An International ICT R&D Journal Sponsored by ZTE Corporation

ISSN 1673-5188
CN 34-1294/ TN
CODEN ZCTOAK

ZTE COMMUNICATIONS

tech.zte.com.cn

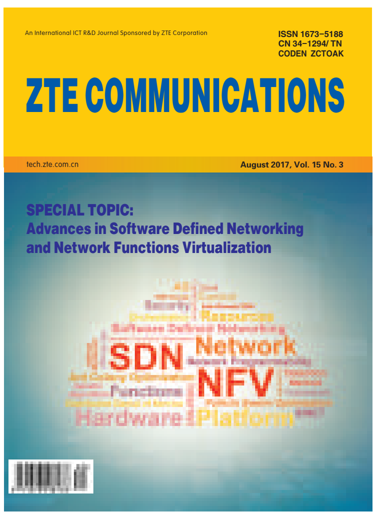
August 2017, Vol. 15 No. 3

SPECIAL TOPIC:
Advances in Software Defined Networking and Network Functions Virtualization

ZTE中兴 ZTE COMMUNICATIONS
VOLUME 15 NUMBER 3 AUGUST 2017

ZTE Communications Editorial Board

CONTENTS



Special Topic: Advances in Software Defined Networking and Network Functions Virtualization

CONTENTS

ZTE COMMUNICATIONS
Vol. 15 No. 3 (Issue 58)
Quarterly
First English Issue Published in 2003

Supervised by:
Anhui Science and Technology Department

Sponsored by:
Anhui Science and Technology Information Research Institute and ZTE Corporation

Editorial Correspondence:
Add: 12F Kaixuan Building,
329 Jinzhai Road,
Hefei 230061, China
Tel: +86-551-65533356
Fax: +86-551-65850139
Email: magazine@zte.com.cn

Published and Circulated (Home and Abroad) by:
Editorial Office of *ZTE Communications*

Printed by:
Hefei Tiancai Color Printing Company

Publication Date:
August 25, 2017

Publication Licenses:

ISSN 1673-5188
CN 34-1294/ TN

Annual Subscription:
RMB 80

Review

Research Paper

Guest Editorial

ZHANG Honggang

Advances in Software Defined Networking and Network Functions Virtualization

ZHANG Honggang

ZHANG Honggang is a full professor with the College of Information Science and Electronic Engineering, Zhejiang University, China. He was International Chair Professor of Excellence for Université Européenne de Bretagne (UEB) and Supélec, France (2012–2014). He is an honorary visiting professor at the University of York, U.K. He is currently active in the research of cognitive green communications and was the leading guest editor of the *IEEE Communications Magazine* special issues on "Green Communications". He is taking the role of Associate Editor-in-Chief (AEIC) of *China Communications* as well as the series editor of *IEEE Communications Magazine* for its Green Communications and Computing Networks Series. He served as the chair of the Technical Committee on Cognitive Networks of the IEEE Communications Society from 2011 to 2012. He was the co-author and an editor of two books with the titles of *Cognitive Communications—Distributed Artificial Intelligence (DAI), Regulatory Policy and Economics, Implementation* (John Wiley & Sons) and *Green Communications: Theoretical Fundamentals, Algorithms and Applications* (CRC Press), respectively.

Software defined networking (SDN) and network functions virtualization (NFV) are being heralded as a fundamental leap forward, which will substantively transform the landscape of the whole telecommunications and networks area. A number of key technical advancements for network softwarization and virtualization, commoditizing networking hardware, simplifying operations, and reducing capital and operational expenditures have been quickly emerging on the fronts.

Basically, decoupling the control plane from the data plane as well as decoupling the network functions from the hardware implementation using virtualization and service abstraction possess distinctive networking flexibility and capability, but also point to a number of functional and operational challenges.

The key target of this special issue is to cover the broad "spectra" of SDN and NFV technologies from various perspectives, ranging from NFV-based emulation platform to inter-domain SDN as well as from Distributed Denial of Service (DDoS) attacks to evolutionary algorithms in software defined networks.

The first article, by LI Haifeng, LI Taixin, and ZHANG Hongke, "A Large-Scale NFV-Based Emulation Platform for Smart Identifier Network" describes a large-scale emulation platform for the proposed Smart Identifier Network (SINET). Specifically, the authors pool and virtualize all hardware resources based on lightweight virtualization technologies, achieving flexibly programmable and dynamically configurable emulation platform via designing the controller and orchestrators to manage emulation tasks and to collect emulation results.

The next article "Survey of Mechanisms for Inter-Domain SDN" by WANG Yangyang and BI Jun highlights the major approaches for extending the SDN mechanism to the inter-domain stages. The deployment of inter-domain SDN and relevant technical problems are discussed with details. Moreover, the authors explain potential applications and technical challenges in various inter-domain SDN scenarios.

The third article "DDoS Attack in Software Defined Networks: A Survey" by XU Xiaoqiong, YU Hongfang, and YANG Kun provides an in-depth survey on how to take advantage of SDN techniques to solve DDoS threats to Internet security. Concretely, various SDN-supported mechanisms against DDoS attacks are introduced via a systematic review of the existing literatures.

Finally, the article titled "Evolutionary Algorithms in Software Defined Networks: Techniques, Applications, and Issues" by LIAO Lingxia, Victor C. M. Leung, and LAI Chin-Feng introduces four types of evolutionary algorithms (EAs): Genetic Algorithms, Particle Swarm Optimization, Ant Colony Optimization, and Simulated Annealing, which can be widely applied to dealing with the complex optimization problems for huge scale of SDN networks. Within the framework of SDN, these four types of EAs are presented by illustrating their key techniques, summarizing their typical applications, and highlighting their future challenges.

We would like to thank all authors for their significant contributions. It is hoped that the articles in this issue will surely stimulate more and more readers of *ZTE Communications* to actively take part in this promising research area from now on.

DOI: 10.3969/j. issn. 1673-5188. 2017. 03. 001
http://kns.cnki.net/kcms/detail/34.1294.TN.20170824.1054.002.html, published online August 24, 2017

A Large-Scale NFV-Based Emulation Platform for Smart Identifier Network

LI Haifeng, LI Taixin, and ZHANG Hongke
(Institute of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China)

Abstract

Network emulation is significant because of its ability to study network architecture running on real operating system and the whole protocol stack. However, conservatively allocating a physical network equipment for a corresponding emulation network is costly, scale-limited and rigid. In this paper, based on network function virtualization (NFV), we present a large-scale emulation platform for the Smart Identifier Network (SINET). We pool and virtualize all hardware resources by lightweight virtualization technology. Controllers and orchestrators are designed to manage emulation and collect monitored SINET emulation results. The controllers are used to implement dynamically synchronously management for large-scale emulation network and link characteristics. The orchestrators are utilized to achieve overload avoidance and green computing with the Xiao's dynamic resource allocation algorithm. We implement flexibly programmable and dynamically configurable emulation.

Keywords

NFV; SINET; emulation; virtualization; control; orchestration

1 Introduction

The current problems of the Internet are a natural consequence of its architecture, which was designed to address the simple data communication needs 40 years ago [1]. The basic requirement of the Internet at that time was merely forwarding data packets among limited trust hosts. With the growth of the Internet, enormous technological innovations in both the application and network layers have emerged, which gives rise to new requirements from the architecture, such as support for trust, security, mobility and low energy consumption. However, the Internet was never designed to meet these requirements. To help evolve the Internet, there are enormous patches made for the current Internet architecture, such as Transport Layer Security (TLS), Content Delivery Network (CDN) and Network Address Translation (NAT). Unfortunately, these patches cannot fundamentally solve the existing shortcomings, and even make the current Internet more and more complex and bulky [2].

An alternative way to address new requirements is designing a new clean-state architecture for the Internet. Along this way, there have been a large number of research efforts that are proposing architectures for the Future Internet [2]–[5]. In this context, the Smart Identifier Network (SINET) has emerged as a promising architecture for the Future Internet [5]. SINET completely removes the restrictions from the triple bindings, including resource and location binding, user and network binding, and control and data binding. Besides, SINET creates three layers vertically and two domains horizontally to fundamentally solve the current Internet problems. Therefore, it can meet new requirements raised by the tremendous growth of the Internet and introduction of new scenarios.

Recently, great progress has been made in SINET in many respects [6], [7]. SINET has been applied for 5G communications [6], and it provides a specific SCN-R scheme to promote efficient and reliable communications in High-Speed Railway (HSR) scenarios. The concept of SINET was also used to deal with mobility-caused outdated mappings [7]. It designed a timer-based pointer forwarding (TBPF) approach to avoid the triangular routing problem. However, as a new clean-state architecture, SINET confronts several technological problems that have not been investigated thoroughly, and still has a long way ahead.

To further promote SINET development, an emulation platform for SINET is urgently needed. In contrast to simulators, emulation platforms provide more realistic environment for developing protocols and techniques in the new network architecture [8], [9]. Typically, an emulation platform uses real hardware resources to virtualize various network equipment and network environments. This means that researchers can use real operating systems, software and protocol stacks to run their experimental network protocols, in order to achieve actual (not simulated) performance measures.

In this paper, we propose an emulation platform for SINET, based on network function virtualization (NFV) technologies [10]. The emergence of NFV simplifies the design of emulation platforms. Besides, NFV endows an emulation platform with more programmable and flexible ability. On the basis of NFV technologies, we pool and virtualize all hardware resources, which include computing, network and special hardware resources. The pooled virtualized resources are managed by controllers and orchestrators of the emulation platform to instantiate SINET emulation.

Our goal is to implement a directly programmable, dynamically configurable and green computing emulation platform for SINET. Programmability is to assure that we can emulate SINET protocols as flexibly as general simulation (e.g., ns-3) can. Dynamical configuration means that emulation topologies and link characteristics can be dynamically and synchronously configured and managed during the whole emulating process. Finally, green computing aims at effectively preventing overload while saving energy used.

To implement preceding goals, we employ the state-of-art techniques to build the NFV-based SINET emulation platform. First, we adopt the separation technique of control and data planes. We decouple the control network and emulation network, enabling the emulation network directly programmable and the underlying physical infrastructure to be abstracted for SINET emulation applications. Second, we design a controller to monitor and manage emulation topology and link characteristics. By running a distributed sleep and busy waits algorithm, the controller implements dynamical and synchronous configuration for large-scale emulation network and link characteristics. Finally, we adopt Xiao's dynamic resource allocation technique [16] into our SINET emulation platform. With this algorithm, we can achieve both overload avoidance and green computing.

The remainder of this paper is organized as follows. Section 2 overviews SINET architecture and related emulation platforms. Section 3 gives some details of our emulation platform including overall structure, emulation control and orchestration. Conclusions are given in Section 4.

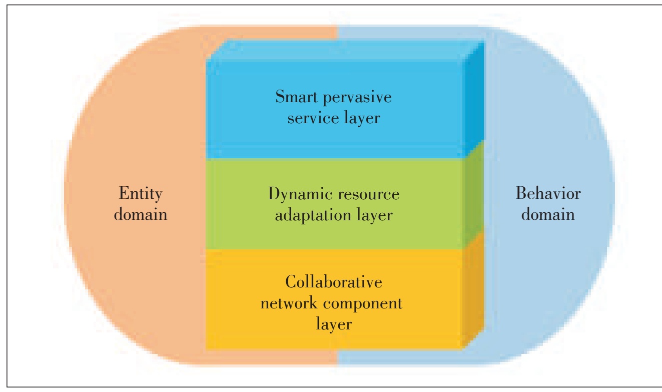
2 Related Work

This section introduces the architecture of SINET and related emulation platforms.

2.1 SINET Architecture

The architecture of SINET (**Fig. 1**) is divided into three vertical layers and two horizontal domains.

The three-layer structure consist of the smart pervasive service layer, dynamic resource adaptation layer, and collaborative network component layer. The smart pervasive service layer is in charge of the naming, registration, and management of various services, including video streaming, web surfing, instant chatting, and file downloading. The dynamic resource adaptation layer is used to manage and organize network function groups, which are in charge of the optimal decision, task allocation, and resource scheduling. Each function group consists of a set of network nodes or components with similar functions such as mobility support, security enhancement, in-path caching, and energy saving. The collaborative network component layer enables network nodes such as routers, content servers, sensors, and interfaces to carry out a specific task, including routing, data transmission/caching, mobility proxy, and power saving (in sleep mode).



▲**Figure 1. SINET reference model.**

The two domains are the entity domain and behavior domain. The entity domain aims at providing identifiers of entities such as network services, function groups, and network nodes. The behavior domain describes properties of different entities, including service characteristics such as service type and service cache and provider signatures.

With three layers and two domains, SINET dynamically gets network status and intelligently meets service requirements. Network function groups and network components in a function group can be properly selected. Hence, SINET can provide smart pervasive service. Meanwhile, network strategies, such as behavior matching, behavior clustering and network complex behavior game are adopted to achieve dynamic adaptation and cooperative scheduling for network resources. In this way, SINET greatly increases the utilization of network resources, further reduces the consumption of energy and improves user experience.

2.2 Related Emulation Platform

There are a number of emulation platforms for different network architecture and purposes [8], [9], [12], [14], [15]. Emulab [8] is well-known for its scalability and Linux Container virtualization (LXC). The design principles in Emulab have a significant influence on the following design of emulation platforms. However, with advances in technology, Emulab techniques proposed in 2008 would be outdated. PlanetLab [9] is a global overlay network for developing and accessing broad-coverage network services. PlanetLab uses the slice technique to run multiple services concurrently and continuously. This slice technique is similar to network slice in 5G, which can be viewed as network virtualization technology. Currently, this technique has been rapidly developed and fully utilized in cloud computing and NFV. In parallel to network virtualization technologies, compute virtualization technologies have been developed and tend to mature, such as LXC, Linux-Vserver, and Docker [11]. In particular, LXC is used as OS-level virtualization technology in Mininet [12]. Mininet is a lightweight emulation platform which is designed specifically for soft defined networking (SDN) [13]. Mininet enables a laptop to produce hundreds of emulating nodes by LXC. Handigol [14] improves Mininet with resource provisioning, isolation and monitoring system, and enormous typical experiments have been reproduced to prove that Mininet has good fidelity.

Besides, we [15] proposed an OpenStack-based Delay Tolerant Networking (DTN) Emulation Platform (EmuStack). EmuStack integrates OpenStack with Linux Traffic Control (TC) tools for managing and emulating the virtual link characteristics that include variable bandwidth, delay, loss, jitter, re-ordering and duplication. Using Docker container technology and network namespaces, EmuStack can support a large-scale topology (including hundreds of nodes) with only several physical nodes.

3 Emulator System Design

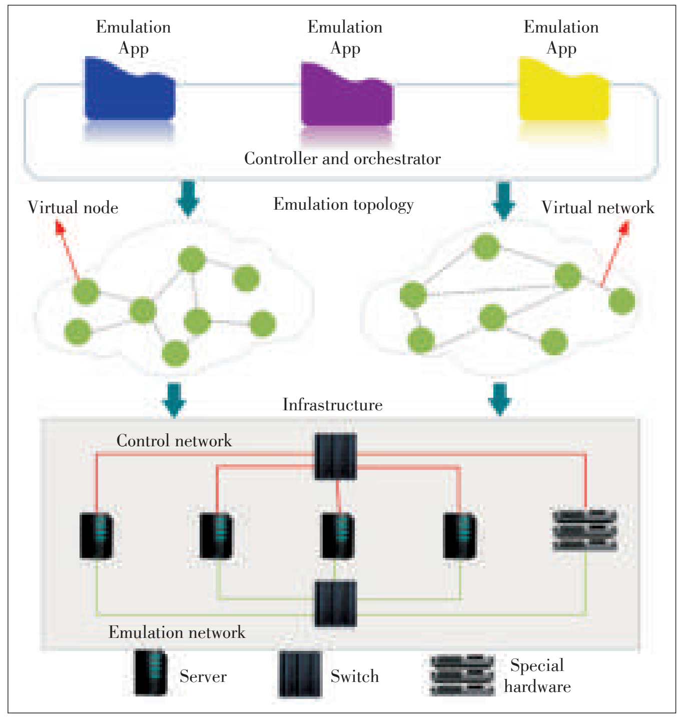
This section describes the overall architecture of NFV-based emulation platform for SINET, and discusses emulation control and orchestration, which are critical for the SINET emulation platform.

3.1 Design Overview

Fig. 2 presents the overall platform design structure. NFV is employed to manage the hardware infrastructure, which is composed of standard X86 servers, high-performance switches and special hardware. In particular, the special hardware is used to accurately emulate complex and professional network environments, such as deep space link and high-speed traffic. It is uniformly described, and can work with X86 servers to carry three types of nodes, namely, emulation nodes, controller nodes and orchestrator nodes.

Emulation nodes consist of physical emulation nodes and virtual emulation nodes. The physical emulation node is special hardware or a standard x86 server which can be virtualized massively to virtual emulation nodes. Virtual emulation nodes can be created by different computing virtualization technologies, such as Docker, Xen and Kernel-based Virtual Machine (KVM). A hybrid deployment is allowed where different computing virtualization technologies can be employed concurrently with multiple physical emulation nodes.

Controller and orchestrator nodes run on standard X86 servers. In the SINET emulation platform, there are at least one controller and orchestrator, which are responsible for managing SINET emulation, monitoring and collecting experimental results. In details, the controller is in charge of creating and managing compute and network resources. On the basis of users' configuration, the controller invokes the virtualized infrastructure manager (VIM) to instantiate virtual emulation nodes and network. After starting an emulation, the controller monitors and collects emulating data. As to the orchestrator, it supervises usage of hardware resources and dynamically orchestrates the whole hardware resources including CPU, memory and network, hence, achieving emulation overload avoidance and green computing.



▲Figure 2. The overall design structure of emulator.

Emulation nodes, controller nodes and orchestrators are equipped with multiple Network Interface Cards (NICs), which are connected with each other by switches to construct the physical network. The physical network is divided into a control network and an emulation network, as shown in the bottom part of Fig. 2. This design enables the network to become directly programmable and dynamically configurable. The control network carries control traffic that consists of network and emulation management information. The emulation network transfers emulation data. Emulation data varies greatly with different SINET experiments, which probably consumes enormous bandwidth of the emulation network. In practice, physical emulation network is the primary limit in the SINET emulation platform. Therefore, we equip each physical emulation node with multiple emulation NICs to carry the large-amount emulation traffic. These NICs are bridged to an Open vSwitch

(OVS), whose internal device is assigned an emulation network IP address. With these bridging technique, we extend bandwidth of emulation network to n times than one NIC, where n is the number of emulation NICs.

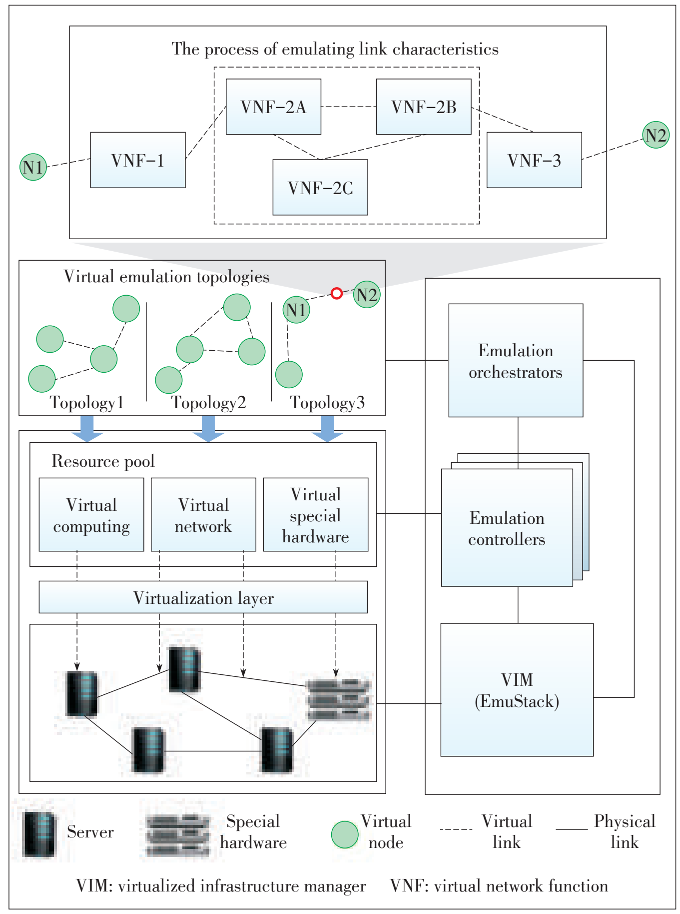
The middle part of Fig. 2 presents the solution to generating virtual emulation topologies in the SINET emulation platform. Virtual emulation topology is composed of virtual emulation nodes and network. Virtual emulation nodes are spawned by the orchestrator with the custom operating system images equipped with SINET experimental protocol software. The virtual emulation network is overlay network built on top of the physical emulation network. SDN and encapsulation techniques are employed to virtualize physical emulation network. The SDN has multiple kinds of network technologies designed to make the network more flexible and agile. The SINET emulation platform uses (Open vSwitch) OVS as SDN switch, and integrates the SDN controller into the emulation controller, hence, achieving the ability to dynamically program and configure virtual emulation network. Additionally, encapsulation techniques consist of Generic Network Virtualization Encapsulation (GNVE) protocol and virtual extensible LAN (VXLAN). These techniques are employed to isolate the user emulation traffic from other users, and accordingly offer each user an oblivious context by which different users can create respective virtual emulation network on the same underlying infrastructure and do not interfere with each other.

Virtual emulation topology is managed by user control emulation applications (Apps). According to his emulating purpose and scenario, a user writes an APP to dynamically control virtual emulation nodes and network. In order to further evaluate SINET performance, the emulation platform offers abundant link characteristic emulations, including link bandwidth, latency, jitter, packet loss, duplication and re-ordering. An App invokes EmuStack [15] to control dynamically these link characteristics. By setting Hierarchical Token Bucket of EmuStack within the network namespace, a controller App can dynamically and independently manage every virtual link characteristic for each virtual emulation topology.

3.2 Emulation Control and Orchestration

Fig. 3 shows emulation control and orchestration. In the SINET emulation platform, the entire hardware resource is virtualized into the resource pool. The controller and orchestrators are employed to control and orchestrate this resource pool.

The emulation controllers are primarily responsible for dynamically and synchronously controlling virtual emulation topology instantiated from the resource pool. For the large-scale SINET emulation platform, it is a challenge. First, there are hundreds of nodes that have stochastic communication delay and background system load, which makes it difficult and complex to control them. Second, different from the simulator (e.g., ns-3) based on discrete event and run in virtual time, the SINET emulation platform runs in real time. Therefore, the controller cannot pause a emulation node' clock to wait for other pending events, which further raises the problems such as synchronization in control.



▲**Figure 3. The diagram of emulation control and orchestration.**

In order to overcome these problems, we employ a distributed pre-allocating control algorithm. In detail, each physical emulation node (compute node) runs as a local control agent, which is responsible for controlling the emulating topologies that are instantiated locally. Before starting a emulation, the controller dispatches control information to the local control agent in advance. Control information primarily consists of management commands of SINET experimental protocols, local link characteristics and topological state of different period. Especially, a solution time stamp is sent with control information, which is used as the experimental start time for each local control agent.

The local control agent employs the sleep and busy waits algorithm to exactly and effectively execute emulating events. When there are amounts of time remained before starting experiment, local control agent goes on sleep-waits mode. Sleep-waits cause local control agent to yield the processor for some amount of time. This specified amount of time is actually converted to an operating system specific granularity, even though it can be passed to nanosecond resolution. For the SINET emulation platform, every X86 server run Linux. In Linux, the granularity is namely Jiffy. Typically, this resolution is insufficient for our needs (generally, ten milliseconds), so we round down and sleep for some smaller number of Jiffies. After hardware real time is close the solution time stamp of experimental start, local control agent is in busy-waits mode and awakened. At this time, we have some residual time to wait. This time is generally smaller than the minimum sleep time, so we busily wait for the remainder of the time. This means that the thread of local control agent just sits in a loop, and consumes cycles until the experimental time arrives.

For all events in emulation scheduling queue, we all adopt the sleep and busy waits algorithm to wait for the time of emulation events. After the previous event combination of sleep and busy waits, the elapsed real-time (wall) clock should agree with the emulation time of the next event, and then the emulation proceeds.

Orchestrators are primarily responsible for dynamically orchestrating hardware resource to ensure there are adequate CPU, memory, network and special hardware resource available to provide users emulation. For the large-scale SINET emulation platform, dynamically large-scale orchestration is also a challenge. First, the resource needs of virtual emulation nodes are heterogeneous due to the diverse SINET experiments, and vary with time as the workloads shrink and grow. It is difficult to forecast the utilization of resource pool when multiple users emulate experimental protocols concurrently. Second, the capacity of physical emulation nodes (standard x86 servers) can also be heterogeneous since multiple generations of servers co-exist in the emulation platform. Thus this heterogeneity of hardware further increases complexity of orchestrating resource.

To solve aforementioned orchestrating problems, we apply Xiao's dynamic resource allocation algorithm to the SINET emulation platform to orchestrate emulation resource in [16]. We aim at effectively avoiding overload but minimizing the number of servers used. In practice, there is an inherent tradeoff between these two goals. To avoid overload, we should keep the utilization of physical emulation node low to reduce the possibility of overload in case the resource needs of virtual emulation nodes increase later. To minimize amounts of used servers, we should keep the utilization of physical emulation nodes reasonably high to utilize efficiently their energy. Fortunately, Xiao's resource allocation algorithm nearly optimally tackled this tradeoff. We apply it to the SINET emulation platform, and adopt Xiao's skewness concept to measure the uneven usage of a server. By minimizing skewness, we can enhance the overall utilization of server. Meanwhile, the load prediction algorithm in [16] is employed to achieve the future resource usages. It helps to reduce the placement churn significantly.

Besides, orchestrators can work either with controller or Virtualized Infrastructure Manager (VIM), and provide management of virtual network functions (VNFs) to emulate more link characteristics. VNFs run on industry-standard X86-based servers like virtual emulation nodes, which provides a well-defined functional behavior, such as forwarding, routing and firewall. Typically, emulation link characteristics including delay, error ratio and link rate are emulated by controller with EmuStack. However, there are still several link characteristics which cannot be realized by EmuStack. For example, it cannot emulate more than 300 s link delay with EmuStack. However, if we emulate SINET for deep space communication, 300 s delay would not meet emulating requirement. To emulate this longer link delay, a VNFs chaining would be orchestrated to endow a virtual link with the longer delay. The VNFs chaining is generally composed of multiple VNFs as shown in the upper part of Fig. 3.

4 Conclusion

In this paper, we present a NFV-based emulation platform for Smart Identifier Network (SINET). We first introduce SINET architecture, and then outline the design of the SINET emulation platform. We design controllers and orchestrators for the SINET emulation platform. With the controllers running a distributed sleep and busy waits algorithm, we achieve dynamical and exactly synchronous emulation management for large-scale SINET emulation. In order to achieve overload avoidance and green computing, we apply the Xiao's dynamical resource allocation algorithms to Orchestrators. Thus, we ensure there are always ample CPU, memory and network resources to provide SINET emulations for different users, while minimizing the number of underutilized servers.

References

[1] G. Xylomenos, C. N. Ververidis, V. A. Siris, et al., "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014. doi: 10.1109/SURV.2013.070813.00063.

[2] H. Luo, Z. Chen, J. Cui, et al., "CoLoR: an information-centric internet architecture for innovations," *IEEE Network*, vol. 28, no. 3, pp. 4–10, 2014. doi: 10.1109/MNET.2014.6843226.

[3] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, Jul. 2012. doi: 10.1109/MCOM.2012.6231276.

[4] L. Zhang, A. Afanasyev, J. Burke, et al., "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, Jul. 2014. doi: 10.1145/2656877.2656887.

[5] H. Zhang, W. Quan, H. Chao, and C. Qiao, "Smart identifier network: a collaborative architecture for the future internet," *IEEE Network*, vol. 30, no. 3, pp. 46–51, 2016. doi: 10.1109/MNET.2016.7474343.

[6] H. Zhang, P. Dong, W. Quan, and B. Hu, "Promoting efficient communications for high-speed railway using smart collaborative networking," *IEEE Wireless Communications*, vol. 22, no. 6, pp. 92–97, Dec. 2015. doi: 10.1109/MWC.2015.7368829.

[7] H. Zhang, H. Luo, and H. Chao, "Dealing with mobility-caused outdated mappings in networks with identifier/locator separation," *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 2, pp. 199–213, 2015. doi: 10.1109/

TETC.2015.2449664.

[8] M. Hibler, R. Ricci, L. Stoller, et al., "Large-scale virtualization in the emulab network testbed," in *USENIX Annual Technical Conference*, Boston, USA, 2008, pp. 113–128.

[9] B. Chun, D. Culler, T. Roscoe, et al., "Planetlab: an overlay testbed for broad-coverage services," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 3, pp. 3–12, Jul. 2003. doi: 10.1145/956993.956995.

[10] H. ZHAO, Y. XIE, and F. Shi, "Network function virtualization technology: progress and standardization," *ZTE Communications*, vol. 12, no. 2, pp. 3–7, Jun. 2014. doi: 10.3969/j.issn.1673-5188.2014.02.001.

[11] M. G. Xavier, M. V. Neves, F. D. Rossi, et al., "Performance evaluation of container-based virtualization for high performance computing environments," in *21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, Belfast, UK, 2013, pp. 233–240. doi: 10.1109/PDP.2013.41.

[12] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Monterey, USA, 2010, article No. 19. doi: 10.1145/1868447.1868466.

[13] Z. Sun, J. Li, and K. Yang, "Software-defined networking," *ZTE Communications*, vol. 12, no. 2, pp. 1–2, 2014.

[14] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown, "Reproducible network experiments using container-based emulation," in *ACM Proceedings of the 8th international conference on Emerging networking experiments and technologies*, 2012. doi: 10.1145/2413176.2413206.

[15] H. Li, H. Zhou, H. Zhang, and W. Shi, "EmuStack: an OpenStack-based DTN network emulation platform (extended version)," *Mobile Information Systems*, vol. 2016, article ID 6540207, 2016. doi: 10.1155/2016/6540207.

[16] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1107–1117, Jun. 2013. doi: 10.1109/TPDS.2012.283.

Manuscript received: 2017-05-26

Biographies

LI Haifeng (haifengli@bjtu.edu.cn) received the B.E. degree in communication and information systems from the Beijing Jiaotong University, Beijing, China, in 2013. He is currently working toward his Ph.D. degree at the School of Electronic and Information Engineering, Beijing Jiaotong University. He has participated in several national research programs of China such as "973" Program and "863" Program. His research interests include future Internet, Delay Tolerant Networking, High Performance Computing and Cloud Computing.

LI Taixin (14111040@bjtu.edu.cn) received the B.S. degree in telecommunications engineering from Beijing Jiaotong University in 2013, and then entered National Engineering Lab for Next Generation Internet Interconnection Devices at BJTU. Currently, he is pursuing the Ph.D. degree in telecommunications and information system at BJTU, China. His research interests include next generation internet, network services and satellite network.

ZHANG Hongke (hkzhang@bjtu.edu.cn) received the M.S. and Ph.D. degrees in electrical and communication systems from the University of Electronic Science and Technology of China (formerly known as Chengdu Institute of Radio Engineering) in 1988 and 1992, respectively. From September 1992 to June 1994, he was a Postdoctoral Research Associate at Beijing Jiaotong University (formerly known as Northern Jiaotong University). In July 1994, he joined Beijing Jiaotong University, where he is a Professor. He currently directs a National Engineering Lab on Next Generation Internet in China. He is a Senior member of IEEE, and has published more than 100 research papers in the areas of communications, computer networks, and information theory. He is the author of eight books written in Chinese and the holder of more than 30 patents. Dr. Zhang received the Zhan Tianyou Science and Technology Improvement Award in 2001, the Mao Yisheng Science and Technology Improvement Award in 2003, the First Class Science and Technology Improvement Award of the Beijing government in 2005, and other various awards. He is now the Chief Scientist of a National Basic Research Program ("973" Program).

DOI: 10.3969/j.issn.1673-5188.2017.03.002
http://kns.cnki.net/kcms/detail/34.1294.TN.20170829.1546.002.html, published online August 29, 2017

Survey of Mechanisms for Inter-Domain SDN

WANG Yangyang [1,3] **and BI Jun** [1,2,3]
(1. Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China;
2. Department of Computer Science, Tsinghua University, Beijing 100084, China;
3. Tsinghua National Laboratory for Information Science and Technology (TNList), Tsinghua University, Beijing 100084, China)

Abstract

Software defined networking (SDN) has been applied increasingly in practical networks. Currently, SDN is mainly used to improve the flexibility and efficiency of datacenter networks, enterprise networks and wide-area networks (WAN). There also emerge some studies that try to deploy SDN to inter-domain settings. In this article, we introduce the progress stages of inter-domain SDN and studies related to each stage. Finally, we discuss the applications and challenges of inter-domain SDN.

Keywords

software defined networking; network programmability; inter-domain SDN

1 Introduction

In recent years, the development of software defined networking (SDN) provides an opportunity for networking innovations. SDN decouples the control plane and data plane of networks. The data plane provides common programmable resources and interfaces of network devices, which makes it possible to program network centrally and greatly simplifies and improves network management and control. Many studies of SDN focus on taking logically centralized control plane in single domain in campus networks, enterprise networks, data centers, and private wide - area networks (WANs). If we extend SDN to inter-domain settings across multiple domains, we can take advantage of its opportunity of programmability and innovation in the Internet. The flexibility of SDN control looks forward to making it possible to optimize scheduling for inter - domain networking resources. However, the inter-domain setting is a scenario of distributed administration. There are more than 500,000 autonomous systems (ASes) in the global Internet. The growth of the number of ASes indicates a super-linear tread based on the statistics from global BGP routing table. Therefore, SDN will meet the scalability issue when applying SDN to the environment of distributed control in inter-domain settings. In a word, there are both challenges and opportunities in the inter-domain SDN researches.

This work was supported in part by the National Key R&D Program of China under Grant No. 2017YFB0801701 and in part by the National Science Foundation of China under Grant No. 61472213.

The "SDN domain" in this article generally refers to the SDN control domain, which is a network domain deployed SDN mechanism and administrated by operators to control independently. An SDN domain can be an AS of the Internet, or a network control domain composed of a number of ASes, and may even be a SDN deployment domain with no AS number in the future. The inter-domain SDN mechanisms studied in this paper are around how to solve the problem of inter-domain control cooperation between SDN domains in the global Internet scale, mainly referring to inter-domain SDN applied to the autonomous system level. If this problem cannot be solved, the SDN mechanism can only be used in a single control domain or local scale networks (such as enterprise networks and data networks), but cannot provide programmability for routing and control applications across multiple domains.

We introduce inter-domain SDN according to the areas of its deployment.

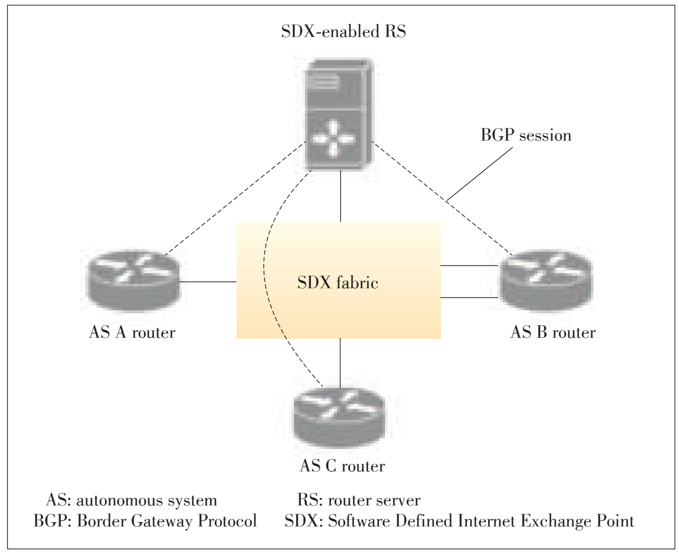
2 Small SDN Networks Interconnected with Border Gateway Protocol (BGP) Domains

Deploying a new SDN network into an autonomous domain is the most common approach. However, how this newly deployed SDN network is interoperable with other BGP network domains is an important issue. The BTSDN mechanism [1] is an example of how to solve this problem. The principle of BTSDN is to deploy a new SDN area in an AS instead of replacing the existing BGP boundary router. On the control plane, the controller of the SDN region exchanges routing information with the BGP routers located at the AS boundary by running the APP of the internal BGP (iBGP), and then connects with other ASes via external BGP (eBGP) sessions. The data plane adopts the mechanisms of Address Resolution Protocol (ARP) Proxy and media access control (MAC) address Rewrite to ensure the delivery of IP packets between the BGP border routers and the SDN region. The deployment cost of BTSDN is very small, which does not need to replace the existing BGP routers and only requires to deploy a new SDN area in an AS. It realizes control function on the controller, and complete the necessary control on the data plane.

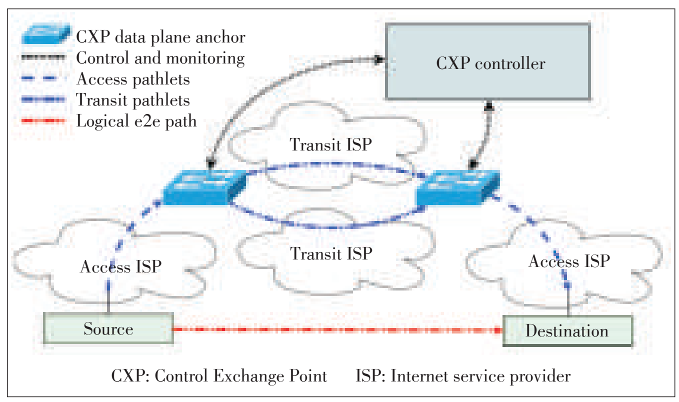
3 Interconnection Between SDN Domains and BGP Domains

There are many representative technologies to interconnect SDN domains with BGP domains, including RouteFlow [2], SDN-IP [3], and Software Defined Internet Exchange Points (SDXs) [4]. SDN devices are adopted to realize this interconnection. Aiming at various targets, these solutions have their own suited scenarios. RouteFlow [2] focuses on providing services of virtual routers. By replacing a commercial router with an OpenFlow switch that can be remotely programmed to control, the control logic function of BGP routers is moved to the virtual router machine, each OpenFlow switch corresponding to a virtual machine. The SDN-IP technology in [3] focuses on the seamless peering interconnection between SDN control domains and traditional BGP routing domains based on BGP, and the realization of SDN control domains as transit networks of the traditional BGP routing domains, in order to promote the gradual deployment of SDN in the existing networks to replace the traditional BGP ASes. The controller of an SDN control domain still uses BGP protocol to exchange routing information with the neighboring traditional BGP routing domains, but uses SDN centralized mode to control local AS's BGP routing calculation and installation. It is beneficial to the efficient control of BGP routing in a local AS.

Princeton University proposed SDX, which mainly transforms the traditional Internet Exchange Point (IXP) and route server (RS) based on SDN, to achieve flexible policies of interconnection between multiple ASes through SDX (**Fig. 1**). The SDX switching infrastructure uses OpenFlow switches to provide flexible, fine-grained inter-domain traffic switching policies for two or more member ASes connected to a SDX. Compared with the traditional IXPs based on RS, the management efficiency and functional flexibility are improved. The limitation of SDX is that SDN control services can only be provided for inter-domain traffic between member ASes. The Swiss Federal Institute of Technology Zurich in Europe proposed the Control Exchange Points (CXPs) [5], as shown in **Fig. 2**. Multiple IXPs stitch multiple segments of inter-domain routing paths under SDN centralized control to form a cross-domain end-to-end path that meets the performance requirements of



▲**Figure 1. The architecture of SDX [4].**



▲**Figure 2. The CXP controller controls multiple IXPs to form end-to-end QoS paths [5].**

Quality of Service (QoS).

The study in [6] tried to build a larger SDN control domain, using a super SDN controller to implement centralized control and management of the ASes that belong to the same organizational structure and are distributed around the world, in order to improve the convergence efficiency of the inter-domain routing among these ASes. The characteristics of these researches are using SDN centralized management and control mode to improve the function and performance of the current BGP routing. These solutions make it easier to find inter-domain SDN applications and requirements in the current networks. However, they are limited by the traditional network compatible scenarios and cannot fully take advantages of SDN for fine-grained and flexible control of traffic forwarding.

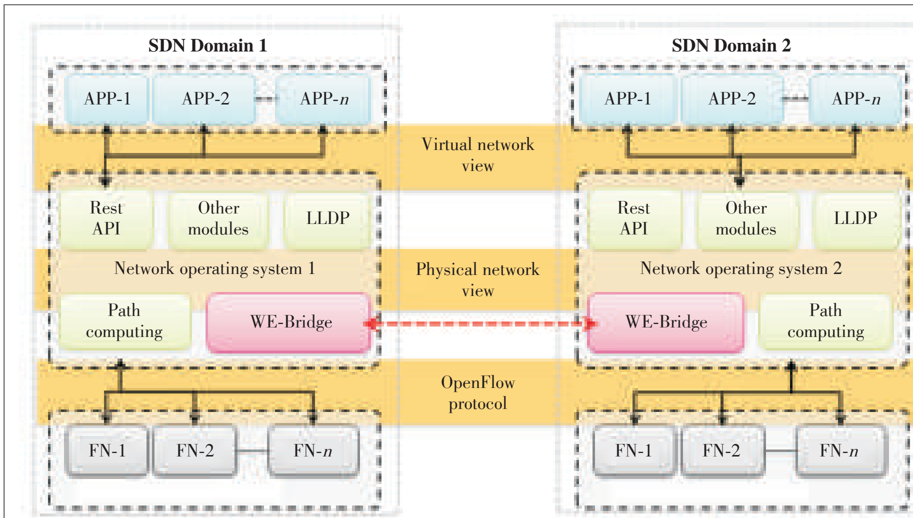
4 Interconnection and Cooperation Between SDN Domains

The interconnection and cooperation between SDN domains are more forward-looking and challenging. In RFC draft [7], BGP was proposed to transfer IP routing reachability messages between federated SDN controllers in data centers. Literatures [8]–[10] propose the hierarchical structure of the controllers in multiple SDN control domains. However, the control plane interconnected by these multi-domain SDN networks is still under centralized management and control, and is not applicable to the distributed and autonomous environment of ASes on the Internet. The literature [11] proposes using NOX controllers and OpenFlow switches to implement a BGP-like distributed inter-domain routing protocol for multiple SDN domains. In some literatures [12], [13], multiple SDN domains exchange information to provide end-to-end cross-domain path service satisfying QoS performance. These proposals are appropriate for the scenario of federated SDN domains, and are not appropriate for large-scale Internet-wide environments. In [14], Extensible Session Protocol (XSP) is proposed. XSP is a high-level session layer protocol located on the transport layer, which is used as the interaction interface between network applications and network services. SDN focuses on the design of a high-level session layer protocol itself that is only for specific application services, but does not address the general underlying interconnection mechanism between domains. In addition, there are a number of projects that try to provide Network Service Interface (NSI) for the upper network applications to realize SDN virtual resource sharing among multiple SDN domains, such as Japan and the EU co-funded project FP7 FELIX [15]. Its main purpose is the high-level inter-domain resource sharing. Although the above researches have their application scenarios and values, they cannot provide a universal inter-domain SDN interconnection mechanism.

The goal of interconnection of SDN domains is to provide general mechanism for global Internet-scale interconnection and cooperation of SDN domains. This kind of research is still in the initial stage. A project team of Tsinghua University has conducted this research earlier in the world, and proposed a new cooperative inter-domain SDN mechanism, West-East Bridge (WE-Bridge) [16].

WE-Bridge mechanism is a new type of SDN East-West interconnection mechanism proposed by Tsinghua University based on the 12th Five-Year "863" national project "Future Network Architecture and Innovation Environment". **Fig. 3** shows the position of the WE-Bridge in the proposed Future Internet Innovation Environment (FINE) architecture in this "863" project. WE-Bridge is an extension of the network operating system layer and virtual platform layer of FINE. The SDN controller exchanges the information of virtual network view of the controlled domain based on WE-Bridge, which provides a collaborative interface for a variety of inter-domain applications (such as new inter-domain routing protocols and path computation), and realizes the cross-domain collaboration for SDN applications. The East-West interface protocol for cross-domain cooperation of WE-Bridge includes a mechanism of establishing peering connection between SDN domains, optimized distribution mechanism of the exchanged information (routing strategy of each SDN domain, virtual network views) among SDN domains, message format and negotiation process. Based on WE-Bridge, a new type of fine-grained inter-domain routing application is proposed to verify the effectiveness of the proposed mechanisms. The proposal of WE-bridge has begun a preliminary attempt for large-scale SDN inter-domain interconnection.

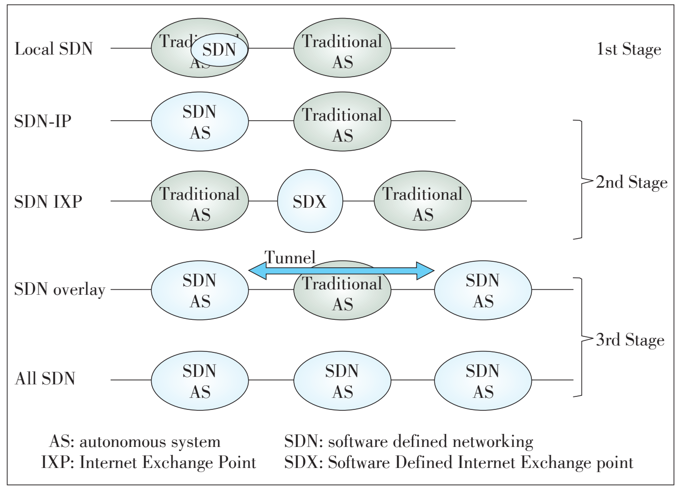
In 2012, the Chinese-American Network Symposium (CANS) established the future Internet/SDN working group, chaired by the Interent2 CTO Dr. Stephen Wolff and Professor BI Jun of Tsinghua University. The main content of this working group charter is to carry out the research of an inter-domain SDN testbed, and application innovations on the inter-domain SDN testbed. The working group has concluded that the inter-domain SDN testbed was unable to adopt a centralized structure, and therefore adopted the WE-Bridge mechanism proposed by Tsinghua University. At the Interent2 and APAN joint conference in Hawaii in January 2013, the working group discussed and determined the technical solution of Tsinghua University. This meeting was held at the East West Center of University of Hawaii System. Inspired by this, this mechanism is named WE-Bridge. On the basis of the design and implementation of WE-Bridge, Chinese Education and Research Computer Network (CERNET), the United States Internet2,



API: Application Programming Interface
APP: application
FN: forwarding node
LLDP: Link Layer Discovery Protocol
SDN: software defined networking
WE-Bridge: West-East Bridge

▲**Figure 3. Inter-domain SDN mechanism of WE-Bridge.**

China Science and Technology Network (CSTNET), Holland academic network SURFnet, the Japanese academic network (APAN-JP/JGN-X) cooperated and established the first cooperative and international inter-domain SDN test network. The graph user interface shows the topology of this testbed in **Fig. 4**. This inter-domain SDN testbed had been demonstrated at such conferences as CANS from 2013 to 2015, the International Conference on SuperComputing 2013 and 2014, APAN 2014 and 2015, the global Open Network Summit (ONS) 2014. The test network was also accepted as the IEEE INFOCOM 2014 demo [18]. Dr. Stephen Wolff published the technical evaluation [19], [20] at the Internet2 website He believes that the current SDN mechanism can only be used in single domain environment, and this pioneering work on inter-domain SDN demonstrates that SDN mechanism can be extended to multi-domain environment in key global scientific cooperation.



▲Figure 5. Deployment process of inter-domain SDN.

5 Deployment Stages of Inter-Domain SDN

SDN network and traditional IP network have a long coexistence transition period. We believe that inter-domain SDN deployment has the processing stages shown in **Fig. 5**.

The first stage is a new SDN network deployed locally in an autonomous system, which is connected with the domains controlled by traditional BGP. The second stage is to use SDN mechanism to control the whole autonomous system, and to connect with the traditional BGP domain. For example, an AS or IXP uses SDN mechanism to control network traffic forwarding and routing policies. The first and second stages mainly focus on how to implement the interworking between a new SDN domain and a traditional BGP domain.

When SDN has been deployed in many ASes, the third stage forwards further to study how to cooperate control between SDN domains, in order to give full play of SDN programming in the inter-domain routing control. The two ASes may be physically adjacent or across multiple traditional IP networks, and control session negotiation and data traffic forwarding are established by overlay scheme. The third stage faces more opportunities and challenges than the previous stages. This stage enables collaboration between many SDN enabled domains, with fine-grained programmable control over Internet routing in a wider range. This stage will help contribute to the new network business model. Internet service providers can provide users with flexible and efficient network channels, improve the performance of the Internet, security and control efficiency, and provide a possible opportunity for the rapid deployment of new network protocols on the Internet. However, achieving this target needs to overcome some challenges such as scalability of SDN applications in large-scale Internet.

APAN-JP: Asia-Pacific Advanced Network Japan CSTNET: China Science and Technology Network

▲Figure 4. The graph user interface of WE-Bridge inter-domain testbed.

6 Applications and Challenges of Inter-Domain SDN

Through the introduction of some typical studies, we summarize the prospects and problems of inter-domain SDN research. The mechanism of inter-domain SDN can bring applications as follows: (1) Inter-domain SDN can provide rich and flexible inter-domain traffic control capabilities, such as the applications in inter-domain traffic engineering and distributed denial-of-service (DDoS) attack defense; (2) inter-domain SDN extends SDN to larger-scale Internet to improve operational efficiency, overall networking programmability, and new technology deployment; (3) inter-domain SDN can provide new business models for operators and users, such as meeting the requirements of QoS-aware end-to-end path service.

In the meantime, inter-domain SDN mechanisms are faced with the following challenges:

1) Scalability problems. A large number of network views will lead to poor scalability of the control plane. The large number of routing entries from numerous ASes also bring burden to both memory and lookup efficiency of the forwarding engine of the SDN data plane. How to avoid or solve these scalability problems is a challenge for constructing large inter-domain SDN network.

2) Increasing deployment and incentives. Increasing deployment of the SDN mechanisms among ASes should be critical for operators to be compatible with traditional networks.
3) Security problems. The deployment of new technologies inevitably results in security problems. How the inter-domain SDN mechanism can reduce its own security problems and improve the security control of the Internet is a challenge.

7 Conclusions

In recent years, SDN has received great attention from both academia and industry, and has made considerable progress in the field of centralized SDN control and application. However, how to extend the SDN mechanism to the inter domain to support new inter-domain SDN applications is an opportunity and challenge.

This article introduces the researches of typical inter-domain mechanisms, including the WE-Bridge, an inter-domain SDN interconnection mechanism proposed by Tsinghua University. This article also discuss the deployment stages of inter-domain SDN, and the challenges that need to be resolved in future researches.

With the gradual deployment of SDN domains, inter-domain SDN interconnection will gradually become popular. We hope this article can be of some reference value in this field.

References

[1] P. P. Lin, J. Bi, and H. Y. Hu, "BTSDN: BGP-based transition for the existing networks to SDN, " *Wireless Personal Communications*, vol. 86, no. 4, pp. 1829–1843, Feb. 2016. doi:10.1007/s11277-015-3145-0.
[2] M. R. Nascimento, C. E. Rothenberg, M. R. Salvador, et al., "Virtual routers as a service: the routeflow approach leveraging software-defined networks," in *Proc. 6th International Conference on Future Internet Technologies*, New York, USA, Jun. 2011, pp. 34–37. doi: 10.1145/2002396.2002405.
[3] P. Lin, J. Hart, M. Kobayashi, et al., "Seamless interworking of SDN and IP," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 475–476, Oct. 2013. doi: 10.1145/2534169.2491703.
[4] A. Gupta, L. Vanbever, M. Shahbaz, et al., "SDX: a software defined internet exchange," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 551–562, Oct. 2014. doi: 10.1145/2740070.2626300.
[5] V. Kotronis, M. Rost, P. Georgopoulos, et al., "Stitching Inter-Domain Paths over IXPs," in *Proc. Symposium on SDN Research*, Santa Clara, USA, Mar. 2016, Article No. 17. doi: 10.1145/2890955.2890960.
[6] H. Kazmi, H. Nawaz, M. A. Gulzar, et al. (2015, Oct. 25). *BGP is High on SDN: Improving BGP Convergence and Security using SDN and Reassertions* [Online]. Available: http://web.cs.ucla.edu/~gulzar/pdf/bgp.pdf
[7] F. Balus, D. Stiliadis, N. Bitar, and K. Ogaki. (2013, Nov.). *Federated SDN-based Controllers for NVO3 (Network Virtualization Overlays)*, draft-sb-nvo3-sdn-federation-02 [Online]. Available: https://tools.ietf.org/html/draft-sb-nvo3-sdn-federation-02
[8] J. McCauley, A. Panda, M. Casado, et al., "Extending SDN to large-scale networks," in *Open Networking Summit*, Santa Clara, USA, 2013, pp. 1–2.
[9] Y. Fu, J. Bi, Z. Chen, et al., "A hybrid hierarchical control plane for flow-based large-scale software-defined networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 2, pp. 117–131, May 2015. doi: 10.1109/TNSM.2015.2434612.
[10] M. Karakus and A. Durresi, "A scalable inter-AS QOS routing architecture in software defined network (SDN)," in *IEEE 29th International Conference on Advanced Information Networking and Applications (AINA)*, Gwangju, Korea, 2015, pp. 148–154. doi: 10.1109/AINA.2015.179.
[11] R. Bennesby, P. Fonseca, E. Mota, et al., "An Inter-AS routing component for software-defined networks," in *IEEE Network Operations and Management Symposium (NOMS)*, Maui, USA, 2012, pp. 138–145. doi: 10.1109/NOMS.2012. 6211892.
[12] K. Phemius, M. Bouet, and J. Leguay, "Disco: distributed multi-domain SDN controllers," in *IEEE Network Operations and Management Symposium (NOMS)*, Krakow, Poland, 2014, pp. 1–4. doi: 10.1109/NOMS.2014.6838330.
[13] S. Civanlar, E. Lokman, B. Kaytaz, et al., "Distributed management of service-enabled flow-paths across multiple SDN domains," in *European Conference on Networks and Communications (EuCNC)*, Paris, France, 2015, pp. 360–364. doi: 10.1109/EuCNC.2015.7194099.
[14] E. Kissel, G. Fernandes, M. Jaffee, et al., "Driving software defined networks with XSP," in *2012 IEEE International Conference on Communications(ICC)*, Ottawa, Canada, 2012, pp. 6616–6621. doi: 10.1109/ICC.2012.6364805.
[15] FELIX Project [Online]. Available: http://www.ict-felix.eu
[16] P. Lin, J. Bi, and Y. Wang, "East-west bridge for SDN network peering," in *Frontiers in Internet Technologies*, J. Su, B. Zhao, Z. Sun, X. Wang, F. Wang, and K. Xu, eds. Berlin/Heidelberg, Germany: Springer, 2013, pp. 170–181. doi: 10.1007/978-3-642-53959-6_16 vol 401.
[17] P. Lin, J. Bi, S. Wolff, et al., "A west-east bridge based SDN inter-domain testbed," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 190–197, 2015. doi: 10.1109/MCOM.2015.7045408.
[18] P. Lin, J. Bi, Z. Chen, et al., "WE-bridge: west-east bridge for SDN inter-domain network peering," in *33rd IEEE International Conference on Computer Communications (INFOCOM14)* Workshops, Toronto, Canada, 2014, pp. 111–112. doi: 10.1109/INFCOMW.2014.6849180.
[19] S. Wolff. (2013, Sep. 20). *Inter-Domain, Software-Defined Networking Testbed and Application Demonstrated By Chinese and American Network Engineers* [Online]. Available: https://www.internet2.edu/blogs/detail/5020
[20] E. Moynihan. (2014, Oct. 14). *Groundbreaking Demonstration Sets Stage for New Interdomain SDN Applications and Research Collaborations* [Online]. Available: https://www.internet2.edu/blogs/detail/7174

Manuscript received: 2017-05-31

Biographies

WANG Yangyang (wangyy@cernet.edu.cn) received his B.S. degree in computer science and technology from Shandong University, China in 2002, M.S. degree from Capital Normal University, China in 2005, and Ph.D. degree from the Department of Computer Science of Tsinghua University, China in 2013. He is currently a postdoctoral scholar in computer science at Tsinghua University. His research interests include Internet routing architecture, future Internet design, and SDN.

BI Jun (junbi@tsinghua.edu.cn) received his B.S., C.S., and Ph.D. degrees from the Department of Computer Science, Tsinghua University, China. He is currently a Changjiang Scholar Distinguished Professor and the Director of Network Architecture Research Division, Institute for Network Sciences and Cyberspace, Tsinghua University. His current research interests include Internet architecture, SDN/NFV, and network security. He successfully led tens of research projects, published over 200 research papers and 20 Internet RFCs and drafts, and also holds 30 innovation patents. He received the National Science and Technology Advancement Prizes, the IEEE ICCCN Outstanding Leadership Award, and Best Paper awards. He is the co-chair of the AsiaFI Steering Group and the Chair of the China SDN Experts Committee. He served as the TPC co-chair of a number of Future Internet related conferences or workshops/tracks at INFOCOM and ICNP. He served on the Organization Committee or Technical Program Committees of SIGCOMM, and ICNP, INFOCOM, CoNEXT, and SOSR. He is Distinguished Member of the China Computer Federation.

DOI: 10.3969/j.issn.1673-5188.2017.03.003
http://kns.cnki.net/kcms/detail/34.1294.TN.20170912.1054.002.html, published online September 12, 2017

DDoS Attack in Software Defined Networks: A Survey

XU Xiaoqiong, YU Hongfang, and YANG Kun
(School of Communication & Information Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract

Distributed Denial of Service (DDoS) attacks have been one of the most destructive threats to Internet security. By decoupling the network control and data plane, software defined networking (SDN) offers a flexible network management paradigm to solve DDoS attack in traditional networks. However, the centralized nature of SDN is also a potential vulnerability for DDoS attack. In this paper, we first provide some SDN-supported mechanisms against DDoS attack in traditional networks. A systematic review of various SDN-self DDoS threats are then presented as well as the existing literatures on quickly DDoS detection and defense in SDN. Finally, some promising research directions in this field are introduced.

Keywords

software defined networks; SDN security; DDoS; detection method; defense mechanism

1 Introduction

Flooding-based Distributed Denial of Service (DDoS) attacks are the most threatening challenge to Internet security today [1]. The DDoS attacker relies on sending an overwhelming number of fake packets to exhaust the resources of victims, such as CPU, memory and network bandwidth. Therefore, requests from benign users cannot be handled because of unavailable system resources. To cope with this kind of attacks, tremendous mitigation techniques have been proposed [2]–[8]. However, few of them have been extensively implemented because of their deployment complexities as well as prohibitive operational costs. One of the main reasons is that such approaches usually require placing large network connection state tables and high-end equipment at routers or switches and sometimes even requires human intervention, which increases extra storage and computational costs. As a consequence, it is desirable to design some automated, lightweight and scalable DDoS mitigation methods.

Software defined networking (SDN) is a new promise networking paradigm that radically changes the network architecture. The separation of control and data plane in SDN allows us to program the control logic and instruct the forwarding plane to behave accordingly. Furthermore, switches can be made lighter and cheaper since they no longer require computing intelligence to perform control plane processing [9].

However, the centralized control and programmability of SDN introduce new fault and attack points [10], [11]. That is to say, SDN creates new threats that are harder to avoid. For instance, a successful DDoS attack on the SDN controller may cripple the entire network. In this paper, we aim at providing an up-to-date overview of DDoS attack in SDN by presenting detection methods and defense solutions related to individual SDN components, i.e. the controller, switch and data-to-control channel.

This paper is organized as follows. Section 2 describes proposals for DDoS attack in tradition networks addressed by the concepts of SDN and introduces SDN-self DDoS attack challenges. In Section 3, attack detection methods for SDN-self DDoS attack are presented. DDoS attack solutions for each of the attack challenges in SDN are discussed in Section 4. The paper is concluded in Section 5.

2 SDN-Supported vs. SDN-Self DDOS Attack

SDN security, especially DDoS attack, has become a popular research field since software defined network was proposed. There is a contradictory relationship between SDN and DDoS attack. On the one hand, the characteristics of centralized control and programmability of SDN make it easy to detect and react to DDoS attack in tradition networks, i.e. SDN-supported security. On the other hand, the same centralized structure is

This work is supported in part by the "973" Program of China under Grant No. 2013CB329103, the National Natural Science Foundation of China under Grant No. 61271171 and No. 61401070, National Key Research and Development Program of China No. 2016YFB0800105, and the "863" Program of China under Grant No. 2015AA015702 and No. 2015AA016102.

considered vulnerable. Consequently, SDN itself may be a target of DDoS attacks.

2.1 SDN-Supported DDoS Attack

SDN-supported security uses new techniques in SDN to deal with DDoS attacks in traditional networks, from DDoS detection [12]–[14] to DDoS defense [15]–[17] (**Table 1**).

2.1.1 DDoS Detection

At present, there are many studies to detect DDoS attacks [5]–[8]. Hong Jiang, et al. [5] proposed a two-stage detection strategy which combining superpoints and flow similarity measurement. By describing the behaviors of DDoS flooding attacks with superpoints, the suspicious flows are located in a better detection strategy. In [6], a more sophisticated anomaly-based system was proposed for detecting DDoS attacks. The proposed system is designed to solve the detection problem from the perspective of computer vision. Tan, et al. [7] put forward a DDoS attack detection system that adopts multivariate correlation analysis. Moreover, with correlation analysis, another method to detect DDoS attacks against data centers was present in [8]. However, these proposed methods can be hardly applied in online detection.

The advantages of SDN give some flexible DDoS detection methods. In [12], leveraging the global flow monitoring capability of SDN, a quickly and precisely method was proposed to adaptively balance the coverage and granularity of attack detection. Based on dynamically scaling the range of detected IP addresses, this method can achieve the most granularity IP address monitoring, and complete the victim and attacker location as well. In [13], an SDN framework for data centers named FlowTrApp was proposed, which performs DDoS detection and mitigation using some bounds on two per flow based traffic parameters (flow rate and flow duration). It attempts to detect attack traffics ranging from low rate to high rate as well as long-lived to short-lived attacks using an SDN engine. CloudWatcher [14] uses SDN to build a framework to efficiently monitor services in large and dynamic cloud networks. The framework enables the network administrator to protect their network easily by writing a simple policy script.

▼Table 1. SDN-supported DDoS attack

SDN-supported	Solution	SDN capabilities exploited	Description
DDoS Detection	Sequential& concurrent method [12]	Global monitoring	Scaling the range of detected IP addresses
	FlowTrApp [13]	Traffic analysis	Using some bounds on two per flow based traffic parameters
	CloudWatcher [14]	Programmability	Protect network by writing a simple policy script
DDoS Defense	SDN/NFV security policy [15], [16]	Centralized-control or programmability	Combining SDN and NFV
	Collaborative framework [17]	Centralized-control or programmability	A self-management scheme

DDoS: Distributed Denial of Service SDN: software defined networking
NFV: network function virtualization

2.1.2 DDoS Defense

SDN separates the control plane from the data plane and hence allows the network operator to automatically steer individual flows via a central programmable interface [18]. This allows a fine-grained security policy enforcement and thus improves overall network security.

Hence, using SDN, Fayaz et al. [15] proposed a DDoS defense system named Bohatei. This system is scalable because its resource management algorithm controls the network to avoid control and data plane bottlenecks. In addition, it exploits network function virtualization (NFV) [19] capability to flexibly place the defense virtual machine (VM) resources at the locations where they are needed. In addition to that, based on SDN and NFV, a scalable security solution was provided for enterprise networks with greater flexibility and lower operational costs [16].

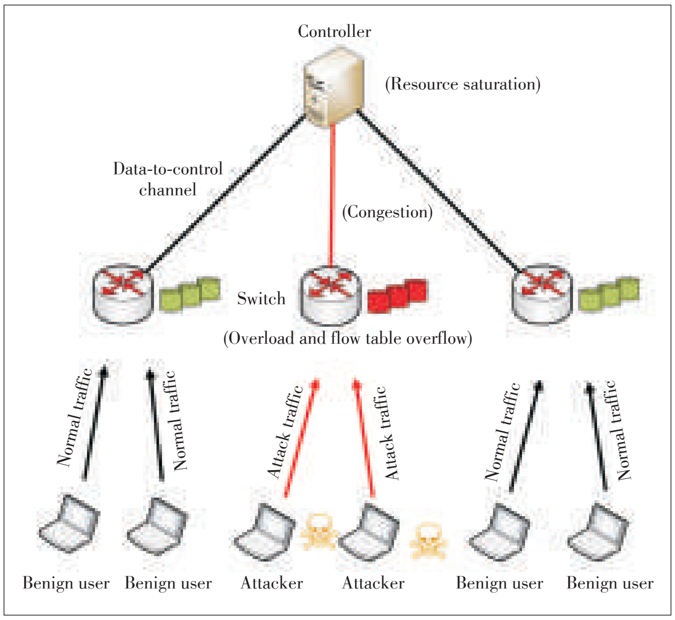
Leveraging the programmability and centralized control offered by SDN, Sahay et al. [17] proposed a self-management scheme, in which an Internet service provider (ISP) and its customers cooperate to mitigate DDoS attack. The ISP collects threat information provided by customers, then it uses this information to enforce security policy and update flow tables in the network accordingly. If a flow is treated legitimate by customers, the ISP controller will mark it with a high priority. Flows with higher priority will get better quality paths.

2.2 SDN-Self DDoS Attack

Despite the advantages of SDN (e.g. programmability, logical centralized control and flexibility) make it easy to detect and defense DDoS attacks in traditional networks, the separation of the control plane from the data plane in SDN introduces new DDoS attack threats. For example, in OpenFlow-based SDN, when a switch receives a new packet, it first checks whether there is an installed flow rule in its Ternary Content Addressable Memory (TCAM) flow table matched this packet or not. If a match is found, the packet is forwarded through the flow rule. Otherwise, the switch buffers this packet and transfers a packet-in message to the controller requesting a new flow rule. The controller then responds with a flow-mod message to instruct all the involved switches with the rules to handle this new packet [20]. An attacker can make use of this characteristic of SDN to launch DDoS attack against the switch, data-to-control channel, and controller, as illustrated in **Fig. 1**.

2.2.1 Switch Overload and Flow Table Overflow

The DDoS attacker can send a large number of table-miss packets to the victim switch. The victim switch should buffer them and generate flow requests sending to the controller since it cannot find matching rules for them. Because of limited re-



▲Figure 1. DDoS attack process in SDN.

sources (CPU and memory), the switch can only generate a limited number of flow requests. Wang et al. [21] show that a hardware switch can only generate less than 1000 requests per second. Thus, the switch may be overloaded, and as a result, flows from benign users may be delayed or dropped. Furthermore, if the controller processed flow requests successfully, a huge number of flow rules should be distributed to the victim switch. Since TCAM is a scarce resource, it only supports a small number of flow rules. For example, the Pronto-Pica8 3290 switch can only hold 2000 rules [22]. Thus, the flow table of victim switch will be filled up quickly and eventually overflow. These two threats in the switch have a local impact as they reduce the throughput of the victim switch.

2.2.2 Data-to-Control Channel Congestion

Following the instructions in 2.2.1, the flow requests flooded by the victim switch are send to the controller though the data-to-control channel, with a lot of bandwidth requirements. In addition, if the buffer of the victim switch fills up, the switch sends an entire packet instead of just a packet header to the controller, resulting in even higher bandwidth consumption. This can overwhelm common bottleneck links, and normal flow requests will experience congestion. The channel congestion affects all hosts with flow requests traversing the congested links.

2.2.3 Controller Resource Saturation

Finally, if the flooded flow requests arrive at the controller, they will consume the controller's resource (i.e. CPU, memory, and bandwidth) for flow rule computation and installation. Without any protection, the controller's resource can be saturated by the flooded requests, and legitimate requests may be dropped. Because the most crucial part of the SDN is the controller, it is also a single point of failure in the entire SDN [23].

3 Attack Detection Methods for SDN-Self DDOS Attack

The most basic but essential task in DDoS research is the detection problem. DDoS attack detection is just the first stage for withstanding DDoS attack. Attack detection mechanisms [8]–[11] used for traditional networks can be adopted in SDN [12]–[14]. Due to the centralized control, the detection for controller DDoS attack is slightly different. This section summarizes the new detection methods for controller DDoS, and then classifies the existing detection algorithms.

3.1 Detection for Controller DDoS

Because only enormous packet-in messages may exhaust controller resources, the simplest detection method is once the amount of the packet-in messages exceeds a predetermined threshold, a DDoS attack against the controller is identified. However, this method may lead to a high false detection rate. In order to improve the detection accuracy, some improved detection mechanisms are proposed [24]–[27].

Considering two facts that new flows can trigger packet-in messages to the controller and low-traffic flows are of high-efficiency for such an attack, an efficient detection method for a novel DDoS attack against SDN controllers was designed by measuring vast new low-traffic flow [25]. This method is based on Sequential Probability Ratio Test (SPRT), a statistical tool which has bounded false negative and false positive error rates. Similarly, S. M. Mousavi et al. [24] proposed an early detection method for DDoS attack against the SDN controller. It assumes that the destination IP addresses are almost evenly distributed in the normal flows, while the malicious flows are destined to a small amount of IP hosts. However, these two methods are not available when attackers generate lots of new low-traffic flows with their destination IP addresses evenly distributed.

In order to detect DDoS attacks against the SDN controller, a lightweight method for DDoS attacks detection based on traffic flow features is presented in [26]. The method contains three modules, i.e. Flow Collector, Feature Extractor and Classifier. The Flow Collector module is responsible for periodically requesting flow entries from all flow tables of OpenFlow switches, then the Feature Extractor module receives the collected flows and extracts features that are important to DDoS attack detection. At last, the Classifier module analyzes whether or not the features correspond to attack or legitimate traffic. This study makes use of the flow statistics character of the OpenFlow switch, but it does not take the controller's overhead caused by flow entries collecting process into consideration. In [27], combining with the sampled flow (sFlow) protocol, the authors reduced flow data gathering by sampling and reduced the required communication between the OpenFlow switches and controller, thus easing the control plane's overload in the large network traffic condition. Moreover, the authors designed a work-wide anomaly mitigation using OpenFlow. However, the authors did not study if the flow sampling may affect the accuracy of anomaly detection.

3.2 Classification of Detection Methods in SDN

According to the detection algorithms used, we can classify the existing detection methods into the machine learning based [26], [28], [29], the entropy based [24], [30] and the graphic based [31], [32].

Machine learning based techniques for handling DDoS attacks have received much attention in the computational intelligence community. DDoS attacks can be detected by using the machine learning algorithm that was trained with attack and normal patterns. Braga et al. [26] classified network traffic by using Self-Organizing Maps (SOM) [33]. In [28], the intrusion detection system utilizes SVM classifier to detect DDoS attacks. In [29], different machine learning algorithms such as Naive Bayes, K-Nearest neighbor and K-means were used in advanced signature-based intrusion detection system (IDS) to find the sets of hosts that have normal or anomalous behavior. However, the machine learning based methods require a large number of training sets and spend a long time for training.

The entropy-based detection mechanism has a relative low calculation overload. The entropy is used to measure the randomness change of the incoming flows during a given time period. Moreover, the flow-based feature of SDN makes it more convenient to calculate the entropy value. Based on the entropy variation of destination IP address of the incoming packets, an early detection method was proposed in [24] for DDoS attacks against the SDN controller. Moreover, an entropy-based anomaly detection model for DDoS flooding attack in SDN is present in [30]. Differing from the mentioned other methods, this detection algorithm runs in the OpenFlow edge switch. By doing statistics and analysis on the network traffic coming to the OpenFlow network, it achieves detecting the attack locally. Although the entropy-based methods are more flexible, they need to combine with other technologies to make threshold determination and multi-element weight assignment.

Among the graphic model based detection methods, a graphic model based on an attack detection method that deals with a dataset shift problem was proposed by Wang et al. [31]. It saves known traffic patterns as a relational graph. If new traffic is generated, the system can determine whether the traffic is malicious by comparing the graphs. SPHINX [32] was proposed to detect both known and potentially unknown attacks on network topology and data plane forwarding originating within a software defined network. SPHINX leverages the novel abstraction of flow graphs, which closely approximates the actual network operations. It enables incremental validation of all network updates and constraints. SPHINX dynamically learns new network behavior and raises alerts when it detects suspicious changes to the existing network control plane behavior. Graphic models are an effective tool to validate normal and abnormal network behavior. However, if the network topology dynamically changes very frequently, several of the learned invariants may be interrupted, resulting in false detection.

3.3 Others

The abovementioned methods only consider a single IDS in software defined networks. With multiple IDSs, the detection performance highly depends on the way by which the suspicious traffic flows are distributed among the multiple IDSs. In [34], considering the detection of malicious attacks against SDN with multiple IDSs, the proposed algorithm distributes the flows to multiple IDSs according to their routing paths. If two flows are close to each other in terms of the routing path, they are forwarded to the same IDS. Moreover, it uses a gravity clustering algorithm to group the flows, and the cluster size is inversely proportional to the sum of the data rates in each group for load balancing.

4 Defense Mechanisms for SDN-Self DDOS Attack

Once a DDoS attack is detected, a timely and effective defense method is required to restore the network function and reduce the network loss. In this section, we summarize the corresponding defensive measures against the three types of DDoS attacks mentioned in Section 2.2 (**Table 2**).

4.1 Reacting Against Switch Overload and Flow Table Overflow

To mitigate such DDoS threats, Dao et al. [35] present a so-

Table 2. An overview of DDoS countermeasures in SDN system

Defense techniques	DDoS treats		
	Switch overload	Channel congestion	Controller resource saturation
IP filtering [35]	√		
Scotch [21]	√		
Lightweight [36]		√	
FlowSec [37]		√	
FloodDefender [44]	√	√	√
MLFQ [38]			√
FRESCO [45]	√	√	√
FloodGuard [40]			√
FlowRanger [39]			√
Avant-Guard [42]			√
SDNShield [41]			√
SDN-Guard [43]	√	√	√

DDoS: Distributed Denial of Service SDN: software defined networking
MLFQ: Multilayer Fair Queueing

lution to protect software defined networks based on IP filtering technique. The proposed scheme analyzes user behavior and uses it to assign the timeouts for the flow entries. Short timeouts are assigned for malicious users flows and long timeouts are used for trusted ones. This solution forces entries of malicious traffic to be quickly removed TCAM tables of the switches. However, this may lead to new packet-in messages to be sent to the controller if the flow duration is higher than the set timeout. Furthermore, this solution drops all malicious traffic, which may be problematic for false positive flows.

Scotch [21] uses an overlay network of software switches as a complement to hardware switches. Since software switches can run on more powerful CPUs, they can generate many more flow requests compared to hardware switches. New flows received by hardware switches would be redirected to software switches, which are responsible for generating flow requests. Data plane traffic can still be forwarded by hardware switches for large throughput. Indeed, Scotch can increase the number of new flows that a switch can handle in benign settings; however, it may not be enough for adversarial settings, where an attacker can flood at a rate even higher than a software switch can handle.

4.2 Reacting Against Data-to-Control Channel Congestion

In [36], a lightweight information hiding authentication mechanism was proposed to prevent DDoS attacks in the SDN control channel. In [37], by enforcing a rate limit on the number of packets sent to the controller, FlowSec was introduced to mitigate an attack on the controller bandwidth. FlowSec collects the switch statistics and computes controller bandwidth dynamically. If there is an attack, FlowSec uses the Floodlight module to collect switch statistics and instructs the switch port to slow down. Although this method can mitigate DDoS attacks in the SDN system, it also hinders other switches and normal traffic.

4.3 Reacting Against Controller Resource Saturation

To our understanding, the most vulnerable component in the SDN architecture is the centralized controller. As a result, in recent years, researchers have proposed a variety of strategies to mitigate controller resource saturation attacks in software defined networks. Among them, P. Zhang et al. [38] proposed a novel queue management method that allows dynamic queue expansion and aggregation named Multilayer Fair Queueing (MLFQ). This method is based on enforcing fair sharing of a controller's resources among switches and hosts in the network. When attacks take place, the controller expands the corresponding queues into multiple lower-level queues to isolate flooded requests. In this way, the controller in general only needs to maintain a small number of queues. Despite its advantages, when the number of attack streams is large, this approach is poorly handled.

L. Wei, et al. proposed FlowRanger [39], a flow prioritizing algorithm which is implemented at the controller side to enhance the Quality of Service (QoS) of regular users. In this scheme, a ranking algorithm is first used to identify regular normal users based on their past requests to the controller. Then, the execution of requests prioritized by using multiple priority buffers. Finally, the packets are processed according to a weighted Round Robin strategy, i.e. the packets in a higher priority buffer are handled with higher priority than those in lower priority buffers. FlowRanger can reduce the impact of DDoS attacks on network performance by guaranteeing that legitimate flows are served first in the controller. However, the flows satisfying these criteria are not necessarily malicious. They may also be benign flows that happen to appear with the attack traffic for their first visit. Therefore, simply blocking these requests is not a good solution.

FloodGuard [40] can defend against general flow request flooding attacks. Once the controller detects an attack, it installs a default rule at the victim switch to redirect all new flows to a data plane cache. The data plane cache is responsible for generating flow requests to the controller. At the same time, the controller proactively generates rules by symbolically executing controller applications, and installs these rules at the victim switch to suppress future flow requests. One possible problem with FloodGuard is that symbolic execution may not exhaust all possible execution paths for complicated controller applications. In addition, FloodGuard needs to deploy extra devices on the data plane.

SDNShield [41], a combined solution towards more comprehensive defense against DDoS attacks on SDN control plane. It uses specialized software boxes to improve the scalability of ingress SDN switches to accommodate the control plane workload surge. It further incorporates a two-stage filtering scheme to protect the centralized controller. It statistically distinguishes legitimate flows from forged ones at the first stage, and recovers the false positives of the first stage with in-depth TCP handshake verification at the second stage.

Avant-Guard [42] is an extension to the existing OpenFlow data plane with the addition of Connection Migration (CM). The Avant-Guard responds to handshake packets if no matching flow entries are found. Only when a connection is established, the packet is sent to the controller to ask for a routing path. The purpose of Avant-Guard is to fight against DDoS attacks based on IP spoofing, by effectively reducing the amount of data to the control plane under DDoS attacks. However, a weakness of Avant-Guard is its implementation on switches. All switches need to be Avant-Guard equipped, otherwise the entire network is still vulnerable.

4.4 Others

In [43], a novel SDN application named SDN-Guard is proposed to protect SDN system against DDoS attacks, and simultaneously mitigate DDoS impact on the SDN controller, data-to-control bandwidth and switch. It can dynamically manage flow routes, rule entry timeouts and the aggregate flow rule entries based on the flow threat probability provided by an IDS.

FloodDefender [44] is a scalable and protocol-independent defense system for protecting OpenFlow networks against SDN-aimed DDoS attacks. It consists of four functional modules: the attack detection, table-miss engineering, packet filter, and flow rule management modules. When no attacks are detected, FloodDefender forwards the packet in messages, actions, and fl ow rules between the controller platform and controller apps. When attacks occur, FloodDefender detours table-missing packets to neighbor switches with wildcard flow rules to protect the communication link from being jammed, filters out attack packets from the received packet in messages to save the computational resources, and constructs a robust flow table in the data plane by separating the flow table into "flow table region" and "cache region" to save the TCAM of OpenFlow switches.

In [45], an OpenFlow security application development framework FRESCO is proposed. As an OpenFlow application, it offers a programming framework that enables security researchers to implement, share, and compose many different security modules and also exports a scripting API that enables security practitioners to code security monitoring and threat detection logic as modular libraries.

5 Conclusions

The emergence of SDN provides a new paradigm to solve DDoS problem in traditional networks by introducing separate layers for routing and data forwarding. At the mean time, SDN DDoS threat has become an open research field for researchers. In this article, we summarize how a traditional network can incorporate the concept of SDN to solve the issue of DDoS attacks. Then we describe SDN-self DDoS attacks followed by a comprehensive survey of proposed detection methods and defense countermeasures.

Although many methods and systems have been developed by the research community, there are still many open research issues that are not well investigated and need to be addressed by future research efforts. For detection, controller modules often aggregate flow rules to conserve switch TCAM. After the flow table is compressed, the switch reports coarse-grained statistics. How to effectively detect attacks in SDN networks with flow table compression is a problem. Meanwhile, in a software defined network with multi-controllers, how to design efficient detection algorithms to balance the overhead and detection accuracy is another problem. For defense, combining with many other promising technologies in next-generation networks, such as NFV and Information Centric Networking (ICN), may bring in some research opportunities. Furthermore, most of the existing mitigation methods only handle abnormal flows, such as discarding or limiting ones. There is barely a complete method to resolve the problems from the attack source. The location of attack sources and victim hosts is also a relatively new research point.

Acknowledgment

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions.

References

[1] S. Stuart, "Akamai releases prolexic Q2 2014 global DDoS attack report," *Database & Network Journal*, vlo. 44, no. 4, Aug. 2014.

[2] C. Jin, H. Wang, and K. G. Shin, "Hop-count filtering: an effective defense against spoofed DDoS traffic," in *Proc. ACM Conference on Computer and Communications Security* (CCS 03), Washington, USA, Oct. 2003.

[3] Z. S. Taghavi, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013. doi: 10.1109/SURV.2013.031413.00127.

[4] Q. Liao, D. A. Cieslak, A. D. Striegel, and N. V. Chawla, "Using selective, short-term memory to improve resilience against DDoS exhaustion attacks," *Security and Communication Networks*, vol. 1, no. 4, pp. 287–299,2008. doi: 10.1002/sec.22.

[5] H. Jiang, S. Chen, H. Hu, and M. Zhang, "Superpoint-based detection against distributed denial of service (DDoS) flooding attacks," in *IEEE International Workshop on Local and Metropolitan Area Networks*, Beijing, China, 2015, pp. 1–6. doi: 10.1109/LANMAN.2015.7114724.

[6] Z. Tan, A. Jamdagni, X. He, et al., "Detection of denial-of-service attacks based on computer vision techniques," *IEEE Transactions on Computers*, vo. 64, no. 9, pp. 2519–2533, 2015. doi: 10.1109/TC.2014.2375218.

[7] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, "A system for denial-of-service attack detection based on multivariate correlation analysis," *IEEE Transactions on Parallel & Distributed Systems*, vol. 25, no. 2, pp. 447–456, 2014. doi: 10.1109/TPDS.2013.146.

[8] P. Xiao, W. Qu, H. Qi, and Z. Li, "Detecting DDoS attacks against data center with correlation analysis," *Computer Communications*, vol. 67, no. C, pp. 66–74, 2015. doi: 10.1016/j.comcom.2015.06.012.

[9] N. McKeown, T. Anderson, H. Balakrishnan, et al., "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008. doi: 10.1145/1355734.1355746

[10] M. Antikainen, T. Aura, and M. Särelä, "Spook in your network: attacking an SDN with a compromised OpenFlow switch," in *Proc. 19th Nordic Conference on Secure IT Systems (NordSec14)*, Tromsø, Norway, Oct. 2014. doi: 10.1007/978-3-319-11599-3_14.

[11] D. Kreutz, F. M. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in *Proc. Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, ser. HotSDN '13*. New York, USA, 2013, pp. 55–60. doi: 10.1145/2491185.2491199.

[12] X. Yang and Y. Liu, "DDoS attack detection under SDN context," in *Proc. IEEE International Conference on Computer Communications IEEE(INFOCOM16)*, San Francisco, USA, 2016. doi: 10.1109/INFOCOM.2016.7524500.

[13] B. Chaitanya and N. Medhi, "FlowTrApp: an SDN based architecture for DDoS attack detection and mitigation in data centers," in *Proc. 3rd International Conference on Signal Processing and Integrated Networks*, Noida, India, 2016. doi: 10.1109/SPIN.2016.7566750.

[14] S. Shin and G. Gu, "CloudWatcher: network security monitoring using OpenFlow in dynamic cloud networks," in *IEEE International Conference on Network Protocols*, Austin, USA, 2012, pp. 1–6. doi: 10.1109/ICNP.2012.6459946.

[15] S. K. Fayaz, Y. Tobioka, V. Sekar, and M. Bailey, "Bohatei: flexible and elastic DDoS defense," in *24th Usenix Conference on Security Symposium*, Washington, D. C., USA, 2015, pp. 817–832.

[16] C. Lorenz, D. Hock, J. Scherer, et al., "An SDN/NFV-enabled enterprise network architecture offering fine-grained security policy enforcement," *IEEE Communications Magazine*, 2017, vol. 55, no. 3, pp. 217–223. doi: 10.1109/MCOM.2017.1600414CM.

[17] R. Sahay, et al. "Towards Autonomic DDoS Mitigation using Software Defined Networking." *NDSS Workshop on Security of Emerging networking Technologies*, 2015. doi: 10.14722/sent.2015.23004.

[18] M. Jarschel, T. Zinner, T. Hossfeld, P. Tran-Gia, and W. Kellerer, "Interfaces, attributes, and use cases: a compass for SDN," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 210–217, 2014. doi: 10.1109/MCOM.2014.6829966.
[19] Q. Duan, N. Ansari, and M. Toy. "Software-defined network virtualization: an architectural framework for integrating SDN and NFV for service provisioning in future networks," *IEEE Network*, vol. 30, no. 5, pp. 10–16, 2016. doi: 10.1109/MNET.2016.7579021.
[20] S. Shin and G. Gu, "Attacking software-defined networks: a first feasibility study," in *ACM SIGCOMM Workshop Hot Topics Software Defined Network (HotSDN13)*, Hong Kong, China, 2013, pp. 165–166.
[21] A. Wang, Y. Guo, F. Hao, T. V. Lakshman, and S. Chen, "Scotch: elastically scaling up SDN control-plane using vSwitch based overlay," in *ACM International Conference on Emerging NETWORKING Experiments and Technologies*, Sydney, Australia, 2014, pp. 403–414. doi: 10.1145/2674005.2675002.
[22] N. Katta, O. Alipourfard, J. Rexford, and D. Walker, "CacheFlow: dependency-aware rule-caching for software-defined networks," in *ACM Symposium on SDN Research*, Santa Clara, USA, 2016, article no. 6. doi: 10.1145/2890955.2890969.
[23] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: a survey, some research issues, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no.1, pp. 602–622, 2016. doi: 10.1109/COMST.2015.2487361.
[24] S. M. Mousavi and M. St-Hilaire. "Early detection of DDoS attacks against SDN controllers," in *Proc. IEEE International Conference on Computing, Networking and Communications*, Anaheim, USA, 2015, pp. 77–81. doi: 10.1109/ICCNC.2015.7069319.
[25] P. Dong, X. Du, H. Zhang, and T. Xu, "A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows," *IEEE International Conference on Communications*, Kuala Lumpur, Malaysia, 2016, pp. 1–6. doi: 10.1109/ICC.2016.7510992.
[26] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *35th Annual IEEE Conference on Local Computer Networks*, Denver, USA, 2011, pp. 408–415. doi: 10.1109/LCN.2010.5735752.
[27] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Computer Networks*, vol. 62, no. 5, pp. 122–136, 2014. doi: 10.1016/j.bjp.2013.10.014.
[28] R. T. Kokila, S. T. Selvi, and K. Govindarajan, "DDoS detection and analysis in SDN-based environment using support vector machine classifier," in *Proc. IEEE Sixth International Conference on Advanced Computing*, Chennai, India, 2015. doi: 10.1109/ICoAC.2014.7229711.
[29] L. Barki, A. Shidling, and N. Meti, "Detection of distributed denial of service attacks in software defined networks," in *Proc. IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Jaipur, India, 2016, pp. 2576 – 2581. doi: 10.1109/ICACCI.2016.7732445.
[30] R. Wang, Z. Jia, and L. Ju, "An entropy-based distributed DDoS detection mechanism in software-defined networking," *IEEE Trustcom/BigDataSE/ISPA*, Helsinki, Finland, 2015, pp. 310–317. doi: 10.1109/Trustcom.2015.389.
[31] B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, "DDoS attack protection in the era of cloud computing and software-defined networking," *Computer Networks*, vol. 81, pp. 308–319, 2015. doi: 10.1109/ICNP.2014.99.
[32] M. Dhawan, R. Poddar, K. Mahajan, and V. Mann, "SPHINX: detecting security attacks in software-defined networks," in *Network and Distributed System Security Symposium*, San Diego, USA, 2015. doi: 10.14722/ndss.2015.23064.
[33] S. Ostermann, B. Tjaden, and M. Ramadas, "Detecting anamalous network traffic with self-organizing maps," in *Proc. 6th International Workshop on Recent Advances in Intrusion Detection*, Pittsburgh, USA, 2003, pp. 36–54. doi: 10.1007/978-3-540-45248-5_3.
[34] T. Ha, S. Yoon, A. C. Risdianto, J. Kim, and H. Lim, "Suspicious flow forwarding for multiple intrusion detection systems on software defined networks," *IEEE Network*, vol. 30, pp. 6, pp. 22–27, 2016. doi: 10.1109/MNET.2016.1600106NM.
[35] N. N. Dao, J. Park, M. Park, and S. Cho, "A feasible method to combat against DDoS attack in SDN network," in *International Conference on Information Networking*, Cambodia, Cambodia, 2015, pp. 309–311. doi: 10.1109/ICOIN.2015.7057902.
[36] O. I. Abdullaziz, Y.-J. Chen, and L.-C. Wang, "Lightweight authentication mechanism for software defined network using information hiding," in *IEEE Global Communications Conference (GLOBECOM)*, Washington, D.C., USA, 2016. doi: 10.1109/GLOCOM.2016.7841954.
[37] M. Kuerban, Y. Tian, Q. Yang, et al., "FlowSec: DOS attack mitigation strategy on SDN controller," in *IEEE International Conference on Networking, Architecture and Storage*, Long Beach, USA, 2016, pp. 1–2. doi: 10.1109/NAS.2016.7549402.
[38] P. Zhang, H. Wang, C. Hu, and C. Lin, "On denial of service attacks in software defined networks," *IEEE Network Magazine*, vol. 30, no. 6, pp. 28–33, 2016. doi: 10.1109/MNET.2016.1600109NM.
[39] L. Wei and C. Fung, "FlowRanger: a request prioritizing algorithm for controller DoS attacks in software defined networks," in *IEEE International Conference on Communications*, London, UK, 2015, pp. 5254–5259. doi: 10.1109/ICC.2015.7249158.
[40] H. Wang, L. Xu, and G. Gu. "FloodGuard: a DoS attack prevention extension in software-defined networks," in *45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, Rio de Janeiro, Brazil, 2015, pp. 239–250. doi: 10.1109/DSN.2015.27.
[41] K. Chen, A. R. Junuthula, I. K. Siddhrau, Y. Xu, and H. J. Chao, "SDNShield: towards more comprehensive defense against DDoS attacks on SDN control plane," in *Proc. IEEE Conference on Communications and Network Security (CNS)*, Philadelphia, USA, 2016. doi: 10.1109/TPDS.2013.146.
[42] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks," in *ACM Sigsac Conference on Computer & Communications Security*, Berlin, Germany, 2013, pp. 413–424. doi: 10.1145/2508859.2516684.
[43] L. Dridi and M. F. Zhani, "SDN-guard: DoS attacks mitigation in SDN networks," in *IEEE International Conference on Cloud Networking IEEE*, Pisa, Italy, 2016. doi: 10.1109/CloudNet.2016.9.
[44] S. Gao, Z. Peng, B. Xiao, A. Hu, and K. Ren, "FloodDefender: protecting data and control plane resources under SDN-aimed DoS attacks," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Atlanta, USA, 2017, pp. 1–9.
[45] S. Shin, P. Porras, V. Yegneswaran, et al., "FRESCO: modular composable security services for software defined networks," *In Proc. Network & Distributed Security Symposium*, San Diego, USA, 2013, pp. 319–332.

Manuscript received: 2017-06-03

Biographies

XU Xiaoqiong (xiaoqiongxu@std.uestc.edu.cn) is currently a Ph.D. student in University of Electronic Science and Technology of China, China. Her research interests include software defined networking and cloud computing.

YU Hongfang (yuhf@uestc.edu.cn) received her B.Sc. degree in electrical engineering in 1996 from Xidian University, China her M.Sc. degree and Ph.D. degree in communication and information engineering in 1999 and 2006 from University of Electronic Science and Technology of China, respectively. From 2009 to 2010, she was a visiting scholar at the Department of Computer Science and Engineering, University at Buffalo (SUNY), USA. Her research interests include network survivability and next generation Internet, and cloud computing.

YANG Kun (kunyang@uestc.edu.cn) received his Ph.D. from the Department of Electronic & Electrical Engineering of University College London (UCL), UK, and M.Sc. and B.Sc. from the Computer Science Department of Jilin University, China. He is currently a Chair Professor in the School of Computer Science & Electronic Engineering, University of Essex, leading the Network Convergence Laboratory (NCL), UK. He is also an affiliated professor at University of Electronic Science and Technology of China, China. Before joining in University of Essex at 2003, he worked at UCL on several European Union (EU) research projects for several years. His main research interests include wireless networks and communications, future Internet technology and network virtualization, mobile cloud computing. He manages research projects funded by various sources such as UK EPSRC, EU FP7/H2020 and industries. He has published 100+ journal papers. He serves on the editorial boards of both IEEE and non-IEEE journals. He is a senior member of IEEE (since 2008) and a Fellow of IET (since 2009).

DOI: 10.3969/j.issn.1673-5188.2017.03.004
http://kns.cnki.net/kcms/detail/34.1294.TN.20170918.1013.002.html, published online September 18, 2017

Evolutionary Algorithms in Software Defined Networks: Techniques, Applications, and Issues

LIAO Lingxia[1], **Victor C. M. Leung**[1], **and LAI Chin-Feng**[2]
(1. Department of Electrical and Computer Engineering, University of British Columbia, Vancouver BC V6T 1Z4, Canada;
2. Department of Engineering Science, National Cheng Kung University, Tainan, China)

Abstract

A software defined networking (SDN) system has a logically centralized control plane that maintains a global network view and enables network-wide management, optimization, and innovation. Network-wide management and optimization problems are typically very complex with a huge solution space, large number of variables, and multiple objectives. Heuristic algorithms can solve these problems in an acceptable time but are usually limited to some particular problem circumstances. On the other hand, evolutionary algorithms (EAs), which are general stochastic algorithms inspired by the natural biological evolution and/or social behavior of species, can theoretically be used to solve any complex optimization problems including those found in SDNs. This paper reviews four types of EAs that are widely applied in current SDNs: Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Simulated Annealing (SA) by discussing their techniques, summarizing their representative applications, and highlighting their issues and future works. To the best of our knowledge, our work is the first that compares the techniques and categorizes the applications of these four EAs in SDNs.

Keywords

SDN; evolutionary algorithms; Genetic Algorithms; Particle Swarm Optimization; Ant Colony Optimization; Simulated Annealing

1 Introduction

As various Internet-connected devices and advanced network applications gain popularity, the Internet traffic has become more and more complex in its data volume, data type, Quality of Service (QoS)/Quality of Experience (QoE) requirements, and security. This increased complexity creates a significant challenge in network management, which calls for flexible solutions through programmability of network devices. Conventional IP networks tightly couple the control logic on dedicated network devices, which are likely provided by diverse vendors, to configure, control, and monitor data flows. This makes it rather difficult for the network devices to cooperate and collect information on network dynamics that are changing in real time, to make decisions based on the network dynamics, and to enforce these decisions by automatically configuring or reconfiguring network devices. This motivates a new networking paradigm called software defined networking (SDN), which decouples network control from conventional network devices to form a logically centralized control plane, while a physically distributed data plane consisting of the network devices as data forwarders efficiently forwards the packets of individual data flows based on the rules generated by the control plane.

Software defined networks have a layered architecture consisting of a data layer, a control layer, and an application layer [1] (**Fig. 1**). The data layer forms a data plane that includes multiple simple network forwarders (switches) providing packet switching and forwarding, and also network statistics collec-

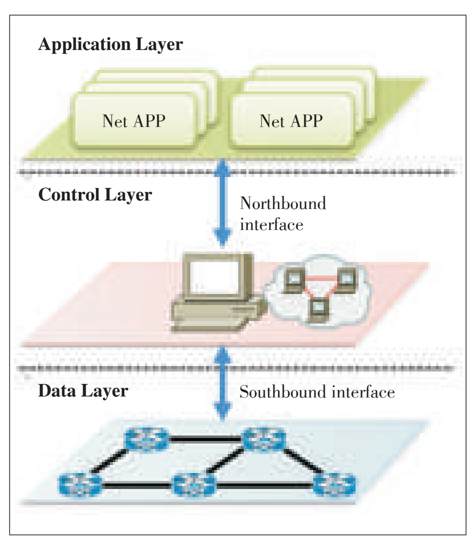


Figure 1. Layered architecture of an SDN system.

tion and reporting capabilities. The control layer is responsible for providing logically centralized control functionality for management of network nodes and flow forwarding. The application layer consists of end-user business applications that control switching devices by invoking the services in the control layer. The control layer and application layer form the SDN control plane. The SDN architecture provides an open standardized south-bound interface (e.g., OpenFlow protocol [2]) to manage the communications between the data and control layers. However, the north-bound interface between the control and application layers and the east-to-west-bound interfaces between the controllers inside the control layer are defined only functionally but not standardized. With these interfaces, a software defined network can maintain a logically centralized global network view in its control layer, allow each network application in the application layer to retrieve data from the global network view, and enforce network-wide management or security policies to optimize network performance, security, and resource usage. SDN makes a perfect architecture to enable network-wide optimization, artificial intelligence (AI), and machine learning mechanisms to form an open, customizable, programmable, and manageable network.

Network-wide optimization, AI, and machine learning problems typically are complex with huge search spaces, large numbers of variables, and multiple objectives. Many algorithms have been proposed to solve these problems in various user scenarios, among which evolution algorithms (EAs) are attractive candidates. EAs are stochastic algorithms inspired by the natural biological evolution and/or social behavior of species [3]. EAs typically have three major processes [4]: the initialization process, evaluation process, and new population generation process (**Fig. 2**). The initialization process randomly generates initial individuals, each of which represents a problem solution directly or indirectly as a string consisting of multiple elements, each element being a variable of the problem. In the evaluation process, each solution is evaluated for its fitness

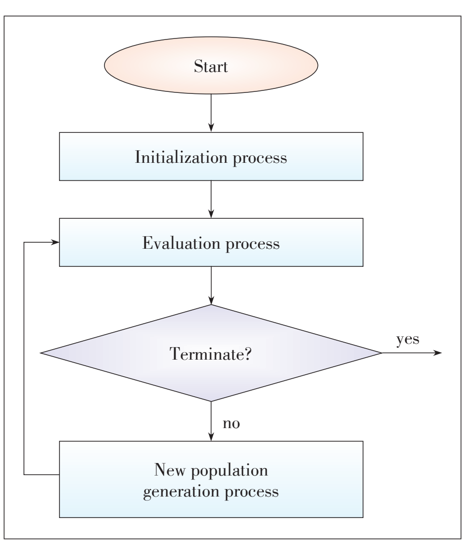


Figure 2. General flow chart of EAs.

against the objectives so that the solutions with higher fitness value will be selected to feed into the new population generation process to generate a new population set for next iteration. Various EAs employ different ways to generate initial populations, evaluate fitness, and generate new populations.

As algorithms for solving complex optimization problems, EAs are general and can adapt to an unknown environment and autonomously decide the parameters of a dynamical optimization problem. By exploiting the diversity of solutions, an EA finds out the best solutions in a set of population with high fitness values and evolve to the next generation. Thus EAs can provide easily implementable scalability for solving a wide range of single- and multi-objective optimization problems [5]. These features motivate the growing interest in applying EAs in SDNs or many other complex systems.

This paper reviews four widely used types of EAs: Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Simulated Annealing (SA). We compare their formulation and characteristics and provide a brief survey of their application in SDNs. The goal of this paper is not to provide an exhaustive survey, but to highlight the features of these EAs, provide a representative sample of the important work, and point out the research issues and challenges in applying EAs in SDNs. Though there have been some surveys on EAs [5], [6] or on SDNs [7]–[9], to the best of our knowledge, our work is the first to summarize the features of these four EAs and review their application in SDNs.

The rest of this paper is organized as follows. Section 2 reviews the related literature. Section 3 introduces the four types of EAs. Section 4 summarizes the applications of these EAs in SDN networks. Section 5 highlights the potential issues and challenges. Conclusions are drawn in Section 6.

2 Review of Related Works

2.1 Surveys on EAs

EAs have been a hot research topic for many years and applied to solve complex problems in a large range of science and engineering fields. Many surveys on EAs have been published in the recent literature. Elbeltagi in [10] compares the formulation and performance of five EAs: GAs, memetic algorithms, PSO, ACO, and shuffled frog leaping. Crepinšek in [11] reviews nearly 100 existing papers since 2013 and summarizes how EAs do exploration and exploitation to achieve close-to-optimal solutions over a short convergence time. Von Lücken in [12], Mukhopadhyay in [13], and Cheshmehgaz in [14] review the application of EAs to solve multiple-objective optimization problems. EAs in data mining are reviewed in [13] and [15], distributed EAs are reviewed in [16], hybrid EAs are reviewed in [17], and approaches to optimize the performance of EAs are reviewed in [18]–[20]. Mehboob's recent survey on the applications of GAs in wireless networks [5] is similar to this paper in summarizing the techniques and applications of EAs and pointing out the major issues and challenges in designing and applying EAs, but it focuses on GAs in wireless networks while this paper reviews the technologies and applications of PSO, ACO, SA as well as GAs for SDN networks.

2.2 Surveys on SDNs

Nunes in [6], Kreutz in [7], and Robertazzi in [8] and many more researchers have presented comprehensive surveys on SDN architecture, protocols, implementations, tools, and applications. Some works are focused on particular issues of SDNs; for instance, security issues have been reviewed by Yan [21], Scott-Hayward [22], and Farhady [23], network virtualization on SDN architecture has been reviewed by Blenk [24] and Jain [25], and flow table optimization has been reviewed by Jain [26]. Other works are focused on combining SDN with other networking technologies; for instance, the architecture, implementation, and applications of wireless SDNs have been summarized in [27], approaches enabling mobile SDNs are summarized in [28], application of SDN in wireless sensor networks is considered in [29], and application of SDN to cloud networks is studied in [30]. The most relevant work to our survey is [9], which reviews the applications of AIs in SDN networks, while our survey is focused on EAs, a subset of AI algorithms, and reviews the algorithms, the applications, and the issues of EAs in SDN networks.

3 Evolutionary Algorithms

EAs are the computational systems that mimic the efficient behavior of species and seek fast and robust solutions for complex optimization problems. They are the stochastic algorithms and can be used to find out the approximate optimal solutions for NP-hard optimization problems. GAs are the earliest EAs introduced by Holland in [31] in 1975. Later on other types of EAs are developed. As described in the introduction section, EAs typically have the same three processes: the initialization process, the evaluation process, and the new population generation process. In the rest of this section, we choose four major EAs widely used in current SDNs: GAs, PSO, ACS, and SA, and discuss how they can form their own initialization, evaluation, and new population generation processes.

3.1 GAs

GAs are inspired by biological genetic evolution that selects the individuals with best fitness values to generate offspring through crossover and mutation operations. As shown in **Fig. 3**, its new population generation process is split into selection, crossover, and mutation sub-processes. In the initialization process, GAs have to randomly generate a population set, each population represents a solution of the optimization problem. GAs often use a string (chromosome) consisting of a number of elements (genes) to represent a solution. The evaluation process defines a fitness function against the objectives to evaluate the fitness of a solution. The selection is used to form a parent set to feed into crossover and mutation functions. With the fitness value of solutions evaluated, the selection sub-process ranks the solutions and the ones with the higher fitness values form the parent set for offspring generation. The selection simulates the survivor of the fittest: the chromosomes with higher fitness are selected with higher probabilities to generate offspring. Crossover sub-process generates a child by mixing the genes of two parents and the mutation sub-process generates a child by randomly changing some of the genes in chromosomes. On one hand, crossover operation exploits the best traits of the current chromosomes, and strong chromosomes (with higher fitness) are more likely to be selected as parents, and hence there is a big chance that the new chromosomes may become similar after several generations, and the diversity of the population may decline and lead to population stagnation. On the other hand, mutation operation explores chromosomes to discover new traits. It injects diversity into the population and avoids the population stagnation. Crossover along with mutation provides the necessary "evolutionary mix of small steps and occasional wild gambles" to facilitate robust search in complex solution spaces [31]. Selected individuals are genetically modified to form the next generation of the population, the usage of crossover and mutation and stochastic selection allow a gradual improvement in the fitness of the solution and allow GAs to keep away from local optima.

Applying GAs to solve an optimization problem has to firstly encode problem's solutions into chromosomes. There are many ways to encode a solution. **Table 1** lists almost all the popular encoding ways proposed in current research, and among them, binary encoding is the earliest encoding method and has been widely used in GAs, but it generates many chro-

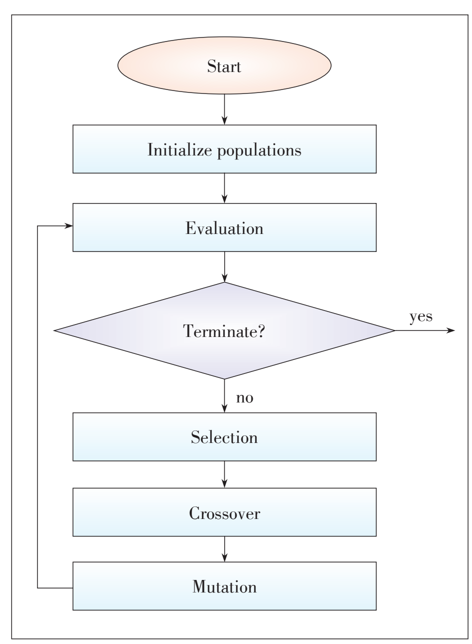


Figure 3. Flow chart of GAs.

mosomes even for a problem with small search space and may not suitable for some optimization problems. Permutation encoding is typically designed for optimizing ordering problems such as traveling salesman problems or job scheduling problems. Tree encoding is particularly used for expressions or evolving programming languages. Value encoding is useful in dealing with optimization problems when the variables are in real numbers or other complicated values, but the crossover and mutation functions may have to be updated to make the real numbers workable.

Fitness functions are corresponding to the objectives. They are typically the objective functions of optimization problems. A fitness function provides a mechanism to evaluate the solutions of a problem. Since the fitness function is utilized by each population to evaluate its fitness at each iteration, carefully designing the fitness function can reduce the convergence time of a GA and hence improve the performance of a GA.

3.1.1 Major Parameters of GAs

A GA has a number of parameters. It firstly needs to determine the population size and the maximum generations. The population size is the number of populations a GA has to maintain in a generation, while the maximum number of iterations is the maximum number of loops the algorithm can run. A GA also has to choose a selection function, crossover function, and mutation function.

A selection function often develops a particular way to select the members with higher fitness value to form a parent set for later crossover and mutation. Different selection approaches may require different methods to generate and assign probability to make sure the diversity and the improvement of the new populations. As listed in Table 1, the typical selection mechanisms used in current GAs include: 1) relative tournament evaluation; 2) roulette wheel selection; 3) relative pooling tournament evaluation; and 4) elitism. Relative tournament evaluation randomly chooses two members from the current population set, and the one with the higher fitness value is selected as a parent for next generation. Roulette wheel selection picks parents from individuals based on their fitness values, the higher the fitness value an individual has, the larger probability an individual has to be chosen as a parent. Relative pooling tournament evaluation throws members of the current population set in a competition, and the winner of the competition will be chosen as parents. Elitism maintains an archive that records all the populations that have been considered so far, and the individuals with the better fitness values in the archive are chosen as parents. Roulette wheel selection and relative pooling tournament evaluation are widely used.

As listed in Table 1, the typical crossover methods consist of: 1) one-point crossover; 2) two-point crossover; 3) uniform crossover; 4) cut and splice; and 5) ordered chromosome crossover. One-point crossover randomly generates a crossover point for two parents, swaps the genes before or after the crossover point to generate a new child chromosome. Two-point crossover randomly generates two crossover points for two parent, and the genes between these two points are swapped to produce new child chromosomes. Uniform crossover uses a fixed mixing ratio between two parents. Cut and splice crossover allows each parent to have its own choice in deciding crossover point. Ordered chromosome crossover consists of multiple crossover methods that change the chromosome by switching the position of genes, and is often used when a direct swap infeasible. Choosing crossover functions needs to make sure the new populations satisfy the constraints of the problems.

The typical mutation methods used in GAs consists of 1) bit-string mutation; 2) flip bit; 3) boundary; 4) uniform; and 5) Gaussian, as shown in Table 1. Bit-string mutation randomly flips the value of genes. Flip bit chooses genes from chromosomes to flip their values. Boundary replaces value of a gene with the upper or lower bound of the value. Uniform mutation replaces the score of the chosen chromosome with a uniform random value selected in the range of the user-specified bounds. Gaussian mutation adds a unit Gaussian distributed random value to the selected chromosome. The same as the crossover function, mutation function has to make sure the new populations satisfy the problem's constraints.

▼Table 1. The major mechanisms in initialization, selection, crossover, and mutation of GAs

Process	Mechanisms	Descriptions
Initialization	Binary encoding	A solution is a bit string with each element as 0 or 1
	Value encoding	A solution is a string with elements integers, real numbers, characters, or objects
	Permutation encoding	A solution is a sequence of number
	Tree encoding	A chromosome is a tree form of objects
Selection	Relative tournament	Two members are randomly chosen, and the parent is the one with higher fitness
	Routlette wheel	The probability an individual to be chosen as a parent is depended on their fitness
	Relative pooling tournament	Populations are thrown into a competition, and the winners are the parents
	Elitism	Populations with the higher fitness from all the populations generated so far from the parent set
Crossover	One-point	Randomly generate a crossover point
	Two-point	Randomly generate two crossover points
	Uniform	Use a fixed mixing ratio between two parents
	Cut and spice	Allow each parent to have its own choice in deciding crossover point
	Ordered chromosome	Switch the position of genes
Mutation	Bit-string	Randomly flip the value of genes
	Flip bit	Flip the value of selective genes
	Boundary	Replace the value of a gene with the upper or lower bound of the value
	Gaussian	Add a unit Gaussian distributed random value to the selected chromosome
	Uniform	Replace the value of a selective gene with a uniform random value within the user-specified bounds

3.1.2 Multi-Objective GAs

Multi-objective GAs (MOGAs) find out a Pareto optimal solution set for multi-objective optimization problems. An MOGA is also an iteration procedure consisting of the same flow chart as a single objective GA, but uses different mechanisms in section, crossover, and mutation to find out the Pareto optimal solution set and maintain the diversity of the set. The first MOGA is introduced by Carlos M. Fonseca and Peter J. Fleming in [32]. It uses a section sub-process where each solution is assigned a rank, the dominated solutions have the rank of 1 and the non-dominated solutions have a higher rank based on the distance between them and a dominated solution. N. Srinivas and Kalyanmoy Debin in [33] propose a MOGA named the none-dominated sorting genetic algorithm (NSGA) based on several layered classification of the solutions. Deb in [34], [35] extends this NSGA to the NSGA-II that initializes a population set, ranks and sorts each population according to none-domination level, creates new pool of offspring by applying crossover and mutation operations, and then combines the parents and offspring before partitioning the new combined pool into fronts. The NSGA-II conducts niching by adding a crowding distance to each member. It uses this crowding distance in its selection sub-process to maintain the member diversity and make sure each member stays a crowding distance apart. This keeps the population diverse and helps the algorithm to explore the fi tness landscape. NSGA-II has been the most widely used MOGA in current research.

3.2 PSO

PSO is proposed by James Kennedy and Russell Eberhart in [36] in 1995 and since then applying PSO to solve different complex optimization problems has been a hot research topic. PSO is inspired by the migrating birds to reach unknown destination. When birds migrating, each bird looks in a specific direction and finds out the migration route through identifying the bird in the best position. When applying PSO to solve an optimization problem, each solution of the problem is a "bird" or referred to a "particle". The population of solutions is a swarm of particles. Each particle has a velocity corresponding to its current place. Once a particle reaches a new position, the best position of each particle (the local best position) and the best position of the whole swarm of particles (the global best position) are updated. The new velocity of a particle corresponding on the new global best position and the local best position can be calculated.

The basic flow chart of a classical PSO algorithm is shown in **Fig. 4**. It is similar as GAs to have an initialization process where initial swarm of particles is randomly generated. Each particle has its velocity and position. Then the evaluation process is to evaluate the fitness of each particle. Each time a particle gets evaluated, its fitness value is compared to the fitness of the global best position and the local best position. If the fitness value is better than the fitness of current global or/and local best positions, the current global or/and local best positions are updated accordingly. The velocity and position of the whole swarm are updated to generate a new swarm for the next iteration until the stopping criterion is met.

The velocity of a particle is calculated using (1), where V^c and V^l are the current and last velocity respectively, P^l is the last position of a particle, L_{best} and G_{best} are the local best position of a particle and the global best position of the whole swarm, w, c_1, c_2 are parameters, and R_1 and R_2 are random variables ranging from 0 to 1. The formula calculating velocity of a particle represents the process that involves social interaction and intelligence so that each particle can learn from their own experience (the local best position) and also from the experience of other particles (the global best position). In (1), w represents the inertia of a particle, c_1 and R_1 represent how much experience needs to be learned from the local best positions, and c_2 and R_2 represent the experience learn from the global best positions. Then the new position of a particle is the sum of the current velocity calculated by (1) and the last position of the particle using (2).

$$V^c = wV^l + c_1 r_1\left(L_{best} - P^l\right) + c_2 r_2\left(G_{best} - P^l\right), \tag{1}$$

$$P^c = P^l + V^c. \tag{2}$$

3.2.1 Major Parameters of PSO

A PSO algorithm often has the following parameters: the number of particles, the w, c_1 and R_1, and c_2 and R_2. A particle in PSO is analogous to a population in GAs. However, PSO fixes the number of particles and adjusts the movement of each particle toward the destination through social behavior, while GAs randomly initialize the populations and then generate new populations for next evolution iteration. Since PSO does not need to sort the fitness of particles in any process, and the movement of each particle in a PSO is guided by the local best position and the global best position, PSO algorithms of-

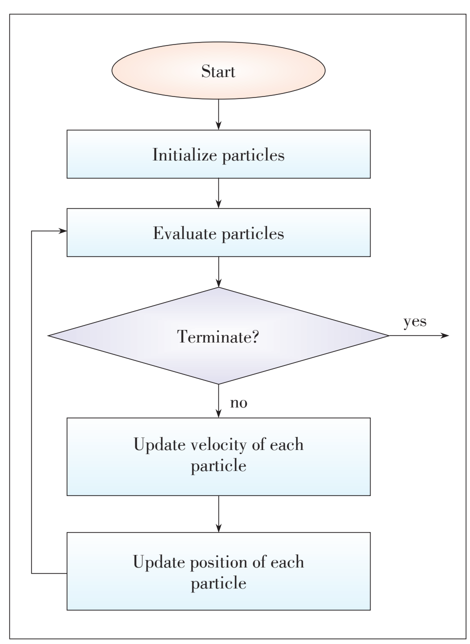


Figure 4. Flow chart of PSO.

ten have short convergence time.

3.2.2 Multi-Objective PSO

Classical PSO algorithms have no mechanisms to maintain the diversity of particles, they are often used to find the global optima for an optimization problem with single objective. However, great efforts have been made to extend classical PSO to solve multi-objective optimization problems. The typical mechanism used by PSO to maintain the diversity of solutions is to maintain two external archives: one for storing the leaders currently being used for performing the movement and the other for storing the final solutions. A crowding factor is used for the selection of non-dominance solutions from the solution archive to form a Pareto frontier for a multi-objective optimization problem [37]. Many more variable multi-objective PSO algorithms have been proposed [3], [38].

3.3 Ant Colony Optimization

ACO is a set of combinational optimization algorithms inspired by ants search for food and depositing pheromone on the route. When ants leave their nest to search for a food source and meet an obstacle, they randomly choose to the right or left directions to go forward. Then the ants in the direction with a shorter distance find a food source and carry the food back and deposit pheromone along their route. The following ants most likely choose the path with the larger amount of pheromone to find a food source. Since the more pheromone a route has, the more ants the route has been taken by, and the higher probability this route has shorter distance between the nest and a food source. Over time, this positive feedback process prompts all ants to choose the shorter path. The amount of pheromone on the ground influences the behavior of ants, the path with the largest amount of pheromone represents the shortest path between the nest and a source of food.

3.3.1 Ant System

The original ACO is the ant system (AS) proposed by Colorni, Dorigo, and Maniezzo in [39]–[41] and used to optimize the traveling salesman problem [42]. To apply AS to solve such a problem, one solution is represented as an ant, which is often a string. Each element of the string is a variable with a value within the given set. **Fig. 5** shows the flow chart of the AS.

AS often starts with generating m random ants and evaluate the fitness of each ant corresponding to an objective function, and then updates the pheromone concentration of each possible trail (variable value) using the following formula:

$$\tau_{ij}(t) = \rho \tau_{ij}(t-1) + \Delta\tau_{ij}, \tag{3}$$

where i is the variable of an ant; j is the option that the value of variable i choose; let l_{ij} be the value of variable i; T is the maximal number of iterations and t is a particular iteration; $\tau_{ij}(t)$ is the revised concentration of pheromone associated with l_{ij} at iteration t, $\tau_{ij}(t-1)$ is the concentration of pheromone at the previous iteration t-1; $\Delta\tau_{ij}$ is the change in pheromone concentration; and ρ is the pheromone evaporation coefficient with value ranging from 0 to 1 to avoid too strong influence of the old pheromone so that premature solution stagnation is incurred. $\Delta\tau_{ij}$ is the sum of the contributions of all ants associated with l_{ij} at iteration t, and can be calculated using (4):

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \tau_{ij}^{\ k}, \tag{4}$$

$$\Delta\tau_{ij}^{\ k} = \sum_{k=1}^{m} \begin{cases} \dfrac{R}{fitness^k} & if\ option\ l_{ij}\ is \text{ chosen by ant } k \\ 0 \text{ otherwise} \end{cases}, \tag{5}$$

where m is the number of ants and $\Delta\tau_{ij}^{\ k}$ is the pheromone concentrate laid on value l_{ij} by ant k. $\Delta\tau_{ij}^{\ k}$ can be calculated by (5) with R being the pheromone reward factor and $fitness^k$ being the value of the objective function for ant k.

Once the pheromone is updated, each ant has to update its route respecting the pheromone concentration and also some heuristic preference. The ant k at iteration t will change the value for each variable according to the following probability:

$$P_{ij}(k,t) = (\tau_{ij}(t)^{\alpha} \times \eta_{ij}^{\ \beta}) / (\sum_{l_{ij}}^{m} \tau_{ij}^{\ \alpha} \times \eta_{ij}^{\ \beta}), \tag{6}$$

where $P_{ij}(k,t)$ is the probability that option l_{ij} is chosen by ant k for variable i at iteration t; $\tau_{ij}(t)$ is the pheromone concentration associated with option l_{ij} at iteration t; η_{ij} is the heuristic factor for preferring among available options and is an indicator of how good it is for ant k to select option l_{ij}; and α and β are exponent parameters that specify the impact of trail and attractiveness, respectively, and take values greater than 0.

3.3.2 Ant Colony System

AS is the first ACO algorithm motivated by real ants and

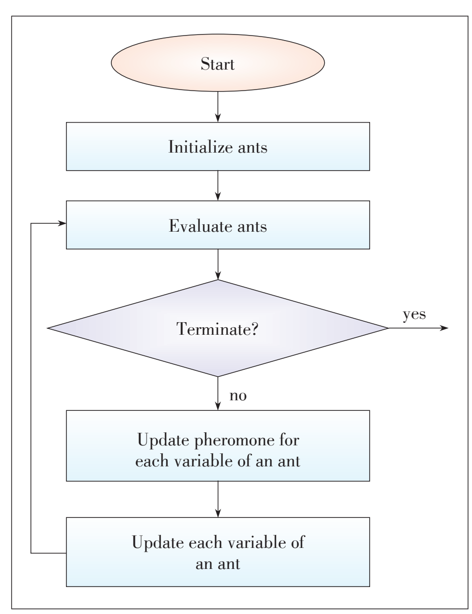


Figure 5. Flow chart of AS.

used to solve traveling salesman problem [42], but its performance cannot compete against the state-of-art heuristic algorithm. Later on, Gambardella and Dorego extend the AS algorithm to be Ant-Q algorithm [43] to improve its performance. The Ant-Q algorithm is further simplified as Ant Colony System (ACS) [44], which becomes the base of many ACO algorithms [45]–[47].

ACS has the same flow chart as AS. The major difference between them is that AS updates the pheromone of a trail using all the pheromone contributed by all the ants, and ACS updates the pheromone of a trail using the following formula,

$$\tau_{ij}(t)=\rho\tau_{ij}(t-1)+(1-\rho)\tau_0, \tag{7}$$

where ρ is still the pheromone evaporation coefficient with the value ranging from 0 to 1 (usually set to 0.9), and τ_0 is the initial pheromone value and can be defined as $\tau_0=(n\times L_{mn})^{-1}$, where L_{mn} is the tour length produced by execution of one ACS iteration without the pheromone concentrate.

The pheromone updating of AS is a globally updating and intends to increase the attractiveness of promising route, while the updating of ACS simplifies the search in a neighborhood of the best tour found so far of the algorithm, and it is local and more effective and can avoid long convergence time.

Once the route updating for each ant is done, each ant chooses its next move by choosing a random probability ρ (which is often fixed to 0.9). With the probability $(1-\rho)$ the next move is chosen randomly with a probability based on η_{ij}^{α} (α often equals to 1) and τ_{ij}^{β} (β often equals to 2), and with the probability ρ_0 the next move is chosen with a probability calculated by (5).

3.3.3 Major Parameters of ACO

The main parameters of a ACO algorithm include the number of ants m, number of iterations t, the attractiveness of a trail η_{ij}, the exponents α and β, the pheromone evaporation rate ρ, pheromone reward factor R, and the probability ρ_0.

3.3.4 Multi-Objective ACO

Angus and Woodward in [48] have reviewed a large collection of multi-objective ACO (MOACO) algorithms. The multi-objective ACO algorithms can be classified by how they make the choice of using single or multiple pheromones and pheromone update or decay, and by how these choices affect the performance of the algorithm as well. Some approaches implicitly or explicitly weight their multiple objectives in some kind of preferential order, and this approach can outperform the alternative pareto-based MOACO for some particular problems, but generating a Pareto optimal solution set so that a decision maker can make his choice based its own strategy is more general. The existing MOACO algorithms seeking the Pareto frontier may use single pheromone matrix or benefit from multiple pheromone matrix.

3.4 Simulated Annealing

Simulated annealing (SA) is a stochastic local search approach motivated by the behavior of physical systems in a heat bath. Local search is an approach that attempts to improve on a solution by a series of incremental and local changes. By given an initial solution, a local search algorithm defines a method that performs loops to find out the optimal solution in the neighborhood of the given initial solution, and expects that the local optima is the global optima. When a solid is put in a heat bath, it firstly raises the temperature of the solid to a point where its atoms can randomly move and then to lower the temperature to force the atoms to rearrange themselves into a crystallization state that minimizes energy of the system at a lower energy state. Carefully selecting the cooling schedule allows a solid to become a crystal that has the lowest energy instead of an amorphous state with higher energy. Since the solution of an optimization problem can be viewed as a solid in a heat bath, the cost of the objective function can be viewed as the energy of a solid, the optimal solution can be viewed as the ground energy state of a solid, moving a solution to a neighboring position can be viewed as the rapid quenching, and the control mechanism adopted by the search algorithm can be viewed as the cooling schedule. In this way, a simulated annealing algorithm can be developed to mimic the physical annealing process of physical material.

3.4.1 Simulated Annealing Algorithm

The SA algorithm is first proposed by Kirkpatrick, Gelatt, and Vecchi in [49]. The basic flow chart of the SA algorithm is shown in **Fig. 6**, where S is a given initial solution, S' is a randomly chosen neighbor of S in the initialization process. In the evaluation process, SA algorithms compare the difference of cost(S') and cost(S) (the cost function is similar as the fitness function in GAs, PSO, and ACO to evaluate how good a solution is in its solution space). In the new population generation process, the SA checks the cost difference between S' and S, generates new S, S', and T based on the cost difference between current S, S', and T.

3.4.2 Major Parameters of SA

The major parameters of a SA algorithm consist of the maximum number of iterations to apply the algorithm, the randomly generated neighbor solution, the temperature T, and the cooling ratio R ranging from 0 to 1.

3.4.3 Multi-Objective SA Algorithms

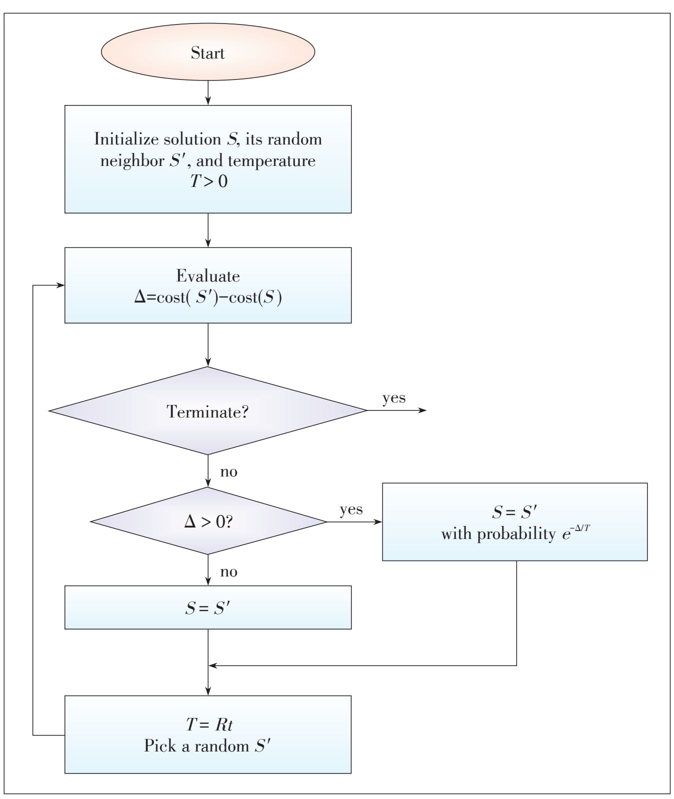
The key issue in extending SA to solve multi-objective optimization problems is to determine how to calculate the probability of accepting a dominated solution [50]. The initial approach introduced by Serafini in [51] proposes a target-vector approach. Given solutions S and S' randomly generated in the neighborhood of S, if S' is non-dominated, it is accepted as the next S and the non-dominated solution set is updated. Serafini's approach allows to maintain an archive of non-dominated solutions so that the generation of several members of the Pareto optimal set in a single run can be calculated, but only the local non-dominance is used to fill up the archive of solutions and a further filtering procedure is required to reduce the number of non-dominated solutions without sacrificing the diversity of the Pareto optimal solutions. Given S and the temperature T, $P(S',S,T)$ (the probability of accepting S' as a non-dominated solution) can be calculated as the following:

$$P(S',S,T)=Min\left(1,\ e^{Max_j}\left(\lambda_j\left(\mathrm{cost}_j(S)-\mathrm{cost}_j(S')\right)/T\right)\right), \quad (8)$$

where the weights λ_j are initialized to one and modified during the search process, and j is the particular iteration during the search process. Based on this approach, many more simulated annealing algorithms for multi-objective optimization problems are developed [52]–[54]. Simon in [3] and Solís in [55] give a comprehensive review on multi-objective SA algorithms.

3.5 Algorithm Summary

As a summary, the discussed 4 types of algorithms can be categorized into EAs with the same general components in Fig. 2 and the need to encode solutions and ensure the generated solutions satisfying the constraints. Each type of the algorithms has its way to initialize a population set. GAs, PSO, and ACO initialize their population set randomly, while AS has to choose an initial solution and then randomly generate another solution inside the neighborhood of the initial one. Each type of the algorithms needs to define a fitness function (cost function in SA) to evaluate the quality of the solutions. GAs evaluate the quality of each individual in the population set, rank and sort the populations according to their fitness number, choose the ones with the higher fitness to be parents for generating offspring; PSO and ACO simply evaluate the quality of each particle and ant, respectively; and SA evaluate the cost difference between the initial solution and its neighbor solution. To generate a new set of population, GA uses crossover and mutation operations; PSO calculates the velocity and updates the position of its particles; ACO updates the pheromone of each trail and computes the trail of each ant will take; and SA calculates the probability of accepting the neighbor solution as the new initial solution and updates new temperature for next iteration.



▲**Figure 6. Flow chart of SA algorithms.**

In general, GA is the only type of EAs that ranks the solutions, and hence the convergence time of GAs increases non-linearly as the population size grows. PSO uses floating point arithmetic to compute velocity and position of a particle. It can generate any potential values of velocity and may lead to high density of members in population set. Since the best positions of a particle and the whole swarm of particles guide the movement of a particle, PSO has short convergence time but may incur premature convergence. ACO is efficient for traveling salesman or similar problems. It can be used in dynamic applications or adapts to the changing environment. However, ACO has uncertain convergence time and it is hard to get theoretical analysis. SA can be used in many optimization problems. SA outperforms all the existing approximation algorithms in graph partitioning problem but suffers poor performance in number partitioning problem. SA algorithms have long convergence time and are not suitable for online optimization problems [56]. **Table 2** shows the major features of each described EA.

4 Applications of EAs in SDN

EAs have been applied in SDN in a wide variety of contexts. We have searched the major applications published since 2013, and categorized them into routing, load balancing, controller placement, security, virtual network mapping, flow entry optimization, and hybrid network migrating. We list the major applications in **Table 3** and discuss the representative applications in each category in the rest of this section.

4.1 Routing

The current routing strategy that forwards a packet along the shortest path between its source and destination cannot always achieve the shortest network delay in a highly dynamic network, because this shortest path may suffer heavy work load or get congested. With a logically centralized control plane that frequently updates the dynamical global network states, SDNs can develop many flexible routing strategies besides only using the shortest path. Since calculating a routing path is based on the huge volume of network statistics, finding out the optimal routing path in an acceptable time in a dynamical network environment is highly demanded. Therefore, an exhaustive algorithm is not time acceptable, and EAs are widely used in recently years to produce approximately optimal solutions in a short time. According to the objectives that a routing strategy wants to optimize, routing algorithms can be categorized to: 1) avoiding link congestion and balance link usage; 2) saving link power; and 3) maintaining QoS or QoE.

4.1.1 Avoidance of Link Congestion and Balancing of Link Usage

Liu in [57] presents a GA to solve the bandwidth-constrained multi-path optimization problem in a SDN. The proposed GA is implemented in a Floodlight [58] controller that manages a SDN network emulated by Mininet [59] for evaluation. The results indicate that the proposed GA can globally, flexibly, and effectively find out a routing path that minimizes the network delay under bandwidth constraint in a multi-path SDN network.

Ren in [60] develops a GA to solve a traffic scheduling problem in switch congestion control. It samples the link utilization ratio, and feeds it into the GA to optimize the switch traffic assignment to avoid link congestion. It also implements the GA in Floodlight controller over a Mininet emulated SDN with data flows generated by Iperf [61] for evaluation. The results show that the proposed GA is able to make the link arrangement utilization more balanced and reasonable.

Maniu in [62] applies GAs to compute the routing path for network flows so that the dynamic resource allocation can be optimized to enable a self-adaptive network with history extrapolation. The chromosome of the GAs is a sequence of nodes representing a routing path from source to destination. Each gene has a value representing the node ID in the network, and the length of the chromosome is the number of nodes in the routing path. Since the length of route is variable, the remaining locus in chromosome is completed with value 0. The Fitness function is a sum of link costs in a route. The evaluation shows the proposed GAs can provide an approximate optimal routing path to save the link cost efficiently.

Kikuta in [63] proposes an effective parallel GA to optimize explicit routing when using general purpose programming on graphic processing unit (GPU) in a SDN. The parallelism of GA consists of the search methods of GA itself, the calculation of each fitness function, and the evaluation of the network congestion ratio. It takes advantage of the multi-core processor for acceleration of graphic processing, and presents 10 times faster than the original parallel GA on GPU, and 9 times faster than the conventional CPU computation when enforcing explicit routing in a SDN.

Stefano in [64] introduces A4SDN for traffic engineering in software-defined networks. A4SDN is based on the alienated ant algorithm, a stochastic-based heuristic approach used to solve combinatorial and multi-constraint optimization problems. Based on artificial ant' behavior, the A4SDN forces the ants to distribute themselves over all the available paths rather than converge to a single one when searching for food. Using this strategy enables an autonomic dynamic routing and leads to a better exploitation of the network bandwidth for best effort data packets. A comparison between A4SDN with two Dijkstra-based shortest path routing solutions shows that A4SDN can guarantee a higher throughput together with a lower packet loss rate and network delay.

Wang in [65] adopts an ACO algorithm to route the network traffic to avoid the network congestion in traffic scheduling. The ACO algorithm is applied to dynamically adjust the calculation parameters of the routing algorithm. The proposed algorithm is compared to the traditional equal cost multi-path routing algorithm, and the results show that the ACO algorithm reduces link or node congestion and effectively improves the link utilization rate.

▼Table 2. Major features of GAs, PSO, ACO, and SA

	GA	PSO	ACO	SA
Motivation	Natural evolution	Bird migration	Ant search food	Solid in heat bath
Population size	Any size	Several	Any size	Typically two
Iterations	Yes	Yes	Yes	Yes
Single-objective	Yes	Yes	Yes	Yes
Multi-objective	Yes	Yes	Yes	Yes
Parallelism	Inherent parallelism	Extended	Inherent parallelism	Extended
Convergence	Slow	Fast	Uncertain	Slow
Solution quality	Near global optima	Near global optima	Near global optima	Near local optima
Applications	General	General	TSP, routing, dynamic and adaptive	General
Theoretical analysis	Hard	Hard	Hard	Hard

ACO: Ant Colony Optimization
GA: Genetic Algorithm
PSO: Particle Swarm Optimization
SA: Simulated Annealing
TSP: Traveling Salesman Protocol

4.1.2 Saving Link Power

The routing application can be optimized for power efficiency by routing flows to minimize the number of links activated. Most of the energy saving strategies on current IP network only aggregate traffic into some links, which leads to imbalance link utilization and seriously impacts the QoS.

Zhu in [66] takes advantage of the centralized control and global vision of a SDN to achieve the network energy saving and load balancing by dynamically aggregating and balancing the traffic while ensuring QoS. It adds actual QoS constraints to the basic maximum concurrent

▼**Table 3. Applications of EAs in SDNs**

Application	Category	EA	Description
Liu in [57]		GA	Avoid link congestion
Ren in [60]		GA	Avoid switch congestion
Maniu in [62]		GA	Optimize link usage
Kikuta in [63]		Parallel GA	Optimize explicit routing in GPU
Stefano in [64]		ACO	Optimize network bandwidth usage
Wang in [65]		ACO	Avoid link congestion
Zhu in [67]	Routing	PSO	Energy saving
Subbiah in [68]		PSO	Finding the best node connector in switches for energy saving
Awad in [69]		PSO	Energy saving under the constraint of size-limited flow table
Dobrijevic in [70]		ACO	QoE aware routing
Tang in [71]		ACO	QoE aware routing
Blaguer in [72]		GA	QoE aware routing
Santl M in[73]		ACO	QoE aware routing
Kang in [74]		GA	Controller load balancing
Chou in [75]		GA	Controller load balancing
AMR in [77]	Load balancing	GA	ink load balancing
Sathyanarayana in [76]		ACO	Controller and link load balancing
Lin in [72]		ACO	Controller load balancing
Lange in [79]		SA	Minimize swi-to-con delay and controller load imbalance
Sanner in [80]		GA	Minimize swit-to-con delay
Jalili in [81]		GA	Minimize con-to-con delay and controller load imbalance
Ahmadi in [82]	Controller placement	GA	Minimize swi-to-con delay, con-to-con delay, controller load imbalance
Gao in [83]		PSO	Minimize swi-to-con delay considering controller capacity
Liu in [84]		PSO	Minimize swi-to-con delay and controller load imbalance
Li in [86]		GA	Detect DDos attacks
Li in [87]		GA,PSO	Detect DDos attacks
Chen in [88]		ACO	Detect DDos attacks
Liu in [89]		ACO	Detect DDos attacks
Ojugo in [90]	Security	GA	Security rule generation
Zhao in [91]		GA	Intrusion action detecting
Bouet in [92]		GA	Single security appliance placement
Famaluddine in [93]		GA	Multiple security appliances placement
Li in [94]	Virtual network mapping	PSO	Optimize network resource usage
Yao in [95]		PSO	Optimize flow table usage
Gao in [96]	Flow table optimization	ACO	Optimize flow table usage
Li in [97]		ACO	Optimize flow table usage
Guo in [98]	Hybrid SDN migration	GA	Migrate routers in hybrid network

ACO: Ant Colony Optimization; DDoS: distributed denial of service; EA: evolutionary algorithm; GA: Genetic Algorithm; GPU: graphic processing unit; PSO: Particle Swarm Optimization; QoE: Quality of Experience; SA: Simulated Annealing

flow problem to formulate a multi-objective mixed integer programming model and proposes a multi-objective PSO algorithm called MOPSO to solve this NP-hard problem.

Subbiah in [67] proposes a PSO based energy aware routing algorithm at open virtual switches. It finds out the best node connectors in the switches from source host to destination host to reduce the path energy consumption. It improves the networking performance compared to the conventional one.

Awad in [68] considers a routing optimization problem for energy saving with a set of practical constraints consisting of the size-limited flow table and discrete link rate. It develops a low-complexity PSO based and power efficient routing heuristic algorithm to solve this problem. Performance evaluation results indicate that the proposed algorithm achieves more than 90% of the optimal network power consumption while requiring only 0.0045% to 0.9% of the computation time in real network topologies.

4.1.3 QoS and QoE

Dobrijevic in [69] applies an ACO approach to flow routing in SDN environments. Based on the global network view and the flexible configuration provided by a SDN, the approach estimates the QoE and seeks to optimize the user QoE for multimedia services. As different service has its QoE affected by network metrics such as packet loss and network delay differently, it proposes an ACO-based heuristic algorithm, based on the service type and its typical integral media flows, to calculate the best routing paths that aware QoE and conform to network limitations and traffic demands. The algorithm is integrated into an OpenDaylight controller, and the evaluation results indicate the proposed approach has promising QoE improvements and lower running time over shortest path routing.

Tang in [70] applies an improved ACO algorithm to the calculation of routes, meeting QoS requirements through obtaining the network topology, resources usage, and network statistics from the controller. If the bandwidth of a single path cannot meet the bandwidth requirement of an application, it can aggregate multiple paths and distribute the work load to these multiple paths to maintain the QoS of the application.

Blaguer in [71] applies a GA to find a routing path for a stream from a source to a target node in a SDN to maximize the concurrent streams without degrading the QoE. Since multimedia streams have to satisfy a certain maximum latency requirements and a minimal bandwidth requirement, finding the optimal routing path that satisfies the constraints costs time, especially in a large scale network. The GA offers an acceptable approximate optimal solution with short convergence time for a large scale of network.

Santl M in [72] applies ACO for QoE-centric flow routing in SDNs. It views a SDN as a weighted graph with a QoE measure as the ultimate metric. QoE depends on QoS and is expressed in terms of delay, jitter, and packet loss rate. Achieving good QoE often has to satisfy combination of multiple QoS metrics. It associates weights to the graph nodes based on values of delay and packet loss rate for each network device. The delay sums up delay of each node on a path, and a packet loss rate is calculated by $1 - \prod \in path(1 - lossRate(node))$. When applying ACO, it specifies an ant type for a flow type, and multiple ants of the same type are sent from the flow source to the destination in iterations, tracking estimated QoE and seeking to maximize the final result. The evaluation is conducted in a SDN emulated by Mininet with Floodlight controller, the proposed ACO indicates promising QoE improvements over shortest path routing as well as low convergence time.

4.2 Load Balancing

In SDN, load balancing can be categorized into two types: controller load balancing and link load balancing. Controller load balancing is a dynamical optimization problem and balances the load of controllers based on the real time changed network state. Link load balancing is often achieved by using ACO algorithms.

Kang in [73] and Chou in [74] propose similar load balancing strategies that use a way to monitor the load of each controller, once controller load imbalance is detected, a GA is applied to generate new switching assignment to balance the controller load.

AMR in [75] also applies a GA to balance the load of each controller corresponding to a set of workload. The performance of the proposed GA outperforms the random and round robin methods.

Sathyanarayana in [76] applies an ACO algorithm to select the best path to reach a controller with the least load to balance the load of both controllers and the paths leading to the controllers. The proposed approach is implemented as a load balancing module in the SDN controller. This module uses resource usage of the controller and the network statistics collected by the controller to find both the best server and the best path for network flows.

Lin in [77] proposes a dynamic load balancing approach based on an ACO algorithm with combined job classification in a layered control plane consisting of a root controller and multiple lower layer controllers. The root controller monitors the state of the whole network, and each lower layer controller manages a subset of network. Every job in the network is firstly sent to the root controller to decide which subset of network and which lower layer controller should take the job based on the job demand for CPU performance, then the job is sent to the corresponding lower layer controller to run the ACO algorithm to calculate the best routing path that minimizes the link load based on the dynamic network load provide by the root controller.

4.3 Controller Placement

Each new flow in a SDN suffers a flow setup delay since each switch in the data plane has to involve the control plane to setup new flows for them. This flow setup delay affects how fast a SDN can forward a new flow and how many new flows the control plane can set per second, and hence limits the network performance and scalability. Therefore, a large scale network or a geographically wide area network may need a logically centralized but physically distributed control plane with multiple controllers to provide a short flow setup delay anticipated by the network. This creates a controller placement problem that is firstly introduced by Heller in [78] to optimize the location of controllers so that the switch-to-controller delay can be minimized in a wide area of SDN. Later on the controller placement problem is extended to a multi-objective optimization problem to minimize the controller-to-controller delay, controller load imbalance, and many other objectives for both wide area of SDNs or data center SDNs.

An exhaustive algorithm can be used to find the global optimal solution in a small scale network with very limited number of controllers, but may not be time acceptable in a large scale network or as the number of controllers in the network increases. Heuristic algorithms have been proposed to solve a particular problem in a particular scenario, but EAs are more general and can solve general optimization problems with approximately optimal solutions in an acceptable computation time.

Lange in [79] proposes a SA algorithm to provide Pareto optimal controller placements to minimize switch-to-controller delay and controller load imbalance under a given maximal controller-to-controller delay. This application is the most representative sample that applies SA in SDN networks.

Sanner in [80] uses a GA to find out the controller placement so that the switch-to-controller latency can be minimized. It encodes a controller placement as a chromosome, but the encoding method is not clearly provided. It only optimizes one objective, and the performance of proposed GA is good comparing to Integer Linear Programmer.

Jalili in [81] applies a multi-object GA to find out the Pareto optimized controller placement so that the controller-to-controller delay and controller load imbalance can be optimized. In particular, it applies NSGA-II with each chromosome representing a controller placement and each gene of the chromosome representing the ID of node in the network. The evaluation is done on finding appropriate placements with 6 controllers for the Internet2 OS3E network topology.

Ahmadi in [82] develops a hybrid NSGA-II to solve the controller placement problem that minimizes the switch-to-controller delay, the controller-to-controller delay, and the controller load imbalance. Comparing to a typical NSGA-II, this hybrid NSGA-II improves the population initialization process by adding an improved local search to generate better solutions in its population set. This hybrid NSGA-II also develops a hybrid crossover function that enforces path-relink strategy and cross-controllers-operator.

Gao in [83] proposes a PSO to find out the optimal controller placement that minimizes the switch-to-controller delay with consideration of the controller capacity. It uses multiple particles, and each particle represents a controller placement consisting of k variables; each variable identifies the position of a controller in the network. The velocity and position of a particle are randomly generated and keep improved using the local best position of the particle and the global best position of the whole swarm of particles. The performance of the proposed PSO is compared to a greedy algorithm and an integer linear programming algorithm, and the results show the proposed PSO performs rapidly and effectively.

Liu in [84] implements a network clustering PSO algorithm to find the best controller placement to minimize the multi-objective optimization problem in SDNs. The proposed PSO takes consideration of the controller capacity, switch-to-controller delay, and the controller load balancing, and formulates an optimization problem that finds out the controller placement to maximize the utilization ratio of each controller and to minimize the switch-to-controller delay under the constraint of a maximal controller load imbalance. Each particle is encoded as a string consisting of n elements representing n switches in a network. The value of an element of the string indicates the position of a controller in the network. The proposed PSO combines a clustering mechanism to calculate the velocity and position of a particle. The evaluation is based on real topology and work load and shows the algorithm´s effectiveness.

4.4 Security

Internet service providers and equipment vendors are subject to many security threats. One of the most prevalent security threats is the distributed denial of service (DDoS) attack, where the attack traffic and attacker's IP address are respectively difficult to detect and trace, because attack traffic is similar to regular traffic and the attack is executed by multiple attackers. An intrusion detection system is a type of security software designed to automatically alert administrators when someone or something is to compromise information system through malicious activities or through security policy violations. Applying EAs in the security of SDN networks can be categorized into 1) detecting attack and 2) optimizing security appliance placement.

4.4.1 Detecting Distributed Denial-of-Service Attacks

EAs can be used in an intrusion detection system to detect the attack traffic. Based on the past behavior, a profile of normal behavior consisting of multiple attributes such as service types, flags, and logs can be created [85]. EAs can detect the unseen patterns and find out the malicious traffic based on this profile, but detecting a new attack is difficult.

Li in [86] proposes a cross validation-GA to enable a support vector machine classification with optimized punish parameter c and the kernel function parameter λ in DDoS attack detection. The proposed algorithm performs better than a typical support vector machine, a clustering model, and a BP neural network model.

Li in [87] combines BP, PSO, and GA to develop a particle swarm BP neural network algorithm for DDoS attack detection. The evaluation shows the proposed algorithm can achieve high detection rate with low miss report rate and convergence time.

Chen in [88] proposes a novel distributed DDoS attack detection and identification framework using an ACO based meta-heuristic approach for low-rate DDoS attacks. The proposed framework consists of three stages: a multi-agent algorithm, an information heuristic rule, and a search method. The proposed framework's time and space complexity is compared to the PSO and probabilistic packet marking. The evaluation shows the proposed framework can solve the problems in using other algorithms and demonstrates better performance than existing methods.

Liu in [89] proposes a random routing mutation based on an improved ACO algorithm to change the data transmission routing dynamically to avoid DDoS attack and improve network security. The ACO algorithm is used to find out the optimal routing path that has a minimal number of overlapping nodes compared to the recently used routing paths and meet the load balancing needs of the entire network.

Ojugo in [90] applies a GA to classify network audit data so that a better signatures for rule based intrusion detect system can be created. A chromosome is a rule consisting of 7-feats, each of which is a fixed length vector including one or more genes of different types. The goodness of chromosomes is evaluated. If the chromosome correctly classifies an attack, it is considered good; otherwise it is bad and not selected for crossover to produce offspring. Therefore, the more attacks a chromosome detects, the higher fitness value. Applying GA to generate the security classification rule reduces security experts from rule creation. The rule generation can speed up and counter new attacks.

Zhao in [91] applies a clustering GA to solve the intrusion detection problem. The proposed clustering GA algorithm consists of a clustering step and a genetic optimization step. It does not only automatically cluster cases, but also can detect unknown intrusion actions. Its overall accuracy can reach up to 95% with a very low false alarm rate.

4.4.2 Security Appliance Placement

Current IP networks need large-scale adoption of security appliances to improve the whole network security and to protect the information privacy. Security appliances can be virtualized and dynamically deployed as pieces of software on commodity hardware. Deploying such software security appliances is costly in terms of license fees and power consumption. Designing cost effective security appliance deployment strategies that meet the security operational constraints is thus mandatory for the adoption of this approach.

Bouet in [92] introduces a security appliance deployment problem that minimizes the number of security appliance and the network load under the management constraints such as the maximal number of security appliance engines, the maximal usable link bandwidth, and the maximal unallocated flows. This optimization problem is a multi-objective optimization with conflicted objectives. Reducing the number of security appliances tends to potentially increase the distance of the paths between the source of a flow and a security appliance, and the usage of link bandwidth along the paths as well. Minimizing the link bandwidth usage increases the number of security appliances to be deployed. Bouet only deploys one security appliance and develops a GA to solve this problem, while Famaluddine in [93] extends this problem to multiple security appliances and solve the same security appliance deployment problem that minimizes the network load and the number of appliances using GAs.

4.5 Virtual Network Mapping

Multi-tenant cloud network needs to map a tenant virtual network onto the physical network substrate to provide resource and information isolation among tenants. To maximize the resource usage without degrading the service level of a tenant virtual network, a virtual network mapping problem can be formulated to maximize the network resource usage under the constraints of tenant virtual network service requirements, network resource limitation, management limitation, and more. Li in [94] proposes a virtual network mapping model based on a PSO algorithm. It proposes a virtual network framework, where each tenant virtual network is controlled by tenant's own controller. The proposed PSO algorithm has better performance than a shortest path algorithm in improving the utilization of network bandwidth.

4.6 Flow Table Optimization

SDNs need to involve their control plane to generate forwarding rules at switches according to the management policy. Network-wide optimization policy often has to be enforced by injecting many local forwarding rules at corresponding switches. However, the tenary content addressable memory (TCAM) used to store the forwarding rules in SDN switches is limited resources. This creates a problem of optimizing the flow table usage without affecting the network-wide management.

Yao in [95] investigates this problem and formulates it as a bounded forwarding-rules maximum flow (BFR-MF) problem, and solves it by applying an improved PSO algorithm. This improved PSO keeps updating the position of particles to maximize the overall feasible traffic. The fairness among flows is maintained to guarantee the QoS requirements of flows. Extensive simulations show that the improved PSO algorithm performs well in optimizing network utilization.

Gao in [96] formulates this problem as an mixed integer linear programing problem that optimizes the TCAM resources under the QoS constraint of flows in a multiple uni-cast session SDN. This problem is actually a routing rule space occupation problem that finds out the best switches to store routing rule without sacrificing the QoS of flows. An ACO algorithm is applied to solve this problem and demonstrates an expected performance in evaluation.

Li in [97] formulates this problem as a BFR-MF problem as similar as Yao in [95], but applies an improved ACO algorithm to optimize the flow table usage with the performance and the level of QoS of flows guaranteed. The simulation indicates that the proposed ACO performs better network utilization under the constraints of QoS in data center networks.

4.7 Hybrid SDN Migration

Though some companies have moved their inter-connecting data centers to fully SDN-enabled networks, many more companies have to look for an incremental deployment of SDN devices in its network due to the existing economical, technical, and organizational challenges. This implies that over a long period of time, a hybrid network consisting of the conventional IP network devices and Openflow enabled network devices is existed. Since the same network protocols can run over a conventional network and a SDN, there is no big technical problem in forming and running such a hybrid network. However, it creates an optimization problem that migrates legacy routers to SDN-enabled routers to maximize the network usage with minimized investment.

Guo in [98] investigates this problem and formulates it as an optimization problem that finds out the optimal migration sequence of routers under the consideration of the traffic engineering performance and the investment. It tries to utilize the potential of network resources in traffic engineering to reduce routers that need to be migrated, and to avoid investing more budgets to migrate more routers that may make little gain for traffic engineering. It applies GA to calculate the migration sequence that minimizes the link usage. The GA uses the permutation of routers as its chromosome, and the fitness function is the sum of the link usage. Once a router is migrated it remains unchanged in the following migration iterations for network stability. The GA approach obtains a better performance compared to a static algorithm or a greedy algorithm in migration.

5 Issues and Challenges

5.1 Requirements of Applying EAs

EAs can theoretically solve any optimization problems with two requirements to be met: 1) encoding candidate solutions and 2) generating fitness function to evaluate solutions. EAs require the solutions of problems to be represented as some kind of expression, for example, the chromosome of a GA, the particle of a PSO, the ant of an ant colony optimization, or the solid of a simulated annealing. These encodings are often represented as a string, each element of which represents a variable with bounded value set for the given problem. The number of the elements in the string represents the number of variables in the given problem and the bound value set of each variable represents the constraints for each variable. Applying EAs also needs a way to evaluate solutions—the fitness function. There is no way to evaluate the rightness of a fitness function. However, the fitness score is necessary for indicating how good a candidate solution is. As long as the encoding scheme and fitness function are developed, there is really no restriction on the types of problems that EAs can solve.

5.2 Tasks That Fit EAs

EAs can be used to solve combinatorial optimization problems and have been used to solve them in a big range of fields, for instance, traveling salesman problems, job shop scheduling problems, vehicle routing problems, multi-commodity distribution network design problems, multi - mode resource constrained project scheduling problems, warehouse design problems, and many other more. GAs and SA algorithms are very general, and can be used in a distributed or paralleled manner to solve complex optimization problems with large search space and multiple conflicted objectives. PSO is mainly used to solve unconstrained, single-objective optimization problems though many mechanisms have been developed to allow a PSO algorithm to support constrained problems or maintain the diversity of solutions for multi-objective optimization problems. Unlike GAs and SA algorithms can optimize problems with variables representing nodes and links of a graph, ACO is typically used for problems that optimize links of a graph, such as travelling salesman problem and network routing problems, and solve them in a dynamical manner. ACO algorithms often run continuously and can adapt to the real time changed environments. The improved ACO can also tackle a problem with unknown features, which makes ACO can be used in data mining, data analysis, and classifying malicious flows and detecting attacks in security problems.

5.3 Implementation Issues

The discussed four EAs have the same three major processes in general as shown in Fig. 2. Therefore, when applying these EAs to solve optimization problems, the first important task is to determine how to represent the solution of the problem. Since GA provides a large flexibility to encode its chromosome, it is important to choose the encoding with less number of genes to reduce search space and easy to enforce constraints. The second task in the implementation of EAs is to maintain the diversity of the populations so that the solution space can be explored globally and near globally optimized solutions can be found. Lack of diversity in the populations often incurs a permature converged solution and results in a local optimal solution rather than a global optimal one. The re-initialization strategy that adds new randomly generated populations to the current population set can avoid this premature convergence process that results in a local optimal solution and is highly recommended in PSO as well as GAs. The third task in implementing EAs is to improve the solution accuracy. Since EAs are approximately algorithms, we expect the solutions generated by them as close optimal as possible. Typically, increasing the size of population set of an EA can allow it to search wider solution space, but it is only suitable for GAs with a flexible population size that can be adjusted according to the solution space of a problem. However, since the density of the populations in the solution space is very little, it is often found that EAs cannot produce high quality solutions with high accuracy to the real global optima. Hybrid EAs can be a fix.

5.4 Open Issues and Future Work

EAs are very difficult to do theoretical analysis. Though EAs can be applied in many optimization problems, and have been shown that their solutions perform better than some other heuristic algorithms in particular circumstance, we cannot expect the same EAs can perform better than the heuristic algorithms on some problems outside the circumstance. One set of populations may bring solutions better than another set of populations, but we do not know why and how. An enhanced theoretical understanding of EAs are needed to increase user's trust in further widely applying EAs in various fields.

Most of EAs are used to tackle a problem with a solution string with up to tens of genes, and we have never seen an EA used to solve an optimization problem with very long solution string in practice. However, as SDN technology used in wireless sensor network, Internet of thing, and 5G mobile networks, the scale of a SDN based network becomes larger and larger. The number of network devices in such networks can increase to thousands or millions level. Calculating the routing path or optimizing the placement of some NFVs over the whole network or a subset of network may have to generate a solution string with hundreds of genes or even more. Solving such EAs with long solution strings and huge solution space creates many practical problems in algorithm designing, distributing and parallelism, and performance tuning.

Since different EAs have their own advantages and disadvantages, a single EA can hardly obtain a good enough solution. PSO is easy to be implemented and has a short convergence time, but suffers from partial optimism incurred by the lack of regulation in particle speed and direction. ACO is efficient for some particular types of optimization problems such as routing and link usage optimization, but suffers from uncertain convergence time and is really hard to get theoretical analysis. GA is inherent parallelism and good at maintaining diversity, but suffers from low accuracy and uncertain convergence time. Many efforts have been put on combining two or three of EAs together to 1) reduce the convergence time, 2) improve the quality of solutions , and 3) incorporate EAs as part of a larger system.

6 Conclusions

EAs are stochastic algorithms inspired by the natural biological evolution and social behavior of species. They typically have the same flow charts consisting of three major components: population initialization, fitness evaluation, and new population generation; but adopt various mechanisms in each component. Although each discussed EA has its own advantage and disadvantage, EAs are general and versatile, and can be used for complex combinatorial problems in a wide variety of circumstances.

With the logically centralized global network vision, SDN's control plane makes a perfect place to develop and deploy network-wide management and optimization solutions, such as optimizing a routing path to avoid network convergence and balance link load, to maintain QoS and QoE, or to reduce link energy consumption; optimising the controller placement for distributed controllers; detecting the DDoS attacks or optimizing the security appliance placement; and optimizing flow tables usage, virtual network mapping, and router migration in a hybrid SDN. Such network-wide management and optimization problems are often complex. EAs can be applied to find out the near optimal solutions for them in an acceptable time.

There are basically two requirements that determine the feasibility of applying an EA to solve an optimization problem: 1) encoding a solution and 2) evaluating the quality of a solution. Any problems that meet these requirements can be solved by EAs. Many practical issues and open issues have been raised when applying EAs to slove such problems, especially regarding their applications in SDNs: theoretical analysis, dealing with the optimization problem for huge scale of SDN networks, and developing hybrid EAs to further improve the solution quality and convergence time.

References

[1] ONF, "Software-defined networking: the new norm for networks," Open Networking Foundation white paper, Apr. 13, 2012, retrieved Aug. 22, 2014.
[2] OpenFlow Switch Consortium. (2009). *OpenFlow Switch Specification Version 1.0. 0* [Online]. Available: http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf
[3] Dan Simon, *Evolutionary Optimization Algorithms*. Hoboken, USA: John Wiley & Sons, 2013.
[4] Daniel W. Dyer, Evolutionary computation in java, a practical guide to the watchmaker framework [Online]. Available: http://watchmaker.uncommons.org/manual/index.html
[5] U. Mehboob, J. Qadir, S. Ali, and A. Vasilakos, "Genetic algorithms in wireless networking: techniques, applications, and issues," *Soft Computing*, vol. 20, no. 6, pp. 2467–2501, 2016.
[6] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014. doi: 10.1109/SURV.2014.012214.00180.
[7] D. Kreutz, F. M. Ramos, P. E. Verissimo, et al., "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015. doi: 10.1109/JPROC.2014.2371999.
[8] T. G Robertazzi, "Software-defined networking," in *Introduction to Computer Networking*. New York City, USA: Springer, 2017, pp. 81–87.
[9] M. Latah and L.Toker, "Application of artificial intelligence to software defined networking: a survey," *Indian Journal of Science and Technology*, vol. 9, no. 44, 2016. doi: 10.17485/ijst/2016/v9i44/89812.
[10] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, no. 1, pp. 43–53, 2005. doi: 10.1016/j.aei.2005.01.004.
[11] M. Crepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: a survey," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, article no. 35, 2013. doi: 10.1145/2480741.2480752.
[12] C. v̌on Lücken, B. Barán, and C. Brizuela, "A survey on multi-objective evolutionary algorithms for many-objective problems," *Computational Optimization and Applications*, vol. 58, no. 3, pp. 707–756, 2014. doi: 10.1007/s10589-014-9644-1.
[13] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, et al., "A survey of multiobjective evolutionary algorithms for data mining: part I," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 4–19, 2014. doi: 10.1109/TEVC.2013.2290086.
[14] H. R. Cheshmehgaz, H. Haron, and A. Sharifi, "The review of multiple evolutionary searches and multi-objective evolutionary algorithms," *Artificial Intelligence Review*, vol. 43, no. 3, pp. 311–343, 2015. doi: 10.1007/s10462-012-9378-3.
[15] P. Larrañaga, H. Karshenas, C. Bielza, et al., "A review on evolutionary algorithms in Bayesian network learning and inference tasks," *Information Sciences*, vol. 233, pp. 109–125, 2013. doi: 10.1016/j.ins.2012.12.051.
[16] Y. J. Gong, W. N. Chen, Z. H. Zhan, el al., "Distributed evolutionary algorithms and their models: a survey of the state-of-the-art," *Applied Soft Computing*, vol. 34 , pp. 286–300, 2015. doi: 10.1016/j.asoc.2015.04.061.
[17] R. C. Purshouse, K. Deb, M. M. Mansor, et al., "A review of hybrid evolutionary multiple criteria decision making methods," in *IEEE Congress on Evolutionary Computation (CEC)*, Beijing, China, 2014. doi: 10.1109/CEC.2014.6900368.
[18] D. J. Munk, G. A. Vio, and G. P. Steven, "Topology and shape optimization methods using evolutionary algorithms: a review," *Structural and Multidisciplinary Optimization*, vol. 52, no. 3, pp. 613–631, 2015. doi: 10.1007/s00158-015-1261-9.
[19] B. Kazimipour, X. Li, and A. K. Qin, "A review of population initialization techniques for evolutionary algorithms," in *IEEE Congress on Evolutionary Computation (CEC)*, Beijing, China, 2014. doi: 10.1109/CEC.2014.6900618.
[20] G. Karafotias, M. Hoogendoorn, and Á. E. Eiben, "Parameter control in evolutionary algorithms: trends and challenges," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 167– 187, 2015. doi: 10.1109/TEVC.2014.2308294.
[21] Q. Yan, F. R. Yu, Q. Gong, et al., "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 602– 622, 2016. doi: 10.1109/COMST.2015.2487361.
[22] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," *IEEE Communications Surveys & Tutorials*,18.1, pp. 623-654, 2016. doi: 10.1109/COMST.2015.2453114.
[23] H. Farhady, H. Y. Lee, and A. Nakao, "Software-defined networking: a survey," *Computer Networks*, vol. 81, pp.79– 95, 2015. doi: 10.1016/j.comnet.2015.02.014.
[24] A. Blenk, A. Basta, M. Reisslein, et al., "Survey on network virtualization hypervisors for software defined networking," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 655– 685, 2016. doi: 10.1109/COMST.2015.2489183.
[25] R. Jain and S. Paul, "Network virtualization and software defined networking for cloud computing: a survey," *IEEE Communications Magazine*, vol. 51, no. 11, pp. 24–31, 2013. doi: 10.1109/COMST.2015.2489183.
[26] X. N. Nguyen, D. Saucez, C. Barakat, et al., "Rules placement problem in openflow networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp.1273–1286, 2016. doi: 10.1109/COMST.2015.2506984.
[27] N. A. Jagadeesan and B. Krishnamachari, "Software-defined networking paradigms in wireless networks: a survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, article no. 27, 2015. doi: 10.1145/2655690.
[28] T. Chen, M. Matinmikko, X. Chen, et al., "Software defined mobile networks: concept, survey, and research directions," *IEEE Communications Magazine* ,

vol. 53, no. 11, pp.126–133, 2015. doi: 10.1109/MCOM.2015.7321981.
[29] D. Gante, A, M. Aslan, and A. Matrawy, "Smart wireless sensor network management based on software-defined networking," in *27th Biennial Symposium on Communications (QBSC)*, Kingston, Canada, 2014. doi: 10.1109/QBSC.2014.6841187.
[30] Y. Jararweh, M. Al-Ayyoub, E. Benkhelifa, et al., "Software defined cloud: survey, system and evaluation," *Future Generation Computer Systems*, vol. 58, pp. 56–74, 2016. doi: 10.1016/j.future.2015.10.015.
[31] J. H. Holland, *Adaptation in Natural and A rtificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Cambridge, USA: MIT Press, 1992.
[32] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization," in *Proc. Fifth International Conference on Genetic Algorithms*, San Mateo, California, pp.416–423, 1993.
[33] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.
[34] K. Deb, S. Agrawal, A. Pratab, et al., "A fast elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Proc. Parallel Problem Solving from Nature VI Conference*, Paris, France, pp. 849–858, 2000.
[35] K. Deb, A. Pratap, S. Agarwal, et al., "A fast and elitist multiobjective genetic algorithm: NSGA–II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002. doi: 10.1109/4235.996017.
[36] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, Perth, Australia, 1995, pp, 1942–1948.
[37] M. R. Sierra and C. A. Coello Coello, "Improving PSO-Based multiobjective optimization using crowding, mutation and ∈-dominance," in *Third International Conference on Evolutionary Multi-Criterion Optimization.*, Guanajuato, Mexico, 2005, pp. 505–519.
[38] S. Lalwani, S. Singhal, R. Kumar, et al., "A comprehensive survey: applications of multi-objective particle swarm optimization (MOPSO) algorithm," *Transactions on Combinatorics*, vol. 2, no. 1, pp.39–101, 2013.
[39] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life.* Cambridge, USA: MIT Press, 1992.
[40] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: an autocatalytic optimizing process," Politecnico di Milano, Tech. Rep. 91-016 revised, 1991.
[41] M. Dorigo, "Optimization, learning and natural algorithms," Ph. D. Thesis, Politecnico di Milano, Italy, 1992.
[42] L. M. Gambardella and M. Dorigo, "Solving symmetric and asymmetric TSPs by ant colonies," in *Proc. IEEE International Conference on Evolutionary Computation*, Nagoya, Japan, 1996. doi: 10.1109/ICEC.1996.542672.
[43] M. Dorigo and L. M. Gambardella, "Ant-Q: a reinforcement learning approach to the traveling salesman problem," in *Proc. 12th International Conference on Machine Learning*, Morgan Kaufmann, USA, 2016, pp. 252–260.
[44] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997. doi: 10.1109/4235.585892.
[45] B. Bullnheimer, R. F. Hart, and C. Strauss, "A new rank based version of the ant system—a computational study," *Central European Journal for Operations Research and Economics*, vol. 7, no. 1, pp. 25–38,1997.
[46] T. Stutzle and H. H. Hoos, "Max-Min ant system," *Future Generation Computer System*, vol. 16, pp. 889–914, 2000
[47] M. Dorigo and T. Stutzle, *Ant Colony Optimization*. Cambridge, USA: MIT Press, 2004
[48] D. Angus and C. Woodward, "Multiple objective ant colony optimization," *Swarm Intelligence*, vol. 3, no. 1, pp. 69–85, 2009. doi: 10.1007/s11721-008-0022-4.
[49] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
[50] P. Czyzzak and A. Jaszkiewicz, "Pareto simulated annealing - A metaheuristic technique for multiple-objective combinatorial optimization," *Journal of Multi-Criteria Decision Analysis*, vol. 7, no. 1, pp. 34-47, 1998
[51] P. Serafini, "Simulated annealing for multiple objective optimization problems," in *Proc. 10th International Conference on Multiple Criteria Decision Making: Expand and Enrich the Domains of Thinking and Application*, Berlin, Germany, vol. 1, pp. 283–292, 1994.
[52] M. P. Hansen, "Generating a diversity of good solutions to a practical combinatorial problem using vectorized simulated annealing," Institute of Mathematical Modelling, Technical University of Denmark, Tech. Rep., 1997.
[53] T. Ray, R. Gokarn, and O. Sha, "A global optimization model for ship design," *Computers in Industry*, vol. 26, no. 2, pp. 175–192, 1995. doi: 10.1016/0166-3615(95)00003-M.
[54] A. J. Ruiz-Torres, E. E. Enscore, and R. R. Barton, "Simulated annealing heuristics for the average flow-time and the number of tardy jobs bicriteria identical parallel machine problem," *Computers and Industrial Engineering*, vol. 33, no. 1–2, pp. 257–260, 1997. doi: 10.1016/S0360-8352(97)00087-9.
[55] J. F. Solís, H. J. Fraire, J. C. Soto-Monterrubio, et al., "Multi-objective simulated annealing algorithms for general problems," in *Handbook of Research on Military, Aeronautical, and Maritime Logistics and Operations*. Hershey, USA: IGI Global, 2016, pp. 280–292.
[56] E. Aarts, J. Korst, and W. Michiels, "Simulated annealing," in *Search Methodologies*, E. K. Burke and J. Korst, Ed. Boston, USA: Springer, 2014, pp. 265–285.
[57] Y. Liu, Y. Pan, M. Wang, et al., "The multi-path routing problem in the software defined network," in *11th International Conference on Natural Computation (ICNC)*, Zhangjiajie, China, 2015. doi: 10.1109/ICNC.2015.7377999.
[58] Project Floodlight. (2017). *Floodlight* [Online]. Available: http://floodlight.openflowhub.org
[59] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *Proc. 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Monterey, USA, 2010, Article No. 19. doi: 10.1145/1868447.1868466.
[60] H. Ren, X. Li, J. Geng, et al., "A SDN-based dynamic traffic scheduling algorithm," in *IEEE International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, Chengdu, China, 2016. doi: 10.1109/CyberC.2016.103.
[61] J. Dugan. (2017). *IPerf—The Network Bandwidth Measurement Tool* [Online]. Available: https://iperf. fr
[62] R. Maniu and L. A. Dumitru, "Exploring the possibilities of a self-regulating SDN controller," *Scientific Bulletin "Mircea cel Batran" Naval Academy*, vol. 18, no. 1 , pp. 58–61, 2015.
[63] K. Kikuta, E. Oki, N. Yamanaka, et al., "Effective parallel algorithm for GPGPU-accelerated explicit routing optimization," in *IEEE Global Communications Conference (GLOBECOM)*, San Diego, USA, 2015, pp. 1–6.
[64] A, D. Stefano, G. Cammarata, G. Morana, et al., "A4SDN-adaptive alienated ant algorithm for software-defined networking," in *10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, Krakow, Poland, 2015.
[65] Y. L. Wang, K. J. Yuan, W. Fang, et al., "Research of a SDN traffic scheduling technology based on ant colony algorithm," in *2016 International Conference on Information Engineering and Communications Technology*. Lancaster, USA: DEStech Publications, Inc., 2016. doi: 10.12783/dtetr/iect2016/3754.
[66] R. Zhu, H. Wang, Y. Gao, et al., "Energy saving and load balancing for SDN based on multi-objective particle swarm optimization," in *15th International Conference on Algorithms and Architectures for Parallel Processing*, Zhangjiajie, China, 2015, pp. 176–189. doi: 10.1007/978-3-319-27137-8_14.
[67] S. Subbiah and V. Perumal, "Energy-aware network resource allocation in SDN," in *IEEE International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, India, 2016. doi: 10.1109/WiSPNET.2016.7566506.
[68] M. K. Awad, M. El-Shafei, T. Dimitriou, et al., "Power-efficient routing for SDN with discrete link rates and size-limited flow tables: a tree-based particle swarm optimization approach," *International Journal of Network Management* , vol. 27, no. 5, 2017. doi: 10.1002/nem.1972.
[69] O. Dobrijevic, M. Santl, and M. Matijasevic, "Ant colony optimization for QoE-centric flow routing in software-defined networks," in *11th IEEE International Conference on Network and Service Management (CNSM)*, Barcelona, Spain, 2015. doi: 10.1109/CNSM.2015.7367371.
[70] Y. Tang, W.Z. Wang, S. N. Dong, et al., "An Improved Ant Colony Algorithm for Routing in Software Defined Networking," *Journal of Computational and Theoretical Nanoscience*, vol. 13, no. 1, pp. 438–442, 2016. doi: 10.1166/jctn.2016.4824.
[71] R. Balaguer. (2017). *Flow embedding algorithms for software defined audio networks* [Online]. Available: http://ftp.tik.ee.ethz.ch/pub/.../MA-2014-14.pdf
[72] O. Dobrijevic, M. Santl, and M. Matijasevic, "Ant colony optimization for QoE-centric flow routing in software-defined networks," in *11th International Conference on Network and Service Management (CNSM)*, Barcelona, Spain, 2015, pp. 274–278.
[73] S. B. Kang and G. I. Kwon, "Load balancing of software-defined network controller using genetic algorithm," *Contemporary Engineering Sciences*, vol. 9, no. 18, pp. 881–888, 2016. doi: 10.12988/ces.2016.66105.
[74] L. D. Chou, Y. T. Yang, Y. M. Hong, et al., "A genetic-based load balancing algorithm in openflow network," in *Advanced Technologies, Embedded and Multi-*

media for Human-Centric Computing, Y.-M. Huang, H.-C. Chao, D.-J. Deng, and J. H. Park Ed. Dordrecht, Netherlands: Springer, 2014, pp. 411–417.
[75] A. M. R. Ruelas and C. E. Rothenberg. (2015). *Implementation of neural switch using OpenFlow as load balancing method in data center* [Online]. Available: https://www.fee.unicamp.br/sites/default/files/departamentos/dca/eadca/eadca - viii/artigos/sessao8/ruelas_rothenberg.pdf
[76] S. Sathyanarayana and M. Moh, "Joint route-server load balancing in software defined networks using ant colony optimization," in *IEEE International Conference on High Performance Computing & Simulation (HPCS)*, Innsbruck, Austria, 2016. doi: 10.1109/HPCSim.2016.7568330.
[77] W. Lin and L. Zhang, "The load balancing research of SDN based on ant colony algorithm with job classification," in *2nd Workshop on Advanced Research and Technology in Industry Applications (WARTIA 2016)*, Dalian, China, 2016.
[78] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proc. 1st Workshop on Hot Topics in Software Defined Networks*, Helsinki, Finland, 2012, pp. 7–12. doi: 10.1145/2342441.2342444.
[79] S. Lange, S. Gebert, T. Zinner, et al., "Heuristic approaches to the controller placement problem in large scale SDN networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 1, pp. 4–17, 2015. doi: 10.1109/TNSM.2015.2402432.
[80] J. M. Sanner, M. Ouzzif, Y. Hadjadj-Aoul, et al. (2016, Nov.). *Evolutionary algorithms for optimized SDN controllers & NVFs' placement in SDN networks* [Online]. Available: https://hal.inria.fr/hal-01421799
[81] A. Jalili, V. Ahmadi, M. Keshtgari, et al., "Controller placement in software-defined WAN using multi objective genetic algorithm," in *2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, Iran, 2015. doi: 10.1109/KBEI.2015.7436121.
[82] V. Ahmadi, A. Jalili, S. M. Khorramizadeh, et al., "A hybrid NSGA-II for solving multiobjective controller placement in SDN," in *2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, Iran, 2015. doi: 10.1109/KBEI.2015.7436122.
[83] C. Gao, H. Wang, F. Zhu, et al., "A particle swarm optimization algorithm for controller placement problem in software defined network," in *15th International Conference on Algorithms and Architectures for Parallel Processing*, Zhangjiajie, China, 2015, pp. 44–54. doi: https://doi.org/10.1007/978-3-319-27137-8_4.
[84] S. Liu, H. Wang, S. Yi, et al., "NCPSO: a solution of the controller placement problem in software defined networks," in *15th International Conference on Algorithms and Architectures for Parallel Processing*, Zhangjiajie, China, 2015, pp. 213–225. doi: https://doi.org/10.1007/978-3-319-27137-8_17.
[85] J. Ashraf and S. Latif, "Handling intrusion and DDoS attacks in software defined networks using machine learning techniques," in *National Software Engineering Conference (NSEC)*, Rawalpindi, Pakistan, 2014. doi: 10.1109/NSEC.2014.6998241.
[86] X. Li, D. Yuan, H. Hu, et al., "DDoS detection in SDN switches using support vector machine classifier," in *2015 Joint International Mechanical, Electronic and Information Technology Conference (JIMET-15)*, Chongqing, China, 2015.
[87] F. Li, "Application of particle swarm BP neural network in DDoS attack detection," *Microcomputer & Its Applications*, no. 3, 2014. doi: 10.3969/j.issn.1674-7720.2014.03.016.
[88] H. H. Chen and S. K. Huang, "LDDoS attack detection by using ant colony optimization algorithms," *Journal of Information Science & Engineering*, vol. 32, no. 4, pp. 995–1020, 2016.
[89] J. Liu, H. Zhang, and Z. Guo, "A defense mechanism of random routing mutation in SDN," *IEICE Transactions on Information and Systems*, vol. 100, no. 5, pp. 1046–1054, 2017. doi: 10.1587/transinf.2016EDP7377.
[90] A. A. Ojugo, A.O. Eboka, O.E. Okonta, et al, "Genetic algorithm rule-based intrusion detection system (GAIDS)," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 3, no. 8, pp. 1182–1194, Aug. 2012
[91] J. L. Zhao, J. F. Zhao, and J. J. Li, "Intrusion detection based on clustering genetic algorithm," in *International Conference on Machine Learning and Cybernetics*, Guangzhou, China, pp. 3911– 3914, 2005. doi: 10.1109/ICMLC.2005.1527621.
[92] M. Bouet, J. Leguay, and V. Conan, "Cost-based placement of virtualized deep packet inspection functions in SDN," in *IEEE Military Communications Conference (MILCOM 2013)*, San Diego, USA, 2013. doi: 10.1109/MILCOM.2013.172.
[93] P. Murukan, D. Jamaluddine, S. Kolhapure, et al. (2016, Feb.). *A cost-based placement algorithm for multiple virtual security appliances in cloud using SDN: MO-UFLP (multi-ordered uncapacitated facility location problem)* [Online]. Available: https://arxiv.org/abs/1602.08155
[94] K. Li, J. Bao, Z. Lu, et al., "A PSO-based virtual SDN customization for multi-tenant cloud services," in *Proc. 11th International Conference on Ubiquitous Information Management and Communication*, Beppu, Japan, Article No. 91, 2017. doi: 10.1145/3022227.3022317.
[95] X. Yao, H. Wang, C. Gao, et al., "Maximizing network utilization for SDN based on particle swarm optimization," in *28th International Conference on Tools with Artificial Intelligence (ICTAI)*, San Jose, USA, 2016. doi: 10.1109/ICTAI.2016.0142.
[96] C. Gao, H. Wang, L. Zhai, et al., "Optimizing routing rules space through traffic engineering based on ant colony algorithm in software defined network," in *28th International Conference on Tools with Artificial Intelligence (ICTAI)*, San Jose, USA, 2016. doi: 10.1109/ICTAI.2016.0026.
[97] S. Yi, H. Wang, X. Yao, et al., "Maximizing network utilization for SDN based on WiseAnt colony optimization," in *IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Sydney, Australia, 2016. doi: 10.1109/HPCC-SmartCity-DSS.2016.0137.
[98] Y. Guo, Z. Wang, X. Yin, et al., "Incremental deployment for traffic engineering in hybrid SDN network," *34th International Performance Computing and Communications Conference (IPCCC)*, Nanjing, China, 2015. doi: 10.1109/PCCC.2015.7410320.

Manuscript received: 2017-06-08

Biographies

LIAO Lingxia (liaolx@ece.ubc.ca) is currently a Ph.D. candidate of Department of Electrical and Computer Engineering of the University of British Columbia (UBC), Canada. She had a bachelor degree from Tsinghua University, China and a master degree from UBC. She was a science researcher in Computer Science Department of UBC, and the R&D director and the general manager of high performance computing of Inspur Group, China. She had over ten years' experience working in computer and network industry. Her current research interests are E-hearthcare, cloud computing, software defined networking, network function virtualization, network monitoring and optimization, and next generation network. She has contributed multiple technical papers and book chapters in these areas.

Victor C.M. Leung (vleung@ece.ubc.ca) is a professor of Electrical and Computer Engineering and holder of the TELUS Mobility Research Chair at the University of British Columbia (UBC). He has contributed some 1000 technical papers, 37 book chapters and 12 book titles in the areas of wireless networks and mobile systems. He was a Distinguished Lecturer of the IEEE Communications Society. He is serving/has served on the editorial boards of the *IEEE Journal on Selected Areas in Communications*, *IEEE Access*, IEEE Transactions on Wireless Communications, IEEE Transactions on Computers, IEEE Transactions on Vehicular Technology, *IEEE Wireless Communications Letters* and several other journals, and has contributed to the organizing and technical program committees of numerous conferences. Dr. Leung was a winner of the 2011 UBC Killam Research Prize, the IEEE Vancouver Section Centennial Award, the 2017 Canadian Award for Telecommunications Research. He co-authored a paper that won the 2017 IEEE Communications Society Fred W. Ellersick Prize.

LAI Chin-Feng (cinfon@ieee.org) is an associate professor at Department of Engineering Science, National Cheng Kung University, China and Department of Computer Science and Information Engineering, National Chung Cheng University, China since 2016. He received the Ph.D. degree in Department of Engineering Science from National Cheng Kung University, China in 2008. He received Best Paper Awards from IEEE 17th CCSE, 2014 International Conference on Cloud Computing, IEEE 10th EUC, and IEEE 12th CIT. He has more than 100 paper publications and 4 papers selected to TOP 1% most cited articles by Essential Science Indicators (ESI). He serves as an associate editor-in-chief, editor, or associate editor for many journals and is TPC Co-Chair for many conferences during 2012–2017. His research focuses on Internet of Things, body sensor networks, E-healthcare, mobile cloud computing, cloud-assisted multimedia network, embedded systems, etc. He has been an IEEE senior member since 2014.

DOI: 10.3969/j. issn. 1673-5188. 2017. 03. 005
http://kns.cnki.net/kcms/detail/34.1294.TN.20170712.1425.002.html, published online July 12, 2017

Distributed Least-Squares Iterative Methods in Large-Scale Networks: A Survey

SHI Lei[1], ZHAO Liang[2], SONG Wenzhan[3], Goutham Kamath[1], WU Yuan[4], and LIU Xuefeng[5]

(1. Georgia State University, Atlanta, GA 30302, USA;
2. Georgia Gwinnett College, Lawrenceville, GA 30043, USA;
3. University of Georgia, Athens, GA 30602, USA;
4. Zhejiang University of Technology, Hangzhou 310023, China;
5. The Hong Kong Polytechnic University, Hong Kong, China)

Abstract

Many science and engineering applications involve solving a linear least-squares system formed from some field measurements. In the distributed cyber-physical systems (CPS), each sensor node used for measurement often only knows partial independent rows of the least-squares system. To solve the least-squares all the measurements must be gathered at a centralized location and then perform the computation. Such data collection and computation are inefficient because of bandwidth and time constraints and sometimes are infeasible because of data privacy concerns. Iterative methods are natural candidates for solving the aforementioned problem and there are many studies regarding this. However, most of the proposed solutions are related to centralized/parallel computations while only a few have the potential to be applied in distributed networks. Thus distributed computations are strongly preferred or demanded in many of the real world applications, e.g. smart-grid, target tracking, etc. This paper surveys the representative iterative methods for distributed least-squares in networks.

Keywords

distributed computing; iterative methods; least-squares; mesh network

1 Introduction

Many physical phenomena can be described by partial differential equations [1] which forms large sparse system of linear equations after further discretized. Problems, such as state estimation, target tracking and tomography inversion, are often formulated as a large-scale linear system based on some field measurements. Those field measurements may contain errors, thus an extra amount of measurement is often sampled to form an over-determined linear system:

$$Ax \approx b , \tag{1}$$

where $A \in \mathbb{R}^{m \times n}(m \geqslant n)$, $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$. Such extra information can smooth out the errors but produces an overdetermined system that usually has no exact solution. The method of least-squares is a common solution to the above problem and can be defined as

$$\min_x \| Ax - b \|_2 . \tag{2}$$

The coefficient A is often modeled from the data obtain from sensors used for observing the physical phenomena such as cyber physical system. Each sensor or node observes partial phenomena due to the spatial and temporal restriction and thus only forms partial rows of the least-squares systems. The large-scale cyber-physical systems are often built on a mesh network, which could be a wired, wireless or wired-wireless hybrid multi-hop network. For instance, the problems from target tracking, seismic tomography and smart grid state estimation all have an inherently distributed system of linear equations. However, the least squares method used currently for solving these problems assumes a centralized setup, where partial row information from all the nodes are collected in a server and then solved using the centralized least-square algorithm.

In many of those cyber-physical systems, the distributed computation in mesh networks is strongly demanded or preferred over the centralized computation approach, due to the following reasons (but not limited to):

1) In some applications such as imaging seismic tomography with the aid of mesh network, the real-time data retrieval from a large-scale seismic mesh network into a central server is virtually impossible due to the sheer amount of data and resource limitations. The distributed computation may process data inside the network in real time to reduce the bandwidth demand as well as distribute the communication and computation load to each node in the network.
2) The mesh network may be disruptive in real world and the data collection and centralized computation may suffer from node failure or link disruption. These become a bottleneck especially when the node failure or link disruption happens

This work is partially supported by US NSF under Grant No. NSF-CNS-1066391 and No. NSF-CNS-0914371, NSF-CPS-1135814 and NSF-CDI-1125165.

near to the sink node which leads to loss of high volume of raw data. However, with distributed computation, the remaining nodes in the network can finish the computation and get the approximated results.

3) In smart grid state estimation, the data collection for centralized computation is even infeasible due to data privacy concerns or inter-agency policy constraints.

4) In some applications that need real-time control, the distributed computation also has advantage over centralized schemes, since some decisions can be made locally in real time. The current state of the art computational device such as smart phones enables us to perform in-network computing and carry out distributed computation over a mesh network.

Iterative methods are natural candidates when it comes for large sparse system and especially for distributed computation of least-squares. Although there are a lot of studies on iterative least-squares, most of them are concerned with the efficiency of centralized/parallel computation, and only a few are explicitly about distributed computation or have the potential to be applied on mesh networks. In mesh networks, since the computers need to communicate with each other through messages passing over a multi-hop network, the key challenges are speeding up the computation and reducing the communication cost. More attention shall be paid to communication instead of the computation cost, especially when solving a big problem in a large-scale mesh network.

In this paper, we select and survey the representative iterative methods from several research communities. These methods have the potential to be used in solving least squares problem over mesh networks. Here, a skeleton sketch of each algorithm is provided and later we analyze the time-to-completion and communication cost of these algorithms and provide the comparison. Some of the algorithms presented here were not originally designed for meeting our requirements, so we slightly modify them to maintain consistency.

The rest of the paper is organized as follows. In section 2, we present the network model and the evaluation criteria for comparison. Then in section 3, we describe the state of art on surveyed algorithms in details, analyze and compare their communication costs and time-to-completion. Finally, we conclude the paper in section 4.

2 Model and Assumption

We denote a wired and/or wireless mesh network with N nodes $v_1, ..., v_N$ which form connected graph and can be reached through multi-hop message relays. Without loss of generality, we assume that the diameter of the network is $\log N$ (i.e., any message can be sent from one node to another through at most $\log N$ hops). We also assume that each node has a single radio and the link between the neighboring nodes has a unit bandwidth. Therefore, the communication delay of one unit data delivery between direct neighbors (either through a unicast to one direct neighbor or multicast/broadcast to all direct neighbors) would be one unit time. It is noted that the link layer supports broadcast, which is often true in many mesh networks. If the link layer only supports unicast, the analysis can be similarly done by considering a one-hop broadcast as multiple unicast and we skip this analysis as it is trivial. We also understand that the link layer communication may take more than one unit time for one unit data due to network interference and media contentions. Therefore, we classify the communication patterns in a mesh network into three categories, unicast (one-hop or multi-hop), one-hop broadcast (local broadcast to all neighbors) and network flooding (broadcast to all the nodes in network). For simplicity and convenience, in the rest of the paper we use the term "broadcast" for local broadcast to one-hop neighbors and "flood" for network flooding. We use the aforementioned assumption on communication cost and delay for the quest of the fundamental limit of each surveyed algorithm in an ideal mesh network.

We assume a random communication network (has to be a connected graph) in our analysis. The influence of the communication network to the performance has been studied in [2], [3]. It is known that the network connectivity ratio, which is defined as the number of edges divided by the number of all possible edgeswill affect the performance. The algorithms are supposed to obtain a faster convergence speed when each node has higher number of neighbors. Higher neighbor count means more nodes can receive the information after one transmission of certain node. It accelerates information diffusion among all the nodes in the network and thus help all the nodes reach consensus faster. To conduct a fair comparison, we use the same communication network (topology) for all the benchmarks.

For comparison and evaluation, the following two performance criteria are considered:

- Communication cost: To solve a least-squares problem of large-size system, communication cost has a big influence on the algorithm performance. Here we refer the communication cost as the cost involved in the messages exchanged in the mesh network during a single iteration of the iterative methods. Since iterative methods typically converge after many iterations, the communication cost of the iterative methods depends on both the cost in one iteration and the iteration number.
- Time-to-completion: The time taken for a network to finish one iteration in the iterative method is referred as time-to-completion in this paper. Note that it is different from the computational time complexity and shall include the consideration of the message size and number of hops the packet has traversed.

We also focus on the analysis of communication delays for time-to-completion while ignoring the computation time in each node. The rationale is summarized as follows. First, the cost of communication is very high in practice. We are highly constrained by the physical bandwidth and energy consumption in large-scale sensor networks in particular. Furthermore, the communication stage is much more time-consuming than the computation step. It turns out that reducing the total communication rounds is the key for speeding up the algorithm in terms of time-to-completion.

The least-squares problem in (2) formed over the mesh network is inherently distributed, i.e. each node v_u only knows part of A and b. We assume that each node in the network holds $m_u = m/N$ consecutive rows of matrix $A \in \mathbb{R}^{m \times n}$ and the corresponding part of vector b. For example in **Fig. 1**, block A_1 indicates the first m/N rows of matrix A which is assigned to node v_1 along with the right hand side vector $b = \{b_1, \cdots, b_{m/N}\}$. Note that the algorithm surveyed in this paper does not require that matrix A and b be equally partitioned over the network. Here the assumption of equal partition is for the simplicity of presentation and analysis and the new distributed equation takes the form:

$$Ax = b , \tag{3}$$

where,

$$A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_N \end{pmatrix}; b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix}; A_u \in \mathbb{R}^{m_u \times n}; b_u \in \mathbb{R}^{m_u} . \tag{4}$$

The least squares problem takes the form $\min_x \|Ax - b\|_2$ and since there is no central coordinator which has entire A and b, the computation of optimum x has to be done distributedly. As mentioned above, the communication cost becomes crucial for distributed solution over sensor network and the goal of this paper is to survey various distributed least squares algorithm originating from different domains. We also try to compare different algorithms under similar criteria as mentioned above so that it provides the reader some basic differences between them and also help them to choose the type of algorithms suitable for their application.

Indices	1	2	3	⋯	$n-1$	n	Block
1	X	X	X	⋯	X	X	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	A_1
m/N	X	X	X	⋯	X	X	
$m/N+1$	X	X	X	⋯	X	X	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	A_2
$2m/N$	X	X	X	⋯	X	X	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$m-m/N+1$	X	X	X	⋯	X	X	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	A_N
m	X	X	X	⋯	X	X	

▲**Figure 1. Row partition of matrix *A*.**

The notations used in this paper are described in **Table 1**.

3 Survey and Analysis

The methods for solving the linear least-squares problems are typically classified into two categories, direct methods and iterative methods. Direct methods are based on the factorization of the coefficient matrix A into easily invertible matrices whereas iterative methods solve the system by generating a sequence of improving approximate solutions for the problem. Until recently direct methods were often preferred over iterative methods [4] due to their robustness and predictable behaviors (one can estimate the amount of resources required by direct solvers in terms of time and storage) [5], [6]. On the other hand, a number of iterative methods have also been discovered, which require fewer memory and are approaching the solution quality of direct solvers [6]. The size of the least squares problem arising from real world three-dimension problem models could be significantly large comprising hundreds of millions of equations as well as the unknowns. Despite such a huge dimension, the matrices arising are typically sparse and can be easily stored. Now given the dimension and sparsity property of the matrix, iterative methods become almost mandatory for solving them [7]. Moreover, iterative methods are gaining ground because they are easier to be implemented efficiently in high-performance computers than direct methods [6].

The methods for solving least-squares problems in distributed networks can be classified into two categories, distributed and decentralized (fully distributed) methods. We will discuss it in detail in this section.

3.1 Distributed Least-Squares Methods

To achieve high performance in computation, researchers have studied distributed iterative methods to solve large linear systems/linear least-squares problems [8], [9]. The researches leverage both shared and distributed memory architecture. In this section, we only present those distributed iterative methods that can be potentially distributed over a mesh network. The Distributed Multisplitting (D-MS), Distributed Modified

▼**Table 1. List of notations**

Notation	Definition
A,E,L,U,Q,R	Matrices
x,y,a,b,r	Vectors
$\alpha,\beta,\delta,\lambda,\gamma$	Scalars
m,n	Rows and columns of matrices
$\mathbb{R}$	Real space
N	Network size
D_{avg},D_{max}	Average and maximum node degree in the network
u,v	Nodes in the network
k	Iteration number of iterative methods

Conjugate Gradient Least-Squares (D-MCGLS), Distributed Component-Average Row Projection (D-CARP), Distributed Cooperative Estimation (D-CE), Distributed Least Mean Squares (D-LMS), Distributed Recursive Least-Squares (D-RLS) methodsare discussed and compared, but only D-LMS and D-RLS are analyzed in details in this paper since they are more relevant and more promising than other distributed algorithms.

Table 2 gives a summary of the communication cost and time-to-completion of the selected algorithms running in the distributed network. The details about the algorithm description and analysis are shown in Section 3. When the least-squares problem in (2) where $A\in\mathbb{R}^{m\times n}(m\geqslant n)$, $x\in\mathbb{R}^n$ and $b\in\mathbb{R}^m$ is considered, we suppose that the iterative algorithm converges within k iterations in the network, and D_{avg} and $D_{\max}$ denote the average and maximum node degrees of the network respectively. The algorithms discussed in this section have been proved to be convergent, but the iteration number highly depends on the matrix condition number; these algorithms may need hundreds to thousands of iterations to converge over a network with hundreds of nodes for a large system. Besides, some algorithms either requires flooding communication in the network per iteration or a Hamiltonian path in the network to perform the computation node by node.

3.1.1 Distributed Least Mean Squares Method

Schizas, Mateos and Giannakis [10]-[13] introduce the D-LMS algorithm. This algorithm lets each node maintain its own local estimation and, to reach the consensus, exchange the local estimation only within its neighbors. The advantage of the methods like D-LMS and D-CE in signal processing is that only local information exchange is required. The problem is that these methods may converge slow in a large-scale network.

In their discussion, the wireless sensor network is deployed to estimate a signal vector $x^*\in\mathbb{R}^{n\times 1}$. Each node v_u has a regression vector $A_u(k)\in\mathbb{R}^{n\times 1}$, where $k=0,1,2,....$ denotes the time instants, and there is a observation $b_u(k)$ on time k; both of them are assumed to have zero mean. One global vector $b(k):=[b_1(k)\cdots b_N(k)]^T\in\mathbb{R}^{N\times 1}$ is used for all the observations on N nodes in the network. $A(k):=[A_1(k)\cdots A_N(k)]^T\in\mathbb{R}^{N\times n}$ is the regression vector combined over the network, and the global LMS estimator is then described as

$$\begin{aligned}\hat{x}(k)&=\arg\min_x E[\|b(k)-A(k)x\|^2]=\\&\arg\min_x\sum_{u=1}^{N}E[(b_u(k)-A_u^T(k)x)^2]\end{aligned}. \tag{5}$$

Let $\{x_u\}_{u=1}^N\in\mathbb{R}^n$ represent the local estimation of the global variable x of one node v_u (each node has its own estimation of the signal vector). In conjunction with these local variables, we consider the convex constrained minimization problem as

$$\begin{aligned}&\{\hat{x}_u(k)\}_{u=1}^N=\arg\min_x\sum_{u=1}^{N}E[(b_u(k)-A_u^T(k)x_u)^2],\\&s.t.x_u=x_{u'},u\in N,u'\in N_u,\end{aligned} \tag{6}$$

where N_u is the neighbor set of node v_u.

The equality constraints above only involve the local estimations of the neighbors of each node and force an agreement among each node's neighbors. Since we assume that the network is connected, the constraints above will introduce a consensus in the network. We can finally have $x_u=x_{u'}$ for all $u,u'\in N$. We find that the distributed estimation problem is equivalent to the original problem in the sense that their optimal solutions coincide such as $\hat{x}_u(k)=\hat{x}(k)$, for all $u\in N$.

To construct the distributed algorithm, the authors resort to the Alternating Direction Method of Multipliers (ADMM) (algorithm, and get the following two equations for estimation updating,

$$\begin{aligned}&v_u^{u'}(k)=v_u^{u'}(k-1)+\frac{c}{2}\left(x_u(k)-\left(x_{u'}(k)+\eta_u^{u'}(k)\right)\right),\\&u'\in N_u,\end{aligned} \tag{7}$$

$$\begin{aligned}&x_u(k+1)=x_u(k)+\mu_u[2A_u(k+1)e_u(k+1)-\\&\sum_{u'\in N_u}\left(v_u^{u'}(k)-\left(v_{u'}^{u}(k)+\bar{\eta}_u^{u'}(k)\right)\right)-\\&c\sum_{u'\in N_u}\left(x_u(k)-\left(x_{u'}(k)+\eta_u^{u'}(k)\right)\right)],\end{aligned} \tag{8}$$

where μ_u is a constant step-size and $e_u(k+1):=b_u(k+1)-A_u^T(k+1)x_u(k)$ is a local priori error. $\eta_u^{u'}(k)$ and $\bar{\eta}_u^{u'}(k)$ denote the additive communication noise present in the reception of $x_{u'}(k)$ and $v_{u'}^{u}(k)$. **Algorithm 1** gives the description of the distributed least mean square algorithm. In detail, during the time instant $k+1$, node v_u receives the local estimates $\{x_{u'}(k)+\eta_u^{u'}(k)\}_{u'\in N_u}$ and plugs them into the equations above to evaluate $v_u^{u'}(k)$ for $u'\in N_u$. Each updated local Lagrange multiplier $\{v_u^{u'}(k)\}_{u'\in N_u}$ is subsequently transmitted to the corr-

▼**Table 2. Analysis of communication cost and time-to-completion**

Algorithm	Communication cost	Time-to-completion
D-MS	kmN^2	$km(N-1)$
D-MCGLS	$(k+1)(m+N)N+k(n+N)N$	$k(m+n+2)(N-1)$
D-CARP	$2knN$	$2n(N-1)+n\log N$
D-CE	knN	$knD_{\max}$
D-LMS	$kN(D_{avg}+1)$	$2kD_{\max}$
D-RLS	$(n+n^2)(N-1)$	$(n+n^2)(N-1)$

N is the network size, $m\times n(m\geqslant n)$ is dimensions of matrix A and k is the number of iterations (usually $m>>N$ and $n>>N$)
D-CARP: Distributed Component-Average Row Projection method.
D-CE: Distributed Cooperative Estimation methods.
D-LMS: Distributed Least Mean Squares method.
D-MCGLS: Distributed Modified Conjugate Gradient Least-Squares method.
D-MS: Distributed Multisplitting method.
D-RLS: Distributed Recursive Least-Squares method.

esponding neighbor $u' \in N_u$. Then upon reception of $\{v_{u'}^{u}(k)+\bar{\eta}_u^{u'}(k))\}_{u'\in N_u}$, the multipliers are jointly used along with $\{x_{u'}(k)+\eta_u^{u'}(k))\}_{u'\in N_u}$ and the newly acquired local data $\{b_u(k+1),A_u(k+1)\}$ to obtain $x_u(k+1)$ via the above equations. The $(k+1)$-st iteration is concluded after node v_u broadcasts $x_u(k+1)$ to its neighbors.

Algorithm 1: D-LMS Method

Each node v_u follows the same routines below

1. Arbitrarily initialize $\{x_u(0)\}_{i=1}^{N}$ and $\{v_u^{u'}(-1)\}_{i\in N}^{u'\in N_u}$
2. **While** not converged **do**
3. Broadcast $x_u(k)$ to neighbors in N_u
4. Update $\{v_u^{u'}(k)\}_{u'\in N_u}$
5. Transmit $v_u^{u'}(k)$ to each $u' \in N_u$
6. Update $x_u(k+1)$

end while

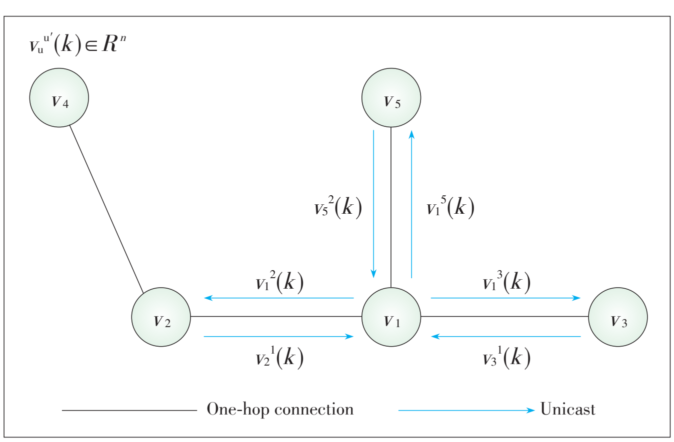
---

1) Communication Cost

Algorithm 1 is simple and only steps 3 and 5 involve communication. Applied to system $Ax=b$, vectors $x_u(k)$ and $v_u^{u'}(k)$ are both of length n (columns of A). For example, in step 3, node v_1 needs to transmit $x_u(k)$ to all its neighbors; in step 5, v_1 needs to transmit different $v_u^{u'}(k)$ to different neighbors (**Fig. 2**). Suppose that the average degree of network is D_{avg}, in each iteration, the communication cost of one node is of $n(D_{avg}+1)$, so the communication of the network is $nN(D_{avg}+1)$. Suppose that after k iterations (the iteration number might be greater than the time instants, so after the k-th sample on node v_u is involved, the first sample is used as the $(k+1)$-st sample), the algorithm converges and the total communication cost is $knN(D_{avg}+1)$.

2) Time-to-Completion

In step 3 of Algorithm 1, node u needs to broadcast $x_u(k)$ to all its neighbors. From the receiver side, each node needs to receive different updates from all its neighbors, and then the delay of the whole network depends on the maximum node degree of the network since the algorithm is synchronous; this delay is $nD_{\max}$. In step 5, node u needs to send different $v_u^{u'}(k)$ to different neighbors, the delay is also $nD_{\max}$. The total communication delay is then $2knD_{\max}$.



▲Figure 2. Communication pattern of D-LMS method.

3.1.2 Distributed Recursive Least-Squares Method

Sayed and Lopes [14] developed a distributed least-squares estimation strategy by appealing to collaboration techniques that exploit the space-time structure of the data, achieving an exact recursive solution that is fully distributed. This D-RLS strategy is developed by appealing to collaboration techniques to achieve an exact recursive solution. It requires a cyclic path in the network to perform the computation node by node. The advantage of this method is the iteration number is fixed (the network size) for a give set of data to solve a least-squares problem, but the problem is a large dense matrix needs to be exchanged between nodes.

The details and analysis of D-RLS strategy are given in this section, and **Algorithm 2** gives the classic RLS procedure [15].

Algorithm 2: Recursive Least-Squares Procedure

Initial: $x^{-1}=\bar{x}$ and $P^{-1}=I$

1. **for** $k \geqslant 0$ **do**
2. $x^k = x^{k-1} + g^k\left[b(k) - A^T(k)x^{k-1}\right]$
3. $g^k = \dfrac{\lambda^{-1}P^{k-1}A(k)}{1+\lambda^{-1}A^T(k)P^{k-1}A(k)}$
4. $P^k = \lambda^{-1}[P^{(k-1)} - g^k A^T(k)P^{(k-1)}]$

end for

To distribute the exact algorithm for estimating the vector x in the network of N nodes, each node v_u has access to regressors and measurement data $A_u(k)$ and $b_u(k), u=1,...,N$, where $b_u(k)\in\mathbb{R}$ and $A_u(k)\in\mathbb{R}^n$. At each time instant k, the network has access to space-time data:

$$b(k)=\begin{bmatrix} b_1(k) \\ b_2(k) \\ \vdots \\ b_N(k) \end{bmatrix} \ and\ A(k)=\begin{bmatrix} A_1(k) \\ A_2(k) \\ \vdots \\ A_N(k) \end{bmatrix}, \tag{9}$$

where $b(k)$ and $A(k)$ are snapshot matrices revealing the network data status at time k. We collect all the data available up to time k into global matrices b and A:

$$b=\begin{bmatrix} b(0) \\ b(1) \\ \vdots \\ b(k) \end{bmatrix} \ and\ A=\begin{bmatrix} A(1) \\ A(2) \\ \vdots \\ A(k) \end{bmatrix}. \tag{10}$$

Note that here is equivalent to solving the least-squares problem of $Ax = b$ by partition A and b row-wise and each node has one partition of consecutive rows of A and b. Applying the RLS algorithm is different to the D-LMS estimator since it gives the least-squares solution of the whole block of data $Ax = b$. Therefore, in the distributed RLS algorithm to solve a normal least-squares problem, we use $A_u(k)$ to indicate the row block on node v_u but not only one vector collected at time instant k (we can treat it as all the data collected till time k).

By assuming an incremental path is defined across the network cycling from node v_1 to v_2 and so forth, until node v_N, The RLS algorithm can be rewritten as a distributed version in **Algorithm 3** [14].

Algorithm 3: D-RLS Method

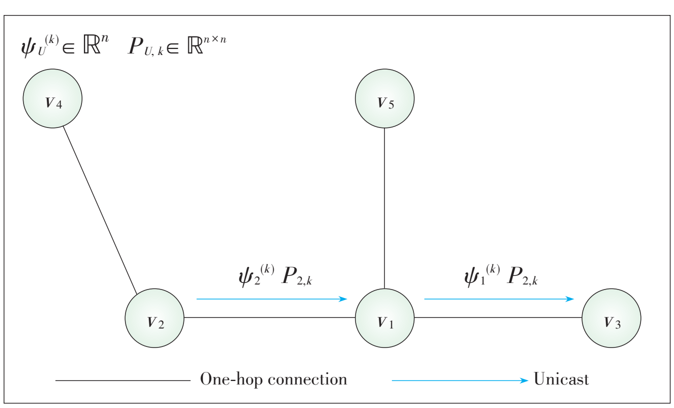
$\psi_0^{(k)} = x^{k-1}$, $P_{0,k} = \lambda^{-1} P^{k-1}$

1. **for** $u = 1{:}N$, node v_u **do**
2. $e_u(k) = b_u(k) - A_u(k)\psi_{u-1}^{(k)}$
3. $\psi_u^{(k)} = \psi_{u-1}^{(k)} + \dfrac{P_{u-1,k}}{\gamma_u^{-1} + A_u^T(k) P_{u-1,k} A_u(k)} A_u(k) e_u(k)$
4. $P_{u,k} = P_{u-1,k} - \dfrac{P_{u-1,k} A_u(k) A_u^T(k) P_{u-1,k}}{\gamma_u^{-1} + A_u^T(k) P_{u-1,k} A_u(k)}$
5. If $u \neq N$ then
6. v_u send $\left\{\psi_u^{(k)}, P_{u,k}\right\}$ to node v_{u+1}
7. **end if**

end for

1) Communication Cost

In the distributed RLS algorithm, the communication is in step 6. Each node shares with its successor node in the cycle path of network the quantities $\left\{\psi_u^{(k)}, P_{u,k}\right\}$, where $\psi_u^{(k)} \in \mathbb{R}^n$ and $P_{u,k} \in \mathbb{R}^{n \times n}$. For example, node v_1 receives the message from v_2 and sends it to v_3 (**Fig. 3**). Therefore, in each iteration, the communication cost only happens in one node and it is $n + n^2$. Since the algorithm can converge after one cycle in the network, the total communication cost is $(n + n^2)(N - 1)$. Note that a Hamiltonian path is required by this algorithm, and to find such a path, extra communication is required. This is another problem and out of the scope of the analysis in this paper, we omit this cost here.



▲Figure 3. Communication pattern of D-RLS method.

2) Time-to-Completion

In distributed RLS algorithm, it is easy to see that the delay in one step is $n + n^2$ and there are totally $N - 1$ steps in the algorithm, so the total time-to-completion is $(n + n^2)(N - 1)$.

3.2 Decentralized Optimization Methods

In recent years, much attention has been paid to fully distributed (decentralized) consensus optimization problems, especially in applications like distributed machine learning, multi-agent optimization, etc. Several algorithms have been proposed for solving general convex and (sub)differentiable functions. By setting the objective function as least-square, the decentralized least-square problem can be seen as a special case of the following model.

$$\min_{x \in \mathbb{R}^n} F(x) := \sum_{i=1}^{p} F_i(x), \tag{11}$$

where p nodes are in the network and they need to collaboratively estimate the model parameters x. Each node i locally holds the function F_i and can communicate only with its immediate neighbors.

Considering the problem in (11), (sub)gradient-based methods have been proposed [16]–[20]. However, it has been analyzed that the aforementioned methods can only converge to a neighborhood of an optimal solution in the case of fixed step size [21]. Modified algorithms have been developed in [16] and [17], which use diminishing step sizes in order to guarantee converging to an true solution. Other related algorithms were discussed in [22]–[28], which share similar ideas. The D-NC algorithm proposed in [16] was demonstrated to have an outer-loop convergence rate of $O(1/k^2)$ in terms of objective value error. The rate is same as the optimal centralized Nesterov's accelerated gradient method and decentralized algorithms usually have slower convergence rate than the centralized versions. However, the number of consensus iterations within outer-loop is growing significantly along the iteration. Shi [29] developed a method based on correction on mixing matrix for Decentralized Gradient Descent (DGD) method [21] without diminishing step sizes.

The algorithms mentioned above are based on synchronous models. Distributed optimization methods for asynchronous models have been designed in [30]–[32]. However, it is worth noting that their convergence rates are usually slower than the counterparts in synchronous models. In the next, we show the derivations of various decentralized optimization methods. In order to have a compact form, let $x \in \mathbb{R}^{np \times 1} := \left[x_1^T, x_2^T, \cdots, x_p^T\right]^T$, where x_i is a column vector containing local estimate of common interest x at node i. Similarly, define $F(x)=\sum_{i=1}^{p}F_i(x_i)$ and $\nabla F(x)\in\mathbb{R}^{np\times 1}:=\left[\nabla F_1(x_1)^T,\cdots,\nabla F_p(x_p)^T\right]^T$, where $\nabla F_i(\cdot)$ denotes the gradient of F_i.

Without loss of generality, we assume $n=1$ (the size of decision variable x), and the aforementioned problem can be formulated as:

$$\min_{x\in\mathbb{R}^p} F(x): Wx=x. \tag{12}$$

The constraint requires x to be consensual due to the property of the mixing matrix W. Now we first derive the DGD algorithm [21] from (5). Assuming W is symmetric and U is symmetric, we get $U^2=I-W$. Then $Wx=x$ if and only if $Ux=0$ holds. The original problem in (12) is thus equivalent to:

$$\min_{x\in\mathbb{R}^p} F(x): Ux=0. \tag{13}$$

If we use external penalty method, an unconstrained (but inaccurate) reformation can be expressed as:

$$\min_{x\in\mathbb{R}^p} F(x)+\frac{\rho}{2}\|Ux\|^2. \tag{14}$$

It is clear to see that (14) will approximate (15) as $\rho\rightarrow\infty$. Applying the gradient descent method to (14), we can obtain the following update rule:

$$\begin{aligned}x_{k+1}&=x_k-\alpha_k\left(\nabla F(x_k)+\rho_k(I-W)x_k\right)=\\&\left(1-\alpha_k\rho_k\right)x_k+\alpha_k\rho_k Wx_k-\nabla F(x_k).\end{aligned} \tag{15}$$

If choosing $\alpha_k\rho_k=1$ for all k, it yields the DGD algorithm:

$$x_{k+1}=Wx_k-\alpha_k\nabla F(x_k). \tag{16}$$

If we choose constant step-size $\alpha_k\equiv\alpha$, this is solving for the fixed $\rho=1/\alpha$, and hence not yielding the optimal consensus solution.

The DGD algorithm is shown in **Algorithm 4.**

Algorithm 4: DGD Method

Initialize $x_i^0,\forall i\in\{0,1,\cdots,p\}$.

1. **for** $k=0,1,\cdots$, node i **do**
2. $x_i^{k+1}=\sum_{j=1}^{p}W_{ij}x_j^k-\alpha\nabla F_i\left(x_i^k\right)$
3. Node i sends its updated value $x_i^{(k+1)}$ to its neighbors.

end for

The same situation happens in the algorithm D-NG (**Algorithm 5**) [16].

$$x_{k+1}=Wy_k-\alpha_k\nabla F(y_k), \tag{17}$$

$$y_{k+1}=x_{k+1}+\frac{k-1}{k+2}\left(x_{k+1}-x_k\right). \tag{18}$$

It is equivalent to applying (a special version of) Nesterov's gradient method [33] to the problem in (14) with increasing value of ρ (i.e. with diminishing α_k). Comparing (18) with (17), we can find that D-NG differs from DGD only in using y_k instead of x_k in (16). y_k defined in (11) is actually an extrapolation of the current x_k and the previous iteration x_{k-1}.

Algorithm 5: D-NG Method

Initialize $x_i^0,\forall i\in\{0,1,\cdots,p\}$.

1. **for** $k=0,1,\cdots$, node i **do**
2. $x_i^{k+1}=\sum_{j=1}^{p}W_{ij}x_j^k-\alpha\nabla F_i\left(x_i^k\right)$
3. Node i send its updated value x_i^{k+1} to its neighbors.

end for

We also show the derivation of Exact First-Order Algorithm (EXTRA) (**Algorithm 6**) [29]. We claim that EXTRA is equivalent to applying Alternating Direction Method of Multipliers (ADMM)method [34] to the original problem in (13). In (13), we would like to solve the exact constraint eventually. To this end, an unconstrained reformulation using augmented Lagrangian method can be described as:

$$\max_{z\in\mathbb{R}^p}\min_{x\in\mathbb{R}^p} F(x)-\rho z, Ux+\frac{\rho}{2}\|Ux\|^2. \tag{19}$$

Solving (19) by ADMM yields the following update equations:

$$z_{k+1}=z_k+Ux_k, \tag{20}$$

$$x_{k+1}=\arg\min_x\left\{F(x)+\frac{\rho}{2}\|Ux-z_k\|^2\right\}. \tag{21}$$

When a linearized preconditioned (approxiate $F(x)$ and linearize $\frac{\rho}{2}\|Ux-z_k\|^2$) version of x_{k+1} step is considered, (21) becomes:

$$\begin{aligned}x_{k+1}&=\arg\min_x\left\{\nabla F(x_k),x+\rho U(Ux_k-z_k),x+\frac{1}{2\alpha_k}\|x-x_k\|^2\right\}\\&=\left(I-\alpha_k\rho U^2\right)x_k+\alpha_k\rho Uz_k-\alpha_k\nabla F(x_k),\end{aligned} \tag{22}$$

where I denotes the identity matrix. Now for constant step-size $\alpha_k\equiv\alpha=1/\rho$, combining (20) and $U^2=I-W$ yields the EXTRA algorithm:

$$x_{k+1}=Wx_k-\alpha\nabla F(x_k)+\sum_{t=0}^{k-1}(I-W)x_t. \tag{23}$$

Note that (23) can also be seen as a "corrected" version of the DGD algorithm [29] comparing to (16).

Algorithm 6: EXTRA Method

Initialize $x_i^0, \forall i \in \{0,1,\cdots,p\}$. Set $\tilde{W} = (I+W)/2$.

1. **for** $k=0,1,\cdots$, node i **do**
2. $x_i^{k+1} = \sum_{j=1}^{p} \tilde{W}_{ij} x_j^k - \alpha \nabla F_i(x_i^k)$
3. $x_i^{(k+2)} = \sum_{j=1}^{p} W_{ij} x_j^{(k+1)} - \alpha \nabla F_i(x_i^{(k+1)})$
4. Node i send its updated value x_i^{k+2} to its neighbors.

end for

Finally, we discuss the Fast Decentralized Gradient Descent (FDGD) method [35] that can be seen as an accelerated version of EXTRA. FDGD does not require diminishing step size and the method is accelerated to reach an optimal $O(1/k^2)$ convergence rate for general convex differentiable functions F_i. We adopte the idea of Nesterov's optimal gradient method for centralized smooth optimization [36] and mixing matrix method in network gossip and consensus averaging algorithms [29], [37]. In **Algorithm 7**, the superscript "*ag*" stands for "aggregated", and "*md*" stands for "middle". Matrix $\tilde{W} = (I+W)/2$ is a half-mixing matrix based on W. At iteration k, each node i sends its current x_k^i to all its immediate neighbors and receives x_k^j from them (one round of communication). The result $x^{i,ag}$ is output as the final solution. Algorithm 7 is a first-order method since only ∇F is required in each iteration, and hence the subproblem has low computation complexity. We do not need to use diminishing step sizes that converge to 0 but still can ensure both of convergence and consensus. Besides, if $\theta_k = 1$ for all k, Algorithm 7 reduces to a version very similar to regular decentralized gradient descent (16). However, by the choice of $\theta_k = O(1/k)$ as below, the change from input x_k^{md} to output x_{k+1}^{ag} is faster than that from x_k to x_{k+1}. This implies that Algorithm 7 can converge faster than regular DGD. The last remark explains intuitively why the multi-step scheme defined in Algorithm 7 can potentially accelerate the convergence. The comparison of various decentralized least-square methods is summarized in **Table 3**.

Algorithm 7: FDGD Method

Initialize $x_0^i, y_0^i = 0, x_0^{i,ag}, \forall i \in \{0,1,\cdots,p\}$. Set $\theta_k = \frac{2}{k+2}$, $L := \max\{L_i\}, \forall i$, where L_i is the Lipschitz constant of node i.

1. **for** $k=0,1,\cdots$, node i **do**
2. $y_{k+1}^i = y_k^i + \sum_{j=1}^{p} (\tilde{W}_{ij} - W_{ij}) x_k^j$
3. $x_k^{i,md} = (1-\theta_k) x_k^{i,ag} + \theta_k \sum_{j=1}^{p} \tilde{W}_{ij} x_k^j$
4. $x_{k+1}^i = \sum_{j=1}^{p} \tilde{W}_{ij} x_k^j - y_{k+1}^i - \frac{1}{L\theta_k} \nabla F_i(x_k^{i,md})$
5. $x_{k+1}^{i,ag} = (1-\theta_k) x_k^{i,ag} + \theta_k x_{k+1}^i$
6. **end for**

Output $x_{k+1}^{i,ag}$

Table 3. Communication cost and convergence speed comparison

Algorithm	Communication cost	Convergence rate
DGD	$O(kN)$	Convergence to a neighborhood
D-NG	$O(kN)$	$O(\log k/k)$ for objective function
EXTRA	$O(kN)$	Ergodic rate $O(1/k)$ for residual
FDGD	$O(kN)$	$O(1/k^2)$ for objective function

N is the network size, and k is the number of communication.
DGD: Decentralized Gradient Descent EXTRA: Exact First-Order Algorithm
D-NG: Distributed Nesterov Gradient FDGD: Fast Decentralized Gradient Descent

4 Conclusions

In this paper, we surveyed some of the developments in distributed iterative methods and parallel iterative methods which can be potentially applied to solve least-squares problems in the mesh network. We covered the traditional iterative methods for solving linear systems including the relaxation methods, the conjugate gradient methods and the row action methods. One algorithm from each category is selected for describing how to apply them to solve least-squares problem in mesh network. We also surveyed some consensus and diffusion based strategies for parameter estimation in signal processing in the mesh network. Such the strategies only require local communications, however, for a large scale network, may take more iterations to converge to the required accuracy to reach an agreement among all the nodes. Algorithm selection depends on the context of the problem and the mesh network.

We also analyzed and compared the performance of the selected representative algorithms in terms of communication cost and time-to-completion. These two concerns are critical for evaluating the performance of distributed algorithms in the context of mesh networks. Besides, we think that a future research direction of distributed computing in mesh networks is the data loss tolerance: will the algorithm still approximate the optimal estimation x^* well if α-percent packets get lost in the network? Different from traditional parallel machines where data delivery is often guaranteed, in many distributed network applications, preventing data losses can either be very expensive (such as sensor networks) as it requires retransmissions, or have a time constraint in real-time applications (such as smart grid) that makes retransmitted data useless.

References

[1] A. Bjõrck, *Numerical Methods for Least Squares Problems*. Philadelphia, USA: SIAM, 1996.

[2] L. Zhao, W. Z. Song, and X. Ye, "Decentralized consensus in distributed networks," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 20, pp. 1–20, 2016. doi: 10.1080/17445760.2016.1233552.

[3] L. Zhao, W. Z. Song, X. Ye, and Y. Gu, "Asynchronous broadcast-based decentralized learning in sensor networks," *International Journal of Parallel, Emer-

gent and Distributed Systems, vol. 32, pp. 1–19, 2017.
[4] Y. Saad and H. A. van der Vorst, "Iterative solution of linear systems in the 20th century," *Journal of Computational and Applied Mathematics*, vol. 123, no. 1–2, pp. 1–33, 2000. doi: 10.1016/S0377-0427(00)00412-X.
[5] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices of Numerical Mathematics and Scientific Computation*. Oxford, UK: Clarendon Press, 1989.
[6] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd edition. Philadelphia, USA: SIAM, 2003.
[7] C. C. Paige and M. A. Saunders, "LSQR: an algorithm for sparse linear equations and sparse least squares," *ACM Transactions on Mathematical Software*, vol. 8, no. 1, pp. 43–71, Mar. 1982.
[8] M. Baboulin, "Solving large dense linear least squares problems on parallel distributed computers. Application to the Earth's gravity field computation," Ph.D. Thesis, Toulouse Institute of Technology, Toulouse, France, 2006.
[9] M. T. Heath, E. Ng, and B. W. Peyton, "Parallel algorithms for sparse linear systems," *SIAM review*, vol. 33, no. 3, pp. 420–460, 1991.
[10] G. Mateos and G. B. Giannakis, "Distributed recursive least-squares: stability and performance analysis," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3740–3754, 2012.
[11] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Performance analysis of the consensus-based distributed LMS algorithm," *EURASIP Journal on Advances in Signal Processing*, 2009. doi: 10.1155/2009/981030.
[12] I. D. Schizas, G. B. Giannakis, S. I. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc WSNs with noisy links—part II: distributed estimation and smoothing of random signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1650–1666, 2008. doi: 10.1109/TSP.2007.908943.
[13] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links—part I: distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350–364, 2008. doi: 10.1109/TSP.2007.906734.
[14] A. H. Sayed, and C. G. Lopes, "Distributed recursive least-squares strategies over adaptive networks. signals, systems and computers," *Fortieth Asilomar Conference on Signals, Systems and Computers (ACSSC' 06)*, Pacific Grove, USA, 2006, pp. 233–237. doi: 10.1109/ACSSC.2006.356622.
[15] V. K. Madisetti, *Digital Signal Processing Fundamentals*. Boca Raton, USA: CRC Press, 2009.
[16] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1131–1146, May 2014. doi: 10.1109/TAC.2014.2298712.
[17] I.-A. Chen, "Fast distributed first-order methods," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, USA, 2012.
[18] I. Matei, and J. S. Baras, "Performance evaluation of the consensus-based distributed subgradient method under random communication topologies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 754–771, 2011. doi: 10.1109/JSTSP.2011.2120593.
[19] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," in *IEEE 52nd Annual Conference on Decision and Control (CDC)*, Florence, Italy, 2013, pp. 6855–6860.
[20] A. Nedic and A. Olshevsky, "stochastic gradient-push for strongly convex functions on time-varying directed graphs," arXiv:1406.2075, 2014.
[21] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," arXiv:1310.7063, 2013.
[22] H. Terelius, U. Topcu, and R. Murray, "Decentralized multi-agent optimization via dual decomposition," in *18th IFAC World Congress*, Milano, Italy, 2011.
[23] J. N. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. Thesis, MIT, Cambridge, USA, 1984.
[24] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks. Information processing in sensor networks," in *Third International Symposium on Information Processing in Sensor Networks*, Berkeley, USA, 2004, pp. 20–27. doi: 10.1145/984622.984626.
[25] G. Shi and K. H. Johansson, "Finite-time and asymptotic convergence of distributed averaging and maximizing algorithms," arXiv:1205.1733, 2012.
[26] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.
[27] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th International Symposium on Information Processing in Sensor Networks*, Piscataway, USA, 2005. doi: 10.1109/IPSN.2005.1440896.
[28] M. Zargham, A. Ribeiro, and A. Jadbabaie, "A distributed line search for network optimization," in *American Control Conference (ACC)*, Montreal, Canada, 2012, pages 472–477. doi: 10.1109/ACC.2012.6314986.
[29] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: an exact first-order algorithm for decentralized consensus optimization," arXiv:1404.6264, 2014.
[30] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem. "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," arXiv:1303.2837, 2013.
[31] E. Wei and A. Ozdaglar, "On the O(1/k) convergence of asynchronous distributed alternating direction method of multipliers," arXiv:1307.8254, 2013.
[32] L. Zhao, W.-Z. Song, L. Shi, and X. Ye, "Decentralised seismic tomography computing in cyber-physical sensor systems," *Cyber-Physical Systems*, vol. 1, no. 2–4, pp. 91–112, 2015. doi: 10.1080/23335777.2015.1062049.
[33] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course (Applied Optimization)*, 1st edition. New York, USA: Springer, 2004. doi: 10.1007/978-1-4419-8853-9.
[34] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, Jan. 2011. doi: 10.1561/2200000016.
[35] L. Zhao, W. Z. Song, and X. Ye, "Fast decentralized gradient descent method and applications to in-situ seismic tomography," *IEEE International Conference on Big Data*, Santa Clara, USA, 2015, pp. 908–917. doi: 10.1109/BigData.2015.7363839.
[36] Y. Nesterov, "Gradient methods for minimizing composite objective function," CORE Discussion Papers, 2007076, Catholic University of Louvain, Louvain-la-Neuve, Belgium, Sept. 2007.
[37] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006. doi: 10.1109/TIT.2006.874516.

Manuscript received: 2017-02-14

Biographies

SHI Lei (lshi1@student.gsu.edu) received his B.Sc. degree in software engineering from Tongji University, China in 2007, and the M.Sc. degree in computer science from Shanghai Jiao Tong University, China in 2010. He has received his Ph.D. degree from the Department of Computer Science of Georgia State University, USA. His research interests include wireless sensor networks, in-network processing and distributed systems.

ZHAO Liang (lzhao2@ggc.edu) received his Ph.D. in computer science from Georgia State University, USA and M.S in electrical engineering from Lehigh University, USA in 2016 and 2012, respectively. He is currently an assistant professor in computer science at Georgia Gwinnett College, USA. His current research interests are in the area of optimization and data analytics for distributed systems.

SONG Wenzhan (wsong@uga.edu) is now a professor at College of Engineering, University of Georgia, USA. His research mainly focuses on sensor web, smart grid and smart environment where sensing, computing, communication and control play a critical role and need a transformative study. His research has received 6 million+ research funding from NSF, NASA, USGS, Boeing, etc. since 2005. He is an IEEE senior member.

Goutham Kamath (gkamath1@student.gsu.edu) received his B.E. degree from India in 2009 and M.S. in electrical engineering from University of Wyoming, USA in 2012. He has received his Ph.D. degree in the Department of Computer Science of Georgia State University, USA. His research interests include wireless sensor networks, distributed systems and mobile ad-hoc networks.

WU Yuan (iewuy@zjut.edu.cn) received the Ph.D. degree in electronic and computer engineering from the Hong Kong University of Science and Technology, China in 2010. He is currently an associate professor at the College of Information Engineering of Zhejiang University of Technology, China. His research interests focus on resource allocations for cognitive radio networks and smart grids.

LIU Xuefeng (csxfliu@comp.polyu.edu.hk) received the M.S. and Ph.D. degrees from Beijing Institute of Technology, China, and University of Bristol, United Kingdom, in 2003 and 2008, respectively. He is currently a research fellow at the Department of Computing of Hong Kong Polytechnic University, China. His research interests include wireless sensor networks, distributed computing, and in-network processing.

DOI: 10.3969/j.issn.1673-5188.2017.03.006
http://kns.cnki.net/kcms/detail/34.1294.TN.20170912.1644.004.html, published online September 12, 2017

High Performance Optical Modulator and Detector for 100 Gb/s Transmission System

ZHONG Fang[1], GONG Pan[1], ZHOU Zhiping[1], HUA Feng[2], and WANG Huitao[2]
(1. State Key Laboratory of Advanced Optical Communication Systems and Networks, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China;
2. Wireline Product Operation, ZTE Corporation, Beijing 100191, China)

Abstract

Silicon photonics, one of the most promising candidates for breaking the bottleneck of current optical transmission systems, has been developing rapidly in both performance and maturity. The analysis and design of the two key components of this technology, the optical modulator and detector, are presented in this paper. The Mach-Zehnder modulator with U-type PN junction is optimized to obtain the modulation efficiency of 0.559 V·cm. The electro-optical 3 dB bandwidth of this device is 30 GHz. The simulation of the PIN waveguide Si based Ge photodetector at 1.55 μm wavelength is also presented. The device shows a very low dark current of about 10 nA at -1 V, and the obtained responsivity and 3 dB bandwidth are appreciable. These results practically meet the requirement of commercial 100 Gb/s optical transmission systems.

Keywords

silicon photonics; modulator; detector

1 Modulation and Detection in Silicon

Traditional copper interconnects have fundamental limitations such as high loss, crosstalk, and low speed [1]–[3]. Such limitations stem from the properties of electrons and therefore make it increasingly difficult and costly to improve copper interconnects. On the other hand, using photons as the carrier of information evades the problems of electrons with excellent potential for several orders of magnitude higher capacity utilizing wavelength division multiplexing (WDM). As a result, the combined integration of electronic and photonic circuits is poised to break through the interconnection bottlenecks in data networks and on-chip interconnections [4]–[6]. Silicon photonics is currently the leading candidate technology to meet the growing demand for network interconnection capacity and speed. Its compatibility with complementary-metal-oxide-semiconductor (CMOS) fabrication process has positioned this technology towards large-scale and dense integration, high reproducibility and low fabrication costs.

Global information communications require optical transmission systems with high spectral efficiency, high channel data rate, and low cost [7]–[9]. Coherent optical transmission has become a solution for high-capacity long-haul communications with channel data rates at 100 Gb/s and beyond [9]–[11], which employs polarization-division-multiplexed quadrature phase-shift keying (PDM-QPSK). Next-generation transport networks may utilize even higher-level modulation formats, such as 16-ary quadrature amplitude modulation (16-QAM). The soaring complexity poses a challenge to current fiber communication systems but at the same time creates opportunities for the versatile silicon photonics.

As one of the most important components in optical interconnection or communication system, silicon optical modulators have undergone significant development in recent years. Many modulation mechanisms were put forward, and the modulation speed increased from megahertz to multi-gigahertz regime [12]–[21]. The power consumption decreased to femto Joules. A modulator is the device for modulating optical parameters such as amplitude, phase and polarization and transforming the electrical signal to the optical signal. Compared with directly modulating optical sources, external modulation using the modulator has several advantages [16]. The optical sources can be cheaper without internal modulation and the modulation speed can be higher. Phase modulation and higher formats are possible, and it can reduce the power budget in multiple channels system with a single light source feeding. In this work, we focus on electro-optic modulators. The real and imaginary part of material refractive index (RI) may be changed when electric field is applied [17]. Depending on which part of refractive index changed, modulators are categorized into electro-refractive (Δn) and electro-absorptive ($\Delta\alpha$). Major electro-optic effects involving Pockels effect, Kerr effect, and Franz–Keldysh effect are used in traditional modulators to change the material RI. Generally, RI is related to the applied electric field in the following relation:

$$n' = n + a_1|E| + a_2|E|^2 + a_3|E|^3 + \cdots, \tag{1}$$

where n' is the RI of the materials when the electric field is applied, n is the RI without applied electric field, and a_1, a_2, a_3 are the different coefficients of different electro-optic effects. The Pockels effect and Kerr effect are described as $\Delta n = a_1 E$

This work is partially supported by ZTE Industry-Academia-Research Cooperation Funds.

and $\triangle n=a_2E^2$, respectively. As one of the most successful commercial modulators, Lithium niobate modulator operates via Pockels effect and exhibits favorable characteristics, especially high speed. However, the electro-optic effects mentioned above are inconsequential in pure silicon at the telecommunications wavelengths of 1.3 μm and 1.55 μm [5]. Besides, silicon has strong thermo-optic effects, but it is too slow to meet the speed requirement for modern telecommunications applications [18]. To date, most modulators based on silicon use the plasma dispersion effect to achieve modulation. The plasma dispersion effect describes the changes of both RI and absorption of silicon in response to the concentration variation of free carriers as follows [19]:

$$\begin{aligned}\Delta n &= -\frac{q^2\lambda^2}{8\pi^2c^2\varepsilon_0 n_0}\left(\frac{\Delta N_e}{m_{ce}^*}+\frac{\Delta N_h}{m_{ch}^*}\right),\\ \Delta\alpha &= \frac{q^3\lambda^2}{4\pi^2c^3\varepsilon_0 n_0}\left(\frac{\Delta N_e}{m_{ce}^{*2}\mu_e}+\frac{\Delta N_h}{m_{ch}^{*2}\mu_h}\right),\end{aligned} \tag{2}$$

where $\triangle n$ is the refractive index change, $\triangle \alpha$ is the absorptive index change, q is the charge on an electron, λ is the wavelength of the incident light, c refers to the speed of light in vacuum, ε_0 refers to the electric constant in vacuum, n_0 refers to the refractive index of intrinsic silicon, $\triangle N\varepsilon$ refers to the concentration variation of electron, $\triangle N_h$ refers to the concentration variation of hole, μ_e refers to electron mobility, and μ_h refers to hole mobility. Besides, $m_{ce}^*=0.26m_0$, which refers to the effective mass of hole, while $m_{ch}^*=0.39m_0$, which refers to the effective mass of electron.

To convert phase modulation of plasma dispersion effect to intensity change, either Mach-Zehnder interferometers (MZI) or resonant structures can be adopted. Limited by its narrow band characteristic, resonant modulators suffer from high susceptibility to fabrication errors and temperature variation. For example, 1 nm increase in the average width of the ring waveguide induces 0.25 nm resonant wavelength shift. Owing to the large thermo-optic coefficient of silicon, resonant modulators are extremely temperature-sensitive. On the other hand, Mach-Zehnder modulators are free from these problems. The performance of Mach-Zehnder modulators are hardly affected by changes in temperature and fabrication process, compared with resonant modulators.

Silicon photodetectors are also one of the essential components in the optical transmission and interconnection system. They convert the incident modulated light into electrical signals, and can be used to either monitor light intensity variations or detect high speed optical signals. In recent years, many high performance photodetectors have been demonstrated [20]-[22].

For long-distance optical transmission applications, the 1.55 μm wavelength range is usually used due to the minimum loss window of silica optical fiber. Consequently, photodetectors working in the 1.55 μm wavelength range have been pursued by researchers ever since. Due to a large bandgap of 1.12 eV, bulk Si material demonstrates a limited maximum absorption wavelength of 1.1 μm. However, in order to make full use of the mature CMOS technology, people also tried to fabricate the hetero-junction photodiodes (PDs) by bonding or epitaxying III-V material on the Si substrate, but their advantages seem not prominent [23].

Ge can demonstrate much higher optical absorption in 1.55 μm wavelength range than silicon, due to the smaller bandgap of 0.67 eV. In addition, Ge, as a group IV material, is the same with silicon; its fabrication process can be compatible with the CMOS technology. Thanks to these advantages, Ge photodetectors are becoming the most promising candidates for Si photonics integration.

Both evanescent [20]-[22] and butt [23] coupling schemes have been used to couple light from silicon waveguides to Ge layers. When integrated with small core waveguides, Ge photodetectors have the advantages of easier power transferring to Ge films and fabrication process in the evanescent coupling case.

The commercial 100 Gb/s optical transmission systems traditionally employ the discrete devices, which leads to low stability and high cost. In this work we present the analysis and design of the silicon Mach-Zehnder modulator and Ge PIN waveguide photodetector, whose performance could meet the requirement of the commercial usage and their design and appearance are aligned with large scale integration. Since PDM-QPSK modulation formats are frequently employed in the 100 Gb/s coherent system, which composed by several modulators and detectors, the goal of this work is to design the devices having high performance at more than 25 Gb/s.

2 Performance Metrics

Several metrics can be used to evaluate the performance of modulators and photodetectors. The most important ones are modulation bandwidth, modulation efficiency $V_\pi L_\pi$, and extinction ratio for modulators. Dark current, responsivity, and 3 dB bandwidth are usually used to describe the performance of photodetectors.

Modulation bandwidth is characterized as the frequency at which the modulation is reduced to 3 dB point. This metric is vital to measure the available information capacity of optical modulator.

$V_\pi L_\pi$ is usually used to evaluate the modulation efficiency of MZI modulators. V_π is the half-wave voltage, and L_π is the length of the modulation arm when the device realize π phase. The smaller $V_\pi L_\pi$ means higher modulation efficiency and more compact footprint.

Extinction ratio is the metric to evaluate the modulation depth. It is defined as the ratio of I_{max}, the maximum transmitted intensity when the modulator works in on-state, to I_{min}, the minimum transmitted intensity when the modulator works in off

-state.

For photodetectors, dark current is the reverse bias current when there is no light incidence on the device. It can reflect the leakage current of the device. The smaller the dark current is, the better performance the device has.

Responsivity is the parameter that tells how the photodetector responses to the incident light and converts it into electrical signals. It (R) can be defined as the ratio of the generated current (I) and the incident power (P) of the light, which can be written as $R=I/P$.

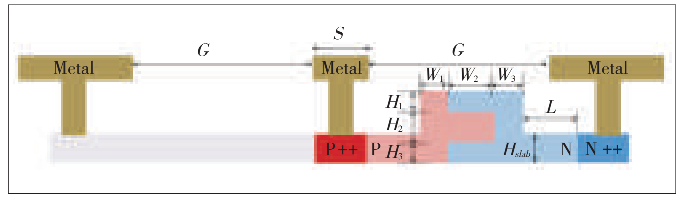
The 3 dB bandwidth is used to describe the device performance in the high speed detection systems, and it is characterized to be the frequency at which the detection responsivity is reduced to the 3 dB point.

In the following two sections, we present the simulation process of the silicon Mach-zehnder modulator and PIN waveguide photodetector which can work well in the commercial 100 Gb/s optical transmission system. For the practical application, all the devices should have 23 GHz bandwidth. The extinction ratioof the modulator is required to be larger than 20 dB, the insertion loss should be less than 5 dB when the driving voltage is 3.3 V_{pp}. The responsivity of photodectector should be larger than 0.65 A/W and the dark current is less than 20 nA under 3.3 V_{pp} driving.

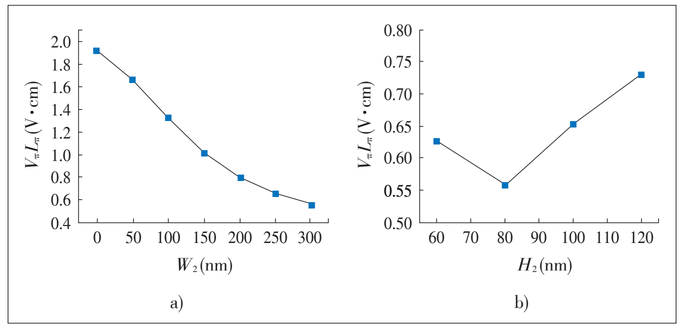
3 Design of Silicon Mach-Zehnder Modulator

For single mode propagation, the height and the width of the rib waveguide are chosen to be 220 nm and 500 nm. Through simulating the above single-mode waveguide with different distances between the high doping and the edge of the rib waveguide (defined as L shown in **Fig. 1**) by COMSOL Multiphysics, the limiting factors of the optical energy in the rib waveguide (Q1) and the limiting factors of the optical energy in the rib waveguide and the L region (Q2) are acquired. We find that the Q1 and Q2 are nearly unchanged when L changes from 400 nm to 1000 nm. For decreasing the absorption loss caused by the high doping area, we choose the L of 1000 nm. Based on the above structure, we then optimize the thickness of the slab, H_{slab} in the same way. At last we chose the H_{slab} of 70 nm.

The modulation efficiency can be improved by increasing the overlapping area between the optical field and the depletion area. The U-type PN junction is a good candidate as shown in Fig. 1. The p-type doping concentration is 1×10^{18} / cm^3, and it is the same for the N-type doping. The parameters W_2 and H_2 have more influence on the modulation efficiency. The change of the modulation efficiency with W_2 and H_2 are illustrated in **Fig. 2**. According to the simulated results, when W_2=300 nm, H_2=80 nm, the PN junction can realize the highest efficiency of $V_\pi L_\pi$=0.559 V·cm. The structure's absorption coefficient α is 14.74 dB/cm. When the driving voltage is 3.3 V, the change of absorption loss is 7.64 dB/cm.



▲**Figure 1. The cross-section diagram of U-type PN junction.**



▲**Figure 2. The curves of modulation efficiency of a) W_2 and b) H_2.**

In an asymmetric device, the modulator is on OFF state with 0 V driving voltage which can be tuned by the working wavelength or the statistic phase shifter, and then is on ON state when the device's driving voltage is on. Considering the limited driving voltage of the practical systems, we assume the driving voltage is 3.3 V_{pp}. Since the absorption coefficients of two arms are different, the insertion loss of the MZI structure can be expressed as follows:

$$IL = 10\log\left(\frac{I^2_{out-on}}{I^2_{in}}\right), \tag{3}$$

$$I_{out-on} = T_{0V}I_1 + T_{3.3V}I_2 + 2\sqrt{T_{0V}T_{3.3V}}\sqrt{I_1I_2}\cos(\Delta\phi) \ . \tag{4}$$

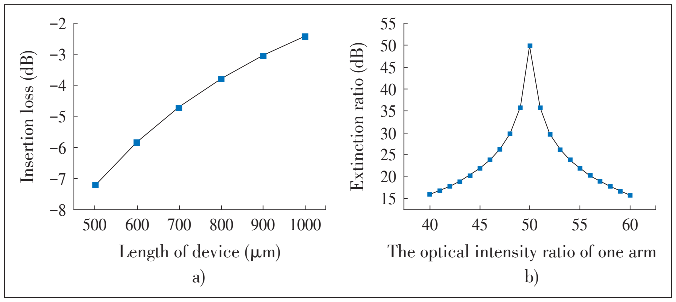
Meanwhile, the extinction ratio can be defined as follows:

$$ER = 10\log\left(\frac{I_{out-on}}{I_{out-off}}\right), \tag{5}$$

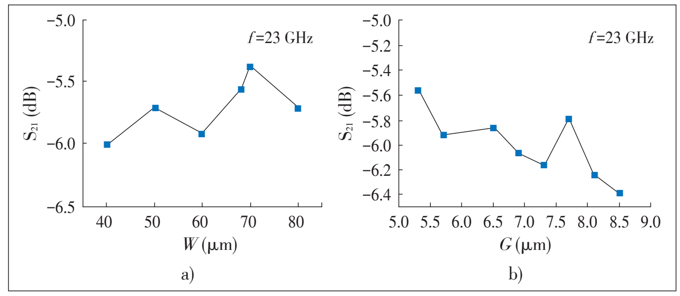
$$I_{out-off} = T_{0V}(I_1 + I_2 - 2\sqrt{I_1I_2}), \tag{6}$$

where I_{in} is the input optical intensity; $I_{out\text{-}on}$ and $I_{out\text{-}off}$ are the output optical power on ON and OFF state; I_1 and I_2 are the optical intensity of the two arms. Besides, T_{0V}, $T_{3.3V}$ are the absorption coefficients on the driving voltage of 0 V and 3.3 V. When the length of the device is larger than 700 μm, the insertion loss of ON state is less than 4.7 dB. The extinction ratio is also influenced by the beam splitting ratio of the two arms, when the beam splitting ratio is better than 45:55 ($I_1:I_2$), the extinction ratio is larger than 20 dB as shown in **Fig. 3**.

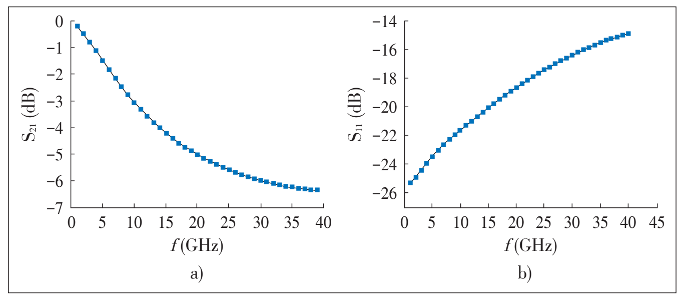
Next, we set up the travelling-wave electrode mode using the commercial simulation software High Frequency Structure Simulator (HFSS). The main parameters of the travelling-wave



▲**Figure 3. a) The insertion loss vs. the length of the modulation arm; b) the extinction ratio vs. the optical intensity ratio of one modulation arm.**



▲**Figure 4. The curves of S_{21} vs. a) *W* and b) *G*.**



▲**Figure 5. The curves of a) S_{21} and b) S_{11} of the designed electrode.**

electrode are the width of the signal line (*W*) and the gap between signal line and ground line (*G*) as shown in Fig. 1.

Since the devices are designed for the 100 Gb/s coherent system, the bandwidth of the modulator is required to be larger than 23 GHz. Considering the insertion loss, the length of the device are chosen to be 700 um. Optimizing the model by parameter sweeping, the change of the S_{21} parameter with *W* and *G* are described in **Fig. 4**. When *W* increases, the microwave loss decreases, which can broaden the bandwidth. Meanwhile, the characteristic impedance decreases with the increasing *W*, and the bandwidth decreases with the impedances' mismatch. When *G* increases, the interaction between signal and ground line decreases and results in less modulation and lower bandwidth.

When *W* is 70 μm and *G* is 5.3 μm, the travelling wave electrode has a lowest loss at 23 GHz. Then, the S parameter of this device was simulated. The S_{21} and S_{11} curves are shown in **Fig. 5**. The 3 dB electro-optical bandwidth of the modulator is nearly the 6 dB electric bandwidth of the travelling-wave electrode. Fig. 5 shows the 6 dB bandwidth of the electrode is roughly 30 GHz.

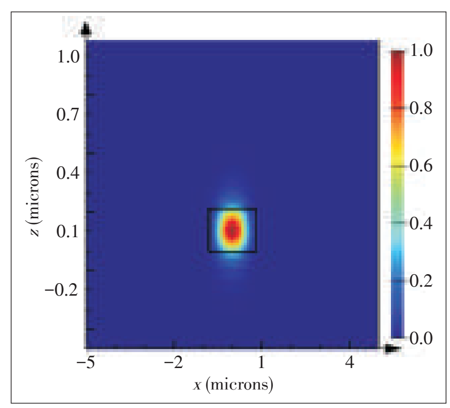


Figure 6.▶ **TE fundamental mode.**

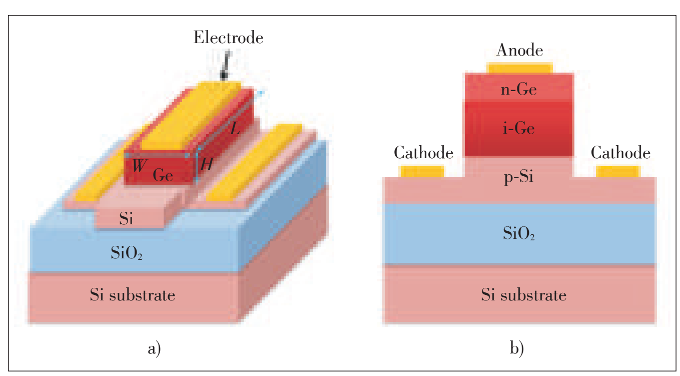
4 Design of PIN Waveguide Photodetector

For single mode propagation, we choose the height and the width of the rib waveguide 220 nm and 600 nm respectively, and the slab thickness is 60 nm. This waveguide can confine the single transverse electric (TE) mode well as **Fig. 6** shows.

The schematic structure and the cross-sectional image of the designed Ge photodetector are shown in **Fig. 7**.

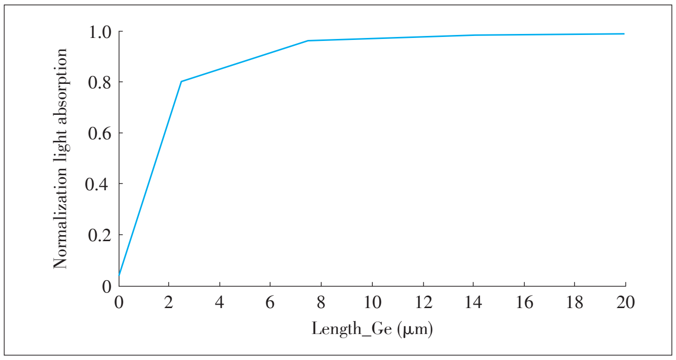
Here we assume the refractive index of the Ge is 4.275 + 0.02835j at C-band. Simulation results show that when the length of the Ge (*L*) reached a certain level, the light absorption would not increase with the length. Therefore we optimized the length of the Ge layer as shown in **Fig. 8**.

When the length reaches 10 um, the light absorption becomes saturated. Since the larger the device area, the smaller the bandwidth limitation, we chose 10 μm as the device length (*L*). Moreover, the width of the Ge layer (*W*) is designed to be 1.6 μm so that high bandwidth can be obtained simultaneously. According to the fabrication situation of the foundry, we follow the same thickness of the Ge layer (*H*) as 500 nm.

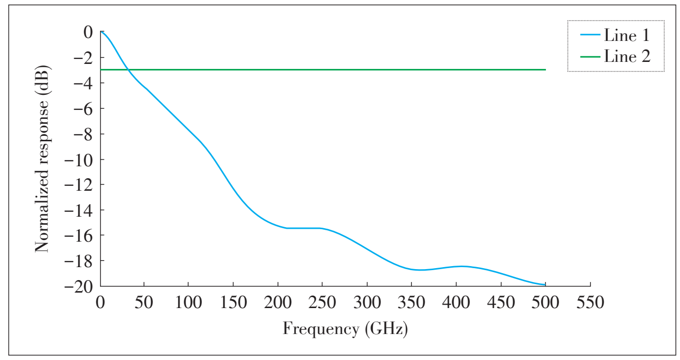


▲**Figure 7. a) The schematic structure and b) the cross-sectional image of the designed Ge photodetector.**

The cathode and anode contacts can be defined in the contacts table and set to simulate a single bias of -1 V in DEVICE module, after importing the photon-generated carriers file gen-

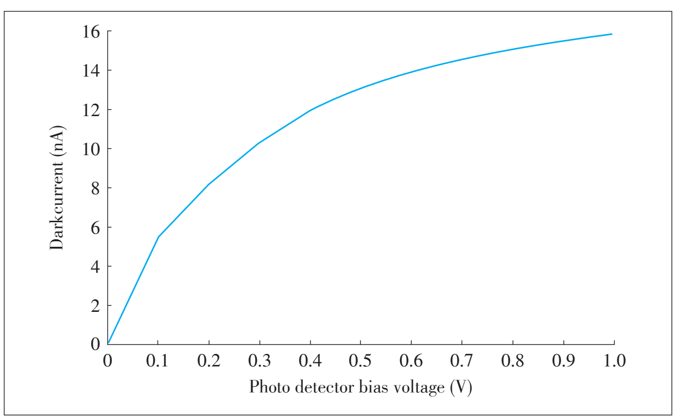


▲**Figure 8. Normalization light absorption vs. the length of Ge characteristic.**

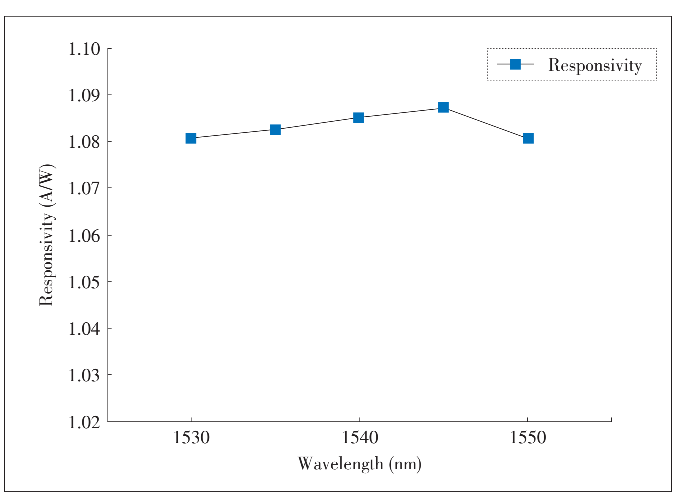


▲**Figure 11. 3 dB bandwidth of the designed Ge photo detector (Line 1) -3 V (Line 2).**

erated by the FDTD simulation. The PIN photo detector is defined by n^{++} type, p^{+} type and p^{++} type implants in the Ge with peak doping density of 1×10^{20} /cm^3, 3×10^{18} /cm^3 and 1×10^{20} /cm^3. **Figs. 9–11** show the dark current of 16 nA at -1 V, responsivity of 1 A/W at -1 V over the C band, and 3 dB bandwidth of 26 GHz at -3 V respectively. The Ge photodetector we have designed can be readily used in the high speed optical



▲**Figure 9. Dark current of the designed Ge photo detector at different reverse biases.**



▲**Figure 10. Spectral responsivity of the designed Ge photo detector.**

transmission systems of 100 Gb/s and beyond.

5 Conclusions

The silicon Mach-Zehnder modulator and PIN waveguide photodetector were analyzed and designed. The modulator with 700 μm long optimized U-type PN junction has high modulation efficiency and bandwidth. The extinction ratio of larger than 20 dB and the insertion loss of less than 5 dB are also acceptable. The PIN waveguide photo detector has high performance and the dark current is only about 16 nA at -1 V, responsivity is more than 1 A/W at -1 V over the C band, and 3 dB bandwidth is more than 26 GHz at -3 V. The performance of the silicon Mach-Zehnder modulator and Ge PIN waveguide photo detector designed can be sufficiently suitable in the commercial 100 Gb/s optical transmission system. The silicon devices bring the highly integrated and low-cost 100 Gb/s optical systems.

Acknowledgements

All the authors would like to acknowledge LI Tiantian (Peking University), TU Zhijuan (Peking University), and ZHANG Qi (ZTE Corporation) for their participation and support to this collaborative project.

References

[1] D. A. B. Miller, "Device requirements for optical interconnects to silicon chips," *Proc. IEEE*, vol. 97, no. 7, pp. 1166–1185, Jul. 2009. doi: 10.1109/JPROC.2009.2014298.

[2] D. Ahn, C. Hong, J. Liu, et al., "High performance, waveguide integrated Ge photo detectors," *Optics Express*, vol. 15, no 7, pp. 3916–3921, 2007. doi: 10.1364/OE.15.003916.

[3] Q. Xu, B. Schmidt, S. Pradhan, and M. Lipson, "Micrometre-scale silicon electro-optic modulator," *Nature*, vol. 435, no. 7040, pp. 325–327, 2005. doi: 10.1038/nature03569.

[4] B. Jalali and S. Fathpour, "Silicon photonics," *Journal of Lightwave Technology*, vol. 24, no. 12, pp. 4600–4615, 2006. doi: 10.1109/JLT.2006.885782.

[5] G. T. Reed, G. Mashanovich, F. Y. Gardes, and D. J. Thomson, "Silicon optical modulators," *Nature Photonics*, vol. 4, no. 8, pp. 518–526, Jul. 2010. doi: 10.1038/nphoton.2010.179.

[6] A. E. J. Lim, J. Song, F. Qing, C. Li, et al., "Review of silicon photonics foundry efforts," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 20, no. 4, pp. 405–416, 2014. doi: 10.1109/JSTQE.2013.2293274.

[7] A. R. Chralyvy, "The coming capacity crunch," in *35th European Conference on Optical Communication*, Vienna, Austria, 2009.
[8] R. W. Tkach, "Scaling optical communications for the next decade and beyond," *Bell Labs Technical Journal*, vol. 14, pp. 3–10, 2010. doi: 10.1002/bltj.20400.
[9] A. D. Ellis, F. C. G. Gunning, B. Cuenot, T. C. Healy, and E. Pincemin, "Towards 1TbE using coherent WDM," in *Joint Conference of the Opto-Electronics and Communications Conference and the Australian Conference on Optical Fibre Technology*, Sydney, Australia, 2008. doi: 10.1109/OECCACOFT.2008.4610550.
[10] P. J. Winzer and R. Essiambre, "Advanced optical modulation formats," *Proceedings of the IEEE*, vol. 94, no. 5, pp. 952–985, May 2006. doi: 10.1109/JPROC.2006.873438.
[11] K. Kikuchi, "Digital coherent optical communication systems: fundamentals and future prospect," *IEICE Electron. Express*, vol. 8, no. 20, pp. 1642–1662, 2011. doi: 10.1587/elex.8.1642.
[12] H. Yi, Q. Long, W. Tan, et al., "Demonstration of low power penalty of silicon Mach-Zehnder modulator in long-haul transmission," *Optics. Express*, vol. 20, no. 25, pp. 27562–27568, 2012. doi: 10.1364/OE.20.027562.
[13] T. Li, J. Zhang, H. Yi, et al., "Low-voltage, high speed, compact silicon modulator for BPSK modulation," *Optics. Express*, vol. 21, no. 20, pp. 23410–23415, 2013. doi: 10.1364/OE.21.023410.
[14] T. Li, J. Zhang, H. Yi, et al., "10-Gb/s 53.1-km BPSK transmission of silicon Mach-Zehnder modulator," in *Asia Communications and Photonics Conference*, Beijing, China, 2013. doi: 10.1364/ACPC.2013.AW4A.3.
[15] T. Li, D. Wang, J. Zhang, et al., "Demonstration of 6.25 Gbaud advanced modulation formats with subcarrier multiplexed technique on silicon Mach-Zehnder modulator," *Optics Express*, vol. 22, no. 16, pp. 19818– 19823, 2014. doi: 10.1364/OE.22.019818.
[16] M. Cignoli, G. Minoia, M. Repossi, et al., "22.9 A 1310 nm 3D-integrated silicon photonics Mach - Zehnder - based transmitter with 275 mW multistage CMOS driver achieving 6 dB extinction ratio at 25 Gb/s," in *IEEE International Solid-State Circuits Conference*, San Francisco, USA, 2015. doi: 10.1109/ISSCC.2015.7063103.
[17] M. Nedeljkovic, R. A. Soref, and G. Z. Mashanovich, "Free-carrier electro-absorption and electro-refraction modulation in group IV materials at mid-infrared wavelengths," in *Proc. SPIE 8266 of Silicon Photonics VII*, San Francisco, USA, 2012. doi:10.1117/12.908650.
[18] J. F. Liu, D. Pan, S. Jongthammanurak, et al., "Design of monolithically integrated GeSi electro-absorption modulators and photo detectors on a SOI platform," *Optics Express*, vol. 15, no. 2, pp. 623– 628, 2007. doi: 10.1364/OE.15.000623.
[19] G. T. Reed and A. P. Knights, *Silicon Photonics: An Introduction*. Chichester, England: John Wiley & Sons, 2004. doi: 10.1002/0470014180.
[20] C. T. DeRose, D. C. Trotter, W. A. Zortman, et al., "Ultra compact 45 GHz CMOS compatible Germanium waveguide photodiode with low dark current," *Optics Express*, vol. 19, no. 25, pp. 24897– 24904, 2011. doi: 10.1364/OE.19.024897.
[21] L. Vivien, A. Polzer, D. M. Morini, et al., "Zero-bias 40 Gbit/s germanium waveguide photo detector on silicon," *Optics Express*, vol. 20, no. 2, pp. 1096–1101, 2012. doi:10.1364/OE.20.001096.
[22] L. P. Virot, L. Vivien, J. M. Fedeli, et al., "High-performance waveguide-integrated germanium PIN photodiodes for optical communication applications," *Photon. Research*, vol. 1, no. 3, pp. 140 – 147, 2013. doi: 10.1364/PRJ.1.000140.
[23] Z. Tu, K. Liu, H. Yi, et al., "A Compact Evanescently-coupled Germanium PIN Waveguide Photo detector," in *Proc. SPIE* 8564 *of Nanophotonics and Micro/Nano Optics*, Beijing, China, 2012. doi:10.1117/12.2001221.

Manuscript received: 2016-12-12

Biographies

ZHONG Fang (fzhong@pku.edu.cn) received the B.E. degree in materials engineering from Peking University, China in 2015. Since 2015, he has been working for his master degree at the State Key Laboratory of Advanced Optical Communication Systems and Networks, School of Electronics Engineering and Computer Science, Peking University, China. His current research interests include silicon based modulators and monolithic integration of high speed transmitters for optical communication systems.

GONG Pan (pankung@pku.edu.cn) received the B.E. degree in electronic from Hebei University of Engineering, China in 2012. He has obtained his master degree from the School of Software and Microelectronics, Peking University, China. His research interests include silicon-based Germanium photodetectors and monolithic integration of high speed coherent receivers for optical communication systems.

ZHOU Zhiping (zjzhou@pku.edu.cn) received the Ph.D. degree in electrical engineering from Georgia Institute of Technology (GT), USA in 1993. From 1993 to 2005, he was with the Microelectronics Research Center at GT, where he engaged research and development in the areas of nanotechnology, integrated optoelectronics, nanophotonic devices, and semiconductor devices. He is currently a Changjiang Professor at Peking University, China, focusing on silicon photonics and microsystems research. He has been credited for more than 360 technical papers and presentations, and 15 patents. He was the Chair of IEEE Wuhan Section, 2007–2008, the Director of IEEE Atlanta Section, 2001– 2003. He also chaired, cochaired, and served on many program committees for various conferences for OSA, IEEE Photonics Society, and SPIE. Dr. Zhou is a Fellow of OSA, SPIE, and IET, and a Life Member of PSC. He serves as the Editor-in-Chief of *Photonics Research* and is on OSA Board of Editors.

HUA Feng (hua.feng@zte.com.cn) received her master degree in optical instrument from Tianjin University, China in 1996. She has worked with ZTE Corporation since 2000. Currently she is a senior engineer focusing on advanced research of cutting-edge optical communication technologies include silicon photonics, SDM, and optical backplane. She has more than 10 patents.

WANG Huitao (wang.huitao@zte.com.cn) received the B.E. degree in electronic science and technology from Harbin Engineering University, China in 2000 and the M.S. degree in physical electronics from Beijing Insititute of Technology, China in 2003. He worked at ZTE Corporation from 2003 to 2017 and was responsible for the technical planning of optical network system and optoelectronics devices. He has been working with ZTE Photonics since 2017, and is committed to the research and planning of high-speed optoelectronic devices in optical communications.

DOI: 10.3969/j.issn.1673-5188.2017.03.007
http://kns.cnki.net/kcms/detail/34.1294.tn.20170824.1424.004.html, published online August 24, 2017

Practical Pattern Recognition System for Distributed Optical Fiber Intrusion Monitoring Based on Φ-COTDR

CAO Cong, FAN Xinyu, LIU Qingwen, and HE Zuyuan
(State Key Laboratory of Advanced Optical Communication Systems and Networks, Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract

At present, the demand for perimeter security system is increasing greatly, especially for such system based on distributed optical fiber sensing. This paper proposes a perimeter security monitoring system based on phase-sensitive coherent optical time domain reflectometry(Φ-COTDR) with the practical pattern recognition function. We use fast Fourier transform (FFT) to exact features from intrusion events and a multi-class classification algorithm derived from support vector machine (SVM) to work as a pattern recognition technique. Five different types of events are classified by using a classification algorithm based on SVM through a three-dimensional feature vector. Moreover, the identification results of the pattern recognition system show that an identification accurate rate of 92.62% on average can be achieved.

Keywords

fiber optics sensors; COTDR; distributed vibration sensing; SVM; pattern recognition

1 Introduction

Nowadays, phase-sensitive optical time domain reflectometry (Φ-OTDR) is widely used in optical fiber perimeter security monitoring system for its distinguished advantages over other technologies [1]. Nevertheless, Φ-OTDR has not been well performed in real applications. In Φ-OTDR system, the trace for analysis is an optical time domain signal, which results from coherent addition of amplitudes of lightwave backscattered from different positions of the fiber [2]. When disturbance occurs at a certain position, the change of the length and the refractive index of the fiber may happen, which will change the phase at that position. Accordingly, the trace mentioned above changes. However, in the real environment, the sensing signal trace is easily interfered by noise sources. Different disturbing events even have the same effect on the trace [3], causing false alarms frequently, which are intolerable for some practical applications. The most important challenge for us is to distinguish the events that we should prevent from all the disturbing ones.

For this purpose, we designed a practical pattern recognition system for distributed optical fiber intrusion system based on phase-sensitive coherent optical time domain reflectometry (Φ-COTDR). The reason for choosing Φ-COTDR is to get a higher Signal Noise Ratio (SNR) and longer monitoring distance [4]. There are two classification levels in the practical pattern recognition system. Along fiber intrusion monitoring channels, the first level classification identifies all intruded channels, while the intruded channels are further analyzed to get the types of intrusion events at the secondary classification level. The sensing signal for analysis is gathered by accumulating the varying trails at different moments at each channel [5]. In our experiment, Event A (a stable state), Event B (walking on the lawn while the fence is exposed to the wind), Event C (shaking the fence), Event D (walking on the lawn) and Event E (vibration exciter) are tested. Event A is used to verify the effectiveness of the intrusion detection (the first level classification). After careful signal analysis, some features are extracted and formed as a feature vector in support vector machine (SVM) for pattern recognition. The experiment results show that the pattern recognition system has a robust performance.

2 Pattern Recognition System for Optic-Fiber Fence Based on Φ-COTDR

The practical pattern recognition system consists of the classifier training and testing stages (**Fig. 1**). In the classifier training stage, the sensing signals we obtained averagely achieve 25 times improvement in SNR. After signal analysis, some features are selected and they are quite important for the subsequent event classification task. The intrusion learning is carried out firstly, which will trigger an alarm if intrusion happens. Meanwhile, the invaded channels are located and the system can preserve the channel information for event classification. This process relieves the data stress of calculating all channels. Then, some SVM classifier models are preserved for testing. In the testing stage, the system can realize intrusion detection and classify the events effectively.

2.1 Functional Requirements

For the optical fiber intrusion monitoring system based on Φ-COTDR, the sensing signal traces are easily interfered by environmental noise, temperature change and other disturbing

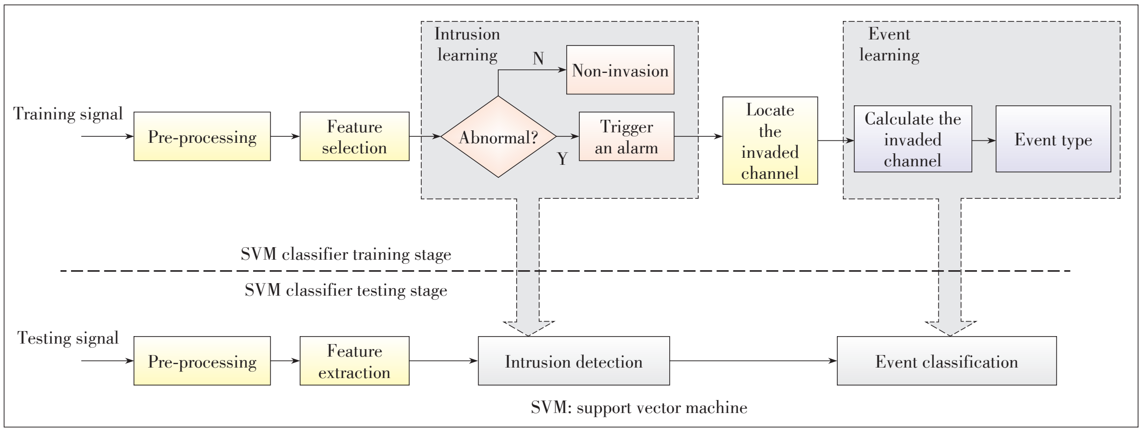


Figure 1. The algorithm flow of the pattern recognition system.

events [6]. In real applications, different events monitored by the system have the similar effect on the Φ-COTDR trace, so that the collected sensing signals show some similar characteristics. Therefore, it is hard for the system to detect the intrusion events and classify them correctly. Moreover, generalization ability is vital to the pattern recognition system, in other words, the system should perform well in the sensing signal for training the classifiers, as well as in testing the new sample. For the real application of the system, the surroundings differ greatly and the data set we collect may show different characteristics. Certain signal processing methods and classification algorithms are adopted depending on the actual features of the data. For the offline training and testing in this paper, depending on the characteristics of the data, we applied fast Fourier transform (FFT) and a classification algorithm derived from SVM to make the system more practical.

2.2 Signal Processing and Feature Selection

At each channel along the sensing fiber, we accumulated the Φ-COTDR traces gathered at different moments to construct a series of temporal signals. The sampling rate of the temporal signal is 400 Hz, the spatial Φ-COTDR traces are sampled at 250 MS/s. After averaging, we got a 400 × 5000 matrix for each sample. As we know, there are rich information in frequency domain of signals. For the time domain signals, we subtracted the average value and applied FFT to get the spectrum of each channel. The total energy, the energy of low frequency, the peak value and the mean value of the spectrum were all calculated for each position. After careful selection, Feature I (energy ratio of a low frequency to total energy), Feature II (total energy), Feature III (ratio of peak value to mean value) are formed as a feature vector in SVM. The normalized method is also applied on the feature vectors. We calculated the feature vectors of 20 samples for each events randomly, and the distributions of Feature I, Feature II and Feature III for 5 events are illustrated in **Fig. 2**. From Fig. 2 a , the vibration exciter is the least and the stable state takes the largest position, which is consistent with the fact and they can be classified effectively. Feature III makes a great contribution to the events classification, which can be seen from Fig. 2c . Certain samples of Event B and Event D are mixed, which means that the identification rate between them may not be so good.

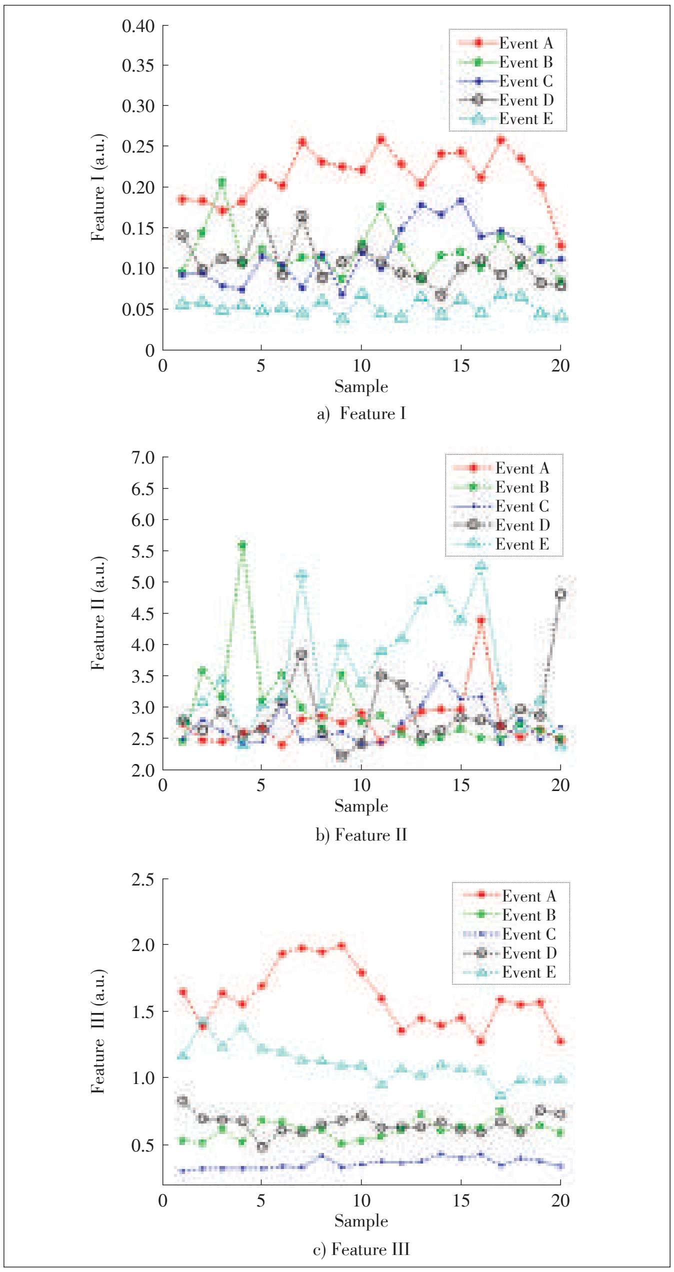
2.3 Support Vector Machine

Nowadays many pattern recognition techniques are applied to optical fiber sensors, especially in distributed vibration sensing [7]. SVM is a powerful and popular supervised learning method based on statistiacal learning theory in machine learning. As shown in Fig. 3, when linear decision hyper-planes are no longer feasible, by using a kernel function, the input space (x_1, x_2, x_3..., x_n) is mapped to a high-dimensional feature space and an optimal margin hyper-plane (OMHP) will be found in the mapped space. The rectangles in dark grey in **Fig. 3** work as a kernel function and stand for support vectors, and $K(x_i,x)$ means the inner product between the input space and the support vector. The output y is a linear combination of the inner products. The way to get the OMHP can be converted into a solution to a quadratic programming (QP) problem, so the solution to an SVM is global and unique [8]. The SVM classification algorithm has a robust generalization ability. On the contrary, artificial neural network (ANN) is a classification method based on empirical risk minimization. Its performance partly depends on the empirical knowledge and prior knowledge of the designer, which cannot ensure its generalization ability [9]. Generally, the back propagation (BP) ANN only converges to locally optimal solutions, while SVM always finds a global optimum value [10]. Neither ANN nor SVM is perfect. However, the SVM is more suitable for our requirements in the practical fiber fence monitoring system, especially in the real scenarios. Therefore, we choose a multi-class classification algorithm based on SVM as a learning method in this paper.

3 Experimental Results and Discussion

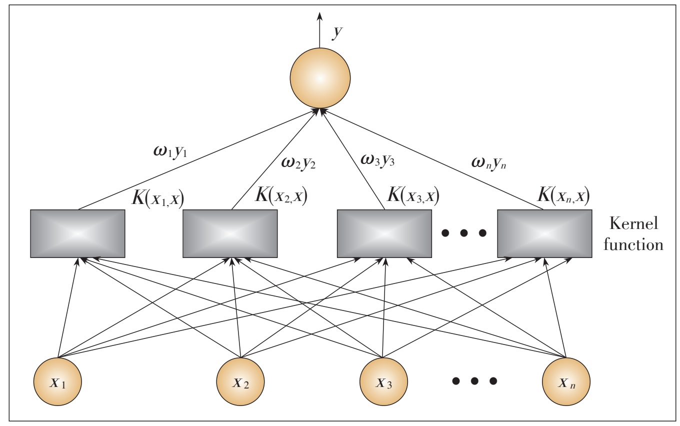
3.1 Overall Optic-Fiber System

The proposed configuration of Φ-COTDR system for evaluating the intrusion monitoring system is shown in **Fig. 4**. An ultra



▲ **Figure 2. The features distribution of the five events.**

-narrow linewidth (<0.1 KHZ) laser (UNLL) with maximum output power of 40 mW is used. The lightwave from the UNLL is split into two arms by using a coupler. The lightwave from one arm of the coupler is modulated by the acoustic-optical modulator (AOM) that is driven by a function generator. A series of modulated pulses generated from the AOM are amplified by the erbium-doped fiber amplifier (EDFA), and then launched into the 50 km sensing fiber through a circulator. The backscattered Rayleigh lightwave produced in the sensing fiber is amplified by using another EDFA. Another arm of the coupler is functioned as a local oscillator and interferes with the backscattered Rayleigh lightwave from the sensing fiber. We use a



▲ **Figure 3. The model of the SVM.**

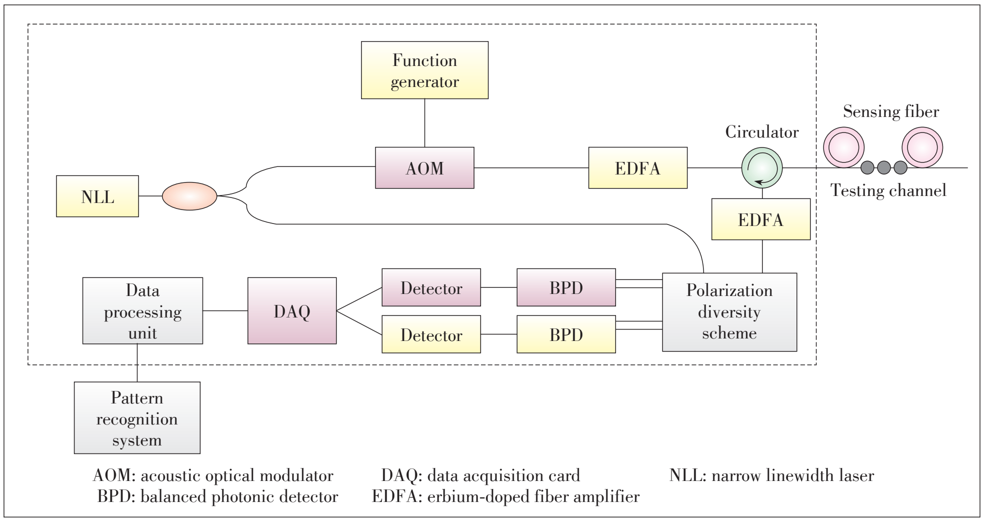
polarization-diversity scheme to avoid the effect of polarization-related problems on coherent detection. The raw Φ-COTDR signal traces are sampled by a data acquisition card (DAQ) (250 MS/s). The signal is processed further in the data processing unit and then transmitted to the pattern recognition system. The triggering pulse repetition rate is set at 400 Hz, which means that the vibration response range is 200 Hz. There are 5000 channels along the sensing fiber, the testing fence is set at the 2038th channel, the lawn is set at the 2035th channel, and the vibration exciter is set at the 2033th channel located at 20.38 km, 20.35 km, and 20.33 km respectively. Five events above are tested in these channels. Especially, each human intrusion event was conducted by different persons to increase the diversity of samples so as to improve the robustness of the pattern recognition system.

3.2 Intrusion Detection and Event Classification Results

To verify the performance of the pattern recognition system, 650 samples are used in the SVM training stage, including 150 Event A, 100 Event B, 150 Event C, 100 Event D, and 150 Event E samples. For the testing stage, a testing sample set that contains 150, 100, 150, 100, 150 for each event is constructed. After pre-processing and feature extraction, a series of feature vectors are put into the SVM. Radial basis function (RBF) is used as the kernel function. By cross validation, we can get some robust SVM classifier models to classify different events. The results are illustrated in **Table 1**. The average identification accurate rate (AIAR), intrusion detection rate (IDR), event classification rate (ECR) and the identification accurate rate for each type (IARE) are calculated for further analysis. The AIAR achieved for the five events is 92.62%. The IDR can be up to 98.6%, which is significant to real application. The ECR of the system is 91.2%. All in all, the test results strongly prove the robustness of the proposed system.

4 Conclusions

A practical pattern recognition system for distributed fiber-



▲**Figure 4. The experimental setup of Φ-COTDR system.**

▼**Table 1. The identification results of the pattern recognition system**

	Event A	Event B	Event C	Event D	Event E	IARE	AIAR	IDR	ECR
Event A	146	0	0	0	4	97.33%			
Event B	4	91	0	4	1	91%			
Event C	0	0	148	2	0	98.67%	92.62%	98.6%	91.2%
Event D	1	29	0	69	1	69%			
Event E	2	0	0	0	148	98.67%			

AIAR: average identification accurate rate
ECR: event classification rate
IARE: identification accurate rate for each type
IDR: intrusion detection rate

optical intrusion monitoring system based on Φ-COTDR is proposed. It is proved quite effective for intrusion detection, by classifying events accurately and reducing the nuisance alarm rate greatly. The experimental indoor test proves the system robustness. In the future, more features will be extracted to further improve the robustness of the SVM classifier, more events will be classified and more field tests in specific outside environments will be carried out. We plan to put more emphases on the function of self-adaption for the pattern recognition system, which is a more challenging work. We believe the Φ-COTDR will have a bright future with the help of the practical pattern recognition system.

References

[1] J. C. Juarez, E. W. Maier, K. N. Choi, and H.F. Taylor. "Distributed fiber-optic intrusion sensor system," *Journal of Lightwave Technology*, vol. 23, no. 6, pp. 2081–2087, Jun. 2005. doi: 10.1109/JLT.2005.849924.

[2] J. C. Juarez, H. F. Taylor. "Field test of a distributed fiber-optic intrusion sensor system for long perimeters," *Applied Optics*, vol. 46, no. 11, pp. 1968– 1971, Apr. 2007. doi: 10.1364/AO.46.001968.

[3] H. J. Wu, X. Y. LI, Z. P. PENG, and Y. J. Rao. "A novel intrusion signal processing method for phase-sensitive optical time-domain reflectometry (Φ-OTDR)," in *Proc. SPIE 9157 23rd International Conference on Optical Fibre Sensors*, Jun. 2014, pp. 1–4, doi:10.1117/12.2058503.

[4] Y. L. Lu, T. Zhu, L. Chen, and X. Y. Bao, "Distributed vibration sensor based on coherent detection of phase OTDR," *Journal of Lightwave Technology*, vol. 28, no. 22, pp. 3243– 3249, Sept. 2010. doi: 10.1109/JLT.2010.2078798.

[5] H. J. Wu, S. K. Xiao, X. Y. Li, et al., "Separation and determination of the disturbing signals in phase- sensitive optical time domain reflectometry (Φ- OTDR)," *Journal of Lightwave Technology*, vol. 33, no. 15, pp. 3156– 3162, Apr. 2015. doi: 10.1109/JLT.2015.2421953.

[6] H. Wu, J. Wang, X. Wu, and Y. Wu. "Real intrusion detection for distributed fiber fence in practical strong fluctuated noisy backgrounds," *Sensor Letters*, vol. 10, no.7, pp. 1557– 1561, Jul. 2012. doi:10.1166/sl.2012.2485.

[7] W. B. Lyons, E. Lewis. "Neural networks and pattern recognition techniques applied to optical fibre sensors". *Transactions of the Institute of Measurement and Control*, vol. 22, no. 5, pp. 385– 404, Dec. 2000. doi: 10.1191/014233100701523882.

[8] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121– 167, Jun. 1998. doi: 10.1023/A:1009715923555.

[9] A. K. Jain, R. P. W. Duin, J. Mao. "Statistical pattern recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, Jan. 2000. doi: 10.1109/34.824819.

[10] J. Li. *An empirical comparison between SVMs and ANNs for speech recognition* [Online]. Available: http://www.cs.rutgers.edu/~mlittman/courses/ml03/ iCML03/papers/li.pdf.

Manuscript received: 2016-01-01

Biographies

CAO Cong(caocong@sjtu.edu.cn) received his B.S. degree in communication engineering from Hohai University, China in 2014; M.S. degree in electrical engineering from Shanghai Jiao Tong University, China in 2017. His research interests are optical fiber sensing technology, signal processing, and pattern recognition.

FAN Xinyu(fan.xinyu@sjtu.edu.cn) received his B.S. and M.S. degrees in applied physics from Shanghai Jiao Tong University, China in 2000 and 2003, respectively; Ph.D. degree in electrical engineering from the University of Tokyo, Japan in 2006. In 2006, he joined NTT Access Network Service Systems Laboratories, Japan, where he has been engaged in research work mainly on optical reflectometry and optical measurement. Since 2012, he joined the Department of Electrical Engineering, Shanghai Jiao Tong University, China, where he is currently a professor. His research interests are optical fiber sensing technology, optical fiber measurement, and fiber-related applications.

LIU Qingwen(liuqingwen@sjtu.edu.cn) received his B.S. and the M.S. degrees in opto-electronics from Tianjin University, China; Ph.D. degree in electronic engineering from the University of Tokyo, Japan. In 2013, he joined the Department of Electrical Engineering, Shanghai Jiao Tong University, where he is currently an associate professor. His research interests are high resolution fiber grating sensors, distributed optical fiber sensors, and photonic sensors.

HE Zuyuan(zuyuanhe@sjtu.edu.cn) received his B.S. and M.S. degrees in electronic engineering from Shanghai Jiao Tong University, China in 1984 and 1987; Ph.D. degree in optoelectronics from the University of Tokyo, Japan in 1999. He joined Nanjing University of Science and Technology, China, as a research associate in 1987, and became a lecturer in 1990. From 1995 to 1996, he was a research fellow with the University of Tokyo, and became a research associate in 1999. In 2001, he joined CIENA Corporation(USA) as a lead engineer. He was with the University of Tokyo as a lecturer in 2003, then became an associate professor in 2005.He is currently a professor of the Center of Excellence in Electrical and Electronic Engineering, University of Tokyo. Since 2012, he has been with Shanghai Jiao Tong University as a Distinguished Professor. His current research interests are optical fiber sensors, optical fiber measurement, and optical information processing.

DOI:10.3969/j.issn.1673-5188.2017.03.008
http://kns.cnki.net/kcms/detail/34.1294.TN.20170904.0922.002.html, published online September 4, 2017

Layered ACO-FOFDM for IM/DD Systems

GUO Mengqi, ZHOU Ji, TANG Xizi, and QIAO Yaojun
(Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract

We propose a layered asymmetrically clipped optical fast orthogonal frequency division multiplexing (ACO - FOFDM) scheme for intensity-modulated and direct-detected (IM/DD) systems. Layered ACO-FOFDM can compensate the weakness of conventional ACO-FOFDM in low spectral efficiency. For FOFDM system, the utilization of discrete cosine transform (DCT) instead of fast Fourier transform (FFT) can reduce the computational complexity without any influence on bit error rate (BER) performance. At transmitter, the superposition of multiple layers is performed in frequency domain, and the iterative receiver is used to recover transmitted signals by subtracting the clipping noise of each layer. We compare the BER performance of the proposed layered ACO-FOFDM system and DC-offset FOFDM (DCO-FOFDM) system with optimal DC-bias at the same spectral efficiency. Simulation results show that in terms of optical bit energy to noise power ratio, the layered ACO-OFDM system has 1.23 dB, 2.77 dB, 3.67 dB and 0.78 dB improvement at the forward error correction (FEC) limit compared with DCO-FOFDM system when the spectral efficiencies are 1 bit/s/Hz, 2 bits/s/Hz, 3 bits/s/Hz and 4 bits/s/Hz. The layered ACO-FOFDM system with zero DC-bias is more suitable for adaptive system, so this system also has potential for application in IM/DD systems.

Keywords

FOFDM; ACO-FOFDM; DCT; spectral efficiency; IM/DD

1 Introduction

The demand for system capacity increases rapidly in optical communication systems. As a multicarrier modulation format, orthogonal frequency division multiplexing (OFDM) has been widely investigated to meet the capacity demand. Its superiorities in high spectral efficiency and robustness against chromatic dispersion and polarization-mode dispersion make it more suitable for high-speed optical communication systems [1]–[4]. Intensity-modulated and direct-detected (IM/DD) technique has been applied in data center interconnections, passive optical network and indoor optical wireless communications due to its advantages in low cost and simple structure [5]–[7]. Therefore, IM/DD optical OFDM system is an attractive technique for both high-speed and short-range optical transmission systems.

The signal transmitted in IM/DD OFDM system must be real and positive. To obtain a real signal, Hermitian symmetry is needed to constrain the input constellations of inverse fast Fourier transform (IFFT). DC-offset OFDM (DCO-OFDM) and asymmetrically clipped optical OFDM (ACO-OFDM) systems are most commonly used to generate positive signals [8], [9]. The performance of DCO-OFDM strongly depends on DC-bias. If the DC-bias is not large enough, the negative values are clipped to zero, which leads to clipping distortion. If the DC-bias is larger than the negative peaks, the system can be inefficient in optical power. The optimal DC-bias depends on the constellation size, which makes DCO-OFDM system not suitable for adaptive system. For ACO-OFDM system, only odd subcarriers are used to carry the signal, zero DC-bias is applied and clipping noise falls on even subcarriers. Although zero DC-bias can reduce the optical power, half of the subcarriers are useless, which results in the spectral efficiency of ACO-OFDM system to be half of that of DCO-OFDM system.

To improve spectral efficiency of ACO-OFDM system, different types of schemes have been proposed. Asymmetrically clipped DC biased optical OFDM (ADO-OFDM) system transmits ACO-OFDM on odd subcarriers and DCO-OFDM on even subcarriers simultaneously, but the even subcarriers still need DC-bias [10]. Hybrid ACO-OFDM uses ACO-OFDM on odd subcarriers and pulse-amplitude-modulation discrete-multitone (PAM-DMT) on even subcarriers, but the real components of even subcarriers are useless [11]. Asymmetrically and symmetrically clipping optical OFDM (ASCO-OFDM) uses two frames of ACO-OFDM with the useful signal on odd subcarriers and one frame of flip-OFDM with the useful signal on even subcarriers [12], which has the same spectral efficiency as Hybrid ACO-OFDM. Afterwards, to improve the spectral efficiency further the superposition of more than two layers are proposed. Layered ACO-OFDM [13] and enhanced asymmetrically clipped optical OFDM (eACO-OFDM) [14] overlap multiple layers with ACO-OFDM in time domain. In ACO-OFDM system, the scaling factors are used to adjust the average power for each layer, which improves the bit error rate (BER) performance at the cost of implementation complexity. Spectrally and energy efficient OFDM (SEE-OFDM) [15] and augmented spectral efficiency discrete multitone (ASE-DMT) [16] overlap multiple layers in frequency domain instead of time domain, which is easier for implementation. SEE-OFDM system employs superposition of multiple layers with ACO-OFDM. ASE-

This work was supported in part by National Natural Science Foundation of China under Grant Nos. 61427813 and 61331010, and in part by ZTE Industry-Academia-Research Cooperation Funds.

DMT uses PAM-DMT on imaginary components of all the subcarriers and layered ACO-OFDM on the remaining real components.

Fast OFDM (FOFDM) based on discrete cosine transform (DCT) has been studied in IM/DD optical communication systems [17]– [19]. In IM/DD system, the transmitted signal should have real values. For fast Fourier transform (FFT) based OFDM system, Hermitian symmetry is needed to generate real signals, but in DCT-based FOFDM system, the real transformation does not need Hermitian symmetry any more, and the one-dimensional modulation has lower computational complexity. For FOFDM system, the interval between subcarriers is half of that in OFDM system. However, the positive frequency of FOFDM system has corresponding image on negative frequency, so the M-PAM FOFDM system has the same spectral efficiency as M^2quadrature - amplitude - modulation (QAM) OFDM system.

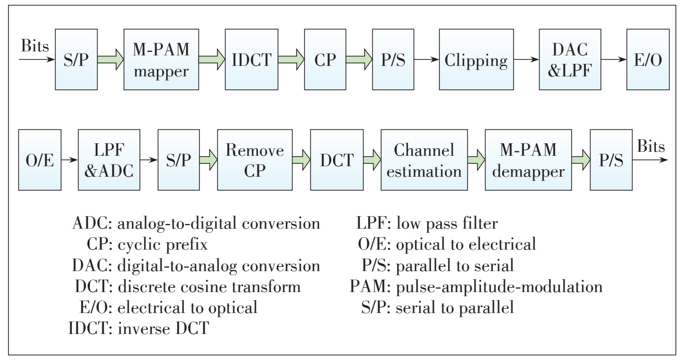
In this paper, we first propose layered FOFDM for IM/DD systems. The superposition of multiple layers is performed in frequency domain, and the iterative receiver is used to subtract the clipping noise of each layer. The spectral efficiency improvement of layered asymmetrically clipped optical fast orthogonal frequency division multiplexing (ACO-FOFDM) system is compared with conventional ACO-FOFDM system and the computational complexity superiority of layered ACO-FOFDM system is investigated. The hard decision is used in the iterative receiver to improve the BER performance. Compared with the DC-offset FOFDM (DCO-FOFDM) system with optimal DC-bias and the same spectral efficiency, layered ACO-FOFDM has better BER performance in terms of optical bit energy to noise power ratio.

The rest of this paper is organized as follows. We discuss the principle of ACO-FOFDM in Section 2. Section 3 describes the detail structures of transmitter and receiver of layered ACO-FOFDM system. In Section 4, we analyze the spectral efficiency, computational complexity and BER performance of layered ACO-FOFDM system. The conclusions are drawn in Section 5.

2 Principle of ACO-FOFDM in Optical Transmission System

FOFDM system based on DCT has been investigated in detail in [19]. Compared with ACO-OFDM, ACO-FOFDM has the same spectral efficiency, power efficiency and BER performance but with lower computational complexity for digital signal processing [18], [19]. **Fig. 1** shows the block diagram of ACO-FOFDM transmitter and receiver. Different from the conventional ACO-OFDM system based on FFT, ACO-FOFDM system uses inverse DCT (IDCT) at the transmitter and DCT at the receiver. The N-point IDCT and DCT are defined as

$$x_n = \sqrt{\frac{2}{N}}\sum_{k=0}^{N-1} W_k X_k \cos\left(\frac{\pi(2n+1)k}{2N}\right), 0 \leqslant n \leqslant N-1, \quad (1)$$



▲**Figure 1. Block diagram of ACO-FOFDM transmitter and receiver.**

$$X_k = \sqrt{\frac{2}{N}} W_k \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi(2n+1)k}{2N}\right), 0 \leqslant k \leqslant N-1, \quad (2)$$

where $W_k = \begin{cases} \frac{1}{\sqrt{2}}, & k=0 \\ 1, & k=1,2,\cdots,N-1 \end{cases}$.

Since IDCT is a real transform, when the real-valued frequency domain M-PAM signal X inputs IDCT, the generated time domain signal is also real-valued. The M-PAM signal X only consists of odd components, which can be expressed as $X = [0, X_1, 0, X_3, ..., X_{N-1}]$.

After IDCT, cyclic prefix (CP) is added to eliminate the intersymbol interference (ISI) after transmission. Then, parallel to serial conversion is performed and all the negative parts of time-domain signal are clipped at zero level. The clipped FOFDM symbol $[X_n]_c$ can be described as

$$[x_n]_c = \begin{cases} x_n, & x_n > 0 \\ 0, & x_n \leqslant 0 \end{cases}, \quad 0 \leqslant n \leqslant N-1, \quad (3)$$

If only odd subcarriers are used to carry the signal, the negative parts of the time domain signal can be clipped at zero level without loss of information. The only influence is the amplitudes of the received signal on odd subcarriers decrease to half of the original values, and all the clipping noises fall into the even subcarriers. After digital-to-analog conversion (DAC) and low pass filter (LPF), the analog electrical signal is modulated to the optical carrier.

At the receiver, the generated electrical signal passes through the LPF and performs analog-to-digital conversion (ADC), the digital signal processing is the reverse process of the transmitter. At last, signal on odd subcarriers is extracted for BER detection.

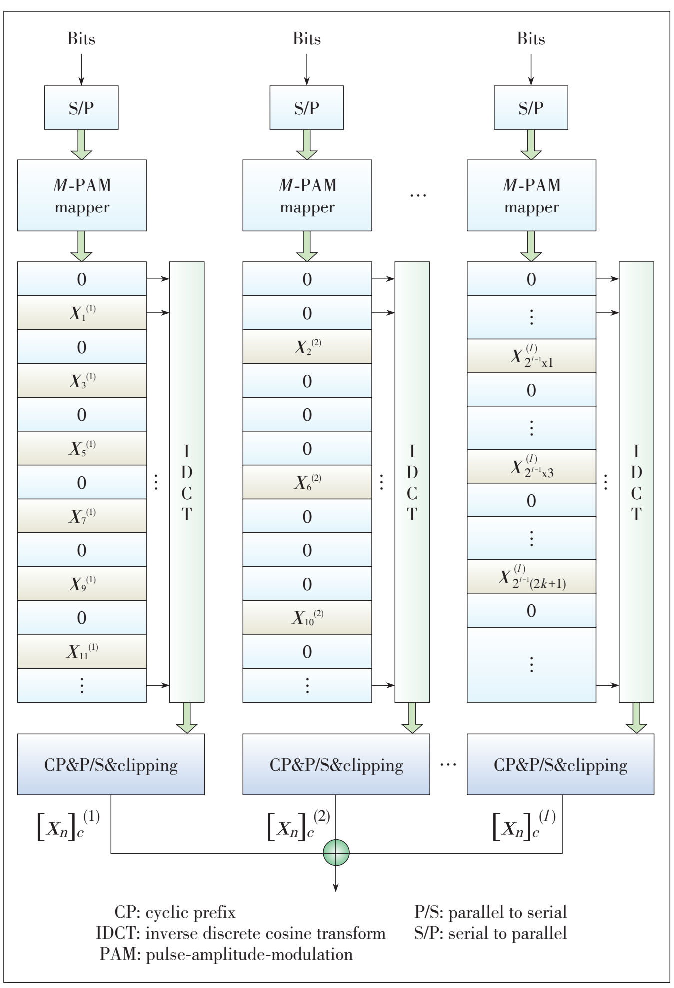
3 Layered ACO-FOFDM System

In this section, the principles of transmitter and receiver of the proposed layered ACO-FOFDM system are described in detail. L represents the total number of layers and l represents the l-th layer in the proposed system.

3.1 Transmitter

At transmitter, the superposition of multiple layers can be performed in time domain [13], [14] or frequency domain [15], [16]. In this paper, the superposition is performed in frequency domain, which is easier for implementation than that in time domain. **Fig. 2** shows the transmitter block diagram of layered ACO-FOFDM. At Layer 1, only odd subcarriers are used to carry the signal (index $2k+1$, $k=0,1,2,...,N/2^1-1$), the clipping noise falls on the even subcarriers, which is the same as conventional ACO-FOFDM system. The odd subcarriers of the remaining unused subcarriers at Layer 1 are used in Layer 2 (index $2(2k+1)$, $k=0,1,2,...,N/2^2-1$), and the clipping noise falls on the even subcarriers of the remaining unused subcarriers of Layer 1. The rest layers can be done in the same manner that odd subcarriers of the remaining unused subcarriers are utilized in each layer. At the l-th layer, only the $2^{l-1}(2k+1)(k=0,1,2,...,N/2^l-1)$ subcarriers are utilized to carry the signal, and the clipping noise of the l-th layer only influences the layers higher than l (i.e., Layer $l+1$ to Layer L). It is not difficult to get the original transmitted signal of each layer at receiver if the clipping noise of its previous layers is eliminated.

Although at Layer $l+1$, half number of subcarriers are used as that at Layer l, all the layers can perform N-point IDCT. The advantages of frequency domain superposition are that a repeater is not needed as in the time domain superposition, and that obtaining chips to conduct IDCT with the same order N is easier than obtaining chips with different IDCT orders in hardware implementation. Moreover, the computational complexity of IDCT/DCT depends on the number of subcarriers with actual data, not just depends on the number of order N. Therefore, the subcarriers without data do not increase the computational complexity. After N-point IDCT, the signal performs adding CP, parallel to serial conversion and zero clipping for each layer. The real and positive time-domain signals are added together at last.



▲**Figure 2. Transmitter block diagram of layered ACO-FOFDM.**

3.2 Receiver

The iterative receiver is used to recover transmitted signals from the layer l from low to high. The asymmetric clipping of the l-th layer at transmitter brings clipping noise, which can influence the layers with higher numbers. Only after the influence of lower layers is eliminated, can the signal at higher layers be recovered successfully. For easy to implement, two different structure types of receiver are introduced in this section, both of them can eliminate the clipping noise in frequency domain. Moreover, the soft decision and hard decision are described in this section. **Fig. 3** reveals two types of receiver structures with soft decision and hard decision.

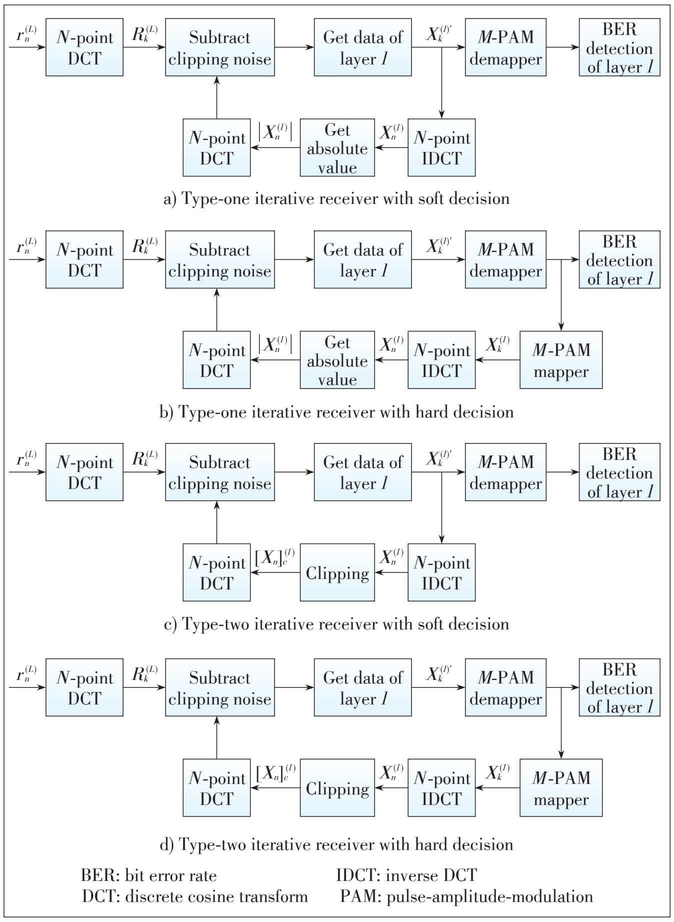
3.2.1 Type One

The first type of receiver can eliminate the influence of the l-th layer to the layers higher than l in frequency domain. For conventional ACO-OFDM system, the received signal on an odd subcarrier o can be represent as $X_o' = \frac{1}{2}FFT(x_n)$ and the received signal on an even subcarrier e can be represent as $X_e' = \frac{1}{2}FFT(|x_n|)$. The same results can be obtained in the ACO-FOFDM system as

$$X_o' = \frac{1}{2}DCT(x_n) \ , \tag{5}$$

$$X_e' = \frac{1}{2}DCT(|x_n|) \ . \tag{6}$$

Fig. 3a shows that after the additive white Gaussian noise (AWGN) channel, the signal on Layer 1 can be directly gotten from the odd subcarriers of $R_k^{(L)}$. Eqs. (5) and (6) reveal that the signal amplitude should double after DCT. Then, the signal $X_k^{(L)'}$ on Layer 1 can perform N-point IDCT, get the absolute value and perform N-point DCT, so the clipping noise on even



▲**Figure 3. Two different types of receiver structures with soft decision or hard decision.**

subcarriers of Layer 1 is obtained. Only the obtained clipping noise on even subcarriers of Layer 1 needs to be subtracted to get the signal on Layer 2. The rest layers can be done in the same manner. Only after the clipping noise of Layers 1 to l has been eliminated, can the signal on Layer l+1 be recovered successfully.

It should be noticed that the soft decision is applied in Fig. 3a, which means the noisy constellation values are used to reconstruct the clipping noise, while in Fig. 3b, an additional M-PAM mapper is needed so that the recovered bits are used to reconstruct the clipping noise. The determinate constellation positions are more accurate than noisy constellation positions to reconstruct the clipping noise, and this hard-decision method was proposed in [15]. The later simulation shows BER performance of the system with hard decision is better than the system with soft decision.

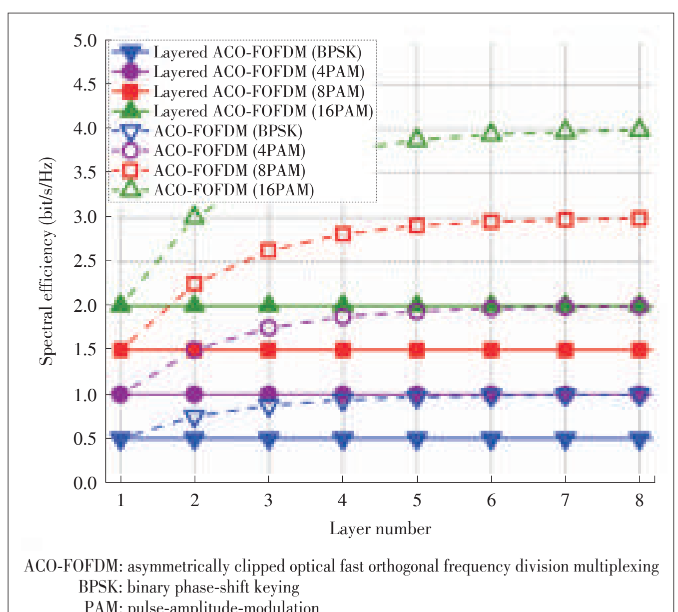
3.2.2 Type Two

Figs. 3c and 3d demonstrate the second type of receiver with soft decision and hard decision, respectively, which also eliminate the influence of the l-th layer to the layers higher than l in frequency domain. The principle can also be found in (5) and (6), and the received signal amplitude should be doubled after DCT. The signal on Layer 1 can be directly gotten from the odd subcarriers of $R_k^{(L)}$. Then, by soft decision the signal $X_k^{(L)'}$ on Layer 1 can perform N-point IDCT, while by hard decision the recovered bits on Layer 1 perform M-PAM mapper and N-point IDCT. After zero clipping and N-point DCT, the amplitude of the signal is doubled, so that the received signals on both odd and even subcarriers of Layer 1 are obtained. Then, we can subtract the obtained clipping noise on even subcarriers of Layer 1 to eliminate the influence of higher layers, and the rest layers can be done in the same manner.

4 Performance of Layered ACO-FOFDM System

4.1 Spectral Efficiency

Fig. 4 compares the spectral efficiency between the layered ACO-FOFDM and conventional ACO-FOFDM systems with BPSK, 4PAM, 8PAM and 16PAM. For conventional ACO-FOFDM system, half of subcarriers are used to carry the signal, which is the same as Layer 1 of the layered ACO-FOFDM system. Layer l+1 carries half number of data compared with Layer l, so the spectral efficiency increases with the increasing of the layer number, and the spectral efficiency converges to twice of that of conventional ACO-FOFDM system when the layer number is large enough. Moreover, in the layered ACO-OFDM system, Hermitian symmetry is needed to generate real-value signals. Therefore, to achieve the same spectral efficiency the size of the complex constellation in FFT-based layered ACO-OFDM system is M^2 while the size of the real constella-



▲**Figure 4. Spectral efficiency comparison between layered ACO-FOFDM system and conventional ACO-FOFDM system.**

tion in DCT-based layered ACO-FOFDM system is M.

4.2 Computational Complexity

As for computational complexity, **Table 1** reveals the comparison of real multiplications and real additions between N-point FFT and N-point DCT. The N-point FFT algorithm has $N\log_2 N-3N+4$ real multiplications and $3N\log_2 N-3N+4$ real additions, while the N-point DCT algorithm only has $\frac{N}{2}\log_2 N$ real multiplications and $\frac{3N}{2}\log_2 N-N+1$ real additions [20]. It is easy to see that the computational complexity of DCT algorithm is lower than that of FFT algorithm in layered ACO-OFDM system. This is the advantage of DCT-based layered ACO-FOFDM system. The multiplications of DCT occupy the most part of computational complexity, so we only considered the real multiplications at transmitter and receiver of the layered ACO-FOFDM system.

At transmitter, for layered ACO-FOFDM system with L layers, half of the subcarriers are utilized to carry the signal and the other half are zeros, so the computational complexity of Layer 1 is $\frac{N}{4}\log_2\frac{N}{2}$. Layer $l+1$ carries half of data compared Layer l, so the computational complexity increases with the increasing of layer number, which is the same as spectral efficiency. Finally, the computational complexity converges to twice of that of conventional ACO-FOFDM system when the layer number is large enough, which can be written as $\sum_{l=1}^{L}\frac{N}{4\times 2^{l-1}}\log_2\frac{N}{2\times 2^{l-1}}\leqslant(2-\frac{1}{2^{L-1}})\frac{N}{4}\log_2\frac{N}{2}$.

At receiver, the computational complexity of N-point DCT is $\frac{N}{2}\log_2 N$. However, for Layer l (l>1), an additional IDCT and an additional DCT are needed to eliminate the clipping noise. Layer 1 does not need to subtract the clipping noise, so the computational complexity for layer 1 is $\frac{N}{2}\log_2 N$, and for ACO-FOFDM system with L layers (L>1), the computational complexity is $\frac{N}{2}\log_2 N+2\sum_{l=1}^{L-1}\frac{N}{4\times 2^{l-1}}\log_2\frac{N}{2\times 2^{l-1}}\leqslant\frac{N}{2}\log_2 N+(4-\frac{1}{2^{L-3}})\frac{N}{4}\log_2\frac{N}{2}$. This result reveals the computational complexity increment of layered ACO-FOFDM receiver is no more than three times of conventional ACO-FOFDM system.

4.3 BER Performance

The advantage of hard decision over soft decision is revealed in **Fig. 5**. The 4-layer ACO-FOFDM with hard decision has nearly the same BER performance as 2-layer ACO-FOFDM with soft decision with a forward error correction (FEC) limit (BER=1×10^{-3}). The required $E_{b(elec)}/N_0$ (electrical bit energy to noise power ratio) to achieve the FEC limit for layered ACO-FOFDM systems with soft decision is far more than that with hard decision. As for hard decision, although the incorrectly decoded bits in lower layers can be translated into distortions on all the subsequent layers, this error propagation effect becomes weaker with the increasing of $E_{b(elec)}/N_0$. The determinating constellation positions are more accurate than noisy constellation positions to reconstruct the clipping noise. Therefore, hard decision is used for iterative receiver of the layered ACO-FOFDM system.

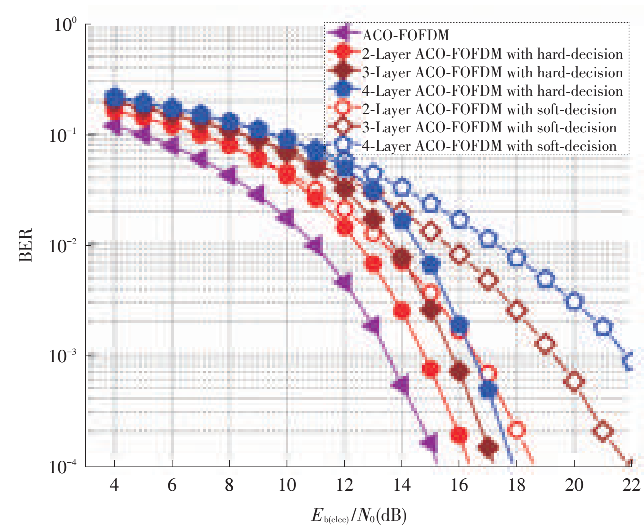
Fig. 6 shows the BER performance comparison between 4PAM layered ACO-FOFDM system and 16QAM layered ACO-OFDM system in AWGN channel. With the same spectral efficiency, BER curves of M-PAM-modulated ACO-FOFDM and those of M^2-QAM-modulated ACO-OFDM coincide with each other, which have been demonstrated in [18]. This result is also suitable for the layered ACO-FOFDM and layered ACO-OFDM systems. With the same constellation size of each layer, the BER curves of M-PAM-modulated layered ACO-FOFDM and M^2-QAM-modulated layered ACO-OFDM coincide with each other when the same simulation parameters are adopted.

As demonstrated in **Fig. 7**, the BER performance of layered ACO-FOFDM system is compared with DCO-FOFDM system with optimal DC-bias in AWGN channel, and the spectral efficiencies range from 1 bit/s/Hz to 4 bits/s/Hz. Although the

▼Table 1. Comparison of real multiplications and real additions between N-point FFT and N-point DCT

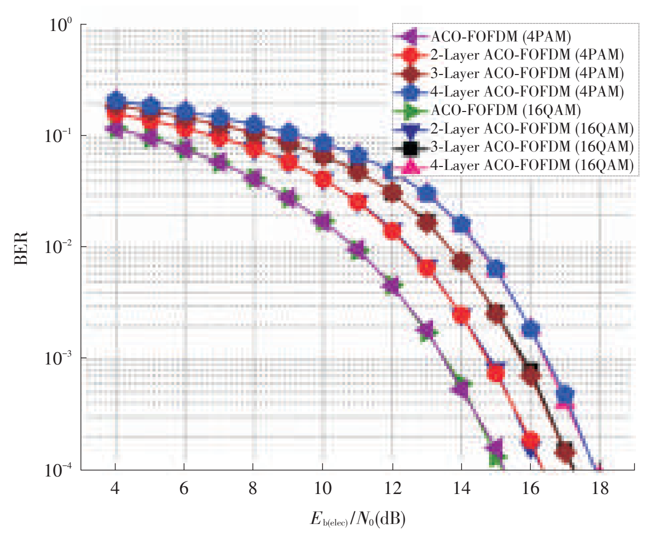
	Real multiplications	Real additions
FFT	$N\log_2 N-3N+4$	$3N\log_2 N-3N+4$
DCT	$\frac{N}{2}\log_2 N$	$\frac{3N}{2}\log_2 N-N+1$

DCT: discrete cosine transform FFT: fast Fourier transform



ACO-FOFDM: asymmetrically clipped optical fast orthogonal frequency division multiplexing
BER: bit error rate

▲**Figure 5. BER performance comparison between soft decision and hard decision in 4PAM layered ACO-FOFDM.**

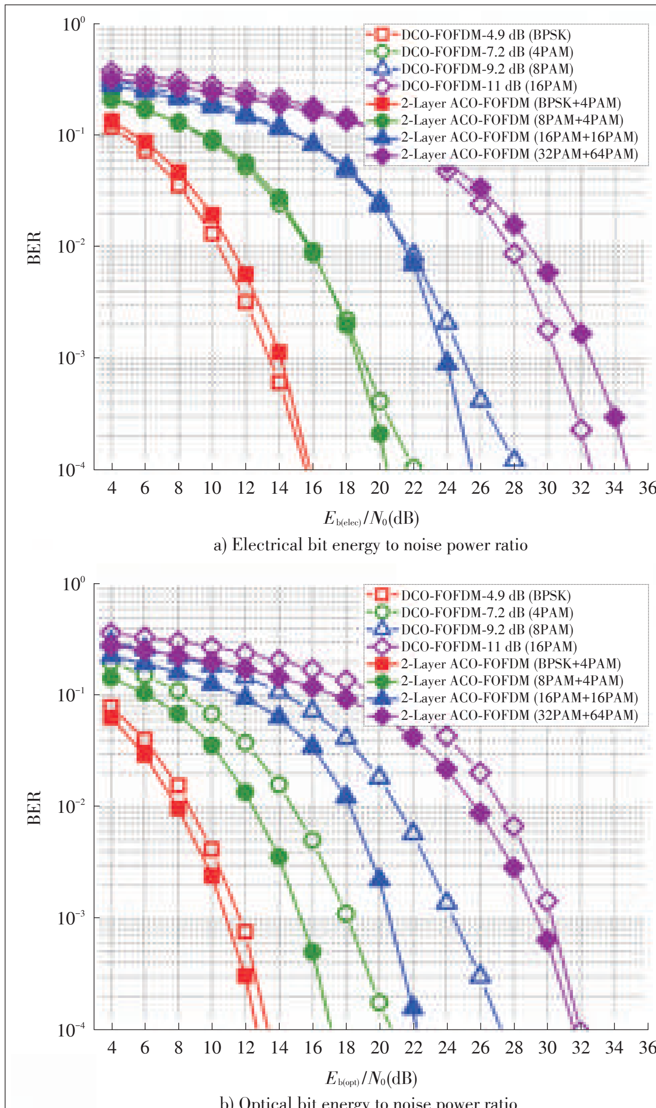


ACO-FOFDM: asymmetrically clipped optical fast orthogonal frequency division multiplexing
BER: bit error rate
PAM: pulse-amplitude-modulation
QAM: quadrature-amplitude-modulation

▲**Figure 6. BER performance for 4PAM layered ACO-FOFDM system and 16QAM layered ACO-OFDM system.**

spectral efficiency of DCO-FOFDM system is twice of that of the ACO-FOFDM system with the same constellation size, the performance of DCO-FOFDM strongly depends on the DC-bias [8]. If the DC-bias is not large enough, the negative values are clipped to zero, which leads to clipping distortion. If the DC-bias is larger than the negative peaks, the system can be inefficient in optical power. To obtain the optimal BER performance of DCO-FOFDM system, we use 4.9 dB, 7.2 dB, 9.2 dB and 11 dB DC-biases for BPSK, 4PAM, 8PAM and 16PAM as investigated in [19]. The disadvantage of DCO-FOFDM is different proper DC biases should be found for different systems. For the layered ACO-FOFDM system, if all the layers use the same constellation size, its spectral efficiency can never be the same as that of DCO-FOFDM system because the infinite number of layers are required for layered ACO-FOFDM system. Therefore, we use 2-layer ACO-FOFDM system with different constellation sizes for layers to obtain the same spectral efficiency as DCO-FOFDM system. The average signal power applied to each layer is proportion to the number of bits transmitted in each layer. The constellation size combination mode with relatively good BER performance is chosen. Moreover, the optimal combination mode of layer number and constellation size can be used in the future work to achieve better BER performance.

Figs. 7a and 7b reveal the BER performance of 2-layer ACO-FOFDM system and DCO-FOFDM system with optimal DC-bias in terms of $E_{b(elec)}/N_0$ (electrical bit energy to noise power ratio) and $E_{b(opt)}/N_0$ (optical bit energy to noise power ratio). For the conversion from electrical bit energy to noise power ratio to optical bit energy to noise power ratio, the average optical power is set to unity [8]. In Fig. 7a, the BER performance



ACO-FOFDM: asymmetrically clipped optical fast orthogonal frequency division multiplexing
BER: bit error rate
BPSK: binary phase-shift keying
DCO-FOFDM: DC-offset fast orthogonal frequency division multiplexing
PAM: pulse-amplitude-modulation

▲**Figure 7. BER performance comparison between layered ACO-FOFDM system and DCO-FOFDM system with the same spectral efficiency. The spectral efficiencies are set to 1 bit/s/Hz, 2 bits/s/Hz, 3 bits/s/Hz and 4 bits/s/Hz. The optimal dc-biases of DCO-FOFDM system are 4.9 dB, 7.2 dB, 9.2 dB and 11 dB for BPSK, 4PAM, 8PAM and 16PAM [18].**

of 2-layer ACO-FOFDM system is only a little bit better than that of DCO-FOFDM system when the spectral efficiency is 2 bits/s/Hz or 3 bits/s/Hz, and its BER performance is worse than that of DCO-FOFDM system when the spectral efficiency is 1 bit/s/Hz or 4 bits/s/Hz. The superiority of 2-layer ACO-FOFDM system is apparent in terms of optical bit energy to noise power ratio, it has 1.23 dB, 2.77 dB, 3.67 dB and 0.78 dB improvement at FEC limit compared with DCO-FOFDM system when the spectral efficiencies are 1 bit/s/Hz, 2 bits/s/

Hz, 3 bits/s/Hz and 4 bits/s/Hz. Consequently, the layered ACO-FOFDM system also has potential for application in IM/DD systems.

5 Conclusions

This paper proposes a layered ACO-FOFDM scheme based on DCT for IM/DD systems. The conventional ACO-FOFDM system only uses odd subcarriers to carry the signal, which leads to low spectral efficiency. For the layered ACO-FOFDM system, the superposition of signal on different layers is performed in frequency domain at transmitter to increase the spectral efficiency. The average signal power applied to each layer is proportion to the number of bits transmitted in each layer. At receiver, the iterative receiver is used to recover transmitted signal from low-to-high layers, and hard decision is applied to improve the BER performance. In the layered ACO-FOFDM system, the spectral efficiency converges to twice of that of conventional ACO-FOFDM system when the layer number is large enough. For the same spectral efficiency, the layered ACO-FOFDM system based on DCT has the same BER performance but lower computational complexity compared with the layered ACO-OFDM system based on FFT. Simulation results demonstrate that the proposed layered ACO-FOFDM system with zero DC-bias is more suitable for adaptive system, and also a promising scheme for application in IM/DD systems.

References

[1] J. Armstrong, "OFDM for optical communications," *Journal of Lightwave Technology*, vol. 27, no. 3, pp. 189–204, Feb. 2009. doi: 10.1109/JLT.2008.2010061.

[2] W. Shieh and I. Djordjevic, *OFDM for optical communications*. New York, USA: Academic Press, 2009.

[3] J. Zhou, Y. Yan, Z. Cai, Y. Qiao, and Y. Ji, "A cost-effective and efficient scheme for optical OFDM in short-range IM/DD systems," *IEEE Photonics Technology Letters*, vol. 26, no. 13, pp. 1372–1374, Jul. 2014. doi: 10.1109/LPT.2014.2325602.

[4] J. Zhou, Y. Qiao, Z. Cai, and Y. Ji, "An improved scheme for Flip-OFDM based on Hartley transform in short-range IM/DD systems," *Optics Express*, vol. 22, no. 17, pp. 20748–20756, Aug. 2014. doi: 10.1364/OE.22.020748.

[5] J. Wei, N. Eiselt, C. Sanchez, R. Du, and H. Griesser, "56 Gb/s multi-band CAP for data center interconnects up to an 80 km SMF," *Optics Letters*, vol. 41, no. 7, pp. 4122–4125, Sept. 2016. doi: 10.1364/OL.41.004122.

[6] J. Zhou and Y. Qiao, "Low-peak-to-average power ratio and low-complexity asymmetrically clipped optical orthogonal frequency-division multiplexing uplink transmission scheme for long-reach passive optical network," *Optics Letters*, vol. 40, no. 17, pp. 4034–4037, Sept. 2015. doi: 10.1364/OL.40.004034.

[7] O. González, R. Perez-Jimenez, S. Rodriguez, et al., "OFDM over indoor wireless optical channel," *IEE Proceedings-Optoelectronics*, vol. 152, no. 4, pp. 199–204, Aug. 2005. doi: 10.1049/ip-opt:20045065.

[8] J. Armstrong and B. Schmidt, "Comparison of asymmetrically clipped optical OFDM and DC-biased optical OFDM in AWGN," *IEEE Communications Letters*, vol. 12, no. 5, pp. 343–345, May 2008. doi: 10.1109/LCOMM.2008.080193.

[9] J. Armstrong and A. Lowery, "Power efficient optical OFDM," *Electronics Letter*, vol. 42, no. 6, pp. 370–372. Mar. 2006. doi: 10.1049/el:20063636.

[10] S. Dissanayake and J. Armstrong, "Comparison of ACO-OFDM, DCO-OFDM and ADO-OFDM in IM/DD Systems," *Journal of Lightwave Technology*, vol. 31, no. 7, pp. 1063–1072, Apr. 2013. doi: 10.1109/JLT.2013.2241731.

[11] B. Ranjha and M. Kavehrad, "Hybrid asymmetrically clipped OFDM-based IM/DD optical wireless system," *Journal of Optical Communications and Networking*, vol. 6, no. 4, pp. 387–396, Apr. 2014. doi: 10.1364/JOCN.6.000387.

[12] N. Wu and Y. Bar-Ness, "A novel power-efficient scheme asymmetrically and symmetrically clipping optical (ASCO)-OFDM for IM/DD optical systems," *EURASIP Journal on Advances in Signal Processing*, vol. 15, no. 1, pp. 1–10, Jan. 2015. doi: 10.1186/1687-6180-2015-3.

[13] Q. Wang, C. Qian, X. Guo, et al., "Layered ACO-OFDM for intensity-modulated direct-detection optical wireless transmission," *Optics Express*, vol. 23, no. 9, pp. 12382–12393, May 2015. doi: 10.1364/OE.23.012382.

[14] M. Islim, D. Tsonev, and H. Haas, "On the superposition modulation for OFDM-based optical wireless Communication," in *Proc. IEEE GlobalSIP 2015*, Orlando, USA, Dec. 2015, pp. 1022–1026. doi: 10.1109/GlobalSIP.2015.7418352.

[15] E. Lam, S. Wilson, H. Elgala, and T. Little. (2015, October 28). *Spectrally and energy efficient OFDM (SEE-OFDM) for intensity modulated optical wireless systems* [Online]. Avaliable: http://arxiv.org/abs/1510.08172

[16] M. Islim and H. Haas, "Augmenting the spectral efficiency of enhanced PAM-DMT-based optical wireless communications," *Optics Express*, vol. 24, no. 11, pp. 11932–11949. May 2016. doi: 10.1364/OE.24.011932.

[17] E. Giacoumidis, A. Tsokanos, C. Mouchos, et al., "Extensive comparisons of optical fast-OFDM and conventional optical OFDM for local and access networks," *Journal of Optical Communications and Networking*, vol. 4, no. 10, pp. 724–733, Oct. 2012. doi: 10.1364/JOCN.4.000724.

[18] J. Zhou, Y. Qiao, Z. Cai, and Y. Ji, "Asymmetrically Clipped Optical Fast OFDM Based on Discrete Cosine Transform for IM/DD Systems," *Journal of Lightwave Technology*, vol. 33, no. 9, pp. 1920–1927. May 2015. doi: 10.1109/JLT.2015.2398441.

[19] J. Zhou, Y. Qiao, T. Zhang, et al., "FOFDM Based on Discrete Cosine Transform for Intensity-Modulated and Direct-Detected Systems," *Journal of Lightwave Technology*, vol. 34, no. 16, pp. 3717–3725, Aug. 2015. doi: 10.1109/JLT.2016.2586526.

[20] M. Vetterli and H. J. Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operations," *Signal Processing*, vol. 6, no. 4, pp. 267–278, Aug. 1984. doi: 10.1016/0165-1684(84)90059-8.

Manuscript received: 2016-12-25

Biographies

GUO Mengqi (guomengqi@bupt.edu.cn) received her B.S. degree from Beijing University of Posts and Telecommunications, China in 2015. Currently, she is a Ph.D. candidate in School of Information and Communication Engineering and State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications. Her research interests include signal processing for optical orthogonal frequency division multiplexing systems.

ZHOU Ji (zhouji@bupt.edu.cn) received his B.S. degree in communication engineering from Beijing University of Posts and Telecommunications, China in 2012. Currently, he is a Ph.D. candidate in School of Information and Communication Engineering and State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications. His research interests include signal processing for optical orthogonal frequency division multiplexing systems and coherent optical communications.

TANG Xizi (tangxizi@bupt.edu.cn) received his B.S. degree from Beijing Technology and Business University, China in 2012. Currently, he is a Ph.D. candidate in School of Information and Communication Engineering and State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications. His research interests include signal processing for passive optical networks.

QIAO Yaojun (qiao@bupt.edu.cn) received his B.S. degree from Hebei Normal University, China in 1994, M.S. degree from Jilin University, China in 1997, and Ph.D. degree from Beijing University of Posts and Telecommunications (BUPT), China in 2000. He was with Lucent and Fujitsu from 2000 to 2007. In 2007, he joined BUPT, where he is currently a professor. His research interests include optical fiber communication system and network.

ZTE Communications Guidelines for Authors

- **Remit of Journal**

ZTE Communications publishes original theoretical papers, research findings, and surveys on a broad range of communications topics, including communications and information system design, optical fiber and electro-optical engineering, microwave technology, radio wave propagation, antenna engineering, electromagnetics, signal and image processing, and power engineering. The journal is designed to be an integrated forum for university academics and industry researchers from around the world.

- **Manuscript Preparation**

Manuscripts must be typed in English and submitted electronically in MS Word (or compatible) format. The word length is approximately 3000 to 8000, and no more than 8 figures or tables should be included. Authors are requested to submit mathematical material and graphics in an editable format.

- **Abstract and Keywords**

Each manuscript must include an abstract of approximately 150 words written as a single paragraph. The abstract should not include mathematics or references and should not be repeated verbatim in the introduction. The abstract should be a self-contained overview of the aims, methods, experimental results, and significance of research outlined in the paper. Five carefully chosen keywords must be provided with the abstract.

- **References**

Manuscripts must be referenced at a level that conforms to international academic standards. All references must be numbered sequentially in-text and listed in corresponding order at the end of the paper. References that are not cited in-text should not be included in the reference list. References must be complete and formatted according to ZTE Communications Editorial Style. A minimum of 10 references should be provided. Footnotes should be avoided or kept to a minimum.

- **Copyright and Declaration**

Authors are responsible for obtaining permission to reproduce any material for which they do not hold copyright. Permission to reproduce any part of this publication for commercial use must be obtained in advance from the editorial office of *ZTE Communications*. Authors agree that a) the manuscript is a product of research conducted by themselves and the stated co-authors, b) the manuscript has not been published elsewhere in its submitted form, c) the manuscript is not currently being considered for publication elsewhere. If the paper is an adaptation of a speech or presentation, acknowledgement of this is required within the paper. The number of co-authors should not exceed five.

- **Content and Structure**

ZTE Communications seeks to publish original content that may build on existing literature in any field of communications. Authors should not dedicate a disproportionate amount of a paper to fundamental background, historical overviews, or chronologies that may be sufficiently dealt with by references. Authors are also requested to avoid the overuse of bullet points when structuring papers. The conclusion should include a commentary on the significance/future implications of the research as well as an overview of the material presented.

- **Peer Review and Editing**

All manuscripts will be subject to a two-stage anonymous peer review as well as copyediting, and formatting. Authors may be asked to revise parts of a manuscript prior to publication.

- **Biographical Information**

All authors are requested to provide a brief biography (approx. 100 words) that includes email address, educational background, career experience, research interests, awards, and publications.

- **Acknowledgements and Funding**

A manuscript based on funded research must clearly state the program name, funding body, and grant number. Individuals who contributed to the manuscript should be acknowledged in a brief statement.

- **Address for Submission**

http://mc03.manuscriptcentral.com/ztecom
12F Kaixuan Building, 329 Jinzhai Rd, Hefei 230061, P. R. China

ZTE COMMUNICATIONS

ZTE Communications has been indexed in the following databases:

- Abstract Journal
- Cambridge Scientific Abstracts (CSA)
- China Science and Technology Journal Database
- Chinese Journal Fulltext Databases
- Inspec
- Ulrich's Periodicals Directory
- Wanfang Data—Digital Periodicals

ZTE COMMUNICATIONS
Vol. 15 No. 3 (Issue 58)
Quarterly
First English Issue Published in 2003

Supervised by:
Anhui Science and Technology Department

Sponsored by:
Anhui Science and Technology Information Research Institute and ZTE Corporation

---

Editorial Correspondence:
Add: 12F Kaixuan Building, 329 Jinzhai Road,
Hefei 230061, P. R. China
Tel: +86-551-65533356
Fax: +86-551-65850139
Email: magazine@zte.com.cn

Published and Circulated (Home and Abroad) by:
Editorial Office of *ZTE Communications*

Printed by:
Hefei Tiancai Color Printing Company

Publication Date: August 25, 2017

Publication Licenses:
ISSN 1673-5188
CN 34-1294/ TN

Annual Subscription: RMB 80