日本ソフトウェア科学会
研究会資料シリーズ No.85
ISSN 1341-870X

Proceedings of Internet Conference 2018

Sponsored by
JSPS 163rd Committee on Internet Technology (ITRC)

Co-sponsored by
WIDE Project (WIDE)

Technically cosponsored by
Japan Society for Software Science and Technology SIG on Internet Technology (JSSST/ITECH)

In-cooperation with
ACM SIGUCCS Tokyo Chapter
IEEE Communications Society Japan Chapter
the Internet Society Japan Chapter (ISOC-JP)
Qshu-Bone Project (QBP)
Kyushu Giga POP Project (QGPOP)
Cyber Kansai Project (CKP)
IEICE Chugoku Chapter
IEICE Kansai Chapter
IPSJ Chugoku Chapter
IPSJ Kansai Chapter
IPSJ Special Interest Group on High Performance Computing (HPC)
IPSJ Special Interest Group on Internet and Operation Technology (IOT)
IPSJ Special Interest Group on system software and Operating System (OS)
IPSJ Special Interest Group on Distributed Processing System (DPS)
IPSJ Special Interest Group on UBIquitous computing systems (UBI)
IEICE Technical Committee on Internet Architecture (IA)
IEICE Technical Committee on Network Systems (NS)
Japan UNIX Society (JUS)

26th, 27th November, 2018
Akihabara Convention Hall, Tokyo, Japan

発行日　2018年11月26日

発行者　インターネットコンファレンス2018 実行委員会
日本学術振興会産学協力研究委員会インターネット技術第163委員会
WIDEプロジェクト

日本ソフトウェア科学会事務局
〒113-0032 東京都文京区弥生 2-4-16 学会センタービル内
日本ソフトウェア科学会
電話 03-5802-2060　FAX 03-5802-3007

日本ソフトウェア科学会
研究会資料シリーズ No.85
ISSN 1341-870X

Proceedings of Internet Conference 2018

Sponsored by
JSPS 163rd Committee on Internet Technology (ITRC)

Co-sponsored by
WIDE Project (WIDE)

Technically cosponsored by
Japan Society for Software Science and Technology SIG on Internet Technology (JSSST/ITECH)

In-cooperation with
ACM SIGUCCS Tokyo Chapter
IEEE Communications Society Japan Chapter
the Internet Society Japan Chapter (ISOC-JP)
Qshu-Bone Project (QBP)
Kyushu Giga POP Project (QGPOP)
Cyber Kansai Project (CKP)
IEICE Chugoku Chapter
IEICE Kansai Chapter
IPSJ Chugoku Chapter
IPSJ Kansai Chapter
IPSJ Special Interest Group on High Performance Computing (HPC)
IPSJ Special Interest Group on Internet and Operation Technology (IOT)
IPSJ Special Interest Group on system software and Operating System (OS)
IPSJ Special Interest Group on Distributed Processing System (DPS)
IPSJ Special Interest Group on UBIquitous computing systems (UBI)
IEICE Technical Committee on Internet Architecture (IA)
IEICE Technical Committee on Network Systems (NS)
Japan UNIX Society (JUS)

26th, 27th November, 2018
Akihabara Convention Hall, Tokyo, Japan

Program Committee and Organization Committee

Program Committee Chair

Teruaki Yokoyama (Kobe Institute of Computing / National Institute of Information and Communications Technology)

Program Committee

Yasuhiro Ohara (NTT Communications)
Katsushi Kobayashi (The University of Tokyo)
Nobuo Okashiwa (Kyoei Gakuen University)
Motoyuki Ohmori (Tottori University)
Keiichi Shima (IIJ Innovation Institute)

Organization Committee Chair

Hiroshi Mano (Koden Techno & Info)

Organization Committee

Motoyuki Ohmori (Tottori University)
Ismail Arai (Nara Institute of Science and Technology)
Minoru Ikebe (Oita University)
Yoshihisa Kawamoto (Osaka Gakuin University)
Katsushi Kobayashi (The University of Tokyo)
Toru Kondoh (Hiroshima University)
Yasuhiro Ohara (NTT communications)
Yasuo Okabe (Kyoto University)
Koji Okamura (Kyushu University)
Toshihiko Shimokawa (Kyushu Sangyou University)
Naomi Terada (National Institute of Information and Communications Technology)
Seiichi Yamamoto (The University of Tokyo)
Teruaki Yokoyama (Kobe Institute of Computing / National Institute of Information and Communications Technology)

TechnicalProgram Committee

Technical Program Committee Chair

Ohara Yasuhiro (NTT Communications)

Technical Program Committee

Hirochika Asai (Preferred Networks, inc.)
Rei Atarashi (IIJ Innovation Institute)
Razvan Beuran (Japan Advanced Institute of Science and Technology)
Romain Fontugne (IIJ Innovation Institute)
Kensuke Fukuda (National Institute of Informatics)
Hiroaki Harai (National Institute of Information and Communications Technology)
Tatsuya Jinmei (Infoblox)
Katsushi Kobayashi (The University of Tokyo)
Aya Matsui (IBM Japan)
Hiroki Matsutani (Keio University)
Rodney Van Meter (Keio University)
Yoshifumi Nishida (GE Global Research)
Masafumi Oe (National Astronomical Observatory of Japan)
Nobuo Ogashiwa (Kyoai Gakuen University)
Yasuhiro Ohara (NTT Communications Corporation)
Motoyuki Ohmori (Tottori University)
Kazunori Sugiura (Keio University)
Yojiro Uo (IIJ Innovation Institute)
Seiichi Yamamoto (The University of Tokyo)
Teruaki Yokoyama (Kobe Institute of Computing / National Institute of Information and Communications Technology)

Program

26th November 2018: 1st day

9:00-10:00 Invited Talk (1)
Internet Civilization
Jun Murai (Keio University)

10:00-10:15 Break

10:15-11:15 Invited Talk (2)
Designing the quantum Internet
Rodney Van Meter (Keio University)

11:15-12:45 Lunch

12:45-13:15 Paper Presentation (1)
サイバー演習の防御演習時におけるシナリオ進行の自動化システムの提案
井上 拓哉 (北陸先端科学技術大学院大学)
Razvan Beuran (北陸先端科学技術大学院大学)

13:15-13:45 Poster Lightening Talk (1)

13:45-14:45 Invited Talk (2)
Beyond Internet Research: By Systemy!
Michio Honda (NEC Labs Europe)

14:45-15:15 Break

15:15-15:45 Research Topic Exchange (1)
Traffy Waste - a smarter way to collect and management waste collection
Wasan Pattara-atikom (NECTEC)

15:45-16:45 Poster Lightening Talk (2)

16:45-17:30 Poster and Demonstration

27th November 2018: 2nd day

9:30-10:30 Invited Talk (4)
Internet Measurement, how to get the relativities right at scale
George Michaelson (APNIC Labs)

10:30-10:45 Break

10:45-11:45 Invited Talk (5)
Past, Present, and Future of DNS Resolution
Paul Vixie (Farsight Security)

11:45-13:15 Lunch break

13:15-13:45 Paper presentation (2)
Kamuee: An IP Packet Forwarding Engine for Multi-Hundred-Gigabit Software-based Networks
Yasuhiro Oharaa (NTT Communications Corporation)
Hiroki Shirokuraa (NTT Communications Corporation)
Abhik Datta Banika (NTT Communications Corporation)
Yudai Yamagishia (NTT Communications Corporation)
Kim Kyunghwan (Independent Engineer)

13:45-14:15 Research Topics Exchange (2)
The Cameroon Internet Shutdowns-Technological Framework
Ndenge Godden Zama (BlackedOut Africa)

14:15-15:15 Invited Talk (6)
Challenges and opportunities from large scale Internet measurement infrastructures
Emile Aben (RIPE NCC)

15:15-15:45 Break

15:45-16:45 Invited Talk (7)
Creating interactive experiences with Mixed Reality
Kelvin Cheng (Rakuten, Inc.)

16:45-17:15 Research Topics Exchange (3)
SmartSantander: an IoT-based Smart City Testbed
Juan Ramón Santana (University of Cantabria)

17:15-18:00 Poster and Demonstration

18:15-18:30 Closing

Table of Contents

Invited Talk

Research Topic Exchange

Table of Contents

Paper Presentation

Poster

Invited Talk (1)

Speaker: Jun Murai (Keio University)

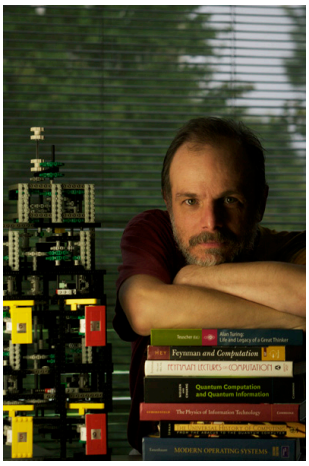
Title: Internet Civilization

Abstract: Some goals of the Internet design have been achieved; such as a global digital space, easy to access, wired and unwired and broadband.
A lot of not yet achieved; such as truly for everyone, full coverage, latency sensitivity, sophisticated distributed processing, things to be connected and eternal preservation of data.
When we define the world today as the `Internet Civilization', what we should work for some essential elements of the Civilization including the architecture of the Internet, coming and expecting technologies and social design would be discussed.

Biography: He received his Ph.D. in Computer Science, Keio University in 1987, majored in Computer Science, Computer Network and Computer Communication. He developed the Japan University UNIX Network (JUNET) in 1984, established WIDE Project in 1988, aiming to research and develop the computer networks.
He is a member of the Strategic Headquarters for the Promotion of an Advanced Information and Telecommunications Network Society (IT Strategic Headquarters), a member of the Cyber Security Policy Council, National center of Incident readiness and Strategy for Cybersecurity(NISC), Cabinet Secretariat, chairs and serves on many other governmental committees, and is active in numerous international scientific associations. He is known as the "Internet samurai" and, in Japan has also been called "the father of the Internet in Japan".
He was a Member of Internet Architecture Board (IAB) from 1993-1995, Board of trustee of Internet Society (ISOC) from 1997-2000, Board of Director of the Internet Corporation for Assigned Names and Numbers (ICANN) from1998-2000. He was inducted into The 2013 Internet Hall of Fame (Pioneer)/ 2011 IEEE Internet Award / 2005 Jonathan B. Postel Service Award.

Invited Talk (2)

Speaker: Rodney Van Meter (Keio University)



Title: Designing the quantum Internet

Abstract: The coming Quantum Internet will bring us new capabilities: advanced cryptographic functions, high-precision sensor networks for uses such as high-resolution astronomy, and secure distributed quantum computing. Experimental progress on the components for quantum repeaters is moving at a dizzying rate, and theorists have proposed various approaches to managing errors to create high-fidelity quantum entanglement. Building quantum networks presents different challenges from building quantum links. I will give an overview of these issues, then discuss the even more daunting challenge of creating a network of networks -- an internetwork -- and show how our simulations are guiding the design of a true quantum Internet.

Biography: Rodney Van Meter received a B.S. in engineering and applied science from the California Institute of Technology in 1986, an M.S. in computer engineering from the University of Southern California in 1991, and a Ph.D. in computer science from Keio University in 2006. His current research centers on quantum computer architecture and quantum networking. Other research interests include storage systems, networking, and post-Moore's Law computer architecture. He is now an Associate Professor of Environment and Information Studies at Keio University's Shonan Fujisawa Campus. He is the Vice Dean of the Graduate School of Media and Governance, and the Vice Center Chair of Keio's new Quantum Computing Center. Dr. Van Meter is a member of AAAS, ACM and IEEE.

Invited Talk (3)

Speaker: Michio Honda (NEC Labs Europe)

Title: Beyond Internet Research: Be Systemy!

Abstract: Research on the Internet has been a good topic for the last three decades, thereby having produced a lot of PhDs. Examples of the work include "new" architectures, measurement and various protocol extensions, such as mobility or multipath support, faster, flexible or secure signaling and better data multiplexing, to name but a few. As it turns out, the Internet has demonstrated its superior scalability, but not extensibility, resulting in ossification phrased as "HTTP as a narrow waist of the Internet", "IP options are not an option" and "Is it still possible to extend TCP?".
In this talk I suggest not to do research on the Internet itself anymore, and discuss possible interesting topics surrounding it. For example, its core technology could be a basis of solving general networking problems. Further, unlike phone lines, the Internet is still in use, introducing a lot of interesting workloads that stress various system components. I will review several topics which do or do not follow this line, and introduce some of our recent and ongoing work, including network stack, load balancer and storage stack.

Biography: Michio Honda is a senior researcher at NEC Labs Europe in Heidelberg, Germany. Before that, he was a software engineer at NetApp in Munich. He received his phd degree in 2012 at Keio University in Japan. He has worked on transport protocols, congestion control, middleboxes, network stacks, scalable software switch, and network/storage stack co-design for persistent memory. He has published in venues including USENIX NSDI and ATC, and ACM HotNets, SOSR, IMC, CCR and SoCC. He received IRTF/ISOC Applied Networking Research Prize in 2011, and best paper award at SOSR'15. He has served as a TPC member of several conferences, including USENIX ATC, and ACM SOSR and ANCS. He has been a contributer to the netmap framework, bringing the research results into the real world and making them available to everybody.

Invited Talk (4)

Speaker: George Michaelson (APNIC Labs)

Title: Internet Measurement, how to get the relativities right at scale

Abstract: APNIC is now eight years into a continuous measurement of IPv6 and DNS, based on web and in-game advertising, which crucially depends on random placement. We are increasingly aware of a worldwide problem establishing "ground truth" for the relative amounts of data seen from different Origin-AS. The talk explores our methodology and dimensions of concern for this question.

Biography: George Michaelson is currently APNIC's senior R&D scientist. Recently, he has been working on long-baseline DNS statistics, services logging, audit and analysis, and design and implementation of the Internet Number Resource Certification framework. George is a member of the BCS, and a founder member of the Australian chapter of the Internet Society. He participates regularly in IETF standardization meetings, and co-authors Request For Comment (RFC) documents, technical drafts, and conference and peer-review papers. George graduated from York University in 1982 with a BSc in Computer Science. His career in the United Kingdom and Australia has pursued research and development in computer science, networking, and systems administration.

Invited Talk (5)

Speaker: Paul Vixie (Farsight Security, Inc.)

Title: Past, Present, and Future of DNS Resolution

Abstract: The Domain Name System has been a critical enabler of Internet growth since its inception in 1987. In the decades since then, the DNS _resolution_ process has evolved from the LAN to the WAN, and to Anycast; it now includes DNSSEC _validation_, Extended DNS (EDNS) Client Subnet, larger message sizes, and I18N. The resolution processs has also been abused for surveillance, advertising insertion, and exfiltration. Today the DNS resolution process is poorly understood, and yet under forced revision. The trend is for DNS to be carried inside HTTPS where it cannot be monitored or controlled except by servers and clients themselves, and the dangers this will yield must be studied and discussed while the future remains flexible. Dr. Vixie (Keio, 2012) will describe the past and present of DNS, and discuss its likely near term future.

Biography: Paul Vixie was responsible for BIND from 1989 to 1999, and is the author of a dozen or so IETF RFC documents about DNS. He also started the first anti-spam company (MAPS), and was the founder and later president of the first U.S.-based commercial Internet Exchange (PAIX). Today he serves as CEO of Farsight Security, home of the Security Information Exchange (SIE) and the world's leading Passive DNS database (DNSDB). He managed the F-root DNS server from 1996 to 2012, and wrote the Cron software used on all UNIX-type computers today. He is also co-inventor of the DNS Response Rate Limiting (RRL) and Response Policy Zone (RPZ) feature-sets now in wide use to protect the operational Internet Domain Name System against online attacks. He received his Ph.D. from Keio University in 2011, and was inducted into the Internet Hall of Fame in 2014.

Invited Talk (6)

Speaker: Emile Aben (RIPE NCC)

Title: Challenges and opportunities from large scale Internet measurement infrastructures

Abstract: The RIPE NCC runs 2 large scale Internet measurement infrastructures, RIPE Atlas and RIPE RIS. In this presentation we'll dig deeper into these infrastructures, challenges in running them and analysing the data coming off of them.

Biography: I'm a system architect/research coordinator at the RIPE NCC, where I work in the science group. I'm a chemist by training, but have been working since 1998 on Internet related things, as a sysadmin, security consultant, web developer and researcher. I am interested in technology changes (like IPv6 deployment), Internet measurement, data analysis, data visualisation, sustainability and security. I'd like to bring research and operations closer together, ie. do research that is operationally relevant. When I'm not working I like to make music (electric guitar, bass and drums), do sports (swimming, (inline) skating, bouldering, soccer), and try to be a good parent.

Invited Talk (7)

Speaker: Kelvin Cheng (Rakuten Institute of Technology, Rakuten, Inc.)

Title: Creating interactive experiences with Mixed Reality

Abstract: The ubiquity of personal mobile devices enables users access to a wealth of online information and services at their fingertips. However, in terms of the interactive experience, there is still a distinct gap between what users physically see and touch at the physical location and the digital content that they are interacting on their mobile device. With the use of Mixed Reality (MR), physical and digital information spaces can be merged, in which digital content can be accessible directly on the physical objects. In this way, MR has the potential to enable a more immersive experience. In this talk, we explore how MR can enhance our daily experience in terms of shopping and exhibitions, and discusses the future of MR and the factors that need to be considered when designing Mixed Reality experiences going forward.

Biography: Dr Kelvin Cheng is currently a Research Scientist at Rakuten Institute of Technology, the R&D lab of Rakuten, Inc. He received his PhD in Computer Science from The University of Sydney. His previous affiliations were Keio-NUS CUTE Center, National University of Singapore, and CSIRO ICT Centre, Australia. He has extensive experience and expertise in the domain of interacting with large surfaces and multi-display environments using bare-hand interaction, and multi-touch devices, and more recently involved in use of augmented and mixed reality platforms to increase consumer experiences. He has previously lectured at the National University of Singapore on Mobile Interaction Design, as well as the use of interactive technologies such as iBeacons and Microsoft Kinect. At Rakuten, he has been involved in projects related to mixed reality shopping, FC Barcelona uniform collection exhibition, and 5G trial demonstrations.

Research Topic Exchange (1)

Speaker: Wasan Pattara-atikom (NECTEC)

Title:Traffy Waste - a smarter way to collect and management waste collection

Abstract: The logistics of waste collecting in Thailand is mostly performed manually with traditional route management without the help of information technology or IoT. Although tracking device are installed in some of the waste collecting vehicles, the tracked data were used for real-time location but the fleeting activities and performance were not yet analyzed and understood.
In this project, we proposed system where the trucks are equipped with a high frequency and high accuracy GNSS-sensor. With this sensors, we can achieve fours key objectives. The first objective is to provide real-time location and detailed activities of the waste collecting logistics such as collecting, in-a-traffic-jammed, or at-terminal. The second objective is to provide performance analytics of the collecting logistics such as truck utilization, the number of and duration of pick-ups per trip, and route trajectory. The third objectives is provide more effecient route recommendation to reduce fuel and operational costs. The location of waste pick-up are identified and extracted automatically using machine learning. The fourth objective is provide two-way communications with citizen using mobile application. Mobile application can be applied to enable the request-for-pick-up with a notification when the vehicle is near the requested pick-up location. This system can improve vehicle utilization, reduce operating cost, make the city cleaner.

Biography: Wasan Pattara-atikom is currently the Head of the Intelligent Transportation Systems Laboratory and Principal Researcher at NECTEC. He earned two Master degrees in Telecommunications and Business Administration, and a Doctor of Philosophy in Information Science from the University of Pittsburgh. His research interest focus on the area of Intelligent Transportation Systems (ITS) and Data Analytics Visualization.
He was responsible for the Traffic Information Dissemination Project (Traffy) which was selected as an outstanding project by the National Science and Technology Development Agency and the Ministry of Science and Technology in 2008 and 2009. Dr. Pattara-atikom published over 80 peer reviewed articles, submitted 7 patents.. He was recently listed on top scientists in Thai Institutions according to their Google Scholar Profiles. He also served in the office the Ministry of Science and Technology in 2012, and in the office of the Deputy Prime Minister in 2013.

Research Topic Exchange (2)

Speaker: Ndenge Godden Zama (BlackedOut Africa)

Title: The Cameroon Internet Shutdowns-Technological Framework

Abstract: Research shows that one of the following techniques, DNS-based blocking, IP address blocking, URL-Based blocking, Search Engine platform censoring, and Deep packet inspection based blocking have been used in the past to block or filter content by governments in Africa. The Cameroon internet shutdown was a total black out. This paper seeks for illustrate how it was done and the impact on innovation and development.

Biography: Ndenge Godden Zama is a Data and Internet Freedom Activist and the founder of BlackedOut Africa. BlackedOut Africa is a organization that fosters policy and advocates for a free, open and accessible internet in Africa. The organization has engaged the private sector and governments all over Africa to keep an open access to the internet as a key catalyst for growth and development. Zama has a BSc in Management and an MA in ICT in Education. He is also research fellow at the African Center for Research, Development and Climate Change

Research Topic Exchange (3)

Speaker: Juan Ramón Santana (University of Cantabria)

Title: SmartSantander: an IoT-based Smart City Testbed

Abstract: SmartSantander testbed is an urban deployment of Internet of Things (IoT) devices in the city of Santander. Such massive deployment has a two-fold approach. On the one hand, it allows to the scientific community to experiment with IoT technologies and Smart City services, including prototyping deployments in a real urban scenario. On the other hand, it also provides a set of services to the citizens, including parking management or environmental monitoring, among others. During the presentation, the SmartSantander testbed will be described, including the architecture and communication technologies used in the testbed. Furthermore, the presentation will also include the current efforts to federate testbeds from Europe and the rest of the world, and the movement to implement open standards for the Smart City through FIWARE.

Biography: Juan Ramón Santana obtained his MSc in Telecommunication Engineering at the University of Cantabria in 2010. He is currently working as research fellow in the Network Planning and Mobile Communications Laboratory, a telecommunication research group from the same university. Prior to his current occupation, he stayed for six months in the Silent Herdsman startup company (formerly known as Embedded Technology Solutions), a spin-off from the University of Strathclyde in Glasgow, working on IoT solutions for the cattle industry. Since then, he has been involved in several Smart City international projects, from which we can highlight SmartSantander, working on tasks such as the integration and deployment of the SmartSantander communication infrastructure; Over the Air firmware update implementation; and the testbed platform development.
Beyond SmartSantander, he has been also working in other EU projects, such as EAR-IT, FIESTA-IoT or FESTIVAL, an EU-Japan collaborative project, working on platforms to federate international testbeds. Finally, it is worth mentioning that he has collaborated in more than 20 publications, including conferences, journals and book chapters.

Paper Presentation

サイバー演習の防御演習時におけるシナリオ進行の自動化システムの提案

井上　拓哉 [1], Razvan Beuran[1]

Abstract

サイバー人材の育成だけでなく，多くの人にサイバー空間の脅威について知ってもらうためにもサイバー演習が必要となる。その中でも，脅威の認識および対処のため，サイバー攻撃に対する防御演習が重要である。しかし，既存の防御演習には教育に関する機能が欠けている。本稿では，防御演習によるセキュリティ教育を普及させるため，防御演習の進行と指導を自動化するためのシステム”DeTMan”を提案する。まずは既存の防御演習について考察し，教育システムとして用いるにあたり必要な機能について検討する。次に，検討された機能をいかにして実装するか検討し，概念実装を行う。最後に，実装した DeTMan について，教育システムとしての防御演習の観点から検証する。

Keywords: サイバーセキュリティ, サイバー演習, 防御演習, サイバーレンジ

1. はじめに

現在，技術の発展に伴い，企業にとっても個人にとってもセキュリティリスクは深刻なものとなっている。しかし，IT 技術を適切に扱うための教育が発展に追いついていない。セキュリティ人材の不足 [1] や，社会のセキュリティに関する意識 [2] が大きな問題となっている。そのため，本稿では，セキュリティ人材の育成に大きな役割を果たしている防御演習に注目した。防御演習では，受講者は与えられた環境に対し実行される攻撃に対処することで，サイバー攻撃の脅威と対策について学ぶ。

無料で参加可能な防御演習として，Hardening[3] と Micro Hardening[4] がある。防御演習の開催には，セキュリティに関する高度な専門知識が必要となる。そのため，現在の主流な防御演習は，特定の人物や団体によって運営されるに留まっている。結果として，防御演習の機会は限られてしまい，需要を満たせていない。防御演習を普及には，より簡単に開催できることが重要である。また，現在の防御演習は演習による訓練に重きが置かれており，演習の参加者に対する指導なども提供される。

本稿では，Hardening と Micro Hardening を参考として防御演習による教育システムについて考察する。そして，教育としての防御演習を提供するサイバー防御演習進行管理システム”DeTMan”を提案する。

[1]北陸先端科学技術大学院大学

2. 既存の防御演習

教育としての防御演習を提供するにあたり，既存の防御演習である Hardening と Micro Hardening を例に考察する。

2.1. *Hardening* の紹介

Hardening とは，Web Application Security Forum(WASForum) が主催するセキュリティ堅牢化の競技大会である。Hardening では，脆弱性を持つ EC サイトの運営して，チーム対抗で売り上げを競う。Hardening における売り上げとは，クローラーによる EC サイトでの自動購入によって成立する。運営側からの攻撃に対して，参加者はシステムを堅牢化することでサービスを維持し，売り上げの最大化を目指す。

防御演習の開催には，図 1 に示す 3 つのステップが必要となる。想定する攻撃者の行動や攻撃パターンなどを策定する (1) 防御演習シナリオの作成，シナリオに沿った防御演習の環境を構築する (2) 防御演習環境の構築，参加者に攻撃を与える (3) 防御演習の実施である。

防御演習シナリオの作成は，防御演習において最も重要な部分である。どのような演習を行うのか，どのような攻撃を行うのか (どのような脆弱性を埋め込むのか)，どのように進行するのかについて決定する。Hardening では様々な分野の専門家が集まり，演習シナリオを作成する。

防御演習環境の構築では，作成した防御演習シナリオに基づいて演習環境を作成する。防御演習では実際に攻撃するため，仮想環境で演習を実施する必

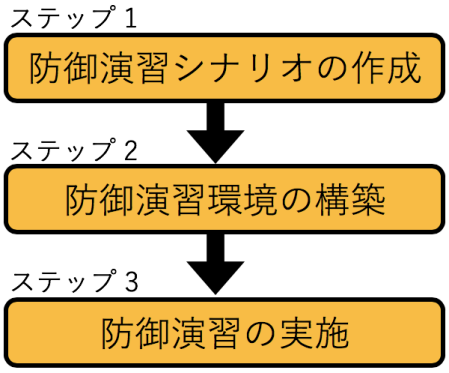


図 1: 防御演習の 3 ステップ

要がある。Hardening における演習環境は，Alfons[5] と呼ばれる環境構築システムを用いて演習環境を構築している。

最後に，実際に演習を実施する。Hardening では，運営側の攻撃はすべて手動で実行される。運営側は会場に設置されたカメラで参加者の様子を確認しながら攻撃するため，チームに合わせて攻撃を調整することができる。そのため，順調に進んでいるチームにはより高度な攻撃を，低調なチームには攻撃をしないといった，柔軟な進行が可能である。

また，Hardening では実環境を想定した仮想ネットワークを使用するだけでなく，顧客対応や上役への報告，さらにはマーケットプレイスと呼ばれる企業のサービス導入なども競技の中で行われる。技術だけなく，サイバーセキュリティに携わる上で必要になると考えられる様々な知識やスキルを学ぶことが可能である。

2.2. Micro Hardening

Micro Hardening は株式会社川口設計の川口洋氏によって提供される競技形式の勉強会である。Hardening Project のサブプロジェクトとして誕生した。Hardening と比較して，カジュアルな演習になっている。競技時間は 1 セット 45 分であり，1 度の演習で，3 セット以上同じ内容を繰り返す。簡素化のために，顧客対応や上役への報告といった，技術的な対策以外の要素は省かれている。

Micro Hardening は防御演習環境の構築から攻撃の実行まで，全て自動化されている。そのため，川口氏 1 人だけでも運営が可能であり，Micro Hardening は日本各地で頻繁に開催されている。

Micro Hardening では，攻撃の実行は時間によって動作するタイムドリブン方式により自動化されている。そのため，すべてのセットにおいて同じタイミングで同じ攻撃が実行される。参加者は，攻撃について調査し，次のセットで対策を施すといった試行錯誤を，1 度の演習の中で繰り返すことができる。

3. 既存の防御演習が持つ教育上の課題

3.1. 開催の困難さ

シナリオの作成は演習において，最も重要なステップである。シナリオは，演習の目的に応じて検討する必要がある。そのため，専門家の知見を活用して作成する。また，作成したシナリオは，演習の目的が同じであれば再利用が可能である。

演習環境の構築には，CyRIS[6] などのサイバーレンジ構築ツールや，Ansible などの構成管理ツールを用いることを推奨する。環境の構築が簡単になるだけでなく，同じ演習を開催する場合に演習環境を簡単に構築できるためである。上記のツール群は，演環境の構築に設定ファイルを用いる。設定ファイルの再利用により，何度でも同じ演習環境を作成することができる。

演習において実行される攻撃は多種多様であるため，様々な分野の技術者が必要となる。また，演習の参加者は，運営側の人数よりも多い。そのため，手動で攻撃する場合には運営側に重い負担がかかる。教育の一環として防御演習を普及させるためには，特別な人材が必要であることや運営側に重い負担を強いることは問題である。

シナリオと演習環境の構築は再利用が可能である。しかし，演習の実施は再利用ができない。そのため，防御演習において演習の実施が負担となっている。

Micro Hardening は，演習の進行を自動化することによりたった 1 人でも開催可能である。そのため，防御演習を教育として提供する場合には自動化による負担軽減が必要である。

3.2. 受講者に応じた演習の進行

Micro Hardening はタイムドリブン方式により自動化されている。しかし，タイムドリブン方式では受講者全員に同じ内容の演習を提供することになる。そのため，演習が基準としているレベルから離れている人は対象外となってしまう。

教育としての防御演習では，Hardening のように受講者それぞれの状況に合わせて演習を自動で進行させる必要がある。

3.3. 受講者に対する指導の不足

DoS 攻撃のような例外を除き，サイバー攻撃とはコンピュータの所有者に気付かれないように実行される。Hardening や Micro Hardening は，実際の環境に近い形で行うため，攻撃されたことに気付くこともまた，演習の一部である。そのため，攻撃に気付

くことなく演習が終了する事態も十分に想定される。これは，教育としては問題である。

文部科学省の高等学校学習要項 [7] において，「基礎的・基本的な知識及び技能を確実に習得させ，これらを活用して課題を解決するために必要な思考力，判断力，表現力その他の能力をはぐくむとともに，主体的に学習に取り組む態度を養い，個性を生かす教育の充実に努めなければならない」と記載されている。また，「個々の生徒の特性等の的確な把握に努め，その伸長を図ること」と記載されている。つまり，教育として防御演習を行うには，受講者の進捗に応じて，セキュリティについて確実に習得させることが必要である。

Hardening や Micro Hardening では演習後の解説により，どのような攻撃されたのかについては知ることができる。しかし，演習後の解説ではいつ・どのように攻撃されたのかはわからない。そのため，攻撃された際にコンピュータはどのような反応を示すのか，知ることができない。教育としては，演習中に指導する必要がある。

また，攻撃について教えるだけでは不十分である。例えば，受講者がどのログファイルを確認するべきなのか知らなければそれ以上の情報について調査できない。加えて，ログ保存の設定が適切でなければ，そもそも確認するための情報が存在しない。

防御演習における指導では，演習中に，どのような攻撃かだけでなく，検知方法や対策まで指導する必要がある。

4. 教育としての防御演習のための自動化システム

防御演習を教育として普及させる場合には，運営の負担を軽くするために進行の自動化が重要である。しかし，既存の自動化された防御演習である Micro Hardening は，Hardening にあった柔軟さが失われている。また，既存の防御演習は教育的な指導が不足している。そのため，教育としての防御演習には以下の点が重要になる。

- 受講者の状況に応じた進行をどのように自動化するか
- どのように指導するか
- どのような振り返りを提供するか

4.1. 受講者の状況に応じた進行

受講者の進捗に合わせるためには，受講者が攻撃に対処するまで待機することと，受講者の状況に応じて異なる攻撃を実行することが必要になる。攻撃を待機させるシステムは，以下の 4 つの機能により実装できると考えられる。

- 進行の独立
- 進行の分岐
- 死活監視機能
- イベントドリブン方式による進行機能

4.1.1. 進行の独立

受講者の状況に応じて進行させるためには，演習の進行を受講者ごとに独立させなければならない。進行を独立させることにより，他の受講者による影響を受けることなく自分でペースで進行可能になる。

4.1.2. 進行の分岐

受講者の状況は，チームごとに異なる。受講者に応じた進行には，順調な受講者にはより高度な攻撃を，低調な受講者には簡単な攻撃や防御に失敗した攻撃を繰り返すといった進行が必要である。そのため，演習の進行を分岐させる必要がある。

4.1.3. 死活監視機能

受講者が管理するネットワークにおいてサービスが停止している場合は，何らかのアクシデントが発生していると考えられる。そのため，サービスが停止している場合は攻撃をするべきではない。

4.1.4. イベントドリブン方式による進行機能

死活監視機能だけでは，受講者が緊急の処置としてサービスを再起動をしただけの場合でも攻撃を再開する。攻撃に対する調査する時間を確保するためにも，死活監視機能以外にも攻撃を待機する機能が必要になる。

イベントドリブン方式を用いることにより，特定のイベントが発生するまで，進行を待機させることが可能である。

4.2. 指導方法

演習によるセキュリティ教育として以下の 3 点が必要だと考えられる。

- 段階的な攻撃の通知
- 受講者の理解を確認
- 演習の振り返り

4.2.1. 段階的な攻撃の通知

本稿の目的は，演習による教育である。1 度に攻撃に関するすべての情報を通知しては受講者自らが考える機会が失われてしまう。そのため，異常の発生・調査すべきファイル・実行された攻撃・対策方法と段階的に受講者に通知する機能が必要である。

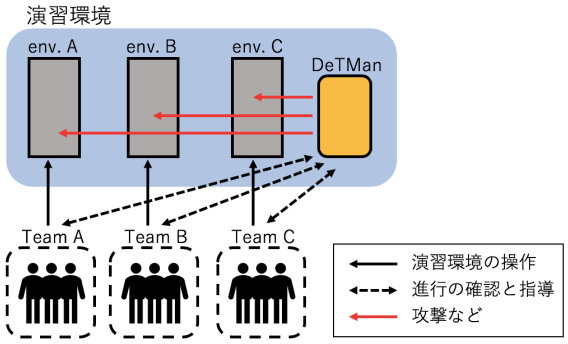


図 2: DeTMan の概要

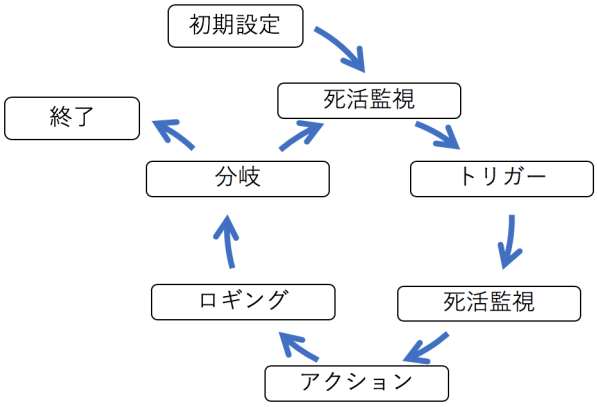


図 3: DeTMan の動作

4.2.2. 受講者の理解を確認

システムに言われるがままに，何も理解せず演習を進めるような事態は避けなければならない。受講者の理解状況を確認する機能が必要である。

4.2.3. 演習の振り返り

攻撃について調査するには以下の情報が必要になる。

- いつ攻撃したのか
- どのような攻撃をしたのか
- 結果はどうだったのか

攻撃のタイミングに関する情報により，各種のログを調査する際に，調査する範囲を限定することが可能である。実行された攻撃の種類による情報により，何を調査するべきか特定可能である。攻撃の成否に関する情報により，受講者の施した対策の効果を知ることが可能である。

また，攻撃に関して任意に調査可能にするため，受講者が任意のタイミングでこれらの情報を確認できる必要がある。

5. 提案システム DeTMan

4 章では，防御演習による教育システムが持つべき機能が明らかとなった。本章では，防御演習自動進行管理システム DeTMan(Defense Training progress Management system) の概念実装を行う。

5.1. *DeTMan* の概要

DeTMan の概要を図 2 に示す。DeTMan は，攻撃と指導をすべて自動で実行する。そのため，演習の実施において，運営側に人を必要としない。

DeTMan は演習の進行を独立させるために，受講者数 (用意された演習環境数) と同数の子プロセスを生成する。演習の進行は子プロセスが担う。1 つの環境について，1 つの子プロセスを割り当てることにより，進行の独立に成功した。

DeTMan の動作を図 3 に示す。DeTMan の各動作において，初期設定以外は子プロセスが担当する。以降の項において，DeTMan の各動作について説明する。

5.2. *DeTMan* の動作

5.2.1. 初期設定

DeTMan ではチームファイルとシナリオファイルという 2 つの設定ファイルを使用する。チームファイルには受講者名 (チーム名) と攻撃対象となるサーバについて記述する。シナリオファイルには演習の具体的な流れについて記述する。シナリオファイルのサンプルを図 4 に示す。

DeTMan において，シナリオファイル内の 1 つのまとまりをステップと呼ぶ。つまり，シナリオファイルとはステップの集合体である。

DeTMan はアクションの成否に応じて動作を変化させることが可能である。シナリオファイルの success と failure には，アクションが成功または失敗した場合について記述されている。また，DeTMan には防御演習に競技要素の持ち込みを可能とするため，ポイント機能がある。success と failure には，next に次のステップを，point にアクション終了後に加減算されるポイントを記述する。

初期設定では，まず，2 つの設定ファイルを読み込む。シナリオファイルでは，trigger，success，failure は省略可能である。ファイルの読み込み後，省略された部分を補完し，DeTMan 用に再構成する。次に，データベースに関する設定を行う。防御演習の進行

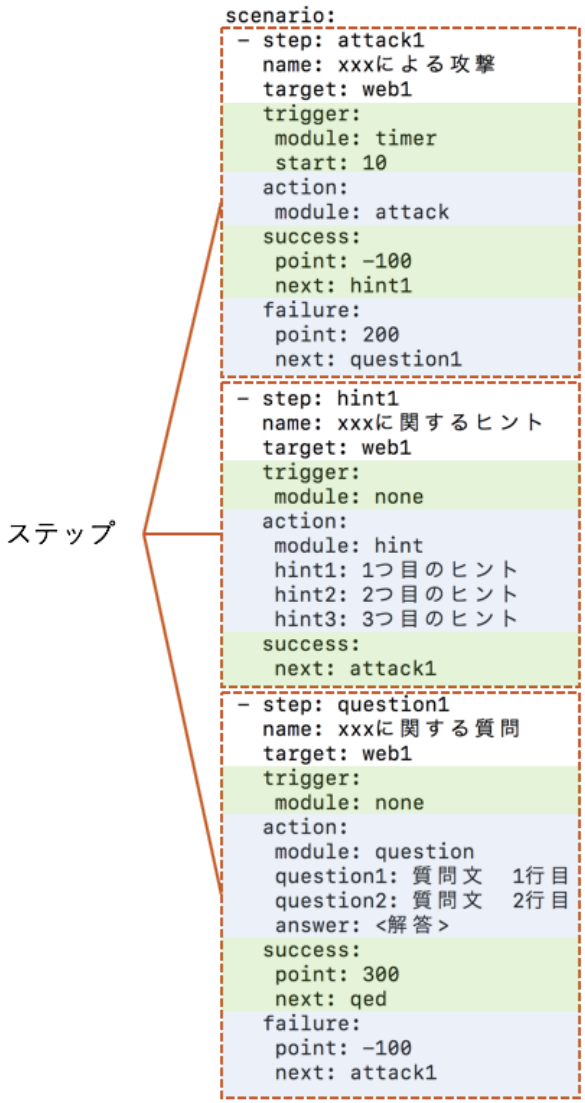


図 4: シナリオファイルのサンプル

状況を，外部から参照可能とするためにデータベースを用いる。データベースには，以下の 4 つのテーブルを作成する。

- 演習環境に関する情報を保管する state テーブル
- 現在実行中のステップとポイントの情報を保管する progress テーブル
- アクションの実行結果に関する情報を保管する log テーブル
- シナリオに関する情報を保管する scenario テーブル
- 受講者への指導に関する情報を保管する board テーブル

最後に，子プロセスを生成する。以降は，子プロセスの終了を待機する。

5.2.2. 死活監視

DeTMan は，通常は ping を用いたネットワークの疎通確認により死活監視を行う。しかし，ネットワークの疎通情報のみでは，攻撃を待機するかの判断には不十分である。そのため，チームファイルに記された演習環境において，サーバ名に web という単語が含まれていた場合に，HTTP リクエストによる HTTP サーバの動作確認を行う。今後，動作確認を行うサービスを追加する。死活監視の結果を state テーブルに書き込む。疎通確認または動作確認が失敗した場合，死活監視が成功するまで攻撃を待機する。

死活監視は，トリガー前と，アクション前に実行する。これにより，トリガーとアクションの際には，演習環境が正常に動作していることが保証される。

5.2.3. トリガー

DeTMan では，イベントドリブン方式におけるイベントをトリガーと呼ぶ。現ステップの trigger に記されたモジュールが実行され，トリガーが発生するまで動作を待機する。

DeTMan は，攻撃を待機するための機能としてイベントドリブン方式を採用した。しかし，非同期 I/O を採用すると同時に複数の攻撃に対処しなければならない事態が想定される。そのため，1 度に 1 つのトリガーのみを待機する。フィッシングやリバースシェルのようなマルウェアを用いる演習は困難になるが，DeTMan では対応しない。

5.2.4. アクション

アクションにおいて，DeTMan は受講者に対して能動的に動作する。現ステップの action に記されたモジュールが実行される。主なアクションとして，攻

撃の実行を想定している。他にも，メールの送信なども想定している。

アクションは，実行の結果とコメントを戻り値として返す。実行の結果は，アクションの成否であり，success または failure である。しかし，例えば，アクションとして実行された攻撃の成功と，メール送信の成功は意味が反対である。攻撃が成功した場合は，受講者に防御をさせるために，次の攻撃には進まない。メール送信が成功した場合は，次の攻撃に進む。次節で説明するロギングされたデータを受講者が見た場合に混乱する。そのため，コメントとして受講者から見た際の結果について返すことにより，混乱を防ぐ。

また，演習中の指導もアクションとして行う。DeTMan はアクションの成否により進行を分岐する。そのため，攻撃が成功した場合にのみに実行するアクションとして，指導が可能である。

DeTMan では，指導のためのアクションとして hint と question を用意した。hint では，board テーブルに情報を格納する。図 4 を参考に説明する。本シナリオでは，hint1，hint2，hint3 の 3 つのヒントが記述されている。DeTMan は，ステップ hint1 が 1 度目に実行された場合，hint1 を格納する。2 度目に実行された場合には hint2 を，3 度目以降は hint3 を格納する。これにより，段階的に受講者に対して情報が提示することが可能である。

question も同様に，board テーブルに情報を格納する。本シナリオでは，question1，question2 の 2 つの質問が記述されている。hint とは異なり，すべての情報が同時に格納される。board テーブルには，hint や question などを区別可能な情報も格納されるため，区別可能である。

5.2.5. ロギング

DeTMan は，実行したアクションに関する情報を log テーブルに格納する。ポイントはロギングの際に計算される。格納される情報は以下の 4 つである。

- 現在の時間
- チーム名
- 実行されたステップの step
- アクションのコメント
- アクション実行後のポイント

また，演習終了後には log テーブルのデータを csv ファイル形式により出力することができる。

表 1: 防御演習の比較

	Hardening	Micro Hardening	DeTMan
演習の用途	訓練	訓練	教育
進行の柔軟さ	◎	×	○
開催難易度	×	◎	○
リアリティ	◎	○	×

5.2.6. 分岐

アクションの成否に応じて，実行ステップを変更する。実行ステップが qed であった場合は演習を終了し，そうでなければ死活監視を行う。そして，実行ステップが qed となるまで，繰り返す。

5.3. WEB UI

データベース内のデータを可視化するために WEB UI を用いる。これにより，受講者の状況を確認可能である。

また，board テーブルのデータも可視化するため，指導にも WEB UI を用いる。データを可視化するだけでなく，未解答の question があった場合は解答フォームも作成する。この解答フォームを用いることにより question に対して解答が可能である。

5.4. 既存の防御演習との違い

表 1 に，Hardening，Micro Hardening，DeTMan の違いをまとめる。既存の防御演習と DeTMan の最大の違いは，演習の用途である。既存の防御演習はスキルアップを目的として開催されるが，DeTMan は教育を目的とする。そのため，DeTMan は既存の防御演習にはなかった教育用の機能を複数持つ。

DeTMan は簡単に柔軟な演習を実施可能だが，演習中に指導を行うためリアリティは損なわれている。実際のインシデントは，DeTMan のように攻撃について教えてくれることはない。そのため，DeTMan は初心者を対象とする。

6. 自動化された演習進行および教育の実証実験

6.1. 実証実験の概要

図 4 に示すシナリオファイルを用いて DeTMan の実証実験を行う。シナリオのフローを，図 5 に示す。本シナリオでは攻撃 attack1 を実行し，防御に成功した場合，attack1 に関して受講者に質問する。攻撃 attack1 の防御に失敗した場合はヒントを表示し，受講者に質問する。受講者が正しく解答した場合は演習を終了し，誤答した場合はもう 1 度攻撃を実行する。

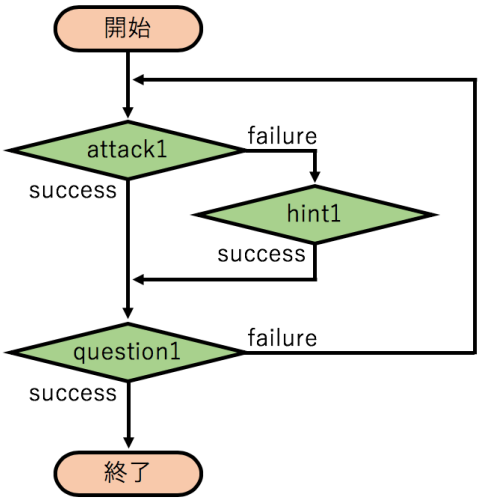


図 5: 演習フロー

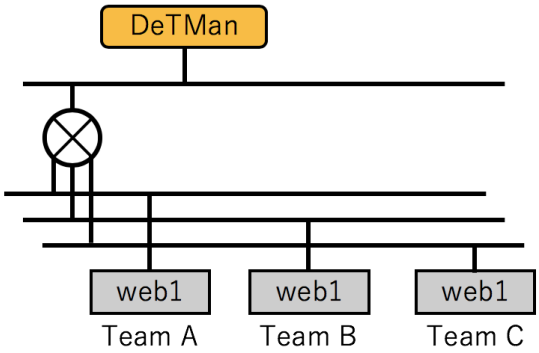


図 6: 演習環境

また，本実験は，図 6 に示す演習環境で実施した。DeTMan は 3 チームが管理する合計 3 台のサーバに対して攻撃する。

Team A は上級者，Team B は中級者，Team C は初心者とし，表 2 のように進行すると定義する。質問には，防御に成功した場合に正答する。例として，Team B は 2 度目に防御に成功するため，質問に 1 度目は誤答し，2 度目に正答する。また，Team B は 2 度目の attack1 におけるトリガー待機中に HTTP サーバが停止し，Team C の演習終了後に再起動する。

6.2. 受講者の状況に応じた進行

演習中のある瞬間において，WEB UI により可視化した progress テーブルの一部を図 7 に示す。progress テーブルより，チームごとに異なる進行をしていることが分かる。

表 2: 各チームにおける演習の進行

チーム	進行
Team A	1 度目に防御成功
Team B	2 度目に防御成功
Team C	5 度目に防御成功

Team	Step
Team A	Complete
Team B	attack1
Team C	question1

図 7: progress テーブルの抜粋

図 8，図 9 に，log テーブルから Team A，Team B に関して抜粋したものを示す。

Team A に関する log テーブル内の情報より，Team A は 1 度目から攻撃 attack1 の防御に成功したため，hint は実行されていないことが分かる。また，question1 にも 1 度で正答したため，そのまま演習が終了した。Team B に関する log テーブル内の情報より，Team B は 1 度目の攻撃 attack1 の防御に失敗したため，次に hint1 が実行されていることが分かる。また，1 度目の question1 に誤答したため，attack1 が再度実行されている。2 度目の attack1 は防御に成功したため，hint1 は 1 度しか実行されていない。DeTMan は，受講者の進捗に応じて進行を変化させてることが確認できた。

また，Team B に関する log テーブル内の情報より，2 度目の attack1 はシナリオ通りならば question1 の 10 秒後に実行されるはずであるが実行されていないことが分かる。しかし，ログには Unavailable と記され，attack1 はシナリオ通りに実行されていない。

この時の可視化された state テーブルから Team B に関して抜粋したものを図 10 に示す。Team B では question1 の後に HTTP サーバが停止したため，DeTMan は HTTP サーバが再起動されるまで攻撃を待機した。

Time	Team	Step	Comment	Point
2018/11/16 07:52:36	Team A	sys	START	1000
2018/11/16 07:52:46	Team A	attack1	Defense success	1200
2018/11/16 07:52:56	Team A	question1	Success	1500
2018/11/16 07:52:56	Team A	sys	COMPLETE	1500

図 8: log テーブルの Team A に関する抜粋

Time	Team	Step	Comment	Point
2018/11/16 07:52:36	Team B	sys	START	1000
2018/11/16 07:52:46	Team B	attack1	Defense failed	900
2018/11/16 07:52:46	Team B	hint1	Success	900
2018/11/16 07:53:06	Team B	question1	Wrong...	800
2018/11/16 07:53:16	Team B	sys	Unavailable	0
2018/11/16 07:54:57	Team B	attack1	Defense success	1000
2018/11/16 07:55:17	Team B	question1	Success	1300
2018/11/16 07:55:17	Team B	sys	COMPLETE	1300

図 9: log テーブルの Team B に関する抜粋

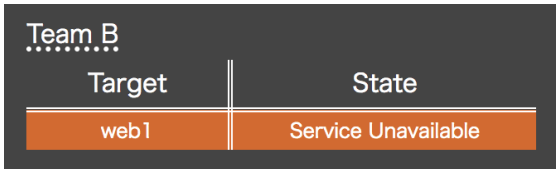


図 10: state テーブルの Team B に関する抜粋

以上の点より，DeTMan は，受講者の状況に応じて演習を進行させていることが確認できた。

6.3. 受講者に対する指導

図 11 に，WEB UI により可視化した board テーブルから Team C に関して抜粋したものの一部を示す。また，図 12 に WEB UI の解答フォームを示す。WEB UI では，対話的であることを強調するため，SNS ライクに表示する。緑のコメントがヒント，赤のコメントが質問，橙色のコメントは受講者の解答である。図 11 において，hint1 が何度目の実行かにより，表示されるヒントが変化していることが確認できる。

DeTMan では，対話的に受講者に対して指導可能であることが確認できた。

6.4. 演習の振り返り

図 7 や図 10 は演習中に撮影したものである。つまり，演習中に受講者は WEB UI により，現在実行されているステップや演習環境の稼働状況について知ることができる。過去に実行された攻撃についても，図 8 や図 9 のように参照可能である。受講者は，WEB UI に表示される情報を参考に演習環境の調査を行う。

また，これらの情報はデータベースに保管されているため，演習後にも残る。DeTMan は log テーブルの内容をチーム別にレポートとして出力する機能を持つため，演習後も振り返りが可能である。

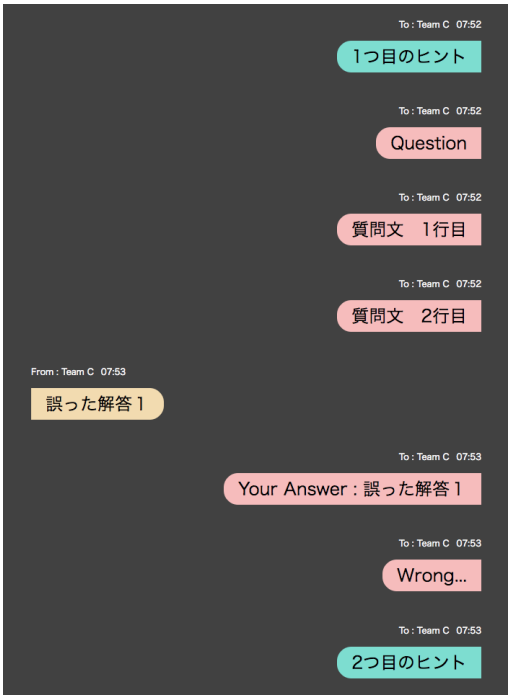


図 11: board テーブルの Team C に関する抜粋

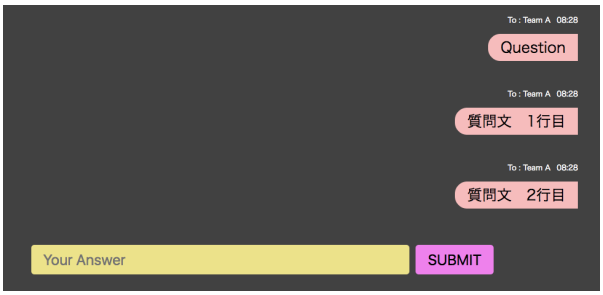


図 12: board テーブルの解答フォーム

7. CyRIS との連携

我々のプロジェクトではサイバー演習統合フレームワーク CyTRON[8] を開発している。その中にサイバーレンジを作成するツール，CyRIS が含まれている。CyRIS において，受講者それぞれに割り当てられる演習環境をインスタンスと呼ぶ。CyRIS は，KVM を用いて作成された基本となるインスタンスを必要なだけ複製し，サイバーレンジを作成する。図 13 に CyRIS と DeTMan の連携について示す。

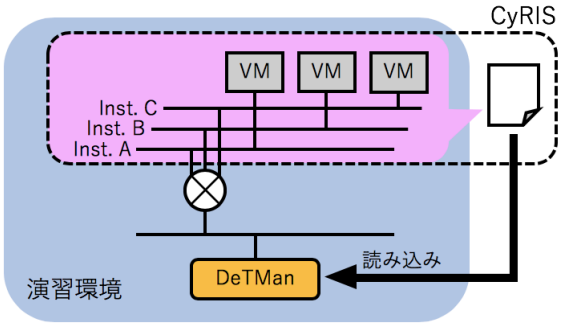


図 13: CyRIS との連携

CyRIS は，作成した仮想マシンに一定のルールに基づいて IP アドレスを割り振り，サイバーレンジの環境情報が記述された yaml ファイルを出力する。このファイルには，DeTMan のチームファイルに必要となる，攻撃対象の IP アドレスなどの情報が格納されており，DeTMan はチームファイルの代わりにこの yaml ファイルを利用可能である。本機能により，CyRIS を用いて演習環境を作成した場合は，チームファイルを作成する手間を省略できる。

8. 今後の課題

8.1. 攻撃側の *IP* アドレスが固定

現在の DeTMan には，攻撃側の IP アドレスを変更する機能がないため，受講者のファイアウォールによる対策により，攻撃がすべて防がれてしまう。 そのため，攻撃者の IP アドレスをランダムで変更する機能が必要である。

8.2. ステップ間の連携

現在の DeTMan は 1 つ 1 つステップが独立してため，あるステップにより情報を奪取しても，別のステップではその情報を活かすことができない。そのため，ステップ同士を連携させるために，攻撃によって得られた情報を保管するための手段が必要になる。

8.3. WEB UI のアクセス制限

現在の WEB UI では，WEB UI にアクセスしたすべての人物が，すべての情報にアクセス可能である。そのため，進行に関する情報が他の受講者に公開されるだけでなく，Board ページで別の受講者に対する質問に解答することも可能である。何らかの形でアクセスを制限することにより，他の受講者に関する情報を閲覧できないようにする必要がある。

9. さいごに

本稿では，既存の防御演習である Hardenig と Micro Hardening を参考に，教育としての防御演習について考察した。考察を元に，教育としての防御演習に必要な機能をどのように実装するか検討した。そして，自動で防御演習の進行と教育をするためのシステム”DeTMan”を実装した。DeTMan が，教育としての防御演習に必要な機能を満たしているか検証した。今後は，DeTMan の完成度を高め，実際に演習を実施することにより検証を継続する。

謝辞

本研究は JSPS 科研費 17K00478 の助成を受けたものです。

参考文献

[1] 経済産業省. IT 人材の最新動向と将来推計に関する調査結果. <http://www.meti.go.jp/press/2016/06/20160610002/20160610002-7.pdf>.
[2] 諏訪博彦, 原賢, 関良明. 情報セキュリティ行動モデルの構築—人はなぜセキュリティ行動をしないのか. 情報処理学会論文誌, Vol. 53, No. 9, pp. 2204–2212, September 2012.
[3] Hardening project. <http://wasforum.jp/hardening-project/>.
[4] 川口 洋. <https://microhardening.connpass.com>.
[5] 安田真悟. Alfons: A mimetic network environment construction system. In *11th EAI International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*, Jun 2016.
[6] Razvan Beuran, Pham Cuong, Tang Thanh Dat, Ken ichi Chinen, Yasuo Tan, and Yoichi Shinoda. Cybersecurity education and training support system: Cyris. *IEICE Transactions on Information and Systems*, Vol. E101-D, No. 3, pp. 740–749, March 2018.
[7] 文部科学省. 高等学校学習指導要領. <http://www.mext.go.jp/component/a_menu/education/micro_detail/__icsFiles/afieldfile/2011/03/30/1304427_002.pdf>.
[8] Razvan Beuran, Tang Thanh Dat, Pham Cuong, Ken ichi Chinen, Yasuo Tan, and Yoichi Shinoda. Integrated framework for hands-on cybersecurity training: Cytrone. In *Elsevier Computers & Security*, Vol. 78C, pp. 43–59, June 2018.

Kamuee: An IP Packet Forwarding Engine for Multi-Hundred-Gigabit Software-based Networks

Yasuhiro Ohara[a], Hiroki Shirokura[a], Abhik Datta Banik[a], Yudai Yamagishi[a], Kim Kyunghwan[b]

[a]*NTT Communications Corporation*
[b]*Independent Engineer*

Abstract

In Software Defined Network (SDN) and Network Function Virtualization (NFV) era, extensible, flexible, and yet high-performance software packet forwarding capability is desirable, as the core functionality of the future Internet. In this paper we present Kamuee: a software IP packet forwarding engine, and its core router version that builds only on commodity hardware. By scaling the number of forwarding CPU cores, Kamuee supports multiple 100GbE interfaces, with near wire-rate traffic. In our benchmarks that use two Intel Xeon Platinum 8180 processors, Kamuee could forward 349.60 Gbps 512B-sized random destination traffic in a BGP full route environment. Further intralaboratory evaluation resulted in 292.17 Mpps 64B forwarding capability, showing the potential of very high-speed software router.

Keywords: NFV, SDN, Software Router, DPDK

1. Introduction

The rapid emergence of services based on technologies such as cloud computing and AI-driven robotics, underlines the desperate requirement for next generation networks, which are more flexible and able to keep pace with changes in usage and capacity. Software-defined networking (SDN), network functions virtualization (NFV), and network virtualization (NV) are the solutions which address this requirement and provide novel methodologies to design, build and operate next generation networks. Thanks to the recent breakthroughs like off-the-shelf hardware or whitebox networking, a massive paradigm shift in networking technology finally took place by decoupling software from the hardware, so that it is no longer constrained by the box that delivers it. SDN and NFV have become indispensable for all telecommunication service providers to (1) ***Drive Innovation*** by creating new types of applications, on-demand services and business models (2) ***Deliver Agility and Flexibility*** by enabling rapid deployment of new applications, services and infrastructure to quickly fulfill their changing requirements (3) ***Reduce OpEX*** by enabling automation and algorithm control through increased programmability of network elements to make it simple to design, deploy, manage and scale networks and (4) ***Reduce CapEx*** by enabling network functions to run on commodity off-the-shelf (COTS) hardware.

Therefore, software-defined networking is not only good for the network, but for the business as well. With SDN, the network can be made programmable. A deeper look into network programmability reveals that it involves both the control plane and the data plane and that both are valuable in containing costs and enabling business growth. Data path programmability offers a platform for rapidly deploying a new service or modifying an existing one. This is particularly important to service providers in fields like security, for example, to mitigate a massive DDoS attack affecting hundreds of vulnerable servers across the globe.

Today, both control and data plane programmability provided by software-based solutions are the desirable characteristics of network services and devices. An integral part of this is software based processing of data packets. But without high performance, software packet processing cannot leverage the potential benefits of SDN and NFV. Recent unprecedented leaps in NFV with performance reaching up to 100 Gbps [1, 2], make the need for software routers supporting speeds above 100 Gbps even more imperative. Keeping this in mind, we endeavored to develop a high performance software IP packet forwarding engine called ***Kamuee*** with comprehensive Layer 3 (routing) functionalities. Kamuee

successfully integrates the capabilities of DPDK (Data Plane Development Kit) [3] with Poptrie [4] to achieve stable and reliable high speed software IP packet routing. Kamuee is a fully fledged software router that can run on COTS PC Server and is capable of forwarding 344.28 Gbps random destination IP traffic with Ethernet frame size greater than 512B and approx. 600,000 full routes.

In this paper we present the design, implementation and evaluation of Kamuee. Section 2 describes the related research. Section 3 highlights the design and implementation methodology of Kamuee. Section 4 elaborates the performance evaluation of Kamuee. Kamuee was used as one of the core routers in Interop Tokyo 2018 and the relevant experience has been shared in section 5. Section 6 concludes the paper.

2. Related work

Achieving high performance in software routing implementations has been a major challenge over the last two decades and has lead to some cutting-edge advances in this field. Slowness of Linux network stack performance has become increasingly relevant issue over the years because of the exponential increase in the amount of data that is being transferred over networks and the corresponding workloads. Even the widespread use of 10 GbE network cards could not resolve this issue because of some bottlenecks in Linux kernel itself that prevent packets from being quickly processed. There have been many attempts to circumvent these bottlenecks with kernel bypass techniques that enable packet processing without involving the Linux network stack such that the application running in the user space communicates directly with networking device. Intel's DPDK [3] is one such solution which takes care of the packet forwarding performance bottleneck. DPDK leverages existing Intel Processor technologies like SIMD instructions (Singles Instruction Multiple Data), Huge-pages memory, multiple memory channels and caching to provide packet processing acceleration with its own libraries. Recent innovation like Poptrie addresses another major bottleneck of IP routing lookup. Poptrie [4] leverages the population count instruction to give the indirect indices to the descendant nodes in order to keep the small memory footprint within the CPU cache and enables extremely high speed IP lookup. Kamuee harnesses (1) packet processing acceleration of DPDK (2) high-speed packet lookup provided by Poptrie and (3) parallel processing for its superior performance. Read-Copy Update (RCU) [5] has been used to achieve the latter, for concurrency control owing to its lock/synchronization specialization.

Kamuee-Zero [6] presents the routing table mechanisms of the previous version of this implementation. The paper shows that 1) in order to achieve near wire-rate performance in 40GbE, more than three queues per port are necessary, 2) the difference in throughput performance between NUMA-aware and NUMA-nonaware is not large (if not negligible), and 3) performance exceeding one hundred Gbps can be achieved using software based router, with four 40GbE interfaces and 128B short packet random traffic. Kamuee-Zero did not have routing capability as it did not support any routing protocol. In contrast, Kamuee (the version presented in this paper) supports all major routing protocols such as BGP, OSPF, and RIP, and other required functions that are necessary to function as a basic router, such as ARP, VLAN, and statistics counter. Also, while Kamuee-Zero supports only 40GbE network hardware, Kamuee can additionally support 100GbE hardware as well.

While NFV tackles the problems posed by legacy proprietary middleboxes [7] by leveraging virtualization technologies to implement network functions (NFs) on commodity hardware, the advantages of NFV come with some downsides [8] as software-based NFs can potentially introduce significant performance overheads. Several research works have been done to address the performance drawback of software based NFV. ClickNP [9] offloads software logic onto programmable hardware like FPGA to accelerate individual NFs. NetBricks [10] runs NFs on a single CPU core instead of virtual machines and containers to improve NF performance. ClickOS [11], DPDK [3] and NetVM [12] optimize and accelerate packet processing from the network hardware to and between virtual machines. Recently, there have been some astounding progress in NFV performance in the past couple of years. Metron [1] realizes high performance NFV service chains at the emerging and extremely challenging link speeds at 100 Gbps using commodity hardware, while significantly reducing latency. Andromeda [2], Google Cloud Platform's network virtualization stack demonstrates that an OS bypass software data-path provides performance competitive with hardware, achieving 32.8Gb/s using a single core. Some other advancements in this area include SafeBricks [13] protecting NFs in cloud environments, a novel programming interface forNon-Volatile Main Memory called PASTE [14] , NFV resource manager ResQ [15] supporting high performance in multi-tenant NFV clusters, and, highly scalable and resilient general purpose L2 switching software FBOSS [16] capable of

running on commodity hardware.

3. Design and implementation

The main motivation of Kamuee design is the following: if the software is good and simple, we should be able to get a good performance out of a good hardware. Furthermore, we should be able to increase the overall performance by adding more hardware resources (i.e., CPU cores). To achieve this, Kamuee was designed and implemented as simple as possible, in the belief that any complexity may lead to performance degradation (i.e., KISS principle).

Kamuee employs the run-to-completion model [17] rather than pipe-line model: we adopted this model because the run-to-completion model is suggested as a better-performance model in a past work called RouteBricks [18]. By employing the run-to-completion model, we can scale-out the packet forwarding process over the multiple CPU cores, enabling the design goal of increased performance by increased hardware.

Some design policies are inspired by others' work. RCU is utilized in Linux kernel and also in some DPDK applications [19]. Use of Tap devices [20, 21] or KNI interface [22] to map the physical NICs in Linux to connect to the open-source software, is a well-known approach and is also supported by VPP [23].

CPU core (equivalently, CPU time) is a very precious resource for our purpose: if we have spare CPU core, we could gain more performance. Thus, we assemble the slow tasks (i.e., the tasks that do not need high performance), such as netlink, RIB, acl, tap, arp, and vty (i.e., virtual terminal), on one CPU core, to avoid wasting CPU core. To achieve this, we used Lthread library [24] included in the DPDK source (located under `examples/performance-thread`). Furthermore, since the function call interface are the same, we can change native DPDK thread to lthread thread and vice versa, enabling balance in performance. For example, because our experience suggested to speed them, in our current recommended setting, rib-manager, tap-manager, and snmp-manager run as the native DPDK threads, consuming one CPU core for each.

Figure 1 illustrates the internal structure of the Kamuee software router. In the fundamental DPDK concepts, Network Interface Cards (NICs) belongs to either Linux space or DPDK space. The figure happens to show the case that two NICs belong to Linux, and four NICs belong to DPDK, but it is configurable. The packets received in DPDK NICs are distributed to a specific CPU core by the RSS/multiqueue technology of the NIC. The assigned CPU core is running a "forwarder"

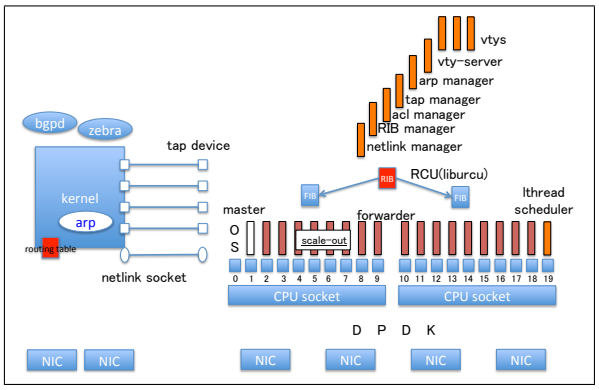


Figure 1: Kamuee internal structure: the ordinary linux space with two NICs and the DPDK space with four attached NICs are illustrated on left and right, respectively. Two CPU sockets are shown on the right, with most of the CPU cores running "forwarder" DPDK thread, enabling *scale-out* of packet forwarding tasks on the multiple cores. The routing table (RIB) is compiled as FIB using Poptrie, and distributed to each CPU socket's L3 cache using RCU. The manager threads such as ones dealing with netlink, tap, and vty, are launched on the right-most core using the "lthread" library. DPDK-attached physical Kamuee NICs are 1-to-1 mapped on the tap devices in the Linux space so that the routing protocol open-source software (e.g., Quagga) can run on the Kamuee NICs. The routes calculated by the routing daemons are first installed in the Linux kernel, and then propagated to Kamuee RIB using netlink socket/messages.

thread of Kamuee, that is built around the DPDK thread and occupies the CPU core. The packet, if it is to be forwarded, is solely handled by that CPU core, without the need of using the other CPU cores; hence the run-to-completion model. The forwarder thread processes the protocol headers such as Ethernet and IP, looks up the routing table (labeled "FIB" in the figure), and then directly forward the packet to the other NIC to emit the packet. Thanks to the Poptrie [4] that can compress some hundreds of thousands of routes into a few megabytes memory footprint, the routing table lookup can be completed in some CPU L3 cache accesses, avoiding a lot of main memory accesses. This simple run-to-completion forwarding process without main memory access, together with the parallelization (in other words, scale-out) on the number of CPU cores, is the main reason of high-performance of Kamuee. This is the key design of Kamuee, and is the main contribution of this paper.

The control protocol packets and the self-destined packets (i.e., the packets that are to be received by the Kamuee host itself) are handled as follows: by routing table lookup in the forwarder, the packet is indicated to be passed to the Linux part of the system. Then, the packet is passed to the "tap manager" (shown top-right in Figure 1). The tap manager delivers the packet to the

```
0.0.0.0/2 nexthop: 192.85.1.3 port: 2 flags:
64.0.0.0/2 nexthop: 193.85.1.3 port: 0 flags:
128.0.0.0/2 nexthop: 194.85.1.3 port: 8 flags:
192.0.0.0/2 nexthop: 195.85.1.3 port: 6 flags:
```

Figure 2: Four default routes or "`default4`"

Linux kernel via the correspondent tap device that is 1-to-1 mapped to the receiving NIC. In this way, the Linux kernel, and thus the Linux user processes, can receive the packet from DPDK attached NICs without problem. The routing protocol daemons use this mechanism: for example, Quagga bgpd (shown top-left in the figure) receives the BGP packets, processes them, and installs the calculated BGP routes in the Linux kernel. The newly installed routes are notified through the Netlink mechanism to the Kamuee's "netlink manager" (shown top-right in the figure). The "rib manager" is informed of the new routes by the netlink manager, and the rib manager installs it in the main routing table (labeled "RIB" in the figure), and produces the FIB using the Poptrie algorithm. One FIB for each CPU socket is prepared to properly support the CPU cache mechanism.

4. Evaluation

4.1. Benchmark setup

In our benchmark method, we measure the performance of the Device-Under-Test by sending the random-destination IP traffic from the network test: Spirent, by having the DUT forward back the traffic, and then by counting the packets returned to the network tester. To achieve this, we install in the DUT four prefixes that cover all IPv4 address space, so that the DUT can return all the packets that it could forward, back to the Spirent tester machine, without causing "route not found" error. We refer to the four prefixes that are shown in Figure 2 as "Four default routes" or "default4".

Wherever BGP full routes are used in the benchmark, the snapshot taken in NTT Communications' TestBed on 2016/12/12 has been used, in addition to Four default routes. The snapshot includes 612,916 prefixes.

If not specified explicitly, the compiler optimization option defaults to "-O3".

We generally focus on the achieved bandwidth rather than transaction performance (i.e., packet per second or pps). The factor limiting the performance is generally

Table 1: Hardware specification of Kamuee

Hardware Type	Product Name
Chassis	SYS-7049GP-TRT
Motherboard	Supermicro X11DPG-QT
CPU	Intel Platinum 8180 ×2
Memory	DDR4-2133 16GB ×12 = 192GB
NIC	Mellanox Connect-X5 100GbE Dual-Port ×5
	Intel X710 10GbE Quad-Port ×1

Table 2: Softwares used in Kamuee

Software Package	Version
OS	Ubuntu 16.04.5 amd64
DPDK	17.11
userspace-rcu	0.9.3
Quagga	0.99.24
net-snmp	6.0.1-2

either transaction performance or the bandwidth of subsystems, such as CPU core, QPI, and PCIe. Since we want to understand the overall performance of the system as a whole, and since the transaction performance can be calculated from the bandwidth performance, we generally focus on bandwidth performance, unless we have a specific interest.

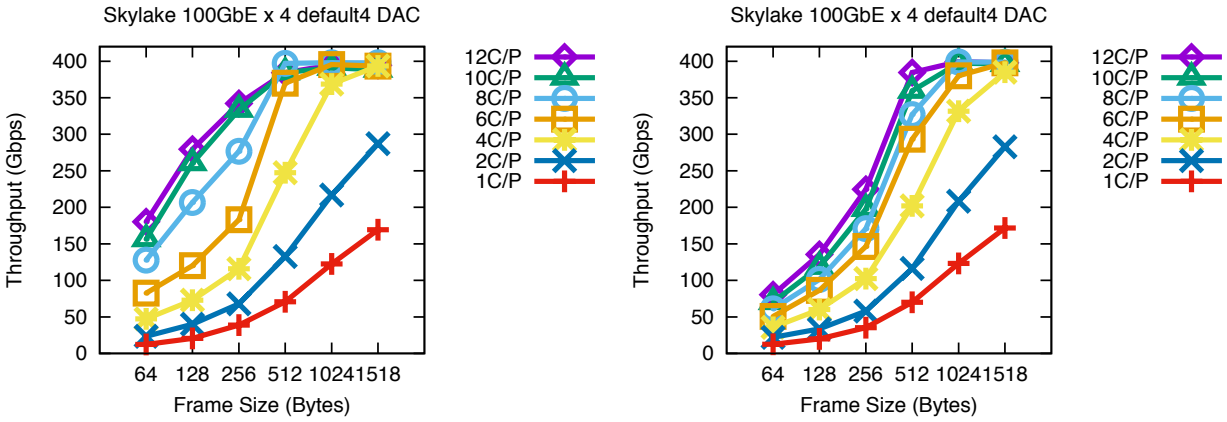
Our version of the software demonstrated a significant fluctuation of the performance over time. We have measured the performance as the average of five samples. It should be noted that our measurement may not cover the significant period of the fluctuation. Sometimes, after 30-40 seconds, Kamuee exhibits performance degradation from 394 Gbps to 383 Gbps in the same setting (this is shown later in Figure 4).

Table 1, 2 lists the hardware and software specification, respectively. Quagga was used to provide BGP4/4+, OSPF and OSPFv2 functionalities on Kamuee.

4.2. Overall Throughput in Bandwidth (BPS)

Figure 3a shows the bandwidth throughput performance of Kamuee when given wire-rate random traffic from all four 100GbE interfaces. The figure compares the performance on one core per port (1C/P) through twelve cores per port (12C/P). It shows that Kamuee successfully increases its performance as the number of CPU cores increase. With 64B shortest Ethernet frame, one core/port setting and twelve cores/port setting exhibit 12.20 Gbps and 180.18 Gbps, respectively. With 1518B longest frame, 12C/P demonstrate almost wire-rate of 394.36 Gbps.

The setting of Figure 3a is normal Linux connected routes plus "`default4`", leaving the Local Loopback Address 127/8 destined to the Linux host's upper layer.



(a) LoopbackTap (b) LoopbackBlackhole

Figure 3: Throughput with/without Tap Route

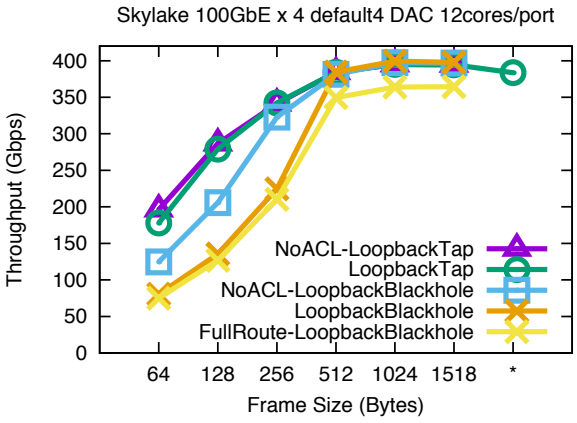


Figure 4: With/Without Tap Route. After 30 or 40 seconds with 1518B 400 Gbps traffic, With-Tap-Route degrades its performance slightly (shown in * mark in the x-axis). Except the one labeled with `FullRoute`, the route-setting is `default4`.

Since we launch fully-random-destination traffic from the tester to the Kamuee, the traffic destined to 127/8 fills and overloads the TAP socket, and sometimes cuts the BGP session through it. To avoid such problem, we installed 127/8 as blackhole route, and unexpectedly it lowered the throughput performance for some unknown reason. The performance is shown in Figure 3b. Since traffic without the TAP destined ones are more realistic, this can be deemed as the real performance value. We can infer the root cause of the performance degradation such that some bug or problem lies in Kamuee or DPDK library. In either way, Figure 3a can be recognized as the potential value of the software router: no matter what bug the root cause is, the performance value of Figure 3a shows the potential capability of the software router, if we trust the reliable Spirent tester.

Figure 4 compares the two settings of Local Loopback Address that are either directed to TAP or Blackhole. With four 100GbE I/Fs, twelve cores per port is the best setting we can get, and the figure only shows the case. 64B case degrades from 180.18 Gbps (at LoopbackTap) to 80.52 Gbps (at LoopbackBlackhole).

We integrated ACL functionality into Kamuee, by incorporating the DPDK ACL library. Our ACL implementation in the benchmark test did not include *ANY* entry, and just searched in the ACL list only to find the default ACL entry. Even in that setting, the ACL function exhibits a significant performance degradation. We show the overhead of the ACL functionality: NoACL-LoopbackBlackhole outperforms LoopbackBlackhole significantly. This suggest that by improving the ACL functionality, we may be able to reduce the performance degradation in the future.

Also we show the impact of FullRoute: comparing LoopbackBlackhole (equipped with `default4`) and FullRoute-LoopbackBlackhole, we can see the impact of holding and looking up the BGP full routes is not large, thanks to the Poptrie technology.

4.3. Overall Throughput in Transaction (PPS)

Figure 5a shows the transaction performance values in Packet Per Second (PPS), in contrast to the theoretical limit labeled as “Wire-rate”. Figure 5b illustrates the achievement rate against the ideal wire-rate. In 64B case, the most realistic setting, i.e., FullRoute-LoopbackBlackhole, exhibits 111.42 Mpps (achievement rate: 0.19), while NoACL-LoopbackTap for refer-

ence shows 292.17 Mpps (achievement rate: 0.49). The NoACL-LoopbackTap's reference value is not a bad value, but the realistic FullRoute-LoopbackBlackhole is not surprisingly fast, and gives a moderate speed performance value.

4.4. Effect of compiler optimization

Figure 6 shows the effect of compiler optimization on the performance of LoopbackTap setting. It demonstrates that gcc optimization option -O1 and above are roughly the same performance when the gcc version is (Ubuntu 5.4.0-6ubuntu1 16.04.9) 5.4.0 20160609. Further discussion such as which optimization option impacts the most is future work.

4.5. Microflow: the Benchmark of a Single IP Flow

Table 3 gives a list of latency measurements that is conducted for five minutes or more. Microflow means the single IP session flow, so the RSS (Receive-Side Scaling) of NIC cannot split the traffic onto multiple cores. We have two NUMA types (Same or Cross, meaning whether the test traffic needs to come accross the different NUMA nodes), and six Ethernet frame sizes.

Overall, around 5 Mpps is performed for the single core forwarding performance. No visible difference in latency was observed between NUMA types of the traffic. The latency is in average 20-30 microseconds for the Ethernet frame longer than 512B, but it was larger (such as 254 and 333 microseconds) in the Ethernet frame shorter than 256B. We suspect that the effect of PCI's Max Read Request Size is involved [25].

4.6. Packet loss

10 Gbps traffic was measured to test the packet loss rate in the not-so-heavy traffic load. This time the IP destination address field is randomized so that the RSS of NIC can split traffic to multiple cores. The 10 Gbps traffic was forwarded without major problem regardless of NUMA type. For the duration of five minutes, a rather small number of frames are dropped such as less than 10,000 frames (Table 4). It suggests the packet loss rate is significantly low to support the real traffic.

4.7. Benchmark for Virtualized Function

In order to investigate the bandwidth performance of Kamuee in virtual environment, VMs have been created using KVM with the same configuration as physical environment. Benchmarking has been done using similar test environment and default4 routes as described earlier. The VMs set up virtual CPUs with similar NUMA configuration and number of cores as the physical environment, with each virtual core using "vcpupin" such that the virtual cores do not operate on the same physical core. NICs are directly connected with SR-IOV using PCI-passthrough.

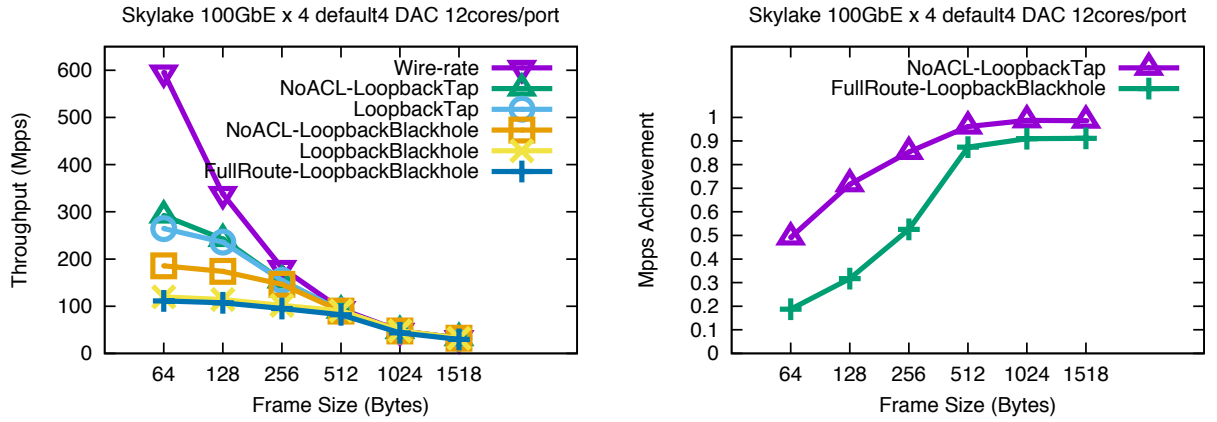
Figure 7 compares the performance of two cores per port (2C/P) through twelve cores per port (12C/P). Here, for packets longer than 1024B, a strange phenomenon is observed: the performance improves on reducing number of cores. The cause of this phenomenon can be IOMMU. The intel_iommu is an option related to DMA for enabling PCI-passthrough in Intel CPUs. Further investigation of the difference in performance due to the presence or absence of intel_iommu=on revealed that performance always degrades when IOMMU is effective regardless of bare-metal or virtual environment. This is illustrated in Figure 8.

5. Experience in Interop Tokyo 2018

We deployed Kamuee as a backup core router in the backbone network of Interop ShowNet 2018. Interop ShowNet has an experimental Internet backbone network deployed during the three-day Interop event. This is one of the largest experimental networks and every year, many product vendors bring in their new products to test their performance and interoperability. This year, over 2600 devices and services were connected to the network and over 450 engineers participated to build this network. As a backup core router of the network, Kamuee was responsible for forwarding all the traffic of this large scale network in case the primary core router fails.

The hardware specification used for Kamuee in Interop ShowNet 2018 is the same as the one in the previous benchmark, and shown in Table 1. Total of six routers were connected directly to Kamuee; four connected using 100GbE-SR4, one connected using 100GbE-LR4, and one connected using 10GbE-LR. The one using 10GbE-LR was the route reflector. Additionally, two network traffic generators were connected, one connected using 100GbE-LR4, and another connected using 10GbE-LR.

Kamuee was configured to provide the best performance for user traffic within the hardware constraints, such as limited number of CPU cores and NICs. We allocated eight cores per port to forwarder threads for 100GbE ports connected to the primary core router and the backup aggregation router. As we had limited number of CPU cores, we chose to only allocate six cores per port to forwarders for 100GbE ports connected to the AS border routers and the network traffic generators



(a) PPS throughput

(b) PPS achievement rate

Figure 5: Throughput in PPS

Table 3: Microflow Latency

				Avg. over 5 samples	mm:ss		Latency (us)		
NUMA	Cable	fsize	Tx load	Rx throughput	Time	#rx-frames	Min.	Avg.	Max.
Same	DAC	64	100Gbps/148.8Mpps	3.53Gbps/5.26Mpps	5:18	1,587,401,784	16.66	69.4	692.25
Same	DAC	128	100Gbps/148.8Mpps	5.63Gbps/4.76Mpps	6:20	1,823,246,252	10.43	71.95	120.82
Same	DAC	256	100Gbps/148.8Mpps	8.71Gbps/3.94Mpps	6:31	1,542,903,228	12.49	253.73	884.24
Same	DAC	512	100Gbps/148.8Mpps	21.49Gbps/5.05Mpps	5:34	1,670,079,851	4.86	22.01	112.36
Same	DAC	1024	100Gbps/148.8Mpps	38.02Gbps/4.55Mpps	8:21	2,378,373,317	5.86	22.18	892.35
Same	DAC	1518	100Gbps/148.8Mpps	61.19Gbps/4.97Mpps	5:54	1,696,514,167	10.87	34.23	184.03
Cross	DAC	64	100Gbps/148.8Mpps	3.32Gbps/4.94Mpps	7:16	2,220,846,968	16.35	69.46	859.5
Cross	DAC	128	100Gbps/148.8Mpps	5.88Gbps/4.97Mpps	7:01	2,089,926,340	12.18	75.63	2,183.4
Cross	DAC	256	100Gbps/148.8Mpps	6.40Gbps/2.90Mpps	5:12	903,051,580	12.02	333.07	369.8
Cross	DAC	512	100Gbps/148.8Mpps	22.07Gbps/5.19Mpps	11:55	3,661,038,680	4.85	22.06	64.8
Cross	DAC	1024	100Gbps/148.8Mpps	39.93Gbps/4.78Mpps	8:05	2,309,901,569	6.57	22.63	950.93
Cross	DAC	1518	100Gbps/148.8Mpps	51.56Gbps/4.19Mpps	12:15	3,194,555,373	9.79	25.37	812.38

as these ports only forwarded limited number of traffic. As 10GbE ports do not require many cores to provide performance, we only allocated two cores per port to forwarders responsible for the 10GbE ports. In total, fourty-six cores were used as forwarders to forward the traffic.

As a core AS router, the routing table of Kamuee consisted of the Internet full routing table and AS internal routes. Over 700K routes were registered on the IPv4 routing table while IPv6 routing table and had only about 59K routes.

From our experience of operating Kamuee in Interop ShowNet 2018, we discovered the following key problems that need to be addressed when running software routers as core routers using COTS devices:

- Nonoptimal cooling inside the chassis
- Differences in NIC implementation per vendor

First problem we encountered was excessive heating of the NICs. Unlike specialized networking chassis which have optimized cooling for NICs and network processors, the COTS server chassis used for Kamuee was not equipped to do so. The temperature sensor readings showed temperature of up to 67 degrees centigrade while the maximum operating temperature for100GbE-LR4 QSFP28 module is 70 degrees centigrade [26]. This calls for some additional cooling design while using COTS hardware as high speed network device to avoid errors due to overheating.

Second problem we faced is the difference in implementations of NIC functionalities and their corresponding DPDK drivers per vendor. Kamuee's initial design used KNI as the packet interface between the dataplane and kernel. Though it worked fine for NICs from a single vendor, it malfunctioned when we started to use NICs from multiple vendors simultaneously. We needed to switch back to the slower Tun/Tap kernel interface to address the malfunctioning problem.

Table 4: 10Gbps traffic packet loss

			Avg. over 5 samples		mm:ss				
NUMA	Cable	fsize	Tx load	Rx throughput	Time	#tx-frames	#rx-frames	#loss*	loss-rate+
Same	DAC	64	10Gbps/14.88Mpps	10.00Gbps/14.88Mpps	5:22	4,793,907,953	4,793,899,029	8,912	1.86e-06
Cross	DAC	64	10Gbps/14.88Mpps	10.00Gbps/14.88Mpps	5:13	4,666,135,849	4,666,135,251	592	1.27e-07

* The tester's tx, rx, and loss counts didn't seem to be consistent for unknown reason.
+ Loss-rate is calculated by dividing #loss by #tx-frames.

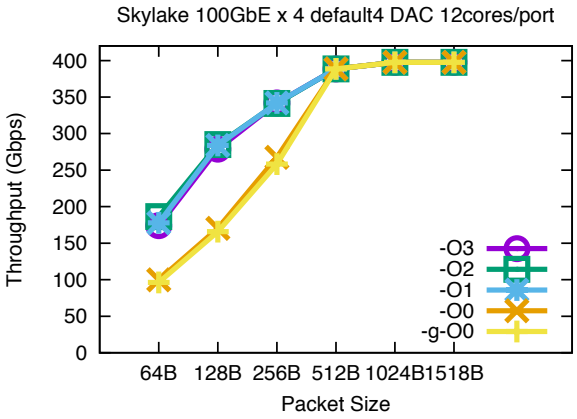


Figure 6: Compiler Effect

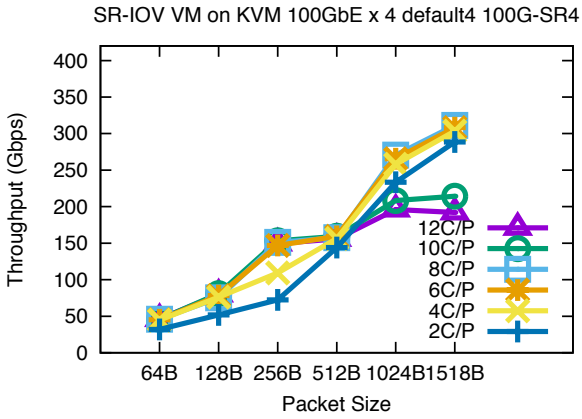


Figure 7: KVM cores per port effect

Kamuee is the first ever software router to be used in Interop Shownet backbone as one of the core routers, and throughout the event Kamuee ran properly without any glitches or errors. Kamuee lived up to the challenge of a fully functional software router interoperable with most of the world's leading router vendors, namely, Cisco, Juniper, and Huawei, and this is a promising stepping stone for its future commercial success as a mainstream software router.

6. Conclusion and future work

We revealed the tricks for high-performance software router design: we think the run-to-completion model in DPDK, the Poptrie algorithm, and the keep it as simple as possible principle, are the main grounds for our superior performance.

We show a good performance benchmark value of Kamuee: for 512B-sized frames or longer, Kamuee can forward around 349.60 Gbps random destination traffic with BGP full routes (at FullRoute-LoopbackBlackhole in Figure 4). Further intralaboratory evaluation of 12C/P at 64B frame in LoopbackTap (Figure 3a), the Kamuee showed the potential of forwarding 292.17 Mpps.

We have seen many unstable characteristics of Kamuee, that might be common to the general software router. The throughput performance fluctuated over time, in a significantly large range of a few tens of Gbps. Further, we sometimes saw performance degradation that we cannot explain (yet). Since the performance of the software router is very high and promising, the need to address the aforementioned drawbacks becomes even more imperative.

Even with some drawbacks, our Interop experience proved that Kamuee satisfies the necessary functions and quality needed to sustain the large scale IP infrastructure. Overall, Kamuee demonstrated a promising performance for the use of future virtual network functions in the NFV environment.

As an open problem for the future network, the algorithms for the Access Control List (ACL) and firewall applications that maintain high performance even with some hundreds of thousands to some millions of ACL entries are the next challenge in this field.

References

[1] G. P. Katsikas, T. Barbette, D. Kostić, R. Steinert, G. Q. M. Jr., Metron: NFV service chains at the true speed of the underlying hardware, in: 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18), USENIX

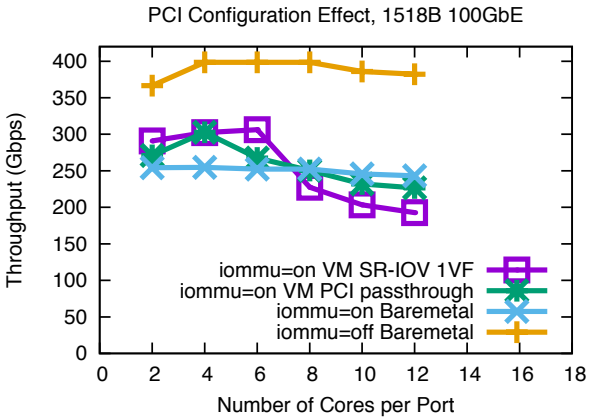


Figure 8: PCI configuration effect

Association, Renton, WA, 2018, pp. 171–186.
URL https://www.usenix.org/conference/nsdi18/presentation/katsikas

[2] M. Dalton, D. Schultz, J. Adriaens, A. Arefin, A. Gupta, B. Fahs, D. Rubinstein, E. C. Zermeno, E. Rubow, J. A. Docauer, J. Alpert, J. Ai, J. Olson, K. DeCabooter, M. de Kruijf, N. Hua, N. Lewis, N. Kasinadhuni, R. Crepaldi, S. Krishnan, S. Venkata, Y. Richter, U. Naik, A. Vahdat, Andromeda: Performance, isolation, and velocity at scale in cloud network virtualization, in: 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18), USENIX Association, Renton, WA, 2018, pp. 373–387.
URL https://www.usenix.org/conference/nsdi18/presentation/dalton

[3] Intel, DPDK – Data Plane Development Kit, http://dpdk.org/.

[4] H. Asai, Y. Ohara, Poptrie: A compressed trie with population count for fast and scalable software ip routing table lookup, in: Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, SIGCOMM '15, 2015.

[5] P. E. McKenney, J. D. Slingwine, Read-copy update: Using execution history to solve concurrency problems, in: Parallel and Distributed Computing and Systems, 1998, pp. 509–518.

[6] Y. Ohara, Y. Yamagishi, Kamuee zero: the design and implementation of route table for high-performance software router, in: Proceedings of Internet Conference 2016, IC 2016, 2016.

[7] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, V. Sekar, Making middleboxes someone else's problem: Network processing as a cloud service, in: Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM '12, ACM, New York, NY, USA, 2012, pp. 13–24. doi:10.1145/2342356.2342359.
URL http://doi.acm.org/10.1145/2342356.2342359

[8] N. W. Paper, Network functions virtualisation: An introduction, benefits, enablers, challenges & call for action. issue 1 (Oct 2012).

[9] B. Li, K. Tan, L. L. Luo, Y. Peng, R. Luo, N. Xu, Y. Xiong, P. Cheng, E. Chen, Clicknp: Highly flexible and high performance network processing with reconfigurable hardware, in: Proceedings of the 2016 ACM SIGCOMM Conference, SIGCOMM '16, ACM, New York, NY, USA, 2016, pp. 1–14. doi:10.1145/2934872.2934897.
URL http://doi.acm.org/10.1145/2934872.2934897

[10] A. Panda, S. Han, K. Jang, M. Walls, S. Ratnasamy, S. Shenker, Netbricks: Taking the v out of NFV, in: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), USENIX Association, Savannah, GA, 2016, pp. 203–216.
URL https://www.usenix.org/conference/osdi16/technical-sessions/presentation/panda

[11] J. Martins, M. Ahmed, C. Raiciu, V. Olteanu, M. Honda, R. Bifulco, F. Huici, Clickos and the art of network function virtualization, in: 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14), 2014.

[12] J. Hwang, K. K. Ramakrishnan, T. Wood, Netvm: High performance and flexible networking using virtualization on commodity platforms, IEEE Transactions on Network and Service Management 12 (1) (2015) 34–47.

[13] R. Poddar, C. Lan, R. A. Popa, S. Ratnasamy, Safebricks: Shielding network functions in the cloud, in: 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18), USENIX Association, Renton, WA, 2018, pp. 201–216.
URL https://www.usenix.org/conference/nsdi18/presentation/poddar

[14] M. Honda, G. Lettieri, L. Eggert, D. Santry, PASTE: A network programming interface for non-volatile main memory, in: 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18), USENIX Association, Renton, WA, 2018, pp. 17–33.
URL https://www.usenix.org/conference/nsdi18/presentation/honda

[15] A. Tootoonchian, A. Panda, C. Lan, M. Walls, K. Argyraki, S. Ratnasamy, S. Shenker, Resq: Enabling slos in network function virtualization, in: 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18), USENIX Association, Renton, WA, 2018, pp. 283–297.
URL https://www.usenix.org/conference/nsdi18/presentation/tootoonchian

[16] S. Choi, B. Burkov, A. Eckert, T. Fang, S. Kazemkhani, R. Sherwood, Y. Zhang, H. Zeng, Fboss: Building switch software at scale, in: Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, SIGCOMM '18, ACM, New York, NY, USA, 2018, pp. 342–356. doi:10.1145/3230543.3230546.
URL http://doi.acm.org/10.1145/3230543.3230546

[17] 8. Poll Mode Driver – Data Plane Development Kit 18.08.0 documentation, http://doc.dpdk.org/guides/prog_guide/poll_mode_drv.html.

[18] M. Dobrescu, N. Egi, K. Argyraki, B.-G. Chun, K. Fall, G. Iannaccone, A. Knies, M. Manesh, S. Ratnasamy, Routebricks: Exploiting parallelism to scale software routers, in: Proceedings of the ACM SIGOPS 22Nd Symposium on Operating Systems Principles, SOSP '09, 2009.

[19] S. Hemminger, Making a virtual router a reality with dpdk, rcu and 0mq, https://events.static.linuxfound.org/sites/events/files/slides/DPDK_RCU_0MQ.pdf.

[20] Universal TUN/TAP device driver., https://www.kernel.org/doc/Documentation/networking/tuntap.txt.

[21] 33. Tun—Tap Poll Mode Driver – Data Plane Development Kit 18.08.0 documentation, https://doc.dpdk.org/guides/nics/tap.html.

[22] Ferruh Yigit, Interworking with the Linux Kernel, https://dpdksummit.com/Archive/pdf/2016Userspace/Day02-Session06-FerruhYigit-Userspace2016.pdf.

[23] VPP Sandbox/router - fd.io, https://wiki.fd.io/view/VPP_Sandbox/router.

[24] Hasan Alayli, lthread, a multicore enabled coroutine library written in C, https://github.com/halayli/lthread.

[25] Y. Ohara, Y. Yamagishi, S. Sakai, A. D. Banik, S. Miyakawa, Revealing the necessary conditions to achieve 80gbps high-speed pc router, in: Proceedings of the Asian Internet Engineering Conference, AINTEC '15, 2015.

[26] Mellanox, Mellanox 100GbE QSFP28 LR4 Optical Transceiver, https://www.mellanox.com/related-docs/prod_cables/PB_MMA1L10-CR_100GbE_QSFP28_LR4_Transceiver.pdf.

Poster

Lightweight Packet Loss Detection and Multicast Delivery Tree Recovery in SDN

Siva Sairam Prasad (cs14resch01003@iith.ac.in), Kotaro Kataoka (kotaro@iith.ac.in)

Indian Institute of Technology Hyderabad, India

Outline

Monitoring quality of multicast streaming is difficult because 1) generally intermediate multicast routers and switches do not maintain the fine-grained state about multicast flow and 2) sampling QoS statistics from destination clients introduces significant operational overhead. Therefore, its difficult to properly locate where packet losses happen and how severe they are. This paper proposes a mechanism to detect and locate packet losses using a packet tagging technique in Software Defined Network (SDN). Our system also dynamically recalculates the multicast delivery tree to bypass a lossy link and mitigate packet loss for the tree.

Problem Description

- In traditional network it is difficult to detect and impossible to locate packet loss for multicast in real-time.
- With the evolution of SDN, calculating packet loss at link level is not feasible in real-time due to variable bit rate and variable duration of the flows.
- By using short lived test flows can get the packet loss at link level, but they lead to overhead in the network and also more TCAM consumption.
- Need to utilize the unused links that are non-lossy for multicast delivery tree calculation.
- Recalculation of entire multicast delivery tree whenever packet loss happens leads to frequent interruptions to the users, also for those are not affected by packet loss.

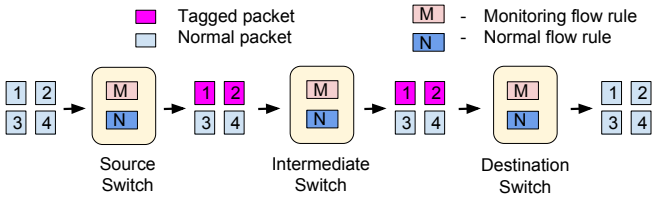


Figure 1: Packet Tagging Mechanism

Proposed solution

Packet Loss Detection

- Install short lived monitoring flow rules on existing traffic.
- Tag packets based on fixed time intervals as shown in Figure.1 for monitoring at source switch.
- Monitor the tagged packets at intermediate switches and remove tag before reaching the destination.
- Get packet loss statistics from monitoring flow rules in periodic intervals.
- Calculate packet loss from statistics and also deduce the lossy link location.

Multicast Delivery Tree Recovery

- Bypass the lossy link in the tree.
- If bypass is not feasible then recalculate the partial tree that is affected by packet loss.

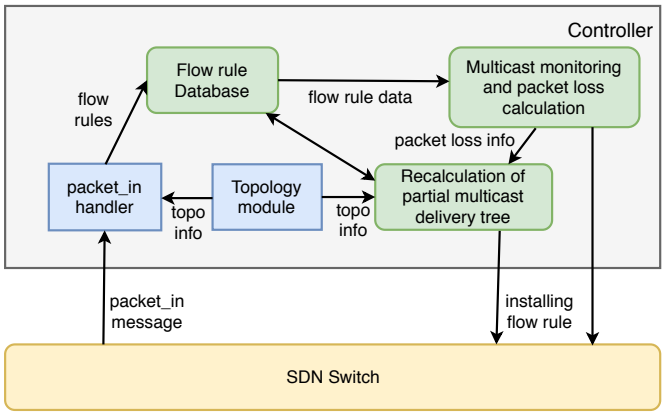


Figure 2: System Diagram

Experimental Setup & Preliminary Results

Network Configuration using Mininet

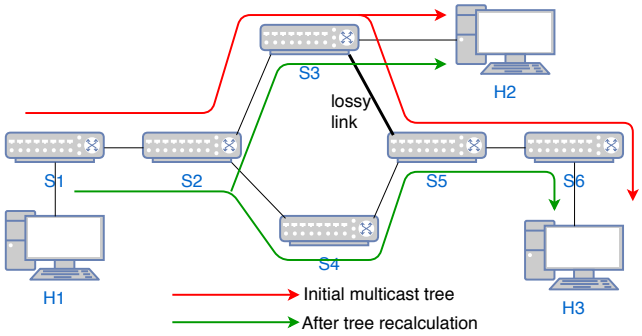


Figure 3: Network Diagram

- In the above network diagram S1-S6 are SDN Switches, H1 is multicast source and H2,H3 are multicast destinations.
- Packet loss rate for (S3,S5) is 10%, for all other edges there is no packet loss.
- Initial and recalculated multicast delivery tree are shown in the above figure.

Preliminary Results

- As shown in Figure 3, the recalculation is done via partial tree recalculation because bypass is not possible.
- Packet loss calculation for every 10 s, time taken to recalculate for one link on above network is 1.8 ms.

Discussion Points

- Minimize the control overhead for monitoring, scalable to large networks.
- Time complexity for each lossy link to bypass is O(E logV), for partial recalculation is O(DE logV) here D is the affected destinations.

Trust Management in Multi-Domain SDN Networks Using Blockchain

Prashanth Podili (cs15resch01003@iith.ac.in), Kotaro Kataoka (kotaro@iith.ac.in)

Indian Institute of Technology Hyderabad

Outline

In Multi-Domain Software-Defined Networks (MD-SDN) such as Internet (Fig.1), controllers communicate to provision end-to-end services across multiple domains besides enabling inter-domain routing[1]. Trust management across the domains provides operational feasibility, transparency, enhanced security and prevention of data abuse. However, quantifying and evaluating trust across domains with heterogeneous SDN implementations is challenging. This work aims to evaluate trust by enabling auditability across domains and builds a distributed trust management system for MD-SDN networks using blockchain. Blockchain allows a) to verify integrity of audit records b) tamper proof trust data storage and dissemination.

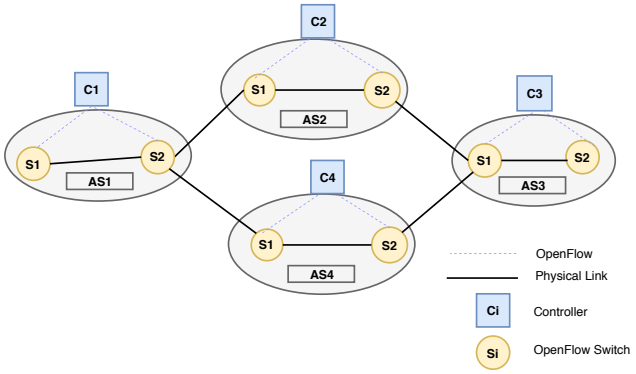


Fig. 1 MD-SDN Network

Background and Motivation

- Programmability of SDN offers flexibility for E2E service provisioning in MD-SDN networks[2].
- Information exchange between domain controllers are not verified (similar to BGP) to ensure integrity of exchanged messages.
- Finding trusted domains among multiple domains is challenging.
- Services with varying QoS values require different trust levels for SDN domains to collaborate.

Trust Management-Challenges

- Heterogeneity of SDN implementation across domains.
- Difficulty to quantify and evaluate trust between domains.
- Mechanism to build truly distributed trust management system.
- Handling collusions/fake trust values of domains.
- Tamper proof trust data storage and dissemination among SDN domains

Proposed Approach

- Trust Definition: Domain A trusts domain B, when domain B operates/provides the services in a manner that is promised to A.
- Evaluating trust by verifying audit logs from service provider domain controllers
- Building Trusted Controller Information Base(TCIB)
  - Exchange Domain Information Message(DIM) (xml/json message) during domain discovery process. DIM includes SDN deployment parameters and services offered with in a domain.
  - Hash of DIMs are verified through blockchain.

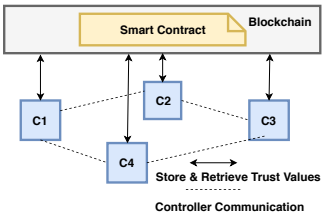


Fig. 2 Blockchain Controller Communication

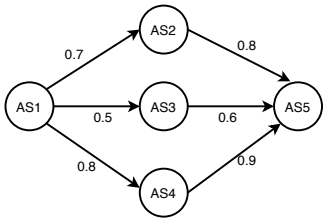


Fig. 3 Sample Trust Graph

- Service Profile Trust
  - Service Profiles classify distributed services with varying QoS. E.g., service profiles for video traffic delivery, virtual networks, service chains etc.
  - Identify (openflow) parameters for every service profile.
  - Negotiate parameters and obtain audit logs from collaborating domains for trust evaluation.
- Quantifying trust between a pair of domains for a service profile
  - Trust value is 1 if parameters conforms to expected (QoS) values, else 0.
  - Aggregated trust value calculated over a period of time, $(T_{agg}) = (m/n)$, m = successful interactions, n= total interactions.
  - Every domain controller puts T_{agg} of domains it interacted to blockchain.
- Trust management using blockchain
  - Smart contract is created to store and distribute trust values.
  - Trust graph (Fig.3) is generated for each service profile across all the domains and global trust is evaluated based on trust transitivity property.

System Design and Implementation

- To evaluate trust in a given domain *D* (Fig.4), the domain controller checks the TCIB for SDN deployment parameters of *D* and retrieves global trust value from blockchain.
- If *D* is trustworthy, the controller establishes a session and negotiates service parameters for auditing.
- Audit log retrieved from *D* is verified for evaluating trust.
- Our Implementation uses Ethereum, Solidity for creating smart contract, Mininet with Open vSwitches and RYU SDN controllers.

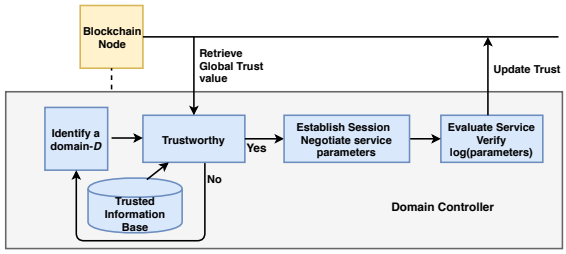


Fig. 4 Trust evaluation of a peer domain in controller

Current Progress and Discussions

- Developed testbed integrating RYU controller and blockchain(Fig. 2) and currently working on design of trust evaluation algorithms.
- Discussion points:a)Frequency of trust value updates to blockchain b)Missing trust values for a pair of domains in trust graph.

References

1. Zhou, Haifeng, et al. "SDN-LIRU: A lossless and seamless method for SDN inter-domain route updates." IEEE/ACM Transactions on Networking 25.4 (2017): 2473-2483.

2. Kotronis, Vasileios, et al. "Stitching inter-domain paths over IXPs."Proceedings of Symposium on SDN Research. ACM, 2016.

On Accurate Packet Loss Estimation for Networks without Traffic Models

Masahiro Terauchi Kohei Watabe Kenji Nakagawa

Graduate School of Engineering, Nagaoka University of Technology, Nagaoka, Niigata, Japan.

Introduction

- It is important to accurately model network traffic when we evaluate Quality of Service (QoS) of networks through simulations.
  - It is difficult to select an appropriate traffic model and tune its parameters.
  - Even if the accurate traffic modeling is achieved, it is also difficult to accurately estimate QoS regarding rare events.

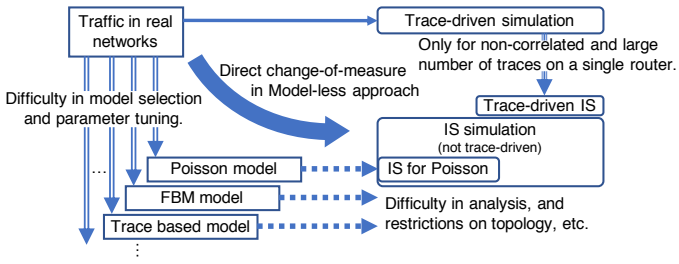


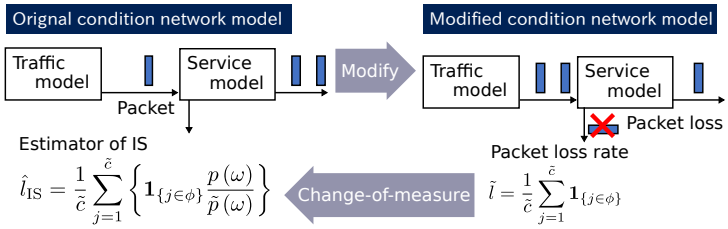
Figure 1: The model-less approach and the conventional simulations.

- Importance Sampling (IS) for accurate estimations of rare events [2]
  - The events occur more frequently in IS simulation.
  - The estimator is obtained by the change-of-measure.
  - The applicable traffic models, topology etc. are extremely limited.
- Trace-driven IS without traffic models [3]
  - It cannot be applied for single flow traffic.
  - It is not applicable for traffic with correlated flows.

Goal of our study

- We propose a model-less approach to accurately estimate a packet loss rate through a simulation without directly modeling traffic, including real network traffic.

Model Based IS



$\tilde{c}$: The number of packets, ϕ : The set of lost packets, ω : The queue length process,

$p(\omega)$: Probability density of an event ω in the original condition,

$\tilde{p}(\omega)$: Probability density of an event ω in the modified condition.

Figure 2: Outline of model base IS

- When IS estimates a loss rate on a single router into which a single flow streams, the change-of-measure is performed based on probability density of a path ω of the queue length process [2].
- The change-of-measure $p(\omega)/\tilde{p}(\omega)$ is analytically derived from a traffic model in model-based IS.

Model-less Approach

- Our goal is to accurately estimate a packet loss rate through a simulation in a real network without assuming any traffic model.

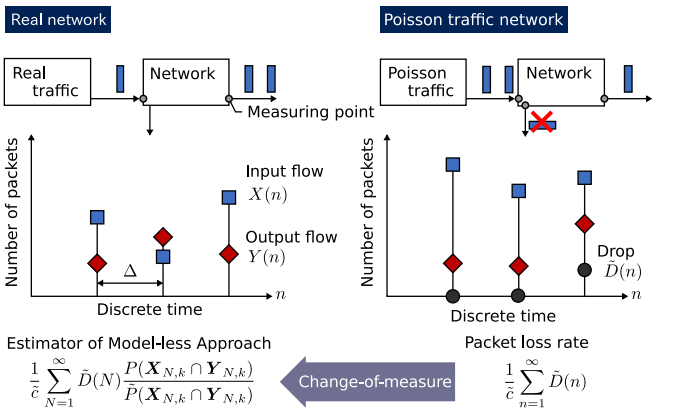


Figure 3: Outline of model-less approach

- The model-less approach follows the procedure below.
  - A simulation with Poisson traffic model is performed.
  - Input traffic, output traffic and loss processes are discretized with Δ.
  - Change-of-measure is based on frequency of discretized traffic pattern.
- Our estimator is

$$\hat{l} = \frac{1}{\tilde{c}} \sum_{N=1}^{\infty} \tilde{D}(N) \frac{P(\boldsymbol{X}_{N,k} \cap \boldsymbol{Y}_{N,k})}{\tilde{P}(\boldsymbol{X}_{N,k} \cap \boldsymbol{Y}_{N,k})}, \tag{1}$$

where

$\boldsymbol{X}_{N,k} = \{X(n)\}_{N-k<n\leqq N}$: Discretized input flow traffic in past k periods.

$\boldsymbol{Y}_{N,k} = \{Y(n)\}_{N-k<n\leqq N}$: Discretized output flow traffic in past k periods.

$\tilde{D}(n)$: Discretized packet loss process.

- When we assume a single router and a single flow, in the limit as $\Delta \to 0$ and $k \to N$, our estimator converges to that of model based IS.
- By expressing the estimator by input and output traffic instead of a queue length process, (1) is applicable for multiple flows on a network with complicated topology.

Experiments

- As a first step in the development, we investigate the case when the packet loss rate of an MMPP/M/1/K system is estimated from an M/M/1/K simulation.
- The simulation time is 2000 [s], simulation sets is 30, $\Delta = 0.025$ [s], and $k = 2$.
- In these systems, since the packet arrivals and a service time are independent.
- Therefore, the change-of-measure can be expressed as $P(X_{1,N,k} \cap Y_{1,N,k})/\tilde{P}(X_{1,N,k} \cap Y_{1,N,k}) = P(X_{1,N,k})/\tilde{P}(X_{1,N,k})$.

Table 1: Target network parameters

Arrive Rate at State 1 [packet/s]	100
Arrive Rate at State 2 [packet/s]	339
Transition Rate of Each State [times/s]	1.00
Mean Service Time [s/packet]	0.001
Queue Size K [packet]	10
Packet Loss Rate [-]	10^{-5}

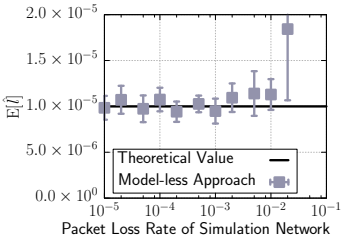


Figure 4: Result of mean

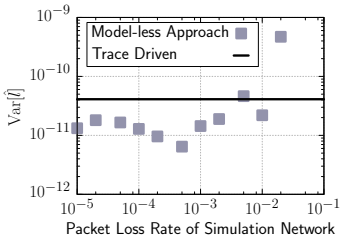


Figure 5: Result of variance

- According to the figures, we can find the region in which the model-less approach can estimate the packet loss rate of the original system.
- Additionally, we can confirm that the variances of the estimators are about 1/3 in the region, compared with the estimator by the trace-driven simulation.

Conclusions and Future Directions

- We proposed the model-less approach to accurately estimate a packet loss rate through simulation with traffic trace without traffic models.
- We will verify the applicability of our approach to the various trace on various networks in our future works.

References

[1] J. Zhang *et al.*, "A Survey of Network Traffic Generation," in *Proc. of CCT 2015.*

[2] N. Kobayasahi *et al.*, "On-line Estimation by Importance Sampling for the Tail Probability of FIFO Queue Length, " in *Proc. of RESIM 2014.*

[3] I. C. Paschalidis *et al.*, "Importance Sampling for the Estimation of Buffer Overflow Probabilities via Trace-driven Simulations," *IEEE/ACM Transactions on Networking*, vol. 12, no. 5, 2004.

This work was partly supported by JSPS KAKENHI Grant Number JP17K00008 and JP18K18035.

Internet Conference 2018
IC2018

An Ocean Target Detection Mechanism in IoT Environment

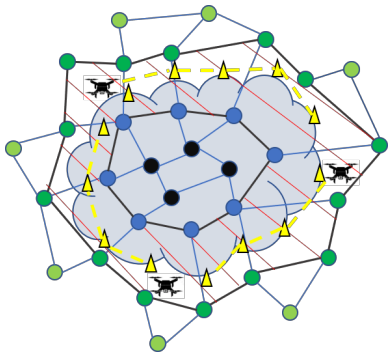
Yaqiang Zhang[1,2], Xiangbo Kong[1], Lin Meng[1], Zhangbing Zhou[2] and Hiroyuki Tomiyama[1],
[1]Graduate School of Science and Engineering, Ritsumeikan University
[2]School of Information Engineering, China University of Geosciences

Background and System

- Toxic target like crude oil leak will cause great damage to human and marine environment.
- With the development of IoT system, devices like UAV can be deployed to detect and track continuous target.

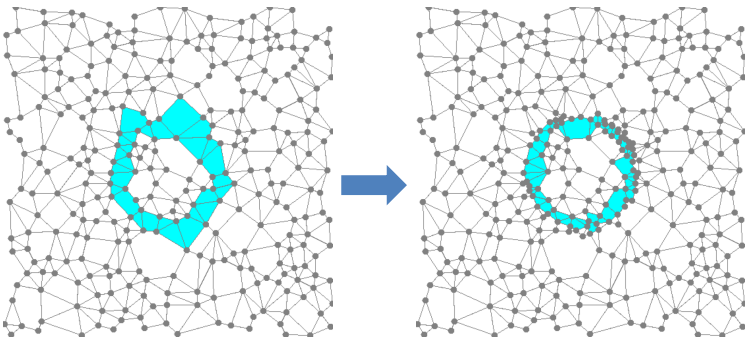
✓ A heuristic algorithm based mechanism is proposed for routing mobile IoT devices like UAV in order to track the boundary region of marine pollutant.

- Static IoT sensors are firstly deployed in the interested ocean network region and separated into different groups according to their reading.
- The gray region is the scope of the marine pollutant.
- Black nodes: a sensor whose reading is exceed the threshold of toxic target and all of its neighbor nodes are event nodes.
- Blue nodes: a sensor whose reading is exceed the threshold of toxic target and some of its neighbor nodes are event nodes while others are not.
- Green nodes: a sensor whose reading is under the threshold of toxic target and some of its neighbor nodes are event nodes while others are not.
- Light green nodes: a sensor whose reading is under the threshold of toxic target and all of its neighbor nodes are normal nodes.



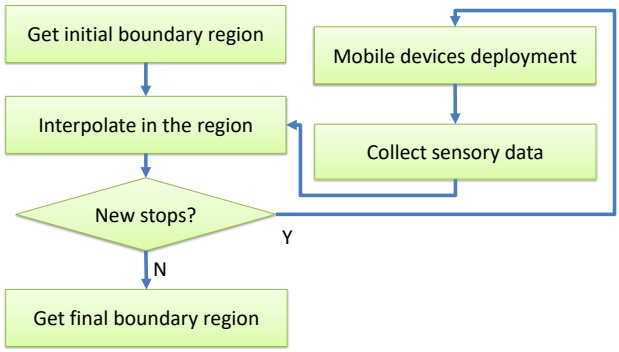
- The yellow dotted line is the predicted boundary line on which all positions are equal to threshold according to interpolated results.
- Yellow triangular marks are stops selected on the predicted boundary line and there is a proper distance between each stop.
- A suitable number of UAV are deployed to traverse stops and genetic algorithm (GA) is applied to route for Multi-UAV in order to optimize the time and energy consumption of UAV.

Target boundary region detection



Initial boundary region　　Final boundary region

Procedure



- Interested network region is divided by static IoT sensor nodes and an initial boundary region is generated.
- According to the sensory data gathered by static IoT sensor nodes, sensing holes that don't have sensory data are interpolated and find a predicted boundary line.
- Some stops are selected on predicted boundary line if there are.
- Multi-UAV are applied to traverse selected stops and get sensory data on stops.
- A new boundary region is generated according to new data and repeat above steps until there are no new stops.

Experimental Results

Changes in boundary region with different number of static sensors.

- Through continuous iteration, the scope of the target boundary region is gradually narrowed and kept stable.
- In different scale of static sensors deployment, the size of the target boundary region varies greatly. When static IoT node deployment is more densely, the result is more accurate.

Conclusions

- The proposed mechanism can effectively detect and track the ocean target boundary region.
- The experimental data shows that the scope of target boundary region is shrunken and it reflects the actual situation of toxic target.
- The GA based mechanism for routing UAV can effectively balance the energy and time consumption.

Reference

[1] L. Shu, M. Mukherjee, and X. Wu, "Toxic gas boundary area detection in large-scale petrochemical plants with industrial wireless sensor networks," IEEE Communications Magazine, vol.54, no. 10, pp. 22–28, 2016.
[2] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges," Future Generation Computer Systems, 2016.
[3] J.-H. Kim, K.-B. Kim, S. H. Chauhdary, W. Yang, and M.-S. Park, "DEMOCO: Energy-efficient detection and monitoring for continuous objects in wireless sensor networks," IEICE Transactions on Communications, vol. E91-B, no. 11, pp. 3648–3656, 2008.

e-Learning System for Cryptography on Moodle

†Osaka Electro-Communication University
‡Shinshu University
§Tokyo University of Technology

Tatsuki Miyamoto†, Shogo Shimura‡, Tatsuki Watanabe†, Hiroyuki Okazaki‡, Yuichi Futa§, Yasuyuki Murakami†

Summary

In Japan, due to the lack of ICT engineers, fostering human resources for ICT is an urgent task. Especially, cryptographic technology is a fundamental element for realizing information security. Learning knowledge and training techniques on cryptography are indispensable not only to researchers and engineers specialized in cryptography but also to ICT engineers such as network engineers and operators. However, the educational environment and teaching materials of cryptographic technology for ICT engineers are not sufficient. There is also a lack of teachers who can educate the theory and the technology of the cryptography.

Recently, e-Learning using the Internet has been widely spread especially at the educational places such as universities. Moodle is popular and widely used as an e-Learning system. In Moodle, students answer questions on quizzes on the browser via Internet and whether the answer is correct or not can be judged depending on whether or not their answer matches the model answer set by the teacher. Moodle is excellent e-Learning system which also has an automatic scoring function. However, learning programming with Moodle is not always easy because the program is not necessarily one correct answer.

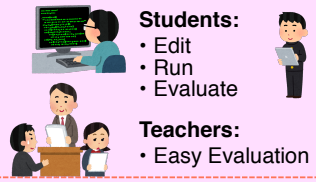
VPL, Virtual Programming Lab for Moodle, is a free system formed by two components: the Moodle plugin and the execution system(Jail-System). The Moodle plugin can be installed as a regular Moodle plugin. The execution system needs to be compiled for installation. The installation is not always easy for teachers.

Formal verification of cryptographic protocols has been studied extensively in recent years. ProVerif is one of the most successful automatic cryptographic protocol verifiers. In the previous work, we developed an e-Learning system for learning C programming based on Moodle with VPL by using VM, Virtual Machine, for easy installation[1]. In this research, we newly support ProVerif to the previously developed system and create the contents for learning ProVerif.

What is Moodle / VPL

Moodle is a free, online Learning Management system. Moodle is widely used as an e-Learning system in many educational institutions. It is easy to install and can be expanded with plugin.

VPL is a free system formed by two components: the Moodle plugin and the execution system. The Moodle plugin can be installed as a regular Moodle plugin. The execution system needs to be compiled for installation.



Students:
- Edit
- Run
- Evaluate

Teachers:
- Easy Evaluation

Developed e-Learning System

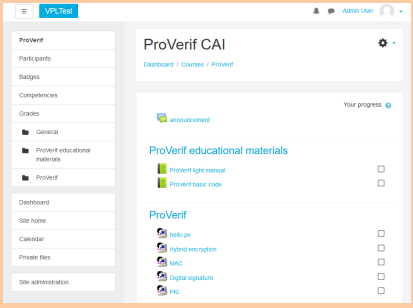


Fig.1 Moodle Screen on Browser

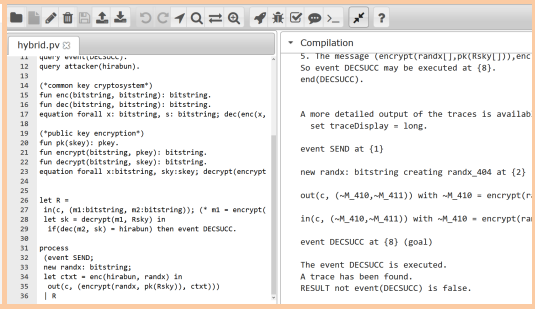


Fig.2 Learning ProVerif on Moodle+VPL

Benefits for Students:
- No compiler required, Browser only required.
  →Available on Tablet or Smartphone.
- No source code submission required.
- Discover mistakes themselves instantly.

Benefits for System Managers:
- Using VM, Any host OS available.
- Easy to Install, manage and update.
- No need to introduce compiler for each client.

Benefits for Teachers:
- Automatic evaluation.
  →No program download, nor compile, nor execution required.
- By the test pattern, it is easy to find mistakes.
- The correct answer is judged as correct even if it is not the model answer.

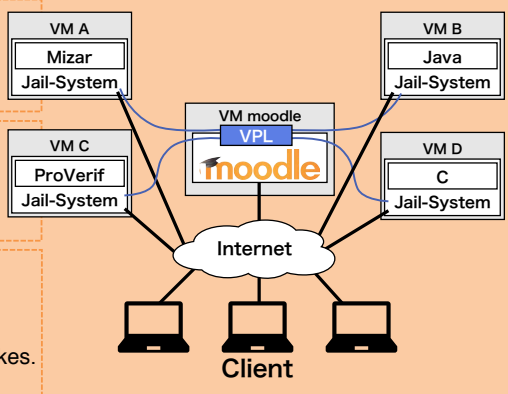


Fig.3 System Configuration

ProVerif Sample

```
free c: channel.
free hirabun: bitstring [private].

type pkey.
type skey.
free Rsky: skey [private].

event DECSUCC.
event SEND.

query event(DECSUCC).
query attacker(hirabun).

(*common key cryptosystem*)
fun enc(bitstring, bitstring): bitstring.
fun dec(bitstring, bitstring): bitstring.
equation forall x: bitstring, s: bitstring; dec(enc(x,s),s) = x.

(*public key encryption*)
fun pk(skey): pkey.
fun encrypt(bitstring, pkey): bitstring.
fun decrypt(bitstring, skey): bitstring.
equation forall x:bitstring, sky:skey; decrypt(encrypt(x, pk(sky)), sky) = x.


let R =
 in(c, (m1:bitstring, m2:bitstring)); (* m1 = encrypt() m2 = ctxt *)
 let sk = decrypt(m1, Rsky) in
  if(dec(m2, sk) = hirabun) then event DECSUCC.

process
 (event SEND;
 new randx: bitstring;
 let ctxt = enc(hirabun, randx) in
  out(c, (encrypt(randx, pk(Rsky)), ctxt)))
 | R
```

Fig.4 ProVerif Sample Source Code

Fig.5 Verifying Process by ProVerif

Conclusion

In this research, we have constructed a programming education support system using Moodle and VPL using VM. We have newly supported ProVerif to previously developed e-Learning system using VM. In addition, we have newly created the contents for learning ProVerif and actually started the learning exercise of ProVerif by using the developed system. As a future plan, we would like to enrich the contents and evaluate the effect of learning the security programming by using the developed e-Learning system.

Acknowledgement: This study was supported in part by JSPS KAKENHI Grant Numbers JP18K02917 and JP17K00182.

[1] M. Nakamura, T, Watanabe, M. Kaneda, H. Okazaki and Y. Murakami, “Programing education support system using Moodle,” 40th Symposium on Information Theory and Its Application, Poster session, SITA2017, Nov. 2017 (in Japanese).

VISIBLE: Application for Vehicle Visibility and Incident Reporting in Real-Time

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Mehul Sharma, Suhel Magdum, Antony Franklin A, Bheemarjuna Reddy Tamma and Digvijay S. Pawar*

Department of CSE, IITH, *Department of Civil Engineering, IITH
Email: [cs17mtech11020, cs17mtech11012, antony.franklin, tbr, dspawar]@iith.ac.in

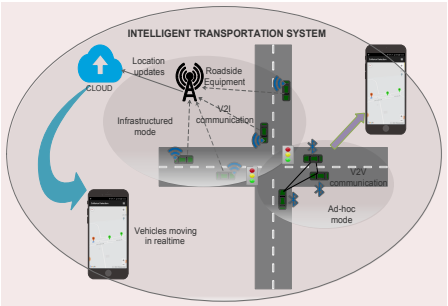
Introduction

- Safety issues in the transportation system are the major concerns.
- V2V/V2I are emerging as an efficient solution for achieving road safety.
- Blind Spots, Intersections, and Ghat Sections are the major accident-prone areas where there is no clear visibility of moving vehicles.
- Internet connectivity is the main concern in areas like ghats.

Objective

- To build a reliable platform that effectively utilizes mobile devices for grasping the traffic situation.
- To develop efficient V2V/V2I communication using cloud technology and P2P.

V2V/V2I Communication



Technologies Used

Conclusion

- We developed a smartphone-based application that can make use of P2P and cloud technology to detect vehicles in the collision domain.
- Future work comprises of audible beeps/alerts if any vehicle comes into danger zone.
- Developing efficient RF (eg. Bluetooth) scanning methods for estimating traffic congestion and speed.

Acknowledgement

This work was supported by M2Smart Project, JST/JICA SATREPS, Japan.

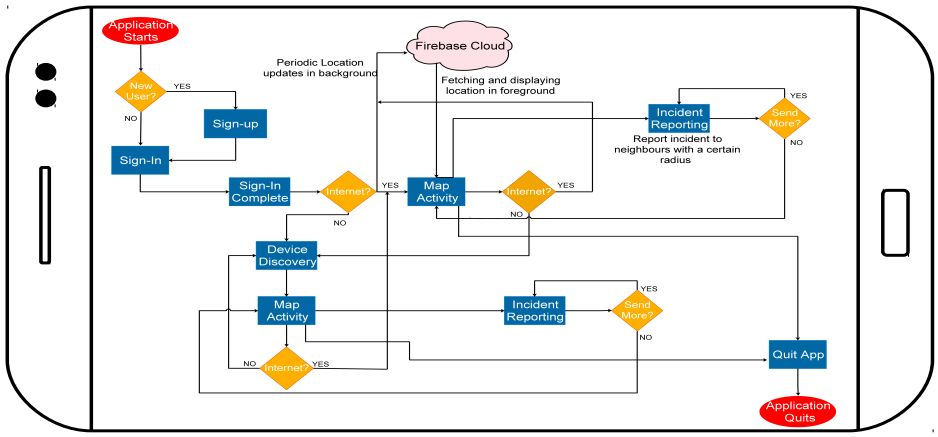
Application Features

1. Real-time visibility of neighbouring vehicles in the collision domain.
2. Incident Reporting within a certain radius.
3. App has two modes, Cloud and P2P.
4. Application automatically switches from cloud mode to P2P mode when there is no Internet.

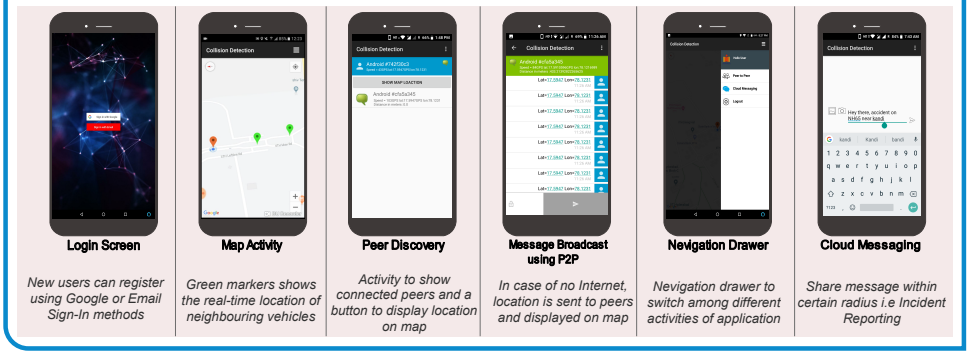
Cloud vs Adhoc Mode

P2P Mode	Cloud Mode
Range is <80m	No bound on range
Unreliable due to link breakage	More reliable
More time to discover presence of neighbour	Less discovery time
More discovery time but less data sharing time	More time to share data i.e. cloud latency
Multihoping	Direct Information sharing using centralized cloud

Application Flow Diagram



Application Screens



Results and Analysis

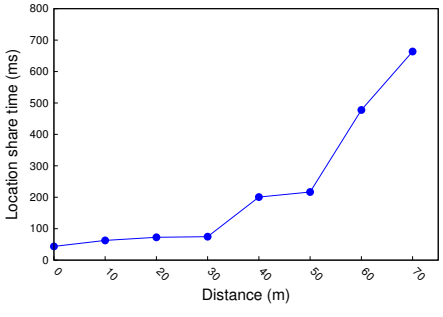


Figure 1: Location share time vs Distance

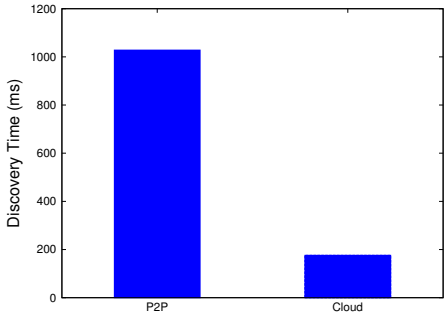


Figure 2: Device Discovery Time

- Fig. 1 shows that time taken to share location to peers in P2P mode increases with increase in inter-node distance.
- Clock of both sender and receiver in P2P is synced using Network Time Protocol (NTP).
- Discovery time is the time taken to discover the presence of neighbour and display its location on the map.

九州大学
KYUSHU UNIVERSITY

An evaluation of method for zero-day malicious email detection using email header information analysis (EHIA) and deep-learning approach

Sanouphab Phomkeona and Koji Okamura
Graduate School of Information Science and Electrical Engineering, Kyushu University, Japan

Introduction

CIS Center for Internet Security®

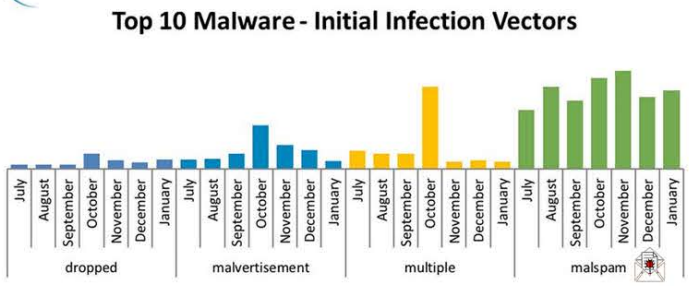


Fig. 1 CIS Cybersecurity report on malware infection vector

1. Email is the most common entry point of targeted attacks
2. About half of all email traffic is malspam, it means about 14.5 billion malspam are sent every single day in Q1 2018
3. Currently, the majority of security systems are unable to detect and stop today's advanced email threats that are specifically designed to fool the security systems

Related Works

Table 1.
Email header features considered by different machine learning malspam filtering techniques

Sheu, 2009 [11]	Ye et al., 2008 [12]	Wu, 2009 [13]	Hu et al., 2010 [14]	Wang & Chen, 2007 [15]	Al-Jarrah et al., 2012 [10]	Our approach
Length of sender field , Sender field, Title more than one category) , Time, Size of email	Received field (domain add., IP add., relay servers, date, time), From field, To field, Date field, Message-ID, X-Mailer	Comparing header fields with syslog	Originator fields, Destination fields, XMailer field, Sender IP, Email subject	Sender address validity, Receiver address (To, CC, BCC), Mail User Agent, Message-ID	Received field # of hops, Span Time, Domain add. Legality, Date & Time Legality, IP add. Legality, sender add. legality, # of Receivers (To, CC, BCC), Mail User Agent, Message-ID, Email subject Date of reception	Span Time, Domain add. Legality, Domain Zone, Date & Time Legality, IP add. Legality, IP Zone, Email Subject, Subject Language Detect, Subject Language Zone, Machine Translate Detect

[10] Omar Al-Jarrah, Ismail Khaterz and Basheer Al-Duwairi, "Identifying Potentially Useful Email Header Features for Email Spam Filtering", ICDS 2012 : The Sixth International Conference on Digital Society.

Email Header Information Analysis

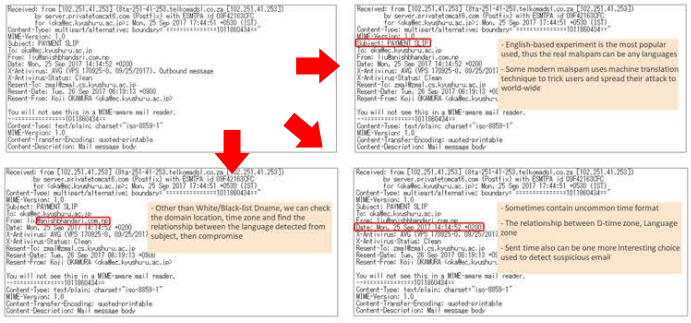


Fig. 2 Example how human being analyze email header

Unlike machine, cybersecurity experts also take consider in email header where suspicious data and their relationship among them are provided. For example:

- A relationship between domain zone and language
- A relationship between time zone and time sent
- Email was written by machine translation detection

Fig. 3 Differnetial of email spawn time between normal email, work email and malspam

From 436 work mails (Green), 4251 normal mails (Blue), and 277 malspam (Red) We can see that most of normal and work mails were sent on work time (8AM-8PM), but the malspam's sent time were varied

Purposed method

Our research focus on developing a new algorithms by using email header information analysis for malspam filtering and also to increase a possibility of zero-day malicious email detection

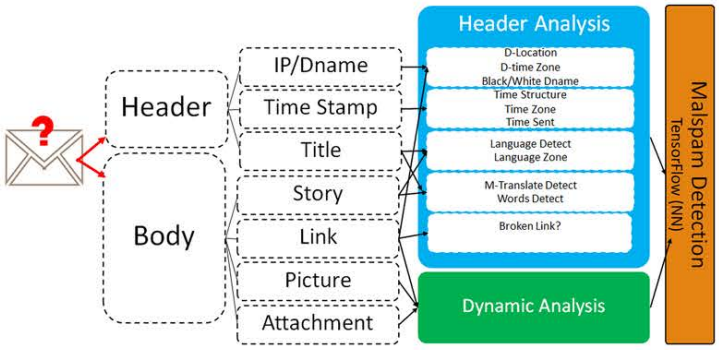


Fig. 4 Design Method for EHIA and Deep-Learning

Email Header Features Extraction

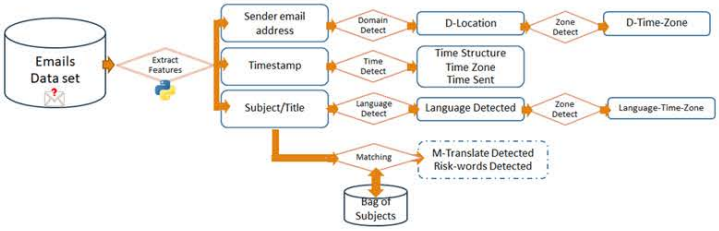


Fig. 5 Features extraction flows chart

From email dataset we first extract 3 features: source address, timestamp and subject
Then we can extract more features from those 3 to get other features in order.

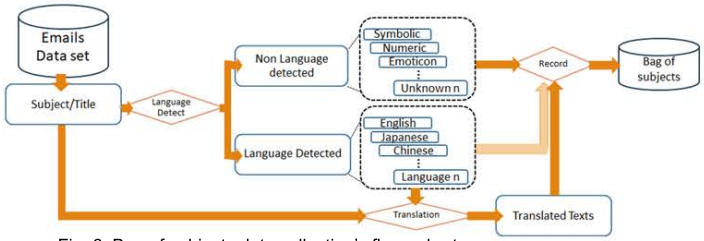
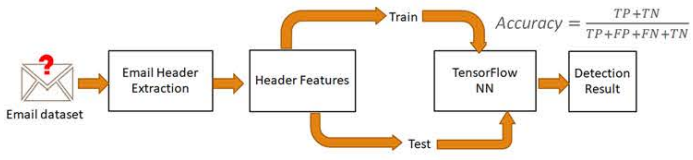


Fig. 6 Bag of subjects data collection's flows chart

Email subject database are created for matching propose and receive M-translate detected, and risk-words detected features

Experiments and Results



Normal Email		Spam Email		Total used	Features		Result
Train	Test	Train	Test				
2350	600	2350	600	5900	1.Sender address 3.D-Zone 5.Time Zone 7.Lang Detected 9.Title	2.D-Location 4.Time structure 6.Time sent 8.Lang Zone	0.7866667

- Normal emails : more than 500,000 (from enron_mail_2015-05, etc.)
- Spam/malspam: more than 500,000 (from http://untroubled.org/spam/ & Cybersecurity Center, Kyushu Univ.)

On progress

Conclusion

In this research, we proposed a method by using new features extracted from email headers and deep-learning approach to detect malspam. From the current experiments, we have not used all the features yet, but we got the best detection result at 78.66% accuracy. Thus, we keep doing more experimentation and improving the method technique to evaluate the detection result

Email: 3IE17305G@s.kyushu-u.ac.jp

九州大学
KYUSHU UNIVERSITY

How Japan's Approach Towards Cybersecurity has Changed: Quantitative Content Analysis of Cybersecurity Strategy from 2013 to 2018

PIYUSH GHASIYA & Prof. KOJI OKAMURA
ISEE, Kyushu University, Fukuoka, Japan

Background

Quantitative Content Analysis helps in obtaining a complete picture and in investigating the features of the data while avoiding the biasness of the researcher.
Moreover, it not only helps in examining the manifest content(visible) but also the latent content (the meaning behind the manifest content) of the text.

Aim of the Research

First, by statistically analyzing Japan's Cybersecurity strategy, the research would try to show how Japan's approach towards cybersecurity has changed over the year.
Next, the researcher will try to analyze whether or not these strategies are effective in making Japanese society safe from the cybersecurity challenges.

Method

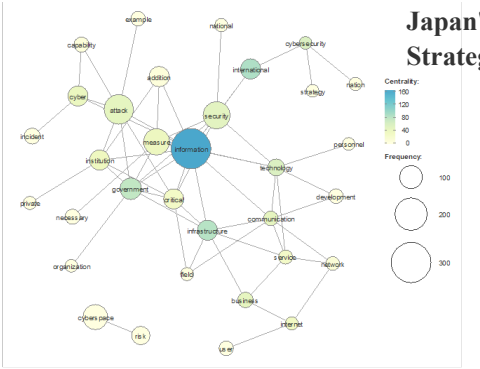
This research will be using KH Coder for analyzing the text (Japan's Cybersecurity Strategy).
Step 1. Extracting the most frequent words.
Step 2. Co-Occurrence Network Analysis of the words with 30 or more frequency.

Term Frequency Analysis

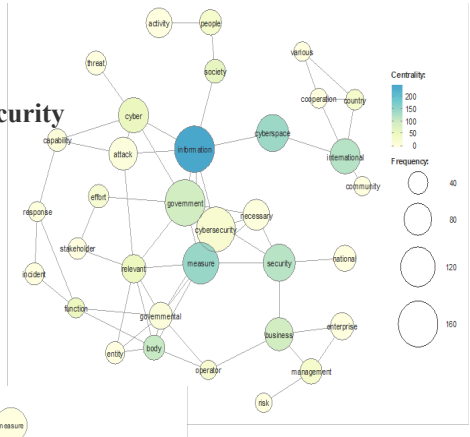
S.No	2013		2015		2018	
	Term	Frequency	Term	Frequency	Term	Frequency
1.	information	304	government	166	cybersecurity	174
2.	attack	165	information	165	government	167
3.	security	142	cybersecurity	153	information	146
4.	measure	132	Japan	146	measure	127
5.	cyberspace	118	measure	132	cyberspace	122
6.	government	90	cyberspace	106	promote	112
7.	cyber	79	security	105	security	89
8.	international	79	include	98	initiative	87
9.	Japan	79	cyber	96	JAPAN	75
10.	addition	78	international	96	service	75
11.	critical	78	business	85	development	73
12.	infrastructure	77	attack	84	technology	71
13.	relate	77	necessary	73	risk	67
14.	institution	76	activity	70	include	63
15.	Security	75	service	66	necessary	63

Co-Occurrence Network Analysis

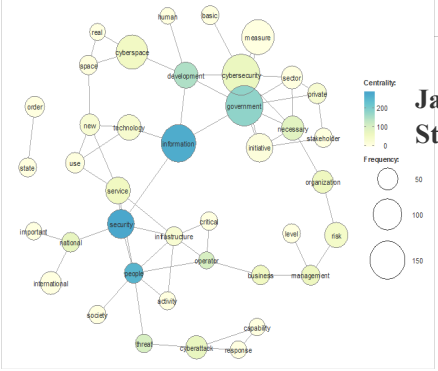
Japan's 1st Cybersecurity Strategy (2013)



Japan's 2nd Cybersecurity Strategy (2015)



Japan's 3rd Cybersecurity Strategy (2018)



The Co-Occurrence Network Analysis shows the centrality (significance) of the particular node (concept or word).
It is clear from the analysis of all three documents that though the policymakers adopted the term "Cybersecurity", however it is acctually "Information Security" that occupies their imagination.

Resources

1. https://www.nisc.go.jp/eng/pdf/cybersecuritystrategy-en.pdf
2. http://www.nisc.go.jp/eng/pdf/cs-strategy-en.pdf
3. https://www.nisc.go.jp/eng/pdf/cs-senryaku2018-en.pdf
4. http://khcoder.net/en/

Presenter's Contact:
Piyush Ghasiya, PhD Researcher,
ISEE, Kyushu University, Fukuoka, Japan
E-mail : piyushghasiya@gmail.com

Internet Conference 2018

A Design of Failure Injection Testing considering Edge Computing Environment

Kenta Hayashi†, Kaori Maeda†, Tohru Kondo‡

† Graduate School of Information Sciences, Hiroshima City University, Japan.
‡ Information Media Center, Hiroshima University, Japan.

Background

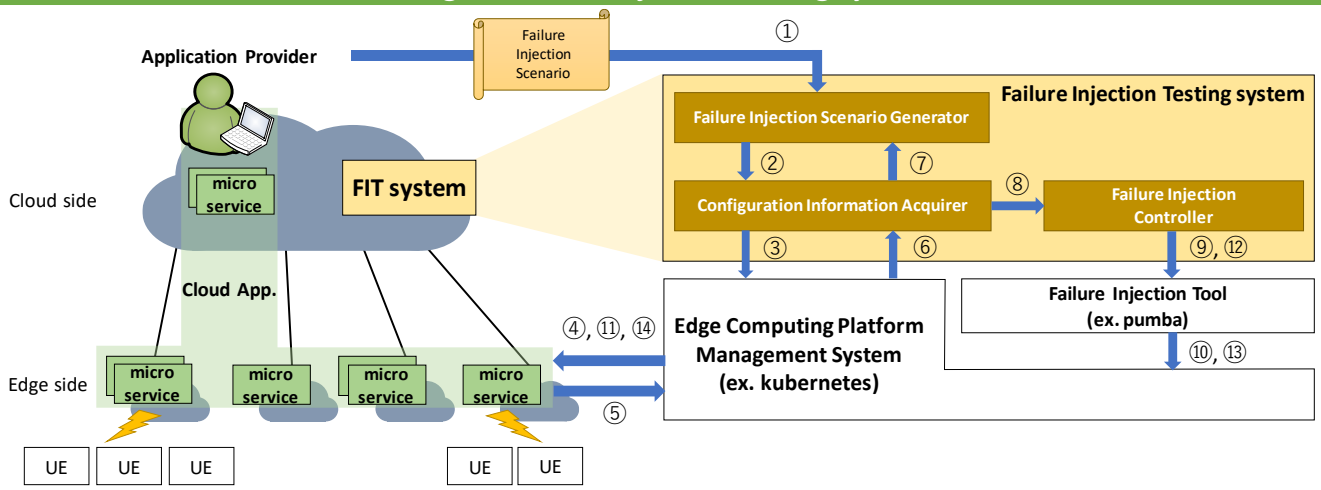
- The current cloud services composed by many distributed micro services[1]
- High complexity of collaboration of micro services
- Requirement for availability and capability without service disruption
- Emerging of Chaos Engineering[2] to improve for resilience of complex distributed systems
- Demand of edge computing for IoT
- Long latency or unstable connections in an edge computing environment have many negative effects on IoT applications
- Edge computing environments including IoT devices have different failure occurrence rates depending on places

The goal of this research

Implementation of a Failure Injection Testing(FIT) system for edge computing environment

- Design of a failure injection scenario which is friendly to application providers
- Implementation of the FIT system based on an arbitrary scenario such as power failure into 50% of edge servers

Design of failure injection testing system



UE (User Equipment) : Edge devices such as sensors

Fig. FIT system configuration

Failure injection flow:

Step 1 (①-⑦) : Acquisition of application configuration information

Step 2 (⑧) : Failure injection preparation

- Creation of failure injection scenario based on configuration information in Step 1

Step 3 (⑨- ⑪) : Failure injection testing based on the scenario which defined in Step 2

Step 4 (⑫ -⑭) : Restoration to the original state

Failure injection scenario

- Indicates structured scenarios for the edge computing environment including the following items
  - Failure injection range (ex. edge side server, access network)
  - Type of failure (ex. packet loss, jitter)
  - Probability of failure occurrence (ex. 50%)
  - Failure injection period (ex. 10 minutes)

Failure injection function

- Micro service operation
  - Micro service stop
    - Slow stop
    - Immediate stop
    - Pause
  - Micro service removal
- Network emulation
  - delay
  - packet loss
  - duplicate
  - corrupt

Future prospects

- Implement a FIT system using pumba[3] for failure injection and kubernetes[4] for edge computing environment
  - Demonstration of the effectiveness of the developed FIT

Reference

[1] Jim Gray, "Why do Computers Stop and What Can Be Done About It?," Tandem Computers Technical Report 85.7, PN87614, June 1985.
[2] Ali Basiri, Niosha Behnam, Ruud de Rooij, Lorin Hochstein, Luke Kosewski, Justin Reynolds, Casey Rosenthal, "Chaos Engineering," IEEE Software, vol.33, no. 3, pp. 3541, May 2016.
[3] "pumba," https://github.com/alexei-led/pumba, (accessed 08/24/2018).
[4] "Production-Grade Container Orchestration, " https://kubernetes.io/, (accessed 08/24/2018).

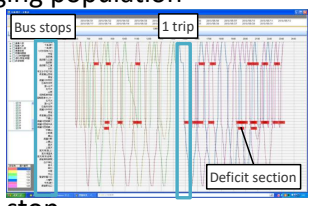
This research and development work was supported by the JSPS KAKENHI Grant Number 15K00122, 16H02808, 18K11266 and MIC/SCOPE #162108102.

Counting Passengers from Images of Drive Recorder Inside Buses by Using Background Subtraction and OpenPose

Hayato Nakashima[1] , Ismail Arai[1] , Kazutoshi Fujikawa[1]
[1] Nara Institute of Science and Technology

Background

- Bus company's management crisis
  - The motorization mainly in local cities
  - The declining birthrate and aging population
  - The progress of depopulation



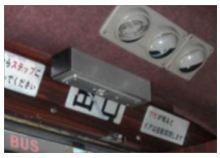
- Optimal route planning
  ex) EAGLE BUS (Saitama, Japan)
  - Count passengers at each bus stop
  - Find out which section on routes is deficit
  - Change bus stop / routes for revenue

Need to know the number of passengers at each bus stop

Problems and purpose

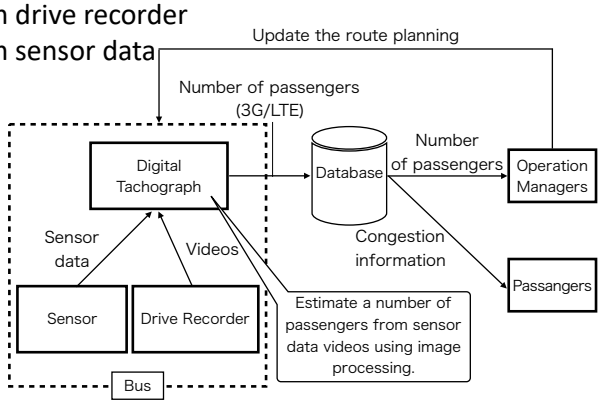
- Past: Manually counting by investigators
  →Only investigate several times a year
- Recent: Automatically counting system
  →Need to install equipment for counting
  →Costly: 300,000 Yen/bus

Propose automatically counting at low cost



Counting method

- Implementation counting passengers
  - From drive recorder
  - From sensor data



Equipment on buses

The bus company in Japan can install equipment with subsidies from the government.

- Drive recorder
  - Verification of accidents and in-vehicle trouble
- Digital tachograph
  - Verification of accidents and in-vehicle trouble
  - Operation management
  - Vehicle data (e.g. vehicle speed, engine speed, GPS etc...) can be stored on memory card or cloud

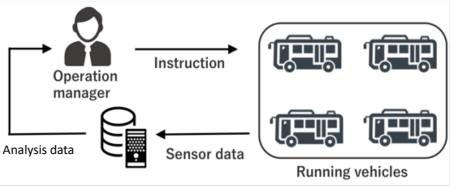


Image analysis method of drive recorder

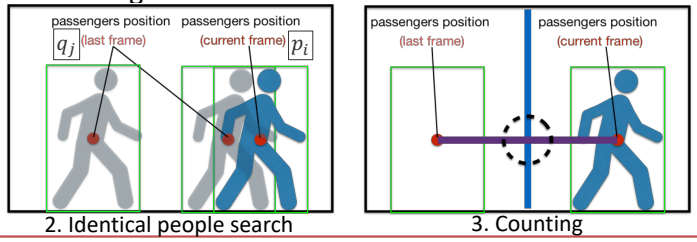
- Try 2 methods
  - Background Subtractor (KNN)
  - Using OpenPose

multi-person keypoint detection library for body

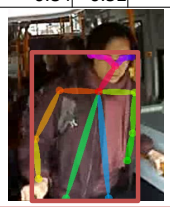
	IN					OUT				
	Correct Peoples	Estimated People	Precision Rate	Recall Rate	F-measure	Correct Peoples	Estimated People	Precision Rate	Recall Rate	F-measure
Background Substraction	20	13	1	0.65	0.64	19	8	0.875	0.37	0.64
OpenPose	69	52	0.75	0.56	0.32	69	43	0.84	0.52	0.64

- Background Subtractor (KNN)
  1. Detection of count target person
     - Calculate rectangular outline and the center of gravity

  2. Identical people search between last & current frames
  3. Counting

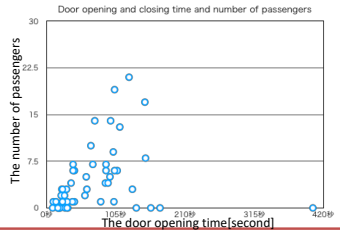


- Using OpenPose
  1. Detection of count target person
     - Using OpenPose
     - Use the position of the neck



Future Work

- Using other sensor data
  - There is a correlation the door opening time for passenger get on/off the bus and the number of passengers.



IDS in the CAN bus using Statistical Analysis and Neural Networks

Araya Kibrom Desta, Ismail Arai, Kazutoshi Fujikawa
Nara Institute of Science and Technology, Inet-Lab

1. Introduction

Present-day vehicles are equipped with multiple Electronic Control Units(ECUs), each of which communicate with one another using a protocol called Controller Area Network(CAN).
Even though, CAN provides its own share of benefits in modernizing automobiles, it also opens a new security hole in the automotive industry. CAN bus doesn't use any mechanism for encrypting or authenticating CAN packets. Security researchers have been able to exploit this security hole to remotely control some critical car components. As a counter measure against this drawback two methodologies of defense, Prevention and Detection, have been proposed. But due to the low processing power of ECUs and a desire of unaltering the CAN de facto standard, we are mainly focusing on a mechanism to detect intrusions inside the CAN bus.

2. Purpose of the Research

The main purpose of this research is to detect cyber attacks inside CAN bus by using different statistical anomaly detection methods and Long Short Term Memory(LSTM) Recurrent Neural Networks(RNN). We will train a neural network to predict subsequent packets, using data from sequences of previously seen messages on the CAN bus. The error difference found by evaluating the actual value and the predicted value will be used for detecting anomalies in a sequence of CAN Packets.

Each CAN packet, along side with other information, has an arbitration ID and timing information. Using this two information and the fact that CAN packets appear in the CAN bus at a fixed frequency[1], we aim to detect malicious message sequences in a fixed time window. Given the available information from the CAN bus and knowledge of attack signatures, we have evaluated some statistical methods that can effectively identify CAN attacks.

3. Replication of existing Researches

To evaluate the methods, we have collected 10 minutes of CAN data from a real car.

1. One Sample/Universal t-test method[2]

This anomaly Detection approach works by calculating statistical data about on going network traffic and comparing them with historical values (μ_{Ht}) measured during training. In every 1 second, we collected the following information for each arbitration ID.

- ✓ ID: arbitration ID
- ✓ N_p : the number of packets in the flow
- ✓ μ_t: the average time difference between successive packets
- ✓ σ^2_t: the variance of the time difference between successive packets
- ✓ T_t: the time difference test value

$$T_t = \frac{\mu_t - \mu_{Ht}}{\sqrt{\frac{\sigma^2}{N_p}}}$$

After we calculated *Tt*, the corresponding p value is solved , and if the p-value is less than a predefined threshold(0.26), the detector alerts the driver to take appropriate measures.

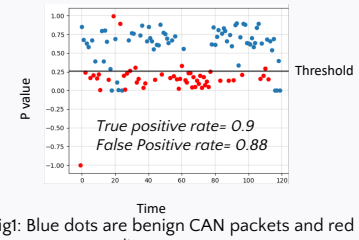


Fig1: Blue dots are benign CAN packets and red dots represent anomalies.

2. IDS using One Class Support Vector Machine[2] ,
we trained OCSVM against N_p, μ_t and σ^2_t

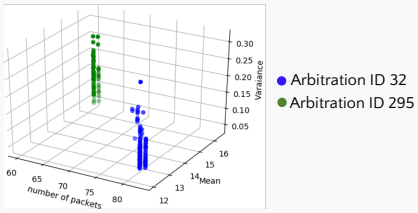


Fig2: data view for IDs 32 and 295

The training gave us good results for both insertion and drop attacks.

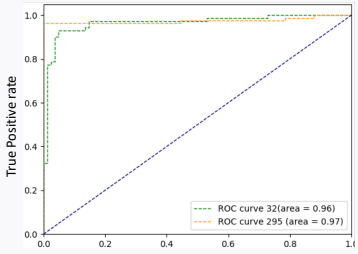


Fig3: ROC curve for arbitration IDs 32 and 295

References

[1] Song, **Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network**. International Conference on Information Networking. Vol. 2016-March IEEE Computer Society, 2016. pp. 63-68
[2] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive CAN bus," in Proc. 2015 World Congress on Industrial Control Systems Security (WCICSS), Dec. 2015, pp. 45–49.

4. Discussion

OCSVM method is more efficient in detecting very short insertion and drop attacks at an acceptable rate, it is even possible to get more practical positive false alarm rates with a higher training data compared to t-test method. In case of the t-test method, its performance can be improved by selecting an optimal threshold value. The training data we used to experiment all the methods are only periodic CAN messages, but all the aforementioned methods fail to detect any anomaly sent with a non-periodic arbitration ID.

5. Future Work

Most of the methods described here mainly focus on periodicity and timing information of CAN packets. None of the methods used any information from the data portion of the packets. And we believe the optimal time that a user should be notified about an intrusion should be in about 1 second. But, for periodic messages which appear in the CAN bus in average of later than 1 second it is impossible to notify the driver before the intrusions cause much more damage.

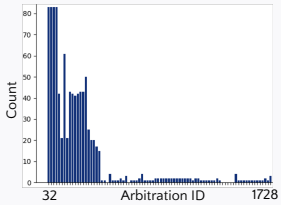


Fig3: Packet count of all arbitration IDs in one second.

In our research we are trying to improve some of the methodologies described here and we will also use sequences of CAN packet data portion to identify anomalies.

A simple strategy for general sequence learning is to use RNN with LSTM[3].

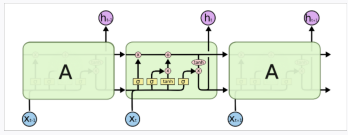


Fig5: Module in an LSTM[4]

With this technology, we will predict data frames using the trained network and depending on how close our prediction was with the actual received data frame, we will determine whether a data sequence has an anomaly or not.

The advantage of using RNN over the statistical methods is, unlike statistical methods RNN have a better anomaly detection capability for a short term anomalies. RNN will also be able to detect anomalies arriving in the CAN bus with non-periodic arbitration IDs.
Furthermore, We will continue to tweak LSTM for better accuracy results and try to continue digging on how to use RNN for identifying anomalies that appear during abnormal car states, like intrusion detection during car crashes. Intrusion detection during this state can be more difficult due to abrupt data packet information changes.

References

[3] F.A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: continual prediction with LSTM. Neural Computation, 12(10):2451–2471, 2000.
[4] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Contact Information

Araya.kibrom_Desta.js3@is.naist.jp
{Ismail,fujikawa}@itc.naist.jp

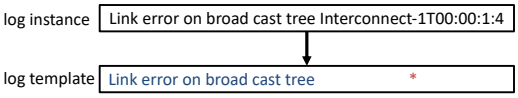
Approach to Better Log Template Generation

Yuya Yamashiro[1], Satoru Kobayashi[2], Kensuke Fukuda[2], Hiroshi Esaki[1]
yuya@hongo.wide.ad.jp, sat@nii.ac.jp, kensuke@nii.ac.jp, hiroshi@wide.ad.jp

1: University of Tokyo, 2: National Institute of Informatics

Introduction

- **Syslog** is widely used for system management
- Huge volume to handle manually
  - 70,000 lines / day (SINET4)
- No “normative grammar”
  - Difficult to extract information from logs
- Translate from raw data to **Log Template**
  - Based on **Software Information**
  - Based on Log Data



Goal

- Automatically generate more accurate template
  - From **open source code**
- Help system operators to extract important information

Dataset

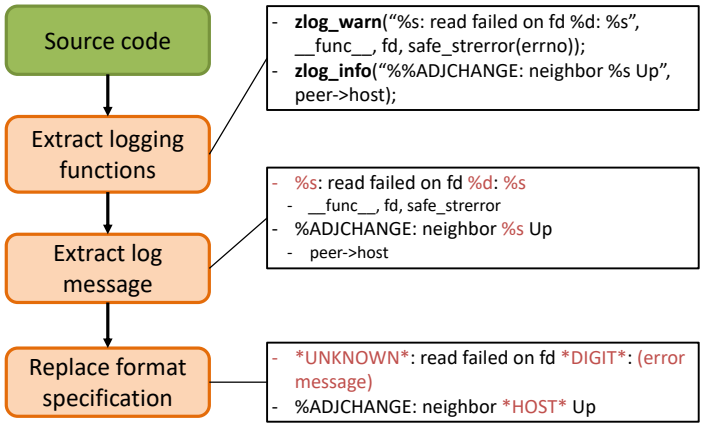
- Vyatta 1.0.8 (napa): Network OS
  - Vyatta Kernel (kernel)
  - Vyatta-quagga (bgpd, ospfd, …)
  - Net-snmp (snmpd)
  - OpenSSH (sshd)
  - Ntp (ntpd)
  - Other many softwares
- Actual vyatta log data of APAN-JP
  - May-June 2018
  - 277,034 lines

Logging Functions

Software	Logging Function
Vyatta Kernel	printk, pr_*
Vyatta-quagga	zlog_*, plog_*
Net-snmp	snmp_log_*, NETSNMP_LOGONCE, DEBUGMSGTL
OpenSSH	fatal, error, sigdie, logit, verbose, debug, debug2, debug3, pam_syslog, helper_log, authlog
Ntp	msyslog

- Difficult to make templates from source code completely automatically

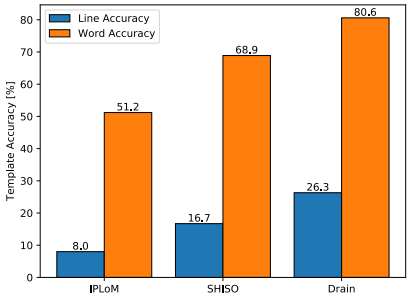
Templates from Source Code



- Total: 189,037 templates

Application of Generated Templates

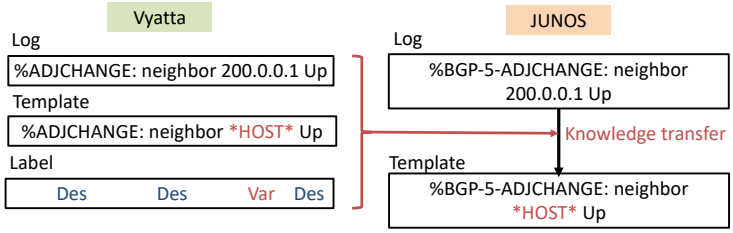
1. Log templates’ accuracy evaluation[1]



a. State-of-the-art clustering based generation algorithms (IPLoM[2], SHISO[3], Drain[4])

b. Use our generated templates as ground truth

2. Transfer learning : from vyatta to junos



References

[1] 山城裕陽, 他 “ソースコードからのネットワークログテンプレート自動生成に関する検討”, 電子情報通信学会IA研究会, p.8, 札幌, 2018
[2] Makanju, A., N. Zincir-Heywood, and E. E. Milios. "Iplom: Iterative partitioning log mining." Tech. Rep. CS-2009-07 (2009).
[3] Mizutani, Masayoshi. "Incremental mining of system log format." Services Computing (SCC), 2013 IEEE International Conference on. IEEE, 2013.
[4] He, Pinjia, et al. "Drain: An online log parsing approach with fixed depth tree." Web Services (ICWS), 2017 IEEE International Conference on. IEEE, 2017.

A method of creating experimental network with routing between virtual hosts

Seiichi YAMAMOTO[1], Eiji KAWAI[2]

1) yama@wide.ad.jp
- National Institute of Information and Communications Technology
- The University of Tokyo

2) eiji-ka@nict.go.jp
- National Institute of Information and Communications Technology

Summary
- A method of creating specific network route for some experiment with virtual hosts.
- Traffic engineering on IP routing structure with docker image.
- L2VPN on IP routing provides Layer 2 connection between the hosts.

Background
- Generally, a tunnel connection between hypervisors is used for network experiment between virtual hosts.
- The tunnel connection is the shortest path between hypervisors.
- So that it is difficult to make flexible path creation.
  - Ex) specific path for NFV verification,
    long-distance path for inter-planet network experiment.

Proposal
- Create router as traffic engineering point (like a “hinge”).
- Use IP routing, because of the ease of creating tunnel and selecting path.
- Practically, Use docker router image.
- “IP routing” is one of the choice to do a traffic engineering.
- However, “Segment routing” would be the another candidate to do the traffic engineering.

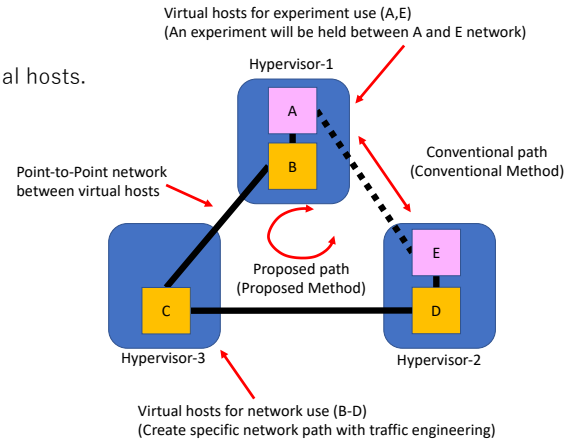


Fig 1. Proposed system

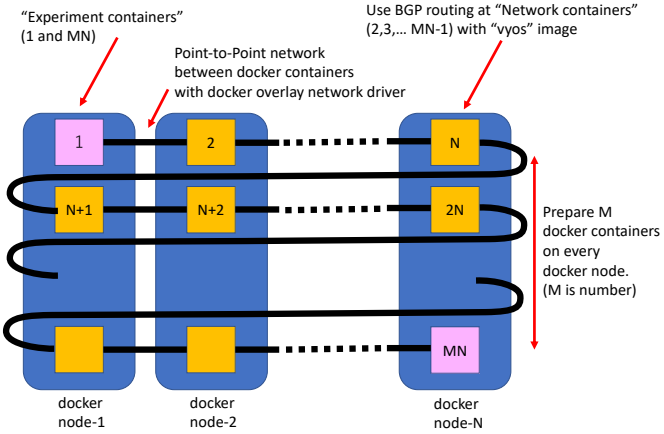
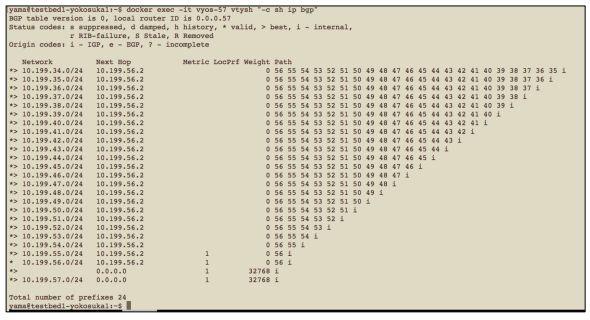


Fig 2. An example of network through multiple containers

Implementation
- Use virtual machines or physical machines as docker nodes (i.e. Hypervisors).
  - Prepare docker swarm cluster to do unified control all docker containers.
  - Use docker-machine tool at docker manager node to control container all together.
- Prepare 2 type of containers, “experiment use” and “network use”.
- Create point-to-point network segment between containers with docker overlay network driver.
- Create IP routing network between network containers.
  - BGP routing is used with IP router container image (vyos) on network container.
  - Use vyos L2VPN, when Layer 2 connection between experiment containers is needed.
  - Detailed policy control is available with BGP routing.
- Provide experiment containers with IP routing network connection.



```
yama@testbed1-yokosuka1:~$ docker exec -it vyos-57 vtysh "-c sh ip bgp"
BGP table version is 0, local router ID is 0.0.0.57
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.199.34.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 i
*> 10.199.35.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 i
*> 10.199.36.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 i
*> 10.199.37.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 i
*> 10.199.38.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 i
*> 10.199.39.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 i
*> 10.199.40.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 i
*> 10.199.41.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 i
*> 10.199.42.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 49 48 47 46 45 44 43 i
*> 10.199.43.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 49 48 47 46 45 44 i
*> 10.199.44.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 49 48 47 46 45 i
*> 10.199.45.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 49 48 47 46 i
*> 10.199.46.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 49 48 47 i
*> 10.199.47.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 49 48 i
*> 10.199.48.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 49 i
*> 10.199.49.0/24   10.199.56.2                            0 56 55 54 53 52 51 50 i
*> 10.199.50.0/24   10.199.56.2                            0 56 55 54 53 52 51 i
*> 10.199.51.0/24   10.199.56.2                            0 56 55 54 53 52 i
*> 10.199.52.0/24   10.199.56.2                            0 56 55 54 53 i
*> 10.199.53.0/24   10.199.56.2                            0 56 55 54 i
*> 10.199.54.0/24   10.199.56.2                            0 56 55 i
*> 10.199.55.0/24   10.199.56.2             1              0 56 i
*  10.199.56.0/24   10.199.56.2             1              0 56 i
*>                  0.0.0.0                  1          32768 i
*> 10.199.57.0/24   0.0.0.0                  1          32768 i

Total number of prefixes 24
yama@testbed1-yokosuka1:~$
```

Fig 3. Routing information through 24 containers

Reference
- Docker
  - https://www.docker.com/
- Mininet
  - http://mininet.org/
- Linux namespace
  - http://man7.org/linux/man-pages/man7/namespaces.7.html
- openstack neutron
  - https://docs.openstack.org/security-guide/networking/architecture.html
- segment routing
  - http://www.segment-routing.net/

Past presentation
- Seiichi YAMAMOTO, 活用研究会を通じたNICT総合テストベッド利用体験事例, スマートIoT推進フォーラム 技術戦略検討部会 テストベッド分科会, Sep. 2018. (Japanese)
  - https://testbed.nict.go.jp/bunkakai/pdf/bunkakai-05-04.pdf

Dynamic Adaptation of Cooldown Period for Auto Scaling of VNFs

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Mohit Kumar Singh, Gaurav Garg, Tulja Vamshi Kiran Buyakar, Venkatarami Reddy, Antony Franklin A, and Bheemarjuna Reddy Tamma

Department of Computer Science and Engineering, IIT Hyderabad, India

Introduction

- Recently, network operators and data centers have moved towards Virtualization of Network Functions (VNFs).
- An Auto Scaling Entity (ASE) monitors the load on the VNF and scale up services depending on the usage of the VNF instance.
- ASE checks the load at each time interval called monitoring period and the VNF needs to wait for cooldown period to ensure that scaling action takes place.
- Existing algorithms, used by the operators to scale the VNFs, use a static cooldown period.
- We propose an algorithm with the dynamic adaptation of cooldown period for scaling of the VNF.

Simulation

- We generate number of HTTP requests based on poisson distribution model to a docker acting as an HTTP server and then, monitoring the CPU utilization of the docker.
- We compare the performances of both the algorithms in terms of number of instances instantiated and number of monitoring requests made over time.

Conclusions

- We conclude that the proposed algorithm is better than the existing approach, where unnecessary scalings are avoided.
- This is because the proposed algorithm checks the utilization at short and regular intervals. This enables the ASE to take the decision at the time when scaling is actually needed.
- However, the proposed algorithm induces an overhead in terms of monitoring requests to CPU.
- As future work, we will try to apply machine learning models to predict the behaviour of traffic to take effective scaling decisions.

Algorithms

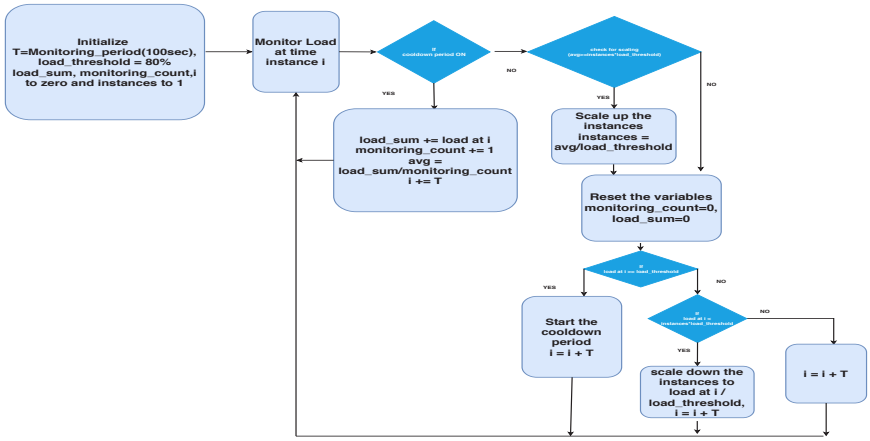


Fig. 1: **Existing Scaling Algorithm.**

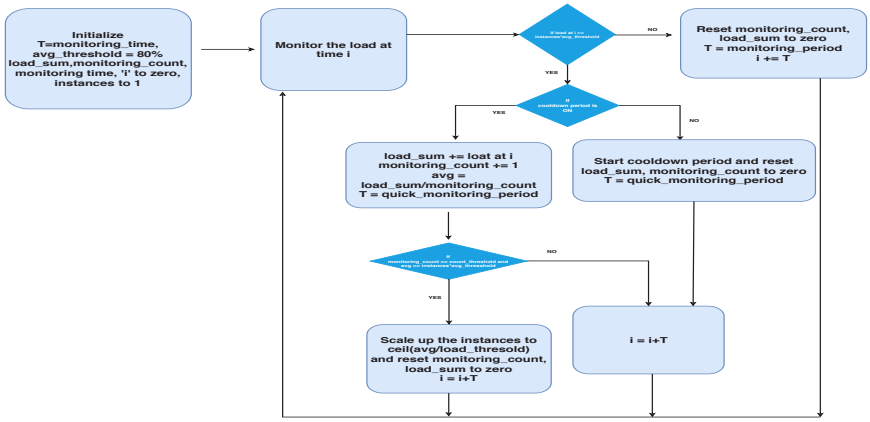
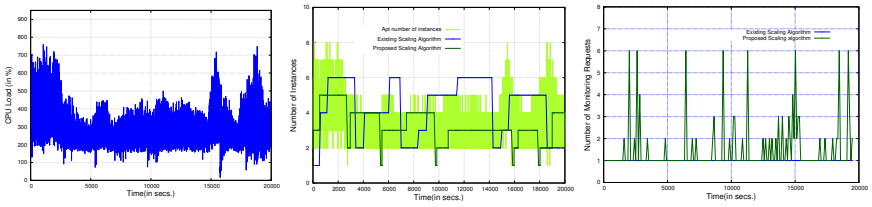


Fig. 2: **Proposed Scaling Algorithm.**

Results



(a) **CPU util. of docker.** (b) **# Active instances.** (c) **# Monitoring requests.**

Fig. 3: **Evaluation of the proposed algorithm**

- Fig. 3 (a) shows the CPU load pattern on the server, on which we test the algorithms.
- Fig. 3 (b) shows how do both algorithms react to traffic. The proposed algorithm uses less number of instances at the times when less number should have been used, and vice-versa.
- Fig. 3 (c) shows number of CPU monitoring requests made in each interval. A total of 200 and 394 monitoring requests have been made by existing and proposed algorithm respectively.

Collecting a great number of active IPv6 addresses

Yudai Aratsu[1] aratsu@hongo.wide.ad.jp
Satoru Kobayashi[2] sat@nii.ac.jp
Kensuke Fukuda[2] kensuke@nii.ac.jp
Hiroshi Esaki[1] hiroshi@wide.ad.jp

1:The University of Tokyo 2: National Institute of Informatics

Background

- Scanning whole IPv4 address takes only 45 min.
- IPv6 has vast address space.
  - to scan efficiently, active IPv6 address Hitlist is needed.

Goal

- Generating IPv6 address Hitlist by employing various methods.

How to collect a large number of active IPv6 addresses?

Methods

- Using server logs
  - Web, Mail, NTP, DNS
- Crawling P2P network
  - BitTorrent, Bitcoin
- rDNS enumeration [1]

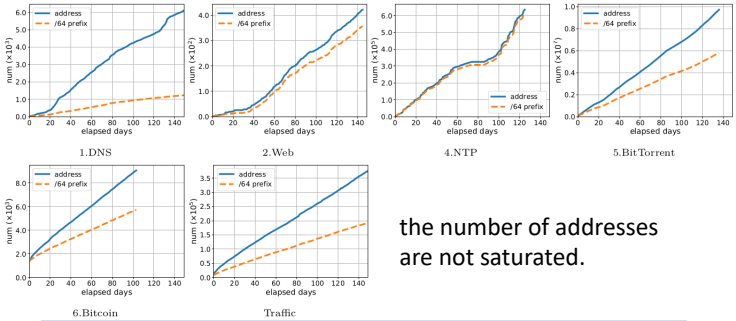
Since Jun, 2018

Result [2]

- Total numbers

		Period	#Address	#/64 prefix	#AS
1.	DNS	151 days	6,155	1,245	418
2.	Mail	146 days	1	1	1
3.	Web	149 days	422	357	101
4.	NTP	123 days	634,941	613,011	341
5.	BitTorrent	137 days	9,734,709	5,813,622	1,981
6.	Bitcoin	104 days	4,428	3,100	668
7.	rDNS enumeration	55 days	7,564,320	118,844	582
	Total	151 days	17,948,250	6,549,576	2,545
	Traffic (mawi)	151 days	377,409	193,168	4,902

- Time evolution



the number of addresses are not saturated.

- IID based classification

Type	1.DNS	2.Web	4.NTP	5.BT	6.BC	7.rDNS	Traffic
"0000"	92.6%	20.6%	14.5%	13.0%	29.1%	91.3%	27.2%
"fffe"	2.3%	0.7%	3.5%	7.0%	9.8%	1.1%	9.8%
Others	5.1%	78.7%	82.0%	80.0%	63.1%	7.6%	63.1%

IP addresses of server or client?

- Response rate (icmp6)

1.DNS	2.Web	5.BT	6.BC	7.rDNS
65.8%	8.1%	0.1%	22.3%	0.2%

mostly IP addressess do not respond.
(not stable?)

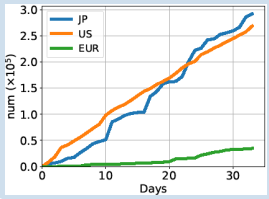
- collecting in three countries
  - Japan (Tokyo), US (California), Europe(Netherland)
  - since Sep, 2018

result of NTP server (33 days)

country based collected address classification

JP server		US server		EUR server	
IN	72.2%	US	84.5%	BR	31.4%
SA	10.1%	IN	11.1%	EU	30.0%
VN	3.2%	BR	0.7%	AT	11.3%
JP	2.5%	CN	0.6%	MX	7.8%
CN	1.5%	GT	0.4%	AR	4.9%

time evolution



Huge bias among server location

[1] T. Fiebig, et al. "Something from nothing (There): Collecting global IPv6 datasets from DNS." PAM'17
[2] 新津, et al. "大規模IPv6アドレス収集手法の検討" 信学技報 2018.09
[3] P. Foremski, et al. "Entropy/IP: Uncovering Structure in IPv6 Addresses" IMC'16
[4] O. Gasser, et al. "Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists" IMC'18

Discussion

How to collect more IPv6 addresses?
- multiple location
- more service
- employing machine learning [3]

How to generate "high quality" Hitlist? [4]
- collecting stable IPv6 addresses
- pseudo active space detection

Fast Logging in Time Series for a Computer Security Incident Response

Motoyuki OHMORI†
†Center for Information Infrastructure and Multimedia, Tottori University

鳥取大学
Tottori University

Abstract

- Time Series Database (TSDB) dedicated for a computer security incident
- Lower storage and faster search
- Performance improvement adopting logging message normalization
- Lock-free clustering support for scalability

Background

In order to avoid data breach, it is important to quickly and accurately identify and confine a suspicious host when a computer security incident happens. When an external organization alerts a suspicious host, an IP address of the host is given. In order to identify the suspicious host, ones may then search for the IP address in related logs. Searching time of logs is then important to shorten a delay to identify a host. To this end, there have been already several logging systems such as fluentd, kibana, and splunk that are based upon text messages of syslog.

These existing logging systems are, however, not so efficient. For example, we have experienced that fluentd caused high CPU usages and lost log messages. In addition, these existing logging systems are not dedicated for a computer security incident. They, therefore, need more time to search for required logging messages for an incident.

Research Goals

Research goal is implementing a fast logging database dedicated for a computer security incident that has:

1. ***Scalable logging database that requires smaller storages.***
2. ***Fast search especially for recent logging messages***.

Basic Ideas of the Fast Logging Database

1. ***Binary Based Key Value Store***

   – **Low performance of existing logging systems based on text message**

   Existing logging systems are basically based upon text messages while logging messages of network or security equipment usually are in pre-defined text format. The fast logging database then stores binary values only in a record in a table, and text messages are indexed in another table.

2. ***Time Series Database (TSDB)***

   – **Each record always has a timestamp**

   Since each logging message must have a timestamp, the fast logging database always stores the timestamp as a primary key. All records are basically stored in ascending order of timestamps. Some records are, however, not strictly in ascending order for lock-free operation described later.

3. ***Fixed Record Length***

   – **Length of all records in a table is always same**

   In order to improve search performance, the fast logging database has the same record length for a table as same as recent Relational Database Management System (RDBMS).

4. ***Timestamp Index***

   – **A location of a record at a timestamp is indexed**

   Regarding searching a record, a timestamp is usually specified for a computer security incident. In order to improve searching speed, a location of a record at a timestamp is indexed. Since the number of logging messages may depend upon daytime or night, timestamp index improves searching a record of a specified timestamp.

5. ***Logging Message Normalization***

   – **Logging message format are automatically normalized:**

   Because a logging message is usually output using ***printf*** functions, the fast logging database indexes a message format. A Logging message is then stored as a tuple of a message format index and variable values, i.e., variable arguments of ***printf***.

6. ***Lock-Free Clustering Support***

   – **Lock free insertion and lookup**

   The fast logging database does not strictly consider an order of a logging message. Timestamps in records are, therefore, not always in ascending order. This nature may delay to finish all search in order to make sure that the all log messages in specified timestamp in search are examined. This nature, however, makes insertion and lookup operations *lock-free*. Lock-free clustering can be then archived.

7. ***Recent in Memory and Old in Disk***

   – **Recently added records are in memory, and older records in disk**

   When a computer security incident happens and quick response is necessary, recent logging messages are searched in most cases. Older messages are not required to be fast to be searched because its search itself is enough delayed already. The fast logging database then always keeps recent logging messages in memory as much as possible, and write the messages to a disk if possible or all memory is consumed. Even after older messages are searched once, these older messages are not in memory in order to prioritize quick response for a recent computer security incident.

Overview of the Fast Logging Database

Figure 1. depicts overview of the proposed fast logging database. The fast logging database consists of multiple database servers. All multiple database severs has the same IP address for IP anycasting. Network equipment or other servers (e.g., Web server, mail server and so on) send logging messages to the fast logging database using syslog protocol. The logging messages are then stored into one of database servers.

When one, say a CSIRT member, searches for logging messages, one sends a search request to one of database servers. The database server receiving a request forward the request to the other database servers. All database severs then sends responses back to one who sends a search request.

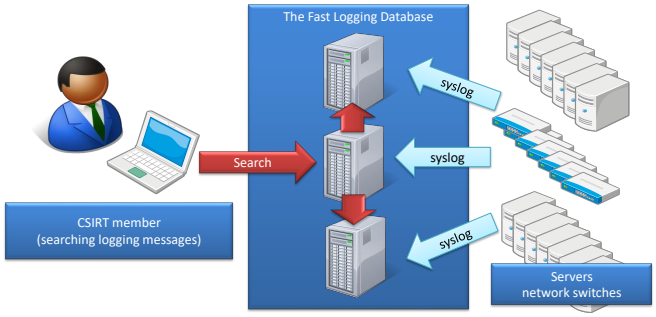


Figure 1. overview of the fast logging database

Development of Local Cloud Environment in the User Vicinity

Tomohiro Yoshida †1 RANDRIANARIVONY Nirinarisantatra†1
Teruaki Yokoyama †2

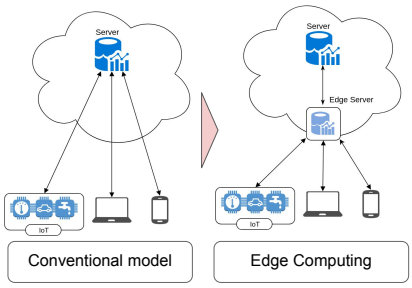
†1 Kobe Institute of Computing Graduate School of Information Technology
† 2 National Research and Development Institute of Information and Communications Technology

Background

As the Internet environment has evolved, terminals connected to cloud services are increasing. This situation is facing new challenges. In order to solve problems such as communication delay, has emerged a processing model called edge computing.

However, in order to perform edge computing, it is required to have the server near the user. Neighboring placement on the Internet can be achieved by installing a server at a data center or the like, while physical edge computing outside the Internet is another way which is used as the de facto standard in construction and operation. With the spread of IoT devices, the demand for local intra-communication that does not need to connect to the Internet is rising, but it is time-consuming and expensive to operate such communication network, especially the construction and the control mechanism in the physical environment.

This study is focusing on the research and development of
physical infrastructure systems for local environment



Conventional model | Edge Computing

Issues

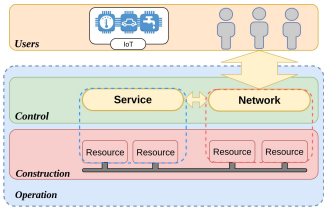
There are some challenges in Service delivery for local environment, communication platform for IoT devices using edge computing, and physical infrastructure implementation

Outdoor, mountains and fields etc...

Construction	Challenges on installation / Reproducibility OS and network equipments set-up for a specific area
Operation	Mobility and failure in local. Configuration expansion issue or hardware upgrade problem.
Control	Flexible control mechanism tailored to the situation Deployment of networks that can accommodate users and IoT devices.

Solution

We propose a compact and inexpensive portable cloud that can solve the problem of physical infrastructure. The scalability and migratory are considered, and it's cheap and it can be exchanged easily. More than one single board computers are used and mounted.

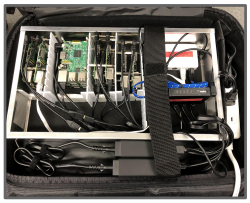


Construction	Since it is possible to prepare multiple units and to unify settings, OS installation and network construction
Operation	Compact size makes it easy to handle and is ideal for moving Because it is inexpensive and easy to obtain, it can cope with trouble, etc.
Control	Services and networks according to requirements. Systems that can be flexibly controlled and managed.

Method

Construction / Operation — Physical environment

Consideration of scalability, mobility, easy to replace with single board computer Multi-equipement implementation.

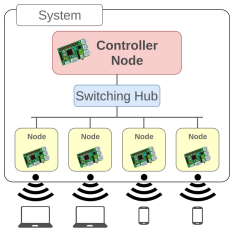


1st Prototype System Configuration
(9 units)
2 small switching hubs

2nd Prototype System Configuration
(5 units)
1 small switching hubs

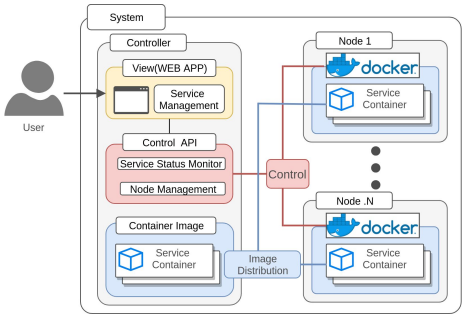
Operation / Control — Network provision

It provides a network for user accommodation the nodes operate as a single board computer



Available access points depending on the number of connected terminals.
can adapt according to the needs.

Control — Service delivery



Internal service provisioning is implemented with Docker. It is possible to activate required applications only, depending on the usage / situation giving more flexibility

Future Works

- Implementation of functions that enable to backup and set up of processed data in the local area using the internet or the data Storage media.
- Improvement of the system configuration from current bag to physical

Acknowledgment: A part of this research was conducted with the support of the Information Technology Promotion Agency (IPA) MITOU Project 2018. https://www.ipa.go.jp/jinzai/mitou/portal_index.html

WIDE