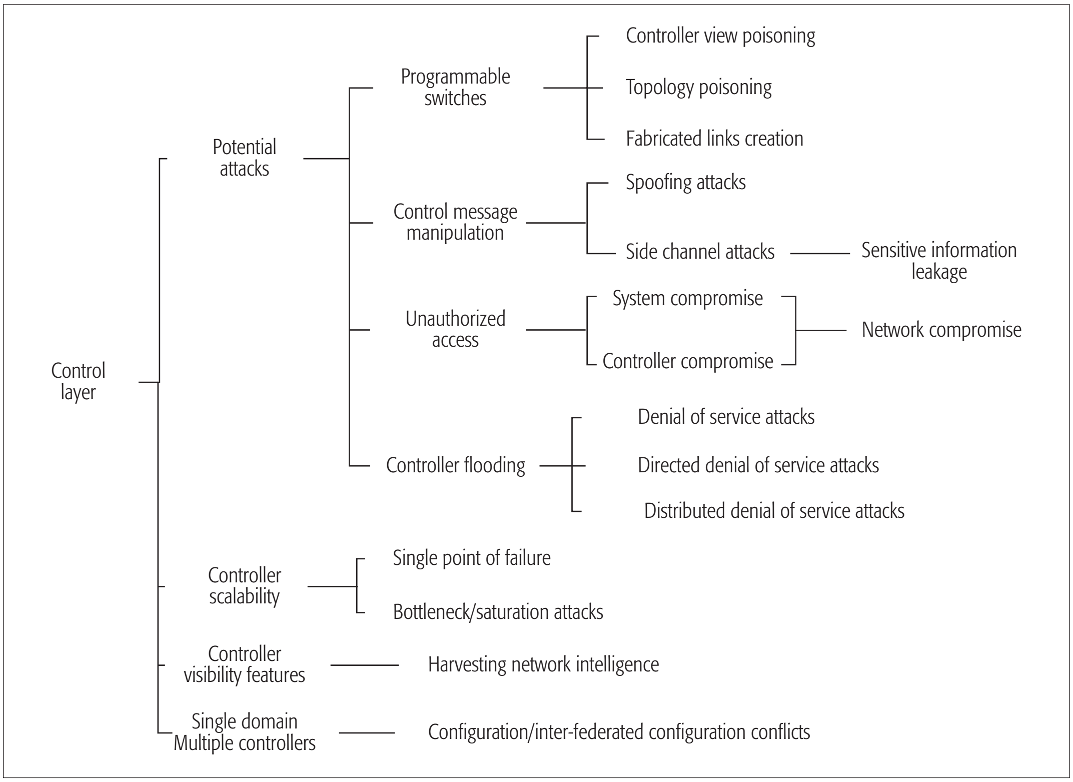

Control layer
Potential attacks
Programmable switches
Controller view poisoning
Topology poisoning
Fabricated links creation
Control message manipulation
Spoofing attacks
Side channel attacks
Sensitive information leakage
Unauthorized access
System compromise
Controller compromise
Network compromise
Controller flooding
Denial of service attacks
Directed denial of service attacks
Distributed denial of service attacks
Controller scalability
Single point of failure
Bottleneck/saturation attacks
Controller visibility features
Harvesting network intelligence
Single domain Multiple controllers
Configuration/inter-federated configuration conflicts

Figure 4. Taxonomy of SDVNs control layer security vulnerabilities, attacks, and challenges.

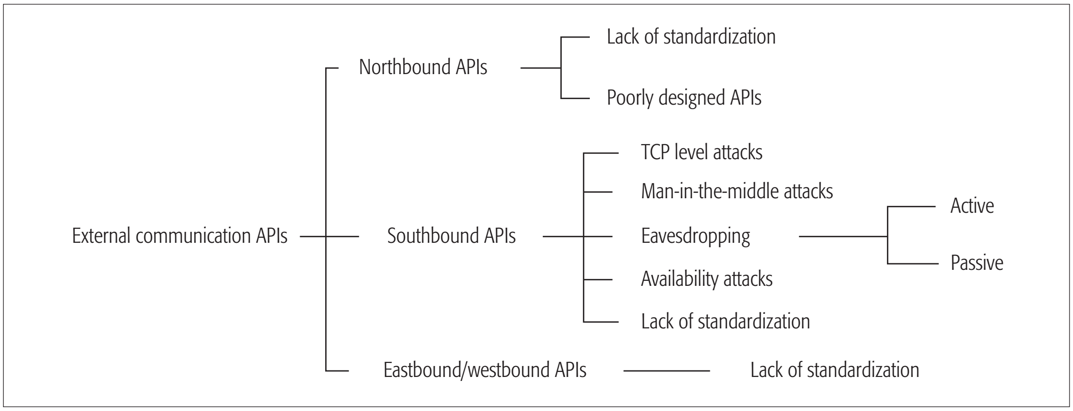


Figure 5. Taxonomy of SDVNs external communication APIs security vulnerabilities, attacks, and challenges.

> Currently, there is lack of standardized eastbound/westbound SDVNs APIs. A compelling and consensus-based eastbound/westbound APIs are needed to address the security vulnerabilities, attacks, and challenges of the corresponding interfaces thoroughly and uniformly.

Flow as a de facto standard, SDVNs lack customized OpenFlow APIs and also face standardization issues. Currently, there is a lack of standardized eastbound/westbound SDVN APIs. Compelling and consensus-based eastbound/westbound APIs are needed to address the security vulnerabilities, attacks, and challenges of the corresponding interfaces thoroughly and uniformly.

Software Defined Vehicular Networks Data Layer (Upper Data Plane)

The data layer comprises SDN-enabled switches and routers that are just dump forwarding entities controlled by the SDVN controller. The potential security vulnerabilities, attacks, and challenges of the upper data plane of the data layer are as follows.

Compromised SDN agents such as hosts and a soft programmable switch can contribute to a wide variety of attacks. This mainly includes the advertisement of fake topologies (logical or physical) by filling up the compromised target switch flow table. Moreover, a compromised host/SDN switch may also trigger dynamic attacks, such as traffic rerouting, traffic hijacking, and network DoS attacks. Furthermore, a compromised SDN agent can manipulate a control message to render the data plane practically offline and can launch a controller DoS attack to place the control plane in an unpredictable state. Compromised SDN-enabled agents can also lead to a variety of man-at-the-end (MATE) attacks.

Flooding a large number of flow entries to exhaust the switch-limited resources can also place the data plane in an unpredictable state. Finally, the data layer can also leverage side-channel attacks to obtain sensitive information. For example, an input buffer can be used to identify flow rules and analyze the packet processing time that may determine the forwarding policy. Hard-

An input buffer can be used to identify flow rules, and analyzing the packet processing time that may determine the forwarding policy. Hardware security is a very important aspect to be considered while securing the data layer. Moreover, it is reported that certain SDN-enabled switch hardware tables cannot process crafted flow rules.

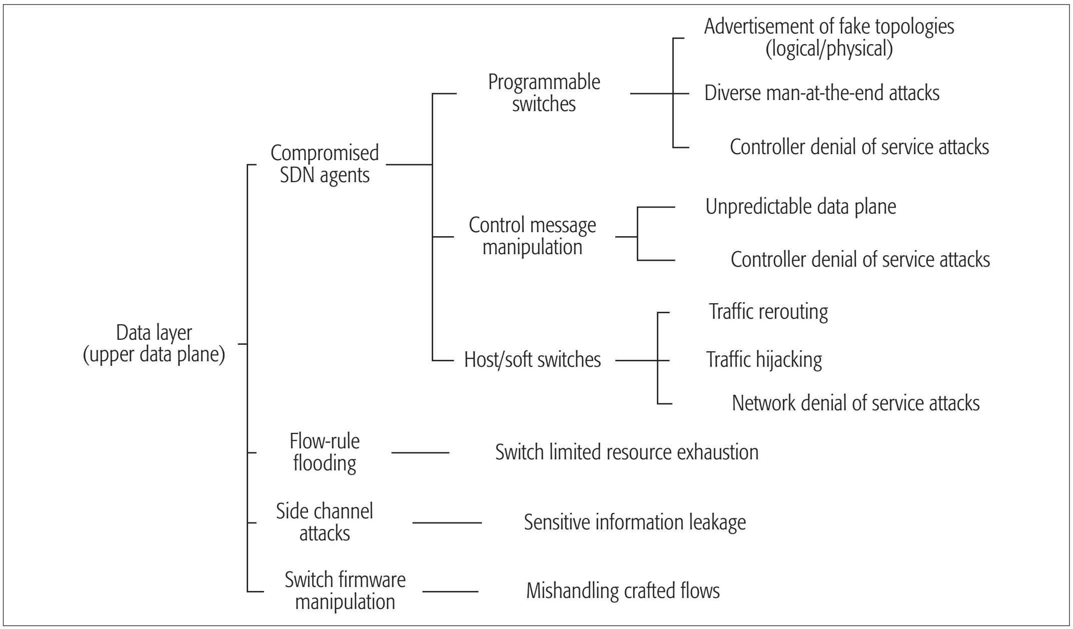

Data layer (upper data plane)
Compromised SDN agents
Programmable switches
Advertisement of fake topologies (logical/physical)
Diverse man-at-the-end attacks
Controller denial of service attacks
Control message manipulation
Unpredictable data plane
Controller denial of service attacks
Host/soft switches
Traffic rerouting
Traffic hijacking
Network denial of service attacks
Flow-rule flooding
Switch limited resource exhaustion
Side channel attacks
Sensitive information leakage
Switch firmware manipulation
Mishandling crafted flows

Figure 6. Taxonomy of the SDVN data layer (upper data plane) security vulnerabilities, attacks, and challenges.

ware security is a very important aspect to consider while securing the data layer. Moreover, it is reported that certain SDN-enabled switch hardware tables cannot process crafted flow rules.

Software Defined Vehicular Networks Data Layer (Lower Data Plane)

The SDVN (lower data plane) comprises diverse entities and SDN-enabled agents, which involves varied standards of wireless communications. Moreover, the newly deployed infrastructure entities and architectural components of SDVNs are also critical from the security point of view. From the vehicular dynamic spectrum access and communication perspective, the article presents the software defined radio (SDR) technology potential security vulnerabilities, attacks, and challenges in SDVNs, which is the most promising and flexible approach for dynamic spectrum management (DSM) or dynamic spectrum access (DSA) and an indispensable architectural component of SDVNs and ITS [14] . The major challenge for wide deployment of SDR in SDVNs is that it requires an adequate level of security. SDRs are able to download new radio applications through various communication links having reconfiguration capability, which makes SDR vulnerable to malicious modifications despite the introduction of integrity checks. A malicious operating environment such as a compromised operating system (OS) or middleware may lead to altered configuration data, which subsequently helps to alter or obtain sensitive user data. Moreover, a malicious modification through a drive by downloads may also lead to an altered software framework that ultimately causes alteration of the operating system software. Consequently, SDR malicious modification can simply lead to data repudiation, which is a serious security concern.

SDR is also vulnerable to device cloning, a commonly used term in conventional wireless communications. Device cloning simply represents authorized access to different services provided by another SDR device. It is a serious security threat and will be even more devastating in SDVNs. Malicious insertions are also possible with SDR through various ways that may directly affect the software framework and consequently the alteration of the operating system software. A malicious insertion may directly cause a software failure and can also lead to illegal use of the provided SDR services. The cognitive control channel (CCC) is also vulnerable to various saturation attacks.

The heterogeneous environment at the lower data plane of the SDVN data layer can cause serious configuration conflicts and inter-federated configuration conflicts. Moreover, the high mobility and networking issues can cause severe delay and disruption in continuous security monitoring and road accidents. Finally, the rapidly dynamic topological changes and unstable wireless channels can also lead to grim security concerns. The data-to-control channels can be targeted with both active and passive eavesdropping to obtain sensitive information [4]. Any type of SDVN, regardless of its design and architecture, relies on people and end users. MATE attacks [15] are fundamentally difficult to resolve under general circumstances, and the problematic part is that humans have become the edge, and auditing the human mind is a complex task. MATE will be an extremely serious concern to tackle, particularly when most of the devices are programmable and easily accessible in open environments. The attack can take the data plane practically offline and can also target the control plane with diverse attacks.

Requirements and Key Enablers for SDVN Security

To build a secure SDVN environment, it is vital to ensure the security of each and every component of the SDVN. Following are some of the essential security requirements for securing the key SDN components.

Securing the SDVN controller is of prime concern as it is responsible for the overall management of the network. Moreover, the controller is a

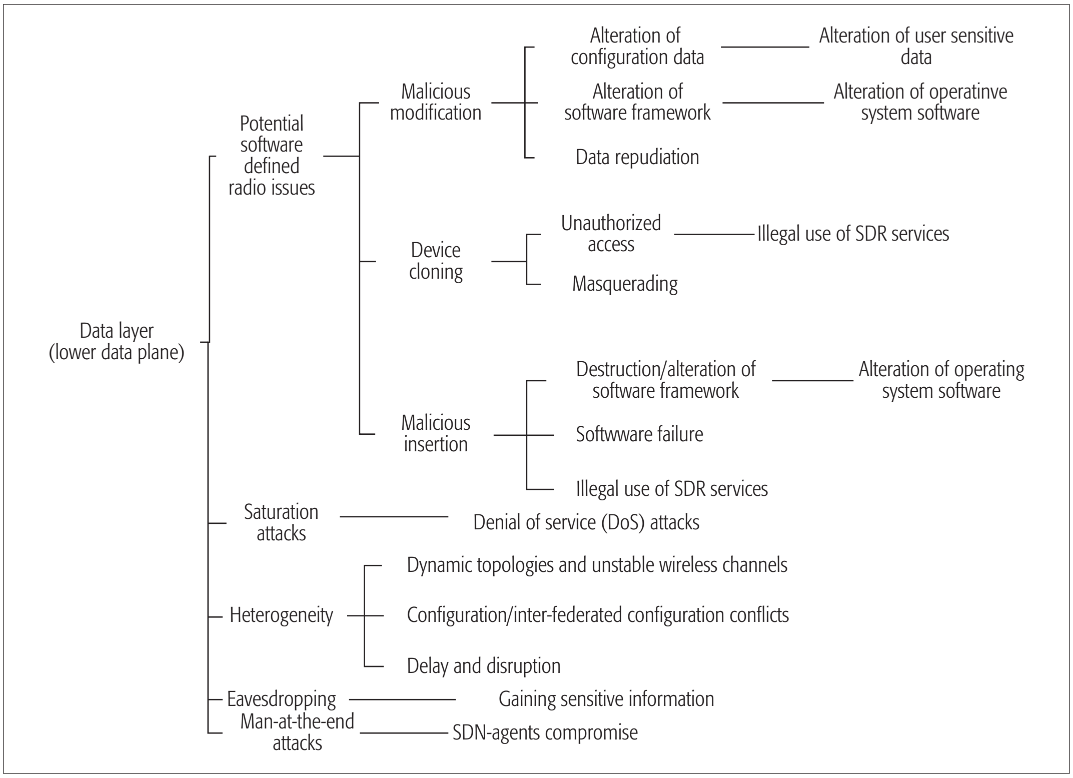

Data layer (lower data plane)
Potential software defined radio issues
Malicious modification
Alteration of configuration data
Alteration of user sensitive data
Alteration of software framework
Alteration of operatinve system software
Data repudiation
Device cloning
Unauthorized access
Illegal use of SDR services
Masquerading
Malicious insertion
Destruction/alteration of software framework
Alteration of operating system software
Softwware failure
Illegal use of SDR services
Saturation attacks
Denial of service (DoS) attacks
Heterogeneity
Dynamic topologies and unstable wireless channels
Configuration/inter-federated configuration conflicts
Delay and disruption
Eavesdropping
Gaining sensitive information
Man-at-the-end attacks
SDN-agents compromise

Figure 7. Taxonomy of the SDVN data layer (lower data plane) security vulnerabilities, attacks, and challenges.

The availability of the SDVNs controller must be ensured through an operating system having no patches and backdoor accounts, vulnerable open ports, services and protocols. The flow paradigm of the SDVNs must be ensured with end-to-end communication security to avoid diverse attacks.

central decision point and a single point of failure in the central control mode of SDVNs. An SDVN controller compromised in any way can put the entire network into chaos. This key component needs defense in depth, which may include the protection of the system containing the SDN controller from physical attacks and cyber threats. The availability of the SDVN controller must be ensured through an operating system having no patches and back door accounts, vulnerable open ports, services, and protocols. The flow paradigm of the SDVNs must be ensured with end-to-end communication security to avoid diverse attacks.

The SDVN agent's security of upper and lower data planes is essentially important as it constitutes the environment. All SDVN agents at the data layer must be tightly physically secured as the environment becomes very open when it comes to the lower data plane. Moreover, most of the SDVN agents are programmable, which creates an even more opportunistic environment for skilled adversaries. SDVN agents require deploying the latest identity management, threat isolation, and mitigation techniques. The most crucial part is hardening the APIs involved in SDVNs. Exploiting poorly designed APIs has already occurred in the security research community. Hardening various SDVN APIs includes the practices of secure coding, deployment of integrity checks, and, most importantly, digital signing of the code.

Open Issues for Securing SDVNs

Security plays a crucial role and is a key obstacle that may hinder the radical growth and overall adoption of emerging SDVNs. This section briefly elaborates the outstanding security issues that must be raised on the agenda toward secure SDVNs.

Programmability Aspects and Open Development Environments

The SDVN has to cope with a potential set of complex problems related to the programmability aspects [12]. The tendency of launching sophisticated phishing, DDoS, malware, spam attacks, and eavesdropping (active, passive) are expected to increase massively. Subsequently, it will change the dynamics of SDVNs. The lack of standardization of various SDVN APIs, poorly designed APIs, and the prevalent open development environments can create opportunities for skilled adversaries to launch severe threats and attacks on various layers of SDVNs. The large cyber-space of SDVNs with diverse programmable devices will tremendously evolve the landscape of digital threats and sophisticated cyber attacks [10, 12].

Security and Dependability on SDNs

First, the complete transformation of the vehicular network into an SDN structure is still questionable. Merely relying on the wired extended part of the SDN structure in SDVNs will bring nearly no change unless it largely shifts to wireless. Second, the security and dependability of SDNs itself has largely been a neglected topic. There are a plethora of open security issues in SDNs [12] that need to be thoroughly taken into consideration toward SDVNs.

Conclusions

The emerging software defined vehicular network is imposing new requirements for network security due to the newly deployed infrastructural entities and architectural components. SDVN is a layered architecture where security implications pertaining to any layer can affect the other layers and are heavily dependent. In order to address meticulously the security issues of emerging SDVNs, a systematic top-down approach is presented as a way forward to secure SDVNs. The security implications of each layer together-with external communication APIs are detailed to systematically tackle the diverse security vulnerabilities, attacks, and challenges of SDVNs. The critical areas of focus of each SDVN layer that must be raised on the agenda toward emerging secure SDVNs are appropriately highlighted. The article provides a tutorial to anticipate secure emerging SDVNs. Moreover, the requirements for securing SDVNs are also identified and presented. Finally, we discuss the open research issues that may assist in wide acceptance of emerging SDVNs.

The lack of standardization of various SDVNs APIs, poorly designed APIs, and the prevalent open development environments can create opportunities for skilled adversaries to launch severe threats and attacks on various layers of the SDVNs. The large cyber-space of the SDVNs with diverse programmable devices will tremendously evolve the landscape of digital threats and sophisticated cyber-attacks.

Acknowledgments

The authors extend their sincere appreciation to the Deanship of Scientific Research at King Saud University for funding this Prolific Research Group (PRG-1436-16).

References

[1] Z. He, J. Cao, and X. Liu, "SDVN: Enabling Rapid Network Innovation for Heterogeneous Vehicular Communication," *IEEE Network*, vol. 30, no. 4, July–Aug. 2016, pp. 10–15. DOI: 10.1109/MNET.2016.7513858.

[2] Y. Ku *et al.*, "Towards Software-Defined VANET: Architecture and Services," *2014 13th Annual Mediterranean Ad Hoc Networking Wksp.*, Piran, 2014, pp. 103–10, DOI: 10.1109/MedHocNet.2014.6849111.

[3] H. Li, M. Dong, and K. Ota, "Control Plane Optimization in Software-Defined Vehicular Ad Hoc Networks," *IEEE Trans. Vehic. Tech.*, vol. 65, no. 10, 2016, pp. 7895–904. DOI: 10.1109/TVT.2016.2563164.

[4] Y. O. Basciftci *et al.*, "How Vulnerable Is Vehicular Communication to Physical Layer Jamming Attacks?," *2015 IEEE 82nd VTC-Fall 2015*, Boston, MA, 2015, pp. 1–5. DOI: 10.1109/VTCFall.2015.7390968.

[5] K. Xu *et al.*, "Toward Software Defined Smart Home," *IEEE Commun. Mag.*, vol. 54, no. 5, May 2016. DOI: 10.1109/MCOM.2016.7470945, pp. 116–22.

[6] A. U. Khan and B. K. Ratha, "Time Series Prediction QoS Routing in Software Defined Vehicular Ad-Hoc Network," *2015 Int'l. Conf. Man and Machine Interfacing*, Bhubaneswar, 2015. DOI: 10.1109/MAMI.2015.7456576 pp. 1–6.

[7] M.A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Software-Defined Networking for RSU Clouds in Support of the Internet of Vehicles," *IEEE Internet of Things J.*, vol. 2, no. 2, 2015, pp. 133–44. DOI: 10.1109/JIOT.2014.2368356.

[8] X. Wen *et al.*, "Towards a Secure Controller Platform for OpenFlow Applications," *Proc. 2nd ACM SIGCOMM Wksp. Hot Topics in Software Defined Networking*, 2013, pp. 171–72. DOI: 10.1145/2491185.2491212.

[9] S. Shin *et al.*, "Rosemary: A Robust, Secure, and High-Performance Network Operating System," *Proc. 2014 ACM SIGSAC Conf. Comp. and Commun. Security*, 2014, pp. 78–89. DOI: 10.1145/2660267.2660353.

[10] H. Song, G. A. Fink, and S. Jeschke, *Security and Privacy in Cyber-Physical Systems: Foundations, Principles and Applications*, Wiley-IEEE Press, 2017.

[11] C. Monsanto *et al.*, "Composing Software Defined Networks," *Proc. 10th USENIX Symp. Networked Systems Design and Implementation*, 2013, pp. 1–13.

[12] A. Akhunzada *et al.*, "Secure and Dependable Software Defined Networks," *J. Network and Computer Applications*, vol. 61, 2016, pp. 199–221; http://dx.doi.org/10.1016/j.jnca.2015.11.012.

[13] D. Kreutz, F. Ramos, and P. Verissimo, "Towards Secure and Dependable Software-Defined Networks," *Proc. 2nd ACM SIGCOMM Wksp. Hot Topics in Software Defined Networking*, 2013, pp. 55–60. DOI: 10.1145/2491185.2491199

[14] W.-T. Chen and C.-H. Ho, "Spectrum Monitoring with Unmanned Aerial Vehicle Carrying a Receiver Based on the Core Technology of Cognitive Radio — A Software Defined Radio Design," *J. Unmanned Vehicle Systems*, 2016. DOI: 10.1139/juvs-2016-0011.

[15] A. Akhunzada *et al.*, "Man-at-the-End Attacks: Analysis, Taxonomy, Human Aspects, Motivation and Future Directions," *J. Network and Computer Applications*, vol. 48, no. 0, 2015, pp. 44–57; http://dx.doi.org/10.1016/j.jnca.2014.10.009.

Biographies

Adnan Akhunzada (a.qureshi@comsats.edu.pk) is currently working as an assistant professor and an active senior researcher at the Centre of Applied Security, Comsats Institute of Information Technology, Islamabad, Pakistan. He taught international modules of the University of Bradford, United Kingdom. He has published several high impact research journals, IEEE transactions, and highly reputable magazine papers. His current research interests include secure design and modeling of software defined networks and future Internet, lightweight cryptography, man-at-the-end attacks, human attacker attribution and profiling, and remote data auditing.

Muhammad Khurram Khan (mkhurram@ksu.edu.sa) is working at the Center of Excellence in Information Assurance, King Saud University, Saudi Arabia. He has edited seven books and proceedings published by Springer-Verlag and IEEE. He has published more than 200 papers in international journals and conferences. He is an inventor of 10 U.S./PCT patents. He is the Editor-in-Chief of a well-reputed journal, *Telecommunication Systems* (Springer), and a member of several editorial boards. His research interests include cybersecurity, biometrics, multimedia security, and digital authentication.

SDN Enabled 5G-VANET: Adaptive Vehicle Clustering and Beamformed Transmission for Aggregated Traffic

Xiaoyu Duan, Yanan Liu, and Xianbin Wang

In order to support the increasing traffic and improve HetNet management, the authors propose an SDN enabled 5G vehicular ad hoc network, where neighboring vehicles are clustered adaptively according to real-time road conditions using SDN's global information gathering and network control capabilities.

ABSTRACT

With the anticipated arrival of autonomous vehicles, supporting vehicle generated data traffic due to the dramatically increased use of in-vehicle mobile Internet access will become extremely challenging in 5G-based vehicular networks. This is mainly due to the high mobility of vehicles on the road and the high complexity of 5G HetNets. In order to support the increasing traffic and improve HetNet management, an SDN enabled 5G VANET is proposed in this article, where neighboring vehicles are clustered adaptively according to real-time road conditions using SDN's global information gathering and network control capabilities. With proposed dual cluster head design and dynamic beamforming coverage, both trunk link communication quality and network robustness of vehicle clusters are significantly enhanced. Furthermore, an adaptive transmission scheme with selective modulation and power control is proposed to improve the capacity of the trunk link between the cluster head and base station. With cooperative communication between the mobile gateway candidates, the latency of traffic aggregation and distribution is also reduced. Computer simulation results show that the proposed design substantially improved 5G users' bit error rate and trunk link throughput rate.

INTRODUCTION

With the recent advancement of artificial intelligence and sensor technologies, autonomous or self-driving vehicles are able to sense their surroundings in real time by the combined use of many techniques including radar, lidar, GPS, and computer vision. Consequently, the autonomous vehicle is closer to reality than we ever thought. According to a recent survey [1], it is predicted that more than 15 million autonomous vehicles will be on the road by 2030. We can imagine that very soon drivers will be relieved of the burden of driving and thus have time for mobile Internet access. As such, handling of the growing vehicle generated data traffic in cellular vehicular ad hoc networks (VANETs) [2], which involves ad hoc communications among nearby vehicles and between vehicles and nearby roadside equipment, has attracted lots of attention from both academia and industry.

However, fifth generation (5G)-VANET has inherent challenges in supporting extremely dynamic vehicle generated data traffic. Due to the high mobility of vehicles and their irregular distribution, a VANET has dynamic network topology, where vehicles can join or leave the network quickly, and the links between vehicles connect and disconnect very often. As such, timely updating of network topology is essential for the operation of 5G-VANET. These challenges are further compounded by the more stringent latency requirements of 5G [3], hence requiring more consistent link quality. When the densified vehicles all directly communicate with the cellular base station (BS) or roadside units during rush hour, the high volume of concurrent vehicle-to-infrastructure (V2I) communications may lead to extremely high signaling overhead and outage probability. Consequently, adaptive, reliable, and situation-aware management of dynamic traffic is critical for the operation of 5G-VANET.

Furthermore, 5G-VANET is also impeded by heterogeneity and ossified cellular network architecture due to the use of diverse access networks and vendor-specific equipment. Due to the inevitable network densification in the quest for high data rate and the mixed use of different wireless technologies, 5G is envisioned to have a heterogeneous network (HetNet) architecture [4]. The intractable interconnection and the limited information sharing between HetNet infrastructures and different operators bring additional difficulty for vehicle traffic management. Additionally, with the increasing complexity of future 5G networks, vendor-specific hardware and protocols make it difficult and remarkably expensive for operators to dynamically adapt their network operations.

In addressing these challenges, we propose a software-defined networking (SDN) enabled 5G-VANET with the capability of adaptive vehicle clustering and beamformed transmission to support the aggregated traffic from the cluster head. Through the separation of data plane and control plane [4], SDN enables 5G-VANET management and facilitates the centralized control over HetNets by providing a global network view and a unified configuration interface despite the underlying HetNets involved. With its open and reconfigurable interface, SDN provides an enabling platform to apply intelligence and consistent policy for 5G-VANET HetNets. In the proposed 5G-VANET, arriving road traffic will

Digital Object Identifier: 10.1109/MCOM.2017.1601160

The authors are with Western University.

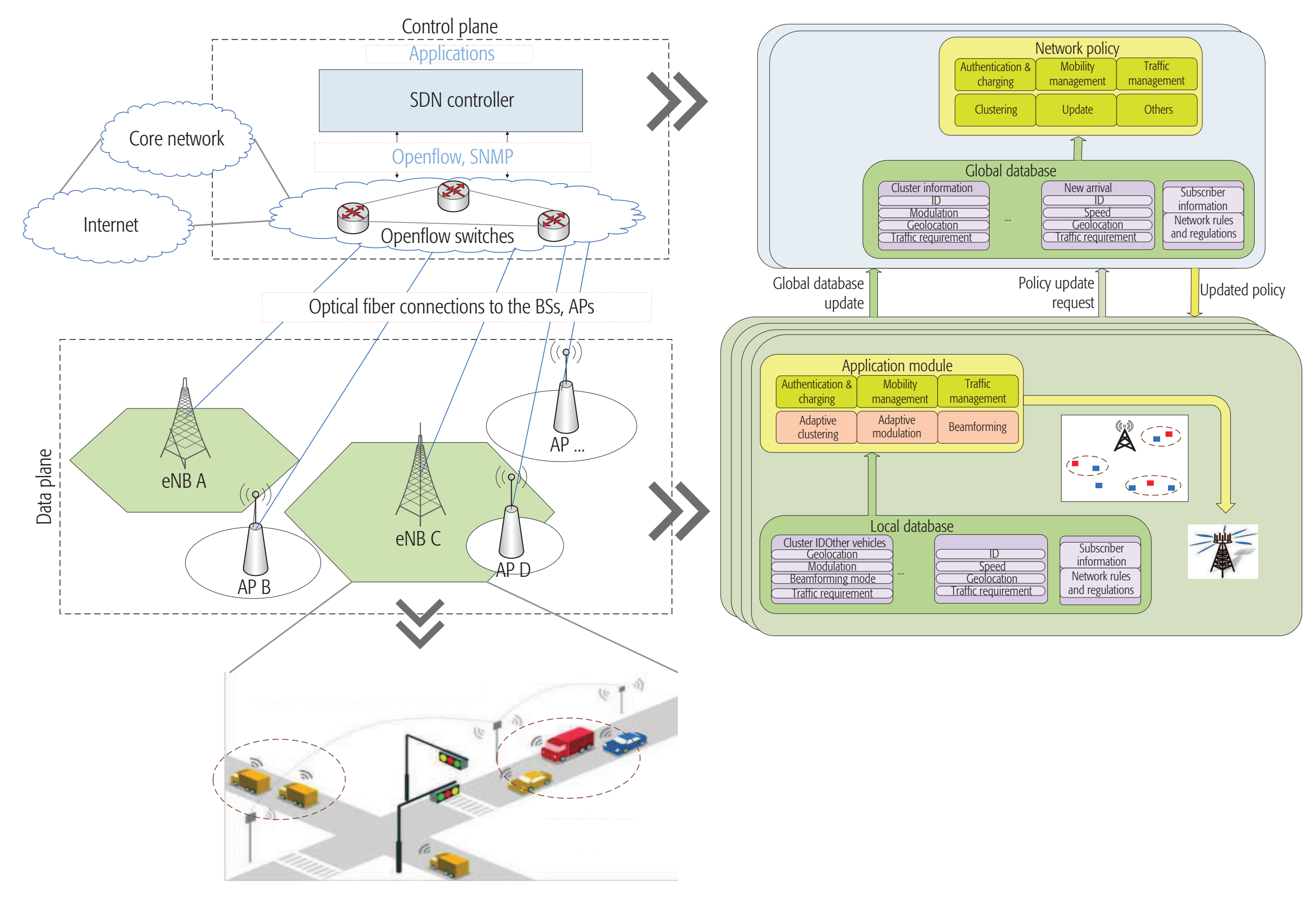


Figure 1. SDN enabled 5G-VANET integrated network architecture. The SDN controller decides the policies related to various network functionalities, while the BSs implement the controller-defined policies.

be predicted with the assistance of SDN to achieve adaptive vehicle clustering. Within each vehicle cluster, a cluster head (CH) is selected to aggregate traffic from other vehicles and communicate with the cellular BS in order to reduce signaling overhead. A dual CH design is then proposed to guarantee the robustness and seamless trunk link communication. Furthermore, when there is high capacity demand over the trunk link from intra-cluster vehicles, an adaptive beamformed trunk link transmission scheme and cooperative communication are considered as potential solutions for improving the 5G-VANET communication quality and capacity and reducing the traffic distribution latency.

The remainder of this article is organized as follows. We first present the network architecture of the SDN enabled 5G-VANET. Based on the network model, the proposed adaptive vehicle clustering and dual CH design enabled by SDN are then elaborated. The beamformed adaptive transmission scheme and cooperative communication are explained, while the performance of SDN enabled 5G-VANET applications is analyzed. The conclusions are drawn in the last section.

Overall Network Architecture of SDN Enabled 5G-VANET

The overall network architecture of the proposed SDN enabled 5G-VANET, which consists of a HetNet environment, as shown in Fig. 1, is designed to support adaptive vehicle clustering and trunk link traffic aggregation schemes. It features a layered architecture consisting of macrocells and small cells, including, base station (BS), access point (AP), and so on. As shown in this figure, vehicles move across the cells, bearing dynamic traffic requirements. In order to provide a consistent policy and global management of the 5G-VANET, macrocell BSs and APs are all controlled by a centralized SDN controller with OpenFlow protocol through high-capacity fiber optic links. SDN removes the control logic from the underlying infrastructure (e.g., BSs and APs) to the control layer so that applications can then be implemented on the central SDN controller to provide new functions, including adaptive clustering and traffic management, over the whole 5G HetNets.

The right side of Fig. 1 shows the operational architecture of the proposed 5G-VANET architecture. The SDN controller is responsible for the global policies related to the access network, including authentication, mobility/traffic management, and other issues, while the BSs (APs) constitute the data plane of the SDN enabled HetNets and implement the controller-defined policies.

Base Stations and Access Points: In the proposed framework for 5G-VANET, each BS and AP has a local database (LDB) and application module. With the support of the LDB, the BS is able to obtain the cell load conditions and facilitates local decision making [5]. In general, the LDB stores the information about vehicles within the cell, In general, LDB stores the information about vehicles within the cell, including the clustering information, geo-location of vehicles, traffic requirement and transmission scheme. Each of these sections is updated when there are new vehicles access or leave the current cell.

including the clustering information, geo-location of vehicles, traffic requirement, and transmission scheme. Each of these sections is updated when there are new vehicles accessing or leaving the current cell.

The information gathered from multiple LDBs constitutes the global database (GDB), which is then utilized by the SDN controller to design network level policy and update the local application modules, as illustrated in Fig. 1. Afterward, clusters are formed adaptively, and the beamformed transmission scheme of aggregated traffic is determined accordingly. The BSs would take care of all local decision making (e.g., the communication of the other cellular users that are not defined by the centralized controller [6]) in order to reduce the processing burden of the controller.

SDN Controller: The SDN controller enables the coordination and information sharing between HetNets through the separation of the control and data planes. As shown in Fig. 1, the SDN controller includes a GDB and a network policy-making module. The GDB contains information about all the vehicle clusters of the service area and is updated regularly by the BSs/APs. With the global view over the whole service area, programmable applications can be run on the controller to realize global functions, including authentication, adaptive clustering, and so on. The updates of the overall network policies could be proactive (after a predefined period of time) or reactive (requested by BS/AP due to cell overloading).

Based on the overall SDN-enabled 5G-VANET network architecture elaborated above, an adaptive vehicle clustering scheme and dual cluster head design are proposed in the following section. After the formation of the vehicle clusters, the beamformed adaptive transmission scheme for the trunk link between cluster head and BS is also discussed in detail in order to support the aggregated traffic from the clustered vehicles.

Adaptive Clustering in SDN Enabled 5G-VANET

Due to the high mobility of vehicles and the restrictions in their range of motion, vehicle clustering is seen as a promising solution in reducing the overhead of cellular networks and providing better communication quality with a low relative speed among clustered vehicles. In related studies, authors in [7] provide a multi-layer cloud radio access network (C-RAN) architecture in order to cluster multi-domain resources for a group of vehicles as well as a single vehicle. However, detailed algorithms are still to be designed under the software-defined vehicular HetNet architecture. In [8], a dynamic clustering-based mobile gateway management mechanism is proposed, which considers vehicle mobility and executes a clustering algorithm periodically. However, how to decide this period still remains an open problem, and the cluster maintenance also dramatically increases the computing load of the cluster head. With the coexistence of multiple HetNet infrastructures in future 5G networks, it is also difficult for a single BS to predict the arriving traffic and execute clustering algorithms adaptively due to limited resources.

In the proposed SDN enabled 5G-VANET, the controller's global view over the HetNets and the timely updating of road traffic topology provide a viable solution in addressing the above challenges. As vehicles usually move fast and APs only have limited coverage, we consider that APs only provide updated information of related vehicles and the clustered vehicles would communicate, with BSs through a selected vehicle, that is, a cluster head (CH). Due to the consistency of moving speed and direction of traveling vehicles, an SDN controller will be able to monitor and predict the location of arriving vehicles using different locationing and data analytics techniques, and then inform the relevant cellular BSs in advance to guarantee adaptive and efficient clustering, as shown in Fig. 2. Based on the high-level "road topology" collected from heterogeneous BSs and APs of different infrastructures or operators, the proposed clustering algorithm would be executed only when needed instead of periodically. We can also define a traffic threshold and take the delay requirement and size of the upcoming in-vehicle data traffic into consideration when making vehicle clustering decisions.

In Fig. 2, vehicles that are moving in two directions are grouped into different clusters. The vehicles that have cellular interfaces (i.e., yellow cars in Fig. 2) are defined as mobile gateway candidates because they are able to communicate with cellular networks. A CH is selected from the mobile gateway candidates, and then all other vehicles in the same cluster communicate with the BS through the CH. Moreover, communication between the CH and other intra-cluster vehicles could be through different wireless protocols (e.g., IEEE 802.11p) to relieve the cellular burden and save licensed spectrum resources. In order to guarantee seamless communication, a backup CH is also selected from the mobile gateway candidates to record a copy of the signaling message, that is, floating car data (FCD), from the existing CH and be prepared for emergencies [9]. Note that in Fig. 2, beamforming is applied to focus the cellular signal in areas with concentrated vehicles. The vehicle cluster colored blue illustrates the uplink traffic collection procedure, while the cluster colored orange shows the downlink traffic distribution. The cooperative multi-receiver coordinated decoding is used in traffic distribution only when the trunk-link traffic volume is higher than a threshold; details are provided later.

Next we elaborate the SDN enabled adaptive vehicle clustering mechanism in 5G-VANET. Specifically, SDN enabled adaptive clustering is realized under the collaboration of cellular BS and mobile gateway candidates. There are three parameters utilized during the clustering procedure: angle of arrival (AoA) (θ), received signal strength (RSS), and inter-vehicular distance (IVD). Below, the adaptive clustering procedure is divided into four steps.

Base station initialized grouping: With the road traffic topology provided by the SDN controller, the BSs are aware of the arriving traffic and prepare themselves in advance. Once the cell is overloaded and clustering conditions are met, the vehicles are roughly classified into groups according to similar AoA and RSS, and the member information of each group is sent back to the mobile gateway candidates in the group.

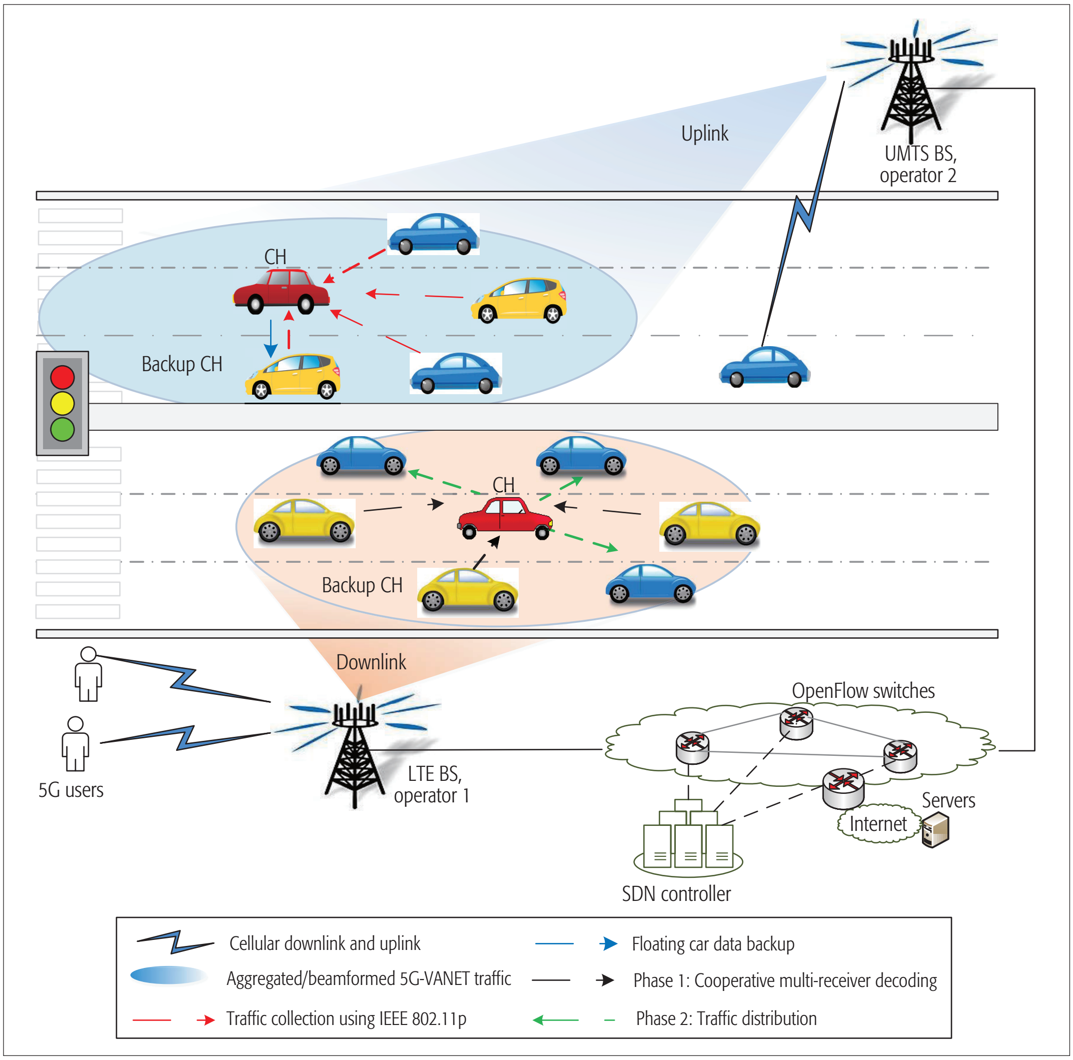

Uplink
UMTS BS, operator 2
CH
Backup CH
CH
Backup CH
Downlink
5G users
LTE BS, operator 1
OpenFlow switches
Servers
Internet
SDN controller
Cellular downlink and uplink
Aggregated/beamformed 5G-VANET traffic
Traffic collection using IEEE 802.11p
Floating car data backup
Phase 1: Cooperative multi-receiver decoding
Phase 2: Traffic distribution

Figure 2. SDN enabled adaptive clustering in 5G-VANET integrated networks. A CH is selected from the mobile gateway candidates, and then the other intra-cluster vehicles communicate with BSs through the CH.

> Due to the high mobility of vehicles and the restrictions in their range of motion, vehicle clustering is seen as a promising solution in reducing the overhead of cellular network and providing better communication quality with a low relative speed among clustered vehicles.

Assume that at the BS side, the received signals from vehicles are classified into N equally divided transmission angles of $360/N$ degrees, and the speed limit of the road is around V_{MAX}. We can define different vehicle groups by the combination of different transmission angle and RSS. Each group is then characterized by

$$\theta_x - \theta_y \le \frac{360}{N} \text{ and } RSS_x - RSS_y \le 1 - e^{-\frac{\Delta V}{a}},$$

where x and y represent two vehicles, ΔV is the speed difference of two vehicles, and a is a constant that defines the rate of variation of the 5G signal strength when the mobility speed increases or decreases by a unit [8].

Vehicle clusters formation: After receiving the vehicle grouping list from the base station, IVD would be used by the mobile gateway candidates to refine the group and form the final cluster. As the vehicle position information measured or predicted by BSs might not be accurate, the mobile gateway candidates use a broadcasting message (For example, IEEE 802.11p has a transmission range of around 250 m) to verify the neighbor vehicles and update the group member list. The IVD of the final clusters is constrained by $d \le R_t \cdot (1 - \varepsilon)$, where R_t denotes the maximum transmission range of IEEE 802.11p and ε reflects the wireless channel fading conditions [10].

Dual cluster head selection: After the formation of the clusters, a CH would be selected in each cluster in order to effectively relay the vehicle related traffic to cellular networks. Assume that there are K vehicles in a cluster; the CH selection could be defined by a linear optimization problem [10]. The objective of the CH selection is to maximize the throughput rate of the trunk link under the constraint of channel quality and moving speed of the vehicle. To be specific, the closer the vehicle speed is to the average cluster speed, the longer this CH candidate would stay in this cluster and the better it can serve as a CH. Similarly, the better the channel quality between CH and BS, the more reliable the trunk link transmission.

Note that the selected CH collects the status (position, velocity, and heading direction) of vehicles (i.e., FCD) and reports to the BSs. This kind of data is characterized as high frequency and small data size, which occupies cellular network resource frequently and impairs other applications. Through clustering mechanism, FCD data is compressed and only transmits through the CH.

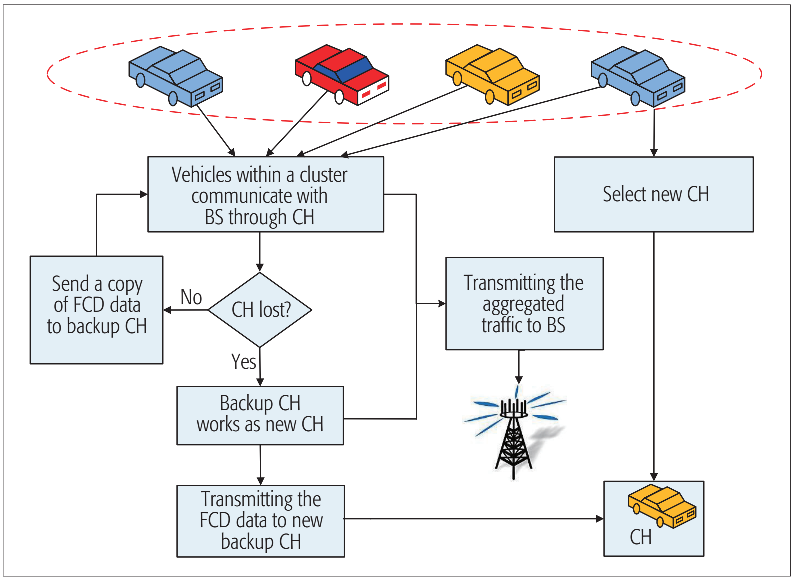

Vehicles within a cluster communicate with BS through CH
Select new CH
Send a copy of FCD data to backup CH
No
CH lost?
Yes
Transmitting the aggregated traffic to BS
Backup CH works as new CH
Transmitting the FCD data to new backup CH
CH

Figure 3. The dual cluster head selection scheme. The proposed algorithm improves the network robustness and guarantees the seamless communication during CH handover.

However, this design increases the vulnerability of the system and poses a potential risk that the CH could be a single point of failure.

For this reason, we further propose a dual CH design in each cluster for SDN enabled 5G-VANET to improve network robustness and guarantee seamless communication during CH handover. In this dual CH scheme, a backup CH is also selected according to the CH selection criteria. The existing CH always sends a copy of FCD data to the backup CH, as shown in Fig. 3. Once there is something wrong with the CH, such as an accident or an unpredictable emergency, a backup CH could be prepared in advance and thus be able to take over the responsibility seamlessly. Moreover, the backup CH also works as a smooth transition during the handover procedure to a new CH. As a result, under the scenario that the existing CH leaves cluster normally, the backup CH becomes the CH immediately, and a new backup CH is selected, as we can see in Fig. 3. The dual CH design is especially beneficial for 5G latency-stringent application with a reduced communication interruption probability.

Cluster maintenance and adaptation: Last but not least, the clusters should be maintained and updated due to frequent road traffic changes in VANETs. In the proposed SDN enabled adaptive clustering scheme, the BS would only inform the corresponding CH if the new arriving vehicles will stay in the CH transmission area for a time period larger than a threshold T_p. The predicted inhabitant time (PIT) is calculated using the angle of a new arriving vehicle to the center of the cluster and the speed of the arriving vehicle [10]. Afterward, the CH would then be prepared for the new traffic and execute a clustering algorithm only when needed.

On the other hand, if the aggregated amount of traffic exceeds the trunk-link capacity, the communication quality would deteriorate and outage probability would increase. Under this situation, some vehicles with high traffic requirement should be removed from the cluster to guarantee the communication quality of service (QoS). The cluster maintenance and adaptation should be monitored as an ongoing procedure in terms of the communication quality index (e.g., outage probability).

Beamformed Adaptive Transmission Schemes in 5G-VANET

After the formation of vehicle clusters and the selection of CH, the CH would aggregate the traffic from other vehicles in the cluster and communicate with the cellular BS. As the volume of the aggregated traffic is much higher, provisioning of a high-capacity trunk link is critical in order to guarantee the communication performance of the clustered vehicles in the SDN enabled 5G-VANET vehicle clustering design. Therefore, in this section, beamformed adaptive transmission of the trunk link between CH and BS is elaborated in detail. Beamforming is used to provide directional coverage of the vehicle clusters with two selective coverage mode. Afterward, the adaptive trunk link transmission scheme is introduced, which consists of dynamic modulation and power control. When the trunk link traffic volume at the CH or latency requirement of the traffic is exceptionally high, cooperative communication of the mobile gateway candidates is also proposed in order to improve communication quality, utilizing diversity gain and reducing the delay of traffic distribution through multi-user decoding.

Directional Coverage of Vehicle Clusters with Beamforming

After vehicle cluster formation, massive multiple-input multiple-output (MIMO) antennas are selected to form a highly directional beam to cover the given cluster or CH. Specifically, the macro BS estimate downlink channel information via uplink pilots under the assumption of channel reciprocity. In high mobility scenarios, use of channel reciprocity between uplink and downlink could introduce large error due to the reduced channel coherence time. In order to improve the system adaptivity, long-term channel prediction could be adopted to cope with the fast channel changes. According to this channel information, desired antennas are selected to design beamforming gain vectors. Using linear precoders such as minimum mean square error (MMSE), zero forcing (ZF), and maximum ratio transmission (MRT) precoding [11], the vehicles in different beamforming sub-bands would be orthogonal in the space domain, so their interference is reduced dramatically.

The two beamforming solutions shown on the right side of Fig. 4 further provide flexibility and adaptivity for the beamforming design of SDN enabled 5G-VANET. In the first solution, the beam only covers the CH to optimize the beam width and reduce intra-beam interference. In the second solution, the beam covers the whole cluster. Although a wider beam in the latter scheme is less focused and achieves lower signal-to-interference-plus-noise ratio (SINR) than a more focused beam, it has the advantage of better coverage and guarantees seamless connection during CH failure. Therefore, due to the high mobility of vehicles, it is preferred that a wider beam is utilized to enhance coverage in the proposed SDN enabled 5G-VANET architecture. Furthermore, when multiple clusters coexist and are close to each other, a narrow beam is applied to reduce interference and improve the trunk link throughput rate.

Adaptive Trunk Link Transmission for Aggregated Traffic

As it is obvious that the amount of V2I traffic would change at different times, an adaptive transmission scheme for the aggregated traffic is designed in this section. The adaptive transmission scheme consists of adaptive modulation and coding (AMC) and power control. Compared to traditional AMC, we introduce a non-orthogonal multiplexed modulation scheme to cope with the varying channel condition of fast moving vehicles. In the proposed adaptive AMC, orthogonal and non-orthogonal modulation and coding schemes are selected adaptively according to channel quality in order to achieve high spectral efficiency. Power control is also used to adapt power allocation to instantaneous channel variations so that a required SINR level can be guaranteed. As QoS is essential for real-time services, such as live video and interactive games, an optimal combination of AMC and power control techniques is used in the adaptive trunk link transmission scheme in order to achieve highest trunk link throughput rate, while maintaining the required QoS (measured by outage probability).

During the adaptive trunk link transmission, the CH first uses traditional orthogonal modulation (OM) to communicate with the BS as OM has low system complexity. When the trunk link traffic volume becomes higher than a threshold (e.g., when intra-cluster vehicles all request high data rate at the same time), non-orthogonal multiplexed modulation (NOMM) would be utilized to improve the trunk-link capacity and reduce the average transmit power. NOMM allows parallel data streams of one user to be modulated simultaneously and partially overlapped on a group of resource elements through sparse spreading code [12]. Compared to orthogonal modulation, NOMM is robust to the varying channel condition due to the fact that data streams belong to the same user and suffer the same channel variation. It can also improve spectrum efficiency through overlapping on resource blocks. Therefore, in order to find a trade-off between throughput rate and receiver complexity, NOMM and OM should be selected adaptively according to varying traffic requirements.

Cooperative Communication in 5G-VANET

A cooperation scheme of virtual MIMO-based cooperative communication is introduced in this section to further improve the communication quality and reduce the latency of the traffic distribution phase when the trunk link traffic amount is high.

When the trunk-link traffic volume and delay requirement are higher than thresholds, cooperative communication is triggered, and several mobile gateway candidates in that cluster could be selected to share their antennas with the CH as virtual antenna arrays and then communicate with the BS. The number of the vehicles participate in the cooperative communication is a trade-off between the performance and complexity. In the uplink transmission, the selected mobile gateway candidates serve as the sub-cluster head and collect traffic from the vehicles nearby. They transmit traffic to the BS simultaneously with the CH and thus improve the throughput rate with multiple trunk links; in downlink transmission, as shown in Fig. 2, the selected mobile gateway candidates also listen to the BS and help the CH in traffic decoding, and thus reduce the latency of traffic distribution through multi-user cooperation. Therefore, the cooperative communication not only brings diversity gain but also potentially reduces latency in decoding.

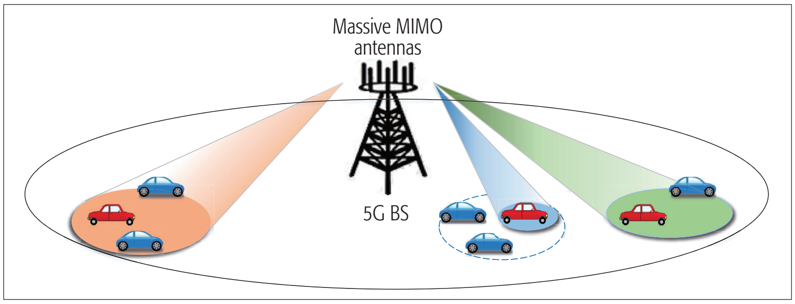


Figure 4. The beamforming directional coverage over the vehicle clusters along a road crossing the cell. Selective beamforming mode with wider or narrow beam is also given in the figure.

Performance Evaluation

In order to evaluate the performance of SDN enabled 5G-VANET applications, MATLAB simulations have been conducted in terms of the SDN enabled adaptive clustering scheme and adaptive transmission scheme of aggregated traffic. It is assumed that there are 6 vehicles randomly distributed within a cluster and the data packet size from each vehicle is 512 bytes [13]. Each vehicle generates 10 packets/s. Note that using the principle of Monte Carlo, all simulations are carried with a 95 percent confidence interval (CI) [2]. SDN enabled adaptive clustering is compared to existing mechanisms, and three scenarios have been considered in the simulations: the proposed scheme, which use signal-to-noise ratio (SNR) combined with average speed to select CH; the traditional method, which chooses the center vehicle as CH; and the scenario in which there is no clustering mechanism (vehicles communicate with BSs through their own connection).

Figure 5 illustrates the simulation results of bit error rate (BER) vs. SNR in terms of the three different scenarios. It can be seen that generally clustering provides better communication quality for vehicles, which is reasonable because vehicle clustering schemes use IEEE 802.11p networks to offload the burden of cellular networks, and thus guarantee the cellular radio link connection quality. It is also clear that the proposed CH selection scheme has the lowest trunk link BER, especially in higher SNR situations. This is because the proposed CH selection scheme takes both SNR and average speed into consideration when choosing a CH, and thus ends up with an optimum CH that has better radio link quality and serves longer in the cluster (less CH handover).

In the simulation of adaptive transmission scheme, quadrature phase shift keying (QPSK)

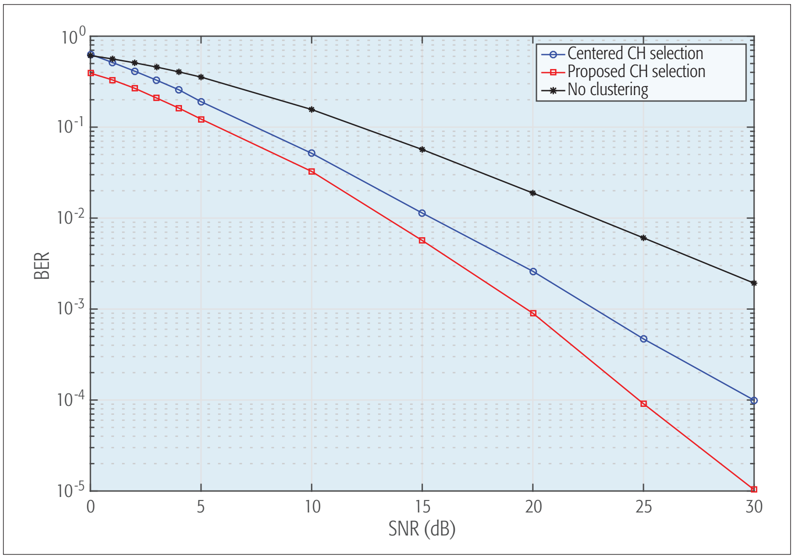


Figure 5. Simulation results of BER vs. SNR in terms of three different vehicle clustering and CH selection methods.

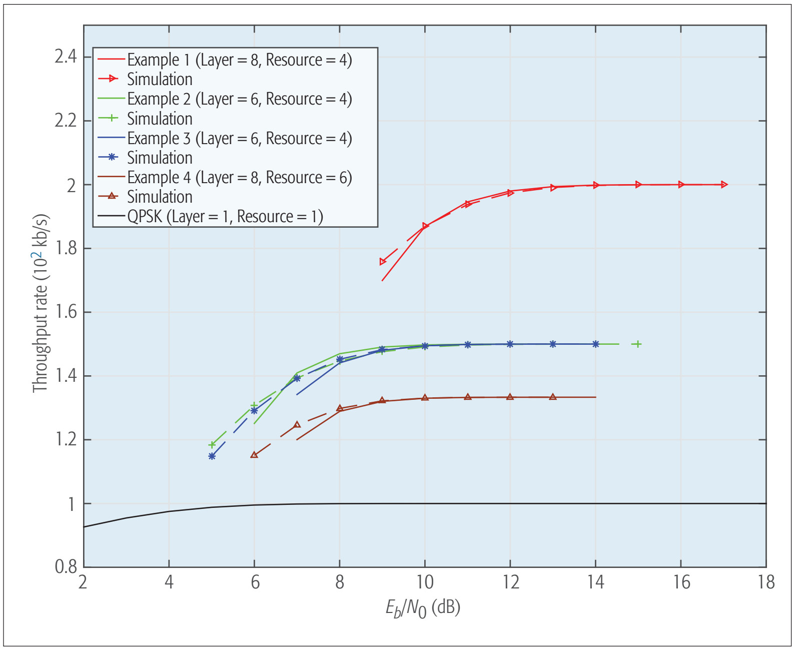


Figure 6. Throughput rate comparison of two trunk link modulation schemes: NOMM modulator and QPSK modulation.

and NOMM are simulated as two modulation schemes to show the transmission performance. In the NOMM scheme, QPSK modulation combined with sparse spreading were used for each layer, and ML detection was applied at the receivers. Link level simulation is implemented including channel coding, modulation, and demodulation. Monte Carlo simulation is also given in order to verify the theoretical analysis. According to [14], there are two important factors for NOMM signature design: the minimum Euclidean distance and the girth (the minimum cycle length of the factor graph). The detection performance becomes better with increasing minimum Euclidean distance and decreasing minimum cycle length. The principle of optimal signature design for a given factor graph is to find a trade-off between the minimum Euclidean distance and the minimum cycle length. Using this principle, some optimal signature matrix examples are designed for the simulation. The load ratio of example 1 to 4 is 2, 1.5, 1.5, and 1.33, while the minimum code distances are 0.83, 1.166, 1.27, and 1.41, respectively [15].

Figure 6 illustrates the throughput rate (dashed lines) and the union bounds (solid lines) [14] of NOMM codes given in examples 1, 2, 3, and 4 [15], and single-user QPSK code. From the simulation results, we can see that:

1. All the simulations coincide well with their union bound most of the time, except a little mismatch at low E_b/N_0 due to noise.
2. For NOMM examples with different bit numbers per symbol, the throughput rate becomes higher with increased bits per symbol. For example, code obtained in example 1 with the optimal signature has the best throughput rate since it has the maximum bits per symbol.
3. The throughput rate of NOMM is better than that of QPSK due to the overlay transmission.

This confirms that NOMM provides higher throughput rate at the cost of complexity. Therefore, adaptive transmission of the aggregated traffic should work in such a way that QPSK and NOMM are selected dynamically according to traffic requirements.

Conclusion

With the anticipated arrival of autonomous vehicles and dramatic growth of in-vehicle mobile data traffic, supporting dynamic vehicle communications in 5G HetNets is expected to be challenging, due to fast varying network topology and high complexity of the heterogeneous infrastructure. In this article, we propose to integrate SDN into 5G-VANET and thus provide a programmable platform to address the challenges. Through the proposed SDN enabled adaptive vehicle clustering and dual cluster head scheme, the signaling overhead of a VANET is significantly reduced along with improved communication quality. The proposed cluster head selection also guarantees seamless access to the operators' services for cluster users. In order to further accommodate the varying traffic over the trunk link and reduce the latency during traffic distribution, an adaptive trunk link transmission scheme and cooperative communication of mobile gateway candidates were proposed for the aggregated V2I traffic transmission in this integrated network. Simulation results show that SDN coordinated vehicle clustering and beamformed transmission are suitable to support fast varying traffic conditions with extremely large dynamic range.

References

[1] D. Mohr *et al.*, "Automotive Revolution-Persepective Towards 2030," McKinsey & Co., tech. rep., Jan. 2016.

[2] E. A. Feukeu, S. M. Ngwira, and T. Zuva, "Doppler Shift Signature for BPSK in a Vehicular Network: IEEE 802.11p," *IEEE ICMA*, Beijing, China, 2015, pp. 1744–49.

[3] J. G. Andrews *et al.*, "What Will 5G Be?," *IEEE JSAC*, vol. 32, no. 6, June 2014, pp. 1065–82.

[4] H. Li, M. Dong, and K. Ota, "Control Plane Optimization in Software-Defined Vehicular Ad Hoc Networks," *IEEE Trans. Vehic. Tech.*, vol. 65, no. 10, Oct. 2016, pp. 7895–7904.

[5] X. Duan and X. Wang, "Authentication Handover and Privacy Protection in 5G Hetnets Using Software-Defined Networking," *IEEE Commun. Mag.*, vol. 53, no. 4, Apr. 2015, pp. 28–35.

[6] A. M. Akhtar, X. Wang and L. Hanzo, "Synergistic Spectrum Sharing in 5G HetNets: A Harmonized SDN-Enabled Approach," *IEEE Commun. Mag.*, vol. 54, no. 1, Jan. 2016, pp. 40–47.

[7] K. Zheng *et al.*, "Soft-Defined Heterogeneous Vehicular Network: Architecture and Challenges," *IEEE Network*, vol. 30, no. 4, July 2016, pp. 72–80.

[8] A. Benslimane, T. Taleb, and R. Sivaraj, "Dynamic Clustering-Based Adaptive Mobile Gateway Management in Integrated VANET-3G Heterogeneous Wireless Networks," *IEEE JSAC*, vol. 29, no. 3, Mar. 2011, pp. 559–70.

[9] A. Al-Fuqaha *et al.*, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surveys & Tutorials*, vol. 17, no. 4, Feb. 2015, pp. 2347–76.

[10] X. Duan, X. Wang and Y. Liu, "SDN Enabled Dual Cluster Head Selection and Adaptive Clustering in 5G-VANET," *IEEE VTC-Fall*, Montreal, Canada, 2016, pp. 6–12.

[11] H. Zaaraoui, Z. Altman and E. Altman, "Beam Focusing Antenna Array Technology for Non-Stationary Mobility," *IEEE WCNC*, Doha, Qatar, 2016, pp. 1–6.

[12] S. Zhang *et al.*, "Sparse Code Multiple Access: An Energy Efficient Uplink Approach for 5G Wireless Systems," *IEEE GLOBECOM*, Austin, TX, 2014, pp. 4782–87.

[13] D. B. Rawat *et al.*, "Enhancing VANET Performance by Joint Adaptation of Transmission Power and Contention Window Size," *IEEE Trans. Parallel Distrib. Sys.*, vol. 22, no. 9, Sept. 2011, pp. 1528–35.

[14] G. Song and J. Cheng, "Distance Enumerator Analysis for Multi-User Codes," *IEEE Int'l. Symp. Info. Theory*, Honolulu, HI, 2014, pp. 3137–41.

[15] Y. Liu, X. Wang, and X. Duan, "Aggregated V2I Communications for Improved Energy Efficiency Using Non-Orthogonal Multiplexed Modulation," *IEEE VTC-Fall*, Montreal, Canada, 2016, pp. 1–6.

Biographies

Xiaoyu Duan (xduan8@uwo.ca) is a Ph.D. candidate at the Department of Electrical and Computer Engineering, Western University, Canada. She received a B.Sc. in communication engineering from Tianjin University in 2010 and an M.Sc. in signal and information processing from Beijing University of Posts and Telecommunications, China, in 2013. Her research interests include software-defined networking, traffic offloading, self-organizing networks, and communication security in 5G heterogeneous networks. She has more than 10 publications in journals and conferences including *IEEE Communication Magazine*, ICC, WCNC, VTC, and PIMRC.

Yanan Liu (yliu2683@uwo.ca) is a Ph.D. candidate at the Department of Electrical and Computer Engineering, Western University. She received a B.Sc. in communication engineering from Dalian Maritime University, China, in 2011 and an M.Sc. in communication engineering from the University of Manchester, United Kingdom, in 2012. From November 2012 to March 2015, she was a system engineer at DaTang Mobile Communication Equipment Co., Beijing, China, where she worked on simulation development and performance analysis for LTE/LTE-A cellular systems. Her research interests include adaptive wireless communication and physical-layer technology of 5G communication systems.

Xianbin Wang [S'98, M'99, SM'06, F'17] (xianbin.wang@uwo.ca) is a professor and Canada Research Chair at Western University. He received his Ph.D. degree in electrical and computer engineering from the National University of Singapore in 2001. His current research interests include 5G technologies, communications security, signal processing for communications, adaptive wireless systems, and locationing technologies. He has over 280 peer-reviewed journal and conference papers, in addition to 26 granted and pending patents and several standard contributions. He has received many awards and recognitions, including Canada Research Chair, CRC President's Excellence Award, Canadian Federal Government Public Service Award, Ontario Early Researcher Award, and five IEEE Best Paper Awards. He currently serves as an Editor/Associate Editor for *IEEE Transactions on Vehicular Technology* and *IEEE Transactions on Broadcasting*. He has also served as Associate Editor/Guest Editor for a number of journals. He has been involved in a number of IEEE conferences including GLOBECOM, ICC, VTC, PIMRC, WCNC, and CWIT, in different roles such as Symposium Chair, Tutorial Instructor, Track Chair, Session Chair, and TPC Co-Chair. He is currently an IEEE Distinguished Lecturer.

Overcoming the Key Challenges to Establishing Vehicular Communication: Is SDN the Answer?

Ibrar Yaqoob, Iftikhar Ahmad, Ejaz Ahmed, Abdullah Gani, Muhammad Imran, and Nadra Guizani

The authors investigate recent premier research advances in the SDVN paradigm. Then they categorize and classify SDVN concepts and establish a taxonomy based on important characteristics, such as services, access technologies, network architectural components, opportunities, operational modes, and system components. They identify and outline the key requirements for SDVNs, and enumerate and outline future challenges in implementing SDVNs.

ABSTRACT

Considerable development in software-based configurable hardware has paved the way for a new networking paradigm called software-defined vehicular networks (SDVNs). The distinctive features of SDN, such as its flexibility and programmability, can help fulfill the performance and management requirements for VANETs. Although several studies exist on VANET and SDN, a tutorial on SDVNs is still lacking. In this article, we initially investigate recent premier research advances in the SDVN paradigm. Then we categorize and classify SDVN concepts and establish a taxonomy based on important characteristics, such as services, access technologies, network architectural components, opportunities, operational modes, and system components. Furthermore, we identify and outline the key requirements for SDVNs. Finally, we enumerate and outline future research challenges.

INTRODUCTION

Remarkable technological advances and the pervasive use of smart devices have realized vehicular ad hoc networks (VANETs), which help improve road safety and efficiency. Vehicular networks typically comprise various communication technologies, including dedicated short-range communication (DSRC), Wi-Fi, fourth generation (4G), 5G, and TV white space. Although these technologies can ensure reliable and ubiquitous mobile coverage, several salient features of VANETs introduce new challenges, such as unbalanced traffic flow in a multi-path topology and inefficient network utilization [1]. Thus, flexible and programmable architectures are key requirements for VANETs.

The convergence of software-defined networking (SDN) with VANET technology can play an important role in addressing most challenges. An illustration of a software-defined vehicular network (SDVN) is provided in Fig. 1. SDN introduces evolvability into VANETs to improve network efficiency. In addition, equipment and radio devices are simply reconfigured in SDN by adding a network programmability feature to vehicular networks through external applications. Consequently, SDN provides flexibility in developing vehicular network infrastructure. Other prominent features of SDN, such as dynamic network resource allocation and centralized control, can satisfy the requirements for VANETs. In the past, SDN-based advances have focused on data center networks, carrier backbone, and access networks. However, SDN/OpenFlow-based advances have recently turned to wireless scenarios. For example, OpenRoads [2] envisions that users will move between different types of wireless infrastructure. Cloud-medium access control (MAC) [3] offers virtual access points. The Wireless & Mobile Working Group of the Open Networking Foundation focuses on wireless backhaul, cellular Evolved Packet Core (EPC), and unified access and management across enterprise wireless and fixed networks (e.g., campus Wi-Fi) [1].

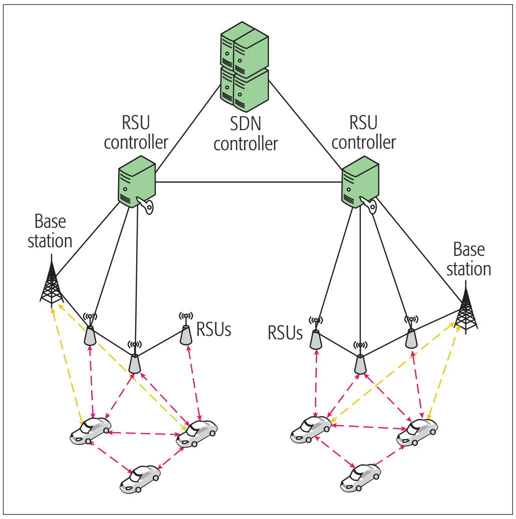


Figure 1. Illustration of a software-defined vehicular network.

The increasing interest in the SDN paradigm pushes for improved understanding of SDN/OpenFlow and the extension of its usage to VANETs . Such interest has motivated this study. Although several studies on VANETs and SDN have been conducted, the convergence of these two areas can be an impetus to academic efforts for promoting SDVNs. Thus, this study investigates recent advances in SDVNs, establishes an SDVN

Digital Object Identifier:
10.1109/MCOM.2017.1601183

Ibrar Yaqoob, IftikharAhmad, Ejaz Ahmed, and Abdullah Gani are with the Centre for Mobile Cloud Computing Research, Faculty of Computer Science and Information Technology, University of Malaya;; Muhammad Imran is with King Saud University; Nadra Guizani is with Purdue University.

Recent advances	Year	Reference	Aim	Use cases
Convergence of SDN with VANET	2016	[5]	Heterogeneous vehicular communication	SDN for heterogeneous vehicular networks
	2014	[1]	Survey on SDN	Programmable networks, SDN architecture and future
		[4]	SDN-based services	Safety services Road surveillance Virtualization
SDN for specific tasks in vehicular networks	2016	[9]	Delay-optimal virtualized radio resource scheduling	Radio resource virtualization
		[10]	Cost-efficient sensory data transmission	Abstraction for sensing data communication
		[11]	Control plane optimization	Transmission delay Cost reduction in cellular download
		[8]	Cooperative data scheduling	Cooperative data dissemination
	2015	[7]	QoE-based flow management	Data flow Interference
SDN and other technologies for emerging vehicular networks	2013	[6]	Improving network management	Control over network configuration
	2015	[12]	SDN for RSU clouds	RSU-based vehicular cloud
		[13]	SDN with fog	Data streaming Lane change services
	2014	[14]	SDN for IoT	SDN framework to handle heterogeneous tasks in the IoT paradigm

Table 1. Summary of the literature.

SDN enables the separation of the data plane from the control plane. This unique feature of SDN provides an abstraction of heterogeneity in vehicular networks, which simplifies network management and configuration by providing a standard interface to different devices within the network.

taxonomy, determines the key requirements for SDN to work efficiently with VANETs, and identifies several open research challenges. The contributions of this study are as follows:

- Recent premier research advances in SDVNs are investigated, highlighted, and reported.
- SDVN concepts are categorized and classified, and an SDVN taxonomy is established.
- The key requirements for SDVNs are identified and outlined.
- Future challenges are enumerated and outlined.

The remainder of this article is organized as follows. We present research advances in the SDVN paradigm. We discuss the established taxonomy. We highlight the key requirements for establishing SDVNs. We present challenges to SDVNs. Finally, we provide the concluding remarks.

Recent Advances

In this section, we investigate recent research efforts related to the SDVN paradigm. A summary of the literature is provided in Table 1.

Convergence of SDN with VANET

In the past decade, cloud computing, the Internet of Things (IoT), vehicular cloud computing (VCC), and the Internet of Vehicles (IoV) have become prominent emerging technologies. At present, these technologies are integrated into one another to explore and develop new services. SDN, which is a new form of programmable networking, is another important emerging technology. In particular, the OpenFlow standard presents an alternative to the implementation of SDN. An early survey on SDN [4] examined its applications and services, with certain research directions. However, this survey focused only on the general network, not on vehicular networks, and showed the early history of SDN.

SDN was proposed primarily for wired networks. At present, however, it is being incorporated into wireless networks and even into wireless ad hoc networks and mobile ad hoc networks, such as VANETs. A model [1] was proposed for the integration of SDN into the wireless environment of vehicular networks. The model solves the flexibility and scalability problems of VANETs by adopting SDN. SDN integrates a programmability feature into the control plane in VANETs to provide a new pool of services, such as surveillance, safety measures, and virtualization of network infrastructure. Furthermore, SDN operational modes enable VANETs to adapt to changes in network topology.

However, merging telecommunication and vehicular networks to form a heterogeneous vehicular network leads to heterogeneity and flexibility issues in the network. In this context, SDN enables the separation of the data plane from the control plane. This unique feature of SDN provides an abstraction of heterogeneity in vehicular networks, which simplifies network management and configuration by providing a standard interface to different devices within the network. The control plane allows rapid configuration management in view of the dynamic nature of vehicular networks and efficiently integrates multiple net-

Other emerging technologies that are suitable for SDN, such as VCC, fog computing, and IoT, are incorporated into SDN for vehicular networks. The integration of these technologies into SDN for vehicular networks provides enhanced synergistic capabilities and leads to new solutions.

work technologies [5]. Although SDN provides a certain degree of control over resource and network management, issues related to heterogeneity, mobility, and flexibility should still be explored further.

SDN- and 5G-Technology-Based Vehicular Networks: A 5G telecommunication network provides wireless communication infrastructure for vehicular communication [5]. 5G network-based vehicular networks typically comprise heterogeneous access technologies. In heterogeneous networks, an SDN controller can gain control over networks by adding a programmability feature to network devices through external applications. In this manner, SDN enables the flexible management of vehicular networks to address the problem of heterogeneity to a certain extent.

SDN for Specific Tasks in Vehicular Networks

The capabilities of SDN have recently been explored to handle specific tasks and issues related to vehicular networks.

Network Management: Network management in vehicular networks is difficult because of the ad hoc nature of such networks. The separation mechanism of SDN for the data and control planes paves the way for new possibilities to control and manage network behavior [6]. The differences in vendor-level specifications can be disregarded by implementing it with SDN methods. SDN improves network management by providing frequent changes to network conditions, high-level support for network configuration, visibility, and control over tasks related to network diagnosis and troubleshooting.

Transmission Interference: The behavior of VANETs is dynamic because of frequent topological changes. In this context, SDN and IEEE 802.11p can be combined to overcome the dynamic topological changes and interference in transmissions within vehicular networks. Efficient sharing of resources among vehicles is required to manage a network. The centralized control of SDN provides control over data flows and vehicle power. The quality-of-experience-based network configuration through SDN is achieved by imposing and controlling the behavior of a vehicular network [7]. However, control over network behavior via SDN may not have a sustainable positive impact on the performance of the network because of short-term connectivity.

Cooperative Data Dissemination: The short-term connectivity problem can be overcome by integrating other access networks into IEEE 802.11p. In this context, the integration of SDN into vehicular networks can provide a new solution. An SDN-based data scheduler in a roadside unit (RSU) cooperatively disseminates data in a heterogeneous vehicular environment [8]. The only drawback of the data scheduler is that it is based on the RSU, which has a high installation cost. However, this scheduler works efficiently, fulfills the data transmission requirements for vehicles, and enhances data propagation.

Delay Reduction: Transmission delay can be reduced by decreasing the complexity of a network and managing its radio resources. An optimal solution, called "software-defined heterogeneous vehicular network," was proposed in [9] to reduce transmission delay. A radio resource scheduling scheme was introduced to allocate radio resources accurately to decrease computational complexity and signaling overhead. Radio resource management in a heterogeneous vehicular environment depends on the location of the SDN controller in the network. Thus, the location of the controller is critical to managing network heterogeneity, which has not yet been addressed in the literature.

Cooperative Sensing: Improvements in information-sharing strategies can reduce the cost of information sharing. The efficiency of sensory data collection in a heterogeneous vehicular network can reduce the cost of data sharing [10]. SDN provides an abstraction of the heterogeneity of vehicular networks in wireless environments, and consequently promotes cost-effective data sharing among vehicles.

The Control Plane for a Vehicular Network: The integration of DSRC and other access networks has recently gained popularity in vehicular networks [11]. An SDN-based strategy is proposed to create a balance between delay and cost. The centralized control of a vehicular network results in the dependence of vehicular networks on other networks, such as Long-Term Evolution (LTE). The mobility and high speed of vehicles causes the SDN controller to lose control over VANETs because of intermittent connectivity. This problem may be avoided by involving other access networks, particularly those that possess a wide coverage range and the capability to manage the mobility of nodes. However, this solution is costly.

SDN Integration into Other Technologies for Emerging Vehicular Networks

Other emerging technologies that are suitable for SDN, such as VCC, fog computing, and IoT, are incorporated into SDN for vehicular networks. The integration of these technologies into SDN for vehicular networks provides enhanced synergistic capabilities and leads to new solutions.

SDN and Vehicular Cloud: The programmability feature of SDN can be integrated into the vehicular cloud to support IoV [12]. The data forwarding and services of the cloud are dynamically configured according to the demands of vehicles. The reconfiguration procedure may be costly for service providers when implemented using cloud resources. The reconfiguration overhead can be higher in vehicular networks because of the high mobility and speed of vehicles. The application of the reinforcement learning approach to reconfiguration can be cost effective and can reduce transmission delay.

SDN and Fog Computing: The challenges to establishing VANETs, such as short connectivity, inflexibility, and non-intelligence, can be overcome by combining SDN and fog computing [13]. SDN enables control over network behavior, whereas fog computing provides time- and location-based services. Thus, their integration optimizes resource utilization and reduces communication delay. The benefits of the proposed approach have been confirmed by use case scenarios of data streaming and lane change. However, resource utilization cannot be optimized without first addressing heterogeneity.

SDN and IoT: The convergence of SDN with

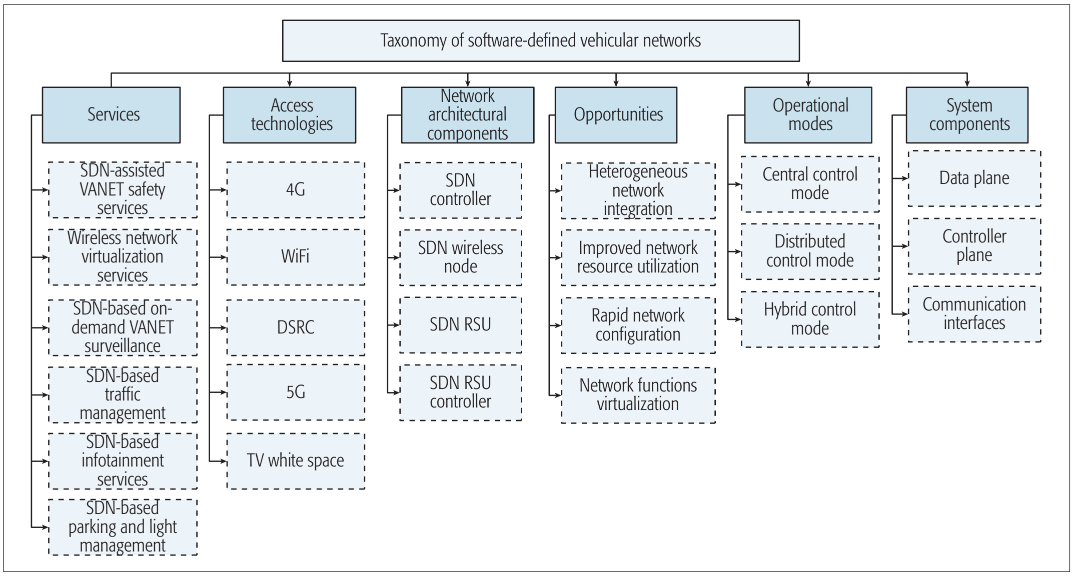

Taxonomy of software-defined vehicular networks
Services
SDN-assisted VANET safety services
Wireless network virtualization services
SDN-based on-demand VANET surveillance
SDN-based traffic management
SDN-based infotainment services
SDN-based parking and light management
Access technologies
4G
WiFi
DSRC
5G
TV white space
Network architectural components
SDN controller
SDN wireless node
SDN RSU
SDN RSU controller
Opportunities
Heterogeneous network integration
Improved network resource utilization
Rapid network configuration
Network functions virtualization
Operational modes
Central control mode
Distributed control mode
Hybrid control mode
System components
Data plane
Controller plane
Communication interfaces

Figure 2. Taxonomy of software-defined vehicular networks.

Infotainment services are likely to attract a huge mass market in vehicular networks and are expected to achieve sales of up to $131.9 billion by 2019. The use of SDN helps efficiently implement these services by separating the control traffic from the actual service data traffic.

IoT can also be beneficial to vehicular networks. SDN provides abstraction and automation, whereas IoT facilitates the connection of resources in a network. An extended SDN prototype was tested in an IoT scenario, in which the vehicular network, electric sites, grids, and set of pilot users were involved [14]. SDN was used to manage the distributed and heterogeneous environment, and consequently achieve differentiated quality levels of tasks in a dynamic scenario. A layered design for the SDN controller was proposed to provide flexibility.

Despite the numerous advantages of these emerging technologies, issues due to intermittent connectivity hinder the solving of problems related to vehicular networks. Thus, exploring these technologies from the appropriate perspective is necessary to solve persistent issues in vehicular networks, and eventually realize effective road traffic management.

Taxonomy

Figure 2 illustrates the SDVN taxonomy, which is established based on the following characteristics:

- Services
- Access technologies
- Network architectural components
- Opportunities
- Operational modes
- System components

Services: SDN-based vehicular networks provide the following key services:

- SDN-assisted VANET safety services
- Wireless network virtualization services
- SDN-based on-demand VANET surveillance
- SDN-based vehicular traffic management
- SDN-based infotainment services
- SDN-based parking and light management

SDN-assisted VANET safety services are provided by dynamically configuring flow rules and assigning them to switches while considering network conditions and the requirements for applications. Wireless network virtualization is the concept of virtualizing network resources or data paths to attain tenant or application segregation. Such segregation is typically required for different reasons, including fault isolation, network abstraction, scalability, and security.

Wireless network virtualization services are provided by applying the concept of SDVNs. SDN-based technologies, such as OpenFlow, are used to achieve path isolation.

SDN-based on-demand VANET surveillance services are provided through an SDN controller. This controller manages and inputs flow rules for data related to surveillance to reach the requesting vehicular nodes. The centralized controller-based approach of SDN facilitates the delivery of network traffic management services to network operators. Network-wide packet-forwarding decisions are made in the controller using all of the network information.

Infotainment services are emerging in vehicular networks, and they have attracted the attention of service providers. Infotainment services include, but are not limited to, advertisements, tourist information, traffic information, and parking directions. These services are likely to attract a huge mass market in vehicular networks and are expected to achieve sales of up to $131.9 billion by 2019. The use of SDN helps efficiently implement these services by separating the control traffic from the actual service data traffic.

SDN-based vehicular networks can also provide smart parking and light management services. In such services, thousands of luminaires, sensors, and cameras form a network in a cloud environment to provide automated parking and lighting services, improve flexibility, and provide information in real time.

Access Technologies: 4G is an access technology that is used in SDN-based vehicular networks. Two potential 4G standards are commercially used: LTE and Mobile WiMAX. LTE can be used to fulfill the requirements for delay-sensitive applications in SDN-based vehicular networks. Mobile WiMAX is suitable for vehicular network applications because it supports high-speed mobility and has wide coverage; consequently, it minimizes network disruptions.

5G-based vehicular networks can support a wide array of applications and realize vehicular

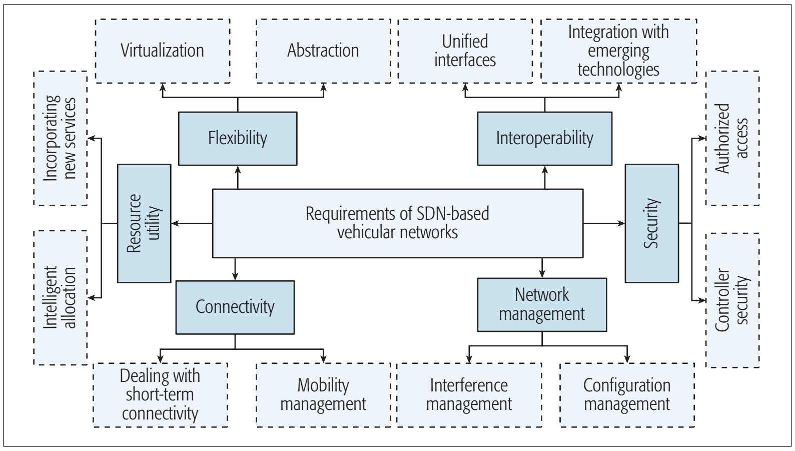


Figure 3. Key requirements for SDVN.

networking for traditional multimedia applications. TV white spaces are also used by vehicular networks with cognitive radio technology to improve efficiency. Vehicular network users utilize the available spectrum in TV bands to fulfill the quality of service (QoS) requirements for applications. DSRC is a point-to-point short-range communication technology. The DSRC Working Group is currently developing a standard for wireless access in vehicular environments, where communication is based on the IEEE 802.11p standard, which is an enhanced version of the Wi-Fi standard IEEE 802.11.

Network Architectural Components: The network architectural components of SDN-based vehicular networks are:
- The SDN controller
- SDN wireless nodes
- SDN RSUs
- The SDN RSU controller

The SDN controller is a key element of SDN-based vehicular networks. It manages the behavior of the entire network. It also controls the flow to routers/switches via southbound interfaces and to applications via northbound interfaces. SDN wireless nodes in SDN-based vehicular networks are the vehicles that perform actions based on the control messages received from the SDN controller. They are the mobile data plane elements that are controlled by the SDN controller. SDN RSUs are stationary structures in SDN-based vehicular networks. RSUs are controlled by the SDN controller and deployed along roadsides. The SDN RSU controller controls a cluster of RSUs that are connected to it through a broadband link. This controller is an OpenFlow-based infrastructure that is responsible for forwarding data, performing emergency services, and storing local information.

Opportunities: Vehicular networks may rely on several heterogeneous wireless networking technologies that pose challenges to fulfilling different QoS requirements for vehicular transport services. Traditional vehicular networks cannot handle the growing demands of a highly dynamic network environment. In contrast, the flexible centralized nature of SDN supports the dynamic environment of vehicular networks and minimizes management overhead. An SDN-based load balancer can help balance traffic load in RSUs to efficiently utilize available resources in SDVNs. The programmability feature of SDN can facilitate the rapid and automatic configuration of vehicular networks. In addition, SDN provides an opportunity for network service providers to deploy network elements in software form.

Operational Modes: The operational modes of SDN-based vehicular networks can be classified into three:
- Central
- Distributed
- Hybrid control modes

In central control mode, the flow rules on how to handle traffic are enforced using the SDN controller. In this mode, the SDN controller manages all the actions performed by SDN wireless nodes, SDN RSUs, and the SDN RSU controller. In distributed control mode, SDN wireless nodes and SDN RSUs are not guided by the SDN controller. This control mode is similar to that in traditional distributed self-organizing networks. In hybrid control mode, the SDN controller does not completely control SDN wireless nodes and SDN RSUs. Instead, it defines only the general policy rules and not all the flow rules. SDN data elements, SDN wireless nodes, and SDN RSUs apply their own intelligence to forward packets and execute flow-level processing.

System Components: The system components of SDN-based vehicular networks are as follows:
- Data plane
- Control plane
- Communication interfaces

The data plane handles packets in accordance with the instructions of the control plane. The tasks performed by the data plane include, but are not limited to, changing, dropping, and forwarding packets. The data plane also provides forwarding resources, such as classifiers. Accordingly, it is also called the forwarding plane. The control plane makes decisions regarding the packet forwarding of network devices and communicates packet forwarding rules to network switches and routers for execution. The control plane is mainly responsible for updating the forwarding table, which is on the data plane, considering the network topology. In the SDN architecture, there are two types of communication interfaces: a) northbound interface and b) southbound interface. The northbound interface allows the control plane and management plane to interact with the application plane, whereas the southbound interface allows the control plane and management plane to interact with the network device.

Requirements for SDN-Based Vehicular Networks

This section outlines the requirements for SDVNs. A detailed description of the requirements is presented in Fig. 3.

Flexibility: SDN deployment requires flexibility at the back end of a VANET. Once SDN is deployed in a VANET, it can systematically manage and configure vehicular networks. SDN virtualizes the network and provides an abstraction of the configuration of devices and nodes in vehicular networks.

Resource Utility: SDVNs require allocating appropriate resources at the right time because resources are available on an ad hoc basis and must be used efficiently and effectively. Although the separate control plane of SDN intelligently allocates resources among vehicles, context-aware resource allocation should be enabled to ensure the successful management of road traffic and vehicular networks.

Connectivity: The problems of short-term connectivity and high mobility rate due to the high speed of vehicles are the remaining concerns that hinder the adoption of SDVNs. Although SDN-based mobility management solutions can help address these problems to a certain extent, these solutions are still immature. Thus, short-term connectivity remains a hurdle that has yet to be addressed to maximize SDN incorporation.

Network Management: As mentioned earlier, the separate control panel of SDN allows for the management and configuration of networks. Although SDN can provide management capabilities to vehicular networks, the dynamic topology of these networks and their interference still need to be addressed.

Security: The strengthening of security and privacy is a key requirement for SDVNs. An SDVN controller should be completely secure because it performs central control over the network. The failure of a controller or the propagation of misinformation can lead to serious road accidents. Only a secure SDN controller can ensure the reliability of overall network operation, which can only be realized by ensuring authorized access.

Interoperability: SDN interfaces should be adequately unified for effective internetwork communication and operation. Abstraction and virtualization are helpful in hiding the heterogeneous details of different networks and vehicles. The incorporation of other emerging technologies, such as VCC and IoT, typically require operational independence from network type and devices [15].

SDN interfaces should be adequately unified for effective internetwork communication and operation. Abstraction and virtualization are helpful in hiding the heterogeneous details of different networks and vehicles. The incorporation of other emerging technologies, such as VCC and IoT, typically require operational independence from network type and devices.

Open Research Challenges

This section discusses key challenges remaining to be addressed to SDVNs. The discussion aims to provide directions to new researchers in the domain.

Mobility Management: The high mobility of vehicles causes a change in SDVN topology and instability in wireless channels. High mobility also hinders the real-time collection of the information of vehicles and the network using the controller. Thus, the controller experiences delays in distributing commands. Efficient control for high mobility management is a significant concern that requires serious attention to promote the adoption of SDVNs. Although several solutions have been proposed to address this challenge, these solutions are still in their infancy and cannot be adopted in SDVNs. The inclusion of the movement behavior of vehicles in predicting network stability can be a solution for the high mobility problem. However, this endeavor is challenging.

Internetworking among Heterogeneous Networks: In VANETs, various types of networks are involved to ensure connectivity among vehicles. However, the lack of efficient internetworking mechanisms leads to connectivity issues among heterogeneous networks in a vehicular network. In SDVNs, interconnection among heterogeneous networks has become a challenge because of the lack of standardized eastbound/westbound application programming interfaces (APIs) and northbound APIs for vehicular applications. The introduction of network functions virtualization to an abstract infrastructure layer has been proposed to overcome this challenge . However, the addition of this new layer introduces new challenges regarding its compatibility with other layers. Thus, considerable attention is required to solve the internetworking issue in the future.

Extent to Which a VANET Should Be Software Defined: The process of determining the extent to which a VANET should be based on SDN is challenging. When only the wired part is changed, no noticeable change will be produced. However, transforming the entire VANET into an SDN-based network is an inefficient solution. Accordingly, a comprehensive performance evaluation should be conducted to identify which control intelligence can be decoupled from the data plane to maximize the benefits of SDN in VANETs. However, the unavailability of testbeds and simulation tools hinders such evaluation. In the future, serious attention must be given to the development of real testbeds and simulation tools for the performance evaluation of vehicles under SDN to determine which control intelligence should be decoupled from the data plane.

Security: In SDVNs, the propagation of misinformation from unauthorized entities can lead to serious accidents. Therefore, security is one of the key concerns that require serious attention. The controller should be protected because it is the centralized decision point in SDVNs. Several threats compromise the forwarding, control, and application layers. First, man-in-the-middle attacks between a switch and the controller are caused by the lack of transport layer security. This lack of security allows adversaries to infiltrate OpenFlow networks undetected. This type of attack can be mitigated by strengthening physical network security. Second, denial-of-service attacks can saturate the flow table and flow buffer. These attacks are caused by the insertion of reactive rules instead of adopting a proactive approach. They can be prevented by using multiple controllers, as in the case of Onix. Other threats include threats based on distributed multi-controllers, threats from applications, illegal access, and security rules and configuration conflicts. Although several solutions have been proposed, these solutions cannot be directly adopted in VANETs because they have different characteristics. The high mobility nature of VANETs requires security mechanisms that can perform real-time authentication; otherwise, latency can cause traffic congestion that impedes the realization of SDVNs. This real-time factor increases difficulty in strengthening security.

Conclusion

The convergence of SDN with VANETs has led to the development of a new computing paradigm called SDVN, which has received considerable attention from the academic and information technology communities. The desirable features of SDN can help overcome most of the limitations of VANETs. This study aims to explore SDVNs. In this article, recent premier research advances in the SDVN paradigm are investigated, highlighted, and reported. SDVN concepts are categorized and classified, and a taxonomy of SDVNs is established based on important characteristics. The key requirements for SDVNs are identified and outlined. Furthermore, several challenges that should be addressed to promote SDVN implementation are discussed. These challenges could serve as future research directions. We conclude that although SDN deployment in vehicular networks can extend network management capabilities and solve numerous challenges in traditional VANETs, the convergence of these two networks engenders several new challenges that should be addressed in the future.

Although SDN deployment in vehicular networks can extend network management capabilities and solve numerous challenges in traditional VANETs, the convergence of these two networks engenders several new challenges that should be addressed in the future.

Acknowledgments

This work is funded by the Bright Sparks Unit and a High Impact Research Grant from the University of Malaya under references UM.C/625/1/HIR/MOE/FCSIT/03 and RP012C-13AFR. Imran's work is supported by the Deanship of Scientific Research at King Saud University through Research Group No. (RG # 1435-051).

References

[1] I. Ku *et al.*, "Towards Software-Defined VANET: Architecture and Services," *2014 13th Annual Mediterranean Ad Hoc Networking Wksp.*, 2014, pp. 103–10.

[2] K.-K. Yap *et al.*, "Openroads: Empowering Research in Mobile Networks," *ACM SIGCOMM Comp. Commun. Review*, vol. 40, no. 1, 2010, pp. 125–26.

[3] J. Vestin *et al.*, "Cloudmac: Towards Software Defined WLANs," *ACM SIGMOBILE Mobile Computing and Commun. Review*, vol. 16, no. 4, 2013, pp. 42–45.

[4] B. A. A. Nunes *et al.*, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Commun. Surveys & Tutorials*, vol. 16, no. 3, 2014, pp. 1617–34.

[5] Z. He, J. Cao, and X. Liu, "SDVN: Enabling Rapid Network Innovation for Heterogeneous Vehicular Communication," *IEEE Network*, 2016, p. 3.

[6] H. Kim and N. Feamster, "Improving Network Management with Software Defined Networking," *IEEE Commun. Mag.*, vol. 51, no. 2, Feb. 2013, pp. 114–19.

[7] E. Bozkaya and B. Canberk, "Qoe-Based Flow Management in Software Defined Vehicular Networks," *2015 IEEE GLOBECOM Wksps.)*, 2015, pp. 1–6.

[8] K. Liu *et al.*, "Cooperative Data Scheduling in Hybrid Vehicular Ad Hoc Networks: VANET as a Software Defined Network," *IEEE/ACM Trans. Net.*, vol. 24, no. 3, June 2016, pp. 1759–73.

[9] Q. Zheng *et al.*, "Delay-Optimal Virtualized Radio Resource Scheduling in Software-Defined Vehicular Networks Via Stochastic Learning," *IEEE Trans. Vehic. Tech.*, vol. 65, no. 10, Oct 2016, pp. 7857–67.

[10] Z. He, D. Zhang, and J. Liang, "Cost-Efficient Sensory Data Transmission in Heterogeneous Software-Defined Vehicular Networks," *IEEE Sensors J.*, vol. 16, no. 20, Oct. 2016, pp. 7342–54.

[11] H. Li, M. Dong, and K. Ota, "Control Plane Optimization in Software-Defined Vehicular Ad Hoc Networks," *IEEE Trans. Vehic. Tech.*, vol. 65, no. 10, Oct 2016, pp. 7895–7904.

[12] M. A. Salahuddin, A. Al-Fuqaha, and M. Guizani, "Software-Defined Networking for RSU Clouds in Support of the Internet of Vehicles," *IEEE Internet of Things J.*, vol. 2, no. 2, 2015, pp. 133–44.

[13] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software Defined Networking-Based Vehicular Adhoc Network with Fog Computing," *2015 IFIP/IEEE Int'l. Symp. Integrated Network Management*, 2015, pp. 1202–07.

[14] Z. Qin *et al.*, "A Software Defined Networking Architecture for the Internet-of-Things," *2014 IEEE Network Operations and Management Symp.*, 2014, pp. 1–9.

[15] A. Al-Fuqaha *et al.*, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surveys & Tutorials*, vol. 17, no. 4, 2015, pp. 2347–76.

Biographies

Ibrar Yaqoob (ibraryaqoob@siswa.um.edu.my) received his Ph.D. degree in computer science from the University of Malaya, Malaysia, in 2017. He earned 550 plus citations, and 50 plus impact factor during his Ph.D. candidature. He worked as a researcher at the Centre for Mobile Cloud Computing Research (C4MCCR), University of Malaya. His research experience spans over more than three and half years. He has published a number of research articles in refereed international journals and magazines. His numerous research articles are very famous and among the most downloaded in top journals. His research interests include big data, mobile cloud, the Internet of Things, cloud computing, and wireless networks.

Iftikhar Ahmad (ify_ia@yahoo.com) is a Ph.D. student in the Computer System & Technology Department at the University of Malaya. He is also working as a lecturer at Mirpur University of Science and Technology, Pakistan. His research interests include communication networks, vehicular ad hoc networks, and the Internet of Vehicles.

Ejaz Ahmed (ejazahmed@ieee.org) is a senior researcher in the High Impact Research project at the Centre for Mobile Cloud Computing Research, University of Malaya. Before that, he worked as a research associate in the Cognitive Radio Network Research Lab SEECS, NUST Islamabad from December 2009 to September 2012, and at the Center of Research in Networks and Telecom, MAJU, Islamabad, from January 2008 to December 2009. His research experience spans over more than nine years. He is an Associate Editor of *IEEE Communications Magazine*, *IEEE Access*, and *Wiley Wireless Communications and Mobile Computing*. He has also served as a Lead Guest Editor/Guest Editor of *Elsevier Future Generation Computer Systems*, *Elsevier Computers & Electrical Engineering*, *IEEE Communications Magazine*, *IEEE Access*, *Elsevier Information Systems*, and *Wiley Transactions on Emerging Telecommunications Technologies*. His areas of interest include mobile cloud computing, mobile edge computing, the Internet of Things, cognitive radio networks, and smart cities. He has successfully published his research work in more than 30 international journals and conferences.

Abdullah Gani [M'01, SM'12] (abdullahgani@ieee.org) is a full professor in the Department of Computer System and Technology, University of Malaya. He received his Bachelor's and Master's degrees from the University of Hull, United Kingdom, and his Ph.D. from the University of Sheffield, United Kingdom. He has vast teaching experience due to having worked in various educational institutions locally and abroad: schools, teaching college, the Ministry of Education, and universities. His interest in research started in 1983, when he was chosen to attend a Scientific Research course in RECSAM by the Ministry of Education, Malaysia. More than 150 academic papers have been published in conferences and respectable journals. He actively supervises many students at all levels of study — Bachelor, Master, and Ph.D. His research interests include self-organized systems, reinforcement learning, and wireless-related networks. He worked on mobile cloud computing with a High Impact Research Grant for the period of 2011–2016.

Muhammad Imran (dr.m.imran@ieee.org) is an assistant professor in the College of Computer and Information Science, King Saud University. His research interests include mobile ad hoc and sensor networks, WBANs, IoT, M2M, multihop wireless networks, and fault-tolerant computing. He has published a number of research papers in peer reviewed international journals and conferences. His research is financially supported by several grants. He is serving as a Co-Editor-in-Chief for *EAI Transactions on Pervasive Health and Technology*. He also serves as an Associate Editor for the *Wireless Communication and Mobile Computing Journal* (Wiley), the *Interscience International Journal of Autonomous and Adaptive Communications Systems*, *Wireless Sensor Systems* (IET), and the *International Journal of Information Technology and Electrical Engineering*. He has served/serves as a Guest Editor for *IEEE Communications Magazine*, *IJAACS*, and the *International Journal of Distributed Sensor Networks*. He has been involved in a number of conferences and workshops in various capacities such as Program Co-Chair, Track Chair/Co-Chair, and Technical Program Committee member. These include IEEE GLOBECOM, ICC, AINA, LCN, IWCMC, IFIP WWIC, and BWCCA. He has received a number of awards such as an Asia Pacific Advanced Network fellowship.

Nadra Guizani (nguizani@purdue.edu) is a Ph.D. student and a graduate lecturer in the Electrical and Computer Engineering Department at Purdue University. Her research work is on data analytics and prediction and access control of disease spread data on dynamic network topologies. Research interests include machine learning, mobile networking, large data analysis, and prediction techniques. She is an active member of the Women in Engineering Program.

Software Defined Architecture for VANET: A Testbed Implementation with Wireless Access Management

Gökhan Seçinti, Berk Canberk, Trung Q. Duong, and Lei Shu

ABSTRACT

Toward ITS, academia and industry aim to utilize all possible radio access technologies in order to support reliable services and applications in VANETs. Thus, the inclusion of already deployed Wi-Fi networks in VANET topology is a crucial step for the next generation vehicular networks. However, the VANET topology also requires preservation of the features already offered by DSRC and the core cellular network. As a result, the coexistence of multiple different access technologies results in high complexity in terms of the control and management of the network infrastructure. To this end, software defined networking provides a promising opportunity to simplify the management and control of clumsy network infrastructures by decoupling the data and control planes in order to provide elasticity for current networks. In this article, we propose an architectural model that exploits this opportunity in order to enhance VANET with Wi-Fi access capability. Moreover, we offer a novel software defined VANET architecture that consists of soft OpenFlow switches with Wi-Fi capabilities as both roadside units and vehicles. In particular, we first investigate existing test tools and environments for software defined wireless networks and also supply a novel testbed architecture in order to provide a feasible test environment for evaluating the proposed architecture. Additionally, we propose a Wireless Access Management (WAM) protocol that provides wireless host management and basic flow admission with respect to the available bandwidth to validate the capability of the offered architecture. The observation results of the deployed testbed prove the conformity of the offered 802.11 architecture to the VANET.

INTRODUCTION

Intelligent transportation systems (ITS) are supported by governments, industry, and academia in order to provide safer and more efficient transportation environments [1]. However, the high mobility of traffic and the dynamic nature of the communication environment has stood as a great challenge to reliable vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communication to improve efficiency of transportation systems. For this reason, vehicular ad hoc networks (VANETs) aim to leverage various radio access technologies cooperatively to meet these demands while overcoming the challenges.

The already widely deployed 802.11 (Wi-Fi) networks are perfect candidates for VANETs. However, the growth of networks and simultaneous use of different radio access technologies present severe difficulties in terms of infrastructure control and management. Specifically, in large-scale deployment of legacy wireless networks, a significant amount of the operational expense (OpEx) is spent on the management of infrastructure devices. Major challenges encountered in mobile wireless access networks include the configuration of the wireless access points, the management of wireless hosts, forwarding the traffic generated by wireless hosts, and management of the radio resources, such as dynamic channel assignment and power planning. Several approaches have been proposed to overcome the aforementioned problems; however, they are not without disadvantages. The existing wireless management solution, the Control and Provisioning of Wireless Access Points (CAPWAP) protocol [2], lacks a traffic offload mechanism. Furthermore, bandwidth and the processing effort spent ON the control plane traffic in CAPWAP becomes underutilized because of traffic overhead.

As an alternative approach, software defined networking (SDN) has been adopted. SDN separates the network control function (the control plane) from the forwarding functions (the data plane), allowing us to control all network functionalities by a controller in a centralized manner. This approach enables an opportunity for a manageable network where it is feasible to implement new protocols. Additionally, SDNs have the ability to install rules to the end devices once the forwarding path for a specific traffic type has been discovered by the controller and installed as a flow to the flow table of the OpenFlow Switch. The data traffic can then be offloaded from the controller. With all the advantages, SDN is considered as a strong solution candidate for wireless access management.

A software defined VANET (SD-VANET) testbed includes soft-switches at the data plane, which are composed of Raspberry Pi as hard-

Software defined networking provides a promising opportunity to simplify the management and control of clumsy network infrastructures by decoupling the data and control planes in order to provide elasticity for current networks. The authors propose an architectural model that exploits this opportunity in order to enhance VANET with Wi-Fi access capability.

Gökhan Seçinti and Berk Canberk are with Istanbul Technical University; Trung Q. Duong is with Queen's University Belfast; Lei Shu is with Guangdong University of Petrochemical Technology.

Digital Object Identifier: 10.1109/MCOM.2017.1601186

0163-6804/17/$25.00 © 2017 IEEE

SD-Wireless Access Management simply benefits this virtualization capability of the soft-switches and provides wireless access virtualization in order to realize control and management mechanisms without increasing signalling overhead or introducing any middle-layers to existing OpenFlow protocol.

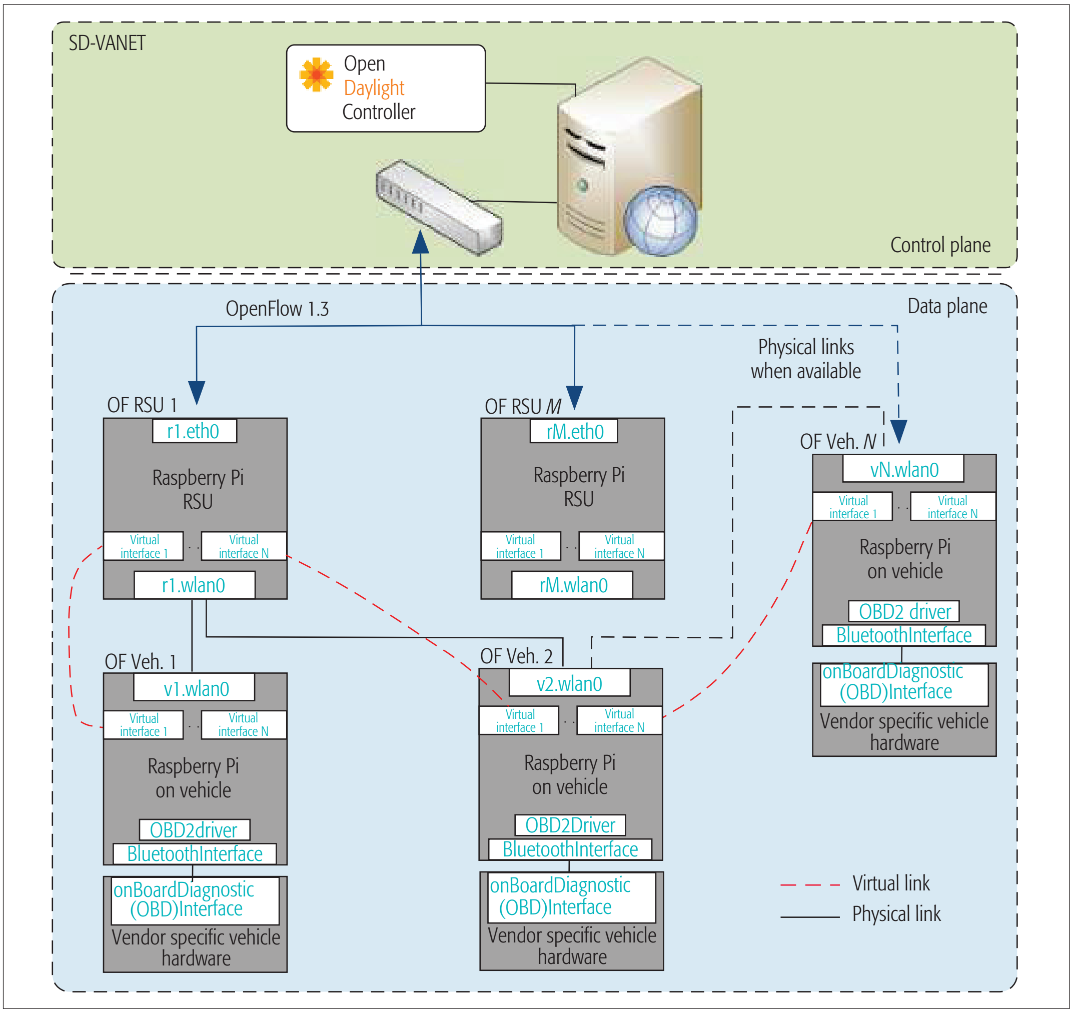


Figure 1. The proposed software defined VANET architecture.

ware and OpenvSwitch (OVS) as the main software component to work as both a roadside unit (RSU) and a vehicle access point. Moreover, these soft-switches are enhanced with virtual port capability, which defines a virtual wireless access port for each client (vehicle and RSU). This virtual port capability provides improved interface to the controller plane, which enables control and management commands to be implemented. Consequently, SD-Wireless Access Management (SD-WAM) simply enhances this virtualization capability of the soft-switches and provides wireless access virtualization in order to realize control and management mechanisms without increasing signaling overhead or introducing any middle layers to the existing OpenFlow protocol.

Utilizing SDN architecture in vehicular and mobile networks in order to benefit currently deployed network infrastructure has already drawn the attention of academia in recent years. In [3], the authors provided an overview for software defined mobile networks and identified current research approaches and challenges. In [4], the authors proposed a framework based on the cloud radio access network (Cloud-RAN) in order to utilize multiple RAN technologies through virtulization in vehicular networks. Moreover, in [5], the authors proposed a software defined mechanism named SERVICE to propose delay optimality. In this manner, they formulated the delay optimality through virtual resource scheduling by using a partially observed Markov decision process. Also, the authors defined a novel dissatisfaction parameter and proposed an offloading mechanism in order to satisfy quality of service (QoS) demands of mobile users in software defined heterogeneous networks in [6]. In addition, the authors in [7] proposed to utilize a slicing mechanism simply based on vehicle driving directions in order to achieve multi-tenant isolation in vehicular networks. However, none of the above studies focused on building an SD-VANET to enable rapid deployment and testing of high-level applications and services.

Although these approaches seem beneficial, they also require huge changes and definitions of new protocols in order to be implemented.

Additionally, there are various studies which focus on the network virtualization approach in SDN in order to tackle management challenges in the wireless environment. In [8], virtualization strategies for the wireless environment were investigated and a novel resource description methodology was proposed to better address challenges in software defined wireless networks. Moreover, in [9], the authors defined and implemented a network virtualization mechanism called "FlowVisor," which introduces a middleware between the data and control planes and creates an opportunity to orchestrate different underlying networks in a single physical data plane. However, this middleware requires computational capabilities to reshape the control packets among the two planes. Thus, it creates another bottleneck problem for SDN architecture where there is already a troublesome bottleneck problem due to the singularity of the controller. To overcome this problem, the authors in [10] proposed a distributed version of FlowVisor. However, their method significantly increases the latency of the control packets when distributing the load of FlowVisor since a database is introduced in order to distribute the knowledge among middle entities.

As new technologies emerge, it is getting complicated for researchers to implement and test their novel protocols and approaches on current physical infrastructures, whose functions are restricted by their vendor-specific nature. The gap between academia and industry has been widened with the difficulty of testing and analyzing new protocols for every new technology, there are a limited number of studies that focus on filling this gap by introducing testbeds and analytical tools for various environmental settings. In [11], a West-East-bridge-based testbed for SDN is introduced in order to manage different domains and inter-domain communication. Two different inter-domain routing protocols to validate their testbed were also proposed. Similarly, the authors designed a testbed to analyze mobile crowd sensing by building a test network in the University of Bologna comprising 300 students as participants in [12]. Furthermore, in [13], the authors investigated existing experimental platforms and testbeds for wireless ad hoc networks and provided insights about the current limitations and challenges for future studies. But none of these testbeds specifically focused on how to provide a testing environment for an SDN that consists of wireless channels as the access network.

Motivated by the above discussion, in this article, we propose an SD-VANET testbed architecture that utilizes already deployed WiFi networks in the Istanbut Technical University (ITU) campus. Moreover, we implement software defined network-WAM (SDN-WAM) to validate our testbed. Our proposed architecture aims to minimize the necessity of changes in current network infrastructure; that is, the conjunction of mobile networks and WiFi networks could be handled with the software defined controller without changing any of the underlying infrastructure. More importantly, our SDN-WAM scheme does not necessitate any change at the control-data plane interface (CDPI), and thus does not increase the latency of the control packets.

The rest of the article is organized as follows. The architecture and the implementation details of the D-VANET testbed are given. Then the system model of SDN-WAM is explained, and the performance of DN-WAM is evaluated. Finally, we conclude the article by summarizing the contributions and emphasizing future work.

SD-VANET Testbed

Architecture

The SD-VANET topology depicted in Fig. 1 consists of two components responsible for handling separate domains of the communication network: the control plane and data plane. The control plane is responsible for supplying the network services required by the access networks, while the data plane is responsible for the forwarding of data packets that match the rules installed by the controller.

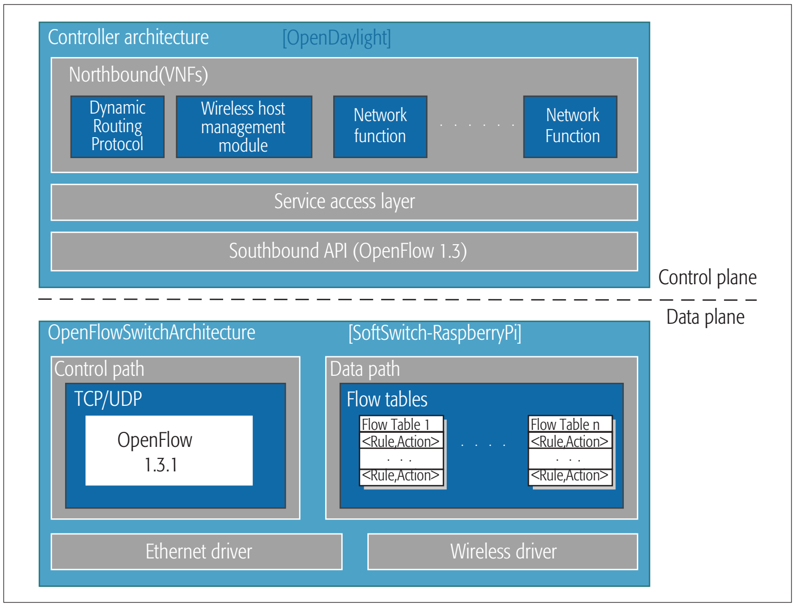


Figure 2. SD-VANET software architecture.

The basic components of an SD-VANET network are the controller and OpenFlow switches. Furthermore, the controller is a software application running on a powerful server computer that has a standard southbound interface such as OpenFlow to communicate with access devices, as depicted in Fig. 1. OpenFlow switches are the access devices of the SD-VANET placed in the backbone of the data plane. As seen in the figure, both RSU and vehicles are modeled by using Raspberry Pi with WiFi capability. There are two types of links defined between vehicles. While physical links represent the actual connection between network interface cards, virtual links provide an opportunity to define overlay networks on which different services and applications are utilized. To this end, SDN-WAM is implemented to validate use of virtual links. In addition, Raspberry Pi on a vehicle is connected through an onboard diagnostic (OBD) interface to the hardware of the vehicle. In this manner, information on the dashboard of the vehicle could be fetched with vendor-specific software.

Our testbed includes both of the aforementioned components of SD-VANET. Moreover, the testbed enables researchers to implement and measure their work, which utilizes either the controller or the OpenFlow Switch.

Implementation

Controller–OpenDaylight: Fundamental requirements of the controller are.

- A southbound application programming interface (API) as seen in Fig. 2 that supports OpenFlow standard to communicate with OpenFlow switches.
- A northbound API that is easy to use and enables rapid development of new network functions.
- Reconfiguration capability

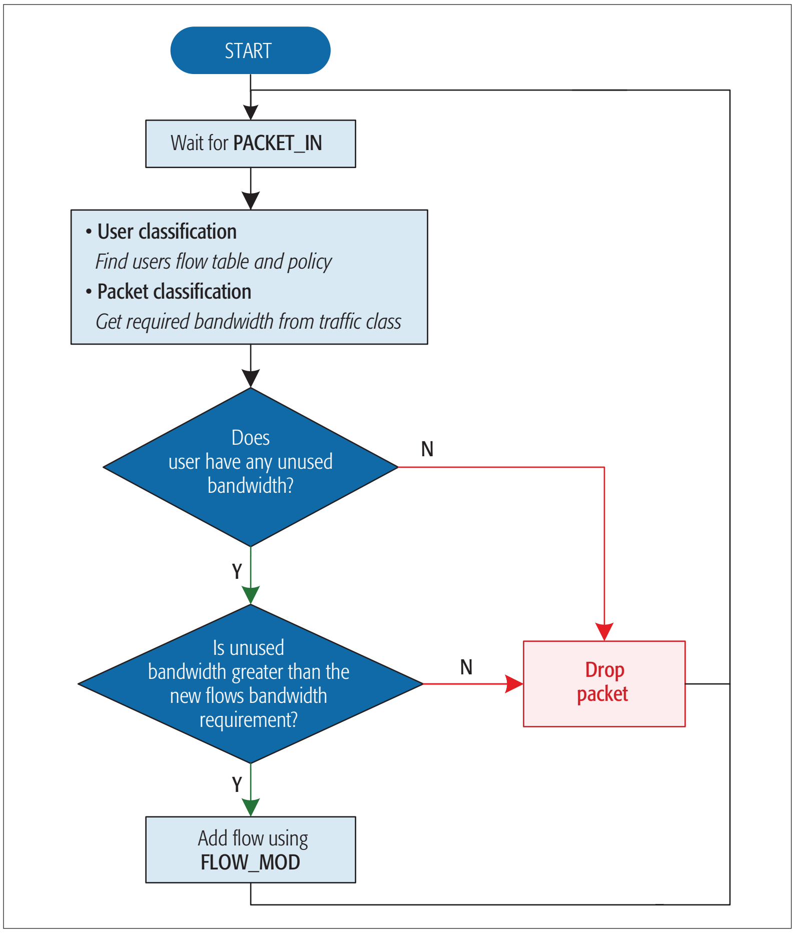

START
Wait for PACKET_IN
• User classification
Find users flow table and policy
• Packet classification
Get required bandwidth from traffic class
Does user have any unused bandwidth?
N
Y
Is unused bandwidth greater than the new flows bandwidth requirement?
N
Drop packet
Y
Add flow using FLOW_MOD

Figure 3. Flow admission implemented on proposed SD-VANET testbed.

The OpenDaylight [14] controller's southbound API supports the OpenFlow protocol. In addition, it is also compatible with the OpenFlow 1.3.1 specification. Moreover, the OpenDaylight controller also provides easy-to-use JAVA APIs to implement network functions on the northbound interface, as seen in Fig. 2. The OpenDaylight controller community consists of technology pioneers of communication networks [14], including Cisco, HP, Extreme Networks, and so on. Having a huge number of high-quality members in the community provides a native support chain for the OpenDaylight controller, which reduces the time required to overcome the obstacles encountered during development. Finally, the OpenDaylight controller's full source code is available free of charge, which allows researchers to implement any kind of customization required during development.

The OpenDaylight controller is installed on a server computer and used as the controller of the testbed because of the aforementioned features.

OpenFlow Switch–OpenvSwitch-Based Raspberry Pi: OpenFlow switches are the access components of the SD-VANET. In addition, the main functionalities of this component are as follows:

- To provide the service chaining infrastructure by supplying flow tables for the controller to install the required <Rule,Action> pairs
- To relay the packets received from the data plane with no matching service chaining rule to the controller using PACKET IN messages defined in OpenFlow to be assessed by the flow admission control algorithm, as seen in Fig. 3

Soft OpenFlow switch implementations are designed to cope with general-purpose inexpensive hardware. As the flow matching algorithm works on a traditional CPU rather than a parallel hardware such as a field programmable gate array (FPGA), a delay is added to the packet forwarding in soft-switch-based SD-VANET. This may lead to a tendency to increase the memory requirements of the general-purpose hardware to handle heavy traffic. For this reason, soft-switches are more likely to be deployed in lightly loaded networks. However, these devices are suitable for gathering useful measurement information for researchers and to prove the applicability of the offered solution to real-time environment even with the software reconfiguration capabilities.

We use Raspberry Pi as the hardware component of the OpenFlow soft-switch implementation. Furthermore, the expansion capability of the device also affected the final decision. The inexpensive hardware makes the device suitable to be used as an OpenFlow switch.

There are two major soft-switch implementations:

- OpenvSwitch(OVS): [1] An open source soft-switch implementation that is natively supported by the Linux kernel. In addition, as the software has integrated components running at the operating system level, this software facilitates the resources in a better way by augmenting the flow tables in Fig. 2 into the existing network device driver.
- CPQD softswitch:[2] An open source user space application that uses tunnels to obtain data from the operating system. This software is useful for inspecting the control plane traffic when there is no need to analyze and orchestrate the traffic.

OpenvSwitch v2.3.90 is ported for Raspberry Pi as the soft-switch implementation of the testbed. OVS provides both the flow table implementation and the OpenFlow communication structure. The Linux kernel is used as the network operating system of the OpenFlow switch. Moreover, the most common Linux distribution used on network devices, OpenWRT [15], is used as the file system of the Raspberry Pi OpenFlow switch. Furthermore, the existence of Raspberry Pi support in OpenWRT makes the file system an even better solution for the testbed. The head version of OpenWRT is used alongside Linux Kernel 4.1. The file system is configured to be compiled with eglibc 2.22 to support the functionalities required by the OVS. The distribution is modified to support a Realtek 5370 Wi-Fi dongle driver. Finally, the Realtek 5370 Wi-Fi dongle is physically attached the device as depicted in Fig. 4, which allows the wireless capability to be attached to the OpenFlow Switch.

Software-Defined-Networking-Based Wireless Access Management

SDN-WAM creates a virtual topology that facilitates the management of the wireless hosts using existing standardized OpenFlow messages. The solution virtualizes each wireless client as an inter-

[1] http://openvswitch.org, accessed on Jan 20, 2106

[2] http://cpqd.github.io/ofsoftswitch13/, accessed on Jan 20, 2106

face of an OpenFlow Switch, as seen in Fig. 1. In addition, management of connected clients are bound to the PORT MOD messages defined in OpenFlow. A PORT MOD message is first introduced in OpenFlow Spec. 1.3.0 and gives the controller the ability to shut the ports of an OpenFlow switch. The offered scheme uses port shutdown messages to provide the controller with the ability to disconnect wireless clients from the OpenFlow switch. In this way, the southbound interface of the controller is preserved as defined in the OpenFlow standard. Furthermore, wireless client virtualization also allows the SDN controller to apply different management policies easily to each client even though clients are connected to the same physical wireless interface.

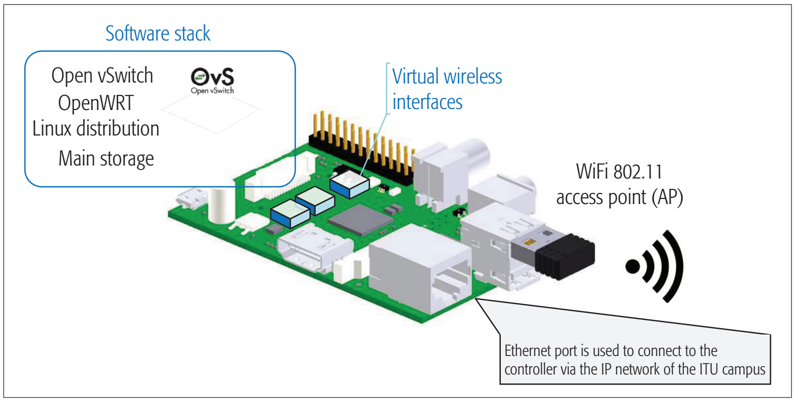

Software stack
Open vSwitch
OpenWRT
Linux distribution
Main storage
OvS Open vSwitch
Virtual wireless interfaces
WiFi 802.11 access point (AP)
Ethernet port is used to connect to the controller via the IP network of the ITU campus

Figure 4. Raspberry Pi — OpenFlow RSU.

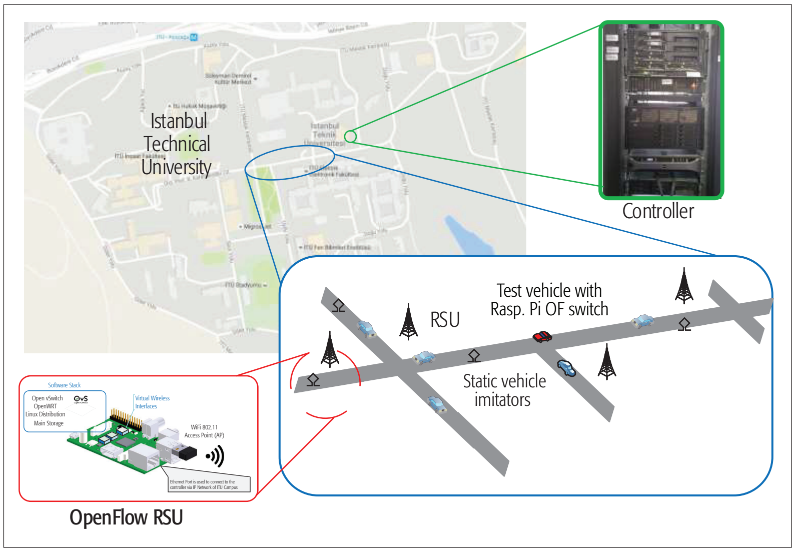

Istanbul Technical University
Controller
Test vehicle with Rasp. Pi OF switch
RSU
Static vehicle imitators
OpenFlow RSU

Figure 5. SD-VANET ITU campus testbed.

Implementation

SDN-WAM is implemented on the offered testbed by developing code for both the controller and the OpenFlow switch. A new network function, wireless host control, is developed and added to the OpenDaylight controller using northbound APIs provided by the controller. Wireless host control keeps track of the bandwidth used by the wireless hosts and implements a basic flow admission control depending on the available bandwidth assigned to the user. In addition, the module also relays the user connect/disconnect events to the wireless access management module for evaluation. The flow admission control algorithm depicted in Fig. 3 is triggered whenever a PACKET IN message is received by the controller. PACKET IN messages are generated by the OpenFlow switches when a packet received from the data plane has no matching service chaining rule in flow tables. The algorithm inserts a new service chaining rule using FLOW MOD messages defined in OpenFlow if the flow is admitted. Otherwise, a packet received through PACKET IN is dropped.

The wireless driver of the Raspberry-Pi-based access point has been altered to create a Linux Netdevice for each connected wireless client. In addition, a soft-switch daemon is triggered by connect/disconnect events of wireless clients to generate a PORT MOD message destined to the controller. This message intends to notify the controller about the status of the current network topology. The shutdown handler of the PORT MOD message generated by the controller is bound to the disconnect action for the interfaces virtualizing wireless clients. Furthermore, each packet received from the physical wireless interface is redirected to the relevant Linux Netdevice interface to satisfy the SDN soft switch forwarding structure that runs on Raspberry Pi.

Deployment

The ITU campus was selected as the test environment to deploy the implemented SDN-WAM testbed. The testbed consists of an OpenDaylight SDN controller and 10 Raspberry Pi OpenFlow switches. Furthermore, the controller was placed in the Computer Engineering Department's Communication Laboratory, and the OpenFlow switches were distributed throughout the campus to several locations. The locations of the controller and OpenFlow switches are depicted in Fig. 5. The green marker points to the location of the controller, while the red circle pinpoints locations of OpenFlow switches. The management ports of the OpenFlow switches were connected to the wired LAN of the university campus. The controller was also connected to the wired LAN of the university campus via Gigabit Ethernet connection. Furthermore, the control plane of the SD-VANET communicated through TCP connection, as seen in the control path of the OpenFlow switch in Fig. 2, over the wired LAN of the university campus. The control plane communication was isolated using the private virtual LAN (VLAN) allocated for the SD-VANET control plane communication within the ITU network.

One dedicated laptop computer with wireless connection was placed within the coverage of each OpenFlow switch. In addition, students of the Computer Engineering Department were notified about the existence of the OpenFlow wireless access points. Students were requested to join the real-time test environment and connect to the Internet using OpenFlow switches over their smartphones.

Wireless hosts connected to the OpenFlow switches were orchestrated by the controller. In addition, the number of users allowed within an OpenFlow switch was also organized by the implemented wireless host management module on the controller. A test that contains several users was run

One of the major future challenges of the proposed architecture is providing scalable software-defined controller for large VANETs which contain highly dynamic topologies. Network virtualization is a promising technique to handle the scalability problem, but it is critical to decide how to create or partition the network into overlay networks or slices based on various applications and service demands.

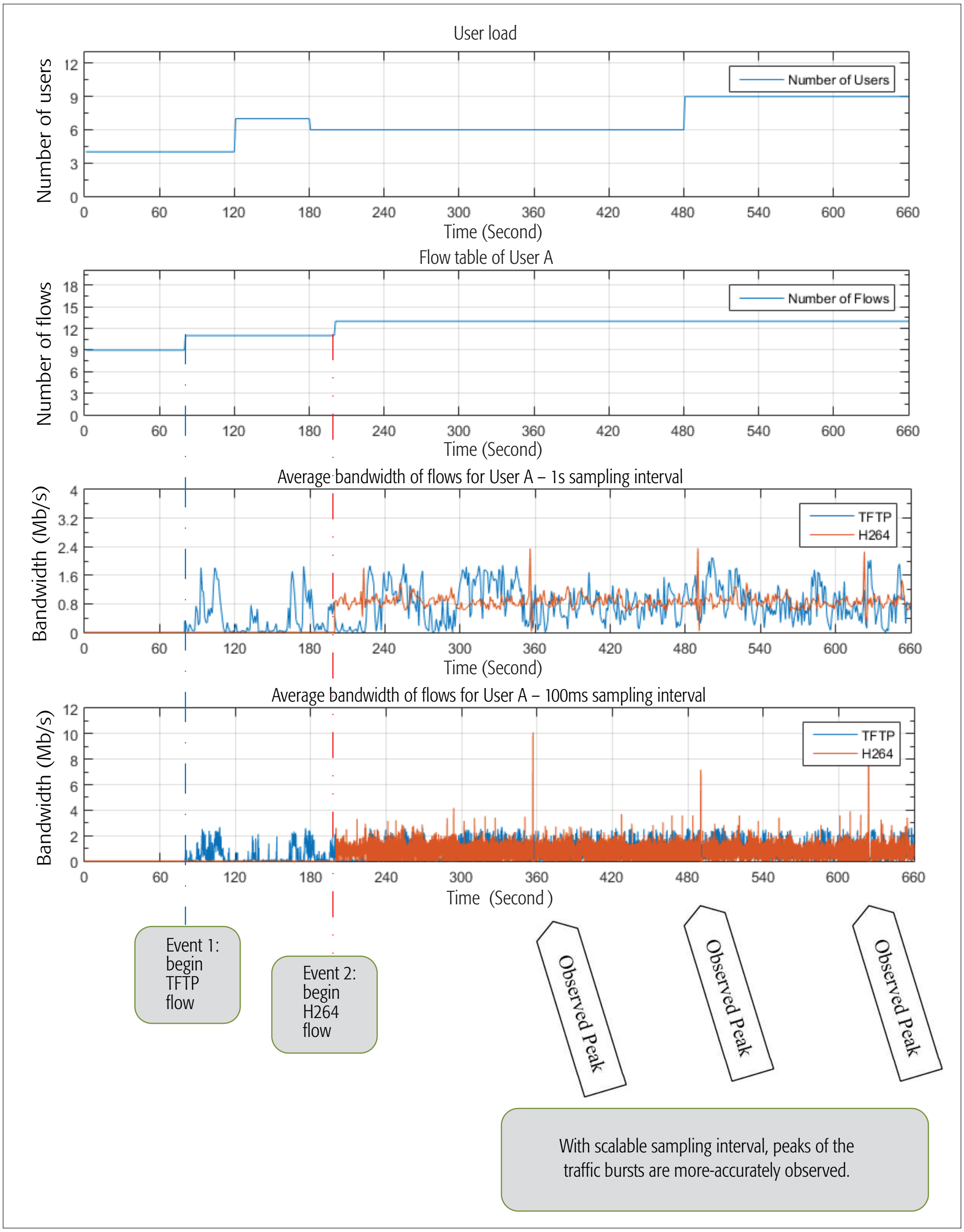


Figure 6. SDN-WAM monitoring results.

to validate the implemented architecture. The two distinct events started by **User A** were observed during the test with two different sampling interval such as 1 s and 100 ms. Additionally, the number of users within the observation period was also tracked by the controller and partially plotted in Fig. 6. Moreover, the basic flow admission control algorithm expressed in Fig. 3 optimized the available bandwidth of the user. The number of flows per user is also depicted in Fig. 6. The bandwidth profile of the H.264 stream and TFTP flows are also depicted in Fig. 6. By changing the sampling period of the bandwidth profiling, the controller also managed to capture the traffic bursts generated by the flows in small periods such as the one generated by H.264 in Fig. 6.

Conclusion

In this article, the SD-VANET architecture proposed for enhancing legacy Wi-Fi network to VANET has been explained in detail. Furthermore, a novel SD-VANET testbed has been explained and implemented. The SDN-WAM protocol, proposed for user management and flow admission control, was implemented on a testbed. Moreover, monitoring and management capabilities of the proposed architecture was shown using the deployed SDN-WAM implementation. Using the aforementioned capabilities, the proposed architecture is nominated as a promising augmentation candidate for fifth generation (5G) network design.

One of the major future challenges of the proposed architecture is providing a scalable software-defined controller for large VANETs that contain highly dynamic topologies. To this end, network virtualization is a promising technique to handle the scalability problem, but it is critical to decide how to create or partition the network into overlay networks or slices based on various applications and service demands. Furthermore, the future architecture should provide an interface to support critical use cases of 5G networks such as smart transportation, vehicles, and infrastructure. With this integration, orchestration of the whole network could be realized more efficiently, and even the network slices could be defined intelligently to address the major scalability problem.

In the near future, we will focus on defining the structure of the SD-VANETs and cellular core network interaction. In addition, we will also expand the offered architecture to address scalability issues that have not been mentioned within this work.

In near future, we will focus on defining the structure of the SD-VANETs and Cellular Core Network interaction. In addition, we will also expand the offered architecture to address scalability issues that have not been mentioned within this work.

Acknowledgment

This work was supported by the Newton Research Collaboration Program under Grant ID NRCP1516/1/131.

References

[1] A. Al-Fuqaha *et al.*, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surveys & Tutorials*, vol. 17, no. 4, 2015, pp. 2347–76.

[2] C. Bernardos *et al.*, "An Architecture for Software Defined Wireless Networking," *IEEE Wireless Commun.*, vol. 21, no. 3, June 2014, pp. 52–61.

[3] T. Chen *et al.*, "Software Defined Mobile Networks: Concept, Survey, and Research Directions," *IEEE Commun. Mag.*, vol. 53, no. 11, Nov. 2015, pp. 126–33.

[4] K. Zheng *et al.*, "Software-Defined Heterogeneous Vehicular Network: Architecture and Challenges," *IEEE Network*, vol. 30, no. 4, July 2016, pp. 72–80.

[5] Q. Zheng *et al.*, "Delay-Optimal Virtualized Radio Resource Scheduling in Software-Defined Vehicular Networks via Stochastic Learning," *IEEE Trans. Vehic. Tech.*, vol. 65, no. 10, Oct. 2016, pp. 7857–67.

[6] Z. Arslan *et al.*, "Sdoff: A Software-Defined Offloading Controller for Heterogeneous Networks," *2014 IEEE Wireless Commun. and Networking Conf.* Apr. 2014, pp. 2827–32.

[7] Z. He, J. Cao, and X. Liu, "SDVN: Enabling Rapid Network Innovation for Heterogeneous Vehicular Communication," *IEEE Network*, vol. 30, no. 4, July 2016, pp. 10–15.

[8] Q. Zhou *et al.*, "Network Virtualization and Resource Description in Software-Defined Wireless Networks," *IEEE Commun. Mag.*, vol. 53, no. 11, Nov. 2015, pp. 110–17.

[9] R. Sherwood *et al.*, "Flowvisor: A Network Virtualization Layer," ONF tech. rep., 2009.

[10] L. Liao, A. Shami, and V. Leung, "Distributed Flowvisor: A Distributed Flowvisor Platform for Quality of Service Aware Cloud Network Virtualisation," *IET Networks*, vol. 4, no. 5, 2015, pp. 270–77.

[11] P. Lin *et al.*, "A West-East Bridge Based SDN Inter-Domain Testbed," *IEEE Commun. Mag.*, vol. 53, no. 2, Feb. 2015, pp. 190–97.

[12] G. Cardone *et al.*, "The Participact Mobile Crowd Sensing Living Lab: The Testbed for Smart Cities," *IEEE Commun. Mag.*, vol. 52, no. 10, Oct. 2014, pp. 78–85.

[13] K. Chowdhury and T. Melodia, "Platforms and Testbeds for Experimental Evaluation of Cognitive Ad Hoc Networks," *IEEE Commun. Mag.*, vol. 48, no. 9, Sept. 2010, pp. 96–104.

[14] OpenDaylight — An Open Source Community and Meritocracy for Software Defined Networking, https://www.opendaylight.org/, accessed Dec. 15, 2015.

[15] Developer's Guide — Basic Approach to OpenWrt, https://wiki.openwrt.org/doc/guide-developer, accessed Dec. 15, 2015.

Biographies

Gökhan Seçinti [S'13] (secinti@itu.edu.tr) is a Ph.D. candidate in the Computer Engineering Programme, Istanbul Technical University (ITU). He serves as a reviewer for *IEEE Communications Magazine*, *IEEE Transactions on Vehicular Technology*, *IEEE Transactions on Wireless Communications*, the *International Journal of Communication Systems*, and the *International Journal of Computer and Telecommunications Networking*. He is a recipient of the IEEE INFOCOM Best Poster Paper Award (2015) and IEEE CAMAD Best Paper Award (2016). His current research includes software-defined networking and performance analysis of 5G networks.

Berk Canberk [S'03, M'11, SM'16] is an associate professor in the Department of Computer Engineering at ITU. Since 2016, he is also a visiting associate professor with the Department of Electrical and Computer Engineering at Northeastern University. He serves as an Editor for *IEEE Transactions in Vehicular Technology*, *Elsevier Computer Networks*, *Elsevier Computer Communications*, and the *Wiley International Journal of Communication Systems*. He is a recipient of the IEEE CAMAD Best Paper Award (2016), the Royal Academy of Engineering (UK) NEWTON Research Collaboration Award (2015), and the IEEE INFOCOM Best Poster Paper Award (2015). His current research areas include software-defined networking, 5G network performance analysis and modeling, and cognitive radio networks.

Trung Q. Duong [S'05, M'12, SM'13] (trung.q.duong@qub.ac.uk) is currently an assistant professor (lecturer) at Queen's University Belfast, United Kingdom. He has been awarded the Best Paper Award at IEEE VTC 2013, IEEE ICC 2014,and IEEE GLOBECOM 2016. His current research interests include 5G networks. He has published more than 250 technical papers (including 138 journals). He is serving as an Editor for *IEEE Transactions on Wireless Communications* and *IEEE Transactions on Communications*.

Lei Shu [M'07, SM'15] (lei.shu@ieee.org) is a professor at Guangdong University of Petrochemical Technology. He is also the executive director of the Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, China. His main research field is wireless sensor networks. He has published over 300 papers. He received GLOBECOM 2010 and ICC 2013 Best Paper awards. He serves as Editor-in-Chief of *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*.

NETWORK AND SERVICE MANAGEMENT

George Pavlou

Jürgen Schönwälder

This is the 23nd issue of the Series on Network and Service Management, which is typically published twice a year, in January and July. The Series provides articles on the latest developments, highlighting recent research achievements and providing insight into both theoretical and practical issues related to the evolution of the network and service management discipline from different perspectives. The Series also provides a forum for the publication of both academic and industrial research, addressing the state of the art, theory, and practice in network and service management.

The key annual event of the network and service management community, the International Symposium on Integrated Network Management (IM 2017) (http://im2017.ieee-im.org/), took place on May 8–12 in Lisbon, Portugal. During IM, the prestigious IEEE/IFIP Dan Stokesberry award, awarded every two years to an individual who has made particularly distinguished technical contributions to the growth of the network management field, was given to Nikos Anerousis of IBM Research, who also gave a keynote speech at the Symposium. In addition, the new IEEE CNOM committee was elected for the next four-year term and comprises the following officers: Filip de Turck (Chair), Ghent University, Belgium; Laurent Ciavaglia (Vice-Chair), Nokia Bell Labs, France; Carol Fung (TPC Chair), Virginia Commonwealth University, United States; and Carlos Raniery (Secretary), Federal University of Santa Maria, Brazil. Congratulations to the incoming officers and thanks to the outgoing ones for their service over the last term. The next major annual event of the network and service management community is the Conference on Network and Service Management (CNSM 2017) (http://www.cnsm-conf.org/2017/), which will take place in November in Tokyo, Japan.

We again experienced excellent interest in the 23rd issue with 23 submissions in total. For all submissions in the scope of our Series, we obtained at least three independent reviews. We finally selected four articles, resulting in an acceptance rate of 21.7 percent. The acceptance rate of all the previous issues has ranged between 14 and 25 percent, making this Series a highly competitive place to publish.

The first article, "REF: Enabling Rapid Experimentation of Contextual Network Management Using SDN" by Fawcett, Mu, Kareng, and Race, presents the design and operational guidelines for a software-defined networking experimentation framework that enables rapid evaluation of contextual networking designs using real network infrastructures.

The second article, "On the Resiliency of Virtual Network Functions" by Han, Gopalakrishnan, Kathirvel, and Shaikh, explains the resiliency requirements for VNFs in order to provide carrier grade services, summarizes the existing solutions in the literature, highlights relevant research challenges, and presents a concrete case study demonstrating how to enhance a router's resiliency.

The third article, "Wi-Fi Self-Organizing Networks: Challenges and Use Cases" by Gacanin and Ligata, surveys challenges and use cases in self-optimizing networks that have not been presented to date, setting the challenging requirement on next-generation Wi-Fi technology to provide seamless and uniform network quality of service.

Finally, the fourth article, "Coordination of SON Functions in Multi-Vendor Small Cell Networks" by Wielgoszewska, Ho, Gacanin, and Claussen, presents a design together with a sample implementation of a coordination scheme between three key self-organizing network functions in small cell networks: cell ID assignment, coverage adjustment, and idle mode control.

We hope that readers of this issue find the articles informative, and we will endeavor to continue with similar issues in the future. We would finally like to thank all the authors who submitted articles to this Series and the reviewers for their valuable feedback and comments on the articles.

BIOGRAPHIES

GEORGE PAVLOU (g.pavlou@ucl.ac.uk) is a professor of communication networks at the Department of Electronic Engineering, University College London, United Kingdom, where he coordinates networks and services research activities. His research interests focus on networking and network management, including traffic engineering, autonomic networking, information-centric networking, and software-defined networks. He has been instrumental in a number of research projects that produced significant results with real-world uptake and has contributed to standardization activities in ISO, ITU-T, and the Internet Engineering Task Force (IETF).

JÜRGEN SCHÖNWÄLDER (j.schoenwaelder@jacobs-universiy.de) is a professor of computer science at Jacobs University Bremen, Germany. His research interests include network management and measurement, network security, embedded systems, and distributed data processing. He is an active member of the IETF, where he has edited more than 30 network management related specifications and standards. He has contributed in various roles to the organization of IEEE and IFIP sponsored academic conferences and journals.

REF: Enabling Rapid Experimentation of Contextual Network Traffic Management Using Software Defined Networking

Lyndon Fawcett, Mu Mu, Bruno Hareng, and Nicholas Race

The author contributes the design and operational guidelines for an SDN experimentation framework (REF), which enables rapid evaluation of contextual networking designs using real network infrastructures. Two use case studies of a QoE-aware resource allocation model and a network-aware dynamic ACL demonstrate the effectiveness of REF in facilitating the design and validation of SDN-assisted networking.

Abstract

Online video streaming is becoming a key consumer of future networks, generating high-throughput and highly dynamic traffic from large numbers of heterogeneous user devices. This places significant pressure on the underlying networks and can lead to deterioration in performance, efficiency, and fairness. To address this issue, future networks must incorporate contextual network designs that recognize application and user-level requirements. However, designs of new network traffic management components such as resource provisioning models are often tested within simulation environments, which lack subtleties in how network equipment behaves in practice. This article contributes the design and operational guidelines for an SDN rapid experimentation framework (REF), which enables rapid evaluation of contextual networking designs using real network infrastructures. Two use case studies of a QoE-aware resource allocation model and a network-aware dynamic ACL demonstrate the effectiveness of REF in facilitating the design and validation of SDN-assisted networking.

Introduction

With the growing popularity of video services and the increasing online presence of traditional broadcasters, online video is believed to be the leading consumer of future networks, generating high-throughput and highly dynamic network traffic [1]. Adaptive media such as HTTP adaptive streaming (HAS) using protocols like TCP or Quick UDP Internet Connections (QUIC) is becoming the de facto standard for online media streaming. The non-cooperative and unsupervised resource competition between adaptive media applications leads to significant detrimental quality fluctuations and an unbalanced share of network resources [2]. Therefore, it is essential for content networks to better understand the application and user-level requirements of different data flows and to manage the traffic intelligently. Traditional network traffic management approaches based on the configuration of proprietary devices are cumbersome and inefficient in the dynamic management of network resources [3]. Software defined networking (SDN) is a network paradigm that decouples network control from the underlying packet forwarding. It continues to gain traction as a vehicle for delivering efficient and flexible context-aware network programming. OpenFlow, first introduced by McKeown *et al.* [4], is commonly used to realize the concepts of SDN, with many networking devices now supporting the protocol. For every rule match specified, OpenFlow automatically maintains and updates packet counters, which may be interrogated on demand by an OpenFlow application. Furthermore, with the introduction of fog computing and network functions virtualization (NFV), the cloud is being brought closer to the user in the form of micro data centers or cloudlets [5]. This opens compute locations that are close to the edge, such as customer premises equipment (CPE), to enable contextual network traffic management services that process and can enforce at the network edge [6]. Context-aware networks are different from traditional networks as they are aware of the flows that have passed through the network, and can make decisions to alter the network based on this information.

Distinctions with Network Emulators and SDN Facilities

Recently there has been pioneering work on exploiting SDN for traffic engineering and network management. Nam *et al.* [7] propose an SDN application to monitor streaming flows in real time, dynamically changing the routing paths for better user experiences. Akella *et al.* [8] harness SDN to provide quality of service (QoS) bandwidth guarantees for priority users through a mathematical model. Mehdi *et al.* [9] argue for using SDN as a security mechanism for the home through anomaly detection and remediation. Wong *et al.* [10] propose to solve peak-hour broadband network congestion problems by pushing congestion management to the network edge using a two-level resource allocation design. However, SDN-assisted novel network programming models are often designed and tested in a simulation or emulation environment such as Mininet [11]. While these test environments do offer a means of experimentation, they do not consider the effects that network protocols, client programs, hardware limitations, physical switches,

Digital Object Identifier:
10.1109/MCOM.2017.1600507

Lyndon Fawcett and Nicholas Race are with Lancaster University; Mu Mu is with the University of Northampton; Bruno Hareng is with Hewlett Packard Enterprise.

0163-6804/17/$25.00 © 2017 IEEE

and other real-world factors may have on the outcomes. Major design flaws may be masked during simulation or emulation and are only discovered in prototyping or early production phases. Emulations can also be limited by the capabilities of software switches such as Open vSwitch.

Many researchers and projects have recognized the following benefits of an experimental testbed that provides an environment close to that of production networks:

1. Proving that SDN applications will operate with real-world hardware, or testing the behavior of specific hardware in each experimental context
2. Experimenting with specific operating system stacks and their network implementations
3. Supporting experiments where hardware constraints (CPU, memory, etc.) are part of the variables under evaluation

Facilities such as Fed4FIRE [12] and their tools provide a means for many researchers to run SDN experiments over geographically distributed hardware, which would otherwise not be possible. However, when slicing the networking resources between multiple users, the outcomes can change on each experimental run due to the load generated by simultaneous experiments, ultimately skewing results. Further impacting this, each experimenter is unaware of the other ongoing experiments, meaning that it is difficult to determine if the results attained were as expected or due to another user on the facility conducting a load intensive experiment.

The contribution in this article differs from existing facilities and software in various ways. One area where the rapid experimentation framework (REF) excels is in its flexible and portable deployment method; a network tested on the experiment facility can also be tested within Mininet, or even executed in production with little changes. As well as this, in contrast to existing facilities that typically provide very detailed low-level control to just the network infrastructure, REF provides higher-level abstractions of both the network and virtualization infrastructures through orchestration, automating the creation, connection, running, and cleaning of nodes in an experiment. Furthermore, it also provides abstraction over the network for making the creation of context-aware traffic management applications as streamlined as possible. Additionally, with the unique configuration using slicing and port multiplexing, REF can create much larger physical networks with limited hardware than its competitors. Finally, the entire REF framework can be used and modified by anyone without any kind of registration or subscription to a federation.

In this article, REF is introduced, an experimentation framework and a guide to building a testbed that together provides a blueprint for an SDN-based contextual network design facility. First, it describes the framework, covering the requirements of the framework, then the purpose of each component within the system as well as the abstractions that it provides to the user. Next, the experiment testbed is detailed, providing a guide on how to construct your own virtualization and network infrastructure for experimentation. After this, both use cases are described and used to show REF in operation, which includes a quality of experience (QoE)-aware resource allocation model and a network-aware dynamic ACL. Finally, the article goes into a discussion on interesting findings that arose during the creation and use of the system.

REF Experimentation Framework

Setting up a functional SDN testbed is a challenging process requiring extensive knowledge and experience. We aim to create a framework that assists researchers in creating their own SDN applications and experiments, while providing isolation to avoid conflict between experiments. Furthermore, the framework should make the most of the hardware available so that researchers can create topologies on a similar scale to those available in simulation environments. Additionally, the framework should allow replicating large-scale localized experiments, and this is useful when modeling a data center, home, or business topology where there is a dense collection of nodes with low latency between each other. This feature is generally not available using a shared testbed due to the equipment being geographically distributed.

To provide a harness capable of supporting rapid deployment and orchestration of experiments, an experimentation platform will need to fulfil the following requirements:

- *Experiments close to practice and at scale*. The system should be able to realize and manage a large number of clients and networks. Meanwhile, to provide both realism and scale, the environment will encompass both physical and virtual elements.
- *Dynamic manipulation of the network*. Rate limiting, queuing, flow redirection, and other features of SDN implementation are required to enforce decisions made by intelligent network traffic management modules.
- *Configurable clients*. The client's configuration (image and resources) should be quickly changeable (automated based on test manifests) after an experiment to set up for a new experiment as well as at runtime.
- *Rapid repeatability of experiments in a clean environment*. Ensure that no residual effects are left over from previous experiments by removing virtual machines (VMs) and networks before a new experiment.

Functional Components

The REF framework (Fig. 1) orchestrates the virtual and physical network infrastructure to assist the execution and statistical data gathering of network-based experiments. It consists of a three-layer architecture: the top layer contains components provided by the researcher including the test manifest and application/user-level functions such as our case studies: QoE and security applications. The middle layer contains the REF orchestrator, which interfaces with, and includes, the infrastructure managers. The bottom layer contains the network and virtualization infrastructure where the experiments are deployed.

The test manifest describes the experiment in a JSON format. It includes each client's IP address, the networks to which each is attached, the VM image to be used, and network emulation requirements. The example manifest below shows

Setting up a functional SDN testbed is a challenging process requiring extensive knowledge and experience. We aim at creating a framework that assists the researcher in creating their own SDN applications and experiments, while providing isolation to avoid conflict between experiments.

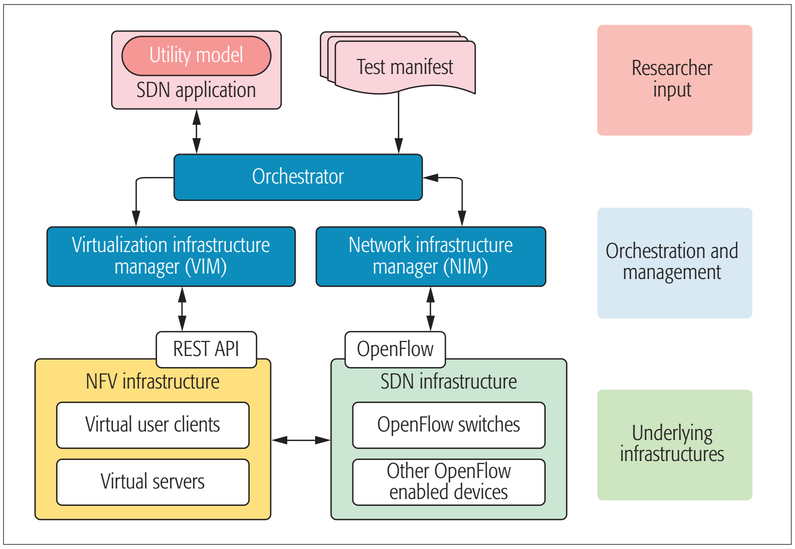


Figure 1. Rapid experimentation framework.

two networks *lan1* and *lan2* who share the same aggregation network (*group1*), and the currently available bandwidth on the aggregation network is configured to be less than the sum of bandwidth on *lan1* and *lan2*.

```
Spec = {
        'name' : "test experiment"
        'keypair' : "openstack_rsa"
        'controller' : "10.30.65.210"
        'credentials' : {'user' : "Test", 'password' : "Test",
        'project' : "Test"},
        'networks' : [{'name' : "lan1", "subnet" :
        "192.168.1.0/24", "rate" : 5000, "group" : 1},
                {'name' : "lan2", "subnet" : "192.168.2.0/24",
                "rate" : 5000, "group" : 1}]
        'groups' : [{'id': 1, rate: "8000"}]
        'hosts' : [{'name' : "h1", 'image' : "Scoot",
        'flavour': "small", 'net' : [{lan1}]},
                {'name' : "h2", 'image' : "Scoot", 'flavour':
                "small", 'net' : [{lan2}]}]
    }
```

The SDN application contains a utility model that captures application-level requirements such as QoE and security measures. As part of the framework, the SDN application is an interchangeable component that communicates with the REF orchestrator through a remote procedure call (RPC) interface providing information about resource allocation on flows. Additionally, information is sent back in regard to the current throughput at different points in the network using SDN-specific control messages such as OpenFlow's *meter statistics* and *flow statistics* messages.

The *REF Orchestrator* handles communication between all the components. It includes two subcomponents, the virtual infrastructure manager (VIM) and network infrastructure manager (NIM). The VIM controls the virtualization infrastructure through a RESTful application programming interface (API), and it launches and configures experiment nodes with information from the test manifest. At the end of the experiment, it resets the test environment by triggering VIM and NIM clean methods, removing networks and virtual machines it instantiated so that the environment is ready for the next experiment.

The NIM controls the network infrastructure and consists of a Ryu OpenFlow controller containing a metering and monitoring application. It installs meter flow mods on request from the SDN application and provides information from the network including current throughput of flows and switches. These abstractions over the network infrastructure are interfaced directly with the orchestration component, which in turn provides a simple RPC API to the researcher's SDN application. This allows the orchestrator to define and configure network setup on the fly through a simple JSON formatted request. A typical request would be to report the current network traffic level for a port or previously defined flow. An example command would be to define a flow (e.g., source/destination IP pair), and request that the flow be limited to a certain level (defined in megabits per second). The response to this command includes a unique identifier that can be used in subsequent requests for traffic data. The VIM is positioned above the virtualization infrastructure (managed by OpenStack), and provides an interface to the orchestrator to provide an instantiation of experiment nodes that are connected to the experiment network. The network infrastructure creates connections between nodes and switches, and provides a platform for configuring link bandwidth.

REF Abstractions

The design for the REF architecture was an iterative process based on initial requirements for context-aware SDN network applications. These desired requirements included: port and flow monitoring, total bandwidth capacity estimation, and controlling bandwidth on a per flow and port basis. We then added functionality to support other state-of-the-art applications created by other researchers using the framework; this included a collection of metering statistics, enabling the ability to define the flow granularity instead of using the same as the forwarding application, and the ability to provide and choose from a catalog of existing forwarding applications. The design of REF stemmed from our experience working with other testbeds and frameworks including Fed4FIRE.

The following lists the main abstractions provided by REF that SDN applications can use. These features are available through a JSON-RPC interface between the researchers' SDN application and REF's orchestrator.

Virtualization and node management abstractions:

- Creating and destroying VMs after each experiment and during when required
- Executing scripts on each client for the experiment
- Recording and aggregating experiment logs from nodes
- Configuring link bandwidth between virtual nodes

Network traffic management abstractions:

- The monitoring of flows at multiple levels while simultaneously logging these for post-experiment analysis
- The monitoring and logging of throughput observed on switch ports

- Providing network forwarding by default, thus reducing the time and difficulty of researchers when creating their utility application
- Enforcing throughput constraints on flows, groups of flows, ports, and groups of ports
- Monitoring and logging of OpenFlow meter counters
- Configuring link bandwidth between physical nodes

The feature list of REF is continuously evolving as SDN matures; for an extensive and current list of the capabilities of this framework, consult the public project webpage for REF (http://lyndon160.github.io/REF/).

Building an Experimentation Testbed

To demonstrate the effectiveness of REF, we provide an implementation guideline (Fig. 2) and two experimental case studies based around the delivery of video to multiple home environments and another based around a smart grid network. For both cases, these connections share a restricted link to the Internet. In addition, the CPEs and the local DSLAM are also under SDN control to provide programmable link configuration for dynamic management of traffic. This reflects our vision of an SDN-assisted pervasive computing and networking environment, allowing granular network control at the very edge of the network.

Virtualization Infrastructure

At the core of the virtualization infrastructure is an OpenStack installation. This provides the means of building and connecting VMs to instantiate a significant amount of dynamically configurable live client applications. The OpenStack installation is standard, with one main modification: virtual LAN (VLAN) trunks are used to break out network interfaces from VMs. These are then mapped one-to-one to exclusive physical interfaces on a switch. We refer to this process as *port-multiplexing*, as it allows an Ethernet switch to implement remote physical interfaces for virtualized machines. This is an essential feature for our experimentation as it allows each client to be directly assigned to a physical port on an SDN controlled switch. The mechanism for this is based on the use of VLANs to carry VM traffic onto the switch. The configuration is such that each VM is allocated an exclusive OpenStack (Neutron) network.

The setup and management of this infrastructure are controlled by VIM. As such, we use an in-house orchestration tool titled MiniStack (https://github.com/hdb3/ministack). Its purpose is to bring the network and client creation automation capabilities shown in Mininet. MiniStack provides the ability to rapidly build, reconfigure, and delete (all within seconds) experimental topologies using a simple and extensible configuration format that includes networks, connections, and clients. Furthermore, as this is a modular component, it can be used by other projects to automate creation and deletion of network topologies.

Network Infrastructure

The network infrastructure used in this example consists of two OpenFlow v1.3 capable switches (switches 2 and 3 shown in Fig. 2) with metering support. In our facility, we use Hewlett Packard Enterprise Aruba's 3800 series (HPE3800) switches, as they fulfill both requirements. However, other compliant switches could be used instead, including OpenFlow switches from Pica and Corsa. The HPE3800 also hosts other important capabilities, such as the ability to flexibly partition a single physical switch into several virtual OpenFlow switches. Each of these is a complete and distinct OpenFlow instance. This is outside of the scope of OpenFlow, but is a feature present on many devices available on the market. This partitioning feature is vital in achieving the scale required in experimentation without incurring the associated cost. However, it is important to note that the switches' memory is shared between instances, reducing the maximum number of flow entries per application. For flow table efficiency, the roles of switches 2 and 3 can be merged by using a switch with multi-table support.

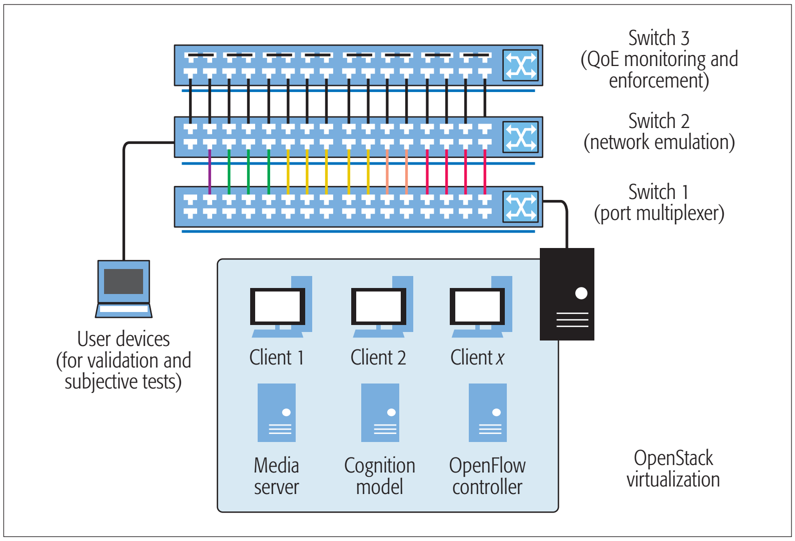


Figure 2. Virtualization and network infrastructure.

Use Case Study 1: QoE-Aware Resource Allocation

We use the evaluation of UFair [13], a QoE model, as a use case study of how REF supports rapid research and experimentation. UFair seeks to reduce the frequency of adaptations over a group of HAS clients, and moderate individual clients' choice of stream bandwidth, to the benefit of all applications on the same network. The core of UFair is a mathematical specification for the optimal bandwidth to be consumed for each member of a group of clients, based on the prevailing network resources and user device capabilities. It is stateful, to retain data about past forced bandwidth changes and thus reduce the impact of resolution changes across the entire client group. UFair operates by using REF's monitoring and enforcing capabilities to get information about the network status and "capping" resources on individual media streams, with the assumption that media clients can adapt their bandwidth utilization in response to network constraints. Therefore, resource allocation or other traffic control can be achieved transparently in the network without cooperation from user applications. The effectiveness of such network-based control is dependent on how application and user-level context is incorporated in network traffic management design and executed by SDN.

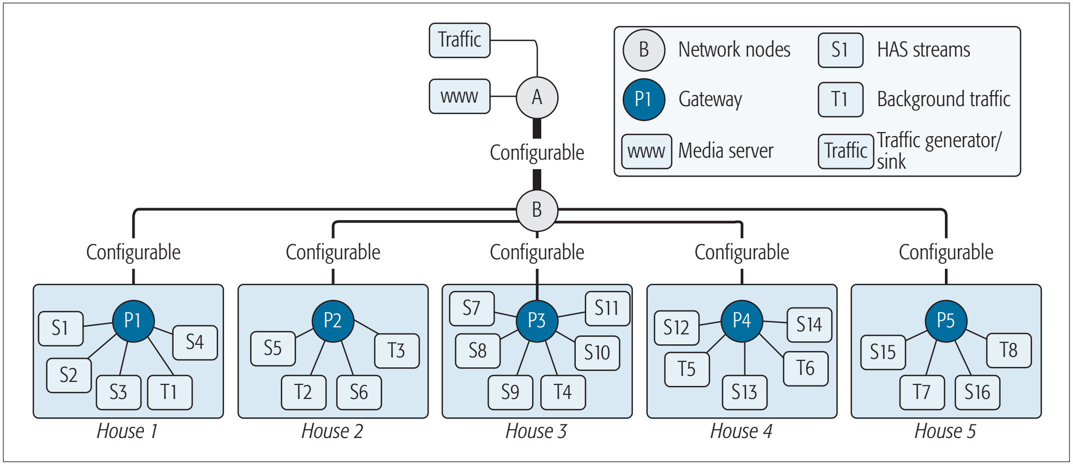


Figure 3 Experimentation topology.

Experiment Topology and Operation: Figure 3 depicts a tiered topology representing a multi-household network. Each of the households contains 4–6 hosts, all of which are connected to a gateway. This gateway is then connected, along with other gateways in the topology, to an aggregation switch (switch B). Over another hop (toward switch A), a foreground and a background server act as endpoints as sources or sinks of respective traffic types.

To emulate a potential home network environment where a household has limited bandwidth, and the link shared between houses is also limited, network link characteristics are dynamically configured. Through REF, the links between switch B and the gateway switches are limited to 20 Mb/s. Similarly, the link between B and A is restricted to 50 Mb/s. The sum of the connectivity available to household links is 100 Mb/s, and is greater than the link between B and A. This results in a situation whereby there is more demand than there is supply in the case of multiple households. In these circumstances, the adaptive streams in each house are affected by hosts within the same house, as well as the behavior of hosts in other houses. Using REF, network configuration is directly programmable as an integral part of the framework to capture the complex and temporal dynamic characteristics of real-world complex networks.

In this setup, the hosts in households are configured as online video players to request MPEG-DASH adaptive video content from media servers, with one or two hosts in the same household generating background traffic. The experiment used REF to monitor the network statistics of each client, as well as each household. This data is then analyzed by the UFair model to determine the most optimal resource allocation strategy. Recommendations given by the UFair model are then applied through REF's traffic enforcement functions, including restrictions per flow and household. We also define a *baseline* experiment, where network statistics are still monitored but no additional traffic management is applied.

Results: Figure 4 depicts the resultant video quality of each video stream when the OpenFlow-assisted UFair model is inactive and when it is active. Video quality is a rate-distortion function that is used to describe the nonlinear relationship between quality and bit rate [13]. The results clearly demonstrate the significant differences of the network provisioning strategy adopted by the user-level model compared to the conventional TCP-based network-level baseline model. The baseline model allows video streams with more intensive requests at the transport layer to acquire more resources, leading to some video streams being heavily penalized. Using the bespoke UFair model, the network management element in the testbed can schedule the resource according to the QoE requirements and link status of every HAS stream. Thus, network resources are dynamically provisioned in such a way that similar video quality is maintained on all related media streams for the entire course of the experiment (Fig. 4). Furthermore, the UFair model was able to avoid any severe video quality fluctuation due to its awareness of all competing media flows in the same network. In this case, we can validate the performance of a utility model by repeating the test 100 times without human intervention. The functions offered by REF, including streamlining the orchestration of

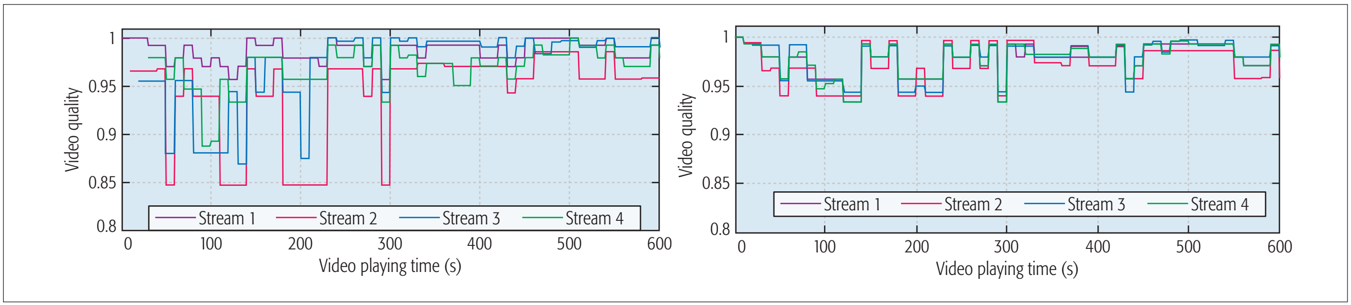


Figure 4. Resource allocation without (left) and with (right) the OpenFlow-assisted UFair model.

utility model, virtualization infrastructure, and physical OpenFlow equipment, allows researchers to focus on application-level design.

Use Case Study 2: Context-Aware Access Control

The Smart-ACL use case study considers how REF supports experimentation with security-based context-aware SDN applications. Smart-ACL is designed to provide protection in the network on top of SDN switches, and reflects an increasing research interest in the adoption of SDN within critical infrastructures. This specific use case considers the use of SDN within a smart electricity substation environment, where protection mechanisms are required against attacks such as denial of service (DoS). This level of protection is necessary to prevent an unwanted event, such as a mass power outage [14]. Moreover, standards bodies such as the International Electrotechnical Commission (IEC) have strict security and resiliency requirements in place to prevent this. Before SDN can be safely adopted in these networks, the above issues need to be addressed, while adhering to IEC standards.

Smart-ACL harnesses multiple OpenFlow features exposed by REF to prevent a multitude of attacks. It operates by using whitelists, rate-limiting on the packet in flow rule, and making remediation decisions based on network context from the REF. Remediation is applied through REF's rate-limiting and blacklists. It takes information from the network about the whitelisted nodes' traffic and classifies this as essential traffic. An average of this traffic is then taken into consideration when rate-limiting non-essential traffic. This value is recalculated periodically with various tolerances to ensure that slow attacks are detected. In this case study, we show how REF has been used to assist in the development of Smart-ACL.

In operation, REF is started and manages the network's connectivity. The Smart-ACL application calls the orchestrator's northbound interface to get information about flows in the network, including flow headers and counters. Using this information, it protects whitelisted services by ensuring that there is enough bandwidth available on the network so that they remain uninterrupted. To do this, Smart-ACL takes information from REF about how much of the total available bandwidth is being used by non-essential traffic (flows not in the whitelist) and rate-limits using REF's enforce feature if the traffic exceeds the total bandwidth minus the whitelists' required bandwidth. Furthermore, using meter drop counts from REF, the application detects if a flow is not behaving within the network constraints; if the drop rate exceeds the threshold, the traffic is blocked for a short while to allow other non-essential traffic fair use of the available bandwidth.

Experiment Topology and Operation: Comparable to the previous case study, hosts, attackers, and link limits are applied automatically through a configuration. In this case study, REF was used to automate three essential nodes with the purpose of maintaining a connection with the traffic sink generating HTTP traffic at a total target rate of 60 Mb/s. Alongside these, two non-essential nodes were connected to the traffic sink sending benign traffic. Additionally, there was a single attack node that was generating UDP traffic with no client-side throughput limits across the network toward the traffic sink. Also, a link limit of 150 Mb/s was set between two of the switches, emulating a constrained environment. Using REF's automation with a test-manifest, this experiment ran for 120 s and was repeated 100 times without any additional human intervention.

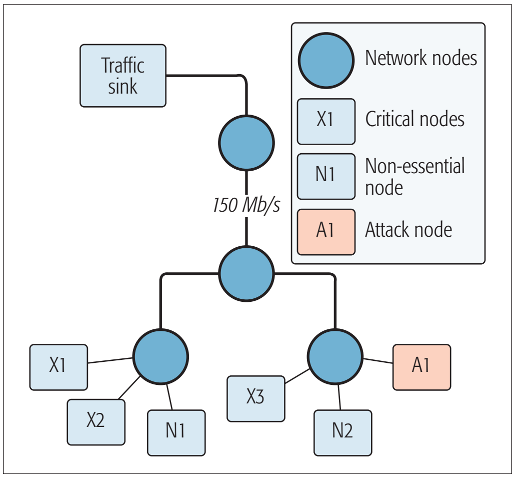


Figure 5. Experimentation topology.

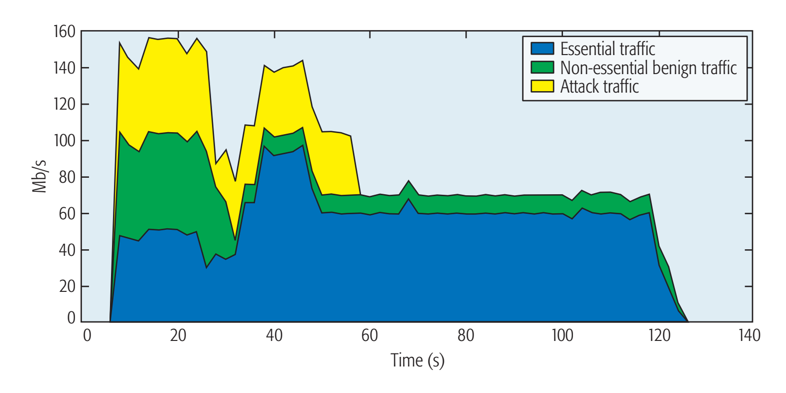


Figure 6. Results with Smart-ACL enabled.

REF assisted the researchers in developing this application by providing an underlying framework to manipulate the network, which also already provided forwarding logic. The use of being able to quickly and automatically repeat the experiment while having an output of the traffic in the network assisted when determining thresholds and timeouts for bandwidth rate-limiting flows, allowing the researchers to improve on the application model with ease to ensure that essential traffic remained unaffected by the ongoing attack.

Results: Figure 6 depicts a stacked graph of traffic logs produced by REF of the switch located at the top of the topology. The results show the effects that Smart-ACL has on the network when a simple UDP Flood DoS attack is triggered. These tests were performed 100 times; then each traffic type was averaged. We can see that the essential traffic remains stable, and that attack traffic along with the non-essential traffic was rate-limited after 25 s. The attack traffic was then identified through excessive packet drops on the meter counter and eventually ceased. Without Smart-ACL, the attack would have continued, limiting the bandwidth available to essential traffic. Further information

We will be continually monitoring progress on the state of the art of software switches to one day integrate them with REF for a hybrid infrastructure of virtual and physical switches; this will provide a means to create experiments on even greater scales without losing experiment rigor.

about this case study as well as the code and results are available on GitHub (https://github.com/lyndon160/Smart-ACL).

Discussions

When acquiring OpenFlow-enabled equipment for research and experimentation, it is essential to investigate the advanced features offered by different vendors and on different generations of equipment. The supported OpenFlow version (e.g., 1.0, 1.3, or 1.5) is often a first indication as to the OpenFlow features a device may offer. However, it is unlikely that all optional features of an OpenFlow specification will be fully implemented. Furthermore, implementation details of features such as metering are often left open to interpretation for the switch vendors, which can result in experiments behaving differently between two switches with the same advertised capability due to differences in implementations. It is worth investigating differences of devices' capabilities and implementation details, as they may have a significant impact on how they support design and evaluation. Thus, we recommend consulting the ONF's OpenFlow conformance list [15] when acquiring a new network switch for an experiment.

Conclusion and Future Work

The proliferation of online media is placing tremendous pressure on QoE and security requirements on existing network infrastructure. This has led to a growing body of research developing novel network traffic management models using software defined networking. Many researchers use simulation tools to evaluate their designs, which can overlook effects that are seen in link delay and link bandwidth emulation in networks and clients. This article introduces REF, a framework that facilitates rapid experimentation of SDN-assisted network designs using a combination of physical equipment and virtualized functions. We carried out two case studies on SDN-assisted QoE and security traffic management applications to validate the REF designs. We also provide detailed guidance and an open source toolset for readers to instantiate a research and experimentation environment of their own. By sharing our experiences, we hope to stimulate cross-site interconnected testbeds to support a research and innovation Internet environment, enabling new uses of the testbeds and thus research.

Leading on from this research, we intend to continue advancing REF as OpenFlow and its features mature. For vendor-specific features such as packet dropping policies, we are currently in the process of creating drivers for different switches to provide researchers with the ability to easily unlock more of these potentially useful features. Furthermore, we plan to open the network and virtualization infrastructures more widely as part of a new project, which starts in early 2017. We expect this facility to federate with other testbeds as part of the development activity within the project. Additionally, we are exploring the idea of creating APIs for other controllers so that a REF application would be portable between controllers. Finally, we will be continually monitoring progress on the state of the art of software switches to one day integrate them with REF for a hybrid infrastructure of virtual and physical switches; this will provide a means to create experiments at even greater scales without losing experiment rigor. Currently, the REF framework is shared between multiple U.K. universities in partnership through the U.K. EPSRC-funded TOUCAN and SDCN project.

References

[1] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2014–2019 White Paper."

[2] M. Mu *et al.*, "User-Level Fairness Delivered: Network Resource Allocation for Adaptive Video Streaming" *2015 IEEE/ACM Int'l. Symp. Quality of Service*, 2015

[3] W. Xia *et al.*, "A Survey on Software-Defined Networking," *IEEE Commun. Surveys & Tutorials*, vol. 17, no. 1, 2015, pp. 27–51.

[4] N. McKeown *et al.*, "OpenFlow," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 38, no. 2, 2008, p. 69.

[5] F. Bonomi *et al.*, "Fog Computing and Its Role in the Internet of Things," *Proc. 1st MCC Wksp. Mobile Cloud Computing*, 2012.

[6] F. M. F. Wong *et al.*, "Improving User QoE for Residential Broadband: Adaptive Traffic Management at the Network Edge," *2015 IEEE 23rd Int'l. Symp. Quality of Service*, 2015.

[7] H. Nam *et al.*, "Towards QoE-aware Video Streaming Using SDN," *2014 IEEE GLOBECOM*, 2014.

[8] A. V. Akella and X. Kaiqi, "Quality of Service (QoS)-Guaranteed Network Resource Allocation via Software Defined Networking (SDN)," *IEEE 12th Int'l. Conf. Dependable, Autonomic and Secure Computing*, 2014.

[9] S. A. Mehdi, K. Junaid, and S. A. Khayam, "Revisiting Traffic Anomaly Detection Using Software Defined Networking," *Lecture Notes in Computer Science*, 2011, pp. 161–80.

[10] F. M. F. Wong *et al.*, "Improving User QoE for Residential Broadband: Adaptive Traffic Management at the Network Edge," *2015 IEEE 23rd Int'l. Symp. Quality of Service*, 2015.

[11] B. Lantz, B. Heller, and N. McKeown, "A Network in A Laptop," *Proc. 9th ACM SIGCOMM Wksp. Hot Topics in Networks*, 2010.

[12] "Fed4FIRE Project," http://www.fed4fire.eu.

[13] M. Mu *et al.*, "A Scalable User Fairness Model for Adaptive Video Streaming over Future Networks," *IEEE JSAC*, 2016.

[14] A. Johnsson and N. Sigfridsson, "Deployment of Smart Substation Standard IEC 61850," *22nd Int'l. Conf. and Exhibition on Electricity Distribution*, 2013.

[15] "ONF OpenFlow Conformant: Certified Product List — Open Networking Foundation," https://www.opennetworking.org/openflow-conformance-certified-products, accessed 14 Dec. 2016.

Biographies

Lyndon Fawcett [S] is a Ph.D. student within the School of Computing and Communications at Lancaster University, United Kingdom, where he is involved in the EPSRC funded TOUCAN project. His primary research interests are in using fog computing infrastructures to enhance NFV and SDN. This research encompasses multiple disciplines, and entails exploring and creating innovative platforms for network and infrastructure orchestration that maintain an awareness of the underlying network and compute resources available.

Mu Mu [M] is a senior lecturer at the University of Northampton, United Kingdom. His Ph.D. in computer science and Master of Science degree were awarded by Lancaster University and TU-Darmstadt, Germany respectively. His research interests include software-defined cognitive networking, quality of experience, human factors in multimedia systems, and immersive media. He has over 50 publications in prestigious conferences and journals. He is a member of ACM.

Bruno Hareng is an SDN and security solution manager at HP Networking, Hewlett-Packard Enterprise Aruba EMEA. His graduated from the ENSIMAG Engineering School of Grenoble with an M.Sc. degree in computer science and applied mathematics. He has more than 20 years of experience in the industry. He has led many projects on communication networks, software-defined networking, QoS/QoE, and security solutions.

Nicholas Race is a reader in networking within the School of Computing & Communications at Lancaster University. His research is broadly around experimental networking and networked media, specializing in the use of SDN and NFV for new network-level services, including in-network media caching, network-level fairness, and network monitoring. His recent work considers the design of NFV service orchestration within fog computing environments.

11
10
9
8
7
6
4

On the Resiliency of Virtual Network Functions

Bo Han, Vijay Gopalakrishnan, Gnanavelkandan Kathirvel, and Aman Shaikh

The authors explain the resiliency requirements for virtual network functions in order to provide carrier grade services, summarize the existing solutions in the literature, highlight several research challenges, and present a concrete case study to demonstrate how to decompose a type of virtual router and thus enhance its resiliency.

Abstract

Network functions virtualization is an emerging technology that can significantly improve the flexibility of network service provisioning and offer potential cost savings. However, it is critical that service providers offer high reliability and availability of the network and services when moving from proprietary hardware appliances to virtualized network functions on commodity servers. In a network consisting of physical appliances, providers can deploy redundant hardware and extra capacity to handle failures, although this is quite expensive in practice. Virtualization of network functions lead to more challenges for resiliency, but also bring new opportunities to address these challenges in a more cost-effective manner. In this article, we explain the resiliency requirements for virtual network functions in order to provide carrier grade services, summarize the existing solutions in the literature, highlight several research challenges, and present a concrete case study to demonstrate how to decompose a type of virtual router and thus enhance its resiliency.

Introduction

Many network and telecommunication service providers have started migrating their infrastructure to take advantage of network functions virtualization (NFV) [1]. The idea behind NFV is to replace dedicated network appliances, such as routers and firewalls, with software that provides that same capability on top of commodity servers. Each individual network function, such as a software-based router, runs in a virtual machine and is called a virtual network function (VNF). Figure 1 illustrates the architectural framework of NFV with three major components: VNFs, NFV infrastructure (NFVI), and an accompanying management and orchestration (MANO) framework (e.g., ECOMP [2] from AT&T and the open source ONAP[1] project). The shown VNFs are virtual network address translation (NAT), firewall (FW), router, deep packet inspection (DPI), and VPN Internet gateway (VIG).

While NFV promises significant flexibility and control to network operators, it also brings more concerns for resiliency. Resiliency has been defined as *"the ability of the network to provide and maintain an acceptable level of service in the face of failures and challenges to normal operation"* [3]. Traditional carrier-grade systems have been engineered to offer higher than 99.999 percent (five 9s) availability (translates to roughly 25.9 s downtime per month). Achieving such availability is extremely difficult for VNFs due to multiple reasons: first, the commodity servers that host VNFs are more prone to errors and failures compared to the dedicated hardware appliances [1, 4]. Next, since the software implementations of these VNFs are at their infancy, they can be rather buggy and are susceptible to failures. To address this issue of bugs, operators and VNF vendors are looking toward continuous integration and continuous deployment (CI/CD) to rapidly roll out changes to VNFs. However, these frequent updates again have the potential to impact service unless handled appropriately. Finally, the availability of a single cloud instance depends on the collective availability of the building, mechanical electrical plumbing (MEP), hardware infrastructure (server, storage, and network), cloud orchestration software (e.g., OpenStack), and so on. Given these constraints, most cloud instances are usually designed to offer 99.9 percent (three 9s) availability (43.2 min downtime per month). Unless addressed carefully, the unavailability of VNFs can result in significant down time for customers and may violate the service level agreements (SLAs) that have been made.

Unfortunately, vendors of VNFs have thus far focused on capabilities and performance to match the physical network functions. As a result, they provide very limited resiliency features. Note that a thorough treatment of resiliency demands that we address all aspects of NFV (VNFs, NFVI, and MANO). The reason is that a service's resiliency depends on all these aspects. On one hand, we should design VNFs by taking advantage of the resiliency offered by NFVI and MANO. On the other hand, VNFs should provide capability that NFVI and MANO can leverage to improve their resiliency. However, given the large scope of the topic, we focus specifically on VNF resiliency in this article and briefly discuss the resiliency of NFVI and MANO.

The resiliency feature of VNFs should be able to gracefully handle both planned maintenance (e.g., upgrading VNF software or reconfiguring VNFs) and unexpected failures [5]. However, due to the heterogeneity of devices, velocity of network evolution, and complexity of management, it is challenging to maintain high availability of services, especially when failures happen [5]. Hence, we need mechanisms built into VNFs that can detect and recover from or even prevent failures. With well designed VNF resiliency, we can maintain the desired service level for customers while reaping the benefits provided by NFV.

[1] https://www.onap.org/ (Accessed on April 14, 2017).

Digital Object Identifier: 10.1109/MCOM.2017.1601201

The authors are with AT&T Labs.

0163-6804/17/$25.00 © 2017 IEEE

For the rest of this article, we first describe the resiliency requirements for VNFs. Next, we review existing solutions that help us meet these requirements. We then highlight some research challenges and future directions. Finally, we provide a concrete case study of EdgePlex [6], a network VNF, by showing how resiliency is achieved by its architecture and design.

Resiliency Requirements for VNFs

This section illustrates the resiliency requirements for VNFs, including failure management, state management, and the awareness of redundancy and correlated failures. We note that although we discuss them separately, in reality they are interconnected and may depend on the infrastructure and MANO resiliency. However, they should avoid being tied too tightly with the infrastructure, which may limit their capabilities.

Failure Management

Similar to other systems, *VNF failure management usually has four parts: design, detection, recovery, and prevention*.

Ideally we want to design VNFs in a way that minimizes the impact of failures. For example, using software quality assurance measures and safe programming languages can prevent many failures. Despite this, in practice, it is difficult to avoid unforeseen failures, and thus VNF design should allow detecting them quickly and recovering from them with minimal impact on the service.

Failure detection covers both deviations from the normal operational behavior of a VNF (not including performance degradation) and errors, for example, by checking the invariants of a building block or an algorithm. The NFV infrastructure/MANO should be able to detect *quickly* both failures that originate from VNFs (e.g., memory crashes) and those associated with the network infrastructure (e.g., link or switch failures).

The goal of automated recovery is to mask a failure to customers. There are five properties desired by any recovery architecture. The first and most fundamental one is correctness. The internal and external state of a recovered VNF instance should be consistent with that before the failure. Second, it should minimize the overhead of failure-free operations, especially the packet-processing latency. Third, the recovery should be fast enough to avoid degrading the end-to-end service, for example, not triggering the TCP timeout (although it may not always be easy to achieve in reality). Fourth, the recovery scheme should be general (e.g.,, not tailored for different VNFs) and should minimize the modifications to VNFs. Finally, it should not significantly increase the operational and management cost.

Failure prevention can take place during runtime by learning from past failures and devising appropriate proactive failure control to prevent them from occurring again. For instance, runtime experience of a VNF may show that CPU utilization above a threshold results in a failure with a high certainty. Given this, once the CPU utilization has reached the threshold, proactive failure control could trigger mechanisms to decrease CPU utilization of the VNF (e.g., by distributing load to other VNF instances).

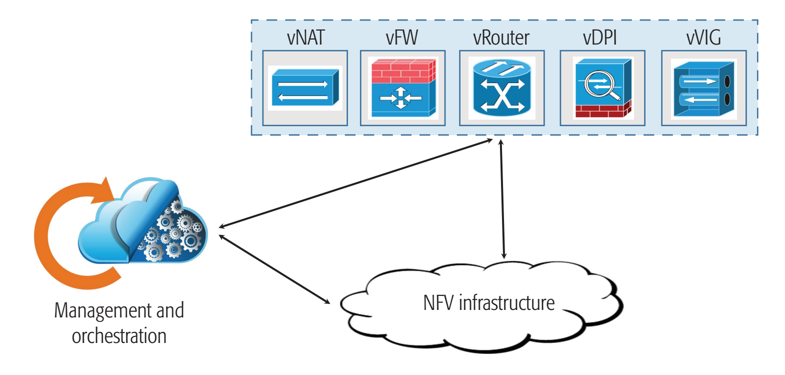


Figure 1. An abstract view of the NFV architectural framework (the concrete view is available in our previous work [1]). In the top dashed box we show a set of VNFs hosted by the NFV infrastructure. Both the infrastructure and VNFs report their status to the management and orchestration, which communicates with VNFs for tasks such as software upgrade and configuration management, and interacts with the infrastructure to spin up, migrate, and shut down VNFs.

State Management

Since most VNFs are stateful, to guarantee service continuity for either unexpected failures or planned maintenance, VNFs need to manage their state intelligently and efficiently or expose their state to the MANO. The level of difficulty to ensure service continuity depends on whether a VNF is stateless or not. For stateless VNFs, such as DNS, it is relatively easy to prevent service discontinuity, as there is usually no state information to preserve both during maintenance and for failure recovery. We note that in order to guarantee service continuity for stateless VNF, we still need to be able to detect a failure and recover from it rapidly.

Most VNFs have networking service related state, including entries in the routing information base (RIB) and forwarding information base (FIB), the mapping between IP addresses in an NAT, the mapping between incoming flows and next-hop instances for a load balancer, flow information in a firewall, and so on. Stateful VNFs should maintain the state internally or externally or use a combination of both. During the normal operational phase, these VNFs need to synchronize the state with either backups or the management systems. They should preserve the state when a failure occurs in order to shield consumers from failures and provide the correctness of rollback recovery. For example, after spinning up a new VNF instance upon a failure, this instance should restore all of the state needed and handle existing traffic with limited or no disruption. For planned maintenance, when upgrading the software of a VNF, we should migrate the state to a backup instance during traffic redirection. It may take time and is sometimes difficult to achieve flawless migration in practice, as the state is usually tightly coupled with data processing [7].

Awareness of Infrastructure Resiliency

In order to gracefully handle unexpected failures, the design of VNFs should be aware of various redundancy schemes provided at different levels. There are multiple dimensions when providing redundancy, active-active vs. active-standby (VNF level), local vs. geographic (infrastructure level), and so on. For mission-critical VNFs and VNFs serving high-value customers, active-active

Although redundancy is required mainly for unexpected failures, it may also help planned maintenance. For example, during the upgrade of VNF software, we can first conduct it on the standby instance, then switch the roles between active and standby, and finally upgrade the software of the old active instance.

mode should be considered such that when one instance fails, the other one can immediately take over the customer traffic. To optimize resource utilization for active-standby, spare resources could be shared among standby instances that are activated only to handle failures, which is hard to achieve for physical network functions.

The risk of failure caused by an OpenStack instance may be mitigated by deploying another OpenStack instance in the same cloud data center, and the risk of failure caused by an entire cloud may be mitigated by deploying a third OpenStack instance at a remote cloud. VNFs should also be conscious of the latency of infrastructure-level redundancy schemes. Local redundancy is required to ensure low latency when switching over to a redundant instance that should have *already* been instantiated on site (e.g., in the secondary OpenStack instance of the same cloud). Geographic redundancy is required to prevent the impact of an entire cloud failure by placing redundant VNF instances in a selected remote cloud. VNFs with geographic redundancy should be able to tolerate a slightly longer failover time.

The design and implementation of a VNF should be cloud-aware to achieve five 9s availability. Typical cloud (e.g., OpenStack) supports application programming interfaces (APIs) and resiliency building blocks to overcome single points of failure such as server failure, full rack failure, single cloud instance failure, or complete data center failure. VNFs can overcome single server failures by leveraging OpenStack's anti-affinity rule and placing VMs on multiple servers. OpenStack's availability zone can also be used to place VMs on different racks to overcome rack failures. VNFs deployed on two or more instances of cloud data center are not likely going to be impacted by a single cloud instance failure. The true five 9s availability can only be achieved by placing VMs on multiple geo-locations (minimum of three data centers), which requires an overarching orchestration framework [2].

Although redundancy is required mainly for unexpected failures, it may also help planned maintenance. For example, during the upgrade of VNF software, we can first conduct it on the standby instance, then switch the roles between active and standby, and finally upgrade the software of the old active instance.

Awareness of Correlated Failures

As the resiliency demands of VNFs, NFVI, and MANO are interrelated, VNF designers should take the correlated failures of NFVI and MANO into consideration. Although VNFs can leverage redundancy to enhance their resiliency, there may be hidden and deep dependencies among these seemingly independent and redundant components/instances/systems, including both hardware and software dependencies. For example, two VNFs in a redundancy group may be running on the same server due to an improper placement policy of OpenStack. As a result, a failure of the server will undermine a VNF's redundancy efforts. Different types of VNFs may also share the same software components/libraries. Thus, a bug/defect in this common part will lead to concurrent failures of these VNFs.

Correlated failures have been extensively investigated in transport networks and cloud services, for example, discovering automatically shared risk link groups [8] and determining the inter-dependencies of cloud infrastructure that are hidden due to proprietary business relationships [9]. VNFs could leverage existing technologies/solutions developed for transport networks, core networks, and cloud services to handle these correlated failures. For example, they should consider catastrophic events and inevitable accidents (e.g., earthquakes and tsunamis), which may all overwhelm redundancy schemes with large-scale correlated failures caused by these disasters. The deployment of VNF instances and the design of service chaining should minimize the impact of correlated failures by exploring the dependency information from MANO and considering the topology of the infrastructure.

Existing Solutions

In this section, we review existing solutions that we can leverage to improve the resiliency of VNFs, including state management, VNF migration, and rollback recovery. We also discuss an advanced approach based on VNF decomposition.

State Management and Synchronization

In general, there are three types of state for most stateful VNFs, control state (e.g., routing entries and firewall rules), per-flow state (e.g., TCP status and the number of packets per flow), and aggregate state (e.g., state machine of an intrusion detection system). A VNF could choose to separate its state information from the corresponding VNF instance and store it in a central location (e.g., a database). A challenge here is the synchronization of the state between the running instances and the centralized state manager.

VNFs usually have two broad classes of state: internal and external. Internal state, similar to application logic, is only used and stored by a given VNF instance. It is unique to a running VNF instance. External state can be viewed as a large distributed data structure that is shared and managed across all replicas. Based on this classification, VNFs can build a split/merge-aware state management framework to facilitate the synchronization among replicas [10], or a control plane architecture that can coordinate quick and fine-grained reallocation of flows across VNF instances during failures while maintaining the consistent VNF state [11], or a centralized data store layer to decouple the state of VNFs from the packet processing component [7].

Make-before-Break VNF Migration

When either the underlying cloud infrastructure or the orchestrator detects a situation that may lead to a failure for a running VNF instance, it may initiate a remediation procedure, for example, by proactively migrating the VNF instance to another server of the same OpenStack instance. We note that if a VNF is completely down, it is too late for migration. Migration is helpful mainly for failure prevention. An efficient VNF migration scheme should minimize the traffic disruption on the data plane.

A possible solution is the so-called *make-before-break* migration that makes the destination VNF instance run at the same time as the original instance for a short period of time. It then shuts down the source when the destination instance can handle both the control plane and data plane correctly and independently. Such techniques have been used in the seamless migration of virtual routers [12]. For example, after migrating the control plane to a new server, a virtual router can clone the original data plane on it by repopulating the FIB and then have two data planes running on both servers.

Rollback Recovery

Rollback is an operation to restore a VNF to its pre-failure state in order to recover from a failure. Checkpointing is the most widely used mechanism for rollback recovery, which periodically takes snapshots of a running VNF instance. Checkpointing does not require any modifications to a VNF. Backup instances can be activated immediately by restoring from the most recent snapshot. However, checkpointing alone cannot guarantee the correctness of recovery simply because the state changes between the most recent snapshot and the failure will be lost.

Checkpointing with buffering [13] ensures the correct recovery by holding outgoing packets in a buffer until a checkpoint has been generated. However, when there is no failure, the maximum delay added to a packet could be as high as the checkpoint interval, which is on the order of tens of milliseconds. Checkpointing with replay [4] first loads the most recent snapshot and then restores the internal state changes since the last snapshot by re-processing all duplicated packets stored in the input logger. The logs constitute a record of every non-deterministic event since the last snapshot. Although checkpointing with replay can reduce the latency of failure-free operations, it requires code modifications to a VNF.

Live VM Migration with Pass-through Devices

There are several challenges for migrating VMs with pass-through single-root input/output virtualization (SR-IOV) devices. One reason is that it is difficult to migrate the internal hardware state of the network interface cards (NICs), as it is directly managed by the virtual function (VF) driver, and the hypervisor cannot simply re-program it as it does with the software state. Moreover, the hypervisor cannot easily track the dirty memory inside the VM, which is modified by the pass-through devices when receiving packets. Thus, the received data during live migration may be lost.

Existing approaches for the live migration of VMs with pass-through devices can be divided into the following three categories [14]. The first solution uses NIC bonding (e.g., Linux Ethernet bonding driver) to bond a primary VF with a secondary virtual network interface that can support live migration. It leverages the secondary interface to forward traffic during the live migration. The second scheme implements an extra layer between the guest operating system (OS) kernel and the VF driver to emulate the hardware state, which cannot be migrated. It tracks the dirty memory by explicitly writing dummy data into the page that has the received data. The third approach enhances the hypervisor by leveraging the physical function (PF) driver of SR-IOV devices to inspect the VF state and track the memory of received packets.

VNF Decomposition

Another solution is the decomposition of a VNF. A straightforward dimension to decompose a VNF is by splitting its control plane and data plane. A VNF can also be decomposed into fine-grained software modules performing different packet processing functions, such as header lookup, flow reconstruction, and decompression [15] After decomposing a VNF into multiple components, the next step is to distribute them either within a cloud or even in more than one site such that failures in a single site do not impact the data packet processing of an end-to-end service. For example, when the server hosting the control plane fails, the data plane on another server can still forward data traffic for customers before the control plane recovers from the failure. In a later section, we illustrate such a solution in detail.

Research Challenges

In this section, we highlight several research challenges and point out a few future directions for VNF resiliency.

Offering resiliency is challenging for layer 3 VNFs due to several key differences between them and application/service VNFs that run at layer 4 or above. First, network VNFs usually interact directly and tightly with the physical network. Some VNFs may need to rely on the network fabric for a subset of their functions. For example, virtual routers may need to leverage the equal-cost multi-path (ECMP) forwarding function of a hardware switch for load balancing among multiple forwarders. Second, various technologies used for application VNFs may not be applicable to network VNFs, such as Domain Name Service (DNS)-based load balancing and anycast routing based failover. For instance, it may not be feasible to use a layer 4 load balancer for network VNFs, as the load balancer itself requires layer 3 support provided by network VNFs. Third, different from application VNFs, which can route packets among their instances using layer 3 devices, network VNFs may need to use various tunneling technologies to forward packets to the next hop.

There are other challenges for VNF resiliency in general. First of all, it is difficult to satisfy most of the requirements summarized earlier, thus requiring further investigation from the research community. In addition, it would be desirable to have a unified API from different VNF vendors for more effective management (similar to OpenFlow for switches), especially to export and import VNF state. There is a trade-off between resiliency, performance, and resource utilization, as offering high availability and reliability may affect the system performance (e.g., lowering throughput or increasing latency) and the overall resource utilization (e.g., due to redundancy). So far, we have discussed the resiliency of individual VNFs. Another challenge is related to improving the resiliency of multiple VNFs when they are chained together to offer networking services (e.g., the fault-tolerant placement of VNFs in a service chain). Finally, instead of migrating VMs among servers, which could be limited by whether a VM uses SR-IOV and other factors, a more efficient and lightweight solution would be to migrate VNFs among VMs.

Case Study: EdgePlex

As a case study, in this section we demonstrate how to enhance the resiliency by decomposing the virtual provider edge (PE) routers. The main functionality of a PE router is to connect customers to a service provider's network. Following the SDN and NFV principles, we have proposed a new architecture, called EdgePlex [6], to improve the reliability and flexibility of PE routers.

There are several challenges for migrating VMs with pass-through SR-IOV devices. One reason is that it is difficult to migrate the internal hardware state of the NICs, as it is directly managed by the VF (Virtual Function) driver and the hypervisor cannot simply re-program it as it does with the software state.

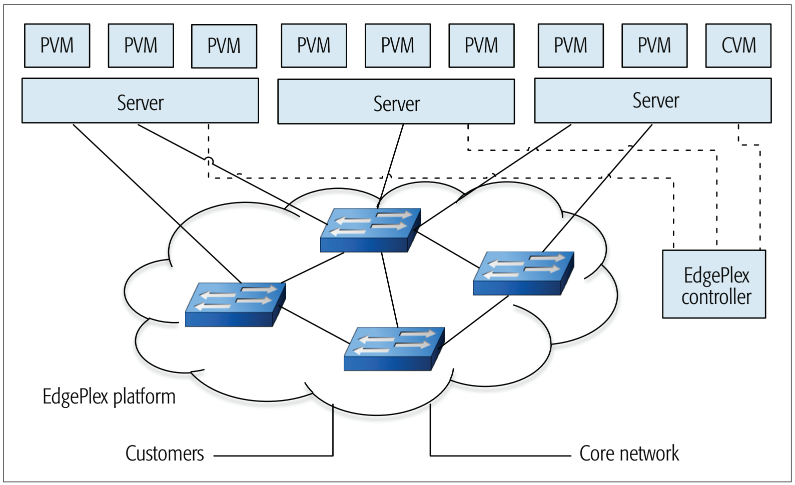


Figure 2. Architecture of the EdgePlex platform.

EdgePlex -- Architecture and Design

We present the EdgePlex architecture in Fig. 2, which is built on top of commodity servers and switches. It also leverages the recent advances in virtualization.

A key novel aspect of EdgePlex is that we use a sandboxed environment (e.g., VMs) to isolate customers. It offers the flexibility to independently move per-customer VMs within or across the platform for either planned maintenance or failure recovery. We assign each customer a VM, called a PortVM (or PVM), which is similar to a physical port on traditional physical routers. EdgePlex stores customer-specific configuration, such as routing and access control, and control and data plane state (e.g., routing and forwarding tables) in PVMs. As a result, we terminate a customer's routing session (e.g., BGP) on its PVM. This design provides network operators with the ability to migrate a customer with limited impact to others.

To connect to the core network, a simple solution is to allow each PVM to directly communicate with the core network, behaving as a virtual router with only one configured customer. However, this approach has scalability limitations, as each PVM needs to have a different multiprotocol label switching (MPLS) label and at least one Border Gateway Protocol (BGP) session to the core network. In order to address this issue, we introduce another VM, called a ControlVM (or CVM), which represents the PE router (from the control plane perspective) to the core network. The CVM speaks BGP with the core routers and relays the control plane information to and from the PVMs.

We have also designed a controller for EdgePlex to manage its components. It instantiates the per-customer PVMs and configures their routing protocols and connectivity in the hypervisors and switches. It also monitors the platform and takes action when needed (e.g., reacts to failures).

Resiliency of EdgePlex

We use existing VM redundancy technologies (i.e., Micro-Checkpointing in KVM) to protect the CVM from unexpected failures, as it is on the control plane, and its failure will make the entire router not available. Micro-Checkpointing takes periodic snapshots of the running VM and stores them on a separate machine. Upon a failure, the backup VM will quickly detect a loss of network connectivity and immediately load and activate the most recent snapshot. By using Micro-Checkpointing, EdgePlex can recover from a CVM failure without affecting its running protocols, because the failover to the backup CVM is usually faster than the timeout of control plane protocols. The backup CVM is activated only when the primary CVM fails, and thus there is no IP address conflict. To leverage the cloud resiliency feature, the backup CVM should be placed, for example, on a different availability zone of the same OpenStack instance.

Due to the number of PVMs and their limited failure impact, we can choose not to apply Micro-Checkpointing to them. If a PVM fails, it will affect only the customer configured on it, and we can re-instantiate a PVM to restore connectivity or redirect traffic to other PVMs. For the planned maintenance of PVMs, we can utilize a make-before-break live migration to reduce the connectivity interruption by modifying the source code of KVM (including libvirt and QEMU). As mentioned earlier, the key idea is to shut down the original source PVM *after* the destination PVM has been running. Because of the identical IP addresses on both PVMs, we use an OpenFlow switch to forward packets to only one of them by configuring the rules accordingly.

For failures of other components, a server failure is isolated to only customer PVMs on that server, and we can re-instantiate these PVMs rapidly. We are also investigating lightweight protection mechanisms for PVMs. The impact of a switch failure depends on how the servers are interconnected using the switches, which can be handled by redundant connections among them.

Performance Evaluation

We have implemented a prototype of EdgePlex, using a combination of custom and open source software. We now evaluate the resiliency features of EdgePlex using this prototype implementation in a testbed, as shown in Fig. 3. It has three sites, locations A, B, and C, with each site hosting an EdgePlex PE. We use the Ryu controller to configure the switches through OpenFlow. We run two groups of experiments. For the checkpointing experiments, the CVM is running on the left server of Location A and periodically sends checkpoints to the right server. For the migration experiments, we migrate the PVM of Client1 from the left server to the right one of location A. We refer interested readers to the EdgePlex publication [6] for more experimental results.

The checkpoint frequency is a key parameter of Micro-Checkpointing. On one hand, if we create checkpoints frequently, the communication overhead may be high, as Micro-Checkpointing needs to periodically transfer dirty memory to the backup VM. On the other hand, if we do not generate enough checkpoints, we may lose the changed network state (e.g., route updates) between backups. We have measured the number of transferred bytes for various checkpoint frequencies under the condition that the CVM receives 200 BGP updates every second. We summarize the results in Table 1. As we can see, the total number of bytes transferred per second decreases as we increase the checkpointing interval. Since we use this approach for only the CVM, the amount of transferred data is still manageable even in the worst case (74.7 MB for the 100 ms interval).

We have evaluated the impact of PVM live migration on the data plane using the make-before-break approach mentioned above. For the existing KVM live migration, the connectivity interruption mainly comes from the modification of switch rules, which can be done only after the migration completes. With the make-before-break PVM migration, we can modify these rules when both the source and the destination VMs are running and thus reduce the interruption to customer traffic. Suppose p is the last packet seen on the left server before the migration and q is the first packet on the right server after the migration. Their timestamps on the receiver side (i.e., either the video server or client) are T_p and T_q, respectively. We estimate the duration of migration to be $T_M = T_q - T_p$. The make-before-break approach can reduce T_M from longer than a second to be less than 100 ms. During the PVM migration, both the video traffic for Location B and the background traffic between Locations A and C were not affected.

Conclusion

When adopting NFV and moving network functions from dedicated appliances to error-prone commodity hardware, network operators should guarantee that the reliability and availability of the offered network services are not affected. In this article, we have summarized the resiliency requirements for VNFs, such as the service continuity and automated failure recovery, and reviewed several existing solutions in this area. As a case study, we have also demonstrated how to improve the resiliency feature of virtual PE routers through VNF decomposition, which offers network operators significantly more flexibility to manage their services. In our future work, we plan to apply the summarized principles to other types of VNFs, such as virtual load balancer and firewall, and further investigate the resiliency of NFVI and MANO (e.g., rerouting traffic around failures).

Acknowledgment

We thank the anonymous reviewers for their insightful comments. We thank Jennifer Yates and Oliver Spatscheck for discussions and suggestions.

References

[1] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network Function Virtualization: Challenges and Opportunities for Innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, Feb. 2015, pp. 90–97.

[2] AT&T Inc., "ECOMP (Enhanced Control, Orchestration, Management & Policy) Architecture," white paper, Mar. 2016.

[3] J. P. Sterbenz *et al.*, "Resilience and Survivability in Communication Networks: Strategies, Principles, and Survey of Disciplines," *Computer Networks*, vol. 54, June 2010, pp. 1245–65.

[4] J. Sherry *et al.*, "Rollback-Recovery for Middleboxes," *SIGCOMM*, 2015.

[5] R. Govindan *et al.*, "Evolve or Die: High-Availability Design Principles Drawn from Google's Network Infrastructure," *SIGCOMM*, 2016.

[6] A. Chiu *et al.*, "EdgePlex: Decomposing the Provider Edge for Flexibilty and Reliability," *SOSR*, 2015.

[7] M. Kablan *et al.*, "Stateless Network Functions: Breaking the Tight Coupling of State and Processing," *NSDI*, 2017.

[8] P. Sebos *et al.*, "Auto Discovery of Shared Risk Link Group," *OFC*, 2001.

[9] E. Zhai *et al.*, "Heading Off Correlated Failures through Independence-as-a-Service," *OSDI*, 2014.

[10] S. Rajagopalan *et al.*, "Split/Merge: System Support for Elastic Execution in Virtual Middleboxes," *NSDI*, 2013.

[11] A. Gember-Jacobson *et al.*, "OpenNF: Enabling Innovation in Network Function Control," *SIGCOMM*, 2014.

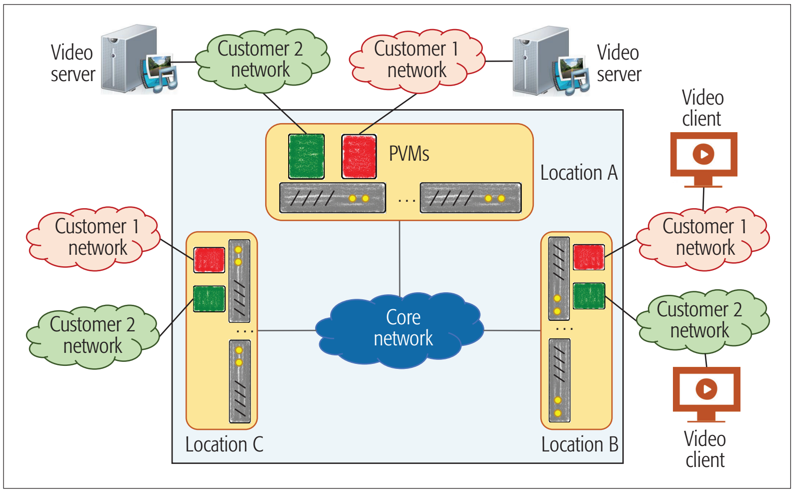


Figure 3. Testbed setup of a prototype implementation of the EdgePlex platform. The two customers are colored in green and red.

Checkpoint interval (ms)	100	200	500	1000
Data transferred (MB)	74.7	39.6	33.2	36.6

Table 1. The number of bytes transferred when checkpointing the CVM for different intervals.

[12] Y. Wang *et al.*, "Virtual Routers on the Move: Live Router Migration as a Network-Management Primitive," *SIGCOMM*, 2008.

[13] B. Cully *et al.*, "Remus: High Availability via Asynchronous Virtual Machine Replication," *NSDI*, 2008.

[14] X. Xu and B. Davda, "SRVM: Hypervisor Support for Live Migration with Passthrough SR-IOV Network Devices," *iVEE*, 2016.

[15] A. Bremler-Barr, Y. Harchol, and D. Hay, "OpenBox: A Software-Defined Framework for Developing, Deploying, and Managing Network Functions," *SIGCOMM*, 2016.

Biographies

Bo Han (bohan@research.att.com) is a senior inventive scientist at AT&T Labs – Research. He received his Bachelor's degree in computer science and technology from Tsinghua University in 2000, his M.Phil. degree in computer science from City University of Hong Kong in 2006, and his Ph.D. degree in computer science from the University of Maryland in 2012. His research interests are in the areas of network functions virtualization, software defined networking, mobile computing, and wireless networking, with a focus on developing simple but efficient and elegant solutions for real-world networking and systems problems.

Vijay Gopalakrishnan [M] is a director in the Network Research Department of AT&T Labs – Research, leading a team focused on systems challenges in the architecture, protocols, and management of networks. His research interests lie broadly in the area of networked systems, where he has worked on topics of network management, content delivery, and the mobile web. Prior to joining AT&T, he got his Master's and Ph.D. in computer science from the University of Maryland, College Park in 2003 and 2006, respectively. Vijay is a member of ACM.

Gnanavelkandan Kathirvel is a lead system architect for AT&T and on the Board of Directors for OpenStack. He leads technology efforts around NFV and AT&T's Integrated Cloud Platform. He is currently responsible for shepherding SDN and NFV projects on to AT&T's Integrated Cloud platform and a big open source advocate. Previously, he led the architecture work to support cloud convergence, building external cloud and a content delivery network for AT&T.

Aman Shaikh is a principal inventive scientist at AT&T Labs – Research. He obtained his Ph.D. and M.S. in computer engineering from the University of California, Santa Cruz in 2003 and 2000, respectively. His current research interests include service quality management, SDN, and NFV. Several tools that have emerged from his research are being used extensively by AT&T operations teams. He has also published results of his research in prestigious conferences and journals.

Wi-Fi Self-Organizing Networks: Challenges and Use Cases

Haris Gacanin and Amir Ligata

Wireless customers expect to have a guaranteed quality of experience at all times, at any location, and through different devices. Wi-Fi has become an access network of preference for service/network providers and customers as well for public and private access. This sets a challenging requirement for next-generation Wi-Fi technology to provide seamless and uniform network quality of service.

Abstract

Wireless customers expect to have a guaranteed quality of experience at all times, at any location, and through different devices. Wi-Fi has become an access network of preference for service/network providers and customers as well for public and private access. This sets a challenging requirement for next-generation Wi-Fi technology to provide seamless and uniform network quality of service). Consequently, the necessity for self-optimization of network and radio frequency segments becomes critical. This article surveys challenges and use cases of Wi-Fi self-optimizing networks (Wi-SONs) that have not been presented to date. We address technology and design challenges that shape Wi-SON as a very complex problem.

Introduction

Advanced communication techniques (e.g., multiple-input multiple-output, beamforming, channel bonding) have enabled data rates of IEEE 802.11 (i.e., Wi-Fi) to exceed 1 Gb/s. This has led to Wi-Fi deployments with an objective to provide broadband services to indoor (home, shopping malls, etc.) and outdoor (stadiums, public transport, etc.) users. Network operators are densifying their already existing private networks with community network deployments. Currently, Wi-Fi network densification is accomplished through two different aspects:

- Physical desnification by adding new access points (APs) and/or logical densification using multiple service set identifiers (SSIDs) in single-operator deployment
- Coexistence of service and/or network operators at the same location, where service operators share the same network infrastructure

Network densification is often part of the strategy to enable broadband services while aiming for uniform network quality of service (QoS). However, densification may not be the best way to achieve that goal due to lack of coordination and many unmanaged (i.e., alien) neighboring APs. Such complex and dense networks bring practical challenges related to network optimization and management (operations, engineering, etc.).

Dynamic optimization has been used for many years in communication networks to adjust configuration parameters with the aim of improving network key performance indicators (KPIs). Wi-Fi optimization [1–6] has been considered to independently maximize particular KPIs (e.g., throughput, packet loss, total transmit power, channel utilization, retransmissions, association failures, authentication failures, packet delay). Suboptimal methods in [1–6] consider Wi-Fi environments without external influences and dependencies. For example, dependencies in multi-operator dense Wi-Fi networks have varying user demands that create stochastic and highly dynamic neighborhoods. In such cases, uncoordinated optimization may trigger frequent and local reconfigurations adversely affecting neighboring APs.

Network self-organization has been largely studied in communication networks (cellular, sensor, ad hoc, and autonomic computing) [7] to automate operations and management. Self-organizing networks (SONs) have been defined as *a set of principles and concepts to add automation to mobile networks requiring less maintenance than traditional networks while improving quality of service* (QoS). Recently, network optimization was further improved in fourth generation (4G) Long Term Evolution (LTE)) networks by using various SON functions [8]. However, cellular and Wi-Fi systems have fundamental differences in the design of physical and media access layer protocols. Use cases may seem similar, but direct application of cellular SONs is not feasible in Wi-Fi systems. For example, unlike coverage optimization in cellular SONs based on antenna and/or power adaptation, a Wi-Fi SON (Wi-SON) may require band steering or adding additional APs. The application of SON functionality is not straightforward due to the following:

- An associated device may connect through a few logical networks (private and/or public SSIDs) originating from a single AP frequently operating on different channels.
- APs enable the network through multiple radios on the same or different channel(s).
- The network complexity is impacted by deployment of extenders/repeaters.
- Network selection is driven by an associated device's communication manager without any influence by the AP.
- Wi-Fi networks often operate on two different frequency bands, each of which is characterized by a very different propagation environment.
- Unlicensed bands are used by different wireless systems, yielding higher inherent interference

Digital Object Identifier: 10.1109/MCOM.2017.1600523

The authors are with Nokia Bell Labs.

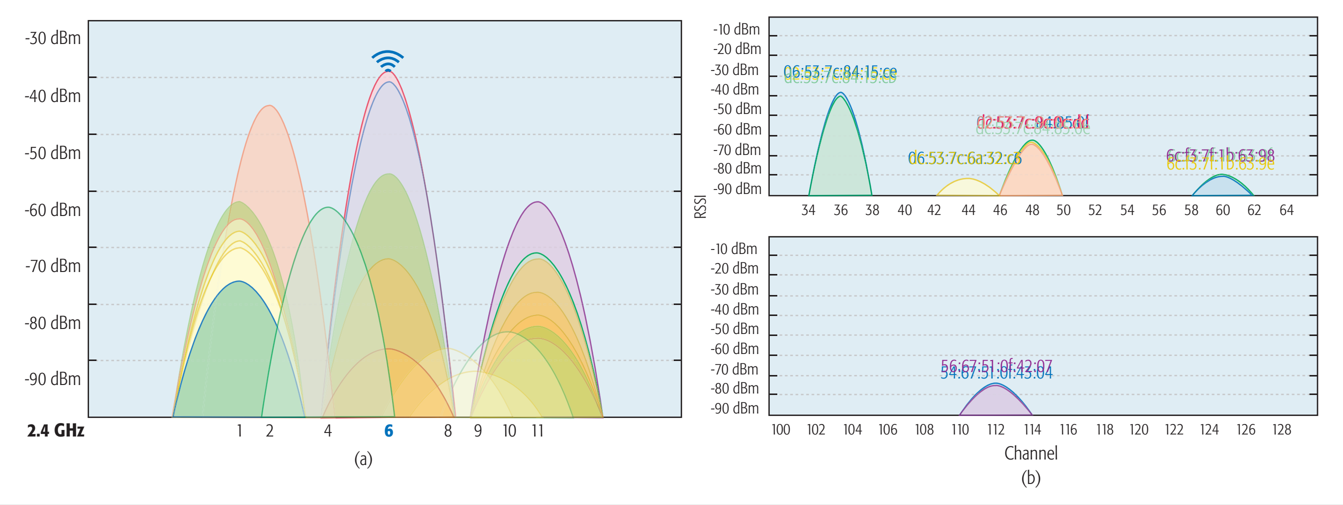


Figure 1. Spectrum utilization in residential environment: a) dense and uncoordinated deployments lead to spectrum overlapping at 2.4 GHz bands due to larger transmission range, which cause great contention and interference problems; b) fewer detected neighboring networks is mainly due to shorter transmission range at 5 GHz bands, which leads to lower contention and interference in comparison with 2.4 GHz bands.

- Wi-Fi networks in residential deployments lack synchronized backhaul, resulting in insufficient or no coordination at all. In enterprise deployment, coordination is done through a centralized controller for a specific deployment.

The above differences pose significant challenges that need to be accounted for when implementing SON capabilities in Wi-Fi networks.

In this article, we present several practically relevant Wi-SON use cases with design guidelines and challenges from the vendor, operator, and technology perspectives. The addressed issues capture new problems and unveil interesting future research directions.

For spectrum utilization in the residential environment:

- Dense and uncoordinated deployments lead to spectrum overlapping at 2.4 GHz bands due to larger transmission range, which causes great contention and interference problems.
- Fewer detected neighboring networks are mainly due to shorter transmission range at 5 GHz bands, which leads to lower contention and interference in comparison with 2.4 GHz bands.

Wi-SON Challenges

Wi-Fi network management is facing large operational challenges due to unmanaged deployments of competing service and network operators. Figure 1 illustrates the consequence of densification in the residential environment. The densification is reflected through spectrum overlapping from physical as well as underlying logical networks at different bands. To address the problems that arise in this context, Wi-Fi network operators target dynamic optimization of network configuration parameters as a reaction to the radio and network KPIs changing. This is not straightforward due to the following:

- Wi-Fi APs are low-cost devices having data-models with very limited parameter sets.
- Wi-Fi medium access control (MAC) has traditionally not been designed to work in direct coordination with neighboring APs (carrier sense multiple access with collision avoidance [CSMA/CA] is considered a very modest form of coordination).
- Unlike cellular systems, where interference is handled by network planning or physical layer design, interference in Wi-Fi is inherently handled by CSMA/CA.
- Optimization complexity in Wi-Fi is high due to dynamics and random access operations, external influences (i.e., unmanaged neighborhood), configuration dependencies, challenging propagation environments, heterogeneous devices and applications, large-scale and dense deployments, and random user behavior.

Wi-SON channel optimization function with the objective to maximize end-user throughput is highly dependent on a location and utilization of the serving and neighboring APs with respect to the target user. Since neighbor activity is stochastic, the solution space is derived from a spatio-temporal problem that is very difficult, if not impossible, to describe analytically. The following presents an overview of practical challenges related to Wi-SON design, standardization, network deployments, and cloud technology.

Design Challenges

Following the guidelines of cellular-SON design [8], the Wi-SON framework adopts three main functional categories: self-configuration, self-optimization, and self-healing, with three major architectures: distributed, centralized, and hybrid. Wi-SON framework design depends on three aspects:

1. Deployment environment (including competitors' deployment density)
2. Operator's service (public vs. private)
3. Costs strategy

The first point is largely dependent on competitors, while the last two points are mainly controlled by the operator. With this in mind, we

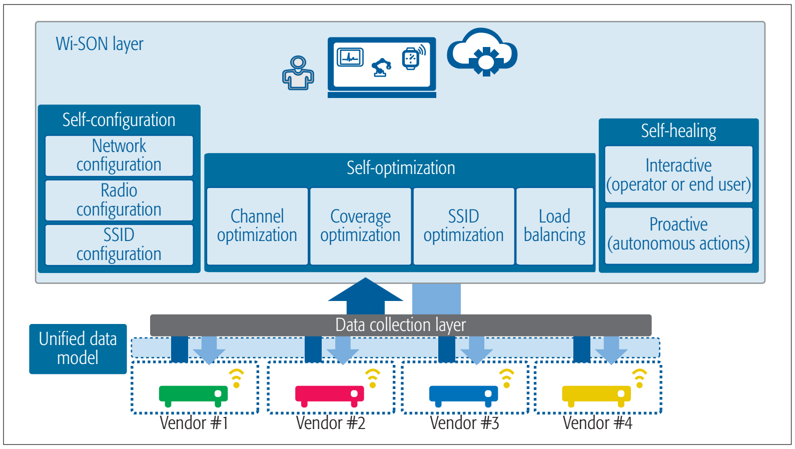


Figure 2. Wi-SON concept.

further discuss challenges related to selection of architecture and management design.

Architecture-Related: In large-scale deployments comprising thousands of managed APs, the Wi-SON design may fall into one of the following categories:

- A centralized cloud-based architecture that collects measurements from various sources and executes algorithms and policies. This approach may provide high-level management for specific use cases and may conveniently handle multi-vendor deployments.
- A distributed architecture that collects measurements and executes algorithms at the AP level. This approach is used for real-time responses to environment changes (i.e., delay-critical actions), where KPIs may be pushed to the management system for network monitoring and analysis.
- A hybrid architecture that may be implemented as a combination of the two above cases, where distributed functions are controlled based on centralized cloud-based policies.

We note here that unless the standardization of data models is properly done, the last two approaches encounter severe difficulties in multi-vendor and multi-service Wi-Fi deployments.

Function-Related: Another design category is related to the following Wi-SON functions:

- *Self-configuration*, which includes network deployment automation (initial device and network provisioning)
- *Self-optimization*, which continuously tunes configuration parameters to provide superior QoS
- *Self-healing*, which automates diagnosis and alerts for troubleshooting and recovery

Unlike cellular SON, where problem dependencies within or across each function are weak, Wi-SON has strong dependencies due to high dynamics. This may cause undesirable performance behavior if Wi-SON functions are independent. Dependencies occur when different functions alter the same parameter within overlapping optimizations. In Wi-Fi, a solution may not be straightforward due to the lack of standardized data models and coordination mechanisms. To ensure proper network operation (uninterrupted service, efficient prevention, detection), resolution of such conflicts is important.

Standardization Challenges

In most cases, a network management system relies on a client-server topology. However, practical implementation of Wi-SON may experience difficulties since APs provided by different vendors may support different management protocols. Currently, for remote management, the operators widely rely on the following strategies [9]:

- Simple Network Management Protocol (SNMP) controls parameters in the management information base (MIB), mainly for implementation of fault systems.
- The Network Configuration Protocol (NETCONF) is another management protocol for configuration of parameter settings by using remote procedure calls (RPCs).
- Broadband Forum's TR-069 CPE WAN management protocol is an IP-based protocol with extensions to different models, including Wi-Fi, LAN devices, voice over IP (VoIP), machine-to-machine (M2M), set-top box (STB), gateways (GWs), femtocells, and so on for configuration, performance testing, and monitoring.

Data collection from multiple (non-standardized) sources is a major problem, where two different vendors may have slightly different implementation of the same parameter. The problem is the fact that these protocols require different supporting architectures (auto-configuration server, database, etc.) for implementation. The third problem is the fact that these protocols have different operational requirements with respect to their ability to initiate collection sessions, support of mass vs. single data collection, data collection frequency, and so forth. Thus, it is necessary for Wi-SON to consider challenging data collection protocols.

Recently, IEEE amendments 802.11k/r/v have been added to IEEE 802.11 [10] to enable network performance optimization within a changing external environment. 802.11k enables nodes to perform or request measurements from other stations and embed this information within a management frame. 802.11r considers configuration and data exchange for fast transitions, minimizing service interruptions. 802.11v adds multicast transmission, beaconing, and sleep modes. To date, these amendments have had low industry adoption due to multi-vendor interoperability in how nodes obtain, exchange, and measure data. Also, these amendments are not suitable for dynamic optimization within the Wi-SON framework. Wider adoption requires standardized and open interface designs for data to be exposed to upper layers.

In 2006, another approach was proposed for control and provisioning wireless access points (CAPWAPs) based on Lightweight Access Point (LWAP) protocol. CAPWAP specifies management of multiple wireless APs from a central controller in multi-vendor Wi-Fi deployments (i.e., interoperability). However, CAPWAP may be considered as a failed approach as it was not adopted by industry since vendors added their own extensions, preventing interoperability.

Deployment Challenges

There are three network deployment scenarios that present challenges to network management and optimization:

- Multiple network operators coexisting
- Multiple service operators sharing the same network infrastructure
- A combination of both

In a multi-operator scenario, challenges arise mainly because the network QoS is assessed by multiple independent service operators. In practice, each service operator has its own service management system (provisioning, activation, monitoring, etc.), which makes practical implementation of Wi-SON frameworks very difficult, if not impossible. In such scenarios, Wi-SON needs to be driven by the underlying network operator with access to service KPIs if required.

In a multi-vendor deployment scenario, it is of crucial importance that optimization frameworks rely on unified data models (e.g., 802.11v). Such models refer to the set of standardized parameters that are exposed by APs and remotely available to the data collection layer in Fig. 2. This may be created in one of the following ways:

- Abstraction across data models of different vendors within the data collection layer
- Vendors following a standardized data model and protocol for remote management
- Operators taking the burden of replacing legacy equipment already deployed in the field or making large-scale firmware upgrades where applicable

Currently, we are witnessing individual efforts that are still asynchronous.

Here, it is worth mentioning two points. Future indoor deployments may consider APs that are integrated in a light system through, for example, luminaires for enterprises or light bulbs in the residential scenario. This may create enormous potential for further densification and low-cost Wi-Fi deployments. However, from an operations and management perspective, such solutions pose challenges due to low-quality RF/baseband elements with limited controlling capabilities. In this case, cloud technologies may play an important role.

Finally, coexistence between cellular and Wi-Fi systems is being addressed in 5G networks, where legacy cellular radio (i.e., LTE), new cellular radio (5G), and Wi-Fi will be integrated. The deployment scenarios being considered are either heterogenous or LTE in unlicensed bands (LTE-U). In the heterogenous scenario the network consists of unlicensed (i.e., Wi-Fi) and licensed (i.e., cellular) systems, where Wi-Fi is used for data offloading or low-cost public hotspots. In the LTE-U scenario both systems operate in the same band, and coexistence is a challenge [11].

Cloud Technologies

Network functions virtualization (NFV) has been proposed to move some network functionality to the cloud, lowering operational costs and improving network flexibility for faster service deployment [12]. Software-defined networking (SDN) was proposed for dynamic reconfiguration to balance (current or predicted) load and demand requirements [13]. Adoption of these technologies within a Wi-SON framework may start with design of open interfaces at the data collection layer and operator's cloud (i.e., B/OSS), as shown in Fig. 2. Cloud technologies may enable a programmable network controller with support of a unified data model (e.g., IEEE 802.11v) for single (multiple)-vendor deployments. Given the real-time requirements and delay-sensitive nature of the use cases (e.g., random access, interference management), they cannot be fully supported by cloud-based control. For this reason, split control of logical functionality between APs and the cloud requires further study [14, 15].

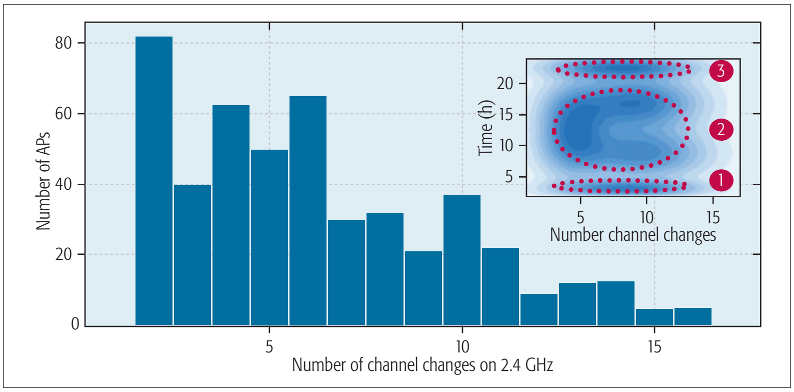


Figure 3. Distribution of the number of (non-coordinated but managed) APs as a function of the number of channel changes and a density plot of the number of channel changes throughout day.

Wi-SON Use Cases

The key features that make up Wi-SON are self-configuration, self-optimization, and self-healing. These allow a Wi-Fi network to mold itself to unique scenarios and adapt to ever changing environments.

Self-Configuration

Configuration of APs is necessary at the customer site during deployment, network maintenance, device swap, or change in the network environment. This is usually done manually by modifying pre-defined configuration files at the APs. In dense deployments, manual configuration must be replaced by self-configuration, which enables intelligent initial parameters settings. This is mostly done by using an auto-configuration server (ACS) based on various management protocols. The most important self-configuration use cases are the following.

Network Initial Configuration: This is done through remote zero-touch configuration and initial settings of newly deployed APs such as server address, IP address, and authentication. The configuration can be enabled via a dynamic host configuration protocol (DHCP) or a bootstrap protocol (BOOTP) agent through ACS. For large multi-vendor deployments, one of the major configuration issues would be remote and intelligent firmware management (i.e., updates on a 10^{4-6} scale) that needs to be dynamically configured and executed. Apart from network configuration, service provisioning may bring different challenges in the case of the multi-operator scenario.

Radio Initial Configuration: Today, for most APs, adaptive setting of initial radio configuration is not available from an operator's management system. This is usually implemented using pre-defined vendor-specific settings that include radio enabling (2.4 GHz vs. 5 GHz band), channel selection, auto-channel selection, modulation and coding scheme (MCS) index, Tx power, and

To detect poor coverage, walk-tests are used with extrapolation of coverage at locations without measurement data. However, these tests require manual interventions by the end user or network technician. This is a time consuming and costly undertaking that Wi-SON can avoid.

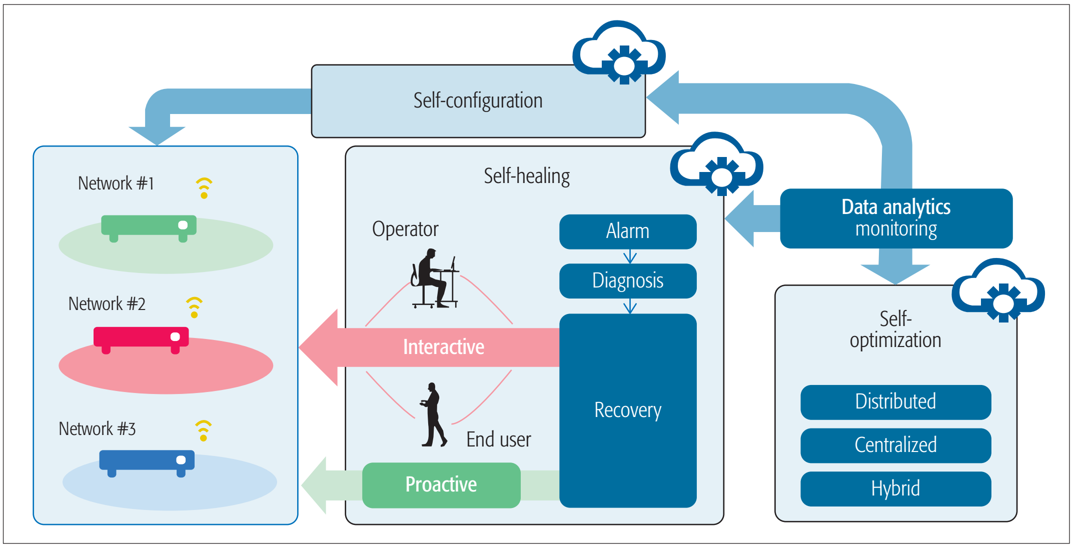


Figure 4. Wi-SON functions.

SSIDs profile installation (public vs. private), and so on. In most cases, channel selection is restricted to non-overlapping channels that may contradict the optimal channel calculated by Wi-SON.

Another example is the MCS index, which is mostly set to adaptive rate in initial settings and may not be a good solution in ultra-dense networks. In principle, a pre-defined approach will cause initial configuration per radio for all logical networks to have the same set of parameters. In Wi-SON, such an approach should be avoided, and optimization function should remotely generate a set of radio configurations for newly deployed APs depending on the neighborhood environment. In principle, the advent of high-throughput Wi-Fi systems (i.e., 802.11ad) that exploit 60 GHz frequency band will further stress the importance of Wi-SON given that the transmission range inevitably shrinks on high frequencies. In addition, higher modulation schemes in amendments like .11ac (and .11ax in 2019) bring huge opportunities for optimization of the MCS.

SSID Initial Configuration: Automated configuration of SSIDs (or neighbor cell list configuration) and updating of SSID (neighbor) relationships is another use case. Unlike in cellular networks, where a cell has a unique identification, in Wi-Fi networks an AP may have a set of different SSIDs (like cell IDs in cellular networks) depending on how many services or service operators are supported by the network. This use case enables efficient network-level planning and optimization, especially when multiple service operators are using the same infrastructure. Thus, Wi-SON neighborhood function needs to generate and update initial SSID configurations based on updates that are done when a new AP or a new service has been deployed in the network.

Self-Optimization

After self-configuration, an intelligent monitoring of the network performance is enabled to keep network KPIs within a pre-defined range. Several practically relevant Wi-SON use cases related to self-optimization are the following.

Channel Optimization: Unlike cellular networks, a target network will operate on the same channel as as its neighbors or an overlapping channel given the limited number of orthogonal channels in unlicensed band. In the event of many alien APs and/or poor channel allocation within the operator's managed network, the interference (hidden node) and/or contention (exposed node) may severely impact associated devices' throughput. To address these problems, Wi-SON function monitors the network KPIs and calculates (on a group of pre-defined clusters) the optimal channel for each AP per cluster and performs re-selection actions for APs belonging to the same cluster. Thus, the channel selection function should be defined to reduce the number of hidden and/or exposed nodes that are directly related to reduction/elimination of AP-to-AP and AP-to-alien negative effects. This is done by accounting for the availability of the entire band measurements such as noise at the AP radio, the clear channel assessment (CCA) indicator, and bandwidth. Finally, the optimal channel re-selection is executed for each cluster's AP.

Figure 3 illustrates the channel changing behavior of a cluster of (about 1000) non-coordinated and managed APs at 2.4 GHz band in a residential area. The channel optimization at a particular AP is frequently triggered, mainly due to contention or during a scheduled scan. At the top right corner, we illustrate a density plot of the number of channel changes throughout a day:

1. Night (0–5 h)
2. Day (9–15 h)
3. Evening (21–23 h)

In the first two periods users' activity influences contention between APs in dense deployments. Consequently, the channel optimization at different APs is triggered independently and more frequently (i.e., cascade effect) throughout a day. Thus, non-coordinated channel optimization cannot reach an optimum channel setting for the given cluster.

Coverage Optimization: Poor coverage is reflected by very low signal strength between an AP and an associated device, which leads to intermittent or no connectivity. To detect poor coverage, walk tests are used with extrapolation of coverage at locations without measurement data. However, these tests require manual interventions by the end user or network technician. This is a time consuming and costly undertaking that Wi-SON can avoid.

Wi-SON function detects poor coverage by monitoring the serving AP and associated device performance. At the first instance, poor coverage is handled through dynamic transmit power control (TPC) by balancing the power among the APs to increase the coverage of the area where the hole is detected. However, the function based on TPC may decrease the impact of our signal on neighboring networks (i.e., increasing or decreasing self-interference in the case of managed neighbors). This lowers the contention on the neighbors and increases channel reuse at the same time. Coverage optimization function should dynamically allocate transmit power at the AP. This will reduce or eliminate the effect of contention and interference on neighboring APs without affecting the performance of devices associated with the serving AP.

SSID Optimization: Most operators enable multiple SSIDs (per radio) in their deployments, creating a high density of service networks and creating the following challenges:

- Neighboring list optimization updates the neighboring list each time a new AP is deployed or removed and generates a table for each AP based on neighboring diagnostics measurements. The list is optimized by using the measurements with device association and radio resource management procedures defined in 802.11r in pre-defined periods. This specification defines radio and network parameters to enable management and maintenance for efficient AP reselection.
- Seamless handovers allow an end-user device to remain connected to the AP if its signal level is above reception sensitivity. Traditional Wi-Fi often has many difficulties, making these transitions in a timely and seamless manner and mobile users remain connected to the public SSID too long. To prevent service disruption (due to frequent association/disassociation), Wi-SON is designed to minimize the number of handovers in high-density AP deployments.
- Power savings are a major concern in Wi-Fi networks where the backhaul is owned by the operator as these networks can have 1000× larger deployments compared to small cells. Wi-SON introduces mechanisms that increase energy efficiency by deactivating APs that do not generate traffic. This is known as idle mode management.

Load Optimization: In Wi-Fi, network load may be handled through channel and/or band balancing.

Channel Balancing: Wi-SON function should optimize a load of the network across all channels within a band without service disruption. For example, on 2.4 GHz band, Wi-SON will select appropriate channel configuration for selected APs to evenly balance total load across all APs or non-overlapping channels. Given the nature of this use case, a fully distributed approach may not provide optimal results. The selection criteria may be based on the number of associated devices per AP, channel utilization per AP, aggregated throughput per channel per AP, bandwidth, and so on. This approach may reduce the efficiency of the channel selection function, whose primary metric in most cases is defined to reduce hidden or exposed nodes (e.g., AP-to-AP and AP-to-alien influences). Such dependencies are taken into consideration.

Band Balancing: Wi-SON allows distribution of associated devices across two available bands (e.g., 2.4 GHz band to 5 GHz) to optimize network utilization and throughput per dual-band dual-concurrent (DBDC) APs. Depending on the action trigger, there may be two practical implementations in this case;

- AP level: In this case, a distributed SON function optimizes a band selection of associated devices by looking at the device traffic type, coverage potential, associated device noise, number of associated devices, device type, and so forth.
- Device level: Here, the best band is selected by a centralized SON function by observing the band with lower interference. A policy-based band reconfiguration can then be done at a target device, or the best band recommendation is pushed to the device by the management system via SMS, email, or app message.

It is important to note that the above rules at both levels will not necessarily lead to the optimal selection with respect to the network QoS. This is because the higher band inherently has lower interference and consequently higher data rate. Thus, the function's maximum efficiency may not necessarily be reached when both bands are perfectly load balanced.

Self-Healing

Self-healing includes several phases, the first of which is providing alerts based on continuous network monitoring. Upon receiving an alert, a diagnosis is initiated to provide an analysis of the root cause. Once the problem is identified, self-healing triggers an appropriate recovery action. Belief network inference can be used to implement these three stages (alert, diagnose, recover).

Unlike in cellular SON, self-healing in Wi-SON may require end-user interaction. This is primarily a result of differing architectures (APs, extenders), multiple bands, coverage overlapping (contention, interference), and so on. As depicted in Fig. 4, self-healing recovery can be implemented in an interactive mode that involves an end user and/or operator, or a proactive mode in which recommendations are executed in an automated fashion.

Interactive Healing: When an issue is detected, the process of self-healing may require additional data to initiate the recovery process. Such data can be obtained through a field-technician application on site and fetched back to intelligent workflows at the operator side. The drawback of such an approach is extended handling time due to additional processing and involvement by an end user.

For instance, in the event of poor coverage, if self-optimization by TPC is unable to eliminate the problem, self-healing is triggered and initiates coverage optimization through deployment of a

Self-healing includes several phases, first of which is providing alerts based on continuous network monitoring. Upon receiving an alert, a diagnosis is initiated to provide an analysis of the root cause. Once the problem is identified, self-healing triggers an appropriate recovery action. Belief network inference can be used to implement these three stages (alert, diagnose, recover).

When an issue is detected, the process of self-healing may require additional data to initiate recovery process. Such data can be obtained through a field-technician application on site and fetched back to intelligent workflows at the operator side. The drawback of such an approach is extended handling time due to additional processing and involvement by an end user.

range extender. In this case, the end user is asked to take multiple measurements within the home, which are then used in workflows at the operator side. The end user is then guided to position the new AP in the most optimal location, which accounts for signal strength to the serving AP, neighboring APs, and hidden nodes.

Proactive Healing: This is a variant of automated healing in which the Wi-SON takes over the tasks of diagnosis and recovery. As illustrated in Fig. 4, data analytics enables network monitoring of vital KPIs. As an example, if a full buffer queue is detected, it can be concluded that the network is overloaded and a device reset is issued. However, given the scale and complexity of networks, proactive healing should be carefully designed.

It is important to note that the output of a self-optimization function (e.g., channel reselection) can be against the recovery process initiated by self-healing (e.g., device reboot). This may lead to possible deadlocks where a high-priority function, such as channel reselection, prevents execution of a low-priority function like device reboot that could solve the problem of network overload. Self-healing thus oversees execution of different functions and resolves possible conflicts by initiating actions that will eventually solve the problem.

Conclusion

In this article, we present a survey of Wi-SON use cases and challenges from both the operator and technology perspectives. We underline differences in comparison to cellular SON to facilitate the understanding of a Wi-SON. We further stress the relevance of standardized data models in the context of multi-vendor deployments and inter-operability. The use cases related to SON functions, and their dependencies and architecture are elaborated to clearly understand their relationships. Wi-SON is a promising concept for network operators to improve their broadband service offering, while reducing operational and maintenance costs. The Wi-SON concept thus embraces the shift of the operator's network management from on demand to interactive and proactive troubleshooting.

Acknowledgment

The authors would like to thank Dr. Mark Wagner for his valuable comments and advice.

References

[1] S. Chieochan, E. Hossain, and J. Diamond, "Channel Assignment Schemes for Infrastructure-Based 802.11 WLANs: A Survey," *IEEE Commun. Surveys & Tutorials*, vol. 12, no. 1, 2010, pp. 124–36.

[2] Y. Bejerano, S. Han, and L. Li, "Fairness and Load Balancing in Wireless LANs Using Association Control," *Proc. ACM Mobicom*, 2004.

[3] S. Pack *et al.*, "Fast-Handoff Support in IEEE 802.11 Wireless Networks," *IEEE Commun. Surveys & Tutorials*, vol. 9, no. 1, 2007, pp. 2–12.

[4] Y. Lee, K. Kim, and Y. Choi, "Optimization of AP Placement and Channel Assignment in Wireless LANs," *Proc. 27th Annual IEEE Local Computer Networks Conf.*, Nov. 2002, pp. 831–36.

[5] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE JSAC*, vol. 18, no. 3, Mar. 2000, pp. 535–47.

[6] Z.N. Kong *et al.*, "Performance Analysis of IEEE 802.11e Contention-Based Channel Access," *IEEE JSAC*, vol. 22, no. 10, Dec. 2004, pp. 2095–2106.

[7] O. G. Aliu *et al.*, "A Survey of Self Organisation in Future Cellular Networks," *IEEE Commun. Surveys & Tutorials*, vol. 15, no. 1, 2013.

[8] Lj. Jorguseski *et al.*, "Self-Organizing Networks in 3GPP: Standardization and Future Trends," *IEEE Commun. Mag.*, vol. 52, no. 12, Dec. 2014, pp. 28–34.

[9] J. Schonwalder, A. Pras, and J.-P. Martin-Flatin, "On the Future of Internet Management Technologies," *IEEE Commun. Mag.*, vol. 41, no. 10, Oct. 2003, pp. 90–97.

[10] M. I. Sanchez and A. Boukerche, "On IEEE 802.11 k/r/v Amendments: Do They Have a Real Impact?," *IEEE Wireless Commun.*, Feb. 2016.

[11] F. M. Abinader *et al.*, "Enabling the Coexistence of LTE and Wi-Fi in Unlicensed Bands," *IEEE Commun. Mag.*, vol. 52, no. 11, Nov. 2014.

[12] B. Han *et al.*, "Network Function Virtualization: Challenges and Opportunities for Innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, Feb. 2015, pp. 90–97.

[13] H. Kim and N. Feamster, "Improving Network Management with Software Defined Networking," *IEEE Commun. Mag.*, vol. 51, no. 2, Feb. 2013, pp. 114–19.

[14] R. Riggio *et al.*, "Programming Abstractions for Software-Defined Wireless Networks," *IEEE Trans. Network and Service Management*, vol. 12, no. 2, June 2015, pp. 146–62.

[15] R. Murty *et al.*, "Dyson: An Architecture for Extensible Wireless LANs," *USENIX Annual Tech. Conf.*, June 23–25, 2010, Boston, MA, pp. 1–14.

Biographies

Haris Gacanin [SM] (haris.gacanin@nokia-bell-labs.com) received his Ph.D. degree in 2008 from Tohoku University, Japan, where he worked as an assistant professor until 2010. Since 2010, he has been with Nokia Bell Labs leading the Indoor Networking Systems Department. His professional interest is related to application of artificial intelligence to enable autonomous networking of mobile and wireless systems. He is a senior member of IEICE with 180+ research publications (journals, conferences, and patents).

Amir Ligata joined the Applications and Analytics division of Nokia in 2016 as a physical layer expert. His research interest is related to customer experience management and analytics.

Coordination of SON Functions in Multi-Vendor Femtocell Networks

Anna Zakrzewska, Lester Ho, Haris Gacanin, and Holger Claussen

ABSTRACT

With the increase of network complexity, there is a high need for network management automation. This is achieved through SON principles that enable self-configuration, self-optimization, and self-healing. However, even though SON functions are meant to be autonomous, a high level of coordination among them is required. To this end, efficient conflict detection and resolution techniques are needed, especially in multi-vendor deployments. This article presents a design together with a sample implementation of a coordination scheme between three key SON functions in femtocell networks: cell ID assignment, coverage adjustment, and idle mode control. This ensures stability and continuity of the network operation even in a situation when the functions have contradicting objectives. The solution is based on the Broadband Forum TR-069 protocol and is applicable to multi-vendor networks. Simulation evaluation has shown that SON coordination reduces mean cell ID conflicts by over 30 percent and, resulting from that, call drop probability by over 40 percent

INTRODUCTION

Small cells have been widely recognized as a way to significantly increase the network capacity. The low-power indoor base stations can easily be deployed in both residential and enterprise settings. The devices are inexpensive, and their plug-and-play deployment model is possible thanks to self-organizing networking (SON) features, enabling autonomous device configuration and optimization. This approach significantly reduces the cost of installation and the need for manual network configuration.

However, as the capacity demand keeps increasing, more small cells are being added to the network. Ultra-dense deployments introduce much higher network complexity and pose additional operational challenges. Moreover, with new upcoming services, such as virtual reality (VR) and the user-centric paradigm, full autonomous adaptability is expected from the network. To configure themselves dynamically, small cells should react to network changes such as user presence or traffic demand and comply with energy consumption constraints [1, 2].

Providing fully autonomous network management is challenging as numerous network configuration parameters heavily depend on each other, and SON functions often intend to modify them at the same time, leading to configuration conflicts. To ensure proper network operation and uninterrupted service, efficient prevention, detection, and finally resolution of SON conflicts are critical. This directly translates to lower network management cost and higher customer satisfaction.

As a result of ongoing infrastructure growth and development, the network may comprise cells from different network equipment vendors (NEVs). This is especially observed in large-scale indoor residential and enterprise scenarios, which is a growing and competitive market. As femtocells are widely available and can be installed easily, we also observe a strong trend toward user deployment, similar to the approach used in Wi-Fi. The enterprise network development is often done in stages, and the supplier is chosen based on several criteria, usually price-driven, introducing a variety of NEVs to the network. An example office environment with dense multi-vendor femtocell deployment is illustrated in Fig. 1.

Importantly, NEVs implement their own proprietary procedures and SON functions, as well as the management model in general (e.g., centralized, distributed, or hybrid SON). In addition, we note that the management systems themselves may also be provided by third parties that rely on centralized processing based on standardized data models. Resolving possible SON conflicts in such a complicated environment becomes very difficult in practice and often requires manual updates by skilled technical personnel, increasing the overall management complexity and operational expenditure (OPEX). Given the growing network complexity, there is a need for coordinated configuration of various network parameters, where a cooperative approach would lead to a more balanced operation and further reduce the requirement for manual conflict resolution.

The problem of SON function coordination has been covered in the literature, and the proposals include two main approaches.

SON Function Co-Design: Clear design and separation of responsibilities for network parameters by particular functions is a way to prevent possible conflicts. These methods are widely discussed in [3], and example design principles are given in [4]. In a disjointed approach functions should ideally be based on separate parameters.

The authors present a design together with a sample implementation of a coordination scheme between three key SON functions in femtocell networks: cell ID assignment, coverage adjustment, and idle mode control. This ensures stability and continuity of the network operation even in a situation when the functions have contradicting objectives.

Anna Zakrzewska, Lester Ho, and Holger Claussen are with Nokia Bell Labs, Dublin; Harris. Gacanin is with Nokia Bell Labs, Antwerp.

Digital Object Identifier:
10.1109/MCOM.2017.1600530

Given the growing network complexity, there is a need for coordinated configuration of various network parameters, where a cooperative approach would lead to a more balanced operation and further reduce the requirement for manual conflict resolution.

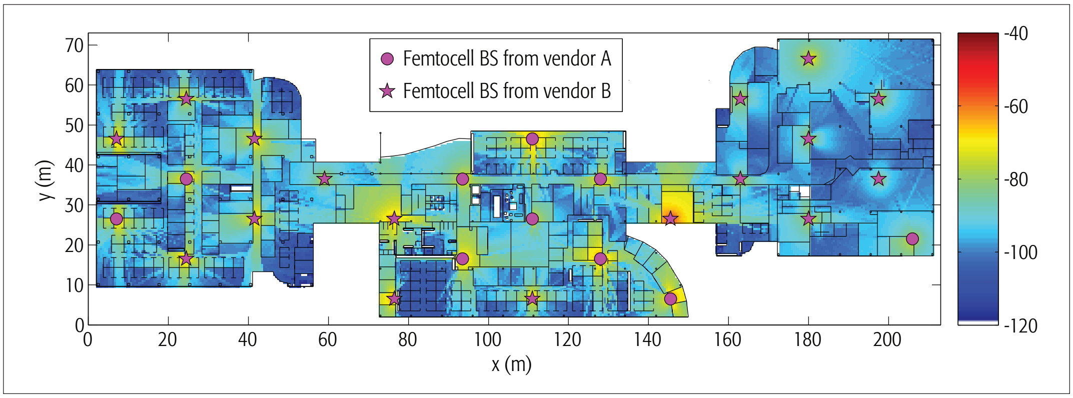


Figure 1. Sample femtocell enterprise deployment of 27 femtocells with equipment provided by different vendors.

However, several functions that rely on the same parameters can be combined into one. Unfortunately, this is not feasible if they have contradicting objectives, like those regarding power control, where one function aims to reduce it while the other tends to increase it.

SON Function Coordination: When co-design is insufficient or impossible, autonomous coordination among several SON features as well as a possibility for manual configuration of the integration between the functions are needed. A number of SON conflicts are analyzed in [5], where significant difficulty in conflict prevention is stated and possible resolution methods are given. The document introduces the SON coordination function that monitors any change requests from underlying functions based on their interdependency. It also provides a set of guidelines on conflict prevention, detection, and resolution by function prioritization. Policy-based coordination was proposed in [6], and the possible types of conflicts between the SON functions were identified and classified in [7]. The authors also provided a self-coordination framework, but that considers only two functions. Reinforcement learning techniques based on function prioritization is used for coordination in [8]. The design was tested in a scenario with three SON functions, but a discussion on the method's applicability to multi-vendor deployments is not given.

Parallel operation of multiple SON functions and satisfying multiple objectives remains an important research direction and challenge, as discussed in [9]. The coordination concepts available in the literature are very generic, giving ideas of the overall coordination process and design principles, but specific problems, especially in the multi-vendor context, have not been solved, and this was indicated as one of the standardization requirements in [10]. Along these lines, [11] covers the multi-vendor plug-and-play aspect, making the equipment from numerous NEVs configurable and operable upon joining the network, but does not extend to include other SON procedures. To the best of our knowledge, multi-vendor SON coordination methods are practically not available.

This article provides a proposal for a coordinated configuration and optimization of multi-vendor femtocell deployments. The technique includes three SON functions: cell ID assignment, coverage optimization, and energy saving management through idle mode control.

The rest of this article is structured as follows. First, the interdependency and conflicts between the three considered functions are discussed. Following that, we present our proposal to coordinate these functions. This is complemented by a description of several coordination examples and a sample implementation together with a performance evaluation. A summary is given in the final section.

Considered SON Functions

The three selected SON functions are the most relevant for network operation. Cell ID is a key network parameter, and its proper configuration affects how users access the network and influences mobility procedures. Coverage adjustment and load balancing ensure adequate coverage and efficient load distribution among the cells. Finally, enabling idle mode and putting cells to sleep when there are no users to serve helps lower energy consumption, and consequently reduce OPEX. These three functions are therefore the most relevant from a network operation point of view, especially in an indoor femtocell deployment scenario characterized by low mobility.

Cell ID Assignment

Cell IDs are parameters broadcasted by cells allowing user equipments (UEs) to discover and distinguish between different cells within their vicinity, and therefore play an important role in procedures related to network access and mobility. There are 512 primary scrambling codes (PSCs) in the Universal Mobile Telecommunication System (UMTS) and 504 physical layer cell identities (PCIs) in Long Term Evolution (LTE), further limited to a relatively small number (e.g., 20) solely for use by small cells, to reduce the amount of scanning by the UEs to lessen delays during mobility and improve battery life. The aim of cell ID allocation optimization is to assign cell IDs from those available in the pool to minimize:

- Collisions. This is where UEs can detect two or more cells that use the same IDs. This causes problems with network access and call failures because of strong interference of the pilot signals.
- Weak collisions. In LTE, this is where UEs are within range of two or more cells that use different IDs but cause interference in the control channels. This occurs when cell IDs are separated by 6 (e.g., cell IDs 1 and 7). This causes decoding problems and increased call drops due to interference.

- Confusions. This is where a cell has two or more neighbors that use the same IDs. This can cause handover failures as the handover may be directed to the wrong neighbor.

In dense small cell deployments, due to the large number of cells located within close vicinity of each other, an optimized allocation of cell IDs from the limited pool is important.

Coverage Adjustment

The pilot channel transmit power dictates the coverage area of a cell, and this has an influence on the amount of load that a cell experiences. The coverage optimization SON algorithm adjusts the coverage of cells to perform load balancing and prevent overload conditions. The considered load is the number of active UEs, as femtocell base station baseband hardware can support a limited number of active UEs. If this limit is hit, additional call attempts and handovers into the cell will be rejected. The coverage algorithm aims to prevent this by offloading load from highly loaded cells to less loaded cells. This is done by monitoring the load of the cells and dynamically increasing the coverage of cells adjacent to highly loaded cells (that are close to congestion), and/or decreasing the coverage of highly loaded cells.

However, the coverage adjustments performed for load balancing have to ensure that no gaps in coverage occur. Therefore, a minimum pilot channel transmit power, Ptx_{lower}, is calculated for each small cell based on the cell's path loss information relative to its neighboring cells, and ensures that the SON algorithm does not reduce the coverage of a cell below the level where gaps in coverage may happen between the cell and its neighbors.

The SON functions that adjust the pilot channel transmit power therefore have to guarantee that the powers do not go below Ptx_{lower} or exceed Ptx_{upper} of a cell. Ptx_{upper} is typically set by default to the highest transmit power as designed in the femtocell hardware implementation, but this can be set to a value that limits the impact of the pilot channel visibility, which is elaborated below.

Idle Mode Control

Idle mode procedures aim to reduce the overall network power consumption by putting small cells into a low-power idle mode during periods of low traffic load. When in idle mode, the small cell base stations' radio transmission functionality is switched off. The SON algorithm managing the transition of cells between idle and active modes has to do so without causing coverage holes or poor service levels. There is a requirement to maintain the coverage of the small cell network throughout the building at all times, even when there are no UEs to serve. The idle mode SON algorithm therefore first determines what are known as "coverage cells." These are femtocell base stations that cannot be placed into idle mode, regardless of load, to provide coverage within the building. Once the coverage cells have been identified, the algorithm monitors the load of the small cells. It invokes idle modes in cells whose load is below a preset threshold, provided that the cell is not a coverage cell, and its neighboring cells will not become overloaded as a consequence of idle mode being invoked. To prevent coverage holes, the idle mode SON algorithm recalculates the Ptx_{lower} of neighboring cells, taking into account the removal of the cell being put into idle mode. These Ptx_{lower} values are then applied to the neighboring cells before idle mode is invoked

SON Function Interdependency Analysis

All the considered SON functions heavily depend on the transmit power settings, and the following conflicts are possible:

- Coverage adjustment-cell ID assignment: An increase of the pilot signal transmit power in a network with stable cell ID assignment may cause cell ID conflicts, leading to dropped calls and handover problems. This happens because with the power increase, visibility of the cells increases as well, which may result in assignment conflicts among the cells that had no such visibility before.
- Idle mode-cell ID assignment: When a cell is switched on, it typically operates at its maximum pilot signal transmit power allowed by the hardware. When it is put in the idle mode, its neighbors have to increase their transmit power to maintain coverage. This can cause very drastic changes in transmit power settings. As every cell needs a cell ID assigned to operate, it is crucial to not only assign a cell ID but also coordinate its power level to mitigate the possible conflicts as explained above.

If not coordinated, cell ID allocations in current SON solutions may result in sub-optimal assignments or require constant reallocations when changes to coverage are made by coverage or idle mode SON functions. This in turn would translate to service disruption for users, such as network access difficulties, decoding problems, mobility issues, and dropped connections. With the exchange of information and coordination between the SON functions, the adjustment of coverage can be done in a way that minimizes cell ID conflicts, and at the same time the cell ID assignments can be optimized to allow more flexibility when adjusting coverage.

With the exchange of information and coordination between the SON functions, the adjustment of coverage can be done in a way that minimizes cell ID conflicts, and at the same time the cell ID assignments can be optimized to allow more flexibility when adjusting coverage.

Multi-Vendor Compatibility

Satisfying these contradictory objectives requires tight coordination and cooperation between the SON functions. This becomes a great challenge in multi-vendor deployments where individual vendors use their own configuration techniques. Enabling multi-vendor SON compatibility requires a method to control certain network parameters regardless of the NEV. The Broadband Forum (BBF) has taken essential steps to enable that. Crucial information exchange for self-optimization purposes can be provided by a centralized controller using the TR-069 signaling protocol for remote device management, and TR-196 [12] specifies the underlying data model, where transmit power is one of the standardized parameters:

- 3G:FAPService.i.CellConfig.UMTS.RAN.RF.PCPICH-PowerInUse
- 4G:FAPService.i.CellConfig.LTE.RAN.RF.Reference-SignalPower

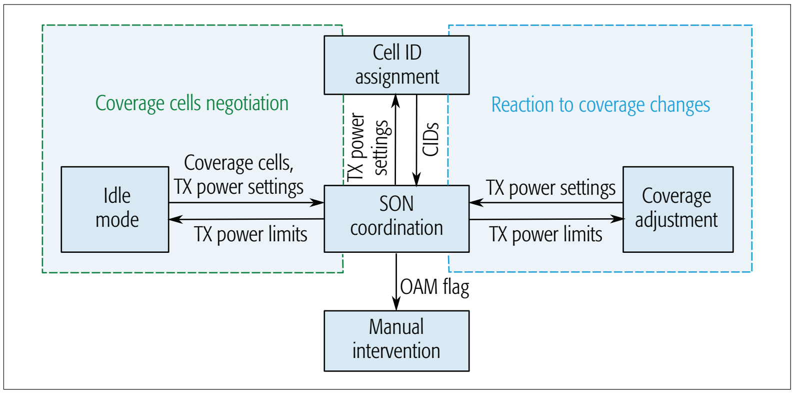


Figure 2. Proposal of coordination between coverage, idle mode, and cell ID assignment functions.

Coordination Scheme Proposal

In this section we propose a practical approach to SON coordination that can be implemented even in multi-vendor deployments. The method facilitates cooperation among three SON functions that are highly dependent on the power settings. The basic principles of the new SON coordination are illustrated in Fig. 2. The goal of a coordination scheme is to minimize the impact of the interactions caused between the adjustment of pilot transmit powers by the coverage and idle mode functions, and cell ID allocations. The proposal is therefore based on the transmit power settings by changing the Ptx_{lower} and Ptx_{upper} parameters that are used by the coverage and idle mode SON algorithms, as described previously. This is done via a central SON coordinator that interfaces all the SON functions and limits the transmit power. Through an interaction with the cell ID function, it influences the choice of the coverage cells, and mitigates possible cell ID conflicts resulting from the coverage change. The building blocks of the coordination scheme are described in detail below.

Coverage Cells Negotiation

Based on the network measurements, the idle mode SON function determines the set of coverage cells together with their minimum transmit power [13]. This happens any time a cell is physically added or removed from the network. This operation can be seen as interactive control between idle mode and coverage adjustment, and a step toward coordination among more SON functions. Since providing a conflict-free cell ID assignment for the selected coverage cells may be impossible, we propose that this is controlled via a central SON coordinator. If a cell ID assignment conflict occurs at the coverage cells, the SON coordinator limits the Ptx_{upper} on the conflicting cells and requests a new set from the idle mode function.

Cell ID assignment

All the considered SON functions rely on the transmit power. While coverage and idle mode functions have direct control over this parameter, the cell ID allocation depends on them. However, thanks to the SON coordinator, the transmit power can also be controlled in a way that is beneficial for the cell ID assignment function. Once the coverage cells have been agreed, we propose that the cell ID assignment is done in two steps: first to the coverage cells, and then to any other active cells (i.e., those not in idle mode). This exchange of information between the idle mode and cell ID assignment functions in the negotiation phase ensures a better cell ID assignment. It results in lower cell ID conflicts when cells are put into idle mode and removes the need for cell ID reassignments when idle modes are invoked. Any of the available cell ID assignment methods can be used at this stage. Therefore, no particular algorithm is presented in this article, but readers are referred to [14] instead.

Reaction to Coverage Changes

A modification of the transmit power at any cell results in a coverage change. This may impact the cell visibility, and therefore it needs to ensure that such a change does not cause serious cell ID assignment conflicts. When the coverage adjustment or idle mode SON functions wish to modify the power settings of any cell, the proposed changes are sent to the SON coordinator that uses the cell ID function to evaluate if this adjustment will result in cell ID conflicts. If the SON coordinator determines that power modification impacts the network in a serious manner and results in a number of conflicts above a certain threshold, it changes Ptx_{upper} of the relevant cells to ensure the stability of the cell ID assignment.

Manual Invervention

Even though the proposed coordination scheme is designed to be fully autonomous, it is essential to provide manual support when the coordination actions described above are unsuccessful. In that case, a flag is raised at the operation, administration, and maintenance (OAM) center to notify the system operator, who can take remedial actions. The operator may proceed with one of the approaches: either accept the proposed solution or change the system settings and recalculate optimal configuration, or to perform r-provisioning by, for example, increasing the number of cell IDs available.

The proposed centralized SON coordination scheme together with all the individual functions could be implemented in a network in one of the two ways. In the native approach, these functions would be a part of the operator's own OAM system interfacing with the femtocells' serving gateway (SGW). Another implementation approach assumes that the device management solution is provided by a third-party vendor which controls the devices using the BBF TR-069 signaling protocol. Since the proposed method relies on the standardized device/network parameters, it is compatible with both approaches.

Example

In this section, we first discuss the key steps using a conflict resolution example and then show an implementation of the proposed SON coordination scheme to demonstrate its principles and gains.

Concept Demonstration

The coordination process is initiated by the SON coordinator and the execution of the idle mode SON function, which calculates a set of coverage cells together with their minimum transmit powers. The set needs to meet the predefined quality criteria specified by the operator (e.g., available cell ID pool or acceptable level of conflicts). If a satisfactory cell ID assignment cannot be found for a suggested coverage cells set, a new set is requested by the coordinator given the new limits on the Ptx_{upper}. The number of such requests can be upper limited, and if the acceptable set cannot be found in a number of attempts, a request for manual support is sent. Through the SON coordinator, the final set of coverage cells is then taken as an input by the cell ID allocation function.

To illustrate possible interactions between the considered SON functions, a conflict-free network configuration is shown in Fig. 3a, whereas in Fig. 3b the user demand significantly increases in the lower part of the deployment. The coverage adjustment algorithm increases the power of Femtocell4 and Femtocell6 to meet the demand. However, that has a serious impact on the operating network. With the power increase, visibility of the cells increases as well, which leads to assignment conflicts: Femtocell4 and Femtocell6 use cell ID 2, so their cell IDs collide and also cause confusion at Femtocell5, as shown in Fig. 3b.

As cell ID reassignment is not possible during normal traffic hours, we propose tight coordination between coverage adjustment and cell ID assignment functions. If the coverage adjustment seriously affects cell ID assignment, the SON coordinator can lower the maximum transmit power, Ptx_{upper}, on the colliding cells. The new imposed value cannot be lower than the nodes' minimum transmit power, Ptx_{lower} to provide minimal coverage. An example adjustment for Femtocell4 and Femtocell6 is shown in Fig. 4a. The coverage is then recalculated with the new input data to mitigate the negative impact on the cell ID assignment. The improved setup accommodates higher transmit powers by cells in the lower part of the figure while ensuring a stable conflict-free cell ID assignment, as shown in Fig. 4b.

Implementation Demonstration

A fourth generation (4G) multi-vendor enterprise femtocell network is considered in the demonstration, as shown in Fig. 1. The femtocells operate at 2 GHz and have a maximum total transmit power of 20 mW. The simulated scenario was generated using the Wireless System Engineering (WiSE) 3D ray tracing tool [15]. The coverage cells have been negotiated, and seven femtocells have been selected to provide minimal coverage [13]. For the cell ID allocation, an algorithm described in [14] is used, and 20 cell IDs are available. Over time, users gradually arrive and leave the enterprise premises, and two such cycles are simulated. There are no users between the two cycles around iteration 400.

In Fig. 5, the x-axis is essentially a timeline, where every iteration represents a time interval (we assume 3 min) when the coverage SON function is executed. These changes in the user presence and load trigger the load balancing and idle mode functions [13]. For the purpose of the idle mode procedure, the underload threshold (UT) was set to one user, so as not to keep an empty cell active. The overload threshold (OT) is configurable, and depends on the hardware and maximum number of connected users. Assuming that

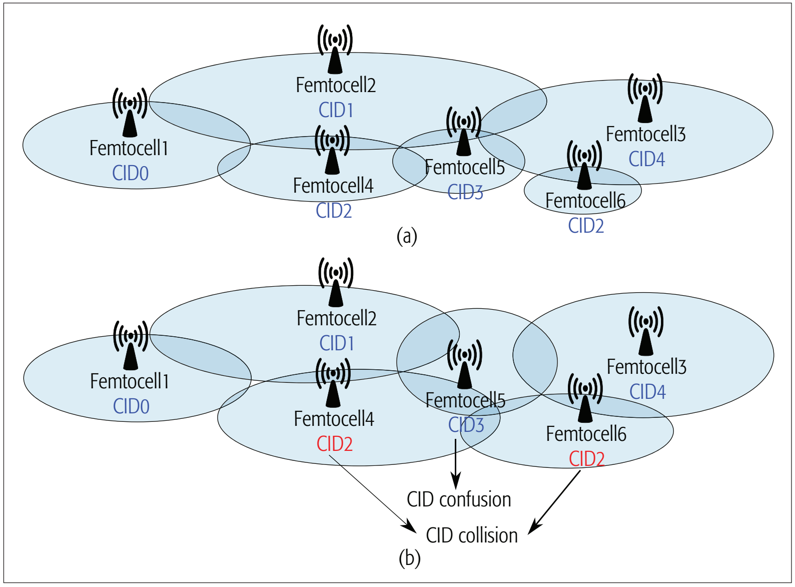


Figure 3. Example of coverage adjustment without coordination: a) stable network operation: 6 femtocells and 5 cell IDs; b) coverage adjustment causes CID conflicts.

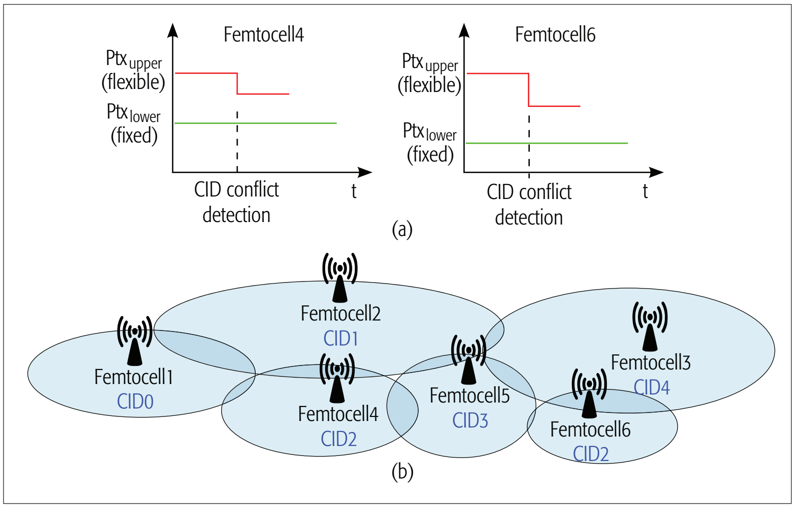


Figure 4. Example of the proposed coordinated resolution of a cell ID conflict: a) maximum transmit power limitation of the conflicting femtocells; b) coverage readjustment after transmit power change.

a femtocell can support 12 active users, the OT is set to 10 users. The SON function coordination aims to minimize the number of cell ID conflicts, which has a large impact on the network performance. As the number of active cells changes due to idle mode, the number of cell ID assignment conflicts per active femtocell is chosen as the key performance indicator (KPI).

To illustrate the impact of the coordination mechanism components, coverage cells, and transmit power limitation, we analyze them as working pairs before discussing their fully coordinated operation. Let us first investigate the effect of the power limits, assuming that cell ID assignment function is not aware of the coverage cells. Focusing on the top two plots from Fig. 5, the green dash-dotted curve shows the cell ID conflicts when power limitations by the SON coordination are active, and markers indicate occasions where power limitations have resulted in conflict reduction. In this case, 9 such situations occurred

The SON function coordination aims to minimize the number of cell ID conflicts, which has a large impact on the network performance. As the number of active cells changes due to idle mode, the number of cell ID assignment conflicts per active femtocell is chosen as the Key Performance Indicator.

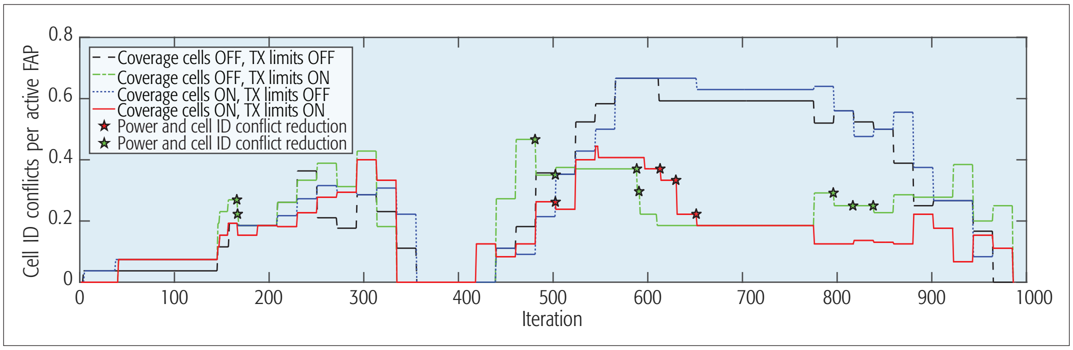


Figure 5. The influence of the two coordination method components, that is, coverage cells and transmit power limitation, for a sample network deployment.

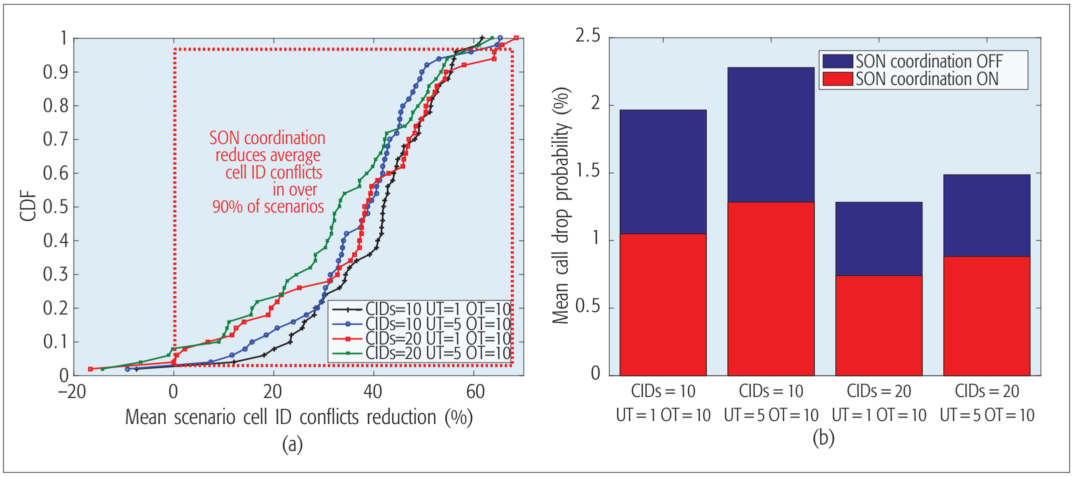


Figure 6. Performance of the SON coordination function: a) cell ID conflict reduction (50 random network deployments of 27 femtocells); b) mean call drop probability (λ = 25 calls/min, t_{call} =3 min).

and 11 cell ID conflicts were eliminated. Enabling power limitation alone reduced the average number of cell ID conflicts per active femtocell in the considered scenario by 29.54 percent, when compared to the baseline where no SON coordination is present (dashed black curve).

Next, we analyze the effect of coverage cells negotiation and consider the scenario where coverage cells are activated and the power limitation option is turned off (blue dotted curve). Please note that although the baseline cell ID assignment function does not take into account coverage cells when allocating cell IDs, the idle mode function is aware of them, thus preventing coverage holes. In the initial phase it is clear that enabling coverage cells causes more cell ID conflicts than when the SON coordination is turned off. This happens as the assignment is more restricted to provide full orthogonality among the coverage cells. On average, maintaining orthogonality among the coverage cells caused 5.89 percent more cell ID conflicts among other femtocells in the network. On the other hand, once numerous cells are turned off to save energy, only coverage cells are operating. Therefore, any conflict at this stage is much more serious, as it affects larger network areas. This is visible toward the end of the experiment.

Finally, we analyze the performance of the proposed coordinated scheme (red solid curve) including both coverage cells by the idle mode and cell ID allocation function, together with the transmit power limitation. In the sample scenario SON coordination yields 42.36 percent reduction of cell ID conflicts per active femtocell when compared to a non-coordinated case.

The example scenario above is used for illustration purposes. To assess the gains of the proposed coordination scheme, it has been evaluated over 50 random deployments of equal size (i.e., 27 cells) in the same enterprise building as in Fig. 1. The user demand changed according to the same pattern as described above throughout all the scenarios, and the evaluation was done for 10 and 20 cell IDs. The results are shown in Fig. 6a. In over 90 percent of the evaluated random scenarios we observe a significant reduction of cell ID conflicts, with a mean between 30 and 40 percent. The scenarios that do not show the gain are related to: 1) very low UT (e.g., UT=1), and a cell in conflict cannot be put into idle mode; 2) very few femtocells in hotspot areas (poorly dimensioned); higher user demand thus requires higher transmit power and may result in additional conflicts if offloading is not possible.

Moreover, it has been demonstrated that cell ID conflict resolution due to SON coordination reduces the call drop probability by over 40 percent. This was evaluated assuming that if a cell-edge user is served by a cell affected by an ID conflict, the chance of disconnecting is 50 percent. Detailed results are given in Fig. 6b.

Conclusions

As the network complexity increases by large numbers of devices and equipment from numerous NEVs, additional steps need to be taken to ensure stable network operation. This requires multi-vendor compatibility and coordination among SON techniques to facilitate network management. A novel standards-compliant multi-vendor coordination mechanism among SON functions has been presented in this article. The method enables coordinated execution of cell ID assignment, coverage, and idle mode optimization SON functions through regulated transmit power adjustment via the TR-069 remote device management protocol. An example method implementation has been presented demonstrating the possible coordination gains. Even though the evaluation was done for femtocell deployments, the principles of the proposed mechanism remain unchanged, and the method can be used in any small cell scenario. In the case of an outdoor network, the scheme may include other power-dependent functions, like those related to mobility robustness optimization. The approach presented in this article can be perceived as one of the first steps in the area of multi-vendor SON coordination, where a high need for standardization has already been identified.

A novel standards-compliant multi-vendor coordination mechanism among SON functions has been presented in this article. The method enables coordinated execution of cell ID assignment, coverage and idle mode optimization SON functions through regulated transmit power adjustment via the TR-069 remote device management protocol.

References

[1] Alcatel-Lucent, "Build a Superior Customer Experience around Small Cells," Strategic White Paper, 2015.

[2] 4G Americas, "Self-Optimizing Networks in 3GPP Release 11: The Benefits of SON in LTE," Oct. 2013.

[3] S. Hamalainen, H. Sanneck, and C. Sartori, *LTE Self-Organizing Networks (SON): Network Management Automation for Operational Efficiency*, Wiley, 2012.

[4] Z. Altman *et al.*, "On Design Principles for Self-Organizing Network Functions," *2014 11th Int'l. Symp. Wireless Communications Systems*, Aug. 2014, pp. 454–59.

[5] 3GPP TS 28.628, "Self-Organizing Networks (SON) Policy Network Resource Model (NRM) Integration Reference Point (IRP); Information Service (IS)," tech. rep., v.13.2.0, Sept. 2015.

[6] T. Bandh *et al.*, "Policy-Based Coordination and Management of SON Functions," *2011 IFIP/IEEE Int'l. Symp. Integrated Network Management*, May 2011, pp. 827–40.

[7] H. Y. Lateef *et al.*, "LTE-Advanced Self-Organizing Network Conflicts and Coordination Algorithms," *IEEE Wireless Commun.*, vol. 22, no. 3, June 2015, pp. 108–17.

[8] O. C. Iacoboaiea *et al.*, "SON Coordination in Heterogeneous Networks: A Reinforcement Learning Framework," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, Sept. 2016, pp. 5835–47.

[9] O. G. Aliu *et al.*, "A Survey of Self Organisation in Future Cellular Networks," *IEEE Commun. Surveys & Tutorials*, vol. 15, no. 1, 2013, pp. 336–61.

[10] L. Jorguseski *et al.*, "Self-Organizing Networks in 3GPP: Standardization and Future Trends," *IEEE Commun. Mag.*, vol. 52, no. 12, Dec. 2014, pp. 28–34.

[11] 3GPP TS 32.508, "Telecommunication Management; Procedure Flows for Multi-Vendor Plug-and-Play eNode B Connection to the Network," tech. rep., v.12.0.0, Sept. 2013.

[12] Broadband Forum, "CPE WAN Management Protocol Femto Access Point Service Data Model (Corrigendum 2)," Feb. 2015; https://www.broadband-forum.org/cwmp/tr-196-2-0-2.html, accessed 3 Nov. 2016.

[13] L. Ho, H. Claussen, and H. Gacanin, "Self-Optimization of Coverage and Sleep Modes of Multi-Vendor Enterprise Femtocells," *Proc. 6th Int'l. Wksp. Self-Organizing Network*, 2016, pp. 42–48.

[14] A. Wielgoszewska *et al.*, "A Centralized Method for PCI Assignment with Common Reference Signal Frequency Shift Control," *Proc. IEEE ICC*, May 2016, pp. 1–6.

[15] S. Fortune *et al.*, "WiSE Design of Indoor Wireless Systems: Practical Computation and Optimization," *IEEE Computational Science Engineering*, vol. 2, no. 1, 1995, pp. 58–68.

Biographies

Anna Zakrzewska [M] is a member of technical staff at Nokia Bell Labs, Dublin, Ireland, working on heterogeneous networks optimization, management, and future radio access. Before, she was with NTT Communication Science Laboratories in Japan and the European Commission Joint Research Centre in Italy. She holds an M.Sc.Eng. degree from Wroclaw University of Technology, Poland (2008), and a Ph.D. from the Technical University of Denmark (2014). Her research work has been recognized with several scholarships and paper awards.

Lester Ho [SM] is a Distinguished Member of Technical Staff at Nokia Bell Laboratories, Dublin. He obtained his B.Eng. degree in electronic engineering in 1999, and his Ph.D. degree on self-organizing wireless networks in 2003, both from the University of London. He joined Bell Labs in 2003, where he performed research in wireless communications, particularly in small cells, self-organizing networks, and network optimization. He has over 40 patents granted and more than 40 publications.

Haris Gacanin [SM] received his Ph.D. degree in 2008 from Tohoku University, Japan, where he worked as an assistant professor until 2010. Since 2010, he has been with Nokia Bell Labs leading the Indoor Networking Systems department. His professional interest is related to application of artificial intelligence to enable autonomous networking of mobile and wireless systems. He is a Senior Member of IEICE with 180+ research publications (journals, conferences, and patents).

Holger Claussen [SM] is leader of the Small Cells Research Department of Nokia Bell Labs. In this role, he and his team are innovating in all areas related to future evolution, deployment, and operation of small cell networks to enable exponential growth in mobile data traffic. He received his Ph.D. degree from the University of Edinburgh, United Kingdom, in 2004. He is an author of more than 100 publications and 120 filed patent families.

Ad Hoc and Sensor Networks

Edoardo Biagioni

Silvia Giordano

Ciprian Dobre

Mobile applications and Internet of Things (IoT) platforms that utilize cloud computing technologies have become increasingly popular in recent years. The cloud provides data storage and processing capabilities that make it possible to run computation-intensive applications on devices with limited processing power. The cloud helps such devices do the "heavy lifting" when necessary. This design is challenged today by Internet delays and networking overheads. However, the cloud has a significant energy footprint and suffers from the drawbacks of extreme centralization [1]. Thus, we witness a return to a more traditional grid-like future, where resources from all over the world are fused together into the grid and commonly used for a greater goal. Instead of externalizing all the business to the cloud, the cloud is brought closer to the business through set-top box equipment and cloudlet constructs, and we witness the rise of paradigms where processing of sensed data is done on machines running closer-than-cloud, whenever possible in the same network as the sensing machines themselves.

An analysis of the work in this direction and a concrete decentralized proposal are presented in the first work, "EXEGESIS: Extreme Edge Resources Harvesting for a Virtualized Fog Environment." In the article, the authors propose to harness unutilized resources at the edge of the cloud, via a three-layer architecture that encompasses the mist, fog, and cloud. The article leverages existing cloud architectures, enabling them to interact with this new edge-centric ecosystem of devices/resources, and benefit from the fact that critical data are available where they can add the most value.

On the same topic of data collection, the second article, "Coordinate-Assisted Routing Approach to Bypass Routing Holes in Wireless Sensor Networks," analyzes face-based geographic routing in wireless sensor networks. The authors identify several issues with existing technology, and further propose a routing algorithm that solves the routing hole problem by using relative coordinate systems. With caching technologies, cloud data is accessed at lower latencies, because it is transferred closer to the destination thanks to content delivery networks. We have mobile networks spreading their operation services, with applications running at the edge of the network thanks to mobile edge computing (MEC). MEC proposes a novel network architecture concept that enables cloud computing capabilities and an IT service environment at the edge of the cellular network [2]. By running applications and performing related processing tasks closer to the cellular customer, network congestion is reduced and applications perform better.

The third article in this issue, "Crowd Associated Network: Exploiting over Smart Garbage Management System," presents an approach toward building future networks that will not rely on dense networking infrastructures — making the case for a crowd associated network (CAN). In such a network, a set of crowds complements possible communication gaps at infrastructure level, and authors demonstrate their concept for city-level implementation of a smart garbage management system (SGMS). The CAN is based on the MEC philosophy by employing a set of dedicated "agents" that run decision tasks for the operation of the network. The common denominator for many technology facelifts today is this: we want to bring data and processing closer to the devices that require it. Processing in the cloud, everyone agrees, will simply not be enough soon — for mobile and business apps, even a few seconds matter. Thus, we see today's Internet moving from a network of computers to a dynamic network of networks, merging smart devices together with traditional computer networks.

The current evolution is heading toward an increasingly interconnected, mobile, pervasive, and ubiquitous Internet of networks, which range from small wireless sensor networks to extended local area networks, all of them remotely accessible. However, all these paradigms are still based on the traditional client/server model: a (mobile/static) device sends a request for an operation, which is served back by a delegated provider. As the authors of the fourth article, "A Hitchhiker's Guide to Computation Offloading: Opinions from Practitioners," remark, we can glimpse a future where decentralization is seen as a complementary solution to today's technology. Some of the communication could be made through device-to-device direct exchanges [3], relieving some of the throughput required of the cellular infrastructure. This is especially true for computation offloading [4] and remote execution for mobile devices. The authors of the fourth article make a thorough review of developments on computation offloading and remote execution. They use their findings to provide designers with guidelines to gain a deep insight into the implementation challenges of a computation offloading system. The authors demonstrate their findings through a pilot Android-based offloading system, evaluating it over two real-time applications.

These articles provide some answers to questions about the continuing evolution of the field of wireless ad hoc networks, both in supporting different applications and in different technologies used to solve specific issues. We thank all the reviewers and the editorial team for their work and their invaluable support.

References

[1] S. Giordano and D. Puccinelli, "The Human Element as the Key Enabler of Pervasiveness," *Proc. 10th IFIP Annual Mediterranean Ad Hoc Networking Wksp. (Med-Hoc-Net)*, IEEE 2011.

[2] Y. C. Hu *et al.*, "Mobile Edge Computing — A Key Technology Towards 5G," ETSI White Paper 11 (2015)

[3] R.-C. Marin and C. Dobre, "Reaching for the Clouds: Contextually Enhancing Smartphones for Energy Efficiency," *Proc. 2nd ACM Wksp. High Performance Mobile Opportunistic Systems*, 2013.

[4] A. Ferrari *et al.*, "Reducing Your Local Footprint with Anyrun Computing," *Comp. Commun.*, Elsevier, 2016.

EXEGESIS: Extreme Edge Resource Harvesting for a Virtualized Fog Environment

Evangelos K. Markakis, Kimon Karras, Nikolaos Zotos, Anargyros Sideris, Theoharris Moysiadis, Angelo Corsaro, George Alexiou, Charalabos Skianis, George Mastorakis, Constandinos X. Mavromoustakis, and Evangelos Pallis

Abstract

Currently there is an active debate about how the existing cloud paradigm can cope with the volume, variety, and velocity of the data generated by end devices (e.g., Internet of Things sensors). It is expected that there will be over 50 billion of these devices by 2020, which will create more than two Exabytes worth of data each day. Additionally, the vast number of edge devices create a huge ocean of digital resources close to the data source, which, however, remain so far unexploited to their full extent. EXEGESIS proposes to harness these unutilized resources via a three-layer architecture that encompasses the mist, fog, and cloud. The mist network is located at the very bottom, where interconnected objects (Internet of Things devices, small servers, etc.) create neighborhoods of objects. This arrangement is enhanced by a virtual fog layer, which allows for dynamic, ad hoc interconnections among the various neighborhoods. At the top layer resides the cloud with its abundant resources that can also be included in one or more virtual fog neighborhoods. Thus, this article complements and leverages existing cloud architectures, enabling them to interact with this new edge-centric ecosystem of devices/resources, and benefit from the fact that critical data are available where they can add the most value.

Introduction and Context

Nowadays a lot of discussion is going on regarding the way the cloud paradigm can cope with the volume, variety, and velocity of the data generated by end devices (e.g. Internet of Things [IoT] sensors). It is expected to have over 50 billion of these end devices [1], currently referred to as "things," by 2020, which will create more than two Exabytes' worth of data each day. It is clear that shipping all of that data to the cloud, and processing and storing them there, as the current paradigm dictates, can run into significant bottlenecks in terms of latency and network capacity. On the other hand, it is hard to miss that the vast number of end devices, most of them utilizing some form of processing power, storage space, and network connectivity, could constitute a pristine "ocean" of digital resources, which could be harnessed and used to address the bottlenecks of the current cloud paradigm by processing and storing data close to where they are created.

In this context, EXEGESIS, building on and extending existing concepts [2, 3] such us as micro data centers, cloudlets, mobile edge computing (MEC), and fog computing (http://www.openfogconsortium.org/news; retrieved July 2016) proposes a novel three-layered architecture that is able to not only reap the resources of end users' devices, but also couple them to the cloud by providing a cross-layer orchestration platform able to deploy services that have a cloud and mist component and to provide a distributed marketplace where these resources can be traded off by any EXEGESIS stakeholder: a local authority in Athens, a small-medium enterprise (SME) in Madrid, or a corporation in Brussels. In this way, EXEGESIS envisages that it can steer new and innovative services and process efficiencies not possible with cloud computing alone.

The EXEGESIS high-level architecture is composed of three layers (Fig. 1). At the very bottom, the mist network is located, where interconnected objects (probes, sensors, cell phones, home appliance devices, small servers, small cell controllers, etc.) create a neighborhood. This arrangement is enhanced by the *virtual fog* (vFog) layer, which allows for dynamic, ad hoc interconnections among the various neighborhoods, allowing sub-groupings called "suburbs" to be formed. At the top layer resides the conventional cloud with its abundant resources that can also be included in one or more "suburbs" in order to provide compute resources and facilitate the interconnection of the various vFog elements. In this context, EXEGESIS complements and even leverages existing cloud architectures as it enables them to interact with this new edge-centric ecosystem of devices/resources and benefit from the fact that critical data are available where they can add the most value.

The key idea and challenge here is to be able to partition the three-layer infrastructure consisting of the mist, vFog, and cloud layers into logical networks whose membership can partially overlap with that of other such logical networks and to be able to dynamically remold this partitioning to ensure optimal performance and utilization of the available resources

It is expected that there will be over 50 billion end devices by 2020, which will create more than two Exabytes worth of data each day. Additionally, the vast number of edge devices creates a huge ocean of digital resources close to the data source, which, however, remain so far unexploited to their full extent. EXEGESIS proposes to harness these unutilized resources via a three-layer architecture that encompasses the mist, fog, and cloud.

Evangelos K. Markakis, George Mastorakis, and Evangelos Pallis are with the Technological Educational Institute of Crete; Kimon Karras, Nikolaos Zotos, Anargyros Sideris, and Theoharris Moysiadis are with Future Intelligence Ltd.; Angelo Corsaro is with PrismTech Corp.; Charalabos Skianis is with the University of the Aegean; Constandinos X. Mavromoustakis is with the University of Nicosia.

Digital Object Identifier: 10.1109/MCOM.2017.1600730

0163-6804/17/$25.00 © 2017 IEEE

The key idea here is to be able to partition the three-layer infrastructure consisting of the mist, fog and cloud layers into logical virtual networks whose membership can partially overlap with that of other vFog networks, and to be able to dynamically remold this partitioning to ensure optimal utilization of the available resources.

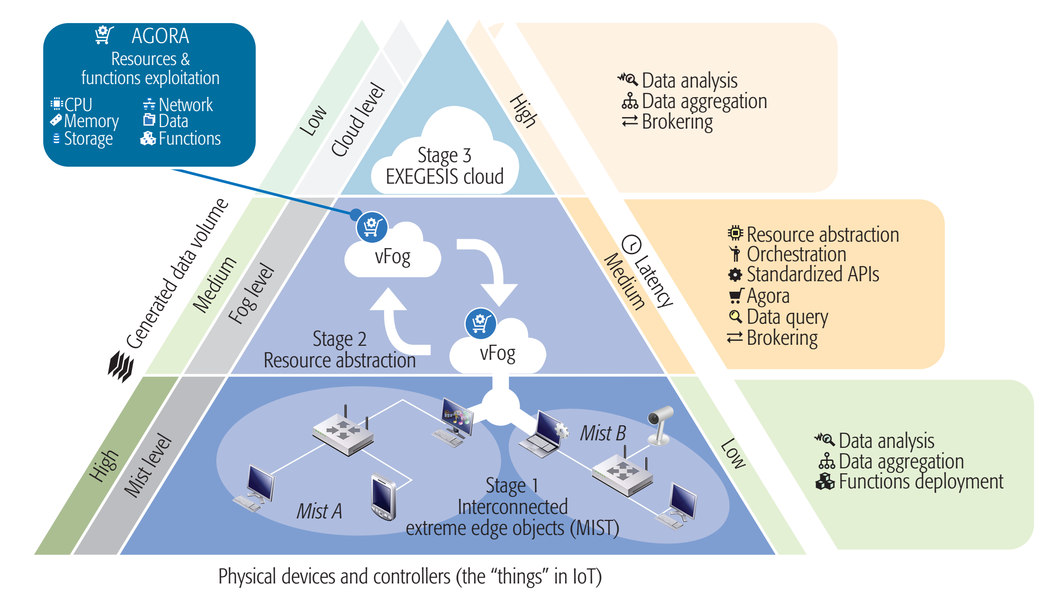


Figure 1. High-level view of the EXEGESIS concept.

Furthermore, EXEGESIS aims to enable business innovation via the deployment and use of suburb-based marketplaces named "AGORAs," stemming from a Greek word which means the place where all social and economic activity takes place. The AGORA is for EXEGESIS the system through which every infrastructure/platform provider offers over-the-top (OTT) and on-demand accelerated service/network/connectivity applications to requesting entities.

In other words, EXEGESIS aims to radically reshape the mist, fog, and cloud landscapes by merging them into one coherent whole, and then slicing and dicing that into logical entities in order to achieve optimal performance and resource utilization.

Background and Related Work

Concept

EXEGESIS, building on the concepts of edge computing [4], frugality of resources [5] and democratization of the digital economy (https://ec.europa.eu/digital-single-market/en/digital-single-market, retrieved July 2016), envisages a future where the processing, storage. and networking resources of the devices residing at the edge of the network can be harnessed and integrated seamlessly and dynamically in a flexible system architecture. In making this a reality, EXEGESIS provides the means to establish vFogs — overlays of interconnected end devices that can be intertwined with cloud resources — forming ad hoc isles of connectivity and computing, setting the basis for a common marketplace where services can easily be deployed across all layers.

The following sections describe the methodology that EXEGESIS follows in order to reach its objectives as well as the technological aspects utilized for realizing these objectives

Technical Approach

EXEGESIS proposes a new interaction ecosystem composed of three layers. At the very bottom, the mist layer is located, where interconnected objects create a neighborhood. This arrangement is enhanced by the vFog layer, which allows for dynamic, ad hoc interconnections among various mist elements, allowing sub-groupings called "suburbs" to be formed. Cloud layer resources can also be included in a suburb in order to provide resources and facilitate the interconnection of the various elements. The key idea here is to be able to partition the three-layer infrastructure consisting of the mist, fog, and cloud layers into logical virtual networks whose membership can partially overlap with that of other vFog networks, and to be able to dynamically remold this partitioning to ensure optimal utilization of the available resources.

Mist for EXEGESIS is the unified extreme edge playground where a variety of end-user (an end user can also be a company that utilizes EXEGESIS solution) devices cooperate toward abstracting, in a common virtual pool, their available resources and as such enable any legitimate entity to use these resources for hosting a variety of compute and networking tasks. The EXEGESIS mist overlay "copies" the hybrid peer-to-peer (P2P) approach where a peer can be *primus inter pares*.[1] In this context, the EXEGESIS mist network has two classes of peers (Fig. 2), *regular mist nodes* (RMNs) and *super mist nodes* (SMNs).

An RMN can be any end device having at least some processing and communication capabilities that will allow EXEGESIS to deploy its solution on it and thus transform the device to a fully operational EXEGESIS mist node. An RMN is able to interact with its corresponding SMN, first to inform it about the device's available resources and second to receive and carry out the assigned computational and/or networking tasks. To that end, a special kind of software, called the vFog agent, runs on each RMN. An RMN can be any physical or virtual entity having even a "pinch" of processing and communication capabilities.

An SMN plays two roles inside the EXEGESIS ecosystem: the role of the mist's intra-manager and the role of the mist's envoy to the vFog orchestrator. As an intra-manager, an SMN:

[1] First among equals.

- Oversees the formation of the mist network by performing operations such as the (de)registering of mist nodes
- Queries the registered mist nodes about their state and their available resources
- Creates a logical topology of the mist network along with a virtual pool of the RMNs' available resources

As an envoy, an SMN interacts with the vFog orchestrator toward:

- (De)registering a mist network to the vFog overlay
- Providing a "copy" of the SMN's virtual pool of resources, therefore enabling the vFog orchestrator to have a clear image about the available resources across the whole vFog overlay
- Mediating between vFog orchestrator and RMNs for reserving resources, assigning computational tasks, or even deploying network functions virtualization infrastructure (NFVI) elements

Following hybrid P2P's paradigm, an SMN is elected from the currently running RMNs taking into account several attributes like processing and memory capabilities, network capacity, and power level/type, among others. Acknowledging that the uncontrolled participation of mist nodes in the election process could pose security threats, EXEGESIS provides the means for "screening" the candidates list based on the EXEGESIS stakeholder's policies. The SMN is elected from the existing RMNs; it manages RMNs, and it is the point of contact to the vFog orchestrator.

The tremendous number and vast heterogeneity of the devices living on the edge of the network poses a significant challenge for EXEGESIS in forming manageable and efficiently operating mist networks. To handle this challenge, EXEGESIS proposes the development and exploitation of a middleware solution that will sit on top of each device's operating system (OS). The middleware utilizes a southbound application programming interface (API) for interacting with the OS and acquiring access to the device's actual resources and a northbound API for communicating with its vFog orchestrator. A hypervisor will be exploited for deploying in containerized form — reducing the system's footprint and increasing services deployability — the RMN/SMN module and, if assigned from the vFog orchestrator, other software units that carry out computational tasks or realize a service.

EXEGESIS proposes the idea of a vFog for managing the underlying mist networks and harnessing their available resources. As the name implies, a vFog assumes the operations of a conventional fog network (e.g., coordination of the fog nodes, provisioning of the available resources to third parties, management operations) but is not deployed over dedicated equipment pre-installed at specific places; a vFog lives on top of mist networks as an overlaid virtual entity (Fig. 2). In these configurations, the underlying SMNs will be the vFog nodes utilizing an election protocol to select, based on a set of predefined criteria (e.g., processing capabilities, storage space, network capacity, power level), the SMN that will undertake the role of the vFog orchestrator; the mind and heart of vFog's overlay. In a nutshell, the vFog orchestrator will carry out the following key tasks:

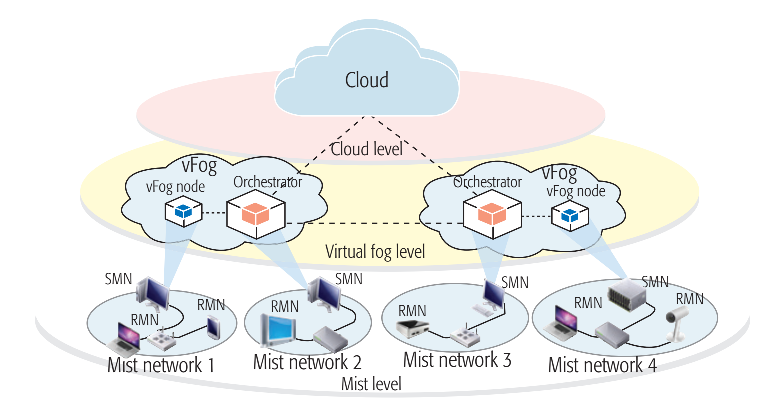


Figure 2. Two vFog neighborhoods accommodating two mist networks each.

- Perform the vertical managerial operations needed to form and maintain the vFog overlay network.
- Query the underlying mist nodes for available resources, and create an abstract pool of them.
- Provide information about the available resources to any authorized third party (including the AGORA).
- Handle horizontal communication operations (e.g., with other vFogs and/or conventional fogs).
- Exchange data with any clouds with which it belongs to the same suburb.
- Accept and forward requests for computational tasks, storage space, and deployment of services to the vFog nodes based on the needed and available resources.
- Deploy the AGORA across the vFog network.

One of the key issues that EXEGESIS attempts to tackle is to stem the tide of data flowing into and out of the cloud. This is done by injecting SMNs into the vFog network that have increased processing capabilities. These nodes will then expose their resources to the orchestration environment so that they can be used for pre-processing and filtering of data. That processing might lead to direct decision making or to a whittled down version of it being uploaded to the cloud for further elaboration. At the core of this process are heterogeneous, programmable, logic-based nodes, which are located in the vFog network and will be used for both processing and vFog suburb management. Programmable logic was selected because it offers the critical combination of high performance, low power, and complete flexibility that is necessary to successfully meet the challenges of this role.

A heterogeneous vFog node within the context of EXEGESIS will consist of a field programmable gate array (FPGA) system-on-chip (SoC), which is an integrated circuit that combines processors, programmable fabric, and, potentially, additional logic. This combination allows us to optimally balance the task load by allowing the processors to handle control-dominated tasks, like managing a vFog network and delegating all compute-intensive tasks to the programmable logic. To accomplish this, the programmable fabric needs to be virtualized so that the orchestration environment can deploy the appropriate application on it at

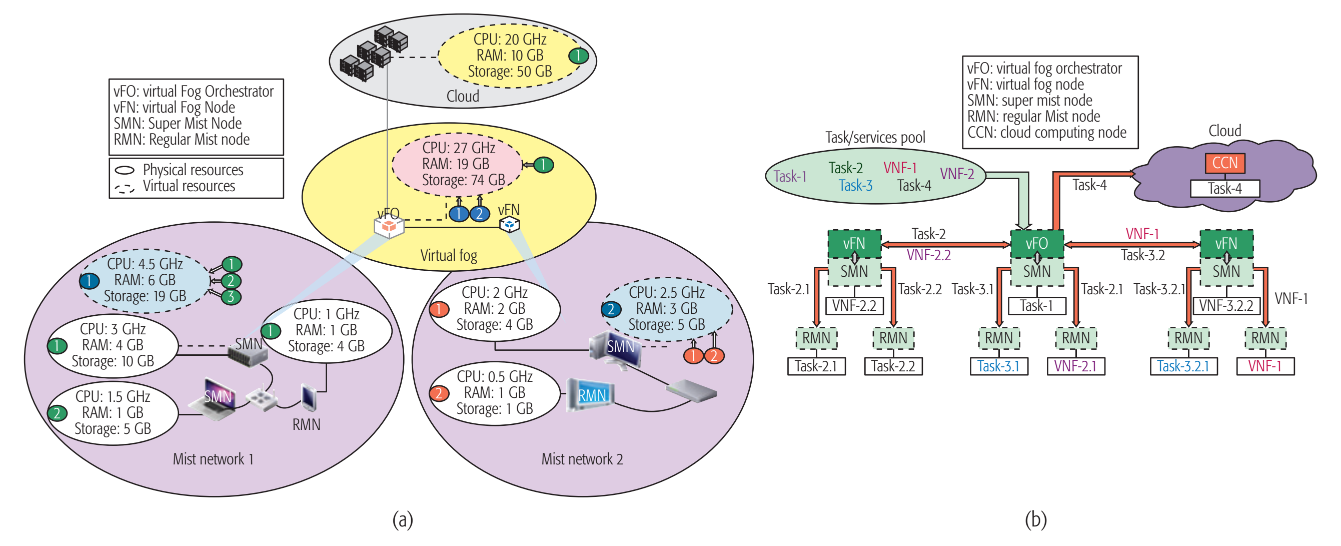


Figure 3. a) Abstraction of resources in EXEGESIS; b) deployment framework for tasks and services.

any given time. This is accomplished by executing cloud software on the processors of the FPGA SoC, which, together with the specialized hardware, enables the deployment of hardware virtual machines on the programmable logic.

Abstraction of Resources in EXEGESIS

Starting at the mist layer (Fig. 3a), each RMN, during its registration process or upon a status update, informs the SMN about the amount and type of physical resources it is willing to provide to the EXEGESIS platform. The SMN in turn abstracts this information to construct a virtual resources pool aggregating the physical resources of all the mist network nodes. Following the same paradigm, each SMN, after registering as a vFog node, delegates information about its virtual resources pool to the vFog orchestrator. At the same time, the vFog orchestrator can request and bind, if needed, more resources from a conventional cloud. In this way, the vFog orchestrator forges a new virtual pool that holds in abstracted form the physical resources across the whole vFog network.

Deployment of Services and Tasks in EXEGESIS

EXEGESIS will deploy services and perform computational tasks following a hybrid operational scheme (Fig. 3b). In such a scheme, the vFog orchestrators can receive the requests for computational tasks and service deployment. After that, the orchestrator, based on the vFog's available resources and policies and also taking into account the incoming task/service requirements, can assign each task or service to one or more vFog nodes (including itself if appropriate). In doing so, the orchestrator will utilize and extend existing work to optimize task allocation [6, 7]. In turn, each vFog node passes the request to its SMN module and, based on the mist's resources and the assigned operation's requirements, forwards the tasks to itself and also, if needed, to the appropriate RMNs. It is noted here that if a task exceeds the capacity of a vFog, the orchestrator can forward the task to another vFog or assign it to cloud computing resources. EXEGESIS' deployment framework has segmentation of tasks and services at its core. In this way, barring any security policies or specific task requirements, EXEGESIS can optimally fragment and distribute tasks to resources as required to ensure that performance targets are met.

Use Cases

Security cameras, mobile phones, machine sensors, environmental sensors, and so on are just a few of the items in daily use that create data that can be mined and analyzed. Add to it the data created in smart cities, manufacturing plants, financial institutions, oil and gas drilling platforms, pipelines, and processing plants, and it is not hard to understand that the deluge of streaming and IoT sensor data can — and will — very quickly overwhelm today's traditional data analytics tools. Organizations are beginning to look to edge computing as the answer. Edge computing exploits vFog and mist, and promotes data thinning at the edge that can dramatically reduce the amount of data that needs to be transmitted to a data center or cloud infrastructure. Without having to move unnecessary data to a central location, analytics or distributed processes at the edge can simplify and drastically speed up analysis while also cutting costs. This drastic shift in data processing paradigm propounded in EXEGESIS can be utilized in many diverse use cases. The proposed concept thus includes and investigates two concrete use cases where the proposed architecture can prove to be a game changer compared to the currently available infrastructure. These use cases, among others, are illustrated in Fig. 4, which demonstrates one possible example of an EXEGESIS architectural configuration where the four scenarios presented in the following sections are served by three vFog suburbs, each with its own mist node neighborhood. All three suburbs share a common cloud infrastructure, while each use case runs different tasks that are executed on their respective suburbs.

Enabling and Enhancing Services for Smart Cities

Cameras are ubiquitous in modern cities, and they can be used for various purposes, among which are traffic management and surveillance. Both of these applications can benefit from acceleration in the form of advanced image processing

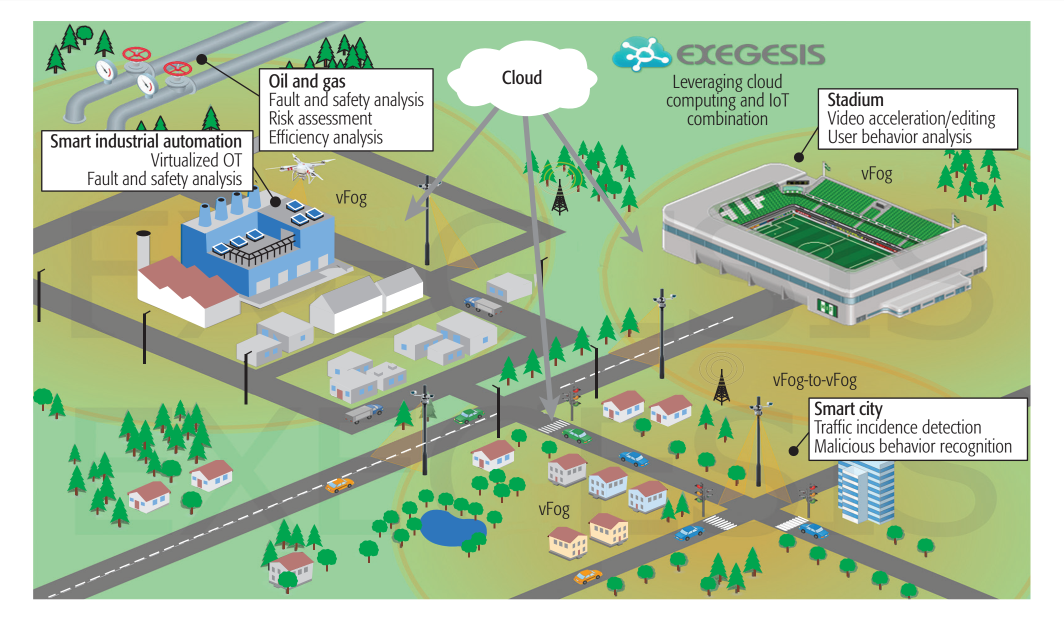

Oil and gas
Fault and safety analysis
Risk assessment
Efficiency analysis
Smart industrial automation
Virtualized OT
Fault and safety analysis
Cloud
EXEGESIS
Leveraging cloud computing and IoT combination
Stadium
Video acceleration/editing
User behavior analysis
vFog
vFog
vFog-to-vFog
Smart city
Traffic incidence detection
Malicious behavior recognition
vFog

Figure 4. EXEGESIS use cases playground.

> The smart city is going to be one of the major revolutions of the coming decades, with large urban areas, under ever-increasing pressure to accommodate a busy, fast-paced life for their citizens, turning to IoT to optimize the use of their infrastructure and thus save on cost and enable new services.

but require that different algorithms be executed (e.g., traffic management requires that the number of cars per lane or the number of cars violating traffic laws are counted, whereas surveillance demands that specific individuals must be identified).

The smart city is going to be one of the major revolutions of the coming decades, with large urban areas, under ever-increasing pressure to accommodate a busy, fast-paced life for their citizens, turning to IoT to optimize the use of their infrastructure and thus save on cost and enable new services. This entails everything from smart lightning to smart water supply to smart security, among others.

There are two issues where today's architecture is lacking: the reuse of existing infrastructure and the complexity in implementing data analysis solutions over that infrastructure. The former means that a set of input devices, say cameras in this scenario, is installed in order to be used only for one function (e.g., traffic monitoring). That function cannot be changed unless the infrastructure itself is physically altered, replaced, or duplicated. The latter refers to the fact that the process of retrieving the data from the input devices, analyzing, reaching a decision, and applying that decision is prohibitively slow and complex since all city infrastructure today is purpose built.

The architecture proposed in this article can solve both issues by creating two separate fog segments, both sharing the same FPGA-accelerated node through which the data pas that performs the appropriate analysis. The orchestrator platform makes sure the accelerated node executes the required functionality at any given time. The switch between the two tasks can be performed very swiftly, which will allow the node to perform both tasks seemingly at the same time much like a typical CPU appears to parallelize thread execution. The results of this analysis can then be either sent on for further processing (e.g., after identifying suspicious activity) to the cloud or trigger automatic reactions in other systems (e.g., manipulating traffic signals when detecting an accident and notifying emergency services automatically).

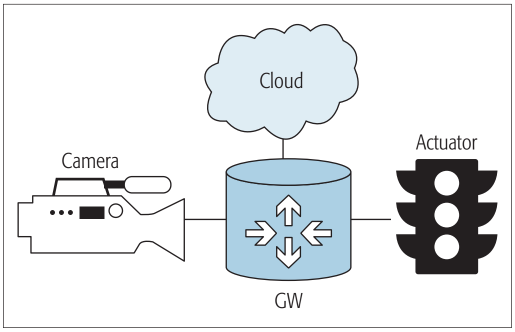


Figure 5. Overview of the simulated scenario.

Even within the narrower confines of smart traffic management, fog computing improves the performance of the application in terms of response time and bandwidth consumption. A smart traffic management system can be realized by a set of stream queries executing on data generated by sensors deployed throughout the city. Typical examples of such queries are real-time calculations of congestion (for route planning) and detection of traffic incidents. One possible case study, further elaborated on later in this article, could compare the performance of a DETECT_TRAFFIC_INCIDENT query on fog infrastructure [8] vs. the typical cloud implementation. In the query, the sensors deployed on roads send the speed of each crossing vehicle to the query processing engine. The operator Average Speed Calculation calculates the average speed of the vehicles from the sensor readings over a given timeframe and sends this information to the next operator. The operator Congestion Calculation calculates the level of congestion in each lane based on the average speed of vehicles in that lane. The operator Incident Detection, based on the average level of congestion, detects whether an incident has occurred or not. This process

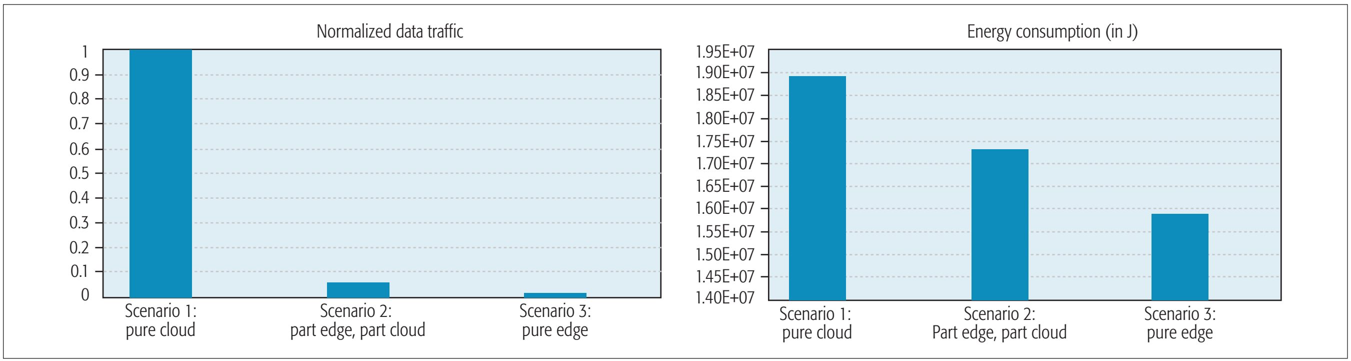


Figure 6. Simulation results showing: a) normalized network usage; b) system energy consumption.

will be implemented and executed on both fog- as well as cloud-based stream query processing engines, which will highlight the faster response times and bandwidth savings offered by the fog-based alternative.

Smart Industrial Automation

The new trend in automation is that of virtualizing as much as possible the operational technologies (OT) side of the system over contemporary IT infrastructure. The idea is simple: as virtual machines have virtualized hardware in IT, the automation industry is trying to virtualize OT hardware such as programmable logic controllers and run them over, more or less, traditional IT infrastructure.

The automation industry has been challenged for several years by the difference in innovation cycles and obsolescence rate existing between OT and IT. The result of this divergence in change rates has left the automation floor replete with obsolete IT technologies that have often introduced security breaches and in general reduce the productivity and usability of the entire system.

Fog and mist computing has been identified as the most natural approach to leverage the benefits of functions virtualization while maintaining the performance constraints typical of OT systems. This, however, is one side of the coin as companies also like to leverage the advantage of the cloud, that is, large storage and massive data analytics to identify issues and bottlenecks in production and flesh them out.

The EXEGESIS platform provides the ideal deployment target for software defined automation as it can enable mist computing to address the deployment and management of virtualized OT functions and services over industrial hardware, and fog computing to address the consolidation of higher-level control and analytics on more computationally capable hardware deployed on the edge of the system.

Preliminary Evaluation

This section provides an initial investigation into how the EXEGESIS edge compute paradigm influences the amount of data flowing throughout a network. This is accomplished by simulating a simple scenario similar to the traffic camera use case described in the previous section. In order to perform the evaluation we use an open source fog environment simulator called iFogSim [9]. We tested three separate scenarios, all of them comprising a camera that collects information, a programmable-logic accelerated gateway device that connects the camera to the cloud, an actuator that receives commands after analysis of the camera data and performs the appropriate actions, and finally the cloud itself, as shown in Fig. 5:

- In the first scenario the camera input stream is forwarded through the gateway to the cloud, which performs the analysis and decision making and returns the decision to the actuator. This scenario is most akin to the current paradigm.
- The second scenario performs motion detection in the fog using the gateway device but sends the clip to the cloud for detailed analysis and decision making, representing a middle ground between a pure cloud and a pure edge approach.
- The third scenario implements all the processing, including motion detection, analysis, and decision making, at the edge on the gateway device and only sends a notification of actions taken to the cloud.

We evaluate two important parameters for all three scenarios. The first is normalized network usage (Fig. 6a), and the second is the energy consumption for the entire system (Fig. 6b).

It is plainly evident that the edge compute variant (scenario 3) is clearly superior in both metrics. Energy usage reduction is to be attributed to the advantages of using programmable logic to perform the computation at the edge but also at the constrained network traffic, which also factors into energy use. Network traffic is whittled down by performing all the processing close to the source and only sending a small action report to the cloud instead of an entire camera stream. These results underpin the claim that the EXEGESIS architecture can yield important potential benefits in multiple areas if realized at scale.

Conclusions

Future 5G networks are being viewed as the key technology that will allow for the realization of a "hyper-connected society" where billions of IoT devices will be able to exchange data and offer/receive services at a high quality of service level. Toward this, the fifth generation (5G) aims to support high data speed at the networks' edges (1–10 Gb/s) and achieve ultra-low end to end latency (~1 ms); however, these alone may not be enough, especially with highly heterogeneous and fragmented network environments, a vast number and huge variety of devices residing at the network edges, and the colossal amount of generated data that are slowly coming to the foreground. To overcome this, EXEGESIS exploits and advances the fog and mist paradigms to propose a beyond 5G ecosystem where heterogeneous fixed and mobile edge nodes (e.g., home gateways, small cells, smartphones, SME servers, IoT devices, vehicles) will form an archipelago of interconnected islands of resources (e.g., storage, computing, network) where each island can be viewed as the successor of a small cell and the archipelago as the evolution of the macrocell. A preliminary simulation-based investigation hinted at the significant benefits that can be derived from moving to the edge-centric EXEGESIS architecture. Future work will involve the implementation of a real-life prototype and the validation of the EXEGESIS paradigm in real-life scenarios.

A preliminary simulation-based investigation hinted at the significant benefits that can be derived from moving to the edge-centric EXEGESIS architecture. Future work will involve the implementation of a real-life prototype and the validation of the EXEGESIS paradigm in real-life scenarios.

References

[1] “Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are”; http://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf, retrieved July 2016.

[2] G. I. Klas, *Edge Cloud to Cloud Integration for IoT*, 2016.

[3] A. Poenaru, R. Istrate, and F. Pop “AFT: Adaptive and Fault Tolerant Peer-to-Peer Overlay — A User-Centric Solution for Data Sharing,” *Future Generation Computer Systems*, May 2016.

[4] M. Chiang, “Fog Networking: An Overview on Research Opportunities,” arXiv preprint arXiv:1601.00835 (2016).

[5] L. M. Vaquero and L. Rodero-Merino, “Finding Your Way in the Fog: Towards a Comprehensive Definition of Fog Computing,” *ACM SIGCOMM Computer Commun. Review*, vol. 44, no. 5, 2014, 27–32.

[6] A. Sfrent and F. Pop “Asymptotic Scheduling for Many Task Computing in Big Data Platforms,” *Info. Sciences J.*, vol. 319, Oct. 2015, pp. 71–91.

[7] J. F. Riera *et al.*, “TeNOR: Steps Towards an Orchestration Platform for Multi-PoP NFV Deployment,” *Proc. 2016 IEEE NetSoft Conf. Wksps.*, Seoul, Korea, 2016, pp. 243–50; doi: 10.1109/NETSOFT.2016.7502419

[8] Y. Nikoloudakis *et al.*, “A Fog-Based Emergency System for Smart Enhanced Living Environments,” *IEEE Cloud Computing Mag.*, Nov./Dec. 2016.

[9] H. Gupta *et al.*, “iFogSim: A Toolkit for Modelling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments,” *CoRR*, vol. abs/1606.02007, June 2016.

Biographies

Evangelos Markakis (markakis@pasiphae.eu) holds a Ph.D. from the University of the Aegean. Currently he acts as a senior research associate for TEI of Crete, and he is the Technical Manager for the HORIZON 2020 DRS-19-2014 “EMYNOS.” His research interests include fog networking, P2P applications, and NGNs. He has more than 30 refereed publications in the above areas. He is a Member of IEEE ComSoc and acts as Workshop Co-Chair for the IEEE SDN-NFV Conference.

Kimon Karras received his Ph.D. in embedded systems design from the Technical University of Munich and has spent four years at Xilinx Research Labs working on data center acceleration through FPGAs and innovative high-level synthesis applications. For the past year, he has been with Future Intelligence Ltd. where he is responsible for the development of the company programmable cloud platform.

Nikos Zotos, CIO, holds an M.Sc. in data communication systems. He has worked for various enterprises and research centers, holding active and key roles and positions. Currently he holds the position of chief innovation officer of Future Intelligence Ltd. His expertise includes design of large-scale heterogeneous networks, QoS over heterogeneous networks, IoT solutions for smart cities, next generation networks, energy efficiency networking, virtualized network environments, and cloud computing technologies.

Anargyros Sideris holds a Ph.D. from the University of the Aegean, Department of Information & Communication Systems Engineering. He joined Research & Development of the Telecommunications Systems Laboratory at the Technological Educational Institute of Crete. His current research activities and interests are in the fields of: network programming, digital interactive television, fog computing, and IoT

Harris Moysiadis is a business development manager at Future Intelligence. He graduated from Athens University of Economics and Business (B.Sc. in business administration) and received his M.Sc. in information systems: business IT from Manchester Business School, United Kingdom. His research interests focus on the business implications of ICTs, mapping their intervention in the business process cycle within the smart cities/agriculture/telecom context. He is a solution-oriented professional who analyzes as-is situations, creatively deconstructs reality, and re-constructs it with out-of-the-box stories.

Angelo Corsaro, Ph.D., is chief technology officer at ADLINK Technology. As CTO he looks after technology strategy and innovation for ADLINK’s Industrial Internet of Things (IIoT) platform. He is a well-known and cited expert in the area of high-performance and large-scale distributed systems with hundreds of publications in referred journals, conferences, workshops, and magazines.

George Alexiou received his Bachelor degree from the Applied Informatics and Multimedia Department of the Technological Educational Institute of Crete in 2014. He has worked as a full stack developer in the web hosting industry. Currently he is a research associate at PASIPHAE Laboratory working on various European funded projects. Additionally, he is doing his Master’s degree on informatics and multimedia in the Department of Informatics Engineering of the Technological Educational Institute of Crete.

Charalabos Skianis is an associate professor and head of ICSD, University of the Aegean, Greece. His work is widely published, and he acts on the TPC and OC for numerous conferences and workshops and as a Guest Editor for scientific journals. He is in the Editorial Boards of journals and a member of professional societies. He is an active member of several committees and organizations, and has participated in several R&D projects.

George Mastorakis received his B.Eng. in electronic engineering from the University of Manchester Institute of Science and Technology in 2000, his M.Sc. degree in telecommunications from University College London in 2001, and his Ph.D. degree in telecommunications from the University of the Aegean in 2008. He is currently serving as an associate professor at the Technological Educational Institute of Crete. He has published more than 150 research articles.

Constandinos X. Mavromoustakis is currently a professor in the Department of Computer Science at the University of Nicosia, Cyprus, where he is leading the Mobile Systems Lab (MOSys Lab., http://www.mosys.unic.ac.cy/) at the Department of Computer Science. He is an active member (Vice-Chair) of the IEEE/R8 Cyprus Section since January 2016, and since May 2009 he has served as the Chair of the C16 Computer Society Chapter of the Cyprus IEEE Section.

Evangelos Pallis holds an M.Sc. and a Ph.D. in telecommunications from the University of East London, United Kingdom. He currently serves as an associate professor at TEI of Crete in the Department of Informatics Engineering and director of PASIPHAE Lab. His research interests are in the fields of wireless and mobile networking. He has more than 200 refereed publications. He is member of IEE/IET, and a Distinguished Member of the Union of Regional Televisions in Greece.

Coordinate-Assisted Routing Approach to Bypass Routing Holes in Wireless Sensor Networks

Haojun Huang, Hao Yin, Geyong Min, Xu Zhang, Weixing Zhu, and Yulei Wu

The authors survey representative face-based geographic routing approaches, including their design prerequisites and philosophy. Then they outline the emerging issues to be addressed in the future and illustrate the forming factors behind them in detail.

Abstract

Geographic routing is becoming an attractive routing solution for WSNs since it offers a radical departure from traditional topology-dependent routing paradigms through use of geographic location in data delivery. However, it often suffers from the routing hole, referring to an area free of nodes in the direction closer to destination, in various real-world environments such as buildings and obstacles, leading to route failure. Currently, most geographic routing protocols tend to exploit face routing to recover the route. The basic idea behind it is to planarize the whole network by eliminating the crossing links before applying routing algorithms, thus achieving suboptimal network performance. In this article, we first survey representative face-based geographic routing approaches, including their design prerequisites and philosophy. Furthermore, we outline the emerging issues to be addressed in the future and illustrate the forming factors behind them in detail. Based on these observations, we then propose a CAGR to address the routing hole problem by employing relative coordinate systems, avoiding planarizing networks and preserving route optimality properties. Simulation results show that the proposed approach is superior to existing protocols in terms of packet delivery ratio, control overhead, and delivery delay in WSNs over a variety of communication sessions passing through the routing holes.

Introduction

Geographic routing, also referred to as localized or position-based routing [1–5], is becoming an attractive routing choice for use in wireless sensor networks (WSNs), since it offers a radical departure from traditional topology-dependent routing paradigms through use of location information in data delivery. Examples of its current and emerging applications include geographic information systems, location-aware services, and content-centric networking [2]. It is built on the location information of neighbors and destination, obtained through GPS or localization approaches [1, 2], to make routing decisions, thereby eliminating the need to establish and maintain route in the whole network. Commonly, it utilizes greedy mode to route data as far as possible and switches to bypass mode to recover the route once encountering a routing hole, which refers to an area free of nodes in the direction closer to the destination [4–8].

In order to address the routing hole problem, a number of bypass approaches, such as face routing [9], geometric routing [2], and flooding-based routing [6], have been proposed over the past few decades. These approaches have been considered as necessities for geographic routing to achieve desired routing goals in an efficient manner. However, the current geographic routing protocols built on these bypass approaches, especially the most prominent face-based geographic routing, which exploits face routing to recover the route from routing holes, still face some urgent issues, including suboptimal network performance and additional routing expenditure, to be addressed before these can be achieved [2].

In this article, we first summarize the existing representative face-based geographic routing approaches, including their design prerequisites and philosophy. Then we outline the emerging issues behind them and illustrate the forming factors in detail. Based on these observations, we then propose a novel coordinate-assisted geographic routing (CAGR) to address these issues related to routing hole detour in WSNs. Without planarizing the network graph, CAGR attempts to bypass the routing holes by employing relative coordinate systems (see Definition 3 for more details). To the best of our knowledge, this coordinate-assisted scheme is first proposed by us as an original innovation and offers meaningful insights for geographic routing design regarding how to recover the route from routing holes. Specially, CAGR avoids planarizing the whole network, thereby preserving route optimality properties. Furthermore, the calculation of CAGR is not significantly harder than existing face-based routing to recover a route, which makes our approach practical.

The remainder of this article is organized as follows. The following section provides a survey on representative face-based geographic routing approaches. The detailed design of CAGR is then described. The performance evaluation is presented next. The final section summarizes and concludes this article.

Digital Object Identifier: 10.1109/MCOM.2017.1600348

Haojun Huang is with Wuhan University; Hao Yin and Xu Zhang (corresponding author) are with Tsinghua University; Geyong Min and Yulei Wu are with the University of Exeter; Weixing Zhu is with PLA University of Science and Technology.

Face-Based Geographic Routing Overview

In this section, we focus on an overview of face-based geographic routing in detail. First, we illustrate its design prerequisites, followed by its design philosophy. Then we outline the urgent issues to be addressed in the future.

Design Prerequisites of Face-based Geographic Routing

Over the last few decades, a variety of face-based bypass approaches [1, 2, 4, 6, 9, 10] have been proposed to address or at least reduce the routing hole problem occurring in geographic routing, which is called face-based geographic routing. Commonly, it begins with greed delivery and exploits face routing to recover the route from routing holes. In addition, it is built on the following mechanisms to achieve the designed goals.

All Nodes Distribute in Two-Dimensional Networks: This means that the height of the network is no longer than the transmission radius of nodes, and the height difference of nodes can be neglected. This case is justified for most applications where all nodes are deployed on the Earth's surface [1, 2, 6, 9, 10].

Each Node Knows Its Own Virtual or Actual Location Information: This can be obtained through a GPS receiver or some other localization scheme.

Each Forwarder Is Aware of the Locations of Neighbors and Packet Destination: Each forwarder knows the locations of neighbors by exchanging beacon messages with them or resorting to receiving packets from neighbors. In order to gather the locations of neighbors, some protocols periodically send a beacon message by one-hop broadcast to all neighbors, whereas others aperiodically send a beaconless message to only nodes located in the relay region. To select the best forwarder, the nodes may require the location information of multihop neighbors [2]. The source node can know the location of the packet destination (by destination location services, receiving the location from a previous packet from that node, or some other mechanism), and then adds it into the packet header. Once receiving these packets, the forwarders in the network can learn its location from the packet header.

The Location of a Void Node Is Embedded into the Packet Header for the Current Forwarder to Check Whether Greedy Delivery Can Begin: In order to return to greedy mode, the location of a void node, referring to a node that exists no candidate closer in the direction to the destination in greedy mode [6, 10], is embedded into the packet header. Thus, each current forwarder receiving this packet can check whether it is closer to the destination than the void node in the network distance represented by Euclidean distance. If not, it continues data delivery as before. Specifically, such location information can be used to find an anchor node to escape routing holes in advance [9].

Design Philosophy of Face-Based Approaches

The basic idea behind these approaches, such as GPSR [9], GOAFR++ [6], and their variants [2, 10], is first to exploit face routing to planarize the network graph into numerous faces, by eliminating the crossing links using an algorithm like Relative Neighborhood Graph (RNG), Gabriel Graph (GG) [9], or their variants [3, 10], and then to apply a right-hand rule or left-hand rule [9] to deliver data along one or possibly a sequence of adjacent faces that provide an advance (see Definition 2 for more details) to the destination. The significant difference among them is how to planarize the network by exploiting face routing.

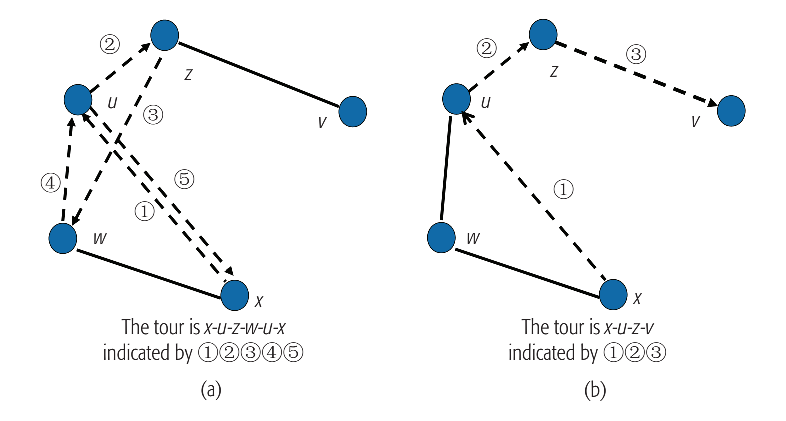


Figure 1. Face routing to bypass routing holes running in a random network raph: a) routing loop running in the full original network graph; b) successful routing running in the RNG network graph.

If a crossing edge exists in the network graph, face routing may encounter a routing loop, as shown in Fig. 1. In Fig. 1a, the route of node *x* that originates a packet to destination *v* is *x-u-z-w-u-x* applies the right-hand rule because this rule traverses the interior of a closed polygonal region (a face) in clockwise edge order. Clearly, there is a routing loop among nodes *x*, *u*, *z*, and *w*. In fact, such a case, elaborated in Fig. 1b, can be fully avoided if edge *zw* is removed by planarizing from the full network graph, and thus the data delivery between node *x* and destination *v* can correctly run along the route *x-u-z-v*. A planar graph obtained via face routing represents the same reachability as the original network with non-crossing links, but provides sparse connectivity due to planarization.

Emerging Issues in Face-Based Geographic Routing

The conventional face-based geographic routing approaches suffer from at least two urgent issues when exploiting face routing to recover the route from routing holes.

First, it incurs additional communication overhead, including the bandwidth used for its transmission and the memory required to store the information of neighbors. This is mainly because it requires the whole network to be planarized and to maintain a local planar graph at each node on one hand, and requires each node, including the node that does not participate in the routing, to broadcast beacon packets to its neighbors periodically for location information exchange on the other hand.

Second, it enables significant low density of connectivity in the network, and thus achieves suboptimal network performance. Taking Fig. 2 as an example, Fig. 2a is the full graph of a random network consisting of 100 nodes; both Figs. 2b and 2c are its planarized network graphs by GG

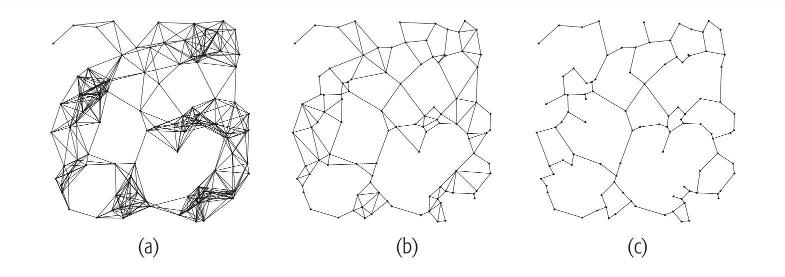


Figure 2. Network graph of a random network consisting of 100 nodes: a) the full original network graph; b) the planarized network graph by GG; c) the planarized network graph by RNG.

and RNG, respectively. Figure 2 reveals that there is an obvious increasing probability that multiple streams are sent to their destinations along the same sparse links or paths in the planarized subnetworks once they encounter the routing holes, which, in turn, leads to suboptimal network performance such as longer delivery delay and larger packet loss ratio. The most fundamental reason is that some optimized links, such as the crossing links and the strongly connected links, have been eliminated from the whole network. In particular, such planarized schemes are not good spanners of the original network graph. For instance, the nodes in the original networks that can be reachable along a path with a few hops might become very far away in the planarized networks. Such a phenomenon for the source-destination pairs passing through the routing holes, as shown in Figs. 2b and 2c, can significantly degrade the network performance, for example, inducing the lower data delivery ratio and longer delivery delay, even though the globally optimal routing protocols are conducted on these subnetworks.

These issues are becoming crucial problems for using face-based geographic routing in practice. However, in spite of some approaches such as BFP [2], Boundhole [7], CS [8], and ABC [13] proposed to address them, the proposed solutions cannot take them into account fully. For instance, all of them are not quite efficient and scalable to deal with such issues in recent WSNs characterized by node mobility, because they cannot tolerate the existence of mobile nodes or node failures.

CAGR: Coordinate-Assisted Geographic Routing

Notice that a variety of geographic routing approaches have been used in a number of areas, such as geographic information systems and location-aware services, while face-based geographic routing, as the most widely accepted of them, still suffers from the above-mentioned issues, which ought to be addressed before this can be achieved. Based on these observations, in this section, we propose CAGR to resolve these issues. Before diving into the details of our proposal, we first introduce some definitions as follows.

Definition 1 (Relay Region): The relay region *r*(*u*,v) for node *u* is defined as the area of the lens formed by the intersection of two circles centered at it and destination v, with radius equal to its transmission radius and the distance *d*(*u*,v) between it and node v, respectively.

Definition 2 (Advance): The advance $\overline{d(u,w)}$ that node *u* obtains by forwarding the packet to node *w* toward destination *v* is defined as the distance *d*(*u*, *v*) between node *u* and node *v* minus the distance *d*(*w*, *v*) between node *w* and node *v*, that is,

$$\overline{d(u,w)} \equiv d(u,v) - d(w,v)$$

where *d*(*u*, *v*) > *d*(*w*, *v*), meaning that each forwarder can achieve a positive advance in greedy mode.

Definition 3 (Relative Coordinate System): The relative coordinate system *RCS*(*u*, *v*) is a perpendicular location axes system that designates *x*-axis and *y*-axis built on the locations of node *u*, its neighbors, and destination *v*. See Fig. 3 for an illustration. The horizontal *x*-axis of *RCS*(*u*, *v*) is determined by the line from node *u* to destination *v*, and intersects the perpendicular *y*-axis at node *u*. Given any neighbor *i* of node *u*, its location received from a GPS device or a separate calibration process act as its coordinate in *RCS*(*u*, *v*), following the east-west and north-south displacements from node *u*. It belongs to one region, that is, one of regions I, II, III, and IV of *RCS*(*u*, *v*), determined by its location.

There are two alternative paths passing through two sides of the routing holes for CAGR to route data to the destination, depending on its traversal direction. In order to facilitate understanding, we mainly illustrate how CAGR bypasses the routing holes only in counterclockwise (clockwise) order, as in [2–7]. This means that CAGR cannot select a candidate from region IV of a coordinate system. In fact, the nodes locating in region IV of a coordinate system belong to regions I and II of their upstream coordinate system and will result in a routing loop if selected as the forwarder, as elaborated in Fig. 3. Therefore, CAGR only chooses the neighbors located in regions I, II, and III of the coordinate systems as the forwarders.

CAGR Overview

The basic idea of CAGR is to employ relative coordinate systems to recover the route from routing holes, thus avoiding planarizing networks and preserving route optimality properties. Figure 3 illustrates the architecture of CAGR, which works in two modes: greedy mode and bypass mode. In the former mode, the current forwarder broadcasts a request-to-send (RTS) message to detect its best next-hop relay that has the maximum advance to the destination. Once receiving the RTS message, only the neighbor in the relay region sets a delay for broadcasting a corresponding clear-to-send (CTS) message based on a discrete delay function. The neighbor that has the minimum delay broadcasts its CTS message first, and the other candidates snooping the CTS message notice that another node has already responded to the request and thus quit the contention process. If no CTS is returned from the relay region, the current forwarder assumes that a routing hole is encountered, then enters into the bypass mode to recover the route. In this case, it divides the networks into four regions by employing a relative coordinate system, and then broadcasts an RTS message to announce its four regions. All the neighbors in regions I, II, and III participate in the next-hop relay selection process using an angle-based delay contention mechanism, such that the first candidate in counterclockwise order responds first. Among the divided regions, the current forwarder finally selects the candidate that provides the minimum angle between forwarder-neighbor and forwarder-destination as the next-hop relay, and then unicasts the data to it. This process continues until either the greedy mode restarts or the destination is reached.

Greedy Delivery

Given any node *u* that intends to deliver data to destination v, it first broadcasts an RTS message, which contains its location and destination v, to detect its best candidate, denoted by *f*{*u*}. Once receiving the RTS message from node *u*, each neighbor *w* sets its contention time

$$\varsigma_{w\to u} \text{ to } \left(1-\frac{\overline{d(u,w)}}{r}\right)\times t_{\max}$$

for broadcasting the CTS message, which contains its own location. Here, the maximum waiting timer $t_{\max}$ is chosen long enough, determined by application requirements, to ensure that node *u* can receive the CTS message from the neighbors in *r*(*u*, *v*), and *r* denotes its maximum transmission range. By Definition 2,

$$\overline{d(u,w)}\le r$$

means that the waiting timer of each node does not exceed $t_{\max}$. Each candidate with different advance responds to the reply at different time instants, thus avoiding unwanted collisions among them. Obviously, $\zeta_{i\to u}$ can ensure that the node, denoted by *i*, with the maximum advance to the destination first broadcasts the reply message. If overhearing a CTS message broadcasted by another candidate before $\zeta_{w\to u}$ is due, node *w* discards its corresponding CTS message; otherwise, it broadcasts its CTS message only when $\zeta_{w\to u}$ is due. Once receiving the CTS message from neighbor *w*, node *u* updates its best candidate if

$$\overline{d(u,f_u)}<\overline{d(u,w)}$$

or f_u is null. Finally, it sends the packet to node *w* by unicast.

Bypass Delivery

Upon broadcasting an RTS message, node *u* sets its waiting timer to $t_{\max}$ and starts the timer. If no CTS message returns from *r*(*u*, *v*) until the timer is expired, it considers that a routing hole is encountered. In this case, such a node serves as a void node since there is no candidate closer in the direction of the destination in greedy mode.

To recover the route from routing holes, node *u* first sets the data packet as the bypass mode and inserts its location into the packet header for any subsequent forwarder in this mode to decide whether to enter into the greedy mode to deliver data again. Then it calculates its local relative coordinate system *RCS*(*u*, *v*) to divide the networks into four regions. After that, it broadcasts an RTS message, which contains the locations of it and its destination *v*, *RCS*(*u*, *v*) and the bypass mode information, to its neighbors, and sets its waiting timer to $t_{\max}$.

For any node $i \in RCS(u, v)$, we call ∠*iuv* shown in Fig. 3 its deflection angle. Let α = ∠*iuv*; we have

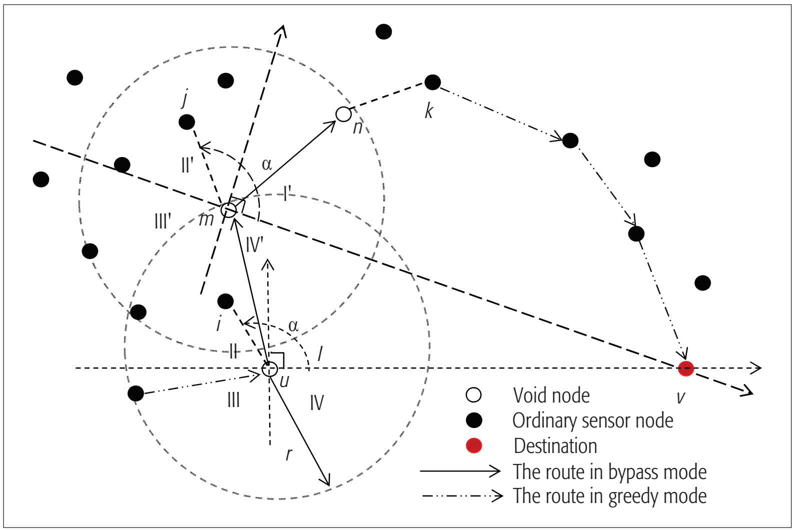


Figure 3. Bypassing routing holes by employing a relative coordinate system.

$$\cos\alpha=\beta=\frac{d(u,i)^2+d(u,v)^2-d(i,v)^2}{2d(u,i)\times d(u,v)}. \tag{1}$$

Then

$$\alpha=\begin{cases}\arccos\beta, & \text{if node } i \text{ is in Regions I or II;}\\ \pi+\arccos\beta, & \text{if node } i \text{ in Region III.}\end{cases} \tag{2}$$

Once receiving the RTS message, node *i* first checks whether it lies in *RCS*(*u*, *v*). If $i \notin RCS(u, v)$, the RTS message is then directly discarded. Otherwise, node *i* uses Eq. 2 and its location to calculate α, and then sets its waiting timer to $\xi_{i\to u}$ for broadcasting the CTS message, which contains its own location, and α. $\xi_{i\to u}$ is given by

$$\xi_{i\to u}=\frac{\alpha}{2\pi}\times t_{\max}, \tag{3}$$

where angle α can be considered in counterclockwise or clockwise order, depending on the traversal direction (right-hand rule or left-hand rule). Before $\xi_{i\to u}$ expires, if node *i* overhears a CTS message broadcasted by another candidate, it deletes its CTS message. Otherwise, it answers node *u* with a CTS message when $\xi_{i\to u}$ expires. When $t_{\max}$ is due, node *u* selects node *m*, which has the minimum angle toward destination *v* shown in Fig. 3, as the forwarder to relay its packet. If no CTS message responds, meaning that no neighbor exists, node *u* then sends the packet to its upstream forwarder to search for another candidate. In this case, the upstream forwarder then regards the current forwarder as an incompetent candidate toward destination *v* to guide the subsequent data delivery.

Once receiving the packet from node *u*, node *m* first checks if the bypass mode can be continued. If *d*(*m*, *v*) > *d*(*u*, *v*), that is, the packet should be routed in bypass mode, it then employs *RCS*(*m*, *v*) to select neighbor *n* as the forwarder to relay the packet. To avoid route loop, node *m* adds its location into the packet header such that it can check whether it has received the same message again in the subsequent data delivery. If this is the case or no CTS message is returned from neighbors, it returns the packet to its upstream forwarder to find one new candidate. This process continues over multiple hops until either the packet arrives at a

While in our approach and BFP, more and more nodes become candidates used for data delivery as the communication sessions increase. Since the candidates just fall in relay region and only need to exchange their location information with the current forwarder. Hence, CAGR and BFP have a similarly lower control overhead than CS with the communication sessions increasing.

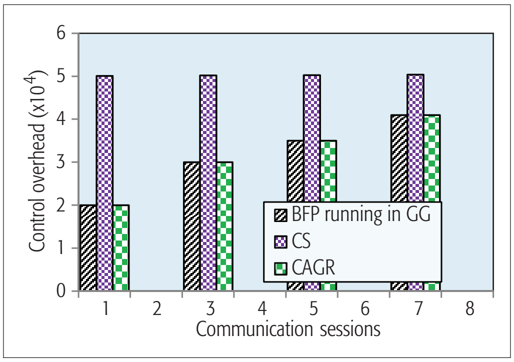


Figure 4. Control overhead with different communication sessions.

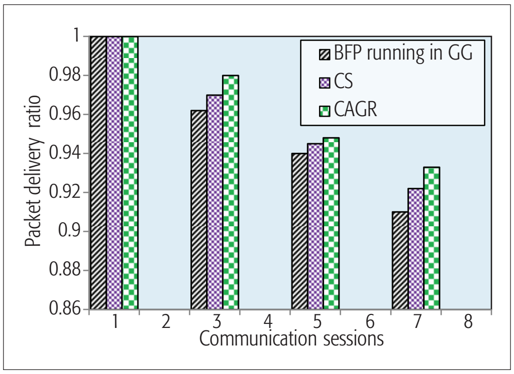


Figure 5. Packet delivery ratio with different communication sessions.

node *k* closer to destination *v* than the void node or destination *v* is reached. Then node *k* removes the locations of all the forwarders in bypass mode from the packet header, and returns to greedy mode to forward it unceasingly if required.

Performance Evaluation

Simulation Environments

The popular network simulation platform, NS2.35, has been revised, built on the real-world benchmark, for the performance evaluation of CAGR, as in [7, 8, 10, 13, 14]. The recent non-face-based geographic routing CS [5] and face-based geographic routing BFP [10] running in GG graph, are implemented and compared to our approach. The underlying medium access control (MAC) protocol is IEEE 802.11, which can adopt RTS/CTS mechanism to exchange locations among neighbors. The network size is set to 500m¥1000m, where 500 nodes are randomly distributed. The node transmission range is set to 40m. Therefore, the average degree of each node is 6.4 nodes, and the network density is 0.001node/m^2.

In order to represent the real application scenarios as far as possible, all nodes are allowed to move randomly but not at the same time, with the speed of 0 $\sim$ 1m/s. The mobility interval is set to 10s. To deal with the unreachable-destination issue due to mobility, the location propagation scheme [15] is employed, by making use of the overhearing feature of wireless transmission to deliver the location information of the mobile destination to source node. Each source node generates CBR flows at 2 kb/s with the packet size of 32 bit. The beacon interval of CS is set to 5s. One routing hole with the size of 80m is set in the center of the network, which makes the bypass mode more likely happen. Each communication session passing through the routing holes is randomly selected in the network. The communication sessions without bypass mode are not considered as our efficient results. The number of communication sessions varies from 1 to 8. Each simulation run lasts for 500s, and every result on the curve is the average of 40 simulation runs. Three key performance metrics, that is, packet delivery ratio, delivery delay and control overhead, are conducted to evaluate the performance of our scheme.

Simulation Results

Figure 4 indicates the control overhead with different communication sessions. In order to know the locations of neighbors, CS requires each node to broadcast its location to neighbors by beacon messages periodically. Such a periodic location update approach has no relation with the number of communication sessions. Therefore, there is almost no change in control overhead for it with different numbers of communication sessions. In our approach and BFP, more and more nodes become candidates used for data delivery as the communication sessions increase since the candidates just fall in the relay region and only need to exchange their location information with the current forwarder. Hence, CAGR and BFP have similarly lower control overhead than CS with the communication sessions increasing.

Figure 5 demonstrates the packet delivery ratio with different communication sessions. As the number of communication sessions increases, many of them may simultaneously bypass a routing hole, resulting in low packet delivery ratio for CS, BFP, and our approach. This is mainly due to the data collisions occurring in the data process on the boundary of the routing holes. In all three geographic routing approaches, the data packets are transmitted to their destinations along the boundary of the routing holes. Because CS and our approach have not planarized the whole network, the data may be routed to multiple destinations along more different paths compared to BFP, thus decreasing the probability of data collisions. Notice that our approach is tolerant of mobility of nodes; therefore, it is clear to see that the packet delivery ratio of CAGR decreases less compared to CS when communication sessions increase.

Figure 6 manifests the delivery delay with different communication sessions. The results indicate that CAGR achieves lower delivery delay compared to CS and BFP. There are two main reasons for the performance gap in our scheme with both of them. First, our scheme does not planarize the whole network, compared to BFP, making the data route to the destination along a closer path with fewer hops as far as possible. Second, in our scheme, multiple sessions are much easier to send to their destinations when there are mobile forwarders in the network, which decreases the delivery delay for each session.

Conclusion

Notice that since face-based geographic routing, as the commonly accepted routing in WSNs, has attracted attention over the past few decades, we have surveyed representative face-based geographic routing approaches, including their

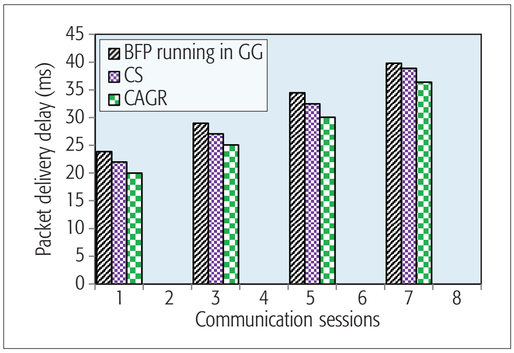


Figure 6. Delivery delay with different communication sessions.

design prerequisites and philosophy, and then outlined their emerging issues addressed urgently in the future. Based on these observations, we have then proposed CAGR, a novel efficient geographic routing approach to address the routing hole problem for communications in WSNs. Without network planarizartion, the packets are routed to destinations along the boundary of the routing holes by employing network coordination, thereby preserving route optimality properties. Simulation results have demonstrated that CAGR significantly outperforms other schemes in terms of control overhead, packet delivery ratio, and delivery delay over a variety of communication sessions passing through the routing holes.

Acknowledgment

This work has been partially supported by the National Natural Science Foundation of China (No. 61402343, No. 61672318, No. U1504614, No. 61631013, and No. 61371125), the National Key Research and Development Program (No. 2016YFB1000102), the Natural Science Foundation of Suzhou/Jiangsu Province (No. BK20160385), and the projects of Tsinghua National Laboratory for Information Science and Technology (TNList).

References

[1] D. Torrieri *et al.*, "Performance Comparisons of Geographic Routing Protocols in Mobile Ad Hoc Networks," *IEEE Trans. Commun.*, vol. 63, no. 11, 2015, pp. 4276–86.

[2] C. Fraser *et al.*, "A Survey of Geographical Routing in Wireless Ad-Hoc Networks," *IEEE Commun. Surveys & Tutorials*, vol. 15, no. 2, 2013, pp. 621–53.

[3] S. S. Lan *et al.*, "Geographic Routing in d-Dimensional Spaces with Guaranteed Delivery and Low Stretch," *IEEE/ACM Trans. Networking*, vol. 21, no. 2, 2013, pp. 663–77.

[4] Q. Fang *et al.*, "Locating and Bypassing Routing Holes in Sensor Networks," *IEEE INFOCOM '04*, Mar. 2004, pp. 7–11.

[5] A. Mostefaoui *et al.*, "Localized Routing Approach to Bypass Holes in Wireless Sensor Networks," *IEEE Trans. Comput.*, no. 63, no. 12, Dec. 2014, pp. 3053–65.

[6] D. Chen *et al.*, "A Survey of Void Handling Techniques for Geographic Routing in Wireless Networks," *IEEE Commun. Surveys & Tutorials*, vol. 9, no.1, 2007, pp. 50–67.

[7] W. Liu *et al.*, "Greedy Routing with Anti-Void Traversal for Wireless Sensor Networks," *IEEE Trans. Mobile Comp.*, vol. 8, no. 7, 2009, pp. 910–22.

[8] Y. Kim *et al.*, "Geographic Routing Made Practical," *USENIX NSDI '05*, vol. 2, May 2005, pp. 217–30.

[9] B. Karp *et al.*, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *ACM MobiCom '00*, Aug. 2000, pp. 243–54.

[10] S. Ruhrup *et al.*, "Message-Efficient Beaconless Georouting with Guaranteed Delivery in Wireless Sensor, Ad Hoc, and Actuator Networks," *IEEE/ACM Trans. Networking*, vol. 18, no. 1, 2010, pp. 95–108.

[11] J. A. Sanchez *et al.*, "Beacon-Less Geographic Routing Made Practical: Challenges, Design Guidelines, and Protocols," *IEEE Commun. Mag.*, vol. 47, no. 8, Aug. 2009, pp. 85–91.

[12] T. Lee *et al.*, "ABC: A Simple Geographic Forwarding Scheme Capable of Bypassing Routing Holes in Sensor Networks," *Elsevier Ad Hoc Net.*, vol. 8, no. 4, June 2010, pp. 361–77.

[13] C. Petrioli *et al.*, "ALBA-R: Load-Balancing Geographic Routing Around Connectivity Holes in Wireless Sensor Networks," *IEEE Trans. Parallel Distrib. Sys.*, vol. 25, no. 3, Mar. 2014, pp. 529–39.

[14] Q. Chen *et al.*, "Adaptive Position Update for Geographic Routing in Mobile Ad-hoc Networks," *IEEE Trans. Mobile Comp.*, vol. 12, no. 3, 2013, pp. 489–501.

[15] F. Yu *et al.*, "A Simple Location Propagation Scheme for Mobile Sink in Wireless Sensor Networks," *IEEE Commun. Lett.*, vol. 14, no. 4, 2010, pp. 321–23.

Biographies

Haojun Huang is with Wuhan University. He received his Ph.D. degree from the School of Communication and Information Engineering of the University of Electronic Science and Technology of China in 2012. He worked as a postdoctoral researcher in the Research Institute of Information Technology at Tsinghua University from 2012 to 2015. His current research interests include wireless communication, ad hoc networks, big data, and software-defined networking.

Hao Yin is a professor in the Research Institute of Information Technology at Tsinghua University. He won the Chinese National Science Foundation Award for Excellent Young Scholars in 2012. His research interests span broad aspects of multimedia communication and computer networks. He is also the vice-director of the Industry Innovation Center for Future Networks, China, and the Secretary-General of the Industry Innovation Alliance of Future Internet, China.

Geyong Min is a professor of high performance computing and networking in the Department of Computer Science at the University of Exeter, United Kingdom. He received his Ph.D. degree in computing science from the University of Glasgow, United Kingdom, in 2003. His research interests include future Internet, computer networks, wireless communications, multimedia systems, information security, high performance computing, ubiquitous computing, modeling, and performance engineering.

Xu Zhang received his B.S. degree in communication engineering from Beijing University of Posts and Telecommunications in 2012 and his Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University in 2017. He was a visiting student in the Department of Electrical & Computer Engineering at the University of Florida from November 2015 to May 2016. His research interests include content delivery networks, network measurement, and multimedia communications.

Weixing Zhu is an associate professor in the Information Management Center at PLA University of Science and Technology. He received his Ph.D.degree in military communications from PLA University of Science and Technology in 2012. His main research focuses on network architecture, cloud computing, big data, military requirements, and software engineering.

Yulei Wu is a lecturer in computer science at the University of Exeter. He received his Ph.D. degree in computing and mathematics and B.S degree in computer science from the University of Bradford, United Kingdom, in 2010 and 2006, respectively. His main research focuses on future Internet architecture, wireless networks and mobile computing, cloud computing, big data for networking, and performance modeling and analysis..

Simulation results have demonstrated that CAGR significantly outperforms other schemes in terms of control overhead, packet delivery ratio and delivery delay over a variety of communication sessions passing through the routing holes.

Crowd Associated Network: Exploiting over a Smart Garbage Management System

Saiful Azad, Arafatur Rahman, A. Taufiq Asyhari, and Al-Sakib Khan Pathan

Most existing non-real-time applications utilize infrastructure based or semi-infrastructure based network architectures. Such a network architecture demands a considerably high installment and maintenance cost. To alleviate the cost, the authors propose an efficient infrastructure-less network architecture named CrAN. In CrAN, a set of crowds play significant roles by completing the communication gaps among various associates in the network; hence the name.

Abstract

Most existing non-real-time applications utilize infrastructure-based or semi-infrastructure-based network architectures. Such a network architecture demands a considerably high installment and maintenance cost. To alleviate the cost, in this article, we propose an efficient infrastructure-less network architecture named CrAN. In CrAN, a set of crowds play significant roles by completing the communication gaps among various associates in the network; hence the name. We show the usability of this proposed architecture to support non-real-time data transmission over an SGMS, where optimum solutions need to be discovered to minimize the management cost. Due to the complexity of the optimization problem, we approximate these optimum solutions using a GA. In the implementation of the GA, we apply new fitness functions to discover a feasible trade-off between distance and waste volume. We then compare the performance of the proposed fitness functions with that of an existing fitness function. The results favorably suggest the necessity of employing the proposed fitness functions to obtain near-optimum solutions.

Introduction

In general, all computer applications can be broadly classified into *real-time* and *non-real-time* applications. In real-time applications, responses to certain events are constrained within a fixed time interval, that is, timeliness is a primary measure of performance. On the other hand, although every *non-real-time* application has its own performance indicator, its required response time is subjective. Consequently, unlike their counterparts, the demand of fixed network architectures by *non-real-time* applications is not imperative. However, *á la* real-time applications, most of the existing non-real-time applications utilize *infrastructure-based* or *semi-infrastructure-based* network architectures. Hence, they charge high expenditures for installation and maintenance. To alleviate the cost of such applications, an inexpensive but efficient *infrastructure-less* network architecture is considered in this article. The usability of the proposed network architecture is specifically demonstrated for an important non-real-time application, that is, a *smart garbage management system* (SGMS).

A number of frameworks have been proposed to satisfactorily manage the garbage problems within the vision of smart cities. For instance, in [1], the authors designed an intelligent solid waste bin to aid the existing waste management system. Their work focused only on the bin design, but designing an appropriate network architecture and optimizing the cost for garbage collection remain out of their scope. In [2], an Internet-of-Things (IoT)-based SGMS is proposed to reduce the amount of food waste by imposing certain constraints. In that proposed SGMS, smart garbage bins (SGBs) communicate among themselves using a wireless mesh network, and transmit data to a router, which then forwards them to a server. All the acquired data are analyzed by the router, which decides service provisioning. Except for SGBs, all other devices are connected via the Internet. Moreover, a direct path between an SGB and the router is assumed. Due to that assumption, the distances between the SGBs are kept notably short, which is again impractical. Other solutions for SGMS have also been addressed in [3–5]. However, most of these solutions utilize infrastructure based or semi-infrastructure-based network architectures. Consequently, the installment and maintenance costs of these networks are considerably high.

On the other hand, our proposed infrastructure-less network architecture aims to support the entire operation of the SGMS, and charges a minimum cost to deploy and maintain. The proposed architecture is compatible for any non-real-time application, where frequent data acquisition is not necessary for proper functioning of that application. In the proposed network, a set of crowds is utilized with other network components to acquire data from a considerably large area (e.g., a town or a city). The details of the proposed network architecture are discussed in the following section with a compatible application after that. Data acquired from the network nodes (SGBs) are further processed to assist in discovering optimum solutions (in terms of reducing management cost) for the SGMS [6, 7]. In this article, we employ a genetic algorithm (GA) to discover such feasible solutions from the acquired data [8, 9]. Within the GA, two new fitness functions are applied and compared to a trivial fitness function. The results hint at the necessity of employing new fitness functions to find feasible solutions.

Digital Object Identifier:
10.1109/MCOM.2017.1600379

Saiful Azad and Arafatur Rahman are with University Malaysia Pahang and IBM Center of Excellence, UMP; A. Taufiq Asyhari is with Cranfield University; Al-Sakib Khan Pathan is with Southeast University, Bangladesh.

Crowd Associated Network

The key concept of the proposed infrastructure-less network architecture is the utilization of the crowd to complete the communication gaps among the associates. Hence, it becomes an inseparable part of the network, and is named a *crowd associated network* (CrAN). In CrAN, two types of components are involved: dedicated agents and non-dedicated agents.

The dedicated agents are those agents that are solely installed in the network to perform some specific tasks. In general, these agents are static and exchange information with non-dedicated agents to achieve the networking goal. On the other hand, the crowd is the latter type of agent who is equipped with necessary devices and acts like an intermediate relay in the proposed network architecture. The crowd completes the communication gaps among the dedicated agents and thereby enables them to function properly. It acquires data from one or multiple dedicated agent(s) and delivers multiple copies to other dedicated agent(s). The members of the crowd may also exchange data among themselves in the hope that the cooperating members will deliver the data to one or multiple dedicated agent(s). This technique improves the performance of the network in terms of data delivery and end-to-end delay. A notable point is that everyone in the crowd is qualified to be a part of the network if he/she complies with the network requirements. However, in reality, not everyone would be interested in contributing. Therefore, from now on, for the sake of distinguishing the non-contributors from the contributors, we will refer to the latter as *volunteers*. They provide services without any expectation of compensation and without any coercion. A volunteer will be given a network component, which he/she has to install in his/her own vehicles (e.g., motorcycle, car, bus). This network component will be called a *volunteer agent* (VA) throughout the rest of the article.

As mentioned earlier, this sort of network architecture is suitable for applications with non-real-time data, such as SGMS, where acquisition can be satisfactorily fulfilled by one or a few successful transmission activities per day.

There are manifold advantages of using this architecture, such as: no required fixed infrastructure (i.e., infrastructure-less), no fixed boundary in terms of deployment, a smaller number of dedicated agents, and lower expected deployment and maintenance costs than any infrastructure-based or semi-infrastructure-based network. Volunteers are the key actors in the proposed CrAN, and their recruitment can be facilitated, for example, through the following provisions.

- The local community may provide incentive to volunteers through revising or reducing tax and/or other service charges.
- A social awareness campaign may also play a significant role in convincing people to become volunteers.
- Other sources that can be explored are employees of a municipal corporation, government offices, or social organizations who live around the coverage area.
- Public buses that travel around cities can contribute to this task; even garbage collecting containers can be equipped with VAs to acquire information.

CrAN for SGMS

In the following subsection we briefly introduce the components that are utilized to install the CrAN. The proposed infrastructure-less network architecture is then detailed. Next, we discuss the communication protocols that are suitable for the proposed network architecture. Then we present the techniques related to data processing and discovering optimum solutions.

Network Components

The CrAN consists of five distinct components: SGB), volunteer agent (VA), *sink, control center* (CC), and *garbage collecting agent* (GCA). All of them have their unique identification numbers. Among them, all except VA are dedicated agents. A collaborative effort of these components envisions delivery of necessary data and discovery of optimum or near-optimum solutions for the SGMS, which contribute significantly in reducing waste management cost. The details of the components are briefly discussed below.

Smart Garbage Bin: Unlike other conventional garbage bins, an SGB is embedded with a sensor that can measure the volume of garbage. The SGB periodically acquires this information and transfers it along with other necessary information to the encountering associates. The SGBs are battery powered, and have low computational abilities and storage capacities. Hence, the following two initiatives are undertaken to enhance the lifetime of an SGB: it can only transfer data whenever necessary without any relaying capability, and instead of continuously delivering packets to all associates (within the range), a priority-based technique is employed to reduce energy dissipa tion.

Volunteer Agent: The objective of this component is to acquire data from the SGBs and exchange them with compatible associates when encountered. This is the only non-dedicated component in the system, and hence its behavior is unpredictable. Therefore, it is prescribed to assign multiple VAs in an area with the idea that at least one of them is able to deliver the data to the appropriate associate(s). In order to obtain a reasonable performance, a VA needs to be supplied with an affluent energy source. In our case, each VA is attached to a vehicle and draws energy from the battery of the vehicle.

Sink: The objective of this component is to exchange information with the VAs when both of them are within communication range. Sinks are dedicated agents that connect to affluent energy sources through electrical wiring. The *destination sink* is a special type of sink with direct connection to the CC. Unlike other sinks, it only forwards data to the CC and never re-transmits any copies to other associates.

Control Center: The primary objective of the CC is to acquire data from the destination sink and subsequently utilize them to obtain optimum solutions for garbage collecting agents with respect to one or multiple parameters (distance, number of containers, etc.). All the computed solutions are stored in a buffer and delivered on a demand basis.

Garbage Collecting Agent: This component is involved in unloading the SGBs by following the optimum solutions provided by the CC. The GCA

There are manifold advantages of using this architecture, such as: no required fixed infrastructure, no fixed boundary in terms of deployment, a smaller number of dedicated agents, and a lower expected deployment and maintenance cost than any infrastructure based or semi-infrastructure based network.

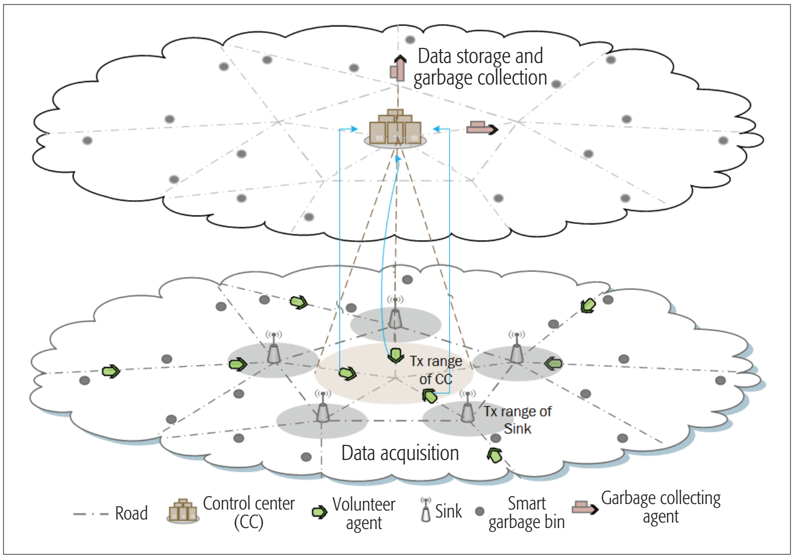


Figure 1. The two-tier architecture of the *CrAN*.

can also be utilized to replace batteries when necessary. When a GCA completes unloading all assigned SGBs, it moves to the dumping zone for releasing the garbage and then returns back to the depot.

Network Architecture

The network architecture of the CrAN is depicted in Fig. 1. It is a two-tier architecture where the first tier is mostly involved in data acquisition, and the second tier is involved in data processing and discovering optimum solutions for the GCAs.

Generally, the SGBs are placed beside roads to ease the garbage collection process. In other words, the SGBs are spatially distributed components installed inside the network area. They periodically acquire waste volume status and generate waste DATA packets and other relevant information. The VAs are the mobile agents in the network that provide voluntary services and gather necessary DATA packets from the SGBs when they encounter the latter. Thereafter, the VAs exchange these packets opportunistically when they come into contact with nearby sinks or other VAs in the hope that these relaying nodes will deliver the packets to the destination (more specifically, to the destination sink). The sinks also apply a similar opportunistic forwarding technique to route the packets to the destination. Consequently, these components have a routing capability to decide which packets to transmit, how many duplicate copies to spread, and so on. Finally, the destination sink receives packets from various sources and delivers them to the CC for further processing.

As noted before, the second tier in the network architecture is involved in data processing, discovering optimum solutions for the GCAs and acquiring feedback from the GCAs. After receiving packets from the destination sink, the CC processes the required data and stores them in a buffer after performing a simple freshness treatment, that is, old data are overwritten with the fresh data. Then, periodically, it computes the optimum solutions with respect to one or multiple parameters as mentioned before. At a later time, these solutions are delivered to the GCAs in order to collect the garbage in the most efficient manner. A GCA unloads an SGB, updates the system if required, and also changes energy source whenever necessary. It may also acquire data from other SGBs, which are not unloaded, but encountered during the trip. At the end, when it returns to the depot, it delivers the feedback and the acquired packets to the CC.

Data Transmission

In CrAN, only the SGBs can generate DATA packets, and all other associates act as intermediate relays to deliver them to the destination sink. In a DATA packet, an SGB encloses the waste volume status as well as the status of the energy source. The prior one indicates when to unload it, and the latter indicates when to change its energy source. A single copy of a DATA packet may result in failure to reach the destination since the nodes experience intermittent connectivity due to a large network area. Therefore, duplication of packets may result in a high probability of reliable delivery to the destination sink within a given timeframe. Hence, an SGB injects L copies of a DATA packet in the network through various associates (mainly *VA*s).

As mentioned earlier, the network components in the CrAN have heterogeneous capabilities in terms of data transmission. An SGB and the destination sink do not require routing capabilities. A simple MAC protocol can enable these components to transmit or receive DATA packets from other associates. In contrast, the rest have to relay packets as they are intermediate nodes, and hence they need routing capabilities. In the following two subsections, we discuss some direction in selecting the most relevant routing and MAC protocols for the CrAN architecture.

Routing: Since the CrAN is an infrastructure-less network, and the operation area can be considerably large, there is a small possibility that a complete end-to-end route can be discovered for delivering a packet to the CC. Thereby, all the components in the network may experience intermittent connectivity or lack of connectivity and time-varying hop-to-hop propagation delays. Hence, the routing protocols that assume direct end-to-end routes before data transmission are not applicable in the CrAN. Conversely, there are opportunistic routing protocols [10–12], which store the packets until an opportunity arises to forward them to another node(s) in the hope that the receiving node is the destination or will at least forward the packets to the destination directly or via other intermediate nodes. These protocols are known as *store-and-forward*-based routing protocols.

Most store-and-forward-based routing protocols can be broadly classified into *replication-based* and *forwarding-based* routing protocols. As the name suggests for the preceding class of protocols, they replicate the packets whenever necessary. A generalized practice, which is observed among these protocols, is that they allow a considerable amount of replication to increase the delivery probability of a DATA packet. In contrast, forwarding-based routing protocols forward a packet until it reaches the destination without any duplication. Although this approach achieves higher efficiency in terms of resource preservation and overhead reduction, it experiences a lower packet delivery ratio and higher end-to-end delay. Hence, they are not preferable for adoption in the CrAN.

There are various replication-based routing protocols proposed in the literature. In [9], an epidemic routing protocol is proposed, which replicates every packet when it encounters a new contact. Hence, its packet delivery ratio is considerably higher than other similar protocols. However, since it is very similar to the *flooding* technique, the network experiences a considerably high overhead. Consequently, the epidemic routing protocol is also not preferable in this network. On the other hand, there are protocols that limit the replication overhead through specific techniques (e.g., [10, 11]). These protocols are considerably easy to implement and demand relatively lower computing power. Therefore, these protocols are good candidates for the proposed network architecture.

Medium access control (MAC): Unlike routing protocols, a MAC protocol is obligatory for all the network components in the CrAN. Among the existing *MAC* protocols, *handshake-based* MAC protocols, such as IEEE 802.11 [12] and IEEE 802.15.4 [13], are suitable for those networks where channel contentions are frequent phenomena and packet drop probability is high due to collisions. In handshake-based MAC protocols, a node has to reserve a channel before initiating any transmission attempt through the handshaking procedure. Conversely, a network architecture like CrAN, where contention and collision are seldom phenomena, these protocols are not applicable due to a considerable amount of overhead they impose before any data transmission. On the other hand, most of the contention-based protocols, such as ALOHA and carrier sense multiple access (CSMA), transmit a packet with an assumption that the next node is within its vicinity. Hence, this type of protocol is also not suitable for the CrAN. For this network architecture, only those MAC protocols that trigger packets when the nodes come within a communication range are preferred. A node must store the packets and transmit them opportunistically. Such a mechanism is embedded within *store-and-delivery-based* MAC (SD-MAC*)* protocol as proposed in [14]. It is a lightweight MAC protocol that is suitable for most sensor nodes.

Unlike routing protocols, a MAC protocol is obligatory for all the network components in the CrAN. Among the existing MAC protocols, handshake-based MAC protocols, such as IEEE802.11 [12] and IEEE802.15.4, are suitable for those networks where channel contentions are frequent phenomena and packet drop probability is high due to collisions.

Data Processing and Discovering Optimum Solutions

After acquiring necessary `DATA` packets through the CrAN architecture, the CC extracts all the required data and then, at a later time, processes these data to find optimum solutions. An SGMS is incomplete if the acquired data are not processed, and optimum results (in terms of minimizing the management cost of the system) are not calculated.

For simplicity, in our forthcoming discussion let us assume that the CC has adequate recent data of the network. It then has to compute feasible solutions and deliver them to the GCAs on a demand basis. In terms of cost optimization, let us assume that we aim to minimize the requirements of the GCAs. Note that if a single GCA can unload all the bins, this problem can be cast into a simple and well studied *traveling salesman problem* (TSP). However, in reality, this latter assumption is less realistic since all the GCAs have limits in terms of capacity. It is therefore necessary to consider this constraint, and we shall refer to the more realistic context as a *garbage collection problem* (GCP).

The GCP resembles the known *capacitated vehicle routing problem* (CVRP). In CVRP, a fixed fleet of delivery vehicles with identical capacity must be utilized to provide service to known customer demands for a single commodity and from a single depot at minimum cost. The objectives of the CVRP include minimizing the vehicle fleet and minimizing the travel time while keeping the total demand of commodities for each route within the capacity of the serving vehicle. However, in GCP, instead of minimizing the travel time, maximizing the garbage collection is envisioned with an assumption that it will reduce the requirement of the GCAs.

All the trivial and new ideas discussed before can be hypothesized as follows:

- Hypothesis I: Minimizing the travel time will minimize the requirements of the number of GCAs.
- Hypothesis II: Maximizing the waste volume collection by a GCA will minimize the requirements of the number of GCAs.
- Hypothesis III: Minimizing the coverage distance for collection per waste volume will minimize the requirements of the number of GCAs.

Among the aforementioned hypotheses, we consider hypothesis I as a trivial (benchmark) objective since it has been widely used in the evaluation of existing algorithms with a similar objective (e.g., CVRP [8]), whereas *hypotheses II* and *III* represent the proposed new objectives.

Similar to its predecessor, the GCP is an NP-hard problem for a large number of SGBs (i.e., $N \geq 100$). It is infeasible to solve this type of problem in polynomial time. Several metaheuristic methods that can produce near-optimum solutions have therefore been proposed since the last decade. Among them, genetic algorithms [4] are widely applied due to their reduced solving time and quality of solutions (if relevant parameters are selected properly). In this article, this technique is employed to obtain viable solutions.

A *GA* utilizes a set of *populations* and creates several *generations* to solve a particular optimization problem. A population consists of a set of solutions (a.k.a. chromosomes), each containing the solution in the form of genes. A *crossover* operation is performed for the reproduction of new chromosomes, whereas a *mutation* operation makes random changes in the solutions or chromosomes. A selection procedure is invoked to select only the fittest solutions as parents, which are then utilized by the crossover operation to create the other fit solutions, which are *offsprings*. At the end of each iteration, a new generation is produced from the combination of the old generation and the new offsprings. Generally, the size of the new generation is larger than the previous one. To keep the size fixed, the fitness values of all the solutions are calculated. At the end, a filtering procedure is applied so that only the fittest nodes survive and get themselves

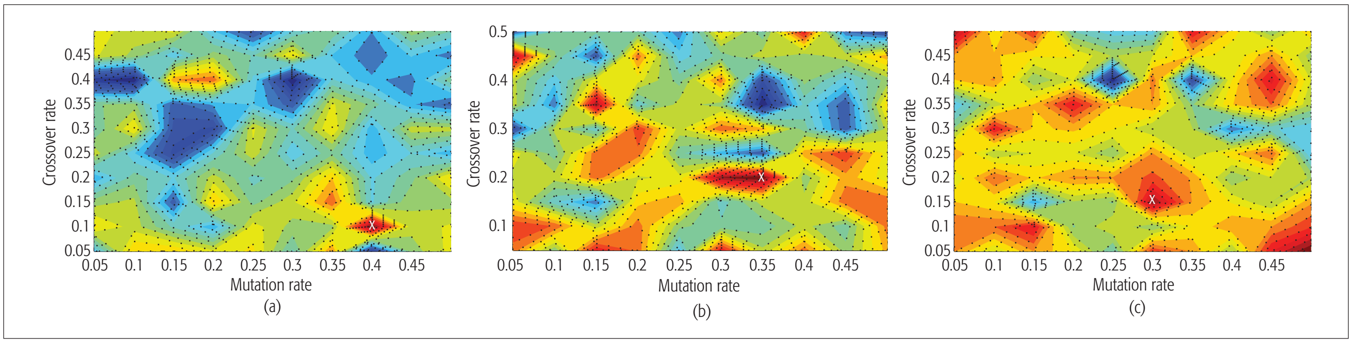


Figure 2. The impacts of various mutation rates and crossover rates on the utilization of the containers for diverse areas. The optimized mutation rate and crossover rate are pointed out using an x mark: a) area: 500 m × 500 m; b) area: 2000 m × 2000 m; c) area: 5000 m × 5000 m.

Area	Mutation rate	Crossover rate
500 m × 500 m	0.4	0.1
2000 m × 2000 m	0.35	0.2
5000 m × 5000 m	0.3	0.15

Table 1. Optimum mutation rates and crossover rates for the three preferred scenarios.

placed in the population. In our case, we need three corresponding fitness functions for three hypotheses, such as

$$\mathscr{F}(i)=\frac{1}{1+\left(\delta_{0,j}+\sum_{j=1}^{m}\delta_{j,j+1}+\delta_{m,0}\right)} \tag{1}$$

$$\mathscr{F}(i)=\sum_{j=1}^{m}\vartheta(j) \tag{2}$$

$$\mathscr{F}(i)=\frac{1}{1+\dfrac{\delta_{0,j}+\sum_{j=1}^{m}\delta_{i,j+1}+\delta_{m,0}}{\sum_{j=1}^{m}\vartheta(j)}} \tag{3}$$

where $\mathscr{F}(i)$ measures the fitness of a particular solution/chromosome i of a certain population, which has m number of genes (i.e., m SGBs), $\delta_{\ell,k}$ denotes the Euclidean distance between SGBs ℓ and k, $\vartheta(j)$ denotes the waste volume of a particular SGB j, $j \in \{1, \dots m\}$. Equations 1–3 are used to select solutions according to hypotheses I, II, and III, respectively. Since the volume of each SGB is considered random against a fixed capacity container, the size of the chromosomes/solutions may vary, which makes the implementation of the GA more challenging.

Evaluation

The proposed hypotheses are evaluated by conducting a comprehensive simulation campaign. The details of this simulation campaign along with parameter optimization and results analysis are discussed in the following.

Simulation Scenario

To evaluate our hypotheses, we consider three Euclidean 2D areas of 500 m × 500 m, 2000 m × 2000 m, and 5000 m × 5000 m, where the SGBs are installed in a random fashion. We consider a variable number of nodes N (ranging from 10 to 100) that are deployed within the area following a uniform probability distribution. Every SGB has a unique identification number, and in this process, 0 is considered as the identification number of the depot. We assign a random waste volume to every SGB, which is assumed to be less than the bin capacity β_c, and the capacity, ζ_c, of each container is assumed to be symmetric. To stress the simulation, all the nodes are considered to have a waste volume, which is larger than the minimum considerable volume μ (i.e., $\vartheta_i > \mu$). The distance of the two nodes is found using a Euclidean distance, δ. We assume that the node which travels within the shortest distance would require minimal time to travel the area. For simplicity, we also assume that the CC has adequate recent data to discover appropriate solutions.

In order to discover feasible solutions using the *GA*, 1-opt crossover and 1-opt mutation are utilized. The following parameters are considered throughout the simulation campaign: ζ_c = 1000 kg, βc = 200, $\mu = 0.5 \times \beta_c$, *generation* = 50, *sizeof(population)* = 2 × N. The length of the chromosomes/solutions varies from the minimum $\lfloor \zeta_c/\beta_c \rfloor$ to the maximum $\lfloor \zeta_c/\mu \rfloor$. Every scenario runs with 100 different seeds, which are then averaged before plotting on a graph. Finally, the simulation program has been implemented in C++, and all the results are tabulated in a plain text file.

Parameter Optimization

For finding appropriate solutions from a GA, it is obligatory to utilize optimum parameter values, which are volatile and can change from one scenario to another. Generally, *mutation rate* and *crossover rate* play important roles in discovering appropriate solutions in any evolutionary algorithm like a GA. Hence, a simulation campaign is carried out to discover optimum *mutation rates* and *crossover rates* for the three preferred scenarios. These values are later utilized in subsequent simulations. In Fig. 2, the impacts of various mutation rates and crossover rates on utilization of containers — where Eq. 3 is specifically selected for the fitness function — are shown using a contour graph for N = 30. In this figure, mild colors represent lower utilization, whereas intense colors represent higher utilization. It can be observed from the figure that multiple mutation-crossover-rate pairs may offer similar types of solutions. Hence, for subsequent simulations, the optimum parameter values in Table 1 are adopted.

Results and Discussion

For evaluating the performance of the three hypotheses and to discover their effectiveness in finding optimal or near optimal solutions, we consider three metrics: the required number of containers, utilization of the GCAs and travel distance per waste volume or, in short, distance per volume. The results are depicted in Figs. 3a–3c. All the results are normalized before plotting on graphs using a max-min normalization technique. Consequently, for each metric, the performance resulting from each given hypothesis does not vary significantly with the size of the area.

From Fig. 3a, it can be observed that since hypothesis II endeavors to maximize the garbage collection for a fixed capacity container, its utilization is considerably higher than the other two hypotheses for any preferred area. It achieves the highest utilization of vehicle capacity (i.e., 1 at N = 100). Since hypothesis III attempts to minimize the distance per volume collection, it achieves considerably higher utilization than hypothesis I (i.e., 0.74). These results of utilization reflect the requirement of the GCAs and are further illustrated in Fig. 3b. Since hypothesis II utilizes the GCAs in the most efficient manner, it requires a lower number of containers than the other two hypotheses. Between hypotheses I and III, the latter outperforms its counterpart. For hypotheses II and III, the required number of containers increase linearly with N. On the other hand, for hypothesis I, the required number of containers have a linear trend with N initially, but seem to have an exponential increase when N is sufficiently large. Moreover, hypotheses II and III appear to have nearly the same performance in terms of the required number of containers.

Although from the aforementioned discussion it may seem that hypothesis II yields superior performance, Fig. 3c shows other important insights. Since hypothesis II attempts to maximize the volume, a GCA has to travel a long distance, which is the longest among the three hypotheses (i.e., 0.82 or more). In contrast, although hypothesis III requires a slightly higher number of containers, its average travel distance is considerably lower than the preceding one. Again, another interesting observation is that initially hypotheses I and III yield almost equal travel distances, but as we increase N, hypothesis III will have a lower average travel distance per volume than its counterpart. From the investigation, it is found that since hypothesis I tries to minimize the distance, the distance for various containers increases chronologically. For instance, the first container has to travel the shortest distance, and the final one has to travel the longest distance, which is even longer than the longest distance of hypothesis III. Consequently, longer distances dominate when the average is calculated. Therefore, if fuel consumption is taken into account when calculating optimum solutions, hypothesis III might offer better performance (in terms of cost) than the other two hypotheses.

Conclusion

In this article, we have proposed a low-cost but efficient infrastructure-less network architecture, which is exploited in the smart garbage management system. Since the crowd is associated inseparably within the architecture, it is named the *crowd associated network*. A set of crowds works like mobile agents (called volunteer agents in this article) who acquire data from various dedicated agents of the network. At a later time, it delivers the acquired data to the other dedicated agent(s)

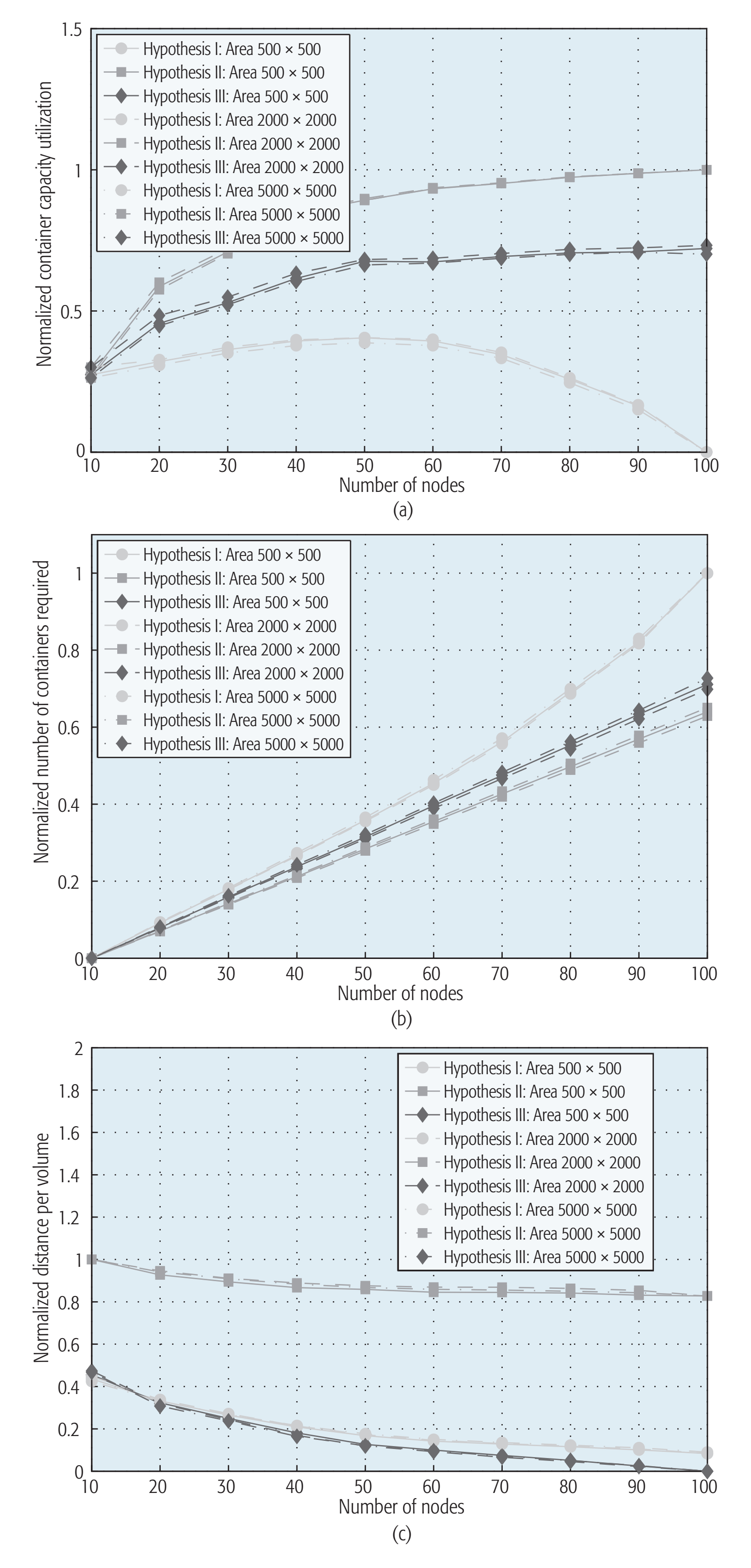


Figure 3. Results of three hypotheses for various metrics vs. number of nodes.

or similar agent(s) in the hope that the other parties can deliver the data to one or multiple dedicated agent(s). This combined effort is envisioned to deliver data to the destination sink, which is further connected to the control center. Thereby, these agents complete the communication gaps among the dedicated agents of the network. After receiving all the packets from various sources, at a later time, the control center computes optimum solutions with respect to garbage collection in order to minimize the management cost of the SGMS. We have employed a genetic algorithm to discover feasible solutions from the acquired data utilizing three objectives: minimizing the travel distance, maximizing garbage collection, and minimizing the travel distance per volume. We have performed an extensive simulation campaign with these objectives and discovered that the third objective seems to offer more feasible solutions than its counterparts.

We have employed the Genetic Algorithm to discover feasible solutions from the acquired data utilizing three objectives, namely minimizing the travel distance, maximizing garbage collection and minimizing the travel distance per volume. The third objective seems to offer more feasible solutions than its counterparts.

Acknowledgment

This work is partially supported by an RDU grant, (no. RDU160360), funded by University Malaysia Pahang (UMP), Malaysia.

References

[1] M. Abdulla Al Mamun *et al.*, "Integrated Sensing Systems and Algorithms for Solid Waste Bin State Management Automation," *IEEE Sensors J.*, vol. 15, no. 1, Jan. 2015, pp. 561–67.

[2] I. Hong *et al.*, "IoT-Based Smart Garbage System for Efficient Food Waste Management," *Scientific World J.*, vol. 2014, Aug. 2014.

[3] B. Chowdhury and M. U. Chowdhury, "RFID-Based Real-Time Smart Waste Management System," *Proc. Australasian Telecommun.Networks and Applications Conf.*, Christchurch, NZ, Dec. 2007.

[4] S. Longhi *et al.*, "Solid Waste Management Architecture Using Wireless Sensor Network Technology," *Proc. 5th Int'l. Conf. New Technologies, Mobility and Security*, May 2012.

[5] V. Catania and D. Ventura, "An Approach for Monitoring and Smart Planning of Urban Solid Waste Management Using Smart-M3 platform," *Proc. 15th Conf. FRUCT Assn.*, 2014.

[6] Y. Kryftis *et al.*, "Resource Usage Prediction Models for Optimal Multimedia Content Provision," *IEEE Systems J.*, vol. PP, no. 99, pp.1–12, doi: 10.1109/JSYST.2016.2548423, 2016.

[7] Y. Kryftis *et al.*, "Epidemic Models using Resource Prediction Mechanism for Optimal Provision of Multimedia Services," *Proc. 20th IEEE Int'l. Wksp. Computer Aided Modelling and Design Commun. Links Networks*, Guildford, U.K., Sept. 2015.

[8] K. Buhrkala, A. Larsena, and S. Ropkea, "The Waste Collection Vehicle Routing Problem with Time Windows in a City Logistics Context," *Proc. Social and Behavioral Sciences*, vol. 39, 2012, pp. 241–54.

[9] N. V. Karadimas, K. Papatzelou, and V. G. Loumos, "Genetic Algorithms for Municipal Solid Waste Collection and Routing Optimization," *Artificial Intelligence and Innovations 2007: From Theory to Applications*, vol. 247, 2007, pp. 223–31.

[10] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," Dept. of Comp. Sci., Duke Univ., tech. rep. CS-2000-06, Apr. 2000.

[11] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," *Proc. ACM WDTN*, Philadelphia, PA, Aug. 2005, pp. 252–59.

[12] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN Routing as a Resource Allocation Problem," *Proc. ACM SIGCOMM*, Kyoto, Japan, Aug. 2007, pp. 373–84.

[13] IEEE-SA, "IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," Apr. 2012.

[14] IEEE-SA, "IEEE 802.15: Wireless Personal Area Networks (PANs)," 2014.

[15] S. Y. Liew *et al.*, "A Store-and-Delivery Based MAC Protocol for Air-Ground Collaborative Wireless Networks for Precision Agriculture," *Proc. 3rd Int'l. Wksp. Internet of Things Technologies*, Melbourne, Australia, Dec. 14–17, 2015.

Biograpnies

Saiful Azad [M'16] received his Ph.D. in information engineering from the University of Padova, Italy, in 2013. Currently, he is serving as a member pf the Faculty of Computer Systems & Software Engineeringt, University Malaysia Pahang, and a Fellow of the IBM Center of Excellence, Malaysia. His research interests include design and implementation of communication protocols for various network architectures, QoS issues, network security, and simulation software design.

Arafatur Rahman [M'14] received his Ph.D. degree in electronic and telecommunications engineering from the University of Naples Federico II, Italy, in 2013. He worked as a postdoctoral researcher at the same university in 2014. Currently, he is an assistant professor with the Faculty of Computer Systems & Software Engineering, University Malaysia Pahang. His research interests include cognitive radio networks, IoT, and 5G, and he has co-authored around 40 journal and conference publications.

A. Taufiq Asyhari [M'13] received his Ph.D. degree from the University of Cambridge, United Kingdom. Since February 2017, he has been a lecturer in Networks and Communications at Cranfield University, United Kingdom. Prior to this position, he was at the University of Bradford, United Kingdom. He has won several academic awards, including a grant from NSC-Taiwan (2013) and the ISWCS 2014 Best Paper Award. His research interest is information theory with applications to wireless and nano-molecular networks.

Al-Sakib Khan Pathan [SM'14] received his Ph.D. in computer engineering in 2009 from Kyung Hee University, South Korea, and his B.Sc. in computer science and information technology from Islamic University of Technology, Bangladesh, in 2003. He is currently an associate professor in the Computer Science and Engineering Department at Southeast University, Bangladesh. He has served as a Chair and committee member of numerous international conferences, and in editorial roles for several renowned journals.

A Hitchhiker's Guide to Computation Offloading: Opinions from Practitioners

Morteza Golkarifard, Ji Yang, Ali Movaghar, and Pan Hui

The authors review developments in the field of pervasive computing on computation offloading. They use this literature review together with their own experience and provide designers with some detailed guidelines to gain a deep insight into the implementation challenges of a computation offloading system. The guidelines empower the reader to choose between the variety of solutions in the literature for developing any offloading system with the consideration of their own system architecture and available facilities.

ABSTRACT

Due to the increasing usage and capabilities of smart devices, mobile application developers build a large number of resource intensive applications, such as WAR applications. Even with the rapid development in hardware technology, the computing capability and battery capacity on wearable devices and smartphones still cannot meet the application demands with heavy computations and high battery drain. Pervasive computing addresses this problem by migrating applications to the resource providers external to mobile devices. The profitability of this method heavily depends on how to implement it and when to use it. Although there are many computation offloading systems proposed in the literature, there is no practical manual that addresses all the implementation complexities on the way to building a general offloading system. In this article, we review developments in the field of pervasive computing on computation offloading. We use this literature review together with our own experience and provide designers with some detailed guidelines to gain a deep insight into the implementation challenges of a computation offloading system. The guidelines empower the reader to choose between the variety of solutions in the literature for developing any offloading system with the consideration of their own system architecture and available facilities. Finally, we evaluate our general offloading system on Android devices with two real-time applications.

INTRODUCTION

Increasing capability and diversity of mobile applications combined with pervasive usage of smart devices makes mobile applications an inevitable part of everyday life. Wearable devices such as Google Glass, endowed with sensors, camera, processing, and communication power, provide users with useful wearable augmented reality (WAR) applications. These applications are fed by information about users' behavior that is captured by sensors on mobile devices, such as camera, GPS, Bluetooth, accelerometer, ambient light, magnetometer, and microphone using an adaptive sampling mechanism.

Recent research studies have introduced assisting systems for cognitively disabled people to improve their daily life quality. Nevertheless, such applications usually require the implementation of some rather complex algorithms that have high demand on both computation and battery. Smart devices, such as wearables and smartphones, have limited resources such as CPU, battery life, storage capacity, and network bandwidth. This limitation is an obstacle for developing resource intensive mobile applications and has been addressed by research in the past decade [1, 2] in the area of pervasive computing.

Computation offloading is a solution to augment the capabilities of mobile devices by migrating computation to more resourceful devices. Many novel WAR applications employ computation offloading to perform heavy tasks, such as recognizing objects, faces, activities, text, and sound. Toward wearable cognitive assistance, the authors of [3] introduced an application prototype that provides cognitive software such as face detection, object detection, OCR application, motion classification, activity inference, and augmented reality. The authors of [4] implemented three practical augmented reality (AR) applications with computationally intensive operations based on wearable devices and leverage the code offloading technique to outsource large computations.

Thanks to the advanced generation of wireless technology in mobile devices such as Bluetooth 4.0, WiFi IEEE 802.11ac, and fourth generation (4G) networking, some capable devices can lend their available resources to other mobile devices to remotely execute their tasks. Figure 1 shows a general system architecture with environmental facilities for computation offloading. Mobile devices in such an environment can adopt three major techniques for computation offloading based on their capabilities and the facilities provided: i) employ cloud services in the presence of high-quality Internet connectivity to offload computation to a server cluster; ii)employ cloudlets, which take advantage of virtual machine (VM) technology to provide remote execution for other connected nearby mobile devices; and iii) use nearby mobile devices as resource providers in order to migrate and process computations. In this scenario, it is assumed that mobile devices are directly connected to Bluetooth or connected to WiFi through a router. In the first method, the biggest concern is the response time to an offloading request from a mobile device. Therefore,

Morteza Golkarifard and Ali Movaghar are with Sharif University of Technology; Ji Yang and Pan Hui are with the Hong Kong University of Science and Technology.

Digital Object Identifier: 10.1109/MCOM.2017.1600481

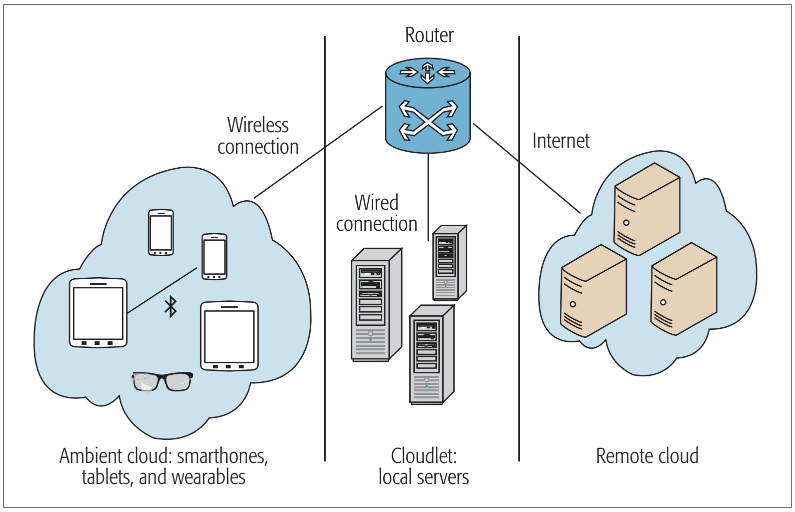


Figure 1. General system architecture for computation offloading.

researchers try to leverage cloudlets. Although it is possible to offload computation in cloudlets without Internet access, cloudlets are as difficult as cloud VMs to configure for offloading.

In this article, we aim to provide comprehensive guidelines from practitioners for implementing a general offloading system. We consider all major complexities one might face in order to build an offloading system. Our extensive suggestions, based on practical experience, for each of these issues pave the way to a complete realization of implementation details regarding any offloading system. Unlike other proposed offloading systems in the literature, our article shows how to build a general offloading system that can use cloud, cloudlets, and nearby devices all as offloading targets. The rest of this article is organized as follows. We begin by reviewing related works and categorize them in the following section. Then we discuss when and where it is worth doing computation offloading. Following that is the main part of the article, providing extensive guidelines. We evaluate the system using some real experiments after that. We conclude this article and discuss some future research directions in the final section.

Categorization of Computation Offloading Frameworks

Offloading systems are categorized based on the offloading target into cloud offloading and device cloud offloading. Cloud offloading is defined as offloading to servers. The device cloud refers to the ambient cloud that is formed by mobile devices. Each device in this ambient cloud can be the offloading target.

Cloud Offloading

Cloud offloading itself can be divided into two subclasses: remote cloud offloading and cloudlet offloading. In this section, we present prominent works in this area.

In 2010, MAUI [5] was introduced to transfer computation from a mobile device to a local server by using fine-grained code offloading. The authors claimed that MAUI improves energy consumption and application performance, while minimizing the burden on programmers by automating program partitioning. The system takes advantage of code portability to create two versions of a code, one running locally on the mobile device and the other running remotely on the server.

CloneCloud [6] operates at thread granularity and enables smartphones to migrate a heavy thread, while simultaneously running other essential threads. In this system, the programmer does not need to annotate the source code in any special idiom. CloneCloud employs a static analyzer and a dynamic profiler along with an optimization solver to automatically decide which part of the application should be retained on the phone and which part should be migrated to the local server.

ThinkAir [7] focuses on scalability and elasticity of the cloud. The system utilizes multiple VM images to parallelize method execution on the cloud. The authors customized Android-x86 to provide native code support on the VMs. ThinkAir takes advantage of an accurate profiler to make correct offloading decisions with low overhead.

COMET [8] elaborated on the idea of using distributed shared memory (DSM) and applied it to offloading. COMET supports multi-threading and uses VM synchronization to keep the runtime execution information in a consistent state. Furthermore, the system has a simple scheduler. It relies on past behavior to decide which local thread should be migrated.

Tango [9] replicates the application and executes it on both client and cloud servers. It uses a *flip-flop* mechanism to allow leadership to switch between replicas. The replica that executes faster is selected as the leader.

Device Cloud Offloading

In this section, we review and summarize several recent systems performing computation offloading using nearby devices.

Misco [10] is the first MapReduce framework for computation offloading using nearby devices. Misco splits a large computation into a number of smaller tasks. These tasks are totally independent, so they can be executed on multiple worker nodes in parallel. The Misco framework includes a master local server and a number of worker nodes. The master server keeps track of user application, while worker nodes are responsible for performing the map and reduce operations.

In 2012, Shi *et al.* proposed Serendipity [11] by considering device-to-device (D2D) communication. This system uses available mobile devices in the environment as remote computational resources. Serendipity selects the same device for data-dependent computations to reduce data exchange between devices. The profiler is similar to MAUI and CloneCloud. The execution time and energy consumption are estimated by running the program multiple times with different input data. Using the information provided by the profiler, Serendipity decides whether to disseminate a task to other nodes or to execute it locally.

HoneyBee [12] leverages a *work stealing* algorithm to load balance the tasks across a network of mobile devices. This algorithm aims to keep every node as busy as possible. The authors did not suggest a convenient way for Android application programmers to benefit from the system; nor did they discuss the energy consumption of their algorithms compared to other similar works.

Sapphire [13] is a distributed runtime framework that helps developers implement mobile/cloud applications in a multi-platform environment. Sapphire is designed for flexibility and extensibility and lets programmers easily modify their distributed application deployments without needing to rewrite major parts of the code. To meet this, Sapphire suggests separation of the application logic from deployment logic, which allows programmers to focus only on the application logic.

Where and When to Offload

The main goals of computation offloading are to maximize the battery life of smart devices and to minimize the execution time of tasks compared to local execution. Smartphone users usually want to acquire more computational resources for faster execution, while they prefer to save energy in extreme cases where the battery level is lower than some level. However, the amount of time and energy saved by computation offloading heavily depends on where your offloading target is and when offloading takes place: *where* to offload and *when* to offload.

The offloading target can be remote or local depending on the offloading strategy, available nearby resources, and the existence of high-speed Internet connection. Offloading target candidates can be a proximal resource-rich computer, a cluster of computers, a more powerful mobile device, or a cloud service with higher configuration.

When to offload indicates when offloading to other devices is more beneficial than local execution in terms of execution time and energy. An offloading system can improve the total execution time for the offloader when the round-trip time of code offloading is smaller than the local execution time. As mentioned in [14], a code offloading procedure contains three steps: sending, invocation, and receiving. This means in remote execution, improvement in the invocation phase should cover the delivery time of a task. Furthermore, high bandwidth between the offloader and offloading target is necessary to cope with the extra overhead of transferring codes, input data, and results in the remote execution. According to Tables 1 and 2, although the latest Bluetooth v. 4 technology provides the Bluetooth Low Energy (BLE) protocol, it is not a proper choice for offloading large tasks compared to WiFi protocols due to limited bandwidth.

In addition to high bandwidth, the proximity of the offloading target can also have an impressive impact on the delivery time. When there is no interference in a local network, connections have negligible latency. However, the latency will be considerable when using cloud services for offloading via the Internet. Using cloud VMs located in different geographical regions would significantly increase the latency, the delivery time, and consequently the total offloading time. For example, if we would like to offload 300 kB of an image frame and codes for an text recognition application in a low-latency network, we need at least a 24 Mb/s data transfer rate to complete the data transmission in less than 0.1 s.

The number of intermediate devices between the offloader and offloading target can have considerable impact on delivery time, too. Although a technique that is called *multihop code offloading* can provide more computational resources for faster execution, its cost in delivery time will sometimes be too high to compensate. We measured the bandwidth of multihop code offloading both in the presence of interference from two cross flows and without it. Figure 2 demonstrates that using multihop code offloading can decrease bandwidth exponentially in both cases.

Version	Max rate
1.2	1 Mb/s
2.0	3 Mb/s
3.0	24 Mb/s
4.0	25 Mb/s

Table 1. Comparison of maximum data transfer rate of different Bluetooth generations.

Protocol	Max rate
802.11a	2 Mb/s
802.11b	11 Mp/s
802.11g	54 Mb/s
802.11n	150 Mb/s
802.11ac	411 Mb/s

Table 2. Comparison of maximum data transfer rate of different WiFi protocols.

How to Offload

In this section, we describe several implementation details regarding the main components of a general offloading system.

Profiler

Similar to [5, 7], our profiler design contains all the information of the hardware parameters, the application parameters, and the network. For the hardware profiler, we monitor the following states:

- CPU utilization and CPU frequency state
- Screen brightness, which ranges from 0 to 255
- Power state of WiFi, Bluetooth, and mobile broadband (e.g., 3G, 4G) interfaces
- Signal strengths of WiFi, Bluetooth, and mobile broadband interfaces indicated by received signal strength indicator (RSSI) value

The application profiler mainly monitors the applications in runtime. It monitors the following information:

- Execution time of a method
- Number of instructions of a method
- The input and output parameter size
- Number of threads invoked by a method
- Thread memory allocation size

The network profiler records the followings parameters:

- RTT of the remote execution servers
- Bandwidth of the remote execution servers
- The connected stream interfaces
- Number of packets transmitted

The offloading target can be remote or local depending on the offloading strategy, available nearby resources and the existence of high speed Internet connection. Offloading target candidates can be a proximal resource-rich computer, a cluster of computers, a more powerful mobile device, or a cloud service with higher configuration.

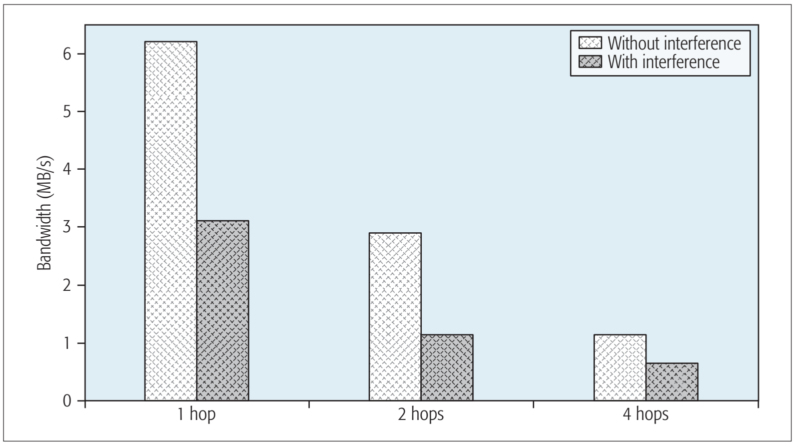


Figure 2. 1-hop bandwidth comparison with multihop. We add two phone pairs to generate two cross traffic flows for inducing signal interference.

Method Selection

The code partitioning component determines *what* to offload. Since most object-oriented programming languages support object serialization and dynamic method invocation, we consider the methods as the job components in our general offloading system. Similar to [5], system developers should notice that certain methods should not be offloaded:

- Methods that interact with user interface, for example, a method trying to update results on the screen
- Methods that interact with I/O devices, such as keyboard, touchscreen, GPS, accelerometer, gyroscope, compass, or barometer
- Methods that interact with any external component such as a method using a pre-installed library package or a static variable

Some methods with high potential to benefit offloading are:

- Methods with CPU or memory intensive tasks, especially with small input and output
- Data-independent methods that are called many times and can be executed simultaneously
- Methods that require large data transfers, such as a method that downloads a large amount of data from the Internet and processes them

After specifying candidate methods, the programmer should create a wrapper for each selected method, encapsulate required information, and offload it. This procedure can be done automatically by a code generator tool, which is responsible for parsing the source code and creating an offloadable version from it.

Object Marshalling

Marshalling an object is the process of converting a data structure stored in memory into a byte array, so it would be suitable for storage and transmission [5]. Several object-oriented programming languages such as Java, Python, and C++ support serialization techniques for object marshalling. We address the mechanism used by some of these languages here in detail. In C++, the `Boost.Serialization` library makes it possible to simply marshall an object. In Objective-C and iOS, there are three main approaches for data serialization using `NSCoding` and `NSCoder` classes: property lists, JSON, and XML.

In Java, two major methodologies can be used for encoding an object into a byte array: `Parcelable` and `Serialization`. Although `Serialization` has a simpler implementation compared to the `Parcelable` technique, it is much slower than `Parcelable` in serializing and deserializing an object because `Serialization` marshalls objects using Java reflection and needs more garbage collection.

Network Connection

In order to initialize offloading, a connection between the offloader and the target device should be established. This connection is made through a WiFi, Bluetooth, or mobile broadband interface.

Bluetooth provides a single-hop connection, which can only be used for device cloud offloading. Android and iOS both offer application programming interfaces (APIs) to work with Bluetooth programmatically. The offloader uses the Bluetooth medium access control (MAC) address to initiate a connection. The Service Discovery Protocol (SDP) provides a universal unique identifier (UUID), which is a 128-bit string, to reach a connection agreement between the client and server. If the connection is successful, a connected Bluetooth socket will be returned, and both devices use this socket to transfer data.

Mobile broadband is used for remote cloud offloading when Internet access is not available through WiFi. Also, WiFi connection is used for device cloud offloading. In this case, tasks in one device are disseminated to other devices by single-hop or multihop communication. For single-hop communication, the offloadee should use WiFi Direct or act as a WiFi access point to which other devices can connect. This approach is practical when users want to connect a wearable device to the smartphone in their pockets for offloading an application with high data transfer like image processing applications. For multihop communication, the offloader connects to a router and offloads tasks to a target device in the device cloud.

Suspend-Resume Mechanism

When the execution of an application reaches an offloading point, the offloading system suspends the executing thread, encapsulates the state of the application, and sends the package to the target device. Finally, the received offloading results (including the new state of the application) are merged to the current state of the application, and the offloading system resumes execution of the current thread [6].

The minimum information that is required for the remote execution of a method includes: a current object of the method's class, method name, types of the method parameters, and values of the method parameters to the offloadee device. All of this information should be packed into a serializable object.

There are several ways to implement the suspend-resume mechanism. Java provides thread synchronization with low-level concurrency primitives such as *synchronized*, *wait()*, and *notify()*, and high-level utilities in the *java.util.concurrent* package such as *Semaphore*, *Future*, *Executors*,

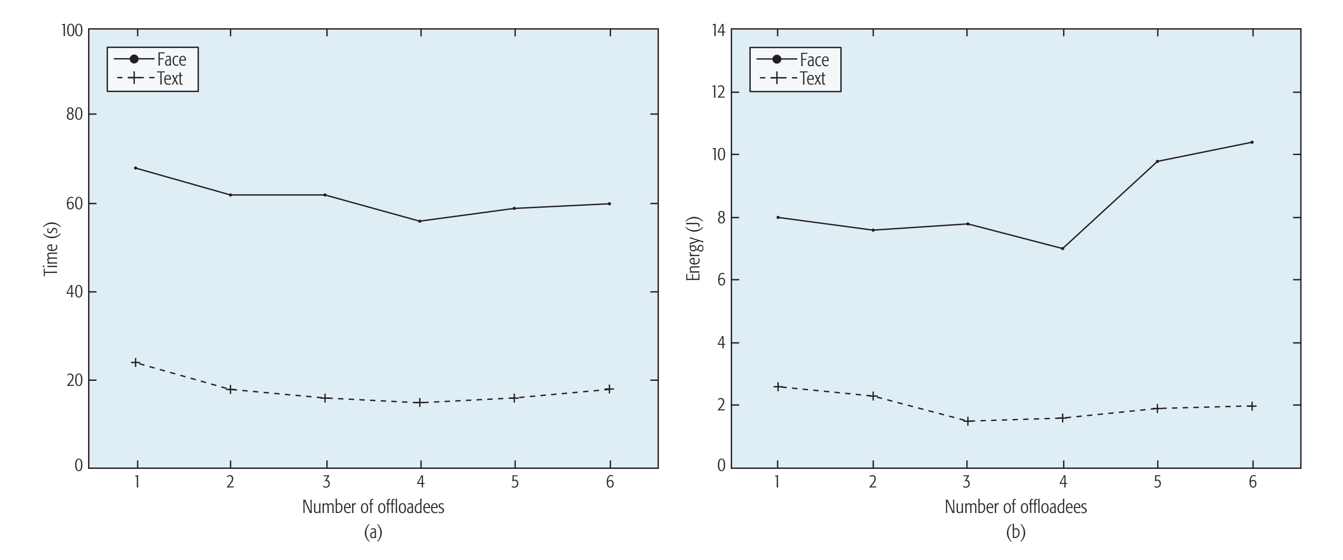


Figure 3. Influence of number of offloadees on offloading efficiency: a) execution time; b) energy consumption.

and *CountDownLatch*. Also, C++ and Objective-C suggest *condition variables* and the *Mutexes* mechanism.

Decision and Scheduling

In our system, we suggest a simple, powerful task scheduling algorithm to improve performance in terms of both execution time and energy. This task scheduler is a two-layer architecture. The first step is to find which kind of offloading architecture to use: local, D2D, or cloud. For this purpose, we design a multiclass support vector machine (SVM)-based model to make decisions. In this model, we train two classifiers: a time incentive classifier and an energy incentive classifier. In the training process, three copies of the application execute on the local device, neighbor mobile device, and cloud server. The profiler records all the features, and labels the data according to the performance of the task. In the second step, we choose the offloadee device in the selected offloading architecture.

If the system selects cloud offloading, the VM managers will schedule the tasks in a way to balance the load [7]. The VM management finds the daily patterns for the traffic and manages the VM creation and destruction to save monetary cost. It can start a new VM instance, destroy an expired VM, and put to sleep/wake a created VM. The creation and destruction of VM instances are implemented by installing the AWS CLI packages.

For D2D offloading, however, we need to design a task dissemination algorithm to disseminate multiple tasks among the list of offloadees. We propose a greedy solution to find an optimal device allocation for each task. We use a linear function to represent the offloader's gain when a task is offloaded to a specific offloadee. This functions is based on some common criteria: energy (*e*), CPU (*c*), and time (*t*). Given the specific decision solution s_i, the evaluation function (*f*) evaluates how much *e*, *c*, and *t* is saved in total:

$$f(s_i) = w_1 e_i + w_2 c_i + w_3 t_i \quad (1)$$

where w_1, w_2, and w_3 are the weight parameters and $w_1 + w_2 + w_3 = 1$. In our system, each device profiles runtime parameters and uses a power model to estimate the energy consumption over an execution time period. For the power model, we take our inspiration from PowerTutor [15], a powerful energy consumption measurement tool. All the information about estimated energy and execution time will be stored in the offloader. The offloader schedules the task using the history of executions and current device workload.

Error Handling

There are two types of exceptions that could possibly happen in the offloading process, `RemoteException` and `NetworkException`. `RemoteException` includes all errors occurring inside offloadee devices. If `RemoteException` happens, offloadees will throw this exception and send it to the offloader with the first priority. The offloader also marks each "exceptional" offloadee. The offloadee will be excluded from the device list if it continuously throws `RemoteExceptions`. `NetworkException` occurs when the connection is lost or the socket stream is blocked. The network module produces an exception to be processed by the Recovery module. The Recovery module tries to re-ping and reconnect to the offloadee. The offloader will wait for the offloadee results provided before a deadline. Otherwise, the system will perform actions similar to `RemoteException`.

Environment Configurations

Traditionally, VM surrogates of mobile devices in the cloud have been seen as the best candidates to offload heavy computation. OSX Server and iOS are licensed only for use on Apple hardware. Therefore, we only discuss cloud offloading for Android devices.To offload Android-compatible programs on an x86 architecture, Android x86 VMs should be deployed on cloud servers. In this section, we discuss the detailed configurations we need on cloudlets and clouds for offloading Android-compatible programs.

Local Server Configuration: There are two ways to run Android on a local server: Android Device Emulator and Android-x86.

The performance is the worst when the tasks are offloaded to Amazon EC2 because the bandwidth is slow when we access it through both public networks and campus networks. Also, cloudlets offloading can have similar real-time performance as D2D. However, this type of offloading service is hard to deploy and access, where there are no servers and WiFi access points.

The Android Open Source Project offers a built-in mobile device emulator in the Android SDK. The Android virtual device manager provides a graphical user interface (GUI) to model an actual device by defining the hardware and software options. It is recommended to install the Intel Hardware Accelerated Execution Manager (HAXM), which is a hardware-assisted virtualization engine, in order to speed up Android application emulation on the x86 machine. This option can be enabled in the option window when we create a new Android virtual device.

The next approach is virtualizing the hardware on a guest local server. There are many suitable virtualization platforms, such as Xen, QEMU, and Oracle's VirtualBox. These platforms are used for virtualizing the hardware and installing a variety of guest operating systems. We set up a VM with minimum of 512 MB of RAM on VirtualBox. Next, we install and run an available version of the Android-x86 port (http://www.android-x86.org/) on the VM [7]. Although this way is straightforward, current android-x86 ports are unofficial and not stable enough for daily use in general.

Amazon EC2 Instance Configuration: The Amazon Elastic Compute Cloud (EC2) is a web service that allows users to rent VMs. Users can boot an Amazon Machine Image (AMI) to create a VM instance. By signing up for the Amazon Web Service (AWS — http://aws.amazon.com/), the user can access all Amazon services including Amazon EC2. On the Amazon EC2 service web page, click "Launch Instance" to define a VM instance. In the displayed window, you can choose an AMI from a list of suggested AMIs. The authors of ThinkAir [7] provide two versions for Android-x86 AMIs (http://claudiu-perta.appspot.com/android-x86/ami/), which can be launched directly on the Amazon EC2 account. Users can choose the virtual server type by selecting a combination of CPU, memory storage, and network capacity. The suggested Android-x86 AMIs are based on i386 architecture and do not support HVM virtualization, so there are only four choices for Android-x86 instance type. It is recommended to select an instance with minimum of 2 GB of RAM and 8 GB of disk storage to run an Android-x86 AMI.

Evaluation

There are plenty of D2D applications that can benefit from our general offloading system. Here, we design and implement Face Detection as a data and computation intensive application and Text Recognition as a real-time application to evaluate our general offloading system.

The Face Detection application takes a bitmap graphic object as input and finds the faces and their rotation angles in the picture. The Text Recognition application continuously captures frames from the camera and recognizes the text inside them. We import the `tess-two` (https://github.com/rmtheis/tess-two) library for the OCR engine, which has comparatively higher recognition accuracy and lower computation complexity.

Number of Offloadees

We run several experiments to show the number of offloadees' influence on the offloading efficiency, as shown in Fig. 3. In these experiments, the offloader is a Google Glass, while offloadees are Nexus5 smartphones. In addition, the offloader and offloadees are static and near each other, which indicates that Bluetooth connection between them is stable. The offloader generates 10 Face Detection requests and 20 Text Recognition requests for this experiment.

Increasing the number of offloadees reduces the pure execution time in the offloader, but transmission time will not be improved since the offloader should send a fixed amount of data to offloadees. Figure 3 demonstrates that we have the best performance and energy consumption when we have four offloadees. This number can change depending on the application execution time and transmission time.

Offloading Target Type

We run the Text Recognition application with D2D, cloudlets, and remote cloud offloading through a WiFi interface. In the D2D case, the offloadee is a Nexus5 smartphone. In the cloudlet case, an Android x86 VM is deployed on one desktop. The VM has four 2.3 GHz CPU cores and 2 GB memory. For the remote cloud, we run the Android x86 on the Amazon EC2 *C3.large* instance, which is located in Singapore and has 3.75 GB memory and two 2.8 GHz virtualized CPU cores. As shown in Fig. 4, we compare the runtime performance in four different scenarios where the Text Recognition application is executed in the offloader (local), offloadee devices (D2D), cloudlets, and Amazon , respectively. In Fig. 4a, we can see that D2D performs the best in all of these four scenarios.

The performance is the worst when the tasks are offloaded to Amazon EC2 because the bandwidth is slow when we access it through both public networks and campus networks. Also, cloudlet offloading can have similar real-time performance as D2D. However, this type of offloading service is hard to deploy and access, where there are no servers and WiFi access points.

Future Research Directions

In this article, we practically investigate a solution to build an offloading system and show application developers the necessary steps to implement such a system. However, there are some important research issues that remain to be solved. In our future work, we will try to optimize the programming framework of computation offloading for application developers. The future work also involves the offloading decision modeling. Current works mainly focus on how to offload rather than offloading efficiency. Most of the previous research used linear programming to perform static analysis to identify which parts of the code need to be offloaded. However, this approach is NP-hard. Future work may need to consider some more practical and efficient offloading decision frameworks.

Furthermore, security is a challenging issue in mobile computing and offloading. Most of the recently proposed frameworks tend to offload the security/privacy related tasks to the cloud, where the cloud is assumed to be trustworthy. Therefore, security is still an open challenge for mobile computing and offloading. There are opportunities to investigate which security measures need to be taken into account for encrypting the communications between the mobile device and the resource providers. In all of the aforementioned

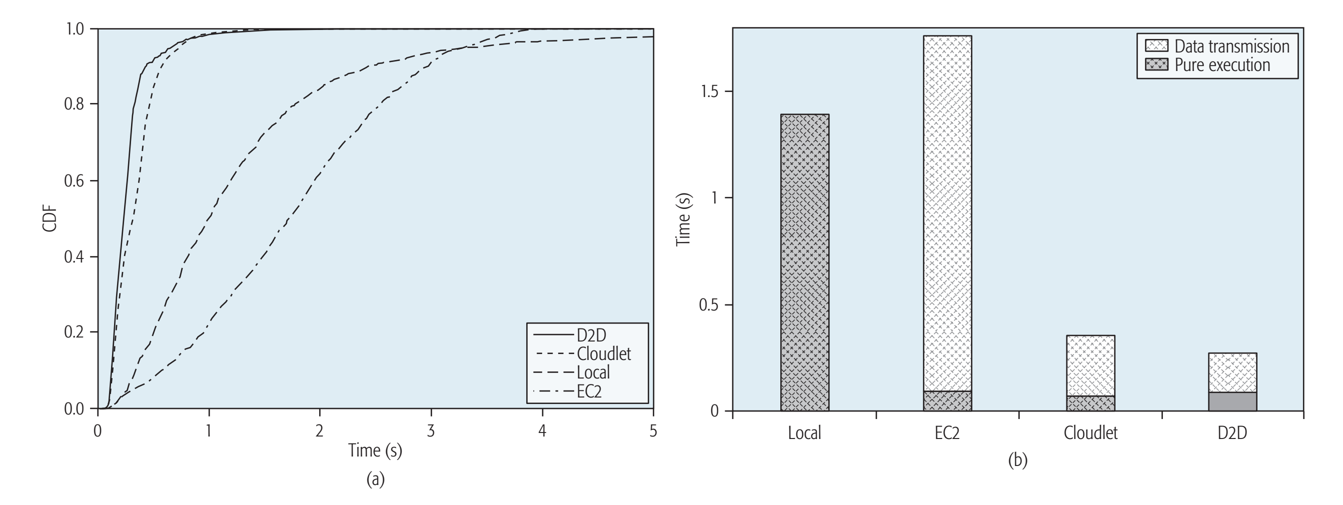


Figure 4. Total time for individual method in four different scenarios: a) execution time CDF; b) average execution time.

works, it is assumed that the initiator and the resource provider nodes are both trustworthy. Nevertheless, the question "how do the offloadees trust the offloader and vice versa?" remains unsolved. In order to make a system robust and defensive, it must be protected from selfish and malicious behaviors. Such a system would be able to achieve improvement in performance and energy. Consequently, we need to investigate which pricing model should be employed to reduce the monetary cost per request to the provider. Furthermore, in a competitive environment, we need to think about what the best strategies are to offer mobile users the best service plan.

Acknowledgments

This research has been supported, in part, by General Research Fund 26211515 from the Research Grants Council of Hong Kong and the Innovation and Technology Fund ITS/369/14FP from the Hong Kong Innovation and Technology Commission.

References

[1] R. Balan *et al.*, "The Case for Cyber Foraging," *Proc. 10th ACM SIGOPS Euro. Wksp.*, 2002, pp. 87–92.

[2] R. K. Balan *et al.*, "Simplifying Cyber Foraging for Mobile Devices," *Proc. 5th ACM Int'l. Conf. Mobile Systems, Applications and Services*, 2007, pp. 272–85.

[3] K. Ha *et al.*, "Towards Wearable Cognitive Assistance," *Proc. 12th Annual Int'l. Conf. Mobile Systems, Applications, and Services*, 2014, pp. 68–81.

[4] B. Shi *et al.*, "Offloading Guidelines for Augmented Reality Applications on Wearable Devices," *Proc. 23rd ACM Int'l. Conf. Multimedia*, 2015, pp. 1271–74.

[5] E. Cuervo *et al.*, "Maui: Making Smartphones Last Longer with Code Offload," *Proc. ACM MobiSys*, 2010, pp. 49–62.

[6] B.-G. Chun *et al.*, "Clonecloud: Elastic Execution between Mobile Device and Cloud," *Proc. ACM EuroSys*, 2011, pp. 301–14.

[7] S. Kosta *et al.*, "Thinkair: Dynamic Resource Allocation and Parallel Execution in the Cloud for Mobile Code Offloading," *Proc. IEEE INFOCOM*, 2012, pp. 945–53.

[8] M. S. Gordon *et al.*, "Comet: Code Offload by Migrating Execution Transparently," *Proc. 10th USENIX Conf. Operating Systems Design and Implementation*, 2012, pp. 93–106.

[9] M. S. Gordon *et al.*, "Accelerating Mobile Applications Through Flip-Flop Replication," *Proc. 13th Annual Int'l. Conf. Mobile Systems, Applications, and Services*, MobiSys '15, 2015, pp. 137–50.

[10] A. Dou *et al.*, "Misco: A Mapreduce Framework for Mobile Systems," *Proc. ACM PETRA*, 2010, p. 32.

[11] C. Shi *et al.*, "Serendipity: Enabling Remote Computing among Intermittently Connected Mobile Devices," *Proc. ACM MobiHoc*, 2012, pp. 145–54.

[12] N. Fernando *et al.*, "Honeybee: A Programming Framework for Mobile Crowd Computing," *Proc. MobiQuitous*, 2013, pp. 224–36.

[13] I. Zhang *et al.*, "Customizable and Extensible Deployment for Mobile/Cloud Applications," *Proc. 11th USENIX Conference on Operating Systems Design and Implementation*, 2014, pp. 97–112.

[14] H. Flores *et al.*, "Mobile Code Offloading: From Concept to Practice and Beyond, *IEEE Commun. Mag.*, vol. 53, no. 3, Mar. 2015, pp. 80–88.

[15] L. Zhang *et al.*, "Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones," *Proc. IEEE CODES/ISSS*, 2010, pp. 105–14.

Bibliographies

Morteza Golkarifard received his B.Sc. and M.Sc. degrees in computer engineering from Sharif University of Technology. He is currently a Ph.D. student and a member of the Performance and Dependability Laboratory (PDL) in the Computer Engineering Department at Sharif University of Technology. He was also a visiting scholar from July 2014 to December 2014 in the Hong Kong University of Science and Technology (HKUST)-DT System and Media Laboratory. His research interests include software defined networking and mobile cloud computing.

Ji Yang received his B.Sc. in 2014 and M.Phil. in 2016, both at HKUST. He is currently a research assistant working in the HKUST-DT System and Media Lab. His research interests are in the area of mobile computing and human-computer interaction.

Ali Movaghar [SM] is currently a professor in the Department of Computer Engineering at Sharif University of Technology, Tehran, Iran. He received his Ph.D. degree in computer, information, and control engineering from the University of Michigan at Ann Arbor in 1985. His research interests include performance/dependability modeling and formal verification of wireless networks, distributed real-time systems, and cyber-physical systems. He is a Senior Member of the ACM.

Pan Hui received his Ph.D. from the University of Cambridge. He is director of the HKUST-DT System and Media Lab. He is an adjunct professor of social computing and networking at Aalto University, Finland, and was a Distinguished Scientist for Telekom Innovation Laboratories (T-labs), Germany. He is an Associate Editor for *IEEE Transactions on Mobile Computing* and *IEEE Transactions on Cloud Computing*, and an ACM Distinguished Scientist.

How Far Are We from WebRTC-1.0? An Update on Standards and a Look at What's Next

Salvatore Loreto and Simon Pietro Romano

Real-time communication between browsers has represented an unprecedented standardization effort involving both the IETF and the W3C. These activities have involved both the real-time protocol suite and the application-level JavaScript APIs to be offered to developers in order to allow them to easily implement interoperable real-time multimedia applications in the web. The authors shed light on the current status of standardization, with special focus on the upcoming final release of the so-called WebRTC-1.0 standard ecosystem.

Digital Object Identifier:
10.1109/MCOM.2017.1600283

Abstract

Real-time communication between browsers has represented an unprecedented standardization effort involving both the IETF and the W3C. These activities have involved both the real-time protocol suite and the application-level JavaScript APIs to be offered to developers in order to allow them to easily implement interoperable real-time multimedia applications in the web. This article sheds light on the current status of standardization, with special focus on the upcoming final release of the so-called WebRTC-1.0 standard ecosystem. It takes stock of the situation with respect to hot topics such as codecs, session description and stream multiplexing. It also briefly discusses how standard bodies are dealing with seamless integration of the initially competing effort known as "Object Real Time Communications."

Background, Rationale and Motivation

Real-time communication in the web has been the subject of a challenging standardization process for the last five years or so. Back in 2011, the Internet Engineering Task Force (IETF) chartered the "Real-Time Communication in WEB-browsers" (RTCWEB) Working Group, with the aim of defining an architecture and a complete suite of protocols for the support of real-time multimedia communications directly between browsers. The RTCWEB WG has since worked on key aspects like the overall communication infrastructure, the protocols and API (application programming interface) requirements, the security model, the media formats (and related media codecs), as well as advanced functionality like congestion/flow control and interworking with legacy VoIP equipment.

In parallel, the World Wide Web Consortium (W3C) has conducted an activity defining a set of APIs exposing functions like exploration and access to device capabilities, capture of media from local devices, encoding/processing of "media streams", establishment of peer-to-peer connections between browsers (and web-enabled devices in general), decoding/processing of incoming media streams and delivery of such streams to the end-user in an HTML5-compliant fashion.

To date, the two mentioned working groups have achieved a major milestone in the field of real-time multimedia communications: the so-called *WebRTC*-1.0 standards suite. The idea behind WebRTC-1.0 is to allow all of the involved stakeholders (browser vendors, telecommunication providers, application providers, web developers, and so on) to converge on a well-defined set of protocols and APIs to be leveraged in order to allow widespread deployment on the market of interoperable products offering end-users a media-rich, web-enabled, real-time experience. To achieve this goal, the standardization process has necessarily had to face a number of obstacles while trying to strike a balance among diverging interests and/or viewpoints.

This article will briefly survey the current state of the art with respect to WebRTC-1.0 completion and introduce the envisioned work program for the second generation of the standard. In doing so, it will touch upon debated topics and illustrate how the community has successfully coped with them.

State of the Art

Related Work

In our previous work on the subject [1] we discussed the evolution of real-time communication in the web, by highlighting the main steps that brought the IETF and the W3C to the launch of the joint standardization initiatives known, respectively, as *RTCWEB* and *WebRTC*. At the time of that writing the standards process had already reached a good level of maturity, even though a number of issues were still open (e.g., congestion control, audio and video codec selection, enhanced use of data channels).

In a subsequent work [2], Jennings *et al.*, focused on security challenges and transport issues, while presenting the solutions and mechanisms proposed within both the IETF and the W3C. They also identified congestion control as an open research question.

Other authors have focused on specific aspects of WebRTC, with special reference to security. Barnes and Thomson [3] provide a thorough description of the security threats associated with peer-to-peer web-based communications, and identify the WebRTC security architecture as a good candidate for the implementation of appli-

Salvatore Loreto is with Ericsson; Simon Pietro Romano is with University of Napoli Federico II.

0163-6804/17/$25.00 © 2017 IEEE

cations that can be secured from tampering by intermediaries. Similarly, Johnston *et al.* [4] discuss issues specific to WebRTC enterprise adoption by focusing on security, compliance, and interoperation.

The objective of this article is to provide an up-to-date view of the current status of standardization, while also identifying challenges that the standardization community will have to tackle once the first release of the WebRTC standards suite has been finalized. The WebRTC standard has in fact had to confront itself with both inner disputes and alternative views. Among the inner disputes we can cite the so-called "codec battle" between the supporters of two prominent candidates for the *Mandatory To Implement* (MTI) WebRTC video codec, namely H.264 and VP8. After an unsuccessful consensus call at IETF 88 (held in Vancouver in November 2013), such a battle ended up with the compromise decision of indicating both codecs as MTI for WebRTC. A further significant issue concerns WebRTC support within browsers. With respect to this particular topic, the current situation is that several browser vendors (Chrome, Firefox, Opera, Edge and Bowser) with differing completion scores,[1] are WebRTC-enabled. An important exception is currently represented by Safari. Apple, in fact, while closely following the standardization activities, has played no active role until now and their browser has no WebRTC capabilities.

Coming to the alternative views, since the beginning of 2014, a brand new initiative has seen the light in the W3C, the ORTC (Object Real-time Communications) Community Group. ORTC has indeed taken over from a previous initiative launched in mid 2013 and called ORCA (OBJECT-RTC API). Both ORCA and ORTC have initially been identified as alternatives to WebRTC. ORCA's explicit goal was to provide an alternative to the existing WebRTC API, aimed at allowing finer grained control to web developers willing to leverage real-time functionality within browsers. The same holds true for its successor ORTC, whose mission is to "define object-centric APIs to enable real-time communications in Web browsers, mobile endpoints, and servers".

Lately, the standardization community has agreed to converge to an agreed-upon solution for the first version of the standard by allowing the ORTC community to contribute to its finalization. At the same time, a common decision has been taken to adopt key concepts proposed with ORTC's low-level object API in the 'Next Version' of the standard, which nonetheless has backward compatibility with the 1.0 release among its foundational requirements.

This is exactly where the community stands now. A step away from completing WebRTC-1.0, with all minds already looking at the emerging initiative informally known as *WebRTC Next Version* (WebRTC-NV).

The WebRTC Architecture

WebRTC extends the classic web architecture semantics by introducing a peer-to-peer communication paradigm between browsers. The WebRTC architectural model draws its inspiration from the so-called SIP (Session Initiation Protocol) [5] Trapezoid. The most common WebRTC scenario is indeed one where both browsers are running the same web application, downloaded from the same application server. In this case, the Trapezoid becomes a Triangle, as shown in Fig. 1. Signaling messages are used to set up and terminate communications. They are transported by the HTTP or WebSocket protocol via the web server, which can modify, translate, or manage them as needed. It is worth noting that the signaling between browser and server is not standardized in WebRTC, as it is considered to be part of the application. As to the data path, the *PeerConnection* abstraction allows media to flow directly between browsers without any intervening servers.

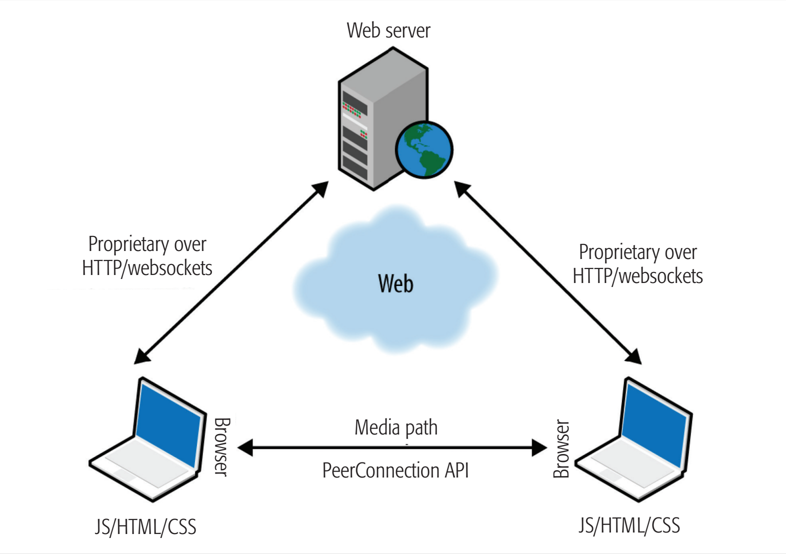


Figure 1. The WebRTC architecture.

A WebRTC web application is typically written as a mix of HTML and JavaScript. It interacts with web browsers through the standardized WebRTC API, as well as other standard APIs, allowing it to properly exploit and control the real-time browser function, both proactively (e.g., to query browser capabilities) and reactively (e.g., to receive browser-generated notifications). The WebRTC API must hence provide a wide set of functions, like connection management (in a peer-to-peer fashion), encoding/decoding capabilities negotiation, selection and control, media control, firewall and NAT element traversal.

Session description represents an important piece of information that needs to be exchanged. It specifies the transport information, as well as the media type, format, and all associated media configuration parameters needed to establish the media path. The IETF is now standardizing the JavaScript Session Establishment Protocol (JSEP) [6]. JSEP provides the interface needed by an application to deal with the negotiated local and remote session descriptions (with the negotiation carried out through whatever signaling mechanism might be desired), together with a standardized way of interacting with the ICE (interactive connectivity establishment) [7] state machine. The JSEP approach delegates entirely to the application the responsibility for driving the signaling state machine: the application must call the right APIs at the right times, and convert the session

[1] See http://iswebrtcreadyyet.com/ for an interesting summarizing table.

From the programmer's perspective, an important update to the WebRTC specification has been the introduction of Promises. Promises currently represent an advanced way for allowing asynchronous communication when using JavaScript. In a nutshell, they are similar to event listeners, but with a couple of fundamental improvements.

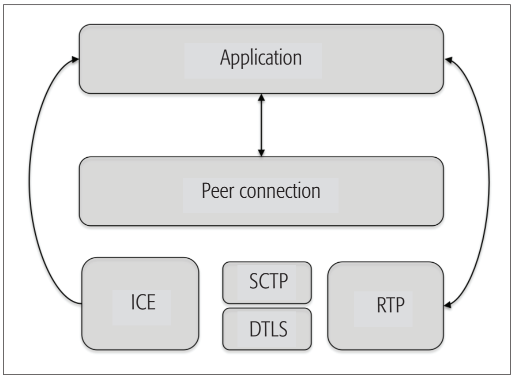


Figure 2. WebRTC: coarse-grained logical decomposition.

descriptions and related ICE information into the defined messages of its chosen signaling protocol.

It is worth mentioning that JSEP offers the possibility of manipulating session descriptions contained inside SDP (Session Description Protocol) messages. This happens within some limits (since browsers try to limit SDP "munging" to avoid disrupting communications) and at the developer's risk.

The W3C WebRTC-1.0 API allows a JavaScript application to take advantage of the novel browser's real-time capabilities. The real-time browser function implemented in the browser core provides the functionality needed to establish the necessary audio, video, and data channels. All media and data streams are encrypted using DTLS [8] (Datagram Transport Layer Security). DTLS is actually used for key derivation, while SRTP [9] (Secure Real-time Transport Protocol) is used on the wire. So, the audio and video packets on the wire are sent using SRTP. Data channel packets are handled by using SCTP [10] encapsulated in DTLS.

Figure 2 sketches, at a very high level, the current structure of the object oriented WebRTC framework. As anticipated, low-level components are for the most part indirectly controlled through the *PeerConnection* structure. Only a restricted form of direct control is allowed for ICE-related and RTP-related functionality. As shown in the figure, RTP allows for some form of control over the behavior of the protocol itself (e.g., for what concerns bandwidth capping). Coming to ICE, with the advent of WebRTC we have assisted to a renewed interest in such a protocol (as well as in its companion protocols STUN and TURN), as witnessed by the creation of the *tram* (TURN Revised and Modernized) working group within the IETF.

Identity Management in WebRTC

The WebRTC API also offers methods to enable verifying user identities. The solution decouples identity provision from communication providers via a third-party *identity provider* (IdP) (supporting a protocol such as OpenID or BrowserID) that can be used to demonstrate their identity to other parties. With this approach, trust between users is built by relying on an external entity [11].

This separation between identity provision and signaling is particularly important in federated scenarios (calls from one domain to another) and when calling via untrusted sites such as when two users who have a relationship via a given social network want to call each other via another, untrusted, site. The solution decouples the browser from any particular identity provider. The browser only needs to know how to load the IdP's JavaScript. Thus, a single browser can support any number of identity protocols. WebRTC offers and answers can in this way be authenticated by using the IdP. The entity sending an offer or answer acts as the *Authenticating Party* (AP) and obtains an identity assertion from the IdP, which it then attaches to the session description. The consumer of the session description acts as the *relying party* (RP) and verifies the assertion.

Toward a First Release of the Standard: WebRTC-1.0

In this section we will briefly discuss some relevant features that are going to be part of the WebRTC-1.0 specification. A non-exhaustive list of such features is reported in Table 1. For each item in the table, we provide a short description, as well as our estimation of its maturity level in terms of inclusion into the standard specifications. The following sections delve into some of the details associated with each of the reported features.

From Legacy JavaScript to ECMAScript Promises

From the programmer's perspective, an important update to the WebRTC specification has been the introduction of *Promises*. Promises currently represent an advanced way for allowing asynchronous communication when using JavaScript. In a nutshell, they are similar to event listeners, but with a couple of fundamental improvements. First, Promises can succeed or fail only once and they can never switch between success and failure states. Second, Promises can be associated with success and failure callbacks that are triggered independently from the exact time when the success/failure event has been raised. This allows applications to react to the outcome of an event rather than focusing on the exact time such an event took place.

All WebRTC-related APIs have lately been modified in order to move from the callback-based approach to the Promise-based approach, with the exception of the well known `navigator.getUserMedia()` method, which has been left unchanged for backward compatibility reasons.

From Streams to Tracks

The W3C *MediaStream* API specified by the "Media Capture and Streams" WG (and used within the WebRTC WG as one of its foundational blocks) has recently been modified in order to increase the level of granularity associated with the various media managed from within the browser. Namely, it has moved from streams to tracks. Streams have initially been interpreted as the most atomic data structure being transmitted over a *PeerConnection*. With the evolution of the specification, they have now been further described as collections of tracks. In summary, the current *MediaStream* objects represent synchronized streams of media that can be recorded or rendered in a media element. For example, a stream taken from camera and microphone

Feature	Function	Expected timeline
Promises	Use of ECMAScript promises in the API. No more callback-based methods exist	Certainly part of the WebRTC-1.0 spec
MediaStreamTrack objects	Allow developers to differentiate stream processing on a per-track basis	Certainly part of the WebRTC-1.0 spec
SDP bundling	Transmission of multiple media flows using a single 5-tuple	Details still under discussion, but most probably part of WebRTC-1.0
Codec priority reordering	Allow codecs to be reordered at the API level	Certainly part of the WebRTC-1.0 spec
RTCP multiplexing	Send both RTP and related RTCP data over a single port	Certainly part of the WebRTC-1.0 spec
Simulcasting	Send the same video stream at multiple resolutions and/or rates	Details still under discussion, but most probably part of WebRTC-1.0
Forward error correction (FEC)	Add redundancy to the encoded information and allow the receiver to compensate for partial data losses	Preliminary discussions ongoing (requirements draft under evaluation in RTCWEB)
Early media	Send media to the remote party before emitting an answer to an already received SDP offer	Certainly part of the WebRTC-1.0 spec
Screen sharing	Capture a user's screen and send it to a remote side in the form of a video stream	Details still under discussion, but certainly part of WebRTC-1.0

Table 1. WebRTC-1.0 Features and timeline.

Session negotiation is an important part of WebRTC. This calls into play the well-known Session Description Protocol (SDP). SDP provides multimedia applications with a standard means to describe a session, in terms of connectivity (i.e., IP addresses and ports), codecs, media attributes, and so on.

inputs has synchronized video and audio tracks representing synchronized streams of media. Each track is represented by a *MediaStreamTrack*. The main reason behind this increased granularity resides in the consideration that developers want to be capable of differentiating stream processing on a per-track basis, for example, to specify which codecs must be adopted, as well as the specific parameters used to configure such codecs. Some key transport properties can now also be set on a per track basis. To name just a few examples, we cite forward error correction (FEC), retransmission policy and bandwidth capping. All of the mentioned configuration actions are actually carried out by leveraging the brand new *RTCRtpSender* and *RTCRtpReceiver* interfaces, which allow applications to control how a given MediaStreamTrack is encoded/decoded and transmitted/received to/from a remote peer.

SDP "Bundling"

Session negotiation is an important part of WebRTC. This calls into play the well known *Session Description Protocol* (SDP). SDP provides multimedia applications with a standard means to describe a session, in terms of connectivity (i.e., IP addresses and ports), codecs, media attributes, and so on. As part of the SDP specification, it is possible to leverage a quite recent feature called *BUNDLE* [12], which refers to the transmission of multiple media flows (i.e., a 'bundle') using a single 5-tuple, that is to say, a single combination of a sending "IP address/port" pair, a receiving "IP address/port" pair, and a specific transport protocol (e.g., RTP). Within the context of WebRTC, the use of this technique has since the outset been encouraged, since it makes it possible to both save port numbers and reduce the number of ICE (Interactive Connectivity Establishment) protocol candidates. The latter point is particularly important since it dramatically reduces session setup time.

Bundling can be properly configured, at the API level, through an ad hoc defined parameter called *RTCConfiguration*, which contains, among other things, a property called `bundlePolicy`. Such a property can assume one of the following values: "Max-bundle", "Max-compat", or "Balanced."

The basic idea is that a WebRTC device will always try to use the bundle mechanism when negotiating a session with another peer. If the remote peer does not support bundle, then the aforementioned policy property comes into play. More precisely, "max-bundle" will instruct the WebRTC device to select a single media flow (among those that had to be bundled) and negotiate such a flow via SDP. If "max-compat" is selected, it will instead negotiate all of the flows separately, just as if bundle had never been introduced. This second approach is indeed the optimal one in case of backward compatibility with legacy (i.e., not aware of the bundle feature) devices. Finally, "balanced" refers to the intermediate approach of choosing two tracks (one audio track and one video track) to be negotiated separately via SDP.

Somehow related to the bundling mechanism is a further feature called "streams multiplexing," which is the possibility of adding multiple streams of the same type (either audio or video) to a single PeerConnection. BUNDLE indeed describes how to transmit/receive audio and video together, but does not explicitly deal with multiple instances of the same media type. This has been the subject of long discussions within RTCWEB, often referred

Simulcast is a relatively new function that draws inspiration from stream multiplexing, that is, a technique whereby a media source simultaneously sends multiple different encoded streams towards a specific destination, for example, the same video source encoded with different video encoder types or image resolutions.

to as the "Plan B vs. Unified Plan" debate, which eventually saw Unified Plan prevailing and being merged in the JSEP specification. The so called MSID (Media Stream Identification) draft [13] in the MMUSIC WG is targeted at allowing this to work, by specifying an SDP grouping mechanism for RTP media streams that can be used to indicate relations between media streams.

Playing with Codec Priority at the API Level

SDP makes it possible, among other things, to specify, for each media stream, the list of supported codecs. Upon session negotiation, the two peers agree on a set of codecs that is computed as the largest subset of common codecs signaled by the two parties. Such a subset is ordered as a list, and the first element is selected as the default codec to be used during the session. All other elements in the subset have to be supported by both parties (since they were advertised in the respective SDPs upon session setup time). Hence, the SDP specification allows for a peer to change codec during the session (provided that the new one belongs in the agreed-upon list of supported options) with no need to renegotiate the session itself.

Given this assumption, the WebRTC specification now makes it possible to programmatically select the desired codec for a PeerConnection with no need to edit the original SDP. More precisely, the API currently makes it possible to:

- Gain access to the bundle of parameters associated with an RTP sender (through the `RTCRtpSender.getParameters()` method).
- Select, within such a structure, the "codecs" property, which is basically an array of supported codecs related to that sender.
- Reorder (or even remove) information contained in the codecs list.
- Commit changes to the RTP sender object (through the `setParameters()` method).

RTCP Multiplexing

The standard way of streaming real-time media across the Internet envisages the use of RTP (Real-time Transport Protocol) for application-level framing of media samples, in conjunction with the companion RTCP protocol used to carry both feedback and minimal session control information back and forth between the two peers. Usually, RTP and RTCP are associated with different ports (e.g., if $2n$ is an even port used for RTP, then $2n + 1$ will be an odd port associated with RTCP control information). With RTCP multiplexing (also known as RTCP MUX), we refer to a way of sending both RTP and related RTCP data over a single port. The idea of leveraging such a function is, once again, to both save allocated port numbers and reduce ICE setup time.

After a good deal of discussions on whether or not to specify RTCP MUX support as optional for WebRTC, there currently seems to be consensus around making it mandatory at least in those cases in which the peers are also using SDP bundle. At recent IETF meetings, a further step was done along the same lines and two major WebRTC browser vendors (namely, Google and Firefox) have clearly stated their will to allow WebRTC endpoints to simply reject legacy (i.e., non multiplexed) RTCP sessions. This resolution, while simplifying things a lot for WebRTC-capable devices, clearly calls for the introduction of a proxying function (provided by some sort of WebRTC gateway intervening along the data path) if the need arises to interact with any legacy application still relying on two different ports for RTP and RTCP.

Simulcasting

Simulcast is a relatively new function that draws inspiration from stream multiplexing, that is, a technique whereby a media source simultaneously sends multiple different encoded streams toward a specific destination, for example, the same video source encoded with different video encoder types or image resolutions. It can be somehow associated with Scalable Video Coding (SVC), namely the mechanisms by which a single encoded video stream can be organized in layers and each participant is allowed to receive (and decode) only the layers that they are able to process. The WebRTC community has long since identified a number of use cases for simulcast. One interesting example is represented by conferencing scenarios involving the presence of a so-called *selective forwarding unit* (SFU). In the mentioned scenario, the clients send to the SFU (which is acting as a conference focus) multiple video streams, each associated with exactly the same scene, but at different resolutions. The SFU can hence properly select the specific incoming stream that has to be forwarded to the other participants. As an example, the SFU might forward a high resolution version of the stream only when the client in question is playing an active role in the conference (e.g., they are currently holding the floor), while relying on the lower resolution version while they are not actively participating in the discussion. Other, more complex, forwarding choices can obviously be applied once the general mechanism described above is available. Just to cite one, the SFU might let the choice depend on considerations associated with optimizing overall bandwidth consumption, while at the same time offering a good-enough service to the end-users in terms of quality of experience (QoE).

Coming to the technical details, until recently, there has been a lack of uniformity in the way simulcasting has been deployed in the wild. The basic mechanism leveraged by all implementations is represented by the insertion of multiple *m* (i.e., media) lines of the same media type (e.g., audio, video, and so on) inside the SDP body. What was lacking in this case was a means to signal to the other party that those m-lines were indeed all associated with a single source. A recent proposal from Google seems to have filled exactly this gap and has gained consensus within the IETF community. In a nutshell, the idea is to add a new identifier in SDP, namely a source stream identifier, that can be leveraged to differentiate sets of media attribute lines.

As a result of this approach, the W3C has allowed some form of manipulation of simulcast streams at the API level. More precisely, within the context of the newly defined *RTCRtpTransceiver* interface (which is basically a combination of an *RtpSender* and an *RtpReceiver* associated with the same SDP media identifier) it is possible to refer to a property called "rid," which is nothing but a copy of the above mentioned source stream identifier. This structure, combined with a new feature called "scaleDownResolutionBy" indicating a scaling down factor relative to the maximum resolution available for the stream, allows the developer to explicitly choose the desired quality of a signaled simulcast stream.

Forward Error Correction

One interesting topic of discussion at recent IETF meetings has been the introduction (and configuration) of *forward error correction* (FEC) [14] capabilities inside WebRTC endpoints. Opus, which is one of the "MTI" (mandatory to implement) codecs for audio, does provide in-band support for it.

FEC is a generic mechanism for the protection of media streams against packet corruption due, for example, to the presence of one or more lossy links along the end-to-end communication path. It adds some level of redundancy inside the encoded information, so to allow the receiving peer to properly compensate for partial data loss with no need for retransmissions.

As it always happens when redundant encoding is introduced, the advanced reconstruction capabilities at the receiving side are paid in terms of increased network overhead. Hence, the challenge in these cases is to try to strike an optimal balance between robustness to packet corruptions and increased bandwidth consumption. This holds particularly true in all those cases in which the network does not provide any form of congestion control. In such cases, indeed, the issue is *congestion* rather than *lossy communication*, and the use of FEC can only make things worse as it contributes to increasing congestion due to the overhead it unavoidably introduces.

Within the standardization community, work is currently in progress in order to allow WebRTC implementations to fine-tune the configuration of FEC parameters (as allowed by the RTP specification), to enforce a fair behavior on the side of the applications. At recent meetings there has also been some preliminary discussion on whether or not to allow such tuning knobs to surface at the JavaScript API level.

With reference to congestion control, it is instead worth mentioning the ongoing work within both the AVTCORE and RMCAT Working Groups within the IETF, with special regard to the so called Circuit Breakers [15] document, which is soon to become an RFC.

Allowing Early Media

Early media is a well-known term in VoIP networks, referring to the capability of sending some media to the other party before emitting an answer to an already received SDP offer. While this might seem awkward, it is a very useful mechanism that real-time applications are used to leverage in order to provide an enriched end-user experience through, for example, playing music while the user is waiting for a call to be connected.

WebRTC has since long looked at early media as a desired functionality, both to seamlessly interact with legacy VoIP applications that already rely on it and to bring its benefits to the WebRTC ecosystem itself. Recently, this function has been standardized. More precisely, it has been specified that an end-point that receives media before getting the answer to its own offer can accept such media provided that:

- It is consistent with the emitted SDP offer (in terms of codecs and other media attributes).
- The end-point in question (i.e., the emitter of the SDP offer) has already created an instance of the *RTCRtpReceiver* object that is to be associated, upon successful completion of the session setup procedures, with the incoming media stream.

The mentioned requirements have been provided through minor modifications to the WebRTC-1.0 specification. Fundamentally, a change was made as to when tracks are created for the offerer. This can now happen either as a result of a call to the `setLocalDescription` method, or as soon as media packets are received. The mentioned modifications ensure that these objects can be created and connected to media elements for play-out when needed. Without digging in too much detail, we just mention as a side note that, in order to prevent potential security breaches, early media cannot happen 'earlier' than the remote DTLS (Datagram Transport Layer Security) fingerprint has been received.

Screen Sharing

Within the context of WebRTC, screen sharing refers to the capability of capturing a user's screen (all or in part) and sending it to a remote side (across a PeerConnection) in the form of a video stream. Such a function leverages an ad hoc defined extension to the *Media Capture API*, which defines a new method called `getDisplayMedia`. Such a method allows for the acquisition of different types of captures, in terms of both the "portion" of the screen one is interested in sharing and the type of display "surface." With respect to this last term, a distinction is made between a *logical* surface and a *visible* one. The former refers to an entire application window, independently from the fact that part of such a window might be covered by another application's window; the latter is instead associated with the part of the window that is visible on the user's side, that is, that is not covered by any other window that is not being shared. As to the portion of the screen that is going to be shared, the following choices are available:

- *Monitor:* one or more physical displays (connected to a user's computer).
- *Window:* a single application window.
- Application: all of the windows associated with a specific application.
- *Browser:* a single browser window (or *Tab*).

Inherently, screen sharing poses a number of security and privacy concerns. The most intuitive risk is related to the fact that users might inadvertently share content that they did not wish to share. A less obvious risk is also associated with display capture. Namely, this new function might weaken the cross site request forgery protections that should be guaranteed by the browser sandbox. As an example, sharing of a window containing a canvas might circumvent standard controls on such an object that do not allow sampling or even conversion to any accessible form if it is not "origin-clean."

Within the standardization community, work is currently in progress in order to allow WebRTC implementations to fine-tune the configuration of FEC parameters (as allowed by the RTP specification), so to enforce a fair behavior on the side of the applications. There has also been preliminary discussion on whether or not to allow such tuning knobs to surface at the JavaScript API level.

A further reason why ORTC supporters proposed a lower-level API concerns the implementation of advanced functionality like simulcasting and Scalable Video Coding (SVC), which both benefit from the possibility of gaining direct access to the basic building blocks of the media pipeline.

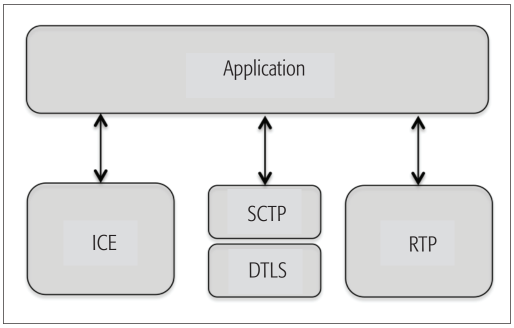


Figure 3. The ORTC architecture.

This and other related issues are currently under discussion within the RTCWEB working group, which has taken at the outset the responsibility of defining the overall security architecture for web real-time communications. With respect to the aforementioned cross-origin protection capabilities, it is strongly advised that users are asked to exhibit elevated permissions before being allowed to access any available display surface.

Bringing ORTC Concepts into WebRTC

Seminal work behind *Object Real-Time Communications* (ORTC) stemmed from the consideration that the SDP-based offer/answer paradigm embraced by the WebRTC API did not fit well the emerging real-time communication models (with special reference to peer-to-peer systems). The core of ORTC is represented by a JavaScript API designed within the ORTC W3C Community Group. Such an API aims at offering finer-grained control over how a real-time web application is implemented, by exposing to the surface most of the objects that the standard WebRTC API typically controls as a single pipelined unit of elaboration through a higher-level configuration interface. Since the outset, the idea has been to allow the coexistence between the SDP-based Offer/Answer approach proposed by WebRTC and the low-level ORTC API. This is achieved thanks to the superposition, on top of the ORTC API, of a WebRTC-compliant *shim* library. With this approach, programmers can choose between ORTC-style raw control of the real-time communications engine on one side and WebRTC-style SDP-based negotiation on the other.

A rough comparison between Figs. 2 and 3 allows us to highlight the major difference between the WebRTC and the ORTC approach. Namely, the two models work, at the lowest layer, with the same set of objects. WebRTC-1.0 relies on the *PeerConnection* abstraction as a glueing component that somehow orchestrates the overall behavior of a peer. ORTC, on the other hand, allows the programmer to gain full direct control over the set of available objects and optionally enables the use of the PeerConnection as an API facility that is provided through the above mentioned shim adapter library.

Based on the considerations above, it is fair to claim that ORTC is not to be considered as a competitor to WebRTC. Full compatibility with the WebRTC-1.0 API is guaranteed by the development of the aforementioned SDP-based JavaScript shim on top of ORTC. Such a library takes on the responsibility of ensuring that SDP parsing and negotiation features are identical and work on top of the ORTC primitives. Compatibility is to be thoroughly checked via unit testing procedures. This is expected to foster interoperability among heterogeneous implementations. A further reason why ORTC supporters proposed a lower-level API concerns the implementation of advanced functionality like *simulcasting* and *scalable video coding* (SVC), which both benefit from the possibility of gaining direct access to the basic building blocks of the media pipeline.

It is important to stress the consideration that, since its foundation as a W3C community group, ORTC has never been really conceived as a competitor to WebRTC. As already anticipated, it has rather been seen as an alternative, yet compliant, approach that can be leveraged by those developers who are targeting scenarios different than the "standard" Offer/Answer based ones. The efforts that have been devoted to the design of the shim library allowing for the seamless operation of a WebRTC application on top of the pipeline-based ORTC framework can indeed be seen as a real added value to the overall WebRTC ecosystem.

The above statement is so true that during a recent WebRTC charter renewal process, key representatives of the ORTC community group have been formally invited to join the WebRTC effort. More precisely, one of the founders of the ORTC initiative has joined the WebRTC chairs, while another ORTC representative has become a member of the WebRTC-1.0 editing team. It has also been decided that all future standardization work in WebRTC will take place within the WebRTC Working Group, while the ORTC community group will fade away and its contributors will join the WebRTC effort. Finally, once done with the WebRTC-1.0 milestone, all energies will be devoted to a brand new initiative called WebRTC-NV, as discussed in the next section.

Discussion and Directions of Future Work

In this article we presented the current state of the art in the field of standardization of web-based real-time communications. We focused on the upcoming new standard known as WebRTC-1.0, by briefly describing both the genesis of this challenging initiative and its evolution toward an agreed upon final specification. We also discussed in some detail the relationship between WebRTC-1.0 and the companion initiative known as *Object Real Time Communications* (ORTC), which has brought a new perspective on how to properly look at and manage the entire media pipeline associated with real-time interaction among web-based devices. Finally, we have highlighted how the two initiatives have eventually converged into a unified effort that has contributed to finalizing the WebRTC-1.0 specification.

The term *WebRTC-NV* refers to the upcoming 'next version' of the WebRTC standard, which has been on purpose called neither WebRTC-1.1 (as proposed by those who are in favor of applying only minor changes to the current spec) nor WebRTC-2.0 (indicating a major departure from the agreed-upon 1.0 version). At the time of this writing, there is indeed no official decision about the direction that will be followed for this new initiative. Unofficial rumors state that the NV initiative will continue to work on ORTC-style low-level controls while maintaining interoperability with WebRTC-1.0. This means that the most important building blocks of the WebRTC-1.0 architecture (SRTP, RTCP, SCTP over DTLS, and so on) will be supported. Similarly to ORTC, SDP support will not be mandatory at all, and the proposed API will offer direct control over the various components of the media pipeline. Apart from this basic set of requirements, discussions are still ongoing as to whether or not the scope of the working group should be expanded in order to cover all or some of the hot topics we mentioned in the article, for example, simulcast, Scalable Video Coding, Forward Error Correction. Finally, contributors will continue to focus on security and privacy as key areas of interest for the working group.

It has been decided that all future standardization work in WebRTC will take place within the WebRTC Working Group, while the ORTC community group will fade away and its contributors will join the WebRTC effort. Finally, once done with the WebRTC-1.0 milestone, all energies will be devoted to a brand new initiative called WebRTC-NV.

References

[1] S. Loreto and S. P. Romano, "Real-Time Communications in the Web: Issues, Achievements, and Ongoing Standardization Efforts," *IEEE Internet Computing*, vol. 16, no. 5, Sept.-Oct. 2012, pp. 68–73, DOI: 10.1109/MIC.2012.115

[2] C. Jennings, T. Hardie, and M. Westerlund, "Real-time Communications for the Web," *IEEE Commun. Mag.*, vol. 51, no. 4, April 2013, pp. 20–26, DOI: 10.1109/MCOM.2013.6495756

[3] R. L. Barnes and M. Thomson, "Browser-to-Browser Security Assurances for WebRTC," *IEEE Internet Computing*, vol. 18, no. 6, Nov.-Dec. 2014, pp. 11–17, DOI: 10.1109/MIC.2014.106.

[4] A. Johnston, J. Yoakum, and K. Singh, "Taking on WebRTC in an Enterprise," *IEEE Commun. Mag.*, vol. 51, no. 4, April 2013, pp. 48–54, DOI: 10.1109/MCOM.2013.6495760

[5] J. Rosenberg *et al.*, "SIP: Session Initiation Protocol," Request for Comments (RFC) 3261, Internet Engineering Task Force (IETF).

[6] J. Uberti, C. Jennings, and E. Rescorla, "Javascript Session Establishment Protocol," Internet Draft (work in progress), draft-ietf-rtcweb-jsep-17.txt, expires: April 24, 2017, Internet Engineering Task Force (IETF).

[7] J. Rosenberg, "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols," Request for Comments (RFC) 5245, Internet Engineering Task Force (IETF).

[8] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security Version 1.2," Request for Comments (RFC) 6347, Internet Engineering Task Force (IETF).

[9] M. Baugher *et al.*, "The Secure Real-time Transport Protocol (SRTP)," Request for Comments (RFC) 3711, Internet Engineering Task Force (IETF).

[10] R. Stewart (Ed.), "Stream Control Transmission Protocol," Request for Comments (RFC) 4960, Internet Engineering Task Force (IETF).

[11] E. Rescorla, "WebRTC Security Architecture," Internet Draft (work in progress), draft-ietf-rtcweb-security-arch-12.txt, expires: Dec. 10, 2016, Internet Engineering Task Force (IETF).

[12] C. Holmberg, H. Alvestrand, and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)," Internet Draft (work in progress), draft-ietf-mmusic-sdp-bundle-negotiation-36.txt, expires: Apr. 30, 2017, Internet Engineering Task Force (IETF)

[13] H. Alvestrand, "WebRTC MediaStream Identification in the Session Description Protocol," Internet Draft (work in progress), draft-ietf-mmusic-msid-15.txt, expires: Jan. 8, 2017, Internet Engineering Task Force (IETF)

[14] S. Holmer, M. Shemer, and M. Paniconi, "Handling Packet Loss in WebRTC," *2013 IEEE Int'l. Conf. Image Processing*, Melbourne, VIC, 2013, pp. 1860–64, DOI: 10.1109/ICIP.2013.6738383

[15] C. Perkins and V. Singh, "Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions," Internet Draft (work in progress), draft-ietf-avtcore-rtp-circuit-breakers-18.txt, expires: February 19, 2017, Internet Engineering Task Force (IETF).

Biographies

Simon Pietro Romano is an associate professor in the Department of Electrical Engineering and Information Technology (DIETI) at the University of Napoli. He teaches computer networks, network security, and telematics applications. He is also the co-founder of Meetecho, a startup and University spin-off dealing with WebRTC-based unified collaboration. He actively participates in IETF standardization activities, mainly in the applications and real time (ART) area.

Salvatore Loreto works for Ericsson Research in Sweden. He is product manager for the MediaFirst Video Delivery end to end solution, the Operator holistic CDN solution including cloud services like transcoding, repackaging and storage. He is also driving and executing the strategy evolution and technology roadmap toward the 5G networks of Media Delivery business line. He works on standardization, both as an IETF working group chair and as an active participant, mainly in the applications and real time (ART) area.

Service-Centric Networking for Distributed Heterogeneous Clouds

Pieter Simoens, David Griffin, Elisa Maini, T. Khoa Phan, Miguel Rio, Luc Vermoesen, Frederik Vandeputte, Folker Schamel, and Dariusz Burstzynowski

The authors first posit that from an analysis of a snapshot of today's centralized and regional data centre infrastructure, there is a sufficient number of candidate sites for deploying many services while meeting latency and bandwidth constraints. They then provide quantitative arguments why both network and hardware performance needs to be taken into account when selecting candidate sites to deploy a given service.

Abstract

Optimal placement and selection of service instances in a distributed heterogeneous cloud is a complex trade-off between application requirements and resource capabilities that requires detailed information on the service, infrastructure constraints, and the underlying IP network. In this article we first posit that from an analysis of a snapshot of today's centralized and regional data center infrastructure, there is a sufficient number of candidate sites for deploying many services while meeting latency and bandwidth constraints. We then provide quantitative arguments why both network and hardware performance needs to be taken into account when selecting candidate sites to deploy a given service. Finally, we propose a novel architectural solution for service-centric networking. The resulting system exploits the availability of fine-grained execution nodes across the Internet and uses knowledge of available computational and network resources for deploying, replicating and selecting instances to optimize quality of experience for a wide range of services.

Interactive Demanding Services in the Cloud

There is vast diversity in cloud-hosted services today, ranging from mobile back-ends, over virtualized set-top boxes and gaming consoles to real-time services providing decision and control support for self-driving cars. These recent cloud services require a crisp experience and/or real-time processing of high data rate streams. High network delays and low throughput to a relatively small number of centralized remote data centers (DCs) may have a serious impact on the quality of experience (QoE). For instance, 30 percent of the US population has a too high latency to one of Amazon's EC2 DCs for cloud-based gaming [1]. Deploying such applications in distributed execution platforms closer to the users reduces network delays and is also the preferred approach for many data intensive applications. Shifting all the data to a centralized service could overwhelm the network, and it is better to bring the computation logic closer to data sources and users at the network edge. As of today, Internet service providers (ISPs) already deploy content delivery network (CDN) proxy servers in their network to save on transit costs and improve the quality of service for their customers [2].

Service developers are thus confronted with the twofold challenge of service instance placement and selection. The central problem in service placement is to determine the cost-optimal set of geo-distributed datacenters where to deploy an instance, and to configure the appropriate scaling policies in each of these datacenters to adequately cope with the expected demand. These distributed nodes have heterogeneous hardware, as they are owned by different entities or deployed at different moments in time. Service instance selection refers to the anycast-style resolution of a service identifier to the network endpoint of the best replica, taking into account service availability, network metrics, and the location of the requesting user.

Service placement and instance selection in distributed clouds are best performed on the grounds of both network and service performance metrics. However, this knowledge is distributed among different business entities in the value chain of application delivery, such as infrastructure providers, ISPs and service developers, and is highly impacted by the specific service requirements as well as the characteristics of the underlying heterogeneous cloud infrastructure. Misaligned objectives and incomplete visibility on policies due to IPR protection mechanisms can lead to suboptimal decisions in terms of service performance and deployment cost [3].

In this article, we introduce the concept of service-centric networking (SCN) as a framework that holistically addresses both service and network aspects when providing functionality for service resolution and placement in a distributed and heterogeneous cloud environment.

The remainder of this article is structured as follows. First we discuss existing frameworks enabling collaboration between ISPs and service providers and for distributed service management. We then focus on the need for close cooperation with the ISP in selecting service instances based on performance and bandwidth/cost grounds, as well as on the importance of DC capabilities being part of the service placement optimization

This work was supported in part by the European Commission through the FUSION project under grant agreement no. 318205.

Digital Object Identifier: 10.1109/MCOM.2017.1600412

Pieter Simoens is with Ghent University and imec; David Griffin, Elisa Maini, T. Khoa Pan and Miguel Rio are with University College London; Dariusz Bursztynowski is with Orange Polska Labs and Warsaw University of Technology; Frederik Vandeputte and Luc Vermoesen are with Nokia Bell Labs; Folker Schamel is with Spinor GmbH.

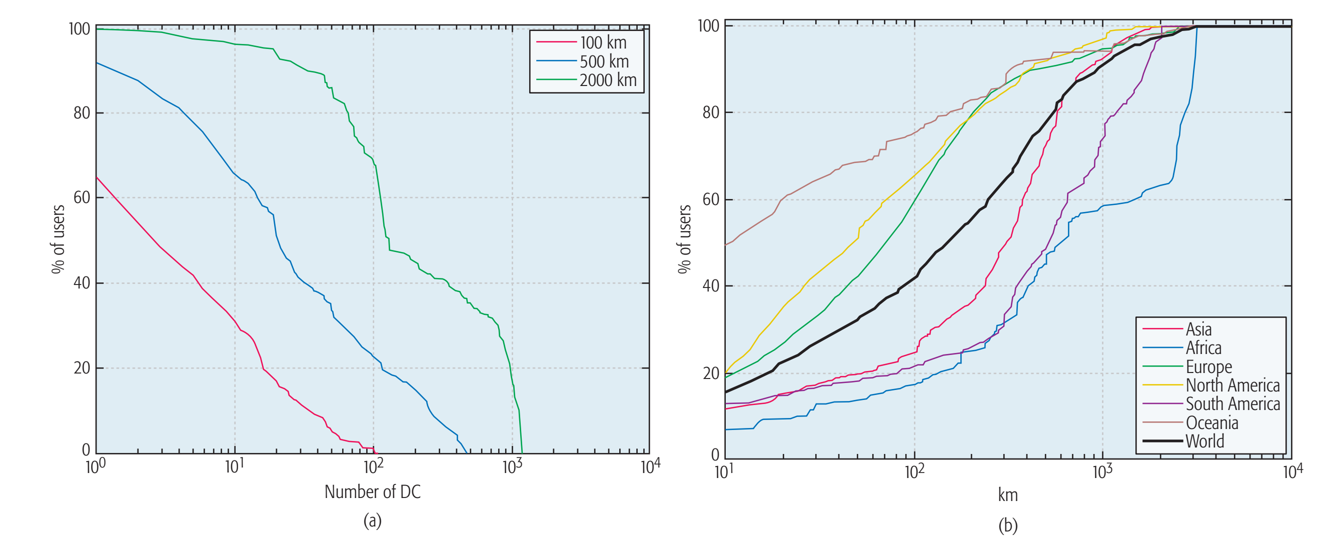


Figure 1. Characterization of the geographical distance between users and DCs worldwide: a) CCDF of number of DCs available within radii of 100 km, 500 km, and 2000 km for all users worldwide; b) CDF of the distance of the fifth closest DC for all users, split by continent and for the global population.

problem. In the last part of the article, we introduce the SCN architecture and its primitives for capability and performance awareness.

Related Concepts

CDNs cache content closer to the user to reduce traffic in interconnection links, and to provide higher downloading speed and lower access delays. CDN typically uses domain name system-based resolution to select the appropriate server. End-user mislocations and the limited view of network bottlenecks have been major drivers for CDN-ISP collaboration to improve server selection and enable on-demand negotiation of CDN surrogates on ISP-owned datacenters [2]. CDNs are often combined with application delivery networks (ADNs) consisting of controllers deployed in datacenters that reduce the service load through load balancing or performing application accelerations such as image transcoding or SSL offload. ADN middleboxes are over-the-top (OTT) proprietary solutions that optimize the service load, but they are black boxes to the ISP. Only the largest enterprises can carry the extensive costs of operating a private WAN that connects geo-distributed datacenters and peers with user ISPs [4].

CDNs and ADNs provide partial solutions to the targeted problems by SCN. CDNs choose between cached content replicas for lower network delays, while SCN also accounts for service-level performance information and service availability. SCN fills the gaps in network-wide service orchestration and introduces service resolution to provide intersection with traffic engineering in transport network and data centers.

Existing research on service resource allocation in geo-distributed clouds can be broadly categorized into approaches that place services in order to minimize latency [5], and approaches that instead focus on (re)placing service instances driven by variations in demand and infrastructure cost [6, 7]. The SCN primitives also account for ISP traffic optimization, service-specific performance metrics, and cloud heterogeneity.

Several distributed service management architectures have been proposed. IRMOS [8] relies on strict QoS guarantees between service components so it fits best to managed networks and needs adoptions for wide area Internet. NGSON is an IEEE standardized overlay framework [9] that provides the means to flexibly interconnect existing deployed services but does not account for service placement and provisioning, scaling, and heterogeneous virtualized capabilities.

While the integration of CDNs, ADNs, NGSON and other known solutions is possible at a conceptual level, it is hard to just take existing technologies in order to achieve the goals of SCN. The most important missing parts are network-wide service orchestration and support for the implementation and propagation of network policies to allow service resolution, taking account of server load, DC resources, and network costs and conditions. The SCN approach is holistic in addressing these problems, and provides additional functionalities oriented to recent evolutions in cloud hardware heterogeneity and lightweight virtualization.

Latency to Distributed DCs

It is often claimed data processing capabilities located at the extreme network edge are required to provide low-latency services. The realization of this edge computing paradigm obviously entails significant capital and operating expenses to ISPs. However, our studies show that the already existing DCs may provide sufficient performance to deliver many high-performance applications, such as cloud gaming, to the vast majority of users worldwide.

We calculated the haversine distance from all cities worldwide listed in the geonames.org database to the address of 3116 DCs identified at www.datacentermap.com. Figure 1a shows the CCDF of the number of DCs within radii of 100 km, 500 km and 2000 km for all users. Network latency, in terms of round-trip-time, can be estimated from haversine distance using a conversion factor of approximately 55km/ms, as determined by the

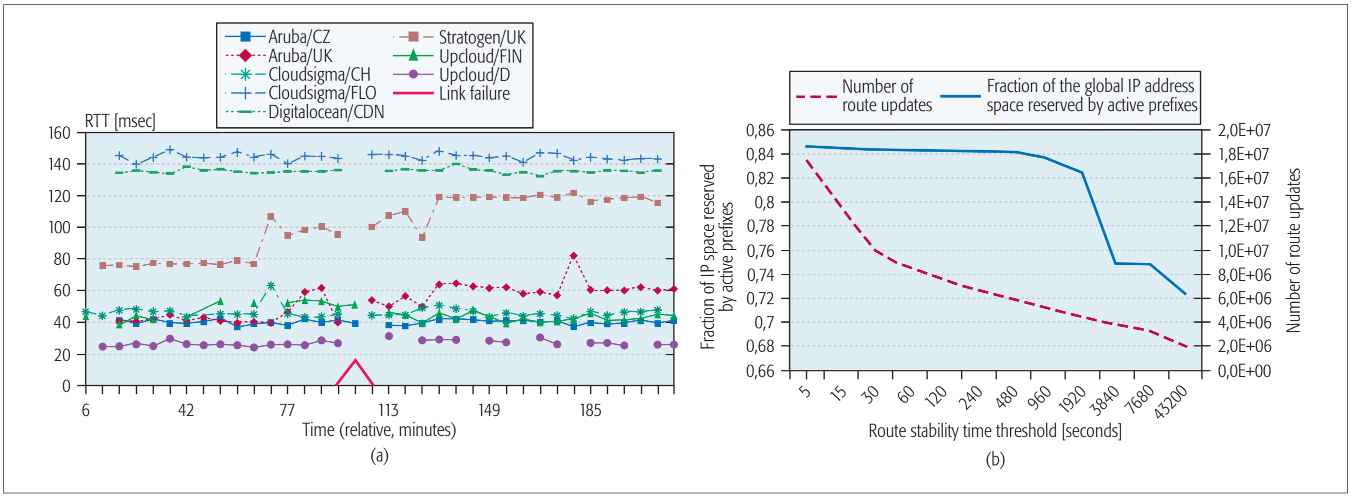


Figure 2. Network and routing statistics from Orange Poland: a) impact of a BGP event on the end-to-end latency to DCs worldwide. The event was observed by the Orange Poland network on Jan 26, 2016 at 12:34:53 CET; b) number of route updates and the fraction of the active IP address space as a function of the minimum time between consecutive route updates, measured from Orange Poland network in the period Jan. 8–Feb. 8 2016.

analysis of global Internet traffic [10]. This conversion factor accounts for queuing delays in intermediate switches and routers. Our model shows that 100 percent of users can reach at least one DC within ~36 ms (2000 km), and ~65 percent of all users can reach a DC within ~2 ms (100 km). It should be noted that this model assumes the best case for access network latency; for higher-latency access networks, the RTT figures should be increased accordingly.

Figure 1b shows the CDF of the fifth closest DC to all users worldwide and per continent. This indicates that for 90 percent of users there is a choice of five or more DCs within 1000 km (~15 ms RTT) for provisioning services.

For 5T tactile services with a response time of 1 ms or less [11], the existing DCs may indeed not be sufficient, and additional micro-DCs within ISP-provided locations may be required to keep latency below 10 ms. On the other hand, latency-tolerant services, such as document editing, can be deployed in a handful of centralized locations. However, even for latency-tolerant services it might be appropriate to deploy replicas in more locations, especially when they are bandwidth-hungry, such as remote video processing or large-scale data analysis. A distributed deployment closer to users and data sources can drastically reduce bandwidth costs.

For the majority of applications that lie between these two extremes and require a response within 30 ms to 100 ms, including audio-visual applications such as video conferencing and cloud gaming, a deployment in a number of the existing DCs is sufficient to meet performance requirements. Service placement optimization is required in order to select the minimum number of locations to run services, and hence reduce cost, while ensuring that the selected DCs are within tolerable performance limits. In addition to network metrics, the infrastructural aspects of the DCs also impact the service placement. We will discuss these in a later section, but we will first study the added value of the ISPs' knowledge of network metrics in placement and resolution.

Network-Aware Service Placement and Resolution

Commercial solutions such as Cedexis or CloudHarmony provide benchmarks of CDNs and cloud providers worldwide on end-to-end network metrics such as latency, jitter, and throughput. Statistics are crowdsourced in an over-the-top manner, by clients accessing HTTP pages with embedded scripts to measure network statistics to selected sites. The accuracy and timeliness of these datasets depends directly on the number of participating clients. ISPs, on the other hand, have a detailed insight into the performance of their own network, and on the BGP routing topology toward other autonomous systems (AS). This inter-AS routing is subject to changes (e.g. due to link failures) and traffic routing policies. A key question is thus whether OTT measurement methods are sufficient for taking resolution decisions or whether this role is better assumed by the ISP.

We measured every six minutes the RTT to 209 DCs worldwide from the Orange Poland network in the period Jan 8–Feb 8 , 2016. Each measurement consisted of downloading 12 times a Javascript that only contains an empty method, and taking the average of only the last 10 downloads to exclude warming-up effects.

We correlated these application-layer latency results to the directly observed changes in BGP inter-domain routing by the ISP. Figure 2a visualizes the impact of a link failure between the Orange Poland network and a Tier-1 network on the end-to-end delay between our probe and a subset of the DCs.

Link failures introduce a storm of BGP updates. After convergence of the BGP rerouting, the RTT of about 10 percent of monitored sites located in Europe and other continents (for the sake of visibility, only a subset are included in the figure) stabilizes on a new value. Although for most DCs the latency observed after BGP convergence does not differ noticeably from before the failure, there is still an impact in terms of lost connectivity: the gaps in the figure correspond to failed measurements during the connectivity downtime.

The period of broken connectivity extends for several minutes, which can have a negative impact on the QoE. Such interruptions can only be detected by OTT probes if measurements are taken very frequently and there are sufficient users in each AS crowdsourcing data. Real-time monitoring of BGP route updates is therefore a more scalable and practical proposition to detect interruptions quickly and to increase the responsiveness to changes in network conditions.

The next question is then how often such BGP route updates occur over time, and how much of the forwarding entries in the routing table are affected. Figure 2b provides insight into the scale of this phenomenon. The dashed plot describes the total number of route updates (forwarding entry changes) during the observation period (one month) such that the time elapsed from the previous update for a given prefix was not less than a given value. We note every such "active" prefix involves a set of IP addresses. Accordingly, the solid line shows the fraction of the IPv4 address space that corresponds to the route updates described by the dashed line.

The general conclusion from this analysis is that BGP route changes are observed for a large portion of the IP address space and over a wide range of time scales, and that BGP route updates are a quick indicator of changes in network performance between end-users and DCs. Although BGP updates could in principle be monitored and processed by non-ISP third parties, this requires probes deployed in various vantage points around the globe. The quantity of information to be processed by OTT providers would easily become prohibitive: BGP route updates observed at different locations must be correlated and the impact on users from each AS must be calculated, which is a complex process considering that BGP changes in a single AS cause a high rate of globally propagated updates. Moreover, ISPs are unlikely to expose the full details of their peering, transit, and uplink connections with third parties, meaning that this information must be indirectly inferred by OTT parties.

In summary, if resolution decisions are made by OTT service providers, they require a significant overhead in terms of network monitoring infrastructure, and the result may be sub-optimal from the perspective of traffic costs to the network operators. ISPs are in a privileged position to make service resolution decisions due to the efficiency and accuracy of direct access to network performance information from the perspective of their users, with the added benefit of being able take network costs into account.

Participating in service resolution decisions has several other advantages to ISPs, in particular to reduce traffic cost. Service replicas will be located in a range of DCs, and the routing paths to those in remote ASs will be over peering and transit links with different monetary costs to the ISP. The ISP is thus able to select service replicas with an appropriate trade-off between service utility and network costs to ensure QoE within acceptable traffic costs for the network operator.

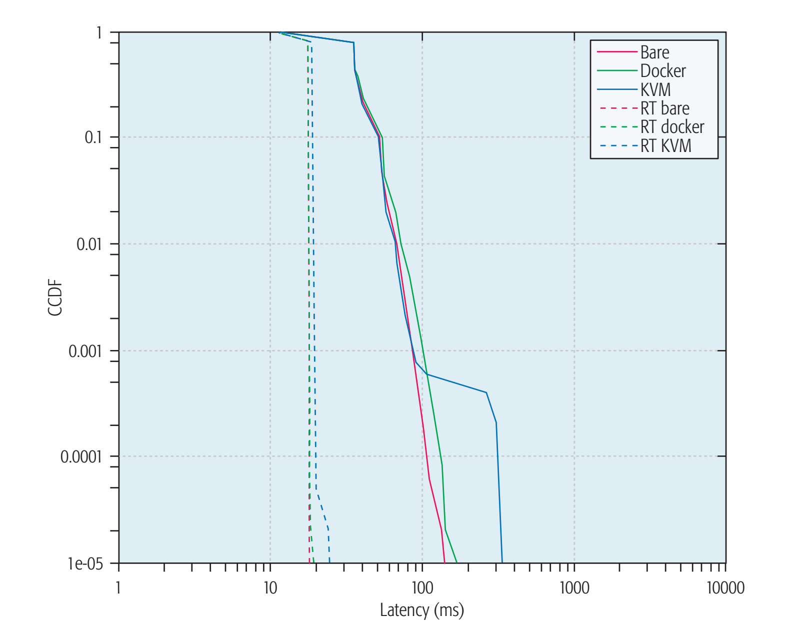


Figure 3. Latency to produce a single 720p video frame. The experiments were conducted on a SuperMicro server blade, with a dual Opteron 6174 CPU and 64 GiB RAM. Full lines: average CCDF of 48 instances with best-effort CPU scheduling of the vanilla Linux kernel. Dashed lines: CCDF for a single instance that was attributed a higher CPU scheduling class, while the other 47 instances were scheduled best-effort.

Performance Variations In Heterogeneous Clouds

Network metrics are not the only factors to be considered in service placement. Demanding services often have specific hardware/software resource and performance requirements to deliver a consistent QoS. For example, media services may depend on certain GPU features such as specific OpenGL extensions, or vendor-specific APIs such as NVIDIA CUDA support.

However, even with identical hardware we can observe huge performance differences across DCs, owing to the configuration and management policies of the infrastructure provider. For economic reasons, infrastructure providers will co-locate many workloads on the same node, balancing resource isolation policies with resource oversubscription, thereby assuming that not all concurrently running applications need their full capacity at the same time.

To demonstrate the impact of resource isolation policies on service performance, we have measured the latency of a media encoding application for producing a single frame in a 720p video stream. Targeting a frame rate of 25 fps, this latency should be kept below 40 ms. We deployed 48 application replicas on bare metal, in a VM managed by the KVM hypervisor, in a Docker container and on bare metal with NUMA-aware placement.

The CCDF plots in Fig. 3 show the probability that the time to produce a single frame exceeds a given latency. The full lines report the average performance of the 48 instances, using the default best-effort settings for CPU isolation of a vanilla Linux kernel. The dashed line indicates the same metric for one instance that was configured with a

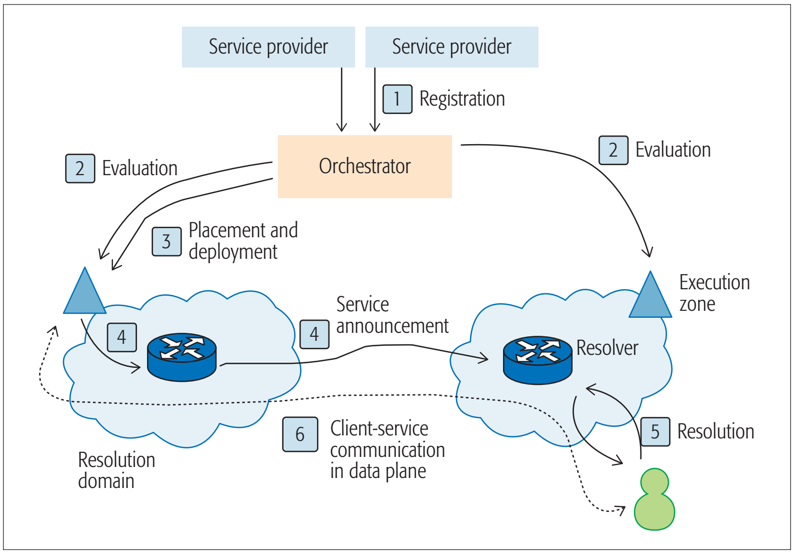


Figure 4. Service life cycle in service-centric networking.

higher priority class, while the other 47 were scheduled with best-effort. It can be clearly observed that the enabled Linux mechanisms result in much stronger guarantees on application performance for all tested virtualization technologies.

The type of hypervisor used and the implementation of the resource isolation mechanisms to provide strict performance guarantees may differ widely among infrastructure providers. Moreover, it is hard for infrastructure providers to come up with a single configuration that is optimal for all applications. First, server workload characteristics continuously change as application instances come and go. Second, there is a wide variety in performance bottlenecks: CPU-intensive, memory-intensive, high I/O, etc. An experimental study concluded that the best-performing configuration for an application in one cloud provider can become the worst-performing configuration for that application in another cloud [12].

Given the impact on service performance of hardware resources, infrastructure management policies, and runtime conditions, it is clear that the cost-versus-quality trade-off of a DC for resource-demanding services is highly application specific. Moreover, the placement decision can only be performed in an optimal way when it is based not solely on static descriptions of DC capabilities, but involves an evaluation of the runtime condition on application-specific requirements.

Service-Centric Networking

The previous discussion reveals that for both service placement and service resolution, detailed knowledge is needed about the capabilities of heterogeneous nodes, the IP network topology, and service performance metrics. This knowledge is scattered between different business entities, such as infrastructure providers, ISPs, and service developers.

In the following, we describe an intermediary service-centric networking (SCN) framework that assists service providers to manage the deployment and operation of services over distributed heterogeneous clouds. This includes the optimal placement of service instances considering the capabilities of DCs, their proximity in terms of network metrics to user demand, dynamic service scaling to meet varying demand, and the resolution of user queries to the best service instance, according to a combination of network metrics, available server capacity, and other operational policies such as minimizing transit costs.

The framework is enabled by several primitives, including evaluator services, session slots, and service catalogs to convey information that is abstract enough to avoid the exposure of IPR on network or service performance, yet contains sufficient detail for service placement and resolution in distributed heterogeneous cloud environments. Placement is performed on a deeper level than the limited set of regions offered by current geo-distributed DC providers, and the service-specific impact of hardware heterogeneity is taken into account when assigning resources to the deployed replicas.

Functional Entities

The SCN framework [4] covers service management and resolution functions implemented by multiple cooperating, but loosely coupled entities: service providers, service orchestrators, DC providers, and service resolvers. The service life cycle across these entities is depicted in Fig. 4.

1. Service providers register their service with an **orchestrator** via an (extended) TOSCA service manifest, containing information such as the service graph identifying service components and their relationship with one another, performance requirements and constraints, and deployment policies.

2. The orchestrator goes beyond cloud infrastructure brokering and also offers advanced instance placement, service life cycle management and monitoring. The orchestrator carries out a detailed **evaluation** of the performance and runtime conditions of a large set of candidate execution locations, named execution zones (EZ). The computational resources may be a dedicated DC of a cloud infrastructure provider, or similar resources co-located with PoP, base stations, etc. provided by an ISP. The placement decision may be based on service-specific evaluator services, a concept further detailed in a later section.

3. The evaluation results are used to **deploy** service replicas in a subset of the EZs, taking into account the service requirements and policies listed in the service manifest.

4. EZs report on their service availability to the **service resolution** subsystem, which is responsible for creating dynamic forwarding paths for end-user queries to be resolved to EZs containing available instances of the requested service. Multiple domain resolvers exchange information on service availability, and each domain has a logically centralized resolver that answers queries from the domain's clients.

5. The resolver returns a **locator** of the service replica to the client. These locators can contain IPv4/IPv6 address, TCP ports, protocol numbers, and/or tunnel identifiers. The location of the resolver for a specific service and/or a given user can be retrieved through standard DNS mechanisms.

6. The client then accesses the service replica over a standard IP connection, **out-of-band** of the SCN framework.

Utility-Based Placement with Evaluator Services

Service placement involves a cost-vs-quality trade-off that is application-specific. The service provider specifies in the manifest the service performance targets by means of a utility function. Utility is defined as a weighted combination of metrics relevant for the service performance and can range from zero to one. Further details on the utility function can be found in [13].

Placement algorithms in the orchestrator need to solve a multi-objective optimization problem to maximize the total utility of all users within budget constraints. We show in the Pareto frontier of the trade-off between placement cost and user utility for the EZs and user demand as described earlier.

Costs are in arbitrary units and are proportional to the published cost of the closest Amazon EC2 for each EZ. The X-axis is the sum of utility each of 1800 user groups received by accessing services in the chosen EZ. Each point "x" on the plane represents a feasible placement solution, but only the points on the Pareto curve represents a maximum utility for a cost constraint value. Each strategy on the Pareto curve shows a particular trade-off between the utility and the cost. Based on this, the service provider can choose an appropriate operating point.

Performance impacting factors such as multi-tenant resource isolation and hardware heterogeneity are only measurable at runtime and/or require in-depth and sensitive knowledge of the service implementation to assess the utility of an EZ. Describing such detailed hardware capabilities and performance dependencies in a static manifest is infeasible. Instead, we propose the concept of evaluator services. These are lightweight services deployed as probes in a selected number of EZs to verify deployment and execution requirements and predict the performance when the application would be deployed in the same environment. Before the service is deployed, the orchestrator deploys one or more evaluator service instances across the candidate EZs. An evaluator service calculates a numerical score for the execution environment. This value, together with network statistics and infrastructure costs, is used as input parameters in the utility function by the orchestrator.

The major advantage of the evaluator service concept is that orchestrators can follow the same evaluation procedure for all services. It is up to the service providers to provide the evaluator services. In the simplest case, the evaluator service only makes a small number of system API calls to verify whether a required hardware or software feature is available; in other cases, a more thorough performance evaluation may be necessary. There should however be a reasonable relation between the complexity of the evaluation and the service itself, as running a complex and time consuming evaluation for a short-lived service would introduce too much overhead.

Both the utility function and the evaluator services are described as policies in a TOSCA service manifest. TOSCA is an OASIS standard for the specification of topology and orchestration of cloud applications [14]. An example is given in Fig. 6. The evaluator service needs to be executed in three regions, and the utility of an execution zone is an equally weighted sum of the end-to-end latency and the numerical score of the evaluator service.

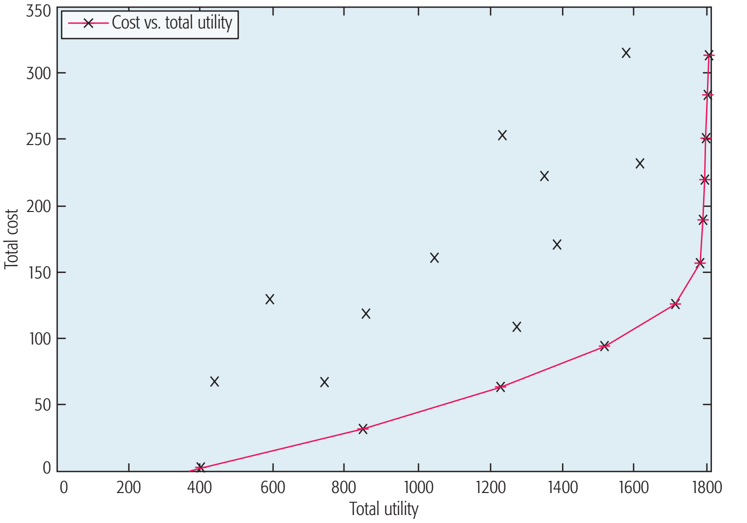


Figure 5. Pareto graph placement cost vs. utility.

Distributed Resolution Based on Session Slots

Service resolution algorithms find the "best" instance among possibly many replicas distributed over the Internet. Simply selecting the closest EZ for each user request or the one that maximizes utility for that individual request can result in sub-optimal performance. As we show in [13], a utility-maximizing service selection approach in SCN can reduce blocking and increase overall utility compared to a classic closest-based selection approach.

The exchange of service availability information consists of two distinct steps: catalogue sharing and service subscription.

Catalogue Sharing: Orchestrators deploy an agent in each EZ that announces the service ID, the utility function, and a representative locator to at least one resolver. This information is further injected into the catalog, which is shared between resolvers using a DHT implementation. This information only updates when a new service is created, all service instances have been deleted, or there are significant changes in network connectivity (e.g. a change of traffic engineering policy). To keep full control of the load on some instances, resolvers may decide to hide the actual locators and replace them with an ALTO provider-defined identifier (PID). ALTO is an IETF standard for dissemination of network-level information between different business entities [15]. The PID is a representative locator for, e.g., a subnet or a metropolitan area that allows other resolvers to assess the potential performance of connections to instances running in that domain. Operators expose cost maps, assigning cost values (e.g. routing cost) to one-way connections between PIDs. Other resolvers can then evaluate the feasibility of service replicas exposed by one resolver, without having full knowledge of the internal network or the operator policies.

Service Subscription: Based on the catalog information, resolvers subscribe to a set of EZ. To obtain enough diversity of service availability,

There are several areas for ongoing study, including: modelling and mitigating policy mismatches between service placement and resolution when deployment and networking costs are not aligned. For extremely low-latency tactile services, additional edge computing nodes may need to be utilized to deploy service instances much closer to users.

```
topology_template:
  node_templates:
    my_service:
      type: tosca.nodes.Compute
     properties:
       # omitted here for brevity
     requirements:
       # other requirements omitted here for brevity
       - evaluator_service:
             node: my_evaluatorServiceFeatureA
             relationship: my_evaluator

    my_evaluatorServiceFeatureA:
     type: tosca.nodes.Compute
     # omitted here for the brevity
  policies:
    -    my_evaluator_placement_policy:
          type: my.policies.evaluator #derived from tosca.policies.Placement
          container type: region
          target_regions: [ regionA, regionB, regionC ]
          evaluator: my_evaluatorServiceFeatureA
          min_score: 250
    -   my_utility_policy:
         type: my.policies.latency_utility #derived from tosca.policies.performance
          R: 0.5*e2e_lat + 0.5*evalFeatureA

  # other resources not shown here ...
```

Figure 6. Sample TOSCA manifest. Two policies are defined, based on a geographic spreading as well as on utility.

resolvers will contact close zones before expanding the subscriptions to more distant zones until enough instances are found. Resolvers will start receiving updates from that EZ on the availability of the service(s) subscribed to. The availability information is conceptualized as session slots. A session slot is a unit of measurement representing how many users can be accommodated simultaneously in a given service instance, group of instances, or EZ. The total number of session slots to be instantiated is decided by the orchestrator, and the current number of available session slots is announced to the service resolvers to help drive the instance selection algorithms.

The resolution overlay can grow organically. In an early phase, orchestrators could act as resolvers to ensure reachability of their managed services. Over time, other parties could attach resolvers to the resolution overlay. As argued earlier, resolvers may be operated by ISPs.

Conclusion

In this article, we present a framework for optimal service placement and resolution in widely distributed heterogeneous cloud infrastructures. SCN leaves the data plane unmodified and therefore aligns with other efforts to improve service delivery, such as software defined networking to manage data flows, and 5G wireless technologies to improve wireless throughput and latency.

The SCN framework has been extensively modelled and prototyped in the FUSION project. Some of the challenges of deploying SCN, as discussed in this article, involve the definition of appropriate abstractions of service requirements and the inclusion of network and service monitoring data in placement and resolution decisions. The primitives of evaluator services, utility, and session slots are able to capture the vast diversity in service requirements at an appropriate demarcation level between different business entities for orchestration and resolution. Together with these primitives, the adoption of standards such as TOSCA and ALTO ease the deployment of SCN. The deployment of SCN is also facilitated by it not requiring to be deployed as a single big-bang solution. For example, service resolution can initially be undertaken by service-specific centralized functions. For more popular services that are more widely deployed, and especially for those that require a more detailed knowledge of network performance metrics than can be provided by OTT monitoring, then the resolution function can be incrementally deployed by ISPs.

There are several areas for ongoing study, including: modelling and mitigating policy mismatches between service placement and resolution when deployment and networking costs are not aligned. For extremely low-latency tactile services, additional edge computing nodes may need to be utilized to deploy service instances much closer to users. Globally centralized placement optimization functions do not scale well at this level of detail, and hierarchical placement frameworks may be needed where algorithms at lower levels in the hierarchy are able to make detailed placement decisions with local knowledge of edge nodes, user locations, and network topology.

References

[1] S. Choy *et al.*, "The Brewing Storm in Cloud Gaming: A Measurement Study on Cloud to End-User Latency," *Proc. 11th Annual Wksp. Network and Systems Support for Games*, 2012.

[2] B. Frank *et al.* "Pushing CDN-ISP Collaboration to the Limit," *ACM SIGCOMM Computer Comm. Review*, vol. 43, no. 3, 2013.

[3] S. Narayana *et al.*, "Joint Server Selection and Routing for Geo-Replicated Services," *Proc. 2013 IEEE/ACM 6th Int'l. Conf. Utility and Cloud Computing*, IEEE Computer Society, Dec. 2013, pp. 423–28.

[4] S. Paul *et al.*,, Application Delivery in Multi-Cloud Environments Using Software Defined Networking," *Computer Networks*, vol. 68, 2014, pp. 166–86.

[5] M. Malekimajd, A. Movaghar, and S. Hosseinimotlagh, "Minimizing Latency in Geo-Distributed Clouds," *J. Supercomputing*, vol. 71, no. 12, 2015, pp. 4423–45.

[6] L. Gu *et al.*, "Optimal Task Placement with QoS Constraints in Geo-Distributed Data Centers Using DVFS," *IEEE Trans. Computers*, vol. 64, no. 7, 2015, pp. 2049–59.

[7] Q. Zhang *et al.*, "Dynamic Service Placement in Geographically Distributed Clouds," *IEEE JSAC*, vol. 31, no. 12, 2013, pp. 762–72.

[8] M. Boniface, *et al.* “Platform-as-a-Service Architecture for Real-Time Quality of Service Management in Clouds,” *2010 5th Int’l. Conf. Internet and Web Applications and Services (ICIW), IEEE*, 2010.

[9] S.-I. Lee *et al.*, “NGSON: Features, State of the Art, and Realization,” *IEEE Commun. Mag.*,” vol. 50, no. 1, 2012, pp. 54–61.

[10] R. Landa *et al.*, “The Large-Scale Geography of Internet Round Trip Times,” *IFIP Networking Conf.*, 2013, pp. 1–9.

[11] G. Fettweis *et al.*, “5G: Personal Mobile Internet Beyond What Cellular Did to Telephony,” *IEEE Commun. Mag.*, vol. vol. 52, no. 2, 2014, pp. 140–145.

[12] D. Jayasinghe *et al.*, “Variations in Performance and Scalability: An Experimental Study in IaaS Clouds Using Multi-Tier Workloads,” *IEEE Trans. Services Computing*, vol. 7, no. 2, June 2014, pp. 293–306.

[13] T. K. Phan *et al.* “Utility-Maximizing Server Selection,” *Proc. Of the IFIP Networking Conf.*, 2016.

[14] OASIS, “TOSCA Simple Profile in YAML Version,” http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.0/TOSCA-Simple-Profile-YAML-v1.0.html, accessed on Sept 19, 2016.

[15] RFC7285, Application-Layer Traffic Optimization Protocol.

Biographies

Pieter Simoens (pieter.simoens@intec.ugent.be) received his Ph.D. degree in 2011 from Ghent University and is now an assistant professor at the same institute. His research interests include distributed real-time systems, with a specific focus on the delivery of advanced services through distributed edge clouds. He has (co-) authored more than 70 articles in journals and conference proceedings.

David Griffin is a principal research associate in the Department of Electronic and Electrical Engineering, University College London. He has a BSc from Loughborough University and a Ph.D. from UCL, both in electronic and electrical engineering. His research interests include planning, management, and dynamic control for providing QoS in multiservice networks and novel routing paradigms for the future Internet.

Elisa Maini is a research associate in the Department of Electronic and Electrical Engineering, University College London. She received her Ph.D. in computer and automation engineering from the University of Naples Federico II. Her current research interests include network optimization and modelling, software-defined networking, and network function virtualization.

Truong Khoa Phan received his Ph.D. degree from INRIA/I3S, Sophia, France. He is currently a research associate in the Department of Electronic and Electrical Engineering, University College London. His research interests include network optimization, cloud computing, multicast, and P2P.

Miguel Rio received the Ph.D. degree from the University of Kent, Canterbury, U.K., and the M.Sc. and M.Eng. degrees in informatics from the University of Minho, Braga, Portugal. He is a reader (associate professor) of computer networks in the Department of Electronic and Electrical Engineering, University College London, London, U.K. He has authored extensively in top ranked conferences and journals. His research interests include network measurement, congestion control, new network architectures, and, more recently, the interaction between cloud and network services.

Luc Vermoesen is a research engineer in the IP Platforms Research Program at Bell Labs in Antwerp, Belgium. He graduated in engineering in 1989 and studied computer science in 1995. In 2000 he joined Alcatel-Lucent, where he worked on projects involving 3G mobile, VDSL prototyping, asynchronous access multiplexer, and IP service routing and switching. In 2007, he joined the Bell Labs Fixed Access team, where he was involved in home networking research and contributed to the Broadband Forum standardization activities. In 2009, he started working on multimedia-related research topics such as novel graphical user interfaces for IPTV and network-based rendering techniques using dedicated HW acceleration. From 2011 onward, he has been involved in cloud computing research with specific interest in virtualization and performance, as well as the applicability of heterogeneous hardware in the cloud. He currently holds over a dozen patents.

Frederik Vandeputte received his Ph.D. degree in 2008 from Ghent University and is now a research engineer at Nokia Bell Labs. His research interests include software parallelization on heterogeneous architectures, heterogeneous cloud systems, network functions virtualization, and performance optimization. He has (co-)authored over a dozen articles in journals and conference proceedings.

Folker Marten Schamel is founder and managing director of Spinor GmbH, a provider of the Shark 3D software for creating interactive virtual worlds in the gaming, movie, and broadcasting industries. He is credited with contributing to the specification of the OpenGL standard. He has a diploma in theoretical physics with mathematics as a minor.

Dariusz Bursztynowski received his Ph.D. in telecommunications from Warsaw University of Technology in 1992. His research interests include network architecture, traffic engineering, network performance modelling and evaluation. He has been involved in a number of Orange activities related to network planning, network management, and traffic engineering. He is currently working at Orange on future network architectures in the field of naming, routing, and autonomic resource management mechanisms.

Shedding Light on the Internet: Stakeholders and Network Neutrality

Angelos Antonopoulos, Elli Kartsakli, Chiara Perillo, and Christos Verikoukis

The authors seek to shed light on the emerging Internet ecosystem and the conflicting interests of its stakeholders. They first identify the different Internet players and describe their interrelationships. In an effort to offer a new perspective on the network neutrality debate, they propose two novel econometric models that employ recent financial data to capture the relationship between the OTT revenues and the financial gains and investments of the telecommunication operators.

Abstract

The latest impressive technological advancements in the telecommunications domain have entailed the involvement of new network operators and over-the-top (OTT) providers that offer their services over the existing networks. This entry of new stakeholders has changed the Internet dynamics and triggered a long-standing conversation on whether different types of data in the network should be prioritized, also known as the network neutrality debate. On the one hand, OTT providers benefit from the current neutral Internet policy of not discriminating against any application or content in order to transfer their data for free, whereas network providers would like to seize the business opportunity and create revenues by supporting the prioritized delivery of data. In this article, we want to shed light on the emerging Internet ecosystem and the conflicting interests of its stakeholders. To that end, we first identify the different Internet players and describe their interrelationships. Furthermore, in an effort to offer a new perspective on the network neutrality debate, we propose two novel econometric models that employ recent financial data to capture the relationship between the OTT revenues and the financial gains and investments of the telecommunication operators. Our empirical results provide tangible answers to fundamental questions that had not been answered before, showcasing that OTT and telecommunication providers have aligned interests and their collaboration could be beneficial to both parties.

Introduction

The phenomenal adoption of mobile devices and applications creates an unprecedented need for providing the end users with ubiquitous and uninterrupted Internet connectivity. This technology evolution has two immediate effects, as the existing network operators are prompted to invest in order to extend and upgrade the network infrastructure, while the telecommunications market expansion motivates new players, such as virtual operators and content delivery providers, to become involved in service provisioning. Besides the aforementioned changes in the networking part, we are also experiencing a paradigm shift, as the proliferation of mobile communications has brought new stakeholders to the spotlight. More specifically, the technological advancements have enabled the introduction of over-the-top (OTT) providers, who offer their services over the existing deployed telecommunication networks and are mainly classified as content distributors (e.g., YouTube), social network operators (e.g., Twitter), and companies that offer communication services (e.g., WhatsApp and Skype) similar to the conventional services provided by the network operators.

The entry of new players with conflicting interests in the field has further complicated the already obscure multi-tenant structure of the Internet. More specifically, network operators argue that the new companies use their network to transfer huge amounts of traffic without generating direct revenues for their benefit. On the other hand, OTT providers, invoking the network neutrality rules [1], consider network providers as common carriers who should not be given the right to prioritize the traffic. Although there have been some recent efforts by telecommunications regulators (i.e., the Federal Communications Commission and the Body of European Regulators of Electronic Communications) to bring this long-standing dispute to an end, the opposing parties do not show willingness to compromise and adhere to their initial stances.

In light of the above discussion, the aim of this article is twofold. First, considering the latest developments and the entrance of new stakeholders into the Internet domain, we try to characterize the distinct roles of the multiple tenants and identify the issues that arise in their interrelationships. Second, given the massive appearance of competitive OTT applications in the market during the end of the last decade, we believe that sufficient time has elapsed to enable a clear analysis of the relationship between communication OTT and telecommunication providers. To that end, we conduct a detailed econometric study to examine the correlation among a series of relevant parameters, including the network providers' growth, network investments, OTT revenues, and Internet penetration. Our findings constitute an important contribution to the network neutrality debate, as they provide some initial tangible answers to two fundamental open questions:

- Do OTT providers constitute a threat for mobile operators and their financial interests?
- Should OTT providers be burdened with extra fees to account for the operator's expenses for the network infrastructure expansion that is required to accommodate the additional traffic demands?

Digital Object Identifier: 10.1109/MCOM.2017.1600417

Angelos Antonopoulos and Christos Verikoukis are with Telecommunications Technological Centre of Catalonia (CTTC/CERCA); Elli Kartsakli is with IQUADRAT Informàtica S.L.; Chiara Perillo is with University of Zurich.

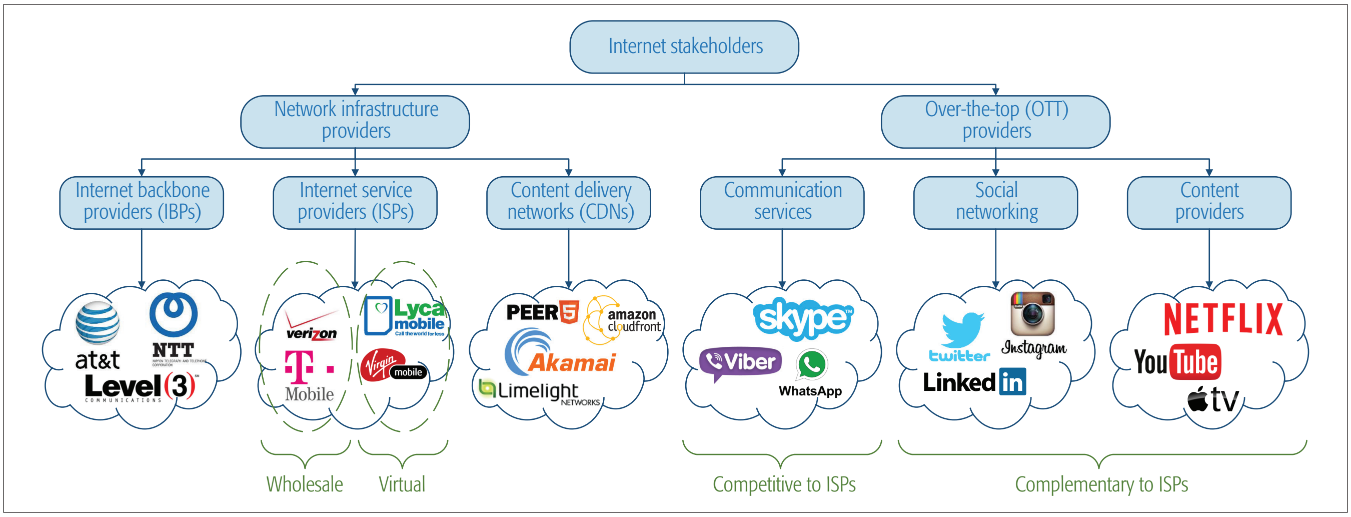

Internet stakeholders
Network infrastructure providers
Over-the-top (OTT) providers
Internet backbone providers (IBPs)
Internet service providers (ISPs)
Content delivery networks (CDNs)
Communication services
Social networking
Content providers
Wholesale
Virtual
Competitive to ISPs
Complementary to ISPs

Figure 1. Internet stakeholders.

In the following section, we briefly present the key Internet stakeholders and their interrelationships, focusing on the network neutrality concept. Subsequently, we provide the two econometric models and the empirical results derived from our analysis, focusing on the OTT providers that offer competitive communication services. Finally, we discuss the practical importance of our results in the network neutrality debate, forming some interesting future research directions.

Internet Stakeholders, Relationships and Network Neutrality

Internet is a broad concept, usually characterized by a physical infrastructure (e.g., servers, routers, base stations) and network services (e.g., voice, video, messaging). As the rapid developments in the telecommunications domain have lately brought new players to the forefront (a general classification is shown in Fig. 1), the aim of this section is to identify the involved parties and their relationships in the evolving Internet ecosystem.

Internet Stakeholders

Network providers can be roughly classified into three broad categories: Internet backbone providers (IBPs), Internet service providers (ISPs), and content delivery network (CDN) providers. There are only a few IBP companies worldwide, forming the backbone (core) Internet, where long-haul and metropolitan fiber-optic connections are deployed. On the other hand, ISPs include the majority of widely known telecommunication companies that offer fixed and mobile Internet services. Going one step further, ISPs can also be distinguished into *wholesale ISPs*, who have full ownership and control of their network infrastructure, and *virtual ISPs* that operate a virtual network by leasing network infrastructure from other wholesale providers. Finally, CDNs are distributed networks of cache servers installed in diverse geographical locations, aiming to store web content closer to the end user, thus reducing network congestion and accelerating delivery. It is worth noting that the classification of the network infrastructure providers is not strict, as there is potential overlapping (e.g., an IBP can act as an ISP, offering access services to the end users). However, this simplified but clear role separation will facilitate the study of the relationship among the different entities, providing interesting insights and directions for the Internet evolution.

The landscape is much clearer with respect to the OTT service providers, which can be classified into three categories:

- Communication service providers that offer Internet-based voice and instant messaging.
- Social networking companies.
- Content providers.

The relationship between OTTs and network providers is controversial, since the offered services may either be similar to those of ISPs (i.e., communication services), leading to direct competition and conflicting interests, or complementary (i.e., social networking and content distribution), thus adding value to Internet connectivity. Apart from the distinctive roles of the OTT providers, the OTT applications have different quality of service (QoS) demands (e.g., bandwidth, delay, jitter, etc.) that define their requirements in network resources. These limitations, along with the multi-tenant nature of the Internet, imply complex ties and correlations among the existing stakeholders that will be discussed in the following section.

Internet Relationships and Network Neutrality

Figure 2 provides a clear view of the key Internet players, highlighting the established relationships among them (depicted as solid arrows). The Internet representation takes the form of a reverse pyramid, where IBPs are placed at the bottom and interact with the network providers of the intermediate layer (ISPs and CDNs) through service level agreements (SLAs). The emerging OTT companies are placed at the top of the pyramid, and their relationships with the network providers (characterized with an interrogation mark) are yet to be consolidated. Finally, the end users are placed at the left side of the pyramid, interacting with both ISPs and OTTs through different pricing schemes.

Existing Relationships: Although there is no clear definition for the Internet structure, it is

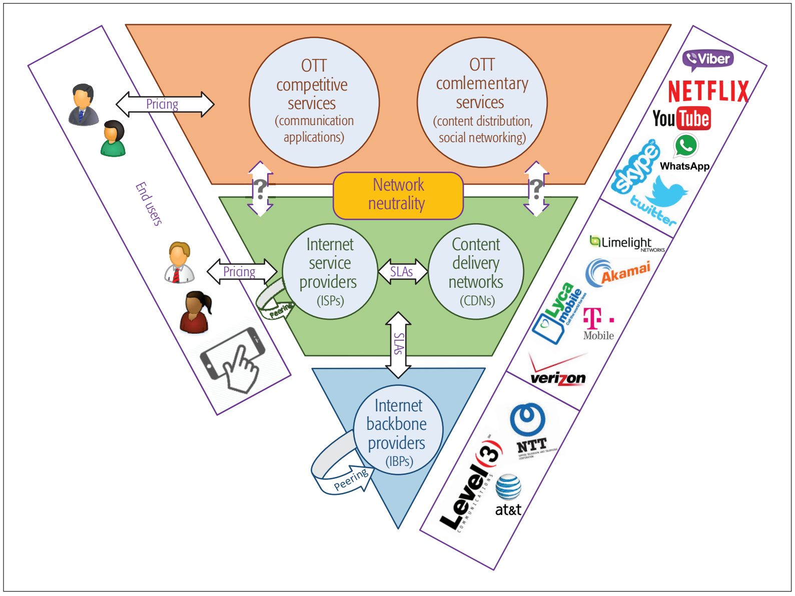


Figure 2. Relationships among key Internet players.

usually conceptually divided into three main tiers (Fig. 3). Tier 1 is formed by the backbone providers (IBPs), while the ISPs in the upper tiers are classified according to their size and the level of control on their networks. Apparently, there is no single network provider that can offer global Internet reachability (i.e., the ability to access every online destination from any address). As a result, end-to-end Internet connectivity relies on the transfer of data through a complex interconnection of networks owned by different providers. This paradigm encourages network providers to establish mutual business agreements to regulate the data transfer through their networks. The two prevalent types of economic agreements between interconnected providers are known as *peering* and *transit*.

Peering agreements concern the data transfer (usually symmetric) between two networks, where both parties benefit equally, and hence no fees are paid. A peering arrangement can be either private or public. In case of *private peering*, which mainly takes place between IBPs (and few very large ISPs), a dedicated physical connection (e.g., fiber links) is created to enable the exchange of large traffic volumes in a secure and reliable way. On the other hand, the majority of ISPs form *public peering* relationships to exchange smaller amounts of traffic through shared networks, known as Internet exchange points. Whereas peering agreements are usually concluded between equivalent partners with mutual expected benefits, smaller ISPs must typically pay transit fees to larger networks (IBPs or large ISPs) to gain global Internet access. Different business models have been developed to match the providers' needs, where transit payment may be determined according to various parameters, e.g., the requested capacity, the actual volume of exchanged traffic, or the particular routing of the data.

The recent deployment of CDNs has driven a new type of commercial agreement between the Internet stakeholders, namely *paid peering*, which, unlike public peering, incurs a charge for the data exchange, but without provision for end-to-end data delivery (unlike transit agreements). The key incentive for alliances between CDNs and ISPs lies in the market share, as the CDNs have a strong customer base consisting of content providers that pay to ensure high-quality content delivery to the end users, while ISPs have direct access to the end users.

Different relationships are developed in the case of virtual operators, who must negotiate the wholesale price for Internet access and the provided service guarantees with the host ISP. In addition, the need for extensive network infrastructure (to meet the increasing service demands) in conjunction with the effort to minimize capital (CAPEX) and operational (OPEX) expenses, have motivated operators to adopt the infrastructure sharing paradigm. This novel concept encompasses various degrees of sharing, including the passive sharing of infrastructure elements (e.g., antenna sites), the active sharing of network components (e.g., routers) and the joint provisioning of user services through roaming-based agreements.

The last, but equally important, class of business relationships among Internet stakeholders refers to the pricing mechanisms between the service providers (either ISPs or OTT providers) and the end users. Regarding the ISPs, the widespread use of broadband technologies has led to a tendency of charging flat rates for voice services. On the other hand, OTTs, as newer players in the market, have adopted more attractive and targeted business models [2], such as periodical subscriptions, payments by transactions, and freemium services (with free basic usage but advanced features under payment), while advertisements, donations, and data monetization are alternative ways to increase their revenues.

The research community has extensively tried to model these complex relationships between the Internet players, from both a technical and economic perspective. Optimal peering decisions according to the network formation have been studied in [3]. Game theory has been a valuable tool for modeling the peering and transit interactions among ISPs. For instance, employing the concept of Shapley value from coalition games has led to the design of a fair profit-sharing mechanism among Internet stakeholders, which further encourages peering arrangements between neighboring ISPs [4]. The potential benefits of a close alliance between ISPs and CDN are discussed in [5], where a set of mechanisms to enable this collaboration is proposed. As new players have been entering the telecommunications sector, their market viability and potential business strategies are explored through techno-economic analysis. The penetration of virtual ISPs (vISPs) to the market has been studied to identify new opportunities and potential threats to traditional operators [6]. Advanced game-theoretic algorithms for the efficient utilization of the shared infrastructure among multiple network operators have also been proposed, aiming to save energy and reduce operation costs [7]. Finally, new challenges in the design of charging and billing mechanisms arise as multiple providers compete for the provision of Internet ser-

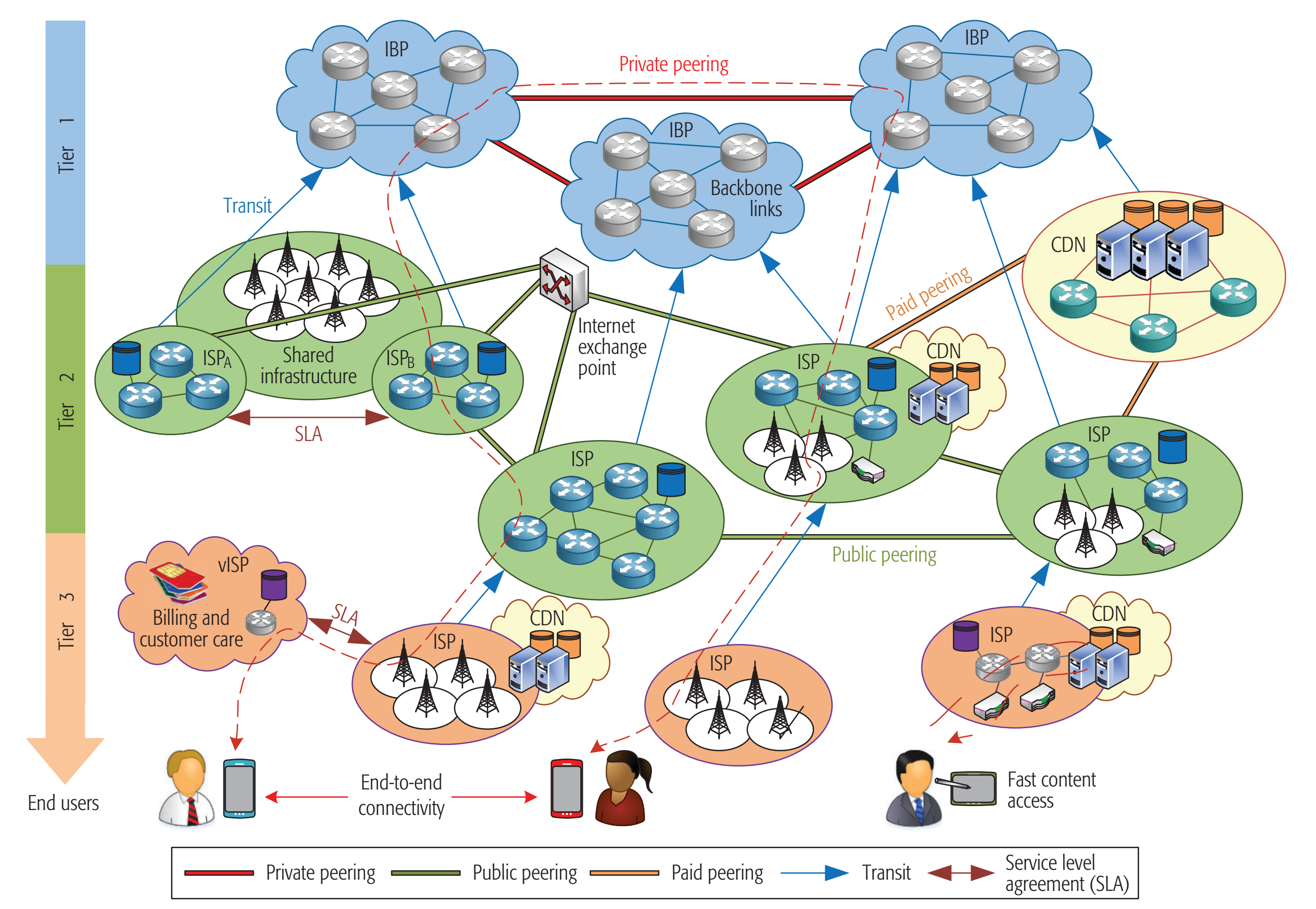


Figure 3. Multi-tier structure of the Internet.

vices, making imperative the need for flexible, scalable and fair pricing schemes [8].

ISPs vs. OTTs and Network Neutrality: Unlike the above discussed existing relationships, which are relatively clear, the recent entry of OTT providers entails new ties that have not yet been definitively developed. This paradigm shift was the main impetus for the opening of a very serious and long-standing conversation about Internet data regulation, commonly referred to as the network neutrality debate. In essence, network neutrality is the absence of discrimination and restrictions on the transmission of content. Although the idea of service differentiation is not new and traffic shaping has already been applied in the Internet (e.g., through DiffServ implementation in network routers), the network neutrality debate was mainly ignited in 2005 by the statement of the AT&T's chairman that he was not willing to let content providers use the network for free,[1] creating two opposing camps, i.e., for and against network neutrality.

Supporters of network neutrality include, among others, the main OTT players, and their basic argument is that the Internet evolution and success have mainly relied on network neutrality. Hence, ISPs should not have any control of the network data, otherwise newer online companies would have a disadvantage, eventually leading to the transformation of the Internet from a market ruled by innovation to one ruled by deal-making.[2] On the other hand, the most zealous opponents of network neutrality are the ISPs, who have set two basic arguments, claiming that OTT companies:

- Have conflicting interests and provide competitive services, thus constituting a threat to their own growth.
- Distort incentives for investment, as they essentially exploit the network already deployed by ISPs, acting as free riders.

Hence, to overcome these issues, they propose to impose extra fees to major OTT providers to prioritize their traffic, using the extra revenue for network expansion and enhancement of broadband access to more consumers.

The importance of the network neutrality debate worldwide [9] has motivated the research community to study the interaction among the different entities from a theoretical point of view. These works usually employ game theoretic tools (e.g., non-cooperative game theory and Stackelberg games) to analyze the forces driving the Internet economics evolution [10], to propose new business models [11, 12], and to study different aspects of the problem as, for instance, the impact of competition between ISPs on network neutrality [13], the feasibility of charging content providers [14], or even the potential of building new OTT infrastructures exclusively employed for content distribution [15]. However, despite the interesting theoretical conclusions of these works, the observation of the actual progress of the firms would be of great importance, providing us with further insights, as we will see in the following section.

[1] http://arstechnica.com/uncategorized/2005/10/5498-2/

[2] T. Wu, "Why You Should Care About Network Neutrality," May 2006, http://www.slate.com/id/2140850

The end of the previous decade was a key point for the Internet evolution, mainly because of the introduction of LTE and the widespread proliferation of smart devices. Over the last few years, these new technologies have been solidly established in our everyday life, giving us a clear picture of the landscape along with empirical observational data.

Network providers	
Country	**ISPs**
USA	AT&T, Verizon, Deutsche Telecom
Japan	NTT DoCoMo, Softbank
UK	BT Group, Vodafone, Telefónica
Germany	Deutsche Telecom, Vodafone, Telefónica
Italy	Telecom Italia, Vodafone
Spain	Telefónica, Vodafone, Orange
France	Orange
OTT providers	
Skype, WhatsApp, Facebook (Messenger)	
Notation	
Variable	**Definition**
R_{ISP}	Revenues of Internet service providers
R_{OTT}	Revenues of over-the-top providers
C_{ISP}	Capital expenses (CAPEX) for network infrastructure of Internet service providers
N_u	Number of Internet users (Internet penetration) per country
GDP	(Real) gross domestic product (reflecting the economic performance) per country

Table 1. Empirical data sources and notation.

ISPs vs Competitive OTT Providers: Empirical Econometric Analysis

The end of the previous decade was a key point for the Internet evolution, mainly because of the introduction of LTE and the widespread proliferation of smart devices. Over the last few years, these new technologies have been solidly established in our everyday life, giving us a clear picture of the landscape along with empirical observational data.

Hence, we believe that the time is ripe to employ econometric tools to examine the interaction of the new stakeholders from a macroscopic viewpoint and quantify the impact of relevant parameters on the development of ISPs and OTT providers. In this section, we provide details about the data and the adopted methodology, and we discuss the outcome of our research.

Methodology and Data

The empirical analysis has been conducted with regard to the period 2008-2013 (both years inclusive), considering data from ten ISPs and three OTT providers that offer competitive communication services. The involved stakeholders and the employed variables of our study are summarized in Table 1.

Variable	x_{min}	x_{max}	$\bar{x}$	σ
R_{ISP}	26 586.99	273 944.80	77 875.44	76 537.74
R_{OTT}	0.05	3616.70	312.58	736.71
C_{ISP}	3636.89	41 902.58	11 731.64	12 016.79
N_u	26.19	266.49	81.07	67.24
GDP	12 823.80	156 960.20	46 913.62	44 634.69

Note: all the economic values (i.e., revenues, CAPEX, GDP) are expressed in millions of US dollars, while the number of Internet users is expressed in millions of people.

Table 2. Descriptive statistics of the variables.

The data set has been constructed by combining inputs from a large number of reliable sources, as there is no available database with the aggregate information.[3] It is worth noting that the calculation of the ISP revenues and CAPEX per country is straightforward, as the ISPs are physically present and offer their services in each country. However, as the revenue of an OTT company in a particular country is not available, we have made an approximation, taking into account the total annual revenue of the company and the portion of the customers in a given country. For instance, assuming that an OTT provider has a total annual revenue of $1M and 20 percent of its total users reside in Italy, we consider that the corresponding revenue in Italy is $200K. For the number of Internet users, we have combined the Internet penetration rate and the population of the country, while the real GDP has been estimated by the nominal GDP and the Consumer Price Index.[4] The descriptive statistics that summarize the basic features (i.e., minimum (x_{min}), maximum(x_{max}), average ($\bar{x}$) values and standard deviation (σ) of the data are presented in Table 2.

The employed data set constitutes a balanced panel, as it contains observations of multiple parameters obtained each year for all parties. The analysis of cross-sectional time series panel data is usually associated with two important linear regression models: fixed and random effects. Their main difference lies in how they characterize the dependent variable. From a theoretical point of view, the fixed effects model is more suitable when each analyzed entity has unique individual characteristics that may have some influence on the variables, whereas the random effects model may be employed if the entities can be considered as random extractions from a population. In our case, the most appropriate approach is the fixed effects model, since the entities under study are countries with distinct characteristics that may be correlated with the considered parameters (e.g., the regulations of a given country could have an effect on the ISP investments or the OTT revenues). Our choice was further validated by the Hausman test, which is typically employed to evaluate the suitability of each method for a given data panel.

Empirical Results

Following the fixed effects approach, we propose two econometric models for the revenue of ISPs and OTTs ($R_{ISP_{it}}$ and $R_{OTT_{it}}$, respectively), for a given country i and year t, formulated as

[3] The collected data of our work have been made publicly available here: https://arxiv.org/abs/1612.06451

[4] All the statistical data have been obtained from the World Bank database (http://data.worldbank.org/)

Model A:

$$R_{ISPit} = \alpha_i + \beta_1 R_{OTTit} + \beta_2 C_{ISPit} + \beta_3 N_{uit} + \beta_4 GDP_{it} + \varepsilon_{it}$$

Model B:

$$R_{OTTit} = \kappa_i + \gamma_1 R_{ISPit} + \gamma_2 C_{ISPit} + \gamma_3 N_{uit} + \gamma_4 GDP_{it} + \varepsilon_{it},$$

to capture the impact of the explanatory variables (right part of the models) on the dependent variables R_{ISP} and R_{OTT}). All variables have already been defined in Table 1, while α_i (in Model A) and κ_i (in Model B) are the unknown intercepts for each country (considered fixed), β_j, γ_j $\forall j \in [1,4]$ are the coefficients of the respective explanatory variables, and ε_{it}, υ_{it} are the error factors.

For the estimation of our models, the statistical software package Stata (Release 13, StataCorp. 2013) has been employed, and the results are provided in Table 3. Our key aim has been to determine the dynamics among the ISPs' growth, the revenues of the competitive OTT providers, and the CAPEX investment (highlighted in the table). As a first observation, it is worth noting that the obtained results for these relationships have a high statistical significance in both models, as indicated by the p-value (i.e., $p < 0.001$), which is used in statistics to support the strength of an empirical conclusion. Moreover, the interpretation of these results reveals two very intriguing insights. First, both models verify the positive correlation between the revenues of ISPs and OTT providers. More specifically, the coefficient of 9.81 (Model A) implies that the increase of one unit in the revenue of the OTT providers is accompanied by an average increase of approximately 10 units in the ISP revenues, while the coefficient of 0.03 (Model B) also confirms that the growth of the ISPs is positively correlated with the growth of the OTT providers. Second, the conclusions for the network investments are also very interesting, as it is shown that CAPEX have a positive influence on the ISP revenues (with a particular coefficient of 3.21), while adversely affecting the OTT income (with a particular coefficient of –0.13). Regarding the total number of users, the results in Model B verify with high significance that Internet penetration is positively related with the OTT profits as expected, while the results in Model A are counter-intuitive, demonstrating a negative relationship between Internet penetration and ISP revenues. Although reasonable explanations can be found for this relationship (e.g., additional operational costs), it should be noted that the significance of this relationship, as indicated by the respective p-value, is lower compared to the previous results, and further research should be done in this direction for more concrete conclusions. Finally, the impact of GDP in both models is not statistically significant.

Dependent variable: R_{ISP}	Model A		Dependent variable: R_{OTT}	Model B	
	Coefficient	t-stat		Coefficient	t-stat
R_{OTT} (β_1)	9.81	(3.68)***	R_{ISP} (γ_1)	0.03	(–6.85)***
C_{ISP} (β_2)	3.21	(7.15)***	C_{ISP} (γ_2)	–0.13	(–4.02)***
N_u (β_3)	–334.70	(–2.17)*	N_u (γ_3)	41.65	(7.50)***
GDP (β_4)	0.20	(0.62)	GDP (γ_4)	0.02	(1.23)
N	42		N	42	
R^2	0.998		R^2	0.917	
Adj. R^2	0.997		Adj. R^2	0.890	

Legend: * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$
N → number of observations
R^2 → coefficient of determination
Adj. R^2 → adjusted coefficient of determination

Table 3. Estimation results.

Discussion and Open Research Lines

The observations of the empirical analysis are very important, as they provide some initial tangible arguments and answers to the questions posed in the network neutrality debate. Let us recall that the ISPs, who are among the most passionate adversaries against network neutrality, consider the OTT companies as a major threat for their own interests, and they have expressed their willingness to charge them with extra fees, or even require the OTT providers to contribute to the expansion of the telecommunication network infrastructure. However, our study has weakened these particular claims, providing empirical evidence that the economic prosperity of the OTT firms is in line with the financial performance of the ISPs. Consequently, it can be concluded that these two important stakeholders fruitfully coexist in the telecommunications and Internet domain, and they should probably work more closely together to achieve a mutually profitable cooperation.[5] In addition, our empirical results also demonstrated the positive effect that network investments have on ISP revenue, while they can be detrimental to the development of the OTT providers. Our results constitute a first step toward refuting the accusations toward OTT companies for free riding and stress the need for additional studies on the causal relationship among the stakeholders.

In a nutshell, our research has brought to light some important facts with regard to the relationship between ISPs and OTT companies. However, as in most serious debates, the truth may lie somewhere in the middle. More specifically, the possible changes in the way the Internet operates should be made considering the common right of everyone to access the Internet safely and with high quality of experience (e.g., prioritizing real-time services) and not according to specific corporate interests (e.g., prioritizing video traffic of a particular company) that may lead to monopoly situations. Furthermore, our work can be considered as an initial effort toward characterizing the dynamics between different stakeholders in the telecommunications market and paves the way for new research lines that will take into consideration the following:

Recent Developments: The most immediate step consists in the extension of our results for the years after 2013. More specifically, updated empirical studies are required to follow the extremely rapid Internet evolution. Furthermore, more variables (e.g., average revenue per user and number of ISPs in a given country) and more accurate data can be taken into account, as more information about the growth of the companies and their market share in different countries

[5] In an effort to bridge this gap, Ericsson recently announced the launch of OTT Cloud Connect (OCC), an open cloud service that allows mobile operators across the globe to "connect" to multiple OTT players to deliver new and creative services to users. See: http://www.reuters.com/article/idUSFWN161019

Next generation wireless networks will embrace a list of new technologies, including Internet of Things (IoT), cloud computing, and software defined networking (SDN), which will create important opportunities for content and application providers to deliver highly innovative services.

become available. Forthcoming work could also take into consideration emerging applications (e.g., Snapchat, Viber, etc.), as well as business activities and transactions (e.g., the WhatsApp acquisition by Facebook).

New Players and Industries: Our study has focused on the relationship between ISPs and competitive OTT providers that offer similar communication services. However, network neutrality also concerns other major OTT players that provide complementary services to the end users. The most representative example consists in the recent agreement between Netflix and Comcast for enhanced QoS, which came in response to the slow connection speeds. In the same context, it would also be interesting to study the relationship between existing service providers and their OTT competitors (e.g., traditional TV operators vs. OTT Internet TV, or even outside the telecom industry with taxi services vs. Uber and hotel services vs. AirBnB) since they have conflicting interests.

Evolving Network Topologies: Although OTT applications have been initially deployed over existing networks, their explosive growth has been driving OTT providers toward acquiring proprietary equipment and infrastructure. Facebook and Google have already initiated efforts toward installing fiber cables across the Pacific ocean, while Akamai has been deploying thousands of servers worldwide. This radical evolution will soon create the need for new Internet maps and theoretical studies for the smooth incorporation of the new infrastructure in the existing networks. Moreover, this development will add more variables to be studied in econometric approaches, while fostering new business models with regard to cost sharing among the involved stakeholders.

5G Enabling Technologies: Next generation wireless networks will embrace a list of new technologies, including Internet of Things (IoT), cloud computing, and software defined networking (SDN), which will create important opportunities for content and application providers to deliver highly innovative services. The experience gained from the empirical and theoretical studies on existing relationships will certainly serve as a guide for capturing future disruptive developments.

Conclusion

In this article, we tried to elucidate the roles of the different Internet stakeholders and interpret the emerging interrelationships, focusing on the network neutrality concept. In addition, through a detailed econometric analysis on a series of parameters (including OTT revenues, ISP network investments, and ISP revenues), we revealed two important findings. First, the interests of OTTs and ISPs are not necessarily conflicting, since the economic gains of the OTTs are positively correlated with the ISP revenues and vice versa. Second, there is no clear motivation for the OTT providers to contribute financially to the network infrastructure, as CAPEX seem to stimulate the economic growth of ISPs, while being detrimental for OTT profits. Our research could serve as a starting point for future studies that will further clarify the interaction among the different entities in the evolving Internet ecosystem.

Acknowledgement

This work has been supported by AGAUR (2014 SGR 1551), CellFive (TEC2014-60130-P), and COMANDER (612257).

References

[1] T. Wu, "Network Neutrality, Broadband Discrimination," *J. Telecommunications and High Technology Law*, 141, 2003.

[2] S. Baldry, M. Steingröver, and M. Hessler, "The Rise of OTT Players: What is the Appropriate Regulatory Response?," *25th European Regional Int'l. Telecommunications Society (ITS) Conf.*, Brussels, Belgium, 2014.

[3] A. Lodhi *et al.*, "Complexities in Internet Peering: Understanding the "Black" in the "Black Art"," *IEEE Conf. Comp. Commun. (INFOCOM)*, Hong Kong, China, 2015, pp. 1778–86.

[4] R. T. B. Ma *et al.*, "Internet Economics: The Use of Shapley Value for ISP Settlement," *IEEE/ACM Trans. Net.*, vol. 18, no. 3, June 2010, pp. 775—87.

[5] B. Frank *et al.*, "Pushing CDN-ISP collaboration to the limit," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 3, July 2013, pp. 34–44.

[6] B. T. Olsen *et al.*, "Technoeconomic Evaluation of the Major Telecommunication Investment Options for European Players," *IEEE Network*, vol. 20, no. 4, July–Aug. 2006, pp. 6–15.

[7] A. Antonopoulos *et al.*, "Energy-Efficient Infrastructure Sharing in Multi-Operator Mobile Networks," *IEEE Commun. Mag.*, vol. 53, no. 5, May 2015, pp. 242–49.

[8] R. Kuhne, G. Huitema, and G. Carle, "Charging and Billing in Modern Communications Networks — A Comprehensive Survey of the State of the Art and Future Requirements," *IEEE Commun. Surveys & Tutorials*, vol. 14, no. 1, 1st Quarter 2012, pp. 170–92.

[9] H. H. Gharakheili, A. Vishwanath, and V. Sivaraman, "Perspectives on Net Neutrality and Internet Fast-Lanes," *ACM Comp. Commun. Rev.*, vol. 46, no. 1, Jan. 2016, pp. 64–69.

[10] R. T. B. Ma, J. C. S. Lui, and V. Misra, "Evolution of the Internet Economic Ecosystem," *IEEE/ACM Trans. Net.*, vol. 23, no. 1, Feb. 2015, pp. 85–98.

[11] R. T. B. Ma and V. Misra, "The Public Option: A Nonregulatory Alternative to Network Neutrality," *IEEE/ACM Trans. Net.*, vol. 21, no. 6, Dec. 2013, pp. 1866–79.

[12] H. H. Gharakheili, A. Vishwanath, and V. Sivaraman, "An Economic Model for a New Broadband Ecosystem Based on Fast and Slow Lanes," *IEEE Network*, vol. 30, no. 2, Mar.–Apr. 2016, pp. 26–31.

[13] P. Coucheney, P. Maille, and B. Tuffin, "Impact of Competition Between ISPs on the Net Neutrality Debate," *IEEE Trans. Network and Service Management*, vol. 10, no. 4, Dec. 2013, pp. 425–33.

[14] N. Kamiyama, "Feasibility Analysis of Content Charge by ISPs," *26th Int'l. Teletraffic Congress (ITC)*, Karlskrona, Sweden, Sept. 2014, pp. 1–9.

[15] D. Saucez, S. Secci, and C. Barakat, "On the Incentives and Incremental Deployments of ICN Technologies for OTT Services," *IEEE Network*, vol. 28, no. 3, May–June 2014, pp. 20–25.

Biographies

Angelos Antonopoulos (aantonopoulos@cttc.es) received his Ph.D. degree (cum laude) from the Signal Theory and Communications (TSC) Department of the Technical University of Catalonia (UPC) in 2012. He is currently a researcher in the Smart Energy Efficient Communication Technologies (SMARTECH) department of the Technological Telecommunications Centre of Catalonia (CTTC). He has authored over 70 publications in peer-reviewed journals and conferences on various topics, including energy efficient network planning and sharing, 5G wireless networks, cooperative communications, radio resource management, and network economics. He has participated in several European and Spanish national projects (e.g., GREENET, Green-T, CO2GREEN, etc.) and has served as an expert evaluator of research projects funded by the Romanian Government through the National Council for Scientific Research. He has been nominated as an exemplary reviewer for *IEEE Communications Letters*. He received the best paper award in IEEE GLOBECOM 2014, the best demo award in IEEE CAMAD 2014, the 1st prize in the IEEE ComSoc Student Competition (as a Mentor) and the EURACON best student paper award in EuCNC 2016.

Elli Kartsakli (ellik@iquadrat.com) received her Ph.D. in wireless telecommunications from UPC in 2012. She holds a degree in electrical and computer engineering from the National Technical University of Athens, Greece (2003), and an M.Sc. in mobile and satellite communications from the Univ. of Surrey, UK (2004). She has participated in several national and European projects and is currently a senior researcher at IQUADRAT. Her primary research interests include 5G networks and architectures, channel access protocols, and energy efficient schemes.

Chiara Perillo (chiara.perillo@bf.uzh.ch) received the M.Sc. degree in economic sciences (monetary and financial markets) from the University of Cagliari in 2014. She holds a B.Sc. degree in economics and finance from the same university (2011). She is currently an early stage researcher with the Department of Banking and Finance at the University of Zurich. Her research interests include empirical finance, econometric methods, and empirical analysis.

Christos Verikoukis (cveri@cttc.es) received the Ph.D. degree from UPC in 2000. He is currently a Head of the SMARTECH Department at CTTC and an adjunct professor at the University of Barcelona. He has published 105 journal papers and over 170 conference papers. He is also a co-author of three books, 14 chapters in other books, and two patents. He has participated in more than 30 competitive projects, and has served as the principal investigator of national projects in Greece and Spain. He has served as the General Chair (CAMAD12, CAMAD13 and CAMAD14, and CAMAD 17) and as TPC Co-Chair (IEEE Healthcom13, IEEE LATINCOM 2014 and IEEE ICT 2017) in several IEEE conferences. He has also served as the co-chair of the CQRM symposium at ICC 2015, ICC 2016, and Globecom 2017, and the chair of the eHealth symposium at Globecom 2015. Dr. Verikoukis received a best paper award at IEEE ICC 2011, IEEE GLOBECOM 2014 and 2015, EUCNC/EURACON 2016, and the EURASIP 2013 Best Paper Award for the *Journal on Advances in Signal Processing*. He is currently Chair of the IEEE ComSoc Technical Committee on Communication Systems Integration and Modeling (CSIM).

IEEE ICC™ 2018

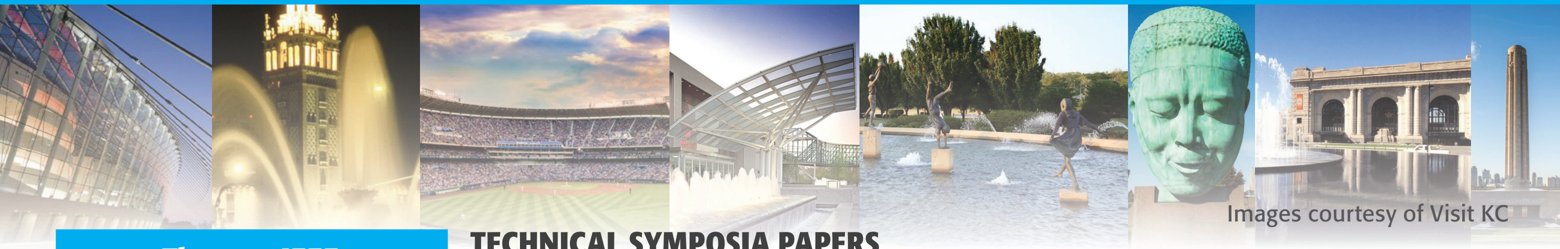
IEEE International Conference on Communications

Communications for Connecting Humanity

20-24 May 2018
Kansas City,
Missouri, USA

Back in the US after 15 years!

CALL FOR TECHNICAL & INDUSTRY SUBMISSIONS



Images courtesy of Visit KC

The 2018 IEEE International Conference on Communications (ICC) will include a Technical Program comprised of 13 specific symposia, tutorials and workshops as well as an Industry Program featuring panels, demonstrations, tutorials and workshops.

TECHNICAL SYMPOSIA PAPERS

Authors are invited to submit original technical papers in the following areas:

- Selected Areas in Communications
  - Access Systems and Networks
  - Big Data
  - Cloud Communications and Networks
  - Data Storage
  - E-Health
  - Internet of Things
  - Molecular, Biological and Multi-scale Communications
  - Smart Grid Communications
  - Powerline Communications
  - Social Networks
  - Satellite and Space Communications
  - Smart Cities
- Ad Hoc and Sensor Networking
- Cognitive Radio and Networking
- Communications and Information System Security
- Communications QoS, Reliability and Modelling
- Communications Software and Services
- Communication Theory
- Green Communications
- Next Generation Networking and Internet
- Optical Networks and Systems
- Signal Processing for Communications
- Wireless Communications
- Wireless Networking

Technical Workshop Proposals

Proposals are sought that emphasize current topics of particular interest to the community on the latest technical and business issues in communications and networking.

Technical Tutorial Proposals

Proposals are sought for new and emerging topics within the scope of communications.

IF&E Proposals

Proposals are sought that focus on latest topics, products and innovations of particular interest to industry and government in communications and networking.

Industry Demonstrations

Hardware and/or software demonstrations are sought that are meant to showcase new and innovative technology.

IMPORTANT DATES

Technical Symposia Papers
Due 15 October 2017

Technical Workshop Proposals
Due 15 July 2017

Technical Tutorials Proposals
Due 15 September 2017

IF&E Proposals
Due 10 November 2017

Industry Demonstrations
Due 5 January 2018

Organizing Committee

General Chairs
Andrzej Jajszczyk, AGH University of Science and Technology, Poland
Ron Marquardt, Sprint, USA

Executive Chair
Deep Medhi, University of Missouri-Kansas City, USA

Executive Vice Chair
Victor Frost, University of Kansas, USA

TPC Chair
Yi Qian, University of Nebraska-Lincoln, USA

TPC Vice Chairs
Rose Qiangyang Hu, Utah State University, USA
Lisandro Zambenedetti Granville, Federal University of Rio Grande de Sul, Brazil

Workshop Co-Chairs
Bala Natarajan, Kansas State University, USA
Byrav Ramamurtny, University of Nebraska-Lincoln, USA
Massimo Tornatore, Politecnico di Milano, Italy

Tutorials Co-Chairs
Tricha Anjali, International Institute of Information Technology, Bangalore, India
Caterina Scoglio, Kansas State University, USA
Rosa Zheng, Missouri University of Science & Technology, USA

IF&E Chair
Durga Satapathy, Sprint, USA

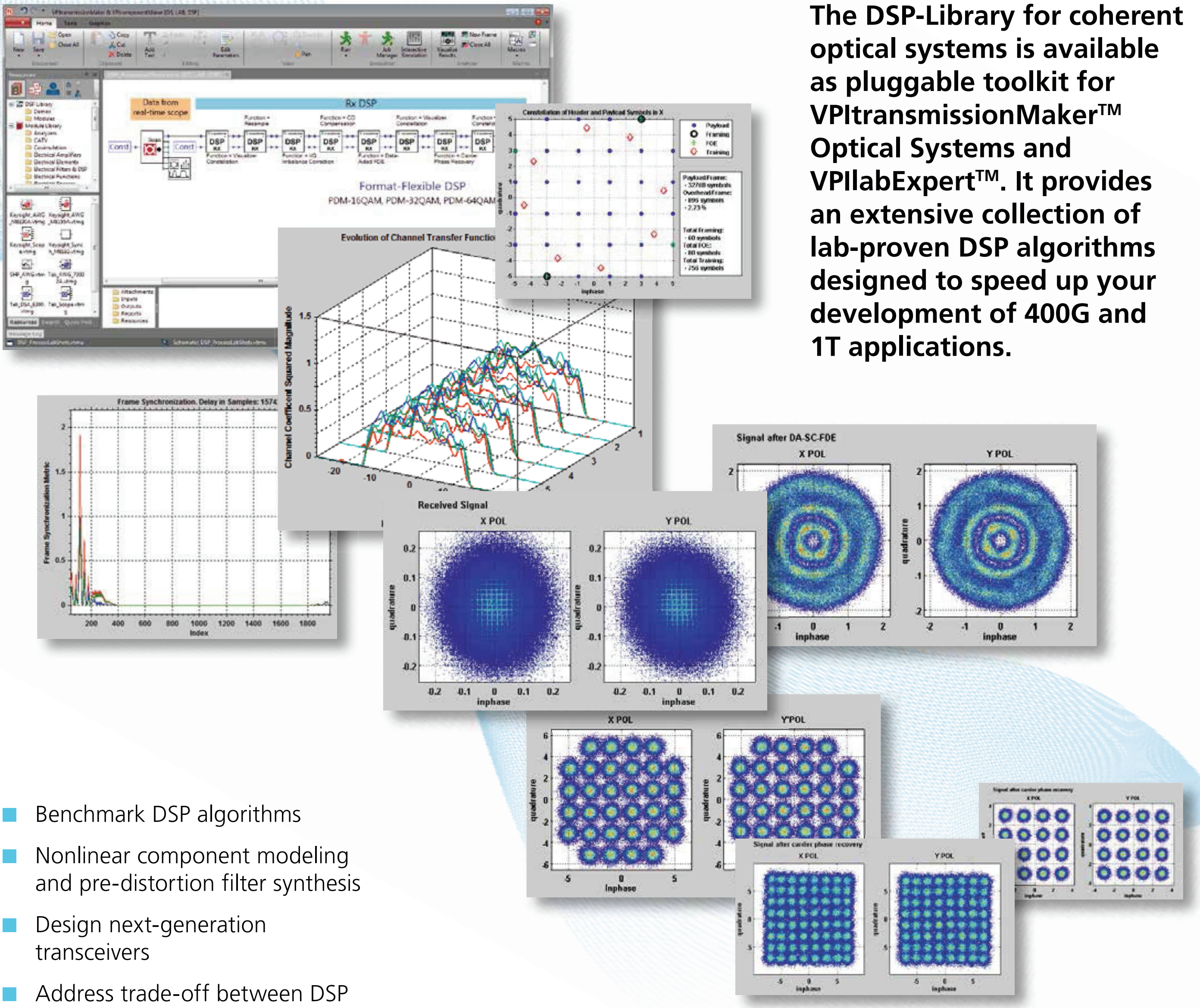
For more information, visit
http://icc2018.ieee-icc.org

IEEE

IEEE ComSoc™
IEEE Communications Society

Fraunhofer
Heinrich Hertz Institute

Ready-to-use DSP-Library for optical system simulations and experiments


Data from real-time scope
Rx DSP
Format-Flexible DSP
PDM-16QAM, PDM-32QAM, PDM-64QAM
Constellation of Header and Payload Symbols in X
Evolution of Channel Transfer Function
Channel Coefficient Squared Magnitude
Frame Synchronization. Delay in Samples: 1574
Frame Synchronization Metric
Index
Received Signal
Signal after DA-SC-FDE
Signal after carrier phase recovery
X POL
Y POL
inphase
quadrature

The DSP-Library for coherent optical systems is available as pluggable toolkit for VPItransmissionMaker™ Optical Systems and VPIlabExpert™. It provides an extensive collection of lab-proven DSP algorithms designed to speed up your development of 400G and 1T applications.

- Benchmark DSP algorithms
- Nonlinear component modeling and pre-distortion filter synthesis
- Design next-generation transceivers
- Address trade-off between DSP complexity and its performance
- Compare modulation formats
- System performance analysis
- Define component requirements

In Cooperation with:

VPIphotonics
DESIGN AUTOMATION

Further Information:
http://www.vpiphotonics.com/Tools/DSPLibrary/