An International ICT R&D Journal Sponsored by ZTE Corporation

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

ZTE COMMUNICATIONS

中兴通讯技术(英文版)

http://tech.zte.com.cn

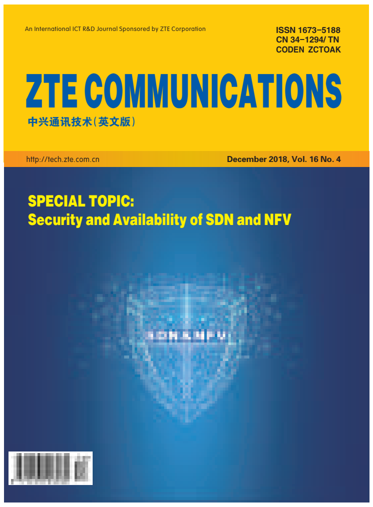
December 2018, Vol. 16 No. 4

SPECIAL TOPIC:
Security and Availability of SDN and NFV

ZTE中兴 ZTE COMMUNICATIONS 中兴通讯技术(英文版) VOLUME 16 NUMBER 4 DECEMBER 2018

The 8th Editorial Board of ZTE Communications

CONTENTS



An International ICT R&D Journal Sponsored by ZTE Corporation

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

ZTE COMMUNICATIONS

中兴通讯技术(英文版)

http://tech.zte.com.cn

December 2018, Vol. 16 No. 4

SPECIAL TOPIC:
Security and Availability of SDN and NFV

SDN&NFV

Special Topic: Security and Availability of SDN and NFV

CONTENTS

ZTE COMMUNICATIONS
Vol. 16 No. 4 (Issue 64)
Quarterly
First English Issue Published in 2003

Supervised by:
Anhui Science and Technology Department

Sponsored by:
Anhui Science and Technology Information
Research Institute;
Magazine House of ZTE Communications

Published and Circulated
(Home and Abroad) by:
Magazine House of ZTE Communications

Staff Members:
Editor−in−Chief: WANG Xiang
Executive Associate
Editor−in−Chief: HUANG Xinming
Editor−in−Charge: ZHU Li
Editors: XU Ye and LU Dan
Producer: YU Gang
Circulation Executive: WANG Pingping
Assistant: WANG Kun

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

Printed by:
Hefei Tiancai Color Printing Company

Publication Date:
December 25, 2018

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

Annual Subscription:
RMB 80

Research Paper

Review

Roundup

DOI: 10.19729/j.cnki.1673-5188.2018.04.001
http://kns.cnki.net/kcms/detail/34.1294.TN.20181029.1705.002.html, published online October 29, 2018

Editorial

Security and Availability of SDN and NFV

► Guest Editor

CHEN Yan received the Ph.D. degree in computer science from the University of California at Berkeley, USA, in 2003. He is currently a professor with the Department of Electrical Engineering and Computer Science, Northwestern University, USA. Based on Google Scholar, his papers have been cited over 10,000 times and his h-index is 49. His research interests include network security, measurement, and diagnosis for large-scale networks and distributed systems. He received the Department of Energy Early CAREER Award in 2005, the Department of Defense Young Investigator Award in 2007, the Best Paper nomination in ACM SIGCOMM 2010, and the Most Influential Paper Award in ASPLOS 2018.

Software defined networking (SDN) and network function virtualization (NFV) have attracted significant attention from both academia and industry. Fortunately, by virtue of unique advantages of programmability and centralized control, SDN has been widely used in various scenarios, such as home networking, enterprise networking, telecommunication networking, data center, and cloud networking. Meanwhile, through adopting virtualization technology to realize various network functions, NFV delivers high-performance networks with greater scalability, elasticity, and adaptability at reduced costs compared to networks built from traditional networking equipment. NFV covers a wide range of network applications, including video, Software Defined Wide Area Network (SD-WAN), Internet of Things (IoT), and 5G.

With SDN and NFV, the flexibility of networks is increased, the utilization of resources is improved, the network operation and maintenance cost is cut down, and the time-to-market of new service is considerably decreased. However, these benefits bring new security challenges for network at the same time. Besides traditional inherent security issues (Distributed Denial of Service (DDoS) attack; Man-in-the-Middle attack), various new attacks are introduced by the new architecture and technology, such as data-to-control plane saturation attack, control plane reflection attack, and control-data plane view inconsistency. Thus, new countermeasures for this new scenario are necessary to defend against attacks and make the whole system more secure and reliable.

This special issue aims at giving a bird view to the concept of SDN and NFV, then analyzing the potential security issues and proposing feasible countermeasures.

Since the birth of SDN, academia and industry have invested a lot of energy in research. However, the deployment of SDN has faced several security issues which put a severe threat to crucial resources of the SDN infrastructure, including resources of the control plane, data plane, and in-between downlink channel. The first paper “Survey of Attacks and Countermeasures for SDN” by BAI et al. analyzes the vulnerability of SDN, presents two kinds of SDN-targeted attacks, namely data-to-control plane saturation attack and control plane reflection attack, and finally proposes the corresponding defense frameworks.

With the development and revolution of network in recent years, traditional hardware based network security solutions have shown some significant disadvantages in cloud computing based Internet data centers (IDCs), such as high cost and lack of flexibility. With the implementation of SDN, network security solutions could be more flexible and efficient, such as SDN based firewall service and SDN based DDoS-attack mitigation service. The second paper “SDN Based Security Services” by ZHANG et al. analyzes some typical SDN based network security services, and provides a research on SDN based cloud security service and its implementation in IDC network.

SDN has been well researched in both academia and industry. The main reason SDN is so concerned is its ability to dynamically change the network states in response to the global view. In particular, the control message processing capability on switches, especially the prevailing Ternary Content Addressable Memory (TCAM) based flow tables on physical SDN switches, proves to be the bottleneck along the policy update pipeline. The limitation has slowed down network updates

Editorial

CHEN Yan

and hurt network visibility, which further constrains the control plane applications with dynamic policies significantly. To solve this serious problem, the third paper “Optimization Framework for Minimizing Rule Update Latency in SDN Switches” by CHEN et al. presents a SDN update optimization framework RuleTris to minimize rule update latency for TCAM-based switches.

NFV is a new network technology which will be widely popularized and used in the foreseen future. Therefore, how to build a secure architecture for NFV is an important issue. Trusted computing has the ability to provide security for NFV and it is called trusted NFV system. The fourth paper “A New Direct Anonymous Attestation Scheme for Trusted NFV System” by CHEN et al. proposes a new NFV Direct Anonymous Attestation (NFV-DAA) scheme based on trusted NFV architecture. It is based on the Elliptic curve cryptography, and transfers the computation of variable D from the Trusted Platform Module (TPM) to Issuer. With the mutual authentication mechanism that those existing DAA schemes do not have and an efficient batch proof and verification scheme, the performance trusted NFV system is optimized. According to the experiment results, NFV - DAA scheme has higher security level and efficiency than those existing DAA schemes.

The aforementioned four excellent works comprehensively set forth the security challenges faced by SDN and NFV from different perspectives, and propose countermeasures and defense frameworks to effectively improve the security and availability of SDN and NFV.

Finally, we would like to thank all the authors, the external reviewers and the staff at *ZTE Communications* for contributing their excellent research work, precious time and energy to this special issue.

DOI: 10.19729/j.cnki.1673-5188.2018.04.002
http://kns.cnki.net/kcms/detail/34.1294.TN.20181015.1722.002.html, published online October 15, 2018

Survey of Attacks and Countermeasures for SDN

BAI Jiasong[1,2,3], **ZHANG Menghao**[1,2,3], **and BI Jun**[1,2,3]
(1. Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China;
2. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China;
3. Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing 100084, China)

Abstract

Software defined networking (SDN) has attracted significant attention from both academia and industry by its ability to reconfigure network devices with logically centralized applications. However, some critical security issues have also been introduced along with the benefits, which put an obstruction to the deployment of SDN. One root cause of these issues lies in the limited resources and capability of devices involved in the SDN architecture, especially the hardware switches lied in the data plane. In this paper, we analyze the vulnerability of SDN and present two kinds of SDN-targeted attacks: 1) data-to-control plane saturation attack which exhausts resources of all SDN components, including control plane, data plane, and the in-between downlink channel and 2) control plane reflection attack which only attacks the data plane and gets conducted in a more efficient and hidden way. Finally, we propose the corresponding defense frameworks to mitigate such attacks.

Keywords

SDN; indirect/direct data plane event; data-to-control plane saturation attack; control plane reflection attack

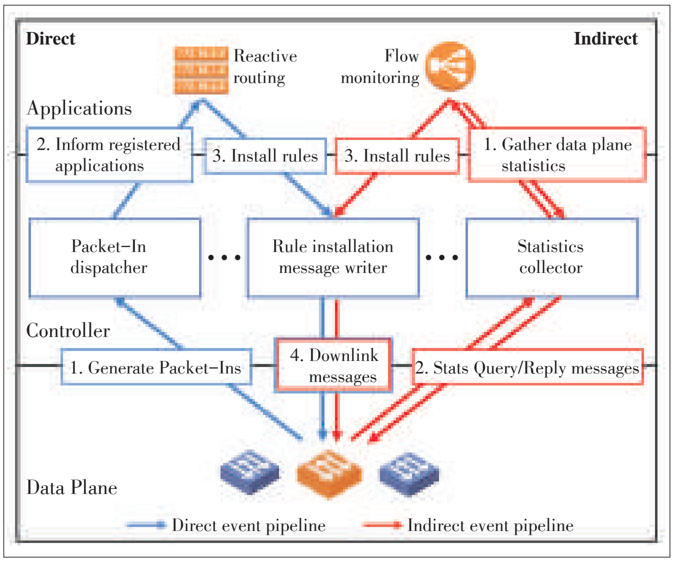
1 Introduction

Software defined networking (SDN) has enabled flexible and dynamic network functionalities with a novel programming paradigm. By decoupling the control plane from the data plane, control logics of different network functionalities could be implemented on top of the logically centralized controller as “applications”. Typical SDN applications are implemented as event-driven programs, which receive information directly or indirectly from switches and distribute the processing decisions of packets to switches accordingly. These applications enable SDN to adapt to the data plane dynamics quickly and make the responses according to the application policies timely. A wide range of network functionalities are implemented in this way, allowing SDN-enabled switches [1] to behave as firewall [2], load balancing [3], L2/L3 routing, and so on.

While the decoupling paradigm has enabled unprecedented programmability in networks, it also becomes the vulnerability of SDN infrastructure. The typical SDN infrastructure consists of three major components: the control plane, the data plane, and a control channel, where the two planes can communicate through standard protocols. To express the logics of control applications, control messages are generated in both the two planes and transferred through the channel. By triggering numerous control messages in a short time, attackers can paralyze the SDN infrastructure by exhausting the available resources of all three components. In particular, the control message processing capability on switches proves to be the bottleneck of the infrastructure, which is constrained by the wimpy central processing units (CPUs) , limited ternary content-addressable memory (TCAM) [4], [5] update rate and flow table capacity due to financial and power consumption reasons. These limitations have slowed down network updates and hurt network visibility, which further constrains the control plane applications with dynamic policies significantly [6].

The applications enable a network to dynamically adjust network configurations based on certain data plane events as illustrated in **Fig. 1**. These events can be categorized into the following two types: direct data plane events (e.g., Packet-In messages) and indirect data plane events (e.g., Statistics Query/Reply messages). In the first case, the controller installs a default table-miss flow rule on the switch. Arriving packets which fail to match any flow rule are forwarded to the control plane for further processing. In the second case, the controller installs a counting flow rule on the switch to record the statistics of arriving packets and periodically polls the flow counter values. A large number of control plane applications combine these two kinds of events to compose complicated network functions.

This work was supported in part by the National Key R&D Program of China under Grant No. 2017YFB0801701, the National Science Foundation of China under Grant No. 61472213 and CERNET Innovation Project (NGII20160123).



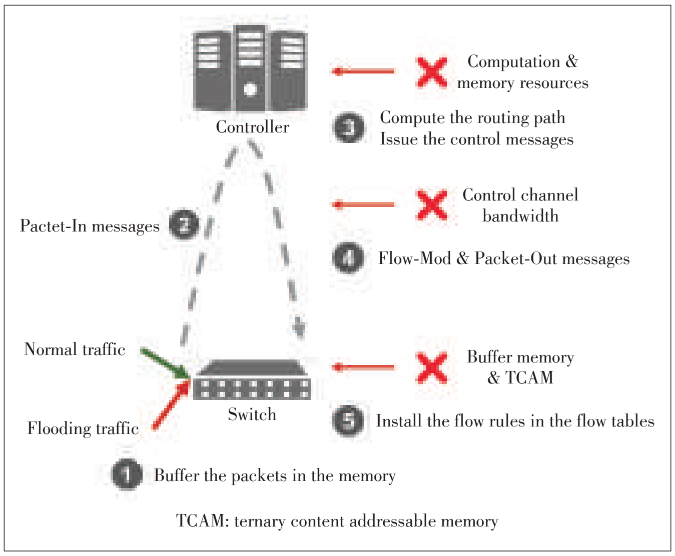
▲Figure 1. Architecture and event pipelines of current software-defined networking.

From our previous study, we find that flow rule update messages from the SDN control plane will be triggered by both kinds of events, which can be exploited by an intentional attacker. In this article, we present two kinds of attacks, i.e., the data-to-control saturation attack [7], a dedicated Denial-of-Service (DoS) attack against SDN essentially, and the control plane reflection attack [8], which can be further categorized into the table-miss striking attack and counter manipulation attack by the type of applied events. Furthermore, we propose the defense frameworks to mitigate these two attacks. In the following, we illustrate the details of two types of attacks in Sections 2 and 3, present the corresponding defense frameworks in Sections 4 and 5, and conclude this article and make some discussion in Section 6.

2 Data-to-Control Saturation Attacks

Intuitively, an attacker could commit the data-to-control saturation attack by producing a large number of short-flows by controlling a number of zombie hosts in an SDN-enabled network. The attack traffic is mixed with benign traffic, making it difficult to be identified. With the reactive routing and fine-grained flow control mechanism taken by the existing mainstream SDN controllers, the unmatched packets in the data plane would be delivered to the controller directly and processed by the corresponding applications. As a result, the data plane, the control channel and the control plane would quickly suffer from the attack, and soon the SDN system could not provide any service for benign traffic.

We start from a simplified motivating scenario to illustrate how an adversary attacks the SDN infrastructure. As depicted in **Fig. 2**, when a new packet arrives at a switch where there is no matching flow entry in the local flow tables, the switch will store the packet in its buffer memory and send a Packet-In message to the controller. The message only contains the packet header if the buffer memory is not full, but will contain the whole packet when the buffer memory is full. After the controller receives the message, it computes the route and takes the corresponding actions on the switches through control messages including Flow-Mod and Packet-Out. Then the switches parse the packets and install the flow rules in the capacity-limited flow tables. The attacker can exploit the vulnerability of this reactive packet processing mechanism by flooding malicious packets to the switches. The header fields of these packets are filled with deliberately forged values that it is almost impossible for them to be matched by any existing flow entries in the switches. After that, numerous table-misses are triggered, and a large number of packet-in messages are flooded to the controller, making the entire SDN system suffer from resource exhaustion. In this adversary model, all three levels of SDN resources are compromised.


Controller
Computation & memory resources
3 Compute the routing path Issue the control messages
Pactet-In messages 2
Control channel bandwidth
4 Flow-Mod & Packet-Out messages
Normal traffic
Switch
Flooding traffic
Buffer memory & TCAM
5 Install the flow rules in the flow tables
1 Buffer the packets in the memory
TCAM: ternary content addressable memory

▲Figure 2. Adversary model of the data-to-control saturation attack.

3 Control Plane Reflection Attacks

Compared with saturation attacks, control plane reflection attacks are much hidden and sophisticated. It does not target at the controller, nor the end host, but it utilizes the limited processing capability of downlink messages in the SDN-enabled hardware switches and easily gain much more prominent effects than saturation attacks.

A general procedure of control plane reflection attacks consists of two phases, i.e., the probing phase and triggering phase. During the probing phase, an attacker uses several kinds of probe packets to learn the conditions that application adopts to issue new flow rule update messages. Upon the information obtained, the attacker can carefully craft the patterns of attack packet stream to trigger numerous flow rule update messages in a short interval to paralyze the hardware switches.

3.1 Table-Miss Striking Attacks

The table-miss striking attack is an enhanced attack vector from the saturation attack. Instead of leveraging a random packet generation method to commit the attack, a striking attack adopts a more accurate and cost-efficient manner by utilizing probing and triggering phases.

The probing phase is to learn the confidential information of the control plane to guide the patterns of attack packet streams. The attacker could first probe the use of direct data plane events by using various low-rate probing packets with deliberately faked headers. By sending these probing packets and observing the response accordingly, the round trip time (RTT) could be obtained. If the first packet has a longer RTT, we can conclude that it is directed to the controller while the others are forwarded directly to the data plane. This indicates that the specific packet header matches no flow rule in the switch. Then the attacker could change one of the header fields with the variable-controlling approach. Within limited trials (42 in the latest OpenFlow specification), the attacker was able to determine which header fields were sensitive to the controller. Then the attacker could deliberately craft attack stream based on probed grains to trigger the expensive flow rule update operations.

3.2 Counter Manipulation Attacks

The counter manipulation attack is based on indirect data plane events and much more sophisticated compared with abovementioned attacks. In order to accurately infer the usage of indirect data plane events, three types of packets are required, i.e., timing probing packets, test packets and data plane streams.

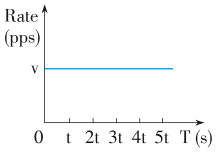
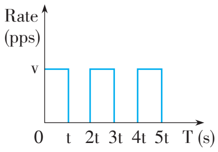
Timing probing packets are used to measure the work load of software agent of a switch, inspired by time pings in [9]. Three properties should be satisfied. First, they should go to the control plane by hitting the table-miss flow rule in the switch, and trigger the operations of corresponding applications. Second, each of them must evoke a response from the network to compute the RTT. Third, they should be sent in an extremely low rate (10 packets per second (pps) is enough) and put as low loads as possible to the switch software agent. There are many options for timing probing packets, e.g., Address Resolution Protocol (ARP) request/reply, Internet Control Message Protocol (ICMP) request/reply.

Test packets are used to strengthen the effect of timing probing packets by adding extra loads to the software agent of the switch. We consider test packets with a random destination IP address and the broadcast destination Media Access Control (MAC) address is an ideal choice. By hitting the table-miss entry, each of them would be directed to the controller. Then the SDN controller will issue Packet-Out message to forward the test packet directly. As a result, the aim of burdening switch software agent is achieved.

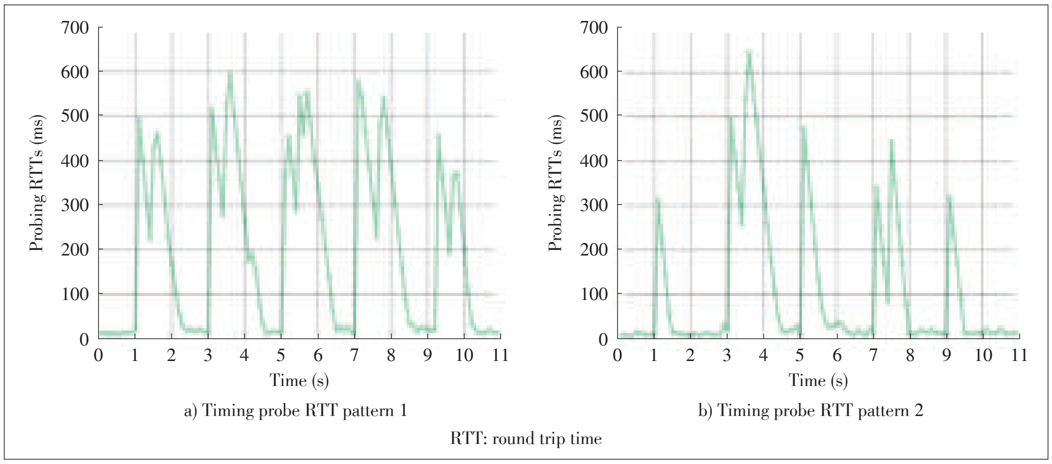
A data plane stream is a series of templates, which should go directly through the data plane to obtain more advanced information such as the specific conditions for indirect event-driven applications. We provide two templates here, as shown in **Fig. 3**. The first template has a steady rate *v* and packet size *p*, which is mainly used to probe volume-based statistic calculation and control method. The second has a rate distribution like a jump function, where three variables (*v*, *t*, *p*) determine the shapes of this template as well as the size of each packet, which is often used to probe the rate-based strategy.

The insight of the probing phase of counter manipulation attacks lies in that different downlink messages have diverse expenses for the downlink channel. Among the interaction approaches between the applications and the data plane, there are mainly three types of downlink messages, i.e., Flow-Mod, Statistics Query, and Packet-Out. Flow-Mod is the most expensive one, Statistics Query comes at the second and Packet-Out is rather lightweight. The latencies of timing probing packets will vary when the switch encounters different message types. Thus, the attacker could learn the type of message issued by the control plane. As for indirect data plane events, the statistic queries are usually conducted periodically by the applications. As a result, each of these queries would incur a small rise for the RTTs of timing probing packets. If a subsequent Flow-Mod is issued by the controller, there would be a double-peak. Based on the double-peak phenomenon, the attacker could even infer what statistic calculation methods the application is taking, such as volume-based or rate-based. With several trails of two templates above and the variations of *v* and *p* in a binary search approach, the attacker could quickly obtain the concrete conditions (volume/rate values, packet number/byte-based) that trigger the expensive downlink messages. The confidential information, such as the query period and exact conditions, helps the attacker permute the packet interval and packet size of each flow. By initiating a large number of flows, Flow-Mod of equal number would be triggered every period, making the hardware switch suffer extremely.

We use a simplified example (**Fig. 4**) to illustrate the attack.

Template Name	Coordinate Axis	Variables
Data plane stream with steady rate		*v*, *p*
Data plane stream with 0-1 rate		*v*, *t*, *p*

▲**Figure 3. Templates for a data plane stream.**



a) Timing probe RTT pattern 1

b) Timing probe RTT pattern 2

RTT: round trip time

◀**Figure 4.**
Timing-based patterns for the counter manipulation attack.

If an attacker obtains a series of successive double-peak phenomenon (Fig. 4a) with the input of data plane stream template 1, where *v* is a big value, and obtains a series of intermittent double-peak phenomenon (Fig. 4b), where *v* is also a significant value. The attacker could determine that packet number volume-based statistic calculation approach is sensitive to stream with a high pps. With the variations of *v* and *p*, the critical value of volume can be inferred to help conduct the attack.

4 FloodShield: Defending Data-to-Control Plane Saturation Attacks

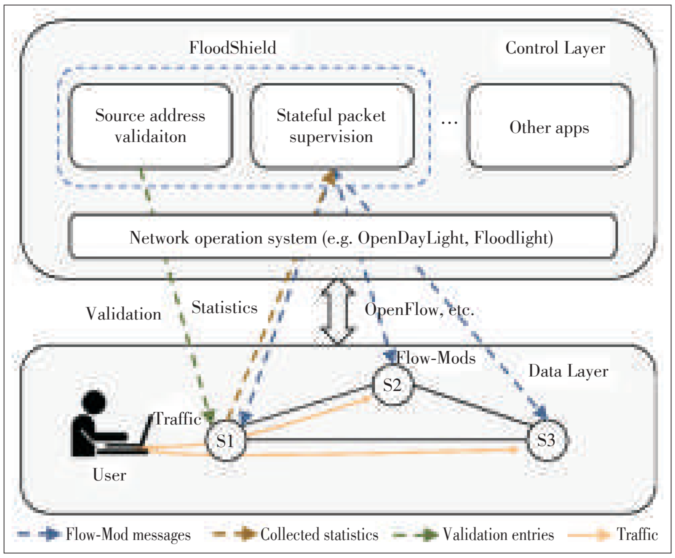
Floodshield [7] is a SDN defense framework against the data-to-control saturation attacks by combining two modules, i.e., source address validation and stateful packet supervision. The former validates the source addresses of the incoming traffic and filters the forged packets directly in the data plane, since attackers tend to commit attacks with a forged source address to hide the locations of attack sources. Based on it, the last module monitors the packet states of each real address and performs network service differentiation according to the evaluation scores and network resource usage.

As depicted in **Fig. 5**, the source address validation module works when a host connects to the SDN-enabled network. By snooping the address assignment mechanism procedure, the module maintains a global Binding Table at the controller to record the mapping between end hosts and their IP addresses. Based on the table, the module then takes advantage of the multi-table pipeline of OpenFlow to install filter rules in table 0 and install normal flow rules in the following tables. Packets with forged IP addresses are dropped in table 0 while trusted packets are directly forwarded to the non-filter flow tables.

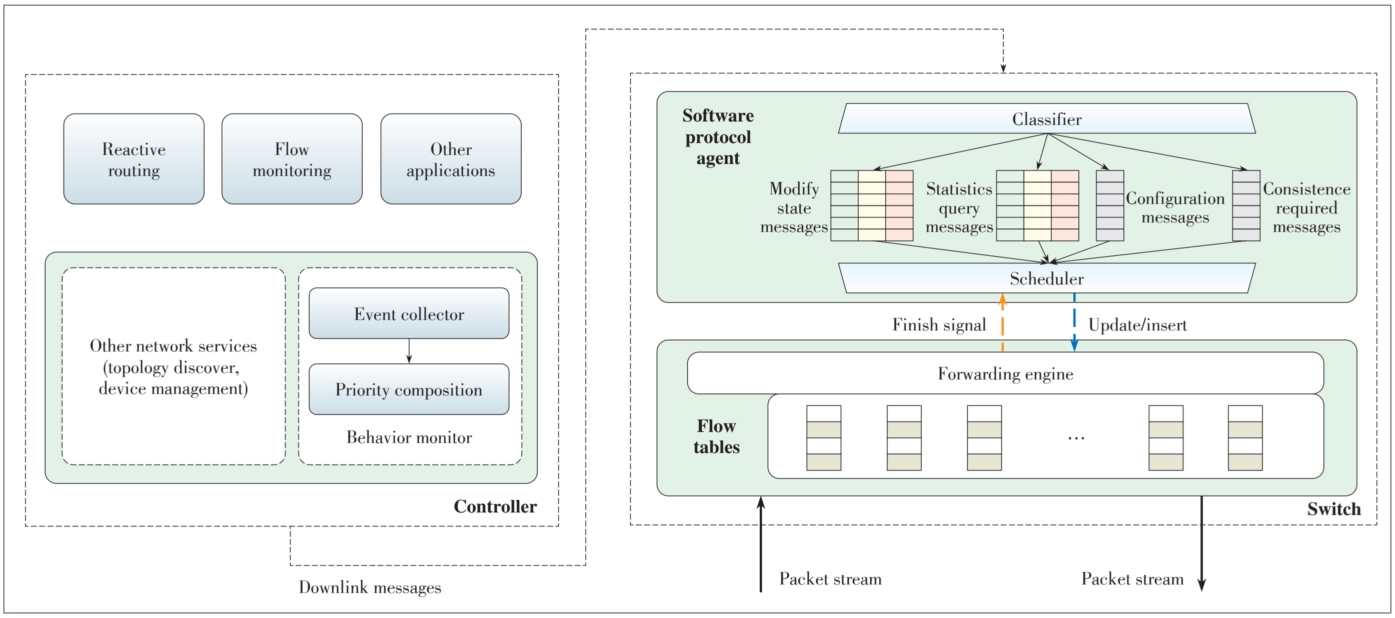
Since packets with real source addresses could also be harnessed to conduct attacks, a stateful packet supervision module is introduced to distinguish flows by traffic features and achieve differentiated services for different user dynamically. The module takes packet-in rate and average flow length as two metrics to evaluate user behavior. Users are divided into three levels according to their evaluation scores and allocated with different priorities. Flows with a high priority are processed as usual while those with a lower priority are limited on the rate or even dropped.

5 SWGuard: Defending Control Plane Reflection Attacks

The basic idea of SWGuard [8] is to discriminate good from evil, and prioritize downlink messages with discrimination results. SWGuard introduces a multi-queue scheduling strategy to achieve different latency for different downlink messages. The scheduling strategy is based on the statistics of downlink messages during the last period, which takes both fairness and efficiency into consideration. When the downlink channel is becoming congested, the malicious downlink messages are inclined to be put into a low-priority scheduling queue and the requirements of good messages are more likely to be satisfied. As shown in **Fig. 6**, SWGuard mainly redesigns two compo-



▲**Figure 5. Framework Design of FloodShield.**



▲Figure 6. Framework Design of SWGuard.

nents of SDN architecture. On the switch side, it changes the existing software protocol agent to multi-queue based structures. On the controller side, it adds a Behavior Monitor module as a basic service which assigns different priorities to different messages dynamically.

SWGuard redesigns the software protocol agent of the existing switch to prioritize the downlink messages. Since different types of downlink messages have diverse requirements, SWGuard summarizes the downlink messages into four categories: 1) Modify State Messages, 2) Statistic Query Messages, 3) Configuration Messages, and 4) Consistency Required Messages. It also designs a Classifier to classify the downlink messages into different queues accordingly. The first two types are related to behaviors of hosts and applications which are sensitive to latency and order, so a multi-queue is allocated for each. The latter two types inherit from the original single queue. With messages in the queues, a Scheduler is designed to dequeue the messages with a time-based scheduling algorithm. For queues with the highest priority are dequeued immediately, messages are dequeued immediately as they arrive. However, for queues with lower priority, different time interval is added to messages before dequeued.

To distinguish different downlink messages with different priorities, SWGuard proposes the novel abstraction of Host-Application Pair (HAP) and use it as the granularity for monitoring and statistics. Packets are recorded for each application of each user. Assuming there are K applications in the control plane, and N hosts in the data plane, packets should be categorized into $K \times N$ groups. SWGuard is designed as attack-driven. When the number of downlink messages in a period is less than a threshold, all packets are allocated with the highest priority. When the reflection attacks are detected, the SWGuard starts to calculate the penalty coefficient for each HAP by comparing their required resources with their real resource occupation. According to the coefficient, downlink messages are enqueued into queues with different priorities. Besides, multi-queues based software protocol agent may violate the consistency of some messages, which need to be sent in a particular order for correctness reasons. To address this issue, a coordination mechanism between the Behavior Monitor and Classifier in software protocol agent is designed.

6 Conclusions

While SDN has offered new opportunities to network automation and innovations, it has also introduced new security concerns. Securing the network infrastructure is crucial to the promotion and adoption of SDN. In this article, we review two SDN-targeted attacks, data-to-control saturation attacks, and control plane reflection attacks, along with the corresponding defense frameworks, FloodShield and SWGuard. The two attacks are both targeted at limited resources of SDN infrastructure, especially resources and limited processing capability of the data plane. Since hardware switching systems share many common designs like TCAM-based flow table, the SDN-targeted attacks also provide new perspectives to the security of other emerging architecture, e.g. the programmable data plane [10].

References

[1] 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.

[2] A. K. Nayak, A. Reimers, N. Feamster, and R. Clark, "Resonance: dynamic access control for enterprise networks," in *Proc. 1st ACM Workshop on Research on Enterprise Networking*, Barcelona, Spain, 2009, pp. 11-18. doi: 10.1145/1592681.1592684.

[3] R. Miao, H. Zeng, C. Kim, J. Lee, and M. Yu, "Silkroad: making stateful layer-4 load balancing fast and cheap using switching ASICS," in *Proc. Conference of the ACM Special Interest Group on Data Communication*, Los Angeles, USA, 2017, pp. 15–28. doi: 10.1145/3098822.3098824.

[4] A. R. Curtis, J. C. Mogul, J. Tourrilhes, et al., "Devoflow: scaling flow management for high-performance networks," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 254–265, 2011. doi: 10.1145/2043164.2018466.

[5] A. Wang, Y. Guo, F. Hao, T. Lakshman, and S. Chen, "Scotch: elastically scaling up SDN control- plane using vswitch based overlay," in Proc. *10th ACM International on Conference on Emerging Networking Experiments and Technologies*, Sydney, Australia, 2014, pp. 403–414. doi: 10.1145/2674005.2675002.

[6] X. Jin, H. H. Liu, R. Gandhi, et al., "Dynamic scheduling of network updates," in *ACM SIGCOMM Computer Communication Review*, Chicago, USA, 2014, pp. 539–550. doi: 10.1145/2619239.2626307.

[7] M. Zhang, J. Bi, J. S. Bai, et al., "FloodShield: securing the SDN infrastructure against denial-of-service attacks," in *17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustComm18)*, New York, USA, 2018, pp. 687–698. DOI:10.1109/TrustCom/BigDataSE.2018.00101.

[8] M. H. Zhang, G. Y. Li, L. Xu, et al., "Control plane reflection attacks in SDNs: new attacks and countermeasures," in *21st International Symposium on Research in Attacks, Intrusions and Defenses (RAID18)*, Heraklion, Greece, 2018, pp. 161–183.

[9] J. Sonchack, A. Dubey, A. J. Aviv, J. M. Smith, and E. Keller, "Timing-based reconnaissance and defense in software-defined networks," in *Proc. 32nd Annual Conference on Computer Security Applications*, Los Angeles, USA, 2016, pp. 89–100. doi: 10.1145/2991079.2991081.

[10] P. Bosshart, D. Daly, G. Gibb, et al., "P4: programming protocol-independent packet processors," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 87–95, 2014. doi: 10.1145/2656877.2656890.

Manuscript received: 2018-06-19

Biographies

BAI Jiasong (bjs17@mails.tsinghua.edu.cn) received his B.S. degree from Department of Computer Science and Technology, Tsinghua University, China in 2017. He is currently a master student in Department of Computer Science and Technology, Tsinghua University. His research interests include SDN, NFV and programmable data plane.

ZHANG Menghao (zhangmh16@mails.tsinghua.edu.cn) received his B.S. degree from Department of Computer Science and Technology, Tsinghua University, China in 2016. He is currently a Ph.D. student in Department of Computer Science and Technology, Tsinghua University. His research interests include the availability and security of SDN and NFV.

BI Jun (junbi@tsinghua.edu.cn) received his B.S., C.S., and Ph.D. degrees from 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. He is also the Director of the Future Network Theory and Application Research Division at Beijing National Research Center for Information Science and Technology. 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-chairs 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.

Call for Papers

ZTE Communications Special Issue on

Data Intelligence

Data-driven intelligence, or data intelligence, is a new form of AI technologies that leverages the power of big data. It is becoming an extremely active research area with broad area of applications such as computer vision, medial and healthy, intelligent transportation system, multimedia system, and social network. With the huge volume of data available in various domains, big data brings opportunities to boost the performance of artificial intelligent system with advanced machine learning especially deep learning techniques. On the other hand, it also presents unprecedented challenges to manage and exploit big data for a variety of applications. This special issue seeks original articles describing development, relevant trends, challenges, and current practices in the field of big data and artificial intelligence. Position papers, and case studies are also welcome.

Appropriate topics include, but are not limited to,

- Computer vision with big data
- Big medial data
- Big transportation data
- Deep learning for big data
- Applications of big data intelligence
- Semantic of heterogeneous data

Guest Editors

- XU Cheng-zhong, Wayne State University (USA)
- QIAO Yu, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences (China)

Important Dates

- Submission Due: May 1, 2019
- Review and Final Decision Due: Jun. 10, 2019
- Final Manuscript Due: Jul. 1, 2019
- Publication Date: Sept. 25, 2019

Manuscript Preparation and Submission

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.

Please submit your manuscript through the online submission system of the journal: https://mc03.manuscriptcentral.com/ztecom.

DOI: 10.19729/j.cnki.1673-5188.2018.04.003
http://kns.cnki.net/kcms/detail/34.1294.TN.20181113.0847.002.html, published online November 13, 2018

SDN Based Security Services

ZHANG Yunyong, XU Lei, and TAO Ye
(China Unicom, Beijing 100032, China)

Abstract

With the development and revolution of network in recent years, the scale and complexity of network have become big issues. Traditional hardware based network security solution has shown some significant disadvantages in cloud computing based Internet data centers (IDC), such as high cost and lack of flexibility. With the implementation of software defined networking (SDN), network security solution could be more flexible and efficient, such as SDN based firewall service and SDN based DDoS-attack mitigation service. Moreover, combined with cloud computing and SDN technology, network security services could be lighter-weighted, more flexible, and on-demanded. This paper analyzes some typical SDN based network security services, and provide a research on SDN based cloud security service (network security service pool) and its implementation in IDCs.

Keywords

SDN; network security; cloud security service

1 Introduction

The introducing of software defined networking (SDN) and network function virtualization (NFV) solution changes the network significantly: general hardware, virtualization software, and programmable services. With SDN and NFV, the network operation and maintenance cost is cut down, the utilization of resources is improved, the network flexibility is increased, and the time-to-market of new services is considerably decreased [1].

Therefore, SDN and NFV are considered as the innovation technology for telecommunications network evolution.

However, SDN and NFV also bring new security challenges for telecommunication networks. These new security challenges include:

- The physical security boundary becomes ambiguous, but the network is still protected by static-deployed security devices/appliances and passive security responses according to provisioned security policies, which leads to low-efficient security operation and maintenance, and delayed responses to security attacks [2].
- Network elements are created and deleted dynamically, but security policies cannot be updated accordingly because most security policies are updated by manual operations. Therefore, security management and protection could not be provided for network elements dynamically and automatically.
- SDN and NFV systems lack centralized security policy scheduling cross different devices and services from different vendors; no collaboration between devices always led to inconsistence of security policy.
- With virtualization, eastbound and westbound traffic flow information cross different virtual machines (VMs) may not be captured and analyzed since current security devices/appliances could not recognize those traffic flows. Thus, the current traffic monitoring and interception mechanisms do not apply to the virtualized system and the security view of the operator's network may not be provided.

Therefore, the static, passive, separate and manual operation of traditional security defense systems does not work for SDN/NFV networks. A dynamic, proactive, centralized and intelligent security management capability is needed.

The new framework shall utilize the key advantages of SDN/NFV technology, such as on-demand capacity scale-in/scale-out, virtualization, centralization, and decoupling the data and control planes. This new framework is a layered one and shall provide security orchestration, centralized and automated security policy management, and intelligent security analysis and response [3].

This paper analyzes the security challenges of SDN/NFV network to identify the requirements, defines a software-defined security framework, and then describes the functionalities of each module; finally the reference implementations are also provided.

The software-defined security framework may include the following components [4]:

- Security controller: A security controller is introduced to centralize and automatically control all the security devices, to gather flow traffic information and system logs from them. The security controller also works as SDN/NFV applications; it interacts with the SDN controller and NFV management and orchestration (MANO) to achieve the flow scheduling and auto scaling of the virtualized security functions. New technology (such as big data and artificial intelligence) can be easily introduced to further improve the intelligence of the security controller [5].

- SDN network: The SDN controller offers flow scheduling function, whole network topology and traffic flow information to the security controller.
- NFV system: The NFV MANO provides virtual machine resources to the virtualized security function and the run-time status of these VMs.

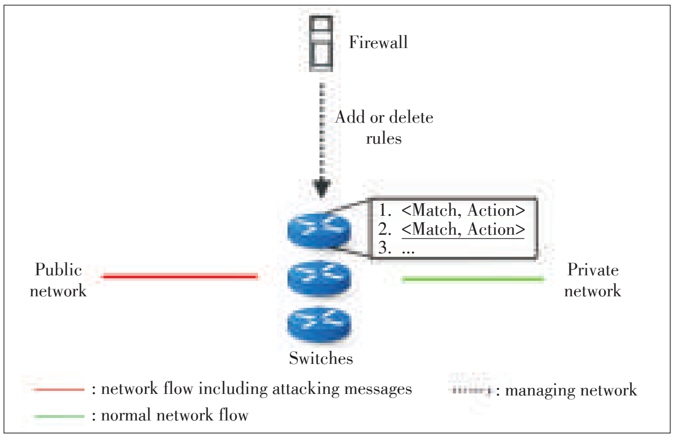
2 Typical SDN Based Security Services

2.1 SDN Based Firewall Service

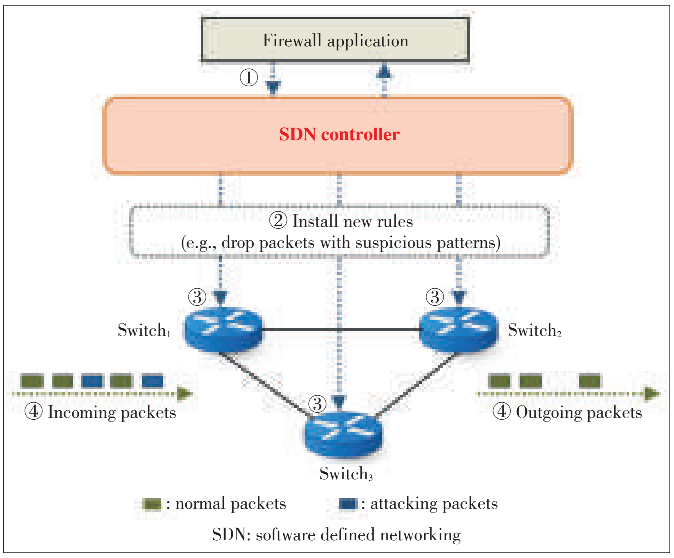
The SDN based firewall service is centralized and can manage network resources and manage firewall rules flexibly. As shown in **Fig. 1**, a SDN based centralized firewall management system could manage firewall rules and switches more flexible.

By using SDN controller, a packet-filtering strategy, which is issued by the firewall application (software or hardware), could be easily converted to a flow table through the controller. However, a protocol between the controller and switches (e.g., openflow and netconf) is only able to match up to the transmission control protocol (TCP) layer at present, and there is no corresponding field to set the identification information of data packets above the TCP layer. Therefore, it cannot be achieved to identify the information above the TCP layer firewall strategy without changing the protocol [6].

Fig. 2 shows an example scenario of centralized firewall service for switches and the process of filtering the attacking network messages through this security system. This scenario concentrates on SDN switches and shows that how a user can manage a centralized firewall service.



▲Figure 1. Centralized firewall service in intra-domain.



SDN: software defined networking

▲Figure 2. An example scenario of centralized firewall service.

As a precondition for this scenario, a security manager should specify a new policy to firewall application when the information about a new attacking network message is recognized. In order to prevent packets from including this attacking network messages, the user adds the new policy to the firewall application running on top of the SDN controller.

The process includes four steps:

- Step 1: A firewall application (could be software or hardware) should specify new security policies when the attacking network messages is warned. And these new policies could be added to the SDN controller.
- Step 2: A new flow entry might be distributed to each switch by a SDN controller after installing it. Therefore, the SDN controller sends a flow insert operation that contains the rules (e.g., "drop packets with the attacking network messages file") to all the SDN switches.

The reported new attacking network messages is either a known attacking network messages or a "zero-day" attacking network messages. As for a known attacking network messages, some mechanisms such as "signatures" and "thumbprints" are developed for firewall service to detect and defend it. However, for a "zero-day" attacking network messages, it should be scanned and detected before any countermeasure is applied to defend it. Attacking network messages deliver malicious payloads that could exploit some vulnerable applications or services. Those attacking network messages might be detected by inspecting the packet payload.

- Step 3: An SDN switch adds a flow entry dropping future packets with the attacking network messages file to its flow table when receiving the flow insert operation about the attacking network messages file. After that, the SDN switch can drop the packets with the attacking network messages file.
- Step 4: When receiving any packets with attacking network

messages file, an SDN switch completely drops the packets. Any packets with attacking network messages files cannot be passed to the switches under the applied rules.

When an SDN switch receives a type of packet that it has not processed before, it deletes this packet and sends a report to the controller about this kind of packets. The controller analyzes whether this is an attack. If this is an attack, the controller sends a message to the firewall application and Step 1 will be executed. If not, the controller keeps a regular flow entry to tell the switches how to handle this sequence of afterwards packets.

2.2 SDN Based Honeypot Service

The SDN-based centralized honeypot can manage honeypot places. As shown in **Fig. 3**, a centralized honeypot manages switches and new routing paths to the honeypots to attract attackers to a place used as a trap. The honeypot is configured as the intended attack target and reports the collected information to the centralized honeypot service [7].

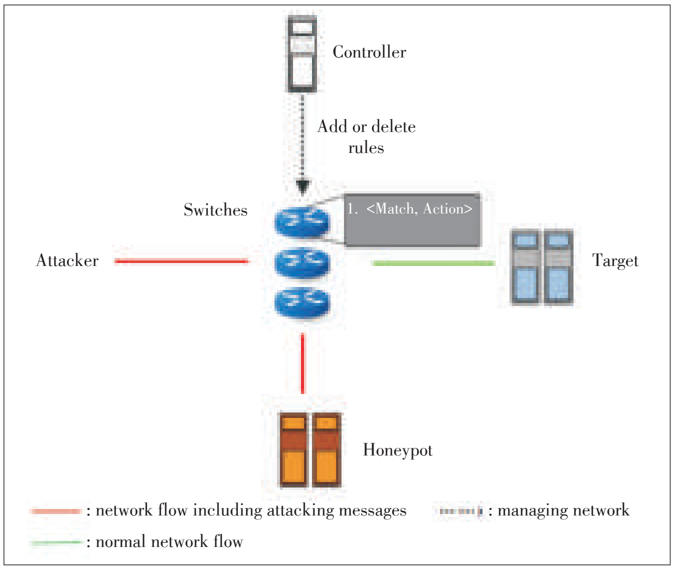
Fig. 4 shows a centralized honeypot service for switches. Adding a routing path to a honeypot scenario shows that how a security manager can use a centralized honeypot system. This scenario concentrates on SDN switches.

The process of adding a routing path to a honeypot instead of the actual target includes four steps as follows:

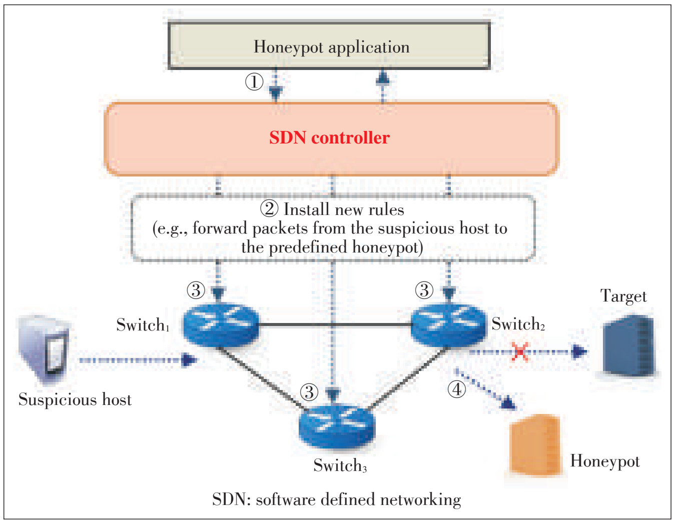
- Step 1: A honeypot application installs new rules to the SDN controller

A honeypot application should specify new rules when the information about a suspicious host is reported. In order to monitor the traffic from the suspicious host, the new rules (e.g., "forward packets from the suspicious host to a honeypot") is added to the SDN controller by honeypot application running on top of the SDN controller.

- Step 2: The SDN controller distributes new rules to appropriate SDN switches



▲Figure 3. Centralized honeypot service in intra-domain.



SDN: software defined networking

▲Figure 4. An example scenario for centralized honeypot service.

The new rules might be distributed to each switch by the SDN controller after installing it. Therefore, the SDN controller sends a flow insert operation that contains the rule (e.g., "forward packets from the suspicious host to a honeypot") to all the SDN switches.

- Step 3: All the SDN switches apply the new rules into their flow tables.

All the SDN switches add a flow entry forwarding future packets from the suspicious host to a honeypot to their flow tables when receiving the flow insert operation about the suspicious host. After that, the SDN switch can forward the packets from the suspicious host to a honeypot.

- Step 4: An SDN switch executes the new rules to support honeypot service

When receiving any packets from the suspicious host, an SDN switch forwards the packets to a honeypot. In this way, any packets from the suspicious host cannot be passed to an actual target host switch under the applied rules. The forwarded packets are collected in the honeypot.

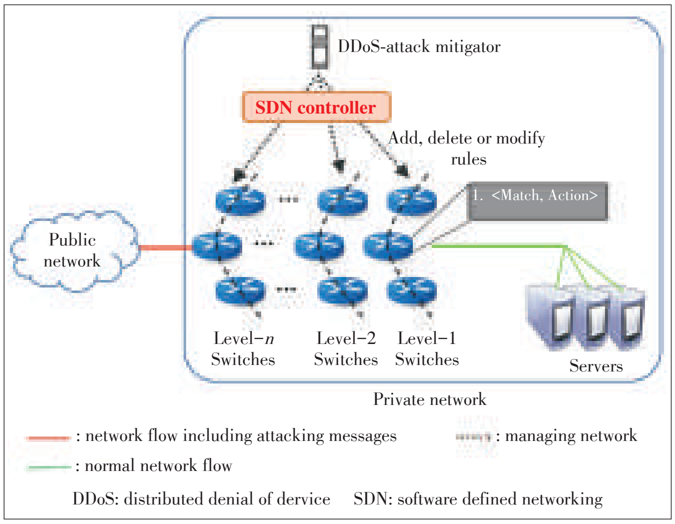
2.3 SDN Based DDoS-Attack Mitigation Service

Fig. 5 shows a centralized distributed denial of service attack (DDoSattack) mitigation service. This service adds, deletes or modifies rules to each switch. Unlike the centralized firewall service, this service is mainly on the inter-domain level.

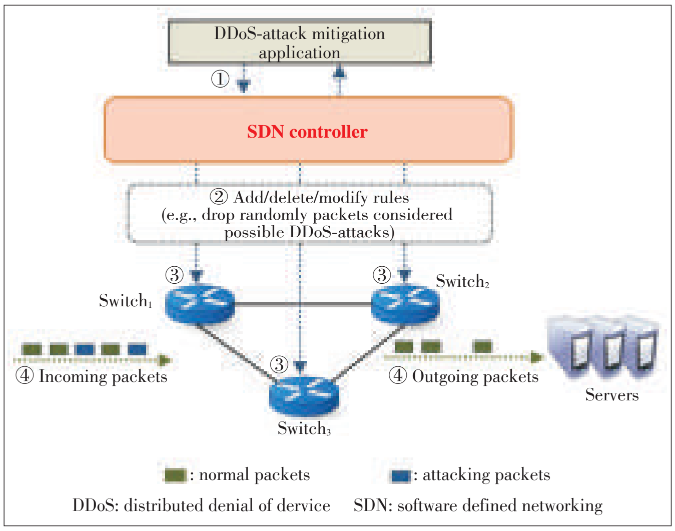
Fig. 6 shows an example scenario of centralized DDoS-attack mitigation for stateless servers. The process against Domain Name Services (DNS) DDoS attacks include four steps as follows.

- Step 1: A mitigation application installs new rules to SDN controller

A DDoS-attack mitigation application should specify new rules when a new DDoS-attack is detected. In order to prevent



▲Figure 5. Centralized DDoS-attack mitigation service in inter-domain.


DDoS-attack mitigation application
①
SDN controller
② Add/delete/modify rules (e.g., drop randomly packets considered possible DDoS-attacks)
③
Switch₁
Switch₂
Switch₃
④ Incoming packets
④ Outgoing packets
Servers
: normal packets
: attacking packets
DDoS: distributed denial of dervice
SDN: software defined networking

▲Figure 6. An example scenario for centralized DDoS-attack mitigation for stateless servers.

packets from reaching servers to waste the servers' resources, the new rule (e.g., "drop DDoS-attack packets randomly with some probability") is added to the SDN controller. This rule addition is performed by DDoS-attack mitigation application running on top of the SDN controller.

- Step 2: An SDN controller distributes new rules to appropriate switches

The new rules might be distributed to each switch by a SDN controller after installing it. Therefore, the SDN controller sends a flow insert operation that contains the rule (e.g., "drop randomly packets considered DDoS attacks with a certain probability") to all the SDN switches.

- Step 3: All the SDN switches apply new rules into their flow tables

All the SDN switches add a flow entry to their flow tables for dropping the packets in a DDoS-attack when receiving the flow insert operation about the DDoS-attack mitigation. After that, the SDN switch can drop these packets with a probability proportional to the DDoS-attack severity.

- Step 4: An SDN switch executes new rules to mitigate DDoS-attacks

An SDN switch completely drops the packets selected when receiving any packets in a DDoS-attack.

Fig. 7 shows an example scenario where the SDN controller can manage a centralized DDoS-attack mitigation.

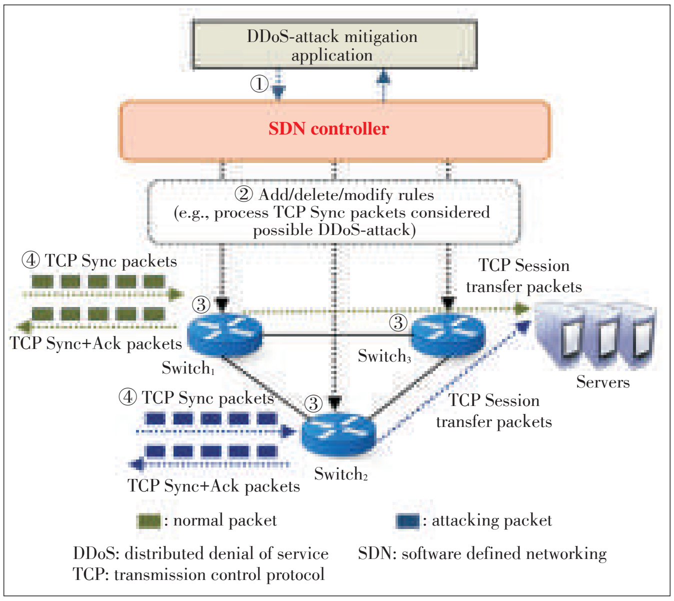
- Step 1: A mitigation application installs new rules to the SDN controller

A DDoS-attack mitigation application should select the switch that performs the role of proxy for TCP service. New rule addition is performed by DDoS-attack mitigation application running on top of the SDN controller.

- Step 2: A SDN controller distributes new rules to appropriate switches

The installed new rules might be distributed to appropriate switches for DDoS attack mitigation by an SDN controller. The SDN controller then sends a flow insert operation that contains the rule (e.g., "generate TCP Sync+Ack for packets considered DDoS attacks") to all the SDN switches. Therefore, a new rule is installed into the selected switch so that it can generate TCP Sync-Ack packets for TCP Sync as request. If the same requests arrive much more frequently than the expected rate, the SDN controller selects a new switch to serve the role of server. For the normal TCP Sync, the switch transfers the TCP session to the corresponding server in the private network. It can also be managed centrally by the SDN controller such that a security manager can determine security policies for their services.

- Step 3: All the SDN switches apply the new rule into their flow tables


DDoS-attack mitigation application
①
SDN controller
② Add/delete/modify rules (e.g., process TCP Sync packets considered possible DDoS-attack)
④ TCP Sync packets
TCP Sync+Ack packets
③
Switch₁
Switch₃
Switch₂
TCP Session transfer packets
Servers
④ TCP Sync packets
TCP Session transfer packets
TCP Sync+Ack packets
: normal packet
: attacking packet
DDoS: distributed denial of service
SDN: software defined networking
TCP: transmission control protocol

▲Figure 7. An example scenario for centralized DDoS-attack mitigation for stateful servers.

All the SDN switches add a flow entry to their flow tables for dropping future packets in any DDoS-attacks when receiving the flow insert operation about the DDoS attacks. After that, the SDN switch can generate TCP Sync-Ack packets with a probability proportional to the DDoS-attack severity.

- Step 4: An SDN switch executes the new rule to mitigate DDoS-attacks

An SDN switch completely responds to TCP Sync packets from an adversary host randomly when receiving DDoS-attack packets. DDoS-attack requests for stateful servers are handled by the switches instead of actual servers.

3 SDN Based Cloud Security Services

With the implementation of SDN, network security service could be more flexible and efficient. As shown in **Fig. 8**, the SDN based cloud security service solution has two main system modules: the SDN based security controller and security service pool [8].

The SDN based security controller provides security for service network control and VNFs management. Based on the SDN controller and cloud computing platform, this module implements SDN based security control, secure VNF and service management, cloud service customer (CSC) management, and service-level agreement (SLA) monitoring [9].

Based on virtualization and NFV technology, the security service pool is implemented with multiple network security VNF or simply integrated with third-party security software with open API. The security resource pool is a combination of original physical security devices (hardware boxes such as firewall, web application firewall (WAF), intrusion prevention system (IPS)) and virtual security devices (such as virtual firewall, virtual wireless application protocol (WAP), and virtual IPS). These devices are abstracted with basic features and unified interface, so that the security controller can orchestrate these security functions, set security policies to them, and obtain flow traffic information from them [10].

Compared with traditional hardware-based network security solutions, SDN based cloud security services have several advantages such as multiple service types and more flexible functions in future networks, Internet data center (IDC) and cloud computing platforms.

Benefited from NFV technology, the security service pool could provide multiple virtual security resources, with lower cost and fewer hardware resources. Many small- and medium-size IDC providers only implement minimized security function, which could only provide basic security functions such as firewall and anti-DDoS devices, due to the high cost of security hardware. With the implementation of security service pool, the IDC provider could provide more types of security services with lower cost, while the security of IDC and tenant network is also enhanced.

Based on the SDN and cloud computing technology, the security controller could bring more flexible security service functions. Security resources could be provided and modified on demand. The service function chain could provide CSC private security network. Security resources could automatically migrate with the migration of tenant network and resources.

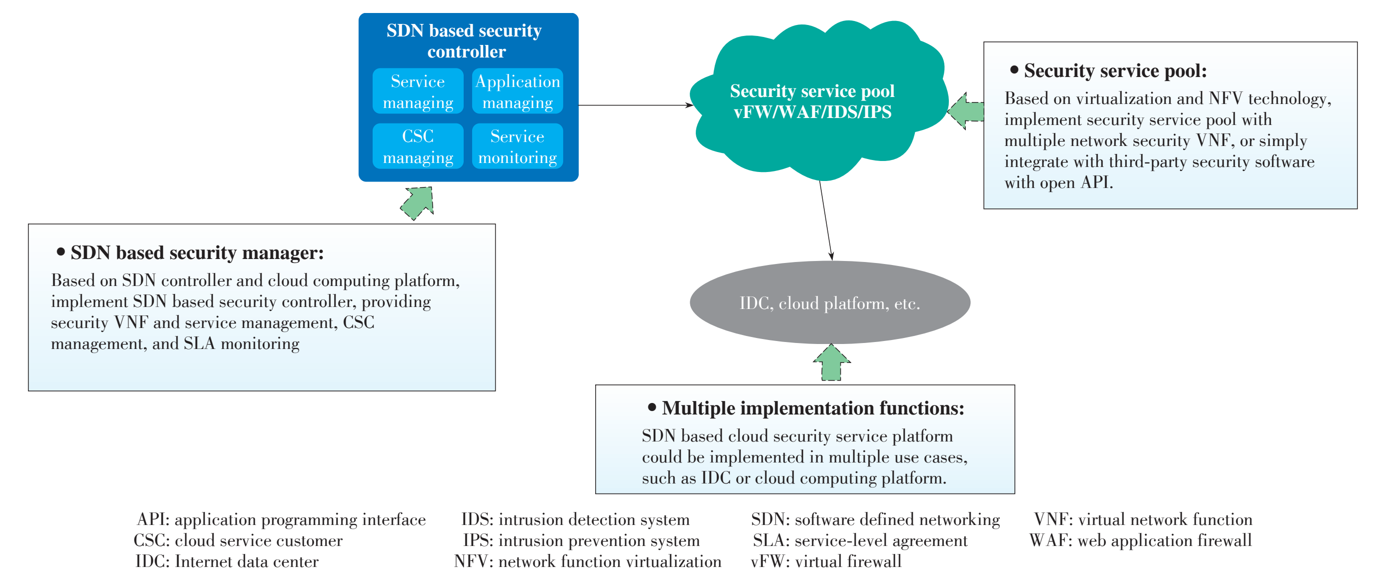
Fig. 9 defines a process of SDN based cloud security service to the CSC, which include the following three steps.

- Step 1: CSC requests cloud security services

CSC could request cloud security services with the system information (such as VLAN id, and IP), service type (such as vFW and WAF), and service quantity and configuration.

- Step 2: Security controller handles the request

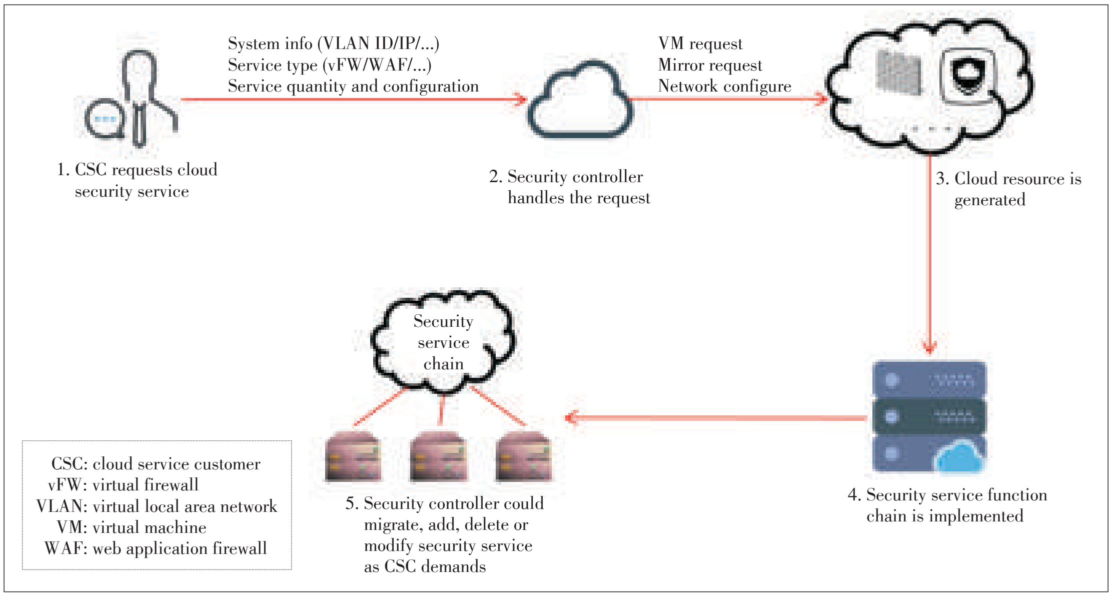
Configuring the CSC request (with some necessary identifi-


SDN based security controller
Service managing
Application managing
CSC managing
Service monitoring
Security service pool vFW/WAF/IDS/IPS
IDC, cloud platform, etc.

- **Security service pool:** Based on virtualization and NFV technology, implement security service pool with multiple network security VNF, or simply integrate with third-party security software with open API.
- **SDN based security manager:** Based on SDN controller and cloud computing platform, implement SDN based security controller, providing security VNF and service management, CSC management, and SLA monitoring
- **Multiple implementation functions:** SDN based cloud security service platform could be implemented in multiple use cases, such as IDC or cloud computing platform.

API: application programming interface
CSC: cloud service customer
IDC: Internet data center
IDS: intrusion detection system
IPS: intrusion prevention system
NFV: network function virtualization
SDN: software defined networking
SLA: service-level agreement
vFW: virtual firewall
VNF: virtual network function
WAF: web application firewall

▲**Figure 8. SDN based cloud security service.**


System info (VLAN ID/IP/...)
Service type (vFW/WAF/...)
Service quantity and configuration
VM request
Mirror request
Network configure
1. CSC requests cloud security service
2. Security controller handles the request
3. Cloud resource is generated
Security service chain
4. Security service function chain is implemented
5. Security controller could migrate, add, delete or modify security service as CSC demands
CSC: cloud service customer
vFW: virtual firewall
VLAN: virtual local area network
VM: virtual machine
WAF: web application firewall

◀**Figure 9.**
Process of SDN based cloud security service.

cation methods), the security controller could handle the request and send another request for the basic cloud resource (VM, cloud mirror type, network configuration, etc.) to the cloud management platform.

- Step 3: Cloud resource generation and service function chain implementation

The cloud management platform handles the request, generates the resource, and then sends the resource information back to the security controller. The controller implements the security function chain with the demands of CSC.

4 Conclusions

With the development of new network technology such as SDN and NFV, network security faces some new challenges, threats, but also opportunities. Combining and implementing SDN and traditional network security functions could bring more flexible, efficient, lower cost network security services to the end customers. As the new Information and communication technology (ICT) technologies such as 5G and artificial intelligence (AI) are being implemented in the Internet and IDC, more network and information security challenges would rise up. Therefore, implementing new ICT technology in security industry would be a new trend [11].

References

[1] J. Carapinha, P. Feil, P. Weissmann, et al., "Network virtualization−opportunities and challenges for operators," in *Future Internet - FIS* 2010, J. Carapinha, P. Feil, P. Weissmann, et al. eds. Berlin/Heidelberg, Germany: Springer Berlin Heidelberg, 2010, pp. 138−147.

[2] D. A. Joseph, A. Tavakoli, and I. Stoica, "A policy-aware switching layer for data centers," in *Proc. ACM SIGCOMM* 2008 *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Seattle, USA. 2008, pp. 51−62. doi:10.1145/1402958.1402966.

[3] *Security Requirements and Reference Architecture for Software-Defined Networking*, ITU-T X.1038, Oct. 2016.

[4] *Service Function Chaining (SFC) Architecture*, IETF RFC 7665, Oct. 2015.

[5] Z. Y. Hu, M. W. Wang, X. Q. Yan, et al., "A comprehensive security architecture for SDN," in *IEEE 18th International Conference on Intelligence in Next Generation Networks*, Paris, France, 2015, pp. 30−37. doi:10.1109/ICIN.2015.7073803.

[6] *Security Services Using the Software-Defined Networking*, ITU-T X.1042, Sept. 2018.

[7] R. Bifulco and G. Karame G, "Towards a richer set of services in software-defined networks," in *2014 Workshop on Security of Emerging Networking Technologies*, San Diego, USA, 2014. doi:10.14722/sent.2014.23006.

[8] *Functional Requirements of Software-Defined Networking*, ITU-T Y.3301, Sept. 2016.

[9] *Security Framework for Cloud Computing*, ITU-T X.1601, Oct. 2015.

[10] *Security Requirements for Software as a Service Application Environments*, ITU-T X.1602, Mar. 2016.

[11] *Functional Architecture of Software-Defined Networking*, ITU-T Y.3302, Jan. 2017.

Manuscript received: 2018-07-01

Biographies

ZHANG Yunyong (zhangyy@chiaunicom.cn) serves as President of China Unicom Research Institute, Vice President of the Ministry of Industry and Information Technology SDN Industry Alliance, China, Vice President of the Technical Committee for New Prominent Forum in China Institute of Telecommunications. He is also a professor-level senior engineer, outstanding member of China Computer Federation, member of the 13th National Committee of CPPCC, national candidate for the Project of Millions of Talents. He was awarded the State Department Special Allowance and the title of "China's Middle-aged and Young Experts with Outstanding Contributions". He has achieved 64 authorized patents and 37 software copyrights.

XU Lei (xulei56@chinaunicom.cn) is a manager of cloud computing with China Unicom Research Institute. His research interests include cloud computing, SDN/NFV, and information security. He has achieved 20 authorized patents and 20 software copyrights. He is an editor of very first worldwide ITU cloud computing standards.

TAO Ye (taoy10@chinaunicom.cn) is the director of the Cloud Security Research Group of China Unicom Research Institute. His research interests include information security, network security, SDN/NFV security, and anti-telecom fraud. He has achieved 10 authorized patents and 10 software copyrights. He is the chief-editor of 2 published ITU standards.

DOI: 10.19729/j.cnki.1673-5188.2018.04.004
http://kns.cnki.net/kcms/detail/34.1294.TN.20181121.1527.004.html, published online November 21, 2018

Optimization Framework for Minimizing Rule Update Latency in SDN Switches

CHEN Yan [1,2], WEN Xitao[3], LENG Xue [1], YANG Bo[4], Li Erran Li [5], ZHENG Peng [6], and HU Chengchen[6]

(1. College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China;
2. Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208, USA;
3. Google Inc., Mountain View, CA 94043, USA;
4. Microsoft, Shanghai 200000, China;
5. Uber Technologies Inc., San Francisco, CA 94103, USA;
6. School of Electronic and Information Engineering, Xi 'an Jiaotong University, Xi 'an 710049, China)

Abstract

Benefited from the design of separating control plane and data plane, software defined networking (SDN) is widely concerned and applied. Its quick response capability to network events with changes in network policies enables more dynamic management of data center networks. Although the SDN controller architecture is increasingly optimized for swift policy updates, the data plane, especially the prevailing ternary content-addressable memory (TCAM) based flow tables on physical SDN switches, remains unoptimized for fast rule updates, and is gradually becoming the primary bottleneck along the policy update pipeline. In this paper, we present RuleTris, the first SDN update optimization framework that minimizes rule update latency for TCAM-based switches. RuleTris employs the dependency graph (DAG) as the key abstraction to minimize the update latency. RuleTris efficiently obtains the DAGs with novel dependency preserving algorithms that incrementally build rule dependency along with the compilation process. Then, in the guidance of the DAG, RuleTris calculates the TCAM update schedules that minimize TCAM entry moves, which are the main cause of TCAM update inefficiency. In evaluation, RuleTris achieves a median of <12 ms and 90-percentile of < 15ms the end-to-end perrule update latency on our hardware prototype, outperforming the state-of-the-art composition compiler CoVisor by ~ 20 times.

Keywords

SDN; SDN-based cloud; network management; access control; unauthorized attack

1 Introduction

As a new network architecture proposed ten years ago, software defined networking (SDN) has been well researched in both academia and industry. The main reason that SDN is so concerned is its ability to dynamically change the network states in response to the global view. However, the response time to the network events determines how many new network applications can become practical. For example, the carrier network has a strict 50 ms requirement for failure recovery [1], entailing a 10 ms to 25 ms delay budget for implementing the rerouting rules. Traffic engineering in data centers has a delay budget as short as 100 ms for the entire control loop [2], leaving less than 20 ms delay budget for implementing flow rules. The advanced malware quarantine [3] in enterprise networks has an even stricter delay budget since the threat detection is done at near line-rate and the quarantine decisions need to take effect as fast as possible.

The processing delay of a user request can be roughly divided into four parts: latency inside the controller, inside switches, and passing through the northbound and southbound communication channels. The transmission delay in the communication channel can be ignored and the recent advances on SDN controller architecture greatly shorten the processing latency of the control plane, which leaves the rule installation latency the primary bottleneck for the SDN control loop. Specifically, the recent measurement [4] exhibits a rule installation delay ranging from 33 ms to 400 ms with a moderate to high flow table utilization on three commercial OpenFlow switches using ternary content addressable memory (TCAM), which is the mainstream hardware to implement OpenFlow compatible

This work is supported by National Key R&D Program of China under Grant No. 2017YFB0801703 and the Key Research and Development Program of Zhejiang Province under Grant No. 2018C01088.

flow tables[1]. In addition, the measurement also finds that the switches can "periodically or randomly stop processing control plane commands for up to 400 ms", which further exacerbates the rule installation latency.

To reduce the latency inside switches, some existing works optimize the policy updates at different levels of the pipeline, however, their improvements are limited. Dionysus [5], for example, significantly reduces multi-switch policy update latency caused by suboptimal scheduling. CoVisor [6] and our previous short paper [7] minimize the number of rule updates sent to switches through eliminating redundant updates. However, since both approaches do not change the update mechanism on physical switches, they all suffer from the aforementioned per-rule update bottleneck. Existing TCAM update optimization techniques, on the other hand, are either dependent on specialized multi-stage Static Random Access Memory (SRAM)/TCAM structure [8]–[10] or only applicable to single-field longest prefix matching [11].

Based on our research, the latency bottleneck within the TCAM-based SDN switches is introduced by policy update and the TCAM update latency is the single dominant factor of the rule update latency. Interestingly, although a single entry update in TCAM usually has a constant sub-millisecond delay, we observe that an OpenFlow rule update sometimes triggers hundreds to thousands of unnecessary entry moves in TCAM to maintain rule dependency due to its unawareness of the minimum dependency information.

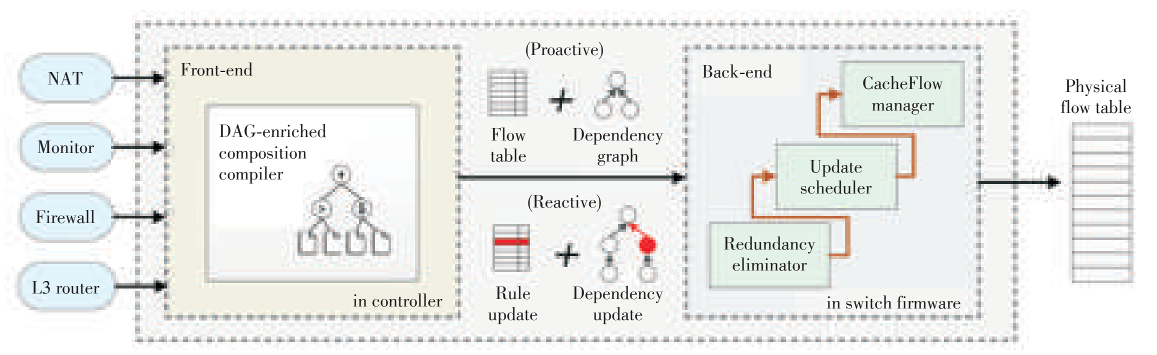
In this paper, we present RuleTris, the first optimization framework for modular composition achieving minimum rule size and optimal rule update cost in TCAM. Our study reveals that the minimum dependency graph (DAG) [10], [12], [13] is the key information towards optimal rule updates. Compared with rule priorities, the DAG is a more fundamental and precise representation of the rule dependency. The DAG not only minimizes the number of rule updates sent to switches, but also minimizes the cost of individual rule updates by cutting 90% to 99% of TCAM micro operations.

As depicted in **Fig. 1**, RuleTris is consisted of a front-end and a back-end. The front-end is a generic policy compiler that produces DAGs while composing multiple flow tables. The DAG produced by the front-end along with the flow table is then passed to the back-end for update optimization. The RuleTris back-end is a set of hardware-specific optimizers that map the DAG into a sequence of TCAM entry moves. The optimizers minimize the flow table size and the number of entry moves by exploiting the minimum dependency information.

To realize such an optimization framework, the primary challenge is to generate DAG efficiently. In fact, the existing DAG extraction algorithm is prohibitively time consuming for our target latency [13]. To this end, we embrace the policy composition paradigm [14]. Our previous short paper proposes to preserve rule dependency within Net Kleene Algebra with Tests (NetKAT) policy compiler [15] to reduce the computation. Extending it for generic policy compilation is quite non-trivial since a common flow table abstraction needs to be employed in the dependency reservation algorithms. Furthermore, to minimize the compilation overhead, the DAG needs to be compiled incrementally as policies evolve over time. On the back-end, an optimal while efficient scheduling algorithm is also needed to map the incremental graph changes into minimum TCAM entry moves.

RuleTris solves these challenging problems with the following contributions.

1) We develop general dependency preserving algorithms that preserve DAG along with flow table composition. The algorithms achieve efficiency by exploiting the dependency implications of composition operators. The algorithms are generic to SDN policy languages that employ policy compositions (sequential, parallel and priority), and are guaranteed to produce the minimum DAG.
2) We further speed up the compilation by incrementally compiling flow table changes. We employ incremental compilation techniques and develop algorithms to handle incremental DAG compositions.
3) We design an efficient and generic front-end policy compiler that generates DAG along with flow table compositions.



DAG: dependency graph NAT: network address translation

Figure 1. ▶ **Overview of RuleTris optimization framework.**

[1] Our survey indicates that at least 32 out of all 48 series of OpenFlow supported switches from 13 major vendors use TCAM to implement OpenFlow compatible flow tables.

The front-end achieves efficiency by exploiting the dependency implications of composition operators with the specialized data structures. Our front-end further speeds up the compilation by incrementally compiling every rule update. What's more, the front-end compiler is generic to all SDN policy languages that employ policy compositions, and is guaranteed to produce the minimum DAG.

4) We develop efficient back-end scheduling algorithms to map incremental DAG changes to rule updates in TCAM. Our back-end components optimize the rule updates to achieve provably minimum entry moves in TCAM, eliminate redundant rules and provide support for efficient rule caching hierarchy to scale up the size of flow tables.

RuleTris can be deployed in a variety of settings. It can be embedded to a policy compiler, so that minimum updates can be generated even for these incremental-agnostic SDN applications that populate non-minimum rule updates. It can also be built as extensions of SDN controllers or controller hypervisors, so that the policy composition of multiple SDN applications or controllers can be updated with minimum number of operations.

We fully implement RuleTris front-end as a standalone composition compiler, and the back-end in the firmware of data-plane programmable hardware-based ONetSwitch [16], [17]. Through hardware evaluation, we demonstrate that RuleTris achieves a median of <12 ms and 90-percentile of <15 ms the per-rule update latency, outperforming the state-of-the-art composition compiler CoVisor deployed on the same hardware switch by ~ 20x. Our large scale emulation indicates even greater speedup on larger TCAM size.

We give background and related work in Section 2, followed by an overview in Section 3. We describe the front-end design in Section 4, priority value assignment algorithm in Section 5 and back-end design in Section 6. We present our implementation in Section 7, evaluation in Section 8, provide discussions on future topics in Section 9 and conclude in Section 10.

2 Background and Related Work

2.1 Background

1) Rule Updates on Physical Switches

TCAM is the mainstream hardware to implement flow tables in hardware SDN switches. Although TCAM offers incomparable lookup performance, current commercial TCAM solutions are slow on rule update. Measurement studies show that a single rule update can bring tens to hundreds of milliseconds of data plane disruption on state-of-the-art switches [4], [18], since typically conducting updates requires locking TCAM from accepting data plane lookup requests.

Maintaining rule dependency is the main reason to blame for the slow updates of TCAM. In fact, one rule update from the controller can often result in massive TCAM entry moves. This is because TCAM implements rule dependency using the relative physical location [11], [19], i.e., a rule located at a higher physical address has a higher matching priority. Upon the arrival of a new rule, the switch firmware may have to move many existing entries to keep the correct rule dependency. Furthermore, since multiple TCAM entry updates cannot be conducted in parallel, the massive TCAM moves eventually lead to significant rule update latency. The approach RuleTris takes to minimize rule update latency is to eliminate unnecessary TCAM entry moves through maintaining a minimum DAG.

2) Rule Dependency

The predicate of a rule specifies the flow space the rule should match. When two rules have an overlapping predicate, the matching ambiguity needs to be resolved by specifying a matching order. In the context of a flow table, we define the rule dependency as the relation between a pair of rules if their matching order changes the actual rule matching semantics. Without loss of generality, we say Rule *A* is dependent on Rule *B* if Rule *B* should be matched first.

Obviously, the dependency relations form a directed acyclic graph, or DAG [10], [12]. The minimum DAG reveals the inherent relationship among rules in a sense that it represents the minimum set of the matching order constraints in order to keep the correct classification semantics of flow space. In this paper, we use the term DAG to refer specifically to the minimum DAG of a flow table.

In fact, assigning rules with integer priority values is the way OpenFlow employs to unambiguously represent rule dependency. However, rule priority does not directly induce a set of minimum dependency relations in a sense that two rules with different priority values are not necessarily dependent.

3) Modular Composition

Modular composition was widely used in network programming languages and hypervisors to provide transparent composition and collaboration of control plane applications [6], [14], [15], [20]. In this paper, we compose applications with three composition operators: parallel operator, sequential operator, and priority operator. The parallel operator (+) creates the illusion that multiple applications to independently process the same traffic. The sequential operator (>) allows one application to process the traffic before another. The priority operator ($) gives one application the priority to act on a subset of the traffic while yielding the control of the rest to other applications.

A composition compiler is typically used to compile the composition of applications into a semantically equivalent flow table to install on the physical switches. Since applications can act on different header fields, the result flow table usually contains many rules that overlap with each other. All existing composition compilers use priorities to keep the dependency.

2.2 Related Work

1) Modular Composition

Several recent SDN policy languages and controllers (e.g., Frenetic [20], NetCore [21], NetKAT [15], Pyretic [14]) support modular composition. Generally, they take high-level policies and generate flow tables that fulfill the semantics of the sequential and parallel composition.

A recent work proposes CoVisor [6], a controller hypervisor that assigns priority value with a convenient algebra without changing the priority of existing rules. Although CoVisor significantly reduces the number of rule updates, it does not optimize the cost of individual rule updates. Further, CoVisor assumes that the guest controllers are able to produce optimal updates, which is still a challenging problem for the guest controllers. In contrast, RuleTris minimizes both the number of rule updates and the cost of individual updates in TCAM, and it also works with incremental-agnostic applications/controllers.

2) Modular Composition Optimization

Our previous short paper [7] first proposed to preserve rule dependency during compilation. It sketched a solution framework with a compiler-specific dependency preserving algorithm and a heuristic-based priority assignment strategy. RuleTris extends the idea with two fundamental improvements. First, RuleTris proposes a compiler-generic dependency preserving algorithm with incremental compilation capacity in the front-end. Second, the back-end now uses rule dependency to minimize TCAM operations instead of rule priorities, leading to a significant reduction in actual TCAM update time.

3) Incremental TCAM Update

Another related and well-explored topic is incremental TCAM updates. TCAM uses the physical location to encode the priority of entries, with lower addresses (or higher addresses, depending on specific implementation) receiving higher priority [19]. During TCAM incremental update, TCAM controller must maintain a correct order of entries based on the limited knowledge of the entry dependency, which may cause moves of existing entries. Although many algorithms have been proposed to infer entry dependency and reduce the update cost [8], [9], [11], it remains computationally challenging to obtain the minimum dependency graph for a flow table with wildcard matching and multiple matching fields. In contrast, we achieve the update cost minimization through leveraging the minimum dependency information generated in policy composition.

4) Incremental Compilation

Most compilers, except Maple [12], do not support incremental policy compilation. In practice, they simply compile the new policies and replace the entire flow table of each switch. On the other hand, although Maple does not support policy composition, it introduces tree-style abstraction to support incremental flow table compilation. However, Maple compiler still makes redundant priority updates due to the consecutively assigned priority values. RuleTris can be integrated into Maple to provide optimal TCAM updates.

CoVisor [6] assigns priorities that lead to an inefficient usage of priority value space with priority multiply, which in turn limits the number of controllers it can support. Also, the large number of priority levels assigned by CoVisor aggravates to slow rule updates of TCAM. In contrast, RuleTris discards priority values and use the DAG to represent rule dependency.

3 Overview of RuleTris

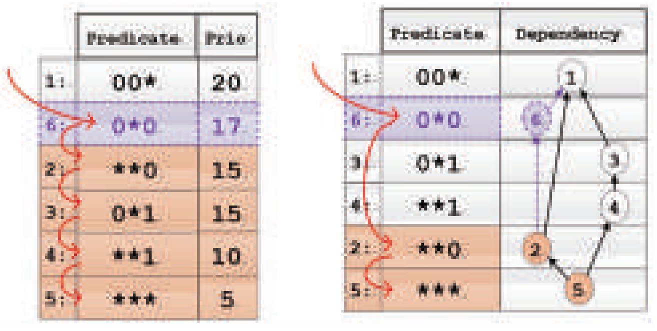
In this section, we first motivate the necessity of the DAG with an example in Section 3.1. We then depict RuleTris optimization framework in Section 3.2, followed by the optimality claims in Section 3.3.

3.1 Benefits of DAG

The key idea for RuleTris to generate minimum update is to represent rule dependency using DAG instead of rule priority until the controller finally compares old and new flow tables. Intuitively, a complete and minimum DAG as the intermediate representation provides the controller the maximum freedom to reuse the priority values of the existing rules, so that the generated updates contain no redundant priority changes.

Generally, optimally updating TCAM tables in physical switches requires a minimum DAG. In implementing a rule update in the TCAM table, integer priority values provide complete dependency information and thus can be used to generate semantically correct update schedule. For example, in **Fig. 2**, Rule 6 is to be added to the flow table. As shown in Fig. 2a, according to the relative priorities, Rule 6 should be placed at a slot with a higher physical address than Rule 2 through Rule 5 and a lower address than Rule 1. Since the only available slot is at the very end, each of Rule 2 through Rule 5 has to be moved one slot down in order to make room for Rule 6.

However, priority values do not guarantee optimality in rule updates. In fact, the integer priority representation implies that all rule pairs with different priority values have dependency, which introduces a huge amount of non-existing dependency



a) Update with priority b) Update with DAG

▲ Figure 2. An example rule insert in a TCAM table. The original TCAM table has five entries (Rules 1-5) and one empty slot in the end. Rule 6 needs to be inserted between Rules 1 and 2. In a), the firmware schedules the insertion plan according to the dependencies implied by the priority values, therefore Rule 2 through Rule 5 are moved in order to preserve their relative positions. In b), however, the DAG indicates the newly inserted Rule 6 has no dependency with Rules 3 and 4, therefore only Rules 2 and 5 need to be moved.

constraints. During the rule update, these redundant dependencies lead to unnecessary TCAM moves.

Instead, the DAG represents a minimum set of dependency constraints and guarantees to produce the optimal update schedule (we will show the optimality in Section 3.3). For example, Fig. 2b shows the optimal update schedule guided by the DAG. Since Rule 6 and Rule 2 has no overlapping flow space with Rule 3 and Rule 4, the optimal update schedule only needs to make two extra entry moves instead of four.

The above example shows the benefit of the DAG in scheduling rule updates. In fact, maintaining the DAG provides a series of other benefits. For example, the DAG makes it straightforward to generate a flow table without rules that are entirely obscured by higher priority rules. By scanning the flow-table in the topological order of the DAG, we can easily eliminate the redundant rules that will never be matched or do not alter the data plane behavior. Also, DAG enables an efficient way to support arbitrarily large flow tables through rule caching [13].

3.2 End-to-End Optimization Framework

The above example shows the importance of the DAG, and leads us to the design of RuleTris optimization framework as in Fig. 1. RuleTris optimization framework is comprised of the front-end composition compiler and the back-end optimizers.

1) Front-End

RuleTris allows administrators to compose multiple controller applications or controllers through composition operators. Such capacity is provided by a general-purpose composition compiler that makes up the RuleTris front-end. The RuleTris composition compiler interfaces with applications or controllers, accepting their proactive or reactive modification of the network policies. Similar with other composition compilers, the RuleTris composition compiler is configured by the administrator to compose the application policies into a single policy implementation for physical network devices. Inspired by previous works, RuleTris allows policy composition with parallel operator (+), sequential operator (>) and priority operator ($) with similar semantics as previous modular composition compilers [6], [7], [14], [20].

Except the compiled flow tables, RuleTris further generates the DAGs to resolve the matching ambiguity, which replaces the integer priority values used in other composition compilers. Upon the arrival of proactive network policy installation, RuleTris compiles the policies in batch, and supplies the back-end with a fresh flow table with the entire DAG. Upon the arrival of reactive policy updates, RuleTris compiles the policy updates in an incremental manner, and supplies the back-end with incremental rule inserts, deletes and modifications together with the updates to the DAG.

RuleTris does not require applications/guest controllers to be dependency-aware. If an application populates prioritized flow tables, RuleTris can extract the DAGs from the prioritized flow tables.

2) Back-End

The RuleTris back-end optimizers exploit the benefits of the DAG and optimize the actual rule installation/update process in the physical switches. For now, RuleTris provides three back-end optimizers. The update scheduler conducts hardware-specific optimization with DAG, and generates minimum-size update schedule to implement rule updates in TCAM tables. The redundancy eliminator removes all the semantically redundant rules. The CacheFlow manager manages multiple-level rule cache structure and conducts rule eviction guided by the DAG [13]. The RuleTris back-end directly generates sequence of TCAM entry moves.

3) Front-End/Back-End Communication

In this paper, we assume the RuleTris back-end is placed in the firmware of physical switches. The front-end to back-end communication is carried through the control channel, e.g., the OpenFlow protocol. RuleTris extends OpenFlow protocol with a DAG extension using the customizable experimenter message, so as to allow the protocol messages to carry DAGs or DAG updates together with flow modification/delete messages. Alternatively, RuleTris back-end can also be co-located with the front-end. In this way, no special front-to-back channel for DAG is necessary but the control channel needs to be extended to expose the TCAM internal layout.

3.3 Optimality Guarantees

RuleTris provides several optimality guarantees with the help of DAG and proper back-end optimizers. We show how the optimality is achieved in Section 6.

Claim 1: With DAG, the back-end can generate a flow table without obscured rules and floating rules.

Through a simple topological scanning, RuleTris can eliminate all the redundant rules generated during modular composition, including the rules obscured by higher priority rules (or obscured rules) and the rule having the same actions with lower priority but more general rules (or floating rules).

Claim 2: With DAG, the back-end can generate the minimum number of entry moves that correctly implements a specific rule update in a TCAM.

This is because the dependency constraint is the only constraint to observe during rule updates in TCAM, and the DAG precisely provides the minimum set of dependency constraints regarding a rule update. The proof is provided in the Appendix.

4 Front-End Compiler

The RuleTris front-end is an incremental composition compiler that compiles forwarding policy updates from SDN applications into rule updates and DAG updates for data-plane flow tables. State-of-the-art incremental compilation technique allows us to compile rule updates with integer priority in a few milliseconds [6]. However, the brute-force way to extract DAG from prioritized flow tables has the high time complexity [7],

[13]. In practice, it can consume minutes in processing a flow table with a few thousand rules.

Alternatively, we choose to maintain the DAG along with the compilation process. The idea was first introduced in our previous short paper [7]. In this section, we extend the NetKAT-specific DAG preservation algorithm into an incremental and compiler-generic front-end by exploiting efficient data structures and algorithms. We first give some background on the modular composition (Section 4.1). Then, we show how we build the DAG along with the composition with linear time complexity (Section 4.2). We present the incremental techniques to further accelerate the compilation of DAG updates (Section 4.3).

4.1 Modular Composition Basics

The ultimate goal of a composition compiler is to combine multiple member policies (or flow tables) into a single result policy. To do so, the existing compilers use the composition configuration (e.g., $(A > B) + C$) to guide the recursive composition compilation. Then, for each composition operator, the compiler combines the two member flow tables (T_1 and T_2) into the result flow table (T_3) according to the semantic of the operator. For parallel and sequential operator, the compiler explicitly iterates over rule pair $(r_{1,i}, r_{2,j}) \in T_1 \times T_2$ in a descending priority order, and calculates the result rule with an operator-specific function $para(r_1, r_2)/seq(r_1, r_2): R \times R \rightarrow R$, where R is the universe set of rules. For parallel operator, the function $para(r_{1,i}, r_{2,j})$ produces a result rule with the match by taking the intersection of $r_{1,i}.match$ and $r_{2,j}.match$ and with the actions by taking the union of $r_{1,i}.actions$ and $r_{2,j}.actions$. For sequential operator, the function $seq(r_{1,i}, r_{2,j})$ produces a result rule with the match by first applying $r_{1,i}.actions$ onto $r_{1,i}.match$ and then intersecting with $r_{2,j}.match$, and with the actions by taking the union of $r_{1,i}.actions$ and $r_{2,j}.actions$. For priority operator, the compiler simply stacks the rules in T_1 on top of T_2 by configuring rules in T_1 with higher priorities than rules in T_2. The reader can refer to previous policy compilers for detailed description of the composition process[2] [15], [21].

4.2 Preserving DAG During Composition

To construct the DAG during the process of a composition operator, the RuleTris compiler needs algorithms to infer the precise dependency relations in the result flow table from the operand DAGs. In addition, we also need efficient data structures to keep the DAGs and the DAG updates.

4.2.1 Parallel Composition

The parallel composition of T_1 and T_2 is calculated by taking cross-product of the operands. Similarly, the DAG of the result flow table is also calculated by taking the equivalent graph cross-product. Denoting two operand graphs as G_1 and G_2, the graph cross-product is defined intuitively as

1) The vertex set of $G_1 \times G_2$ is the set cross-product $V(G_1) \times V(G_2)$;

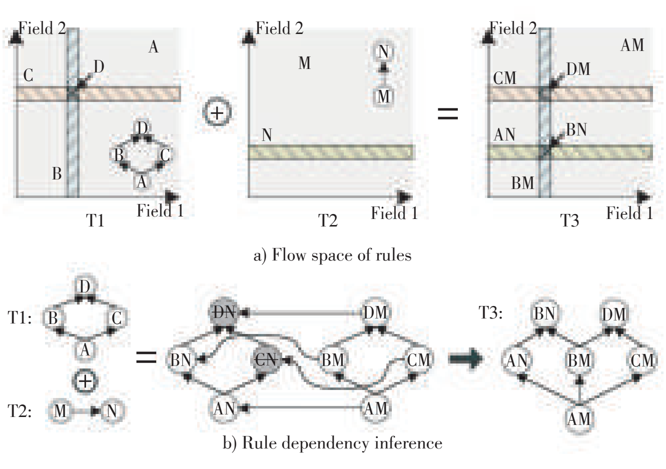
2) There is a directed edge $\langle r_{1,i}, r_{2,m} \rangle \rightarrow \langle r_{1,j}, r_{2,n} \rangle$ in $G_1 \times G_2$ if and only if either i) $r_{1,i} = r_{1,j}$ and $r_{2,m} \rightarrow r_{2,n}$; or ii) $r_{2,m} = r_{2,n}$ and $r_{1,i} \rightarrow r_{1,j}$.

The correctness proof is intuitive. Consider rule r_1 depends on rule r_2, i.e., r_1 overlaps with r_2 and semantically r_2 has a higher priority than r_1. When we intersect both of them with a third rule r, the two result rules $(r_1 \cap r)$ and $(r_2 \cap r)$ still overlap with each other, unless either of them has an empty match.

There are two cases that need special treatment. First, when the parallel composition of any rule pair results in an empty match, the corresponding vertex of this rule should not be added to the result DAG. For example, in **Fig. 3**, we have two flow tables T_1 and T_2 taking the parallel composition. Specifically, T_1 contains four rules (A, B, C, D) and T_2 contains two rules (M, N). In the figure, the match space of the rules is visualized and the actions are omitted. To obtain the result DAG, the compiler first takes a cross-product of the operand DAGs. Then, the compiler crosses out the vertices of all the rules with empty match (DN and CN), and removes their adjacent edges from the DAG as well. Finally, the minimum DAG is obtained as shown on the right.

The second case is when two result vertices are adjacent but the corresponding rules have the same match. In this case, the higher priority rule entirely obscures the other one, so the latter becomes redundant. Although the redundant rules should be maintained within the compiler for the correctness of the future incremental rule removals, it is favorable to eliminate such redundancy in the current output.

We design a two-level nested graph structure to efficiently handle such redundancy. On the higher level, the compiler uses the rule match as the key to index the vertices, which we



a) Flow space of rules

b) Rule dependency inference

▲Figure 3. Example 1 of dependency construction in parallel composition: cross-product and empty rule removal.

[2] We assume all flow tables have a default match-all rule with a pseudo "pass" action, which passes the packet to the next flow table composed with the priority operator or drop the packet if there is not the one.

call key vertices. Therefore, multiple rules with the same match will fall into the same key vertex. If more than one rule is inserted into one key vertex, the dependency relations between those rules are recorded as a nested sub-graph. Within any key vertex, there must exist one single highest priority rule, because otherwise the composed flow table is ambiguous. When the compiler populates the flow table from the DAG, the highest priority rule is used to represent the key vertex, as it obscures all other rules in this key vertex.

Fig. 4 shows an example of the parallel composition of T_1 and T_2. After the cross-product of the operand DAGs, we see several sets of vertices have the same match (e.g., *BN*, *DN* and *DM*). The compiler indexes these equivalent vertex sets with the nested graph data structure, which populates the flow table without redundant matches.

4.2.2 Sequential Composition

In Section 4.1, the existing compilers calculate sequential composition of T_1 and T_2 in a two-level loop. The inner loop is similar to parallel composition. Each rule $r_{1,i}$ in T_1 produces a partial flow table $r_{1,i} > T_2$. For the outer loop, different partial flow tables are stacked by the priorities in T_1. This is because if $r_{1,i}.priority > r_{1,j}.priority$, the partial flow table produced by $r_{1,i}$ will always be matched prior to that by $r_{1,j}$.

The DAG of the sequential composition can be also obtained through a similar two-level loop. For each rule $r_{1,i}$ in T_1, the DAG of the partial flow table $r_{1,i} > T_2$ is calculated by taking a cross-product, similar to the parallel composition. Then, the partial DAGs of the partial flow tables are stitched together according to the dependencies in T_1, i.e., if $r_{1,i} \rightarrow r_{1,j}$ in T_1, the partial DAG induced by $r_{1,i}$ is also dependent on the partial DAG by $r_{1,j}$.

Fig. 5 shows an example of the sequential composition between T_1 and T_2. As shown in the middle of Figure 5b, the partial DAGs in the three large circles are derived from the dependencies of T_2, e.g., $X \rightarrow W$ derives $AX \rightarrow AW$, $BX \rightarrow BW$ and $CX \rightarrow CW$. Meanwhile, the dependencies between partial DAGs are derived from the dependencies of T_1, e.g., $C \rightarrow A$ derives $(CW, CX, CY, CZ) \rightarrow (AW, AX, AY, AZ)$. Finally, after eliminating empty and redundant rules, we get the optimal flow table and its DAG of T_3 shown on the right of Fig. 5b.

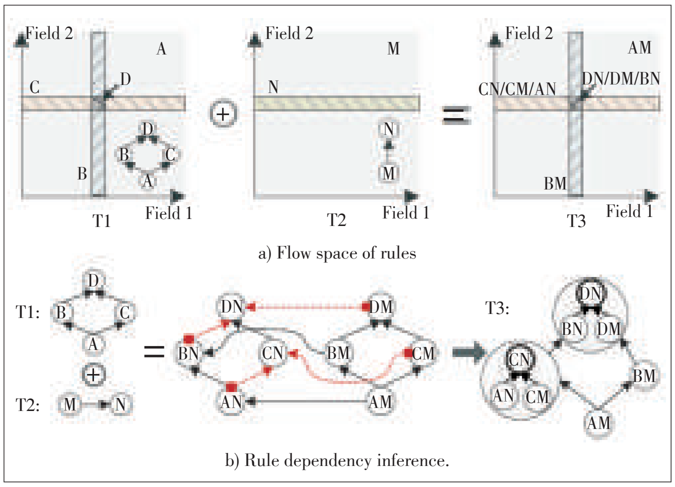
In some cases, the dependency relations between partial DAGs (or “mega” dependencies) need further refinement to produce a minimum set of the dependency relations. More precisely, we can create a mega edge from rule set *A* to rule set *B*, if for every rule pair $<a, b>$ ($a \in A$, $b \in B$) we have either $a \rightarrow b$ or *a* is independent with *b*. We defer the detailed discussion to Section 4.2.3.

4.2.3 Priority Composition

The priority composition of T_1 and T_2 is derived by stacking the flow tables by priority. Therefore, the priority composition of DAGs can be calculated by stitching the operand DAGs with a mega dependency relation from T_2 to T_1.

The challenge comes from resolving the mega dependency between T_1 and T_2 into dependencies between individual rules. Theoretically, the dependency relation between T_1 and T_2 does not necessarily derive the dependency between an arbitrary rule in T_1 and an arbitrary rule in T_2, since they may not overlap with each other. In order to obtain a minimum set of the dependency relations, the compiler needs to efficiently verify any possible rule dependency.

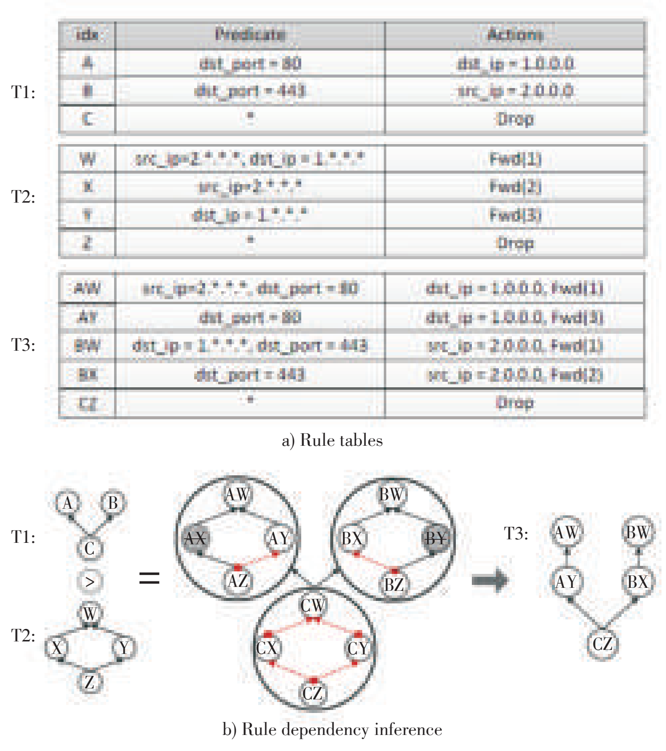
RuleTris compiler resolves the mega dependency relations



a) Flow space of rules

b) Rule dependency inference.

▲Figure 4. Example 2 of dependency construction in parallel composition: equivalent rule reduction.



	idx	Predicate	Actions
T1:	A	dst_port = 80	dst_ip = 1.0.0.0
	B	dst_port = 443	src_ip = 2.0.0.0
	C	*	Drop
T2:	W	src_ip=2.*.*.*, dst_ip = 1.*.*.*	Fwd(1)
	X	src_ip=2.*.*.*	Fwd(2)
	Y	dst_ip = 1.*.*.*	Fwd(3)
	Z	*	Drop
T3:	AW	src_ip=2.*.*.*, dst_port = 80	dst_ip = 1.0.0.0, Fwd(1)
	AY	dst_port = 80	dst_ip = 1.0.0.0, Fwd(3)
	BW	dst_ip = 1.*.*.*, dst_port = 443	src_ip = 2.0.0.0, Fwd(1)
	BX	dst_port = 443	src_ip = 2.0.0.0, Fwd(2)
	CZ	*	Drop

a) Rule tables

b) Rule dependency inference

▲Figure 5. Example of sequential composition.

with the following recursive procedure.

First, the mega dependency from T_2 to T_1 is resolved to a set of tentative dependency relations from every sink vertex of T_2 to every source vertex of T_1, where source vertices (sink vertices) are defined as the vertices that has no incoming (outgoing) edges. For example, in **Fig. 6**, the mega dependency relation is resolved to tentative edges $A \rightarrow Z$ and $B \rightarrow Z$.

Then, for each tentative dependency relation (or edge) $r_2 \rightarrow r_1$, the compiler explicitly checks whether the matches of the two rules r_1 and r_2 overlap. If so, edge $r_2 \rightarrow r_1$ is put into the result DAG. Otherwise, the compiler recursively generates tentative edges as follows.

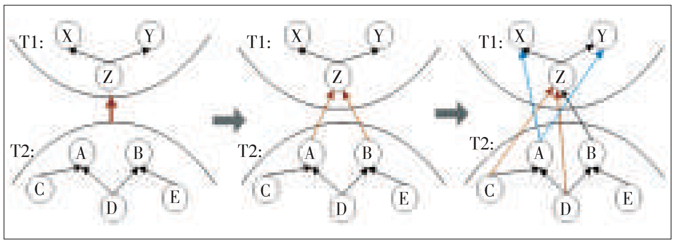
- For every predecessor of r_2, say r_3, if edge $r_3 \rightarrow r_1$ does not exist in the DAG already, the compiler adds it to the set of tentative edges, as r_3 has a more general match than r_2 and may overlap with r_1. For example, in Fig. 6, assuming A and Z do not overlap, the compiler will add $C \rightarrow Z$ and $D \rightarrow Z$ as tentative edges (red dashed edges).
- For every successor of r_1, say r_4, if edge $r_2 \rightarrow r_4$ does not exist already, and meanwhile $r_1.match$ is not strictly more general than $r_4.match$ (meaning $r_1.match - r_4.match$, $\varnothing$ in flow space), the compiler also adds the edge $r_2 \rightarrow r_4$ to the set of tentative edges. This is because r_2 may overlap with r_4 on the excessive flow space $r_1.match - r_4.match$. For example, in Fig. 6, the compiler will also add $A \rightarrow X$ and $A \rightarrow Y$ as tentative edges (blue dashed edges).

In this way, the compiler continues resolving tentative edges until the set of tentative edges is empty.

Finally, **Fig. 7** shows an example of the priority composition between T_1 and T_2. The compiler first adds a mega edge between the DAGs of T_1 and T_2. Then, the mega edge is resolved to a tentative edge from W to C. Because W does not overlap C, this tentative edge sprouts to tentative edges $X \rightarrow C$ and $Y \rightarrow C$. Note, $W \rightarrow A$ is not added as a tentative edge because $A.match$ is strictly smaller than $B.match$. Finally, edge $X \rightarrow C$ is added to the result DAG.

4.3 Incremental Compilation

Ideally, when processing a rule update, the composition compiler should only recompile the rules and the partial DAG that change during the update. We observe that most part of a DAG will not change during a rule update, which indicates the opportunity of dramatic performance improvement over recompilation from scratch.



▲Figure 6. Resolving mega dependency relations.

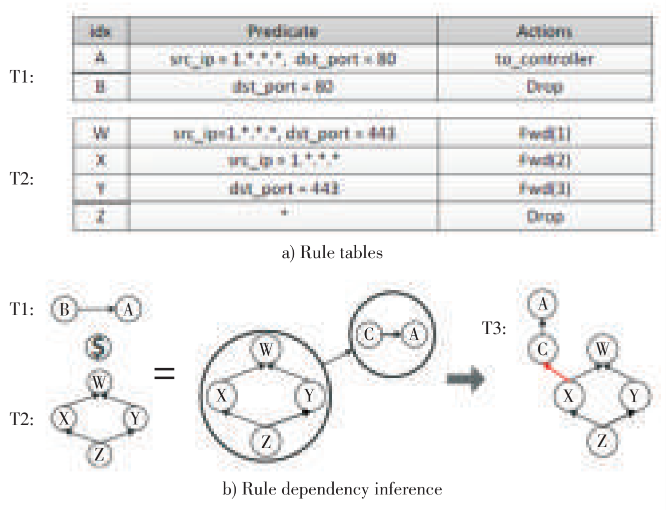
RuleTris' s incremental compilation technique is built on top of existing incremental composition technique. Previous study [6] proposes an efficient indexing structure for flow tables, which allows the compiler to efficiently find the rules that overlap with a target rule. RuleTris employs this technique to avoid redundant computation.

The key technique RuleTris introduces is the mechanism to compile DAG update. Upon the arrival of a rule update with dependency change in the member policy, the RuleTris compiler calculates the delta DAG as follows.

1) Rule insert

Consider a composition of T_1 and T_2. When the compiler receives a rule insert r_1 with the dependency change in T_1, the compiler first computes all the additional rules to be added in result similar to CoVisor. For parallel and sequential composition, it does so by looking up T_2' s index for the rules that overlap with r_1, and apply composition function $para(r_1,r_2)/seq(r_1, r_2)$. For priority composition, r_1 is simply inserted into the result flow table. Then, the compiler calculates the changes in the DAG of T_3. It adds vertices representing the rules inserted into the DAG. Further, the compiler handles dependency changes for the composition operators as follows:

- For parallel composition, the compiler takes a cross-product of the additional partial DAG in T_1 and the full DAG of T_2, and the result partial DAG is added to $T_3.graph$.
- For sequential composition, if r_1 belongs to the left operand (i.e., $T_1 > T_2$), the compiler composes r_1 with T_2 and adds the result partial graph to $T_3.graph$. The compiler also adds the edges associated with r_1 to $T_3.graph$ as mega dependency relations, and resolves them with the same procedure in Section 4.2.2. If r_1 belongs to the right operand (i.e., $T_2 > T_1$), the compiler composes every rule in T_2 with r_1, and adds the result partial graph to $T_3.graph$. The compiler also resolves the mega dependencies in $T_3.graph$, since r_1 may change the actual edges the mega edges are resolved to.



	idx	Predicate	Actions
T1:	A	src_ip = 1.*.*.*, dst_port = 80	to_controller
	B	dst_port = 80	Drop
T2:	W	src_ip=1.*.*.*, dst_port = 443	Fwd(1)
	X	src_ip = 1.*.*.*	Fwd(2)
	Y	dst_port = 443	Fwd(3)
	Z	*	Drop

a) Rule tables

b) Rule dependency inference

▲Figure 7. Example of priority composition.

- For priority composition, the compiler first adds the edges associated with r_1 to $T_3.graph$, and then resolves the mega dependency relation created by the priority operator.

RuleTris further accelerates the above graph compositions with the rule indexing structure. When taking a partial cross-product or sequential composition, the compiler only processes the partial DAG of T_2 whose rules overlap with r_1, because composing r_1 with any rules not overlapping it will result in an empty rule.

2) Rule delete

When a rule is deleted in a member flow table, all the rules that are composed from the deleted rule are to delete in the result flow table. If a deleted rule has both predecessors and successors in the DAG, the compiler will add tentative edges from every rules in the predecessor set to every rules in the successor set. Then, the compiler verifies the tentative edges in the same way as in Section 4.2.3.

3) Rule modification

RuleTris handles rule modification equivalently as one delete plus one insert.

5 Assigning Priority Values

Another challenging step in RuleTris minimum update framework is to assign discrete priority values to the rules in the new flow table. The priority value assignment must observe both the dependency constraint and the integer priority constraint. The objective of this step is to reuse as many rule priorities as possible, so as to minimize the number of priority changes on existing rules. We formulate the optimization problem as follows.

5.1 Problem Formulation

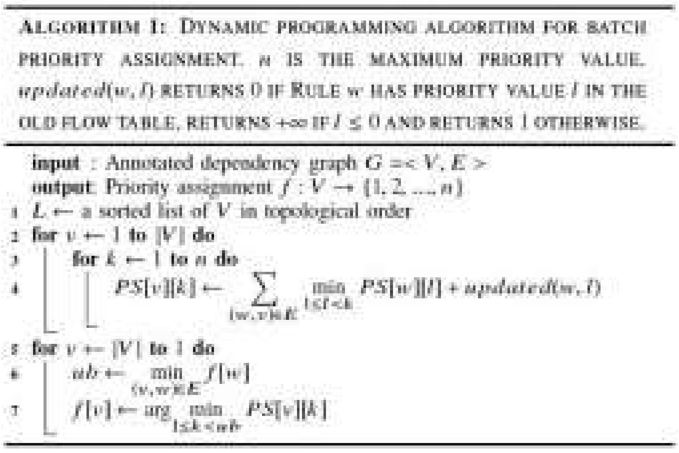
The prioritizer takes as input a directed dependency graph with its vertices representing rules. There are two types of vertices. Some vertices are annotated as retained and each of them is associated with a priority value, which is an integer within a given range. The other vertices are annotated as new and they are not associated with any value. The output of prioritizer is a mapping from the vertex set to the set of priority values that preserves the dependency constraint, i.e., if there is an edge from Vertex A to Vertex B, their priority values must satisfy $pri(A) < pri(B)$.

When we only consider one batch policy update consisting of multiple rule updates, the quality of the output is measured by the number of priority changes, i.e., the number of retained vertices whose priority values are changed. We define batch priority assignment problem as the problem to find the priority assignment with the minimum number of priority changes.

When we consider a sequence of batch policy updates, where upon each update the prioritizer does not know about the future updates, the quality of the output sequence is then measured by the total number of priority updates. We define it as online priority assignment problem, which is an online version of the previous optimization problem.

5.2 Batch Priority Assignment

We solve batch priority assignment problem optimally through dynamic programming. The key idea is to iteratively find the optimal priority assignment for a subset of the new flow table. The algorithm is detailed in **Algorithm 1**.



ALGORITHM 1: DYNAMIC PROGRAMMING ALGORITHM FOR BATCH PRIORITY ASSIGNMENT. n IS THE MAXIMUM PRIORITY VALUE. $updated(w, l)$ RETURNS 0 IF RULE w HAS PRIORITY VALUE l IN THE OLD FLOW TABLE, RETURNS $+\infty$ IF $l \leq 0$ AND RETURNS 1 OTHERWISE.

```
input : Annotated dependency graph $G = < V, E >$
output: Priority assignment $f : V \rightarrow \{1, 2, ..., n\}$
$L \leftarrow$ a sorted list of $V$ in topological order
for $v \leftarrow 1$ to $|V|$ do
    for $k \leftarrow 1$ to $n$ do
        $PS[v][k] \leftarrow \sum_{(w,v) \in E} \min_{1 \leq l < k} PS[w][l] + updated(w, l)$
for $v \leftarrow |V|$ to 1 do
    $ub \leftarrow \min_{(v,w) \in E} f[w]$
    $f[v] \leftarrow \arg\min_{1 \leq k < ub} PS[v][k]$
```

The algorithm traverses the new flow table in the topological order regarding the dependency graph, so that when it visits a vertex, the optimal solution for all its parent vertices have been calculated. $PS[v][k]$ records the minimum number of priority changes of all v's ancestor vertices when v is assigned with priority value k. As it proceeds, the algorithm incrementally explores all possible priority assignments based on previous optimal solutions. Thus, this algorithm guarantees to output a global optimal priority assignment.

5.3 Online Priority Assignment

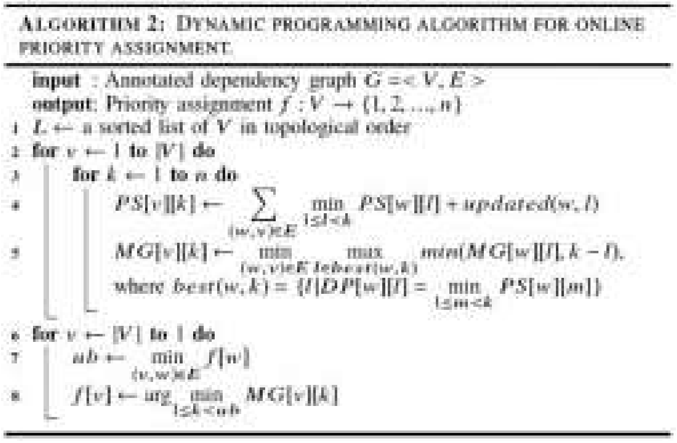
Assigning priority values entails a stochastic process requiring an online strategy: the previous priority assignment decision will become an “existing state” and affect priority assignment of the next policy update. A static optimization solution is impossible due to the uncertainty of the future updates. Instead, we opt for a heuristic approach based on the intuition that a more evenly scattered distribution of priority values reduces the chance of future priority changes.

We formulate the evenness of a priority distribution as the minimum priority gap, i.e., the smallest priority value difference between two adjacent rules. By maximizing the minimum priority gap, we achieve a more “balanced” priority value distribution.

We integrate the heuristic into the previous algorithm by using it to select the most balanced priority assignment among the huge amount of optimal assignments discovered by the previous algorithm. The algorithm is detailed in Algorithm 2. Specifically, $MG[v][k]$ records the evenness indicator of all optimal priority assignments of all v's ancestor vertices with v assigned to priority k.

5.4 Improve Algorithm Speed

Denoting the flow table size as m and the maximum priority number as n, the time complexity of Algorithm 1 and **Algorithm 2** is the state table size $O(mn)$ times the complexity of state transition function $O(n)$. Considering a typical m of thousands and n of 65536, it can take days to calculate an optimal assignment. Therefore, speeding up the algorithm is necessary.



ALGORITHM 2: DYNAMIC PROGRAMMING ALGORITHM FOR ONLINE PRIORITY ASSIGNMENT

```
input  : Annotated dependency graph $G = < V, E >$
output: Priority assignment $f : V \rightarrow \{1, 2, ..., n\}$
$L \leftarrow$ a sorted list of $V$ in topological order
for $v \leftarrow 1$ to $|V|$ do
    for $k \leftarrow 1$ to $n$ do
        $PS[v][k] \leftarrow \sum_{(w,v)\in E} \min_{1\le l<k} PS[w][l] + updated(w, l)$
        $MG[v][k] \leftarrow \min_{(w,v)\in E} \max_{l\in best(w,k)} min(MG[w][l], k - l)$,
        where $best(w, k) = \{l \mid DP[w][l] = \min_{1\le m<k} PS[w][m]\}$
for $v \leftarrow |V|$ to $1$ do
    $ub \leftarrow \min_{(v,w)\in E} f[w]$
    $f[v] \leftarrow \arg\min_{1\le k<ub} MG[v][k]$
```

The key idea to speed up the algorithm is to compress the state table size. Intuitively, considering assigning priority for the highest priority rule in a sub-flow table, the minimum number of priority change ($PS[v][k]$) is always a step function of k, because the function value only reflects the combination of retained/changed state of the ancestor rules. Specifically, we can prove the step function only has no more than $3d$ stages, where d is the length of the longest path of the dependency graph. Therefore, instead of recording n PS values for each vertex, we only need to characterize the step function with less than $3d$ step points of k and the corresponding function values.

As a result, the time complexity of Algorithms 1 and 2 can be reduced to $O(md^2)$. In practice, the optimized algorithms typically calculate the optimal assignments within a few hundred milliseconds for flow tables with thousands of rules.

6 Back-End Optimizer

The DAG and DAG updates generated by RuleTris front-end are exploited by RuleTris back-end to conduct optimization to TCAM updates. RuleTris has three back-end optimizers: update scheduler, duplication eliminator and CacheFlow manager. With them, RuleTris can provide guarantees to conduct rule updates with minimum number of TCAM moves, to compile minimum-size flow tables with no redundant rules and to provide support for efficient rule caching hierarchy.

6.1 Update Scheduler

The update scheduler exploits the DAG to optimize the rule update process in TCAM. When there are entries conflicting with each other on the match patterns, the entry located on the highest physical address wins. As a result, the switch firmware must maintain a correct ordering of rules during TCAM update.

Typically, the switch firmware works as follows. Upon the arrival of a rule insert, the firmware first checks the dependency relations (usually in the form of priority) with the layout of existing rules and looks for the range of locations that satisfy the dependency requirements. Then, it checks if there are empty slots within that range. If so, it picks a slot and writes the new rule in it. Otherwise, the firmware has to move the existing rules for an extra slot.

Integer priority value provides a poor clue of actual rule dependencies, and leads to massive redundant TCAM moves. RuleTris update scheduler exploits the DAG to optimize the TCAM updates. The RuleTris update scheduler first checks if there is an empty slot that satisfies the dependency constraints of the new rule. If so, the new rule is written to the slot. Otherwise, the update scheduler calls **Algorithm 3** to search for an entry moving chain, which starts with the new rule and ends with an empty slot (e.g. $J \rightarrow D \rightarrow A \rightarrow Slot_{top}$ in **Fig. 8**). Finally, the new rule is inserted by moving every rule in the moving chain one slot downstream. The optimality proof of Algorithm 3 is provided in the Appendix.

For example in Fig. 8, Rule J is to be inserted and its relative dependency is shown with the dotted arrows. The scheduler first finds the inserted location range between D and E, which has no slot available. Next, the scheduler looks for the nearest slots, which are located on the top and bottom of the figure. Then, the scheduler searches for moving chains, which are $J \rightarrow D \rightarrow A \rightarrow Slot_{top}$ on the upper side and $J \rightarrow E \rightarrow F \rightarrow Slot_{bottom}$ on the lower side. Since the number of entry moves is the same, a final update decision is picked on a random basis.

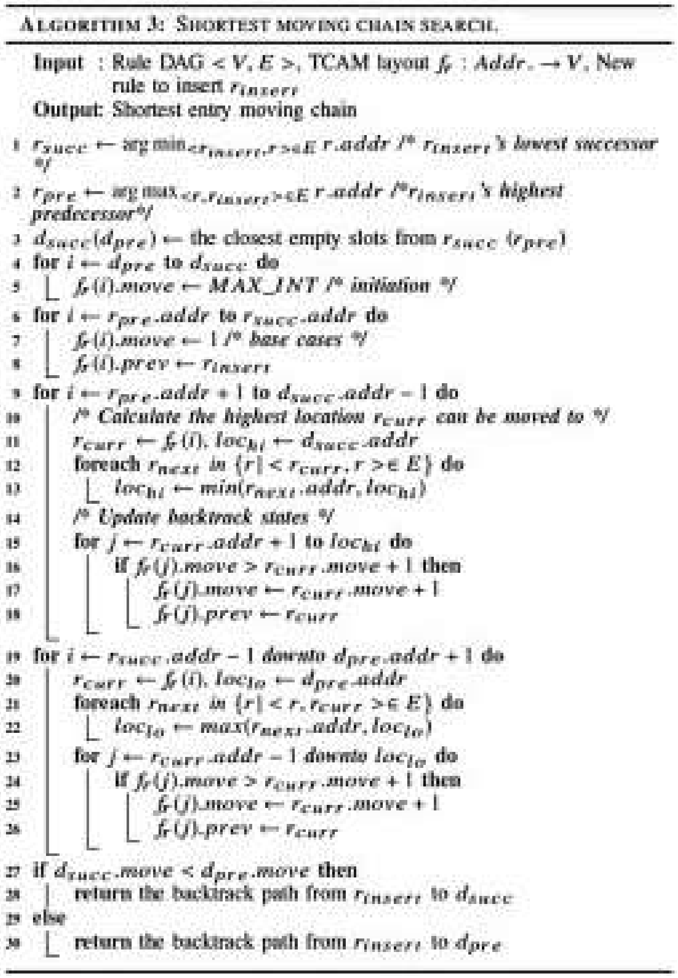
6.2 Redundancy Eliminator

The redundancy eliminator uses the DAG to remove redundant rules. Specifically, we observe two types of redundancy in flow tables:

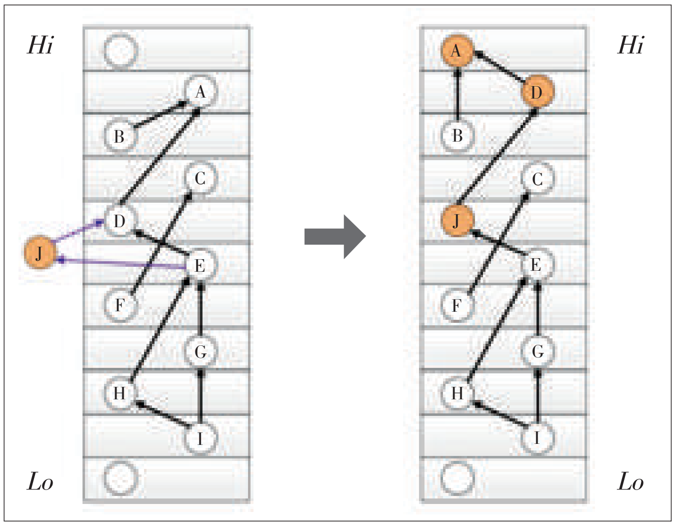
1) Obscured rules. If a rule is entirely obscured by higher priority rules, no data plane packet will match this rule.

2) Floating rules. Consider two rules immediately adjacent in DAG. If they share the same actions and the lower-priority rule has a more general match than the higher-priority one, the higher-priority rule is redundant because removing it does not change the data plane behavior of the flow table.

RuleTris redundancy eliminator conducts one-time scan in a topologically decreasing order to remove the above types of redundant rules. Specifically, for each rule visited, the redundancy eliminator accumulates the match with a flow space union. If a visited rule is entirely obscured by the current accumulated match, it is an obscured rule and should be removed. If a visited rule has the same actions with any of its predecessors and its match is narrower than the predecessor, it is a floating



```
ALGORITHM 3: SHORTEST MOVING CHAIN SEARCH.
Input  : Rule DAG $< V, E >$, TCAM layout $f_F : Addr. \rightarrow V$, New
         rule to insert $r_{insert}$
Output: Shortest entry moving chain
$r_{succ} \leftarrow \arg\min_{<r_{insert}, r> \in E} r.addr$ /* $r_{insert}$'s lowest successor
  */
$r_{pre} \leftarrow \arg\max_{<r, r_{insert}> \in E} r.addr$ /*$r_{insert}$'s highest
 predecessor*/
$d_{succ}(d_{pre}) \leftarrow$ the closest empty slots from $r_{succ}$ $(r_{pre})$
for $i \leftarrow d_{pre}$ to $d_{succ}$ do
    $f_F(i).move \leftarrow MAX\_INT$ /* initiation */
for $i \leftarrow r_{pre}.addr$ to $r_{succ}.addr$ do
    $f_F(i).move \leftarrow 1$ /* base cases */
    $f_F(i).prev \leftarrow r_{insert}$
for $i \leftarrow r_{pre}.addr + 1$ to $d_{succ}.addr - 1$ do
    /* Calculate the highest location $r_{curr}$ can be moved to */
    $r_{curr} \leftarrow f_F(i)$, $loc_{hi} \leftarrow d_{succ}.addr$
    foreach $r_{next}$ in $\{r \mid < r_{curr}, r > \in E\}$ do
        $loc_{hi} \leftarrow min(r_{next}.addr, loc_{hi})$
    /* Update backtrack states */
    for $j \leftarrow r_{curr}.addr + 1$ to $loc_{hi}$ do
        if $f_F(j).move > r_{curr}.move + 1$ then
            $f_F(j).move \leftarrow r_{curr}.move + 1$
            $f_F(j).prev \leftarrow r_{curr}$
for $i \leftarrow r_{succ}.addr - 1$ downto $d_{pre}.addr + 1$ do
    $r_{curr} \leftarrow f_F(i)$, $loc_{lo} \leftarrow d_{pre}.addr$
    foreach $r_{next}$ in $\{r \mid < r, r_{curr} > \in E\}$ do
        $loc_{lo} \leftarrow max(r_{next}.addr, loc_{lo})$
    for $j \leftarrow r_{curr}.addr - 1$ downto $loc_{lo}$ do
        if $f_F(j).move > r_{curr}.move + 1$ then
            $f_F(j).move \leftarrow r_{curr}.move + 1$
            $f_F(j).prev \leftarrow r_{curr}$
if $d_{succ}.move < d_{pre}.move$ then
    return the backtrack path from $r_{insert}$ to $d_{succ}$
else
    return the backtrack path from $r_{insert}$ to $d_{pre}$
```



▲Figure 8. Example of TCAM move optimization.

rule and should be removed.

6.3 CacheFlow Manager

CacheFlow manager maintains a hierarchy of rule caches and helps scale up the size of physical flow tables with larger but slower flow table implementations, such as in SRAM. This technique was proposed in previous work [13]. The key idea is to maintain the correct dependency of the partial flow table in high-speed cache by inserting "cover-set" rules that redirect data plane packets to low-speed matching hardware. We refer the reader to the original paper for details.

7 Implementation

We implement RuleTris front-end composition compiler with 5k lines of Java code. For comparison, we also implement a baseline composition compiler, which recompiles from scratch for each update, and the CoVisor composition compiler [6], which does efficient incremental composition using the priority algebra.

We implement RuleTris back-end optimizers by extending the ONetSwitch firmware with 3k lines of C code [16]. ONetSwitch is hardware based all programmable SDN switch which allows us to fully amend the firmware for RuleTris. We extend OpenFlow v1.3 protocol with DAG support using experimenter messages. The extension can carry both full DAGs and incremental DAG updates from the front-end to the firmware back-end. In the experiments, RuleTris composition compiler uses the extended OpenFlow to talk to RuleTris back-end firmware, while the baseline compiler and the CoVisor compiler use the original ONetSwitch firmware with full OpenFlow v1.3 support.

8 Evaluation

8.1 Methodology

a) Experiment Setup

We evaluate RuleTris under three scenarios. The first two evaluate the rule update efficiency of RuleTris with parallel and sequential compositions. The third one evaluates the rule swapping efficiency with the CacheFlow back-end. In each scenario, we conduct hardware experiments using aforementioned ONetSwitch with a 256 entry TCAM flow table, and stress RuleTris with larger flow table updates through firmware emulation. Except as otherwise noted, we maintain a reasonably high TCAM load factor of 0.90 in the emulation experiments.

We run all composition compilers on top of Ryu controller [22]. The front-end compilation and the back-end emulation are done on a Linux workstation with 4 cores at 2.8 GHz and 8 GB memory.

In the experiments, we compare RuleTris with the following composition compilers.

- Baseline. The baseline compiler recompiles the new flow table from scratch for every rule update and assigns sequential priority values to the new flow table.
- CoVisor. The CoVisor compiler conducts incremental compilation to rule updates with the efficient rule indexing struc-

ture. It assigns priority to new rules using a convenient algebra that prevents reprioritizing.

b) Dataset

- L3-L4 monitoring + L3 router. In this scenario, the L3-L4 monitoring app collects flow statistics in parallel with a L3 router conducting IP-based forwarding. We generate monitoring rules using network filter generation tool ClassBench [23] with the firewall configuration. L3 router rules are also generated using ClassBench, but with the IP chain configuration.
- L3-L4 NAT > L3 router. L3 router rules are generated similar as above. L3-L4 network address translation (NAT) tables are randomly generated based on the IP addresses and TCP/User Datagram Protocol (UDP) ports of the router rules.
- CacheFlow rule swapping. CacheFlow picks a subset of rules from a full rule set to put in cache. In our experiment, the full rule set is a forwarding rule database with 1000 rules generated similar as previous L3 router rules. A set of rules is randomly selected to be installed in the TCAM, as well as the necessary cover-set rules that ensure correct matching semantics. Then, a sequence of swap-in/swapout operations is randomly generated to mimic the cache swapping behavior.

c) Metrics

In Figs. 9, 10, and 11 the bars show the median, and the error bars show the 10th and 90th percentiles.

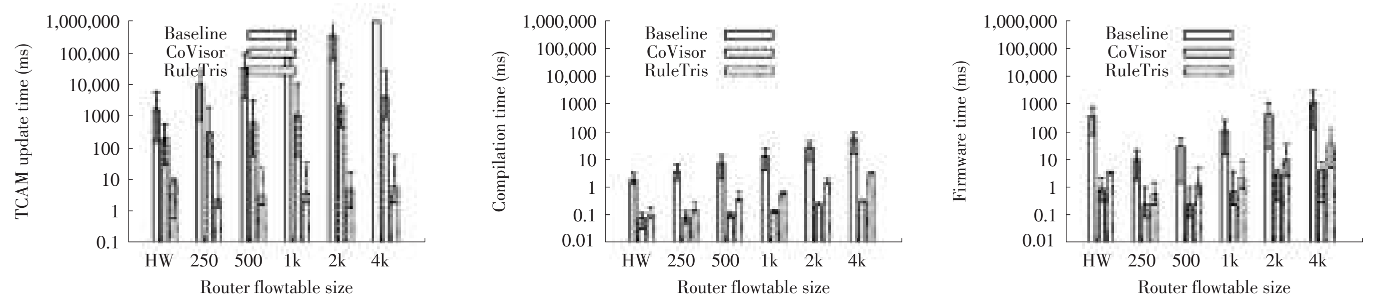
- Compilation time. It is the computation time for compiling the rule update in the front-end.
- Firmware time. It is the computation time for calculating the update schedule from a priority-based or dependency graph-based rule update in the switch firmware. In hardware experiments, this time is measured on the 800 MHz ARM Cortex-A9 CPU on ONetSwitch by switch firmware. In the emulation experiments, the firmware time is measured on the workstation emulating the physical switch.
- TCAM update time. It is the actual time to conduct rule updates on the TCAM. Since TCAM moves are conducted sequentially and each TCAM move costs a fairly constant amount of time, we use the total number of moves times the average latency of a TCAM move (0.6 ms) to estimate the TCAM update time in emulation experiments.

8.2 Experimental Results

Fig. 9 shows the results of L3-L4 monitoring + L3 router. In this experiment, we initiate L3-L4 monitoring table with 100 rules and L3 router with 250 to 4k rules to show how the overhead increases. We sequentially feed 1000 updates to compilers, each update contains one rule delete and one rule insert to the L3-L4 monitoring table. The size of L3 routers is set to 78 in the hardware experiment (first group) in order to fit the 256-entry TCAM.

The TCAM update time, compilation time and firmware time are shown in Figs. 9a, 9b and 9c respectively. The baseline compiler is by far the slowest regarding all three metrics. This is because it recompiles the flow table in every round with new priority value assignments, and thus generates a large amount of redundant rule updates that only modifies the rule priority. In the hardware experiment, RuleTris exhibits 20x faster total update time than CoVisor adding all three latency components together. And emulations indicate even greater differences. Among three latency components, TCAM update time contributes the most. RuleTris has the fastest and a fairly constant latency in TCAM updates. This is because RuleTris maintains the DAG that helps the firmware to calculate the optimal update schedule. Since CoVisor does not keep DAG, it is the fastest in compilation and firmware time, but spends 1 to 3 orders of magnitude more time on TCAM update. Note, the hardware experiment shows a higher firmware time than emulations because of the different capacity of the processors.

Fig. 10 shows the result of L3-L4 NAT > L3 router. Same as the previous experiment, we initiate L3-L4 NAT table with 100 rules and L3 router with 250 to 4k rules to show how the overhead increases. We sequentially feed 1000 updates to compilers, each update contains one rule removed from and one rule inserted to the NAT table. The size of L3 routers is set to 126 in the hardware experiment. Again, we observe RuleTris exhibits about 20x faster total update time than CoVisor due to the time saved in the TCAM updates.



a) TCAM update time

b) Compilation time

c) Firmware time

HW: hardware TCAM: ternary content-addressable memory

▲**Figure 9. Rule update overhead of L3-L4 monitoring + L3 router. The first group (HW) is hardware experiment results and the rest are emulation results.**

Fig. 11 shows the result of CacheFlow rule swapping. In this experiment, we create a two-level CacheFlow with the physical switch as the first level. We vary the load factor of the first level from 0.8 to 1.0. We compare the rule swapping efficiency of RuleTris with the priority-based update firmware. We initiate the CacheFlow manager with a thousand L3 forwarding rules. We randomly select 205 to 256 rules (according to the load factor) to install into the first level. We sequentially feed 1000 updates to the CacheFlow manager; each update contains one rule delete and one rule insert to the TCAM table.

The TCAM update time and firmware time are shown in Figs. 11a and 11b respectively. As expected, RuleTris's DAG based updates show a dominant advantage over the priority-based updates. The median of RuleTris TCAM update time ranges from 0.6 to 1.2 milliseconds, whose bars can be barely seen in the figure. In contrast, priority-based updates costs 40 to 100 milliseconds per rule swapping, and the per-operation cost increases significantly with the TCAM load factor. The long tail of the RuleTris update time is due to some of the swap-in rules that have dense dependency with the rules in cache, which leads to multiple entry moves in TCAM.

9 Discussion

1) Multiple Tables

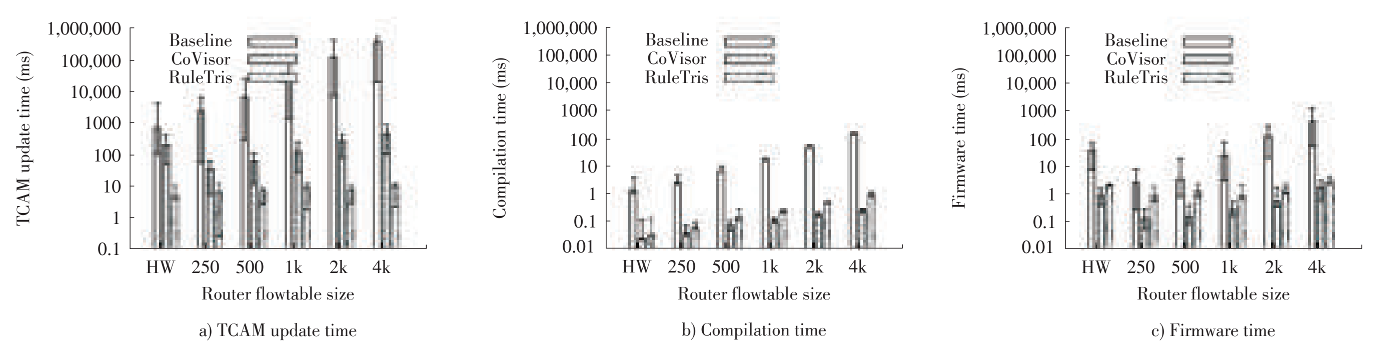
RuleTris currently optimizes updates to a single flow table. Switches typically have multiple tables. Depending on the order of execution of the tables, we can further minimize the rule updates. For example, if we have two TCAM tables in a pipeline, the dependencies between the two modules in a sequential composition can be decoupled by placing the first one in the first TCAM and the second module in the second TCAM. Similarly, if we have two TCAM tables that operate in parallel and the actions are both applied, we can decouple the dependencies of the two modules in a parallel composition. However, the number of tables in a hardware switch is limited. RuleTris can support more module compositions than the number of physical flow tables. We leave the effective distribution of rules to multiple flow tables to our future work.

2) Hardware Specific Optimizations

Tango [24] and Mazu [18] have shown that different switches can have very different latency behavior depending on the order of rule updates. For example, given two ordering of a batch of rules, one is in increasing priority and the other in decreasing priority. One switch has a much lower latency for the first order. Techniques [18], [24] proposed to exploit hardware behavior can be usefully combined with RuleTris.

3) Minimal Network Update

RuleTris considers per switch flow table updates independently. Coordination among flow tables of several switches can



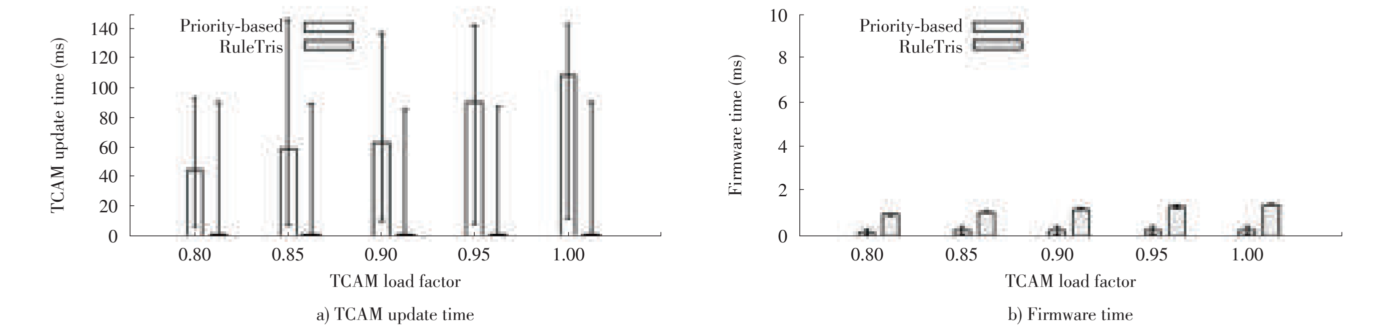
a) TCAM update time

b) Compilation time

c) Firmware time

HW: hardware TCAM: ternary content-addressable memory

▲Figure 10. Rule update overhead of L3-L4 network address translation (NAT) > L3 router.



a) TCAM update time

b) Firmware time

TCAM: ternary content-addressable memory

▲Figure 11. Rule update overhead of single rule swaps with CacheFlow. Results are from hardware experiments.

be combined with RuleTris to further reduce the number of updates [5], [25]–[28].

For future work, we would like to release the source code of RuleTris and build RuleTris into controller platforms such as OpenDaylight and Open Network Operating System (ONOS), and hypervisors such as OpenVritex. We would also like to exploit the benefits of multiple flow tables and the gains across switches.

10 Conclusions

To enable effective modular programming in software defined networks, it is crucial that modular composition be optimized end-to-end for both compilation time and switch update time and flow table size. We present the first end-to-end optimization framework, RuleTris that incrementally keeps DAG during policy compilation and exploits DAG for optimal TCAM updates. We fully implement RuleTris and demonstrate its optimality with both hardware experiments and emulations.

Appendix

We first introduce the definition of a valid entry moving chain. We define an entry moving chain as valid if after inserting the new rule by moving entries along the entry moving chain, all the entries still satisfy the dependency constraints indicated by the DAG.

Now we prove Algorithm 3 generates one of the shortest valid entry moving chains.

Theorem 1. Algorithm 3 generates a valid entry moving chain.

Proof: There are two types of entry moves in the generated entry moving chain.

First, the new rule r_{insert} is written into the location of an existing rule (the loop between Line 6 and Line 8). Considering the range of the loop variable, the possible destination location of r_{insert} is from the location of r_{insert} ' s highest predecessor r_{pre} and the location of r_{insert} ' s lowest successor r_{succ} . If the destination location is between r_{pre} and r_{succ} exclusively, it is obviously that the destination location of r_{insert} is higher than all its predecessors and lower than all its successors, thus the dependency holds. If the destination location is at r_{pre} (r_{succ}), the following moving chain searching code at Line 26 (Line 18) determines that r_{pre} (r_{succ}) is moved to the a lower (higher) location. Therefore the dependency holds.

Second, some existing rules are moved from the previous location to a new location (the two loops between Line 9 to Line 23). Without loss of generality, we consider the downstream search loop (Line 9 to Line 15), which searches for the shortest downstream moving chain. Specifically, Line 15 determines that the possible destination location of an existing rule r_{curr} is between $r_{curr.}addr+1$ and r_{curr} ' s lowest successor ' s location. Therefore, the new location of r_{curr} is still lower than all its successors.

Theorem 2. Given the input rule DAG is minimum, no other valid entry moving chain has fewer entries than the entry moving chain generated by Algorithm 3.

Proof: Without loss of generality, we still only consider the downstream search. We prove the theorem by induction on the loop variable i of the loop from Line 9 to Line 15.

Base case: When $i=r_{pre}.addr+1$, the length of the moving chain is one (set by at loop from Line 6 to Line 8), and it is the minimum possible length of a valid entry moving chain.

Induction: Assuming for all $i=r_{pre}.addr+1$ to i_{curr-1}, the shortest entry moving chains are known, i.e., $f_r(i).move$ and $f_r(i).prev$ store the correct length of its shortest entry moving chain.

Consider $i=i_{curr}$. Assuming the previous entry on the shortest entry moving chain is i_{last}, the index of the lowest successor of $f_r(i_{last})$ must be larger or equal to i_{curr} , because otherwise moving $f_r(i_{last})$ to i_{curr} would introduce a DAG edge inversion between $f_r(i_{last})$ and its lowest successor. Since the DAG is minimum, a DAG edge inversion is necessarily a dependency violation.

The loop between Line 15 to Line 18 guarantees that if $f_r(i_{last})$ is larger or equal to i_{curr} , $f_r(i_{last})$ must have been considered to move to i_{curr} , therefore we have

$$f_r(i_{curr}).move \leqslant f_r(i_{last}).move+1. \tag{1}$$

On the other hand, for all i'_{last} that has $f_r(i'_{last}).move=f_r(i_{last}).move-1$, the index of the lowest successor of $f_r(i'_{last})$ must be smaller than i_{curr}, because otherwise moving $f_r(i'_{last})$ to i_{curr} would be valid and $f_{\mathrm{r}}(i_{\mathrm{last}})$ would not be the previous entry of $f_{\mathrm{r}}(i_{\mathrm{last}})$ on the shortest entry moving chain, which contradicts with the assumption. The loop between Line 15 to Line 18 guarantees $f_r(i'_{last})$ will not be considered to move to i_{curr}, therefore we have

$$f_r(i_{curr}).move>f_r(i'_{last}).move+1=f_r(i_{last}). \tag{2}$$

Since all the values are integers, we can combine 1 with 3 and have

$$f_r(i_{curr}).move=f_r(i_{last}).move+1, \tag{3}$$

which is the correct length of $f_r(i_{curr})$ ' s shortest moving chain.

References

[1] *Requirements of an MPLS Transport Profile*, IETF RFC 5654, 2009.

[2] M. Al - Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic flow scheduling for data center networks." in *7th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, San Jose, USA, 2010, pp. 19–19.

[3] ONF. (2013, Oct. 8). *Solution brief: SDN security considerations in the data center* [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/solution-briefs/sb-security-data-center.pdf

[4] M. Kuzniar, P. Perešíni, and D. Kostic, "What you need to know about SDN flow tables," in *International Conference on Passive and Active Measurement*, New York, USA, 2015, pp. 347–359. doi: 10.1007/978-3-319-15509-8_26.

[5] X. Jin, H. H. Liu, R. Gandhi, et al., "Dynamic scheduling of network updates," in *ACM Conference on SIGCOMM*, Chicago, USA, 2014, pp. 539–550. doi: 10.1145/2619239.2626307.

[6] X. Jin, J. Gossels, J. Rexford, and D. Walker, "CoVisor: a compositional hypervisor for software-defined networks," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI' 15)*, Oakland, USA, 2015, pp. 87–101.

[7] X. T. Wen, C. X. Diao, X. Zhao, et al., "Compiling minimum incremental update for modular SDN languages," in *Third Workshop on Hot Topics in Software Defined Networking (HotSDN)*, Chicago, USA, 2014. doi: 10.1145/2620728.2620733.

[8] J. Van Lunteren and T. Engbersen, "Fast and scalable packet classification," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 4, pp. 560–571, May 2003. doi: 10.1109/JSAC.2003.810527.

[9] T. Mishra and S. Sahni, "DUO-dual TCAM architecture for routing tables with incremental update," in *IEEE International Symposium on Computers and Communications (ISCC)*, Riccione, Italy, 2010, pp. 503–508. doi: 10.1109/ISCC.2010.5546713.

[10] H. Y. Song and J. Turner, "Fast filter updates for packet classification using TCAM," in *IEEE GLOBECOM*, San Francisco, USA, 2006. doi: 10.1109/GLOCOM.2006.342

[11] D. Shah and P. Gupta, "Fast updating algorithms for TCAMs," *IEEE Micro*, vol. 21, no. 1, pp. 36–47, Jan. 2001. doi: 10.1109/40.903060.

[12] A. Voellmy, J. Wang *et al.*, "Maple: simplifying SDN programming using algorithmic policies," in *ACM SIGCOMM*, Hong Kong, China, 2013, pp. 87–98. doi: 10.1145/2534169.2486030.

[13] N. Katta, O. Alipourfard, J. Rexford, and D. Walker, "Infinite cacheflow in software-defined networks," in *Third Workshop on Hot Topics in Software Defined Networking (HotSDN)*, Chicago, USA, 2014, pp. 175–180. doi: 10.1145/2620728.2620734.

[14] J. Reich, C. Monsanto, N. Foster, J. Rexford, and D. Walker. (2013). *Composing software defined networks* [Online]. Available: http://frenetic-lang.org/publications/composing-nsdi13.pdf

[15] C. J. Anderson, N. Foster, A. Guha, *et al.*, "NetKAT: semantic foundations for networks," in *41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL' 14)*, San Diego, USA, 2014. doi: 10.1145/2535838.2535862.

[16] C. C. Hu, J. Yang, H. B. Zhao, and J. H. Lu. (2014). Design of all programmable innovation platform for software defined networking [Online]. Available: https://www.usenix.org/system/files/conference/ons2014/ons2014-paper-hu_chengchen.pdf

[17] ONetSwitch. (2018). *ONetSwitch45* [Online]. Available: http://onetswitch.org/hardware45.html

[18] K. He, J. Khalid, S. Das, et al., "Mazu: taming latency in software defined networks," University of Wisconsin-Madison, Tech. Rep., 2014.

[19] K. Pagiamtzis and A. Sheikholeslami, "Content-Addressable Memory Circuits and Architectures: A Tutorial and Survey," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, pp. 712–727, Mar. 2006. doi: 10.1109/JSSC.2005.864128.

[20] N. Foster, R. Harrison, M. J. Freedman, et al., "Frenetic: a network programming language," in *16th ACM SIGPLAN international conference on Functional programming*, Tokyo, Japan, 2011, pp. 279–291. doi: 10.1145/2034773.2034812.

[21] C. Monsanto, N. Foster, R. Harrison, and D. Walker, "A compiler and run-time system for network programming languages," in *39th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, Philadelphia, USA, 2012, pp. 217–230. doi: 10.1145/2103656.2103685.

[22] Ryu SDN Framework Community. (2015, Jan. 29). *Ryu OpenFlow controller* [Online]. Available: https://osrg.github.io/ryu/index.html

[23] D. E. Taylor and J. S. Turner, "ClassBench: A Packet Classification Benchmark," *IEEE/ACM Transactions on Networking*, vol. 15, no. 3, pp. 499–511, Jun. 2007. doi: 10.1109/TNET.2007.893156.

[24] A. Lazaris, D. Tahara *et al.*, "Tango: simplifying SDN programming with automatic switch behavior inference, abstraction, and optimization," in *ACM International on Conference on Emerging Networking Experiments and Technologies (CoNext)*, Sydney, Australia, 2014, pp. 199–211. doi: 10.1145/2674005.2675011.

[25] C.-Y. Hong, S. Kandula, R. Mahajan, et al., "Achieving high utilization with software-driven WAN," in *ACM SIGCOMM*, Hong Kong, China, 2013, pp. 15–26. doi: 10.1145/2486001.2486012.

[26] H. H. Liu, X. Wu, M. Zhang, et al., "zUpdate: updating data center networks with zero loss," *ACM SIGCOMM*, Hong Kong, China, 2013. doi: 10.1145/2534169.2486005.

[27] M. Reitblatt, N. Foster, J. Rexford, C. Schlesinger, and D. Walker, "Abstractions for network update," in *ACM SIGCOMM*, Helsinki, Finland, 2012. doi: 10.1145/2342356.2342427.

[28] N. P. Katta, J. Rexford *et al.*, "Incremental consistent updates," in *Second Workshop on Hot Topics in Software Defined Networking (HotSDN)*, Hong Kong, China, 2013, pp. 49–54. doi: 10.1145/2491185.2491191.

Manuscript received: 2018-07-17

Biographies

CHEN Yan (ychen@northwestern.edu) received the Ph.D. degree in computer science from the University of California at Berkeley, USA, in 2003. He is currently a professor with the Department of Electrical Engineering and Computer Science, Northwestern University, USA and a distinguished professor with the College of Computer Science and Technology, Zhejiang University, China. Based on Google Scholar, his papers have been cited over 10,000 times and his h-index is 49. His research interests include network security, measurement, and diagnosis for large-scale networks and distributed systems. He received the Department of Energy Early CAREER Award in 2005, the Department of Defense Young Investigator Award in 2007, the Best Paper nomination in ACM SIGCOMM 2010, and the Most Influential Paper Award in ASPLOS 2018.

WEN Xitao (xitao.wen@gmail.com) received the B.S. degree in computer science from Peking University, China, in 2010, and the Ph.D. degree in computer science from Northwestern University, USA, in 2016. His research interests span the area of networking and security in networked systems, with a current focus on software-defined network security and data center networks.

LENG Xue (lengxue_2015@outlook.com) received the B.S. degree in computer science and technology from Harbin Engineering University, China, in 2015. She is currently pursuing the Ph.D. degree major in computer science and technology with Zhejiang University, China. Her research interests are software-defined networking (SDN), network function virtualization (NFV), microservice and 5G protocol verification. She is a student member of the IEEE and CCF.

YANG Bo (ybo2013@zju.edu.cn) received the B.S. degree in information security from the Huazhong University of Science and Technology, China, in 2013, and the M.S. degree in computer science from Zhejiang University, China, in 2016. He is currently a software engineer with Microsoft, Shanghai, China. His research interests include software-defined network and network security.

Li Erran Li (lierranli@gmail.com) received the Ph.D. degree in computer science from Cornell University, USA. He was a researcher with Bell Labs. He is currently with Uber and also an adjunct professor with the Computer Science Department, Columbia University, USA. His research interests are in machine learning algorithms, artificial intelligence, and systems and wireless networking. He is an ACM Distinguished Scientist. He was an associate editor of the IEEE Transactions on Networking and the IEEE Transactions on Mobile Computing. He co-founded several workshops in the areas of machine learning for intelligent transportation systems, big data, software defined networking, cellular networks, mobile computing, and security.

ZHENG Peng (zeepean@gmail.com) received the B.S. degree in information security from Northwestern Polytechnical University, Xi'an, China, in 2015. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Xi'an Jiaotong University, China. He was a visiting research fellow at Duke University, USA from July to August 2017 and Brown University from July to October 2018, respectively. He has authored papers in CoNEXT, ICDCS, ICNP, etc. His research interests span the area of computer networking and systems, with a focus on the programmable network and software-defined networking.

HU Chengchen (chengchenhu@gmail.com) received the B.S. degree from the Department of Automation, North-Western Polytechnical University, China, and the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, China, in 2003 and 2008, respectively. He worked as an assistant research professor with Tsinghua University from July 2008 to December 2010. After that, he joined the Department of Computer Science and Technology, Xi'an Jiaotong University, China, where he is currently a full professor. His main research interests include computer networking systems and network measurement and monitoring.

DOI: 10.19729/j.cnki.1673-5188.2018.04.005
http://kns.cnki.net/kcms/detail/34.1294.TN.20181112.1001.002.html, published online November 12, 2018

A New Direct Anonymous Attestation Scheme for Trusted NFV System

CHEN Liquan[1], **ZHU Zheng**[1], **WANG Yansong**[2], **LU Hua**[2], **and CHEN Yang**[1]
(1. School of Information Science and Engineering, Southeast University, Nanjing 210096, China;
2. Nanjing R&D Center, ZTE Corporation, Nanjing 210021, China)

Abstract

How to build a secure architecture for network function virtualization (NFV) is an important issue. Trusted computing has the ability to provide security for NFV and it is called trusted NFV system. In this paper, we propose a new NFV direct anonymous attestation (NFV-DAA) scheme based on trusted NFV architecture. It is based on the Elliptic curve cryptography and transfers the computation of variable D from the trusted platform module (TPM) to the issuer. With the mutual authentication mechanism that those existing DAA schemes do not have and an efficient batch proof and verification scheme, the performance of trusted NFV system is optimized. The proposed NFV-DAA scheme was proved to have a higher security level and higher efficiency than those existing DAA schemes. We have reduced the computation load in Join protocol from $3G_1$ to $2G_1$ exponential operation, while the time of NFV-DAA scheme's Sign protocol is reduced up to 49%.

Keywords

NFV; trusted computation; DAA; bilinear pairings

1 Introduction

Network function virtualization (NFV) is a new network architecture based on standard virtualization technology. It can realize the network entities such as servers, switches and storages on industrial standard hardware platforms, achieving various network functions by running different software on the virtualized platform. Network entities loaded onto the virtualized platform can achieve dynamic resource allocation, and network flexibility and scalability enhancement. Moreover, replacing the existing special hardware devices with industrial standardized servers could decrease the operators' network cost. Low cost and high flexibility are the great features of NFV technology [1].

Since NFV technology is being widely used in the foreseen future, the security issues in NFV system need resolving. In an NFV system, the virtualized network function (VNF) is the key model of achieving network functionality and it possibly becomes the first target to be attacked. Furthermore, compared to other modules in NFV, VNF needs to interact with outside environments (e.g. another VNF) frequently, which makes it to be another breakthrough for the malicious counterpart. Therefore, the VNF security is the most important part for the entire NFV system. To guarantee the security of interaction among different VNF modules, a two-way authentication protocol for mutual authentication is necessary. At the same time, a security channel needs to be established for this authentication protocol, which prevents the conversation between VNF parties from eavesdropping. Based on direct anonymous attestation (DAA), we propose a new NFV-DAA scheme that is applied to the authentication between VNF modules. The proposed scheme also provides VNF with identification and mutual authentication, and establishes secure communication channel between the VNF parties.

The DAA scheme was first developed by Brickell, Camenisch, and Chen [2] for remote authentication of a trusted computing platform while preserving the privacy of the platform. It has been adopted by trusted computing group (TCG) in the trusted platform module (TPM) specification version 1.2 [3]. DAA is a new group signature scheme without the capability to open signature but with a mechanism to detect rogue members. It draws on the techniques that have been developed for group signatures, identity escrow and credential systems. In the DAA scheme, a suitable signature scheme is employed to issue certificate on a membership public key generated by a TPM. This certificate can help one platform to be authenticated as a group

This work was supported by the national Natural Science Foundation of China (NSFC) under grant No. 61372103 and the ZTE Industry-Academia-Research Cooperation Funds.

member. A valid TPM proves to the verifier that it possesses a certificate. Each TPM has a secret key, which is used to sign a credential and detect rogue TPMs by the verifier. Many researchers have proposed different DAA schemes to meet the requirements in different applications and environments [4]–[6].

Generally, the DAA schemes have the characteristics of efficiency, anonymity, and privacy.

The Join protocol, by which the issuer receives the TPM's application for joining and sends back the credential to TPM, runs once a new platform with TPM begins to join this trusted system. When this platform receives the DAA credential from the issuer, it can use this credential to conduct the following sign/verify processes many times. Compared to the privacy certificate authority (CA) scheme, the issuer in DAA has no need to conduct in each of the following sign/verify processes [2]. Therefore, DAA is more efficient than the Privacy CA scheme that have a bottleneck because the Privacy CA server has to be included in every processing section.

Since the DAA scheme uses zero-knowledge proof theory to prove the trust of a new platform which possesses legitimate credential, it prevents any adversary from seeking the identity of the real communicating TPM. Meanwhile, many DAA schemes use the credential randomization technique to mask the real transmitted credential [7], [8]. It is difficult for the adversary to track the identity of the target TPM even when the verifier can collude with the credential issuer.

The trusted credential issuer has endorsement key (EK) lists to check the legitimation of the applying TPM, and the verifier employs the Camenisch-Lysyanskaya (C-L) signature scheme [9] and the discrete logarithms based proofs to prove the possession of a certificate, while the unforgeability, privacy and anonymity are guaranteed under the decisional Diffie-Hellman (DDH) assumption.

The DAA scheme in [2] is based on the strong RSA assumption and is named as RSA-DAA. Theory analysis results have shown that the protocols and algorithms in RSA-DAA are complicated and inefficient. In recent years, researchers have worked on how to create new DAA schemes with elliptic curves cryptography (ECC) and bilinear pairings [10], [11]. We call these DAA schemes as ECC-DAA for short. Generally speaking, ECC-DAA is more efficient in both computation and communication than RSA-DAA. The operation of TPM is much simpler and the key and signature length is shorter in ECC-DAA than that in RSA-DAA.

However, there are no existing DAA schemes proposed to meet the requirements of NFV system by now. According to the security requirements of mutual authentication between the signer and verifier, bundling rogue check of TPM and host in NFV system, an enhanced DAA scheme (hereinafter as NFV-DAA scheme) with mutual authentication which can meet all the above requirements is proposed. A remote anonymous authentication architecture for NFV system is constructed. The proposed NFV-DAA scheme has the following advantages with efficiency and security.

1) We put off J, K variables and those computations in the sign/verify stage in [10], and use a new variable $c_2 = H_2(f \parallel bsn)$ instead. In order to realize rogue list checking and user-controlled-linkability, the verifier can check the received c_2 with the RogueList to find out those rogue TPMs. Meanwhile, with the same bsn (base name) value, we can control the verifier to find out what messages are coming from the same TPM by getting out the same c_2 value. This scheme can reduce one scalar multiplication induced by the J, K pair computation.
2) Considering the low computing ability of the TPM, the computation of variable D is transferred from the TPM to issuer.
3) An efficient batch proof and verification scheme is used to reduce the computation of both the TPM and Host. In our NFV-DAA scheme, the TPM needs only to perform one exponentiation in the sign stage. However, this operation requires at least three exponentiations in the existing DAA schemes that provide the same functionality.
4) Elliptic curve is used in the Join, Sign and Verify protocols. It is shown in theoretical analysis that the protocols and algorithms used in RSA-DAA are complex and inefficient compared to those in ECC-DAA. Generally, TPM has lower computation load and higher communication efficiency in ECC-DAA, while the length of key and signature is also shorter.
5) The identity of TPM and Host is tied up and checked by the issuer and verifier. This technology prevents the attack of plugging a valid TPM into a malicious host. This security problem has not been considered in the existing other DAA schemes.
6) Traditional DAA schemes do not take mutual authentication into account. Despite that the issuer has a thorough mechanism to check the identity of TPM, TPM does not have any method to check the authenticity of the issuer and host. Therefore, the TPM and host would receive a forged certificate from a fake issuer, or the TPM be deceived by a fake host. In NFV-DAA scheme, a thorough mutual authentication is proposed, which ensures the legitimacy of the identity of all the protocol parties.

The rest of the paper is organized as following. Section 2 presents the NFV-DAA scheme for trusted NFV system. In Section 3, the security and performance analysis of the proposed scheme is presented. Finally, we conclude the paper in Section 4.

2 NFV-DAA Scheme for Trusted NFV System

2.1 Preliminary knowledge

1) Bilinear mapping

G_1, G_2 and G_T are cyclic groups with order of prime q,

$G_1=\langle P_1\rangle$, and $G_2=\langle P_2\rangle$, where P_1, P_2 is a generator of G_1, G_2 respectively. And calculating discrete logarithm on groups G_1, G_2, and G_T is difficult. Here, we also use the G_1, G_2 and G_T to represent the computation costs of the group G_1, G_2 and G_T.

If the mapping $\hat{t}:G_1\times G_2\rightarrow G_T$ satisfies the following conditions:

- $P_1\in G_1$, $P_2\in G_2$, $1\in G_T$, and $\hat{t}(P_1,P_2)\neq 1$
- $x\in G_1$, $y\in G_2$, then $\hat{t}(x,y)$ can be computed in polynomial time
- for $x\in G_1$, $y\in G_2$, and $a,b\in Z_P$, $\hat{t}(x,y)^{ab}=\hat{t}(x^a,y^b)$

Then $\hat{t}:G_1\times G_2\rightarrow G_T$ can be called as bilinear mapping.

2) The CDH problem

The computational Diffie-Hellman (CDH assumption) is the assumption that a certain computational problem within a cyclic group is hard. Consider a cyclic group $G_1=\langle P_1\rangle$. The CDH assumption states that, given aP_1, bP_1, and $a,b\in Z_q$ it is computationally difficult to get the value of abP_1.

Moreover, each party in the NFV-DAA scheme is presented as follows.

- Trusted Center (TC) and Issuer: They are the entity who issues the certificate. In this paper, we make no distinction between these two terms.
- TPM: It is the trusted platform module.
- Host: the host for the TPM. It is also the physical platform in trusted NFV system which providing resources for different VNF.
- Verifier: It verifies the signature. In this paper, the verifying operation is carried out by the counterpart VNF' s Host, therefore, the counterpart Host plays the role of Verifier in the NFV-DAA scheme. However, in the following discussion, we still put Host and Verifier as different logical entity.

The overall NFV-DAA scheme similarly includes Setup protocol which establishes system parameters, Join protocol which obtain certificate and the Sign/Verify protocol which do the Sign and authentication process. Specific description of each protocol is shown as follows.

2.1.1 Setup Protocol

Assuming that the bilinear pairs are $\hat{t}:G_1\times G_2\rightarrow G_T$ and $H_1:\{0,1\}^*\rightarrow Z_q$, we define the common parameter set $par_c=(G_1,G_2,G_T,\hat{t},P_1,P_2,q,H_1)$. Given that the public key and private key parameters of the issuer are ipk and isk respectively. Here, isk is $x,y\leftarrow Z_q$, ipk is (X,Y), while $X=xP_2\in G_2$, $Y=yP_2\in G_2$. In addition, the issuer will generate a pair of key (PK_I,SK_I) for mutual authentication. Finally, the issuer will provide a unique value H_2 to generate secret value f. Then we get the issuer parameter set $par_I=(ipk,K_I,PK_I)$.

Suppose that the parameter par_R of TPM is H_2, $H_2:\{0,1\}^*\leftarrow Z_q$. The public key of TPM is par_T, and the public and private EK key is (PK_T,SK_T), $par_T=PK_T$. The public and private key pair of the host is (PK_H,SK_H), $par_H=PK_H$. The parameters of sign/verify are par_s: $H_3:\{0,1\}^*\rightarrow Z_q$, $H_4:\{0,1\}^*\rightarrow Z_q$. After the protocol setup, the system public parameter set par is defined as $(par_C,par_I,par_T,par_H,par_S)$.

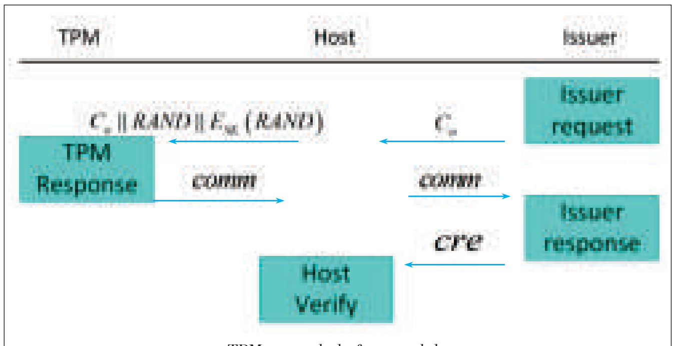
2.1.2 Join Protocol

The Join protocol is realized based on the request/response interaction between the issuer and TPM/host. We can divide the Join protocol into four parts in chronological order: the issuer request, TPM response, issuer response, and host verification. The overall process of the Join protocol is shown in **Fig. 1**.

The operation process of issuer request is shown in **Fig. 2**.

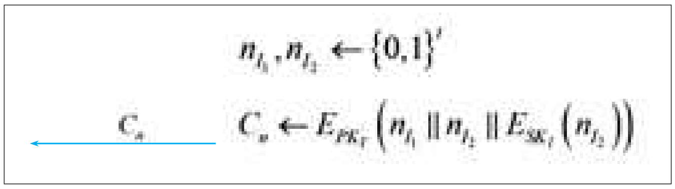
The issuer needs to confirm firstly that it is a legitimate trusted platform who has issued the DAA certificate, while the TPM also needs to check the legitimacy of the issuer. That is to say, an authentication channel should be established between the issuer and TPM in advance. The establishment is completed with the random numbers n_{I_1} and n_{I_2} chosen by the issuer, while n_{I_2} is encrypted with SK_I and then $n_{I_1}\|n_{I_2}\|E_{SK_I}(n_{I_2})$ is encrypted with PK_T to the host. Similar to the issuer, the host also generates a random number $RAND$, encrypts $RAND$ with pre - shared private key SK_H, and sends $C_n\|RAND\|E_{SK_H}(RAND)$ to TPM.

The TPM decrypts $E_{SK_H}(RAND)$ with the pre-shared public key PK_H to get $RAND^{'}$. The result of comparison between $RAND^{'}$ and $RAND$ indicates whether the host is legitimate. Only when $RAND^{'}$ is equal to $RAND$ will the TPM continue the protocol. Next, if the TPM can successfully decrypt C_n with SK_T, obtain the value of $n^{'}_{I_1}$ and return the hash value of



TPM: trusted platform module

▲Figure 1. Overall process of the Join protocol.


$n_{I_1},n_{I_2}\leftarrow\{0,1\}^{l}$
C_n
$C_n\leftarrow E_{PK_T}\left(n_{I_1}\|n_{I_2}\|E_{SK_I}\left(n_{I_2}\right)\right)$

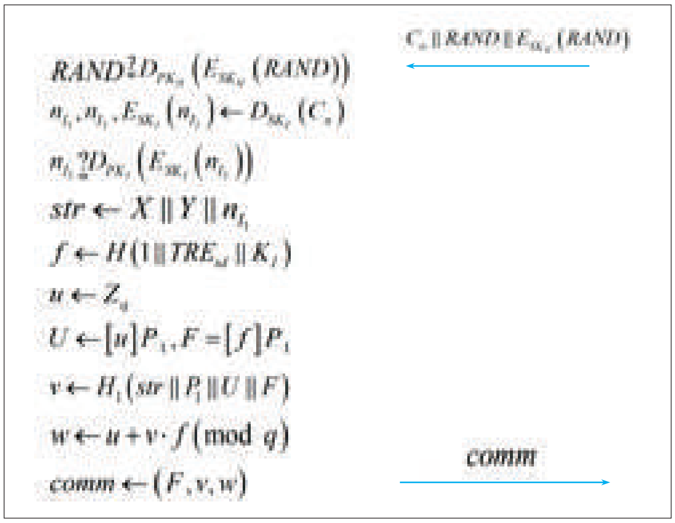
▲Figure 2. Flowchart of the issuer request.

n'_{I_1} to the issuer, it indicates that the TPM owns its legitimate EK private key. Besides, the TPM decrypts $E_{SK_I}(n_{I_2})$ with the legitimate EK public key of the issuer and compares the decrypted result n'_{I_2} with n_{I_2}. If they are equal, it indicates the legitimacy of the issuer. In this way, mutual authentication between the issuer and TPM is completed. **Fig. 3** shows the operating process of TPM response.

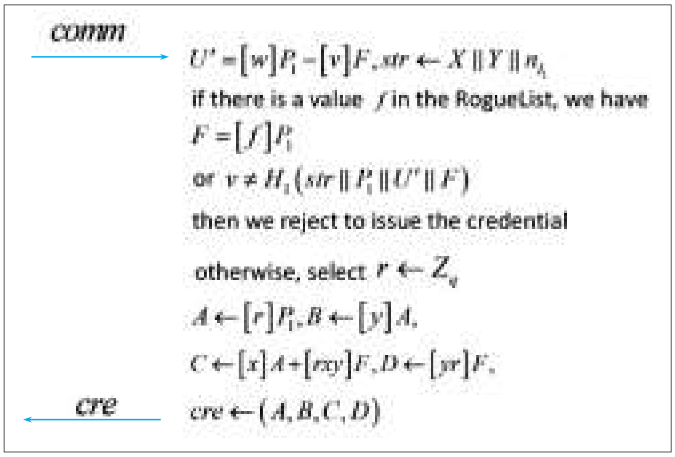
The TPM generates secret value f with K_I and TRE_{id} (a stable security parameter stored in TPM) and $f = H(1 \| TRE_{id} \| K_I)$. It also generates the comm value from f, and delivers it to the certificate issuer. **Fig. 4** shows the process of the issuer response.

The issuer will authenticate the *comm* value, that is to say, it will judge the zero-knowledge proof process the TPM performed to discrete logarithm f, and check whether the TPM owns the legitimate f or not. If the authentication is successful, the issuer will generate DAA certificate with F in comm. It is important to note that, the computation of D in certificate *cre* uses $D = [yr]F$, which uses F provided by TPM rather than f to compute $D = [f]B$ [12]. This is mainly due to the fact that F itself is generated from value f. For the generation of DAA certification (A, B, C, D), according to the B-bL-RSW principle of blind-bilinear assumption, the computation amount of TPM Join in NFV-DAA is reduced from $3G_1$ to $2G_1$. This computation amount is the lowest among the existing DAA schemes which are based on the LRSW or DDH difficulty assumption.

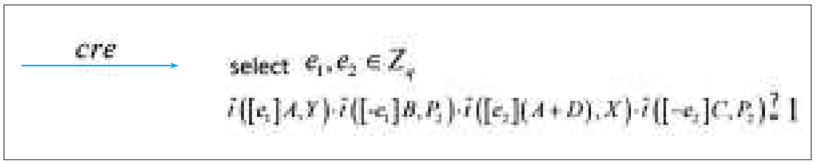
Furthermore, the process of host authentication is shown in **Fig. 5**. After receiving the certification *cre*, the host verifies correctness of *cre*. Based on the batch authentication technology, the host finds out whether the certificate is correct or not by using a P^4 computation. The P^4 computation will cost less than four independent bilinear pair computations (4*P*) [13]. The platform here does not have to perform very strict authentication of certificate and simple authentication is enough. The reason is that it does not affect the security of the entire DAA even if we cannot completely guarantee the dependability of certificate at this time. In subsequent Sign/Verify, there are other operations to verify the certificate.


$C_a \| RAND \| E_{SK_H}(RAND)$
$RAND \overset{?}{=} D_{PK_H}(E_{SK_H}(RAND))$
$n_{I_1}, n_{I_2}, E_{SK_I}(n_{I_2}) \leftarrow D_{SK_T}(C_a)$
$n_{I_1} \overset{?}{=} D_{PK_I}(E_{SK_I}(n_{I_2}))$
$str \leftarrow X \| Y \| n_{I_1}$
$f \leftarrow H(1 \| TRE_{id} \| K_I)$
$u \leftarrow Z_q$
$U \leftarrow [u]P_1, F = [f]P_1$
$v \leftarrow H_1(str \| P_1 \| U \| F)$
$w \leftarrow u + v \cdot f \pmod{q}$
$comm \leftarrow (F, v, w)$
comm

▲Figure 3. Response flowchart of the trusted platform module.


comm
$U' = [w]P_1 - [v]F, str \leftarrow X \| Y \| n_{I_1}$
if there is a value f in the RogueList, we have
$F = [f]P_1$
or $v \neq H_1(str \| P_1 \| U' \| F)$
then we reject to issue the credential
otherwise, select $r \leftarrow Z_q$
$A \leftarrow [r]P_1, B \leftarrow [y]A,$
$C \leftarrow [x]A + [rxy]F, D \leftarrow [yr]F,$
cre
$cre \leftarrow (A, B, C, D)$

▲Figure 4. Flowchart of the issuer response.


cre
select $e_1, e_2 \in Z_q$
$\hat{t}([e_1]A, Y) \cdot \hat{t}([-e_1]B, P_2) \cdot \hat{t}([e_2](A+D), X) \cdot \hat{t}([-e_2]C, P_2) \overset{?}{=} 1$

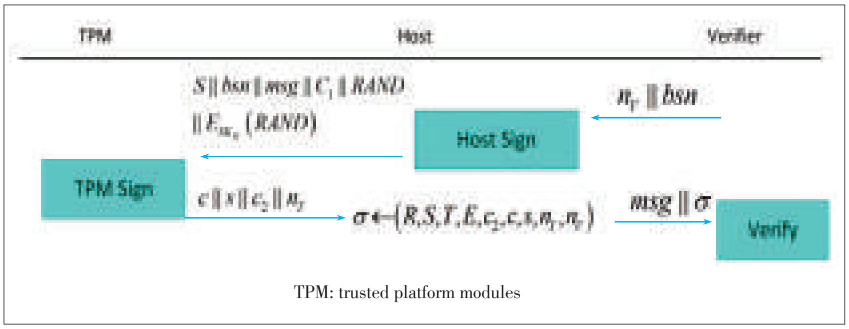
▲Figure 5. Flowchart of host verify.

2.1.3 Sign/Verify Protocol

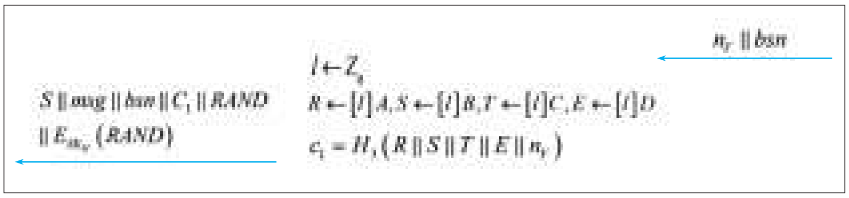
The Sign/Verify protocol refers to the process in which the TPM, together with the host, performs the knowledge sign of message *msg* and generates DAA signature σ, and then sends σ to the verifier. The operations in chronological order in the Sign/Verify protocol can be divided into three parts: host sign, TPM sign, and verify. The total framework of the Sign/Verify protocol is shown in **Fig. 6**.

The operation of host sign is shown in **Fig. 7**. The host performs blind computation of DAA certification value (A, B, C, D) after receiving n_V and base name $b\,\text{sn}$. Then the host generates blind certificate (R, S, T, E). Meanwhile, the host generates a random number *RAND* and encrypts it with the pre-shared private key SK_H, which is similar to the procedure in the Join protocol. Next, the host sends $S \| bsn \| msg \| c_1 \| RAND \| E_{SK_H}(RAND)$ to TPM.

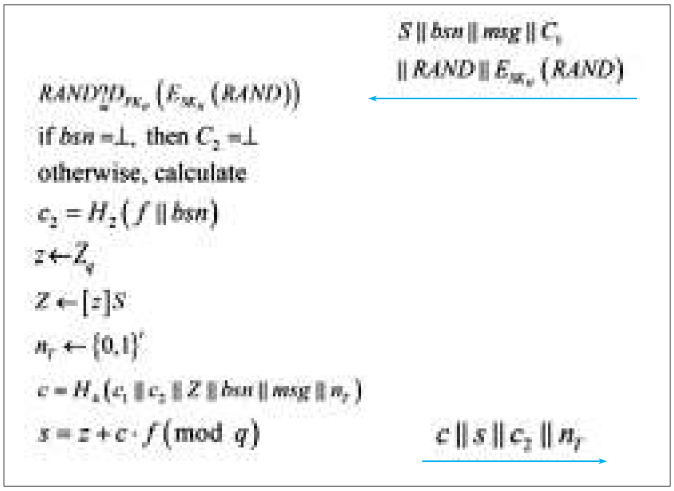
The operation of TPM sign is shown in **Fig. 8**. After receiving from the host, the TPM verifies the legitimacy of the verifier and host using the same method as the Join protocol. The TPM computes $RAND'$ in the way of decrypting the $E_{SK_H}(RAND)$ with the pre-shared key PK_H, and compares



▲Figure 6. Overall framework of the Sign/Verify protocol.


$n_V\|bsn$
$l\leftarrow Z_q$
$S\|msg\|bsn\|C_1\|RAND$
$\|E_{sk_v}(RAND)$
$R\leftarrow[l]A, S\leftarrow[l]B, T\leftarrow[l]C, E\leftarrow[l]D$
$c_1=H_1(R\|S\|T\|E\|n_V)$

▲Figure 7. Flowchart of the host sign operation.



▲Figure 8. The sign operation of the trusted platform module.

$RAND'$ with $RAND$. Only when the values are equal is the host proved to be legitimate.

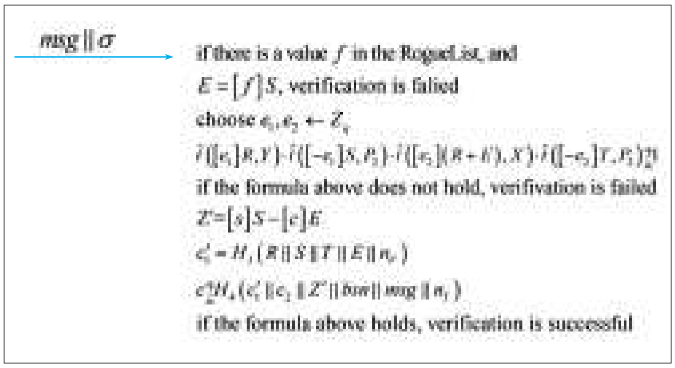
The TPM continues to finish the rest computation of the signature value after checking the legitimacy of the host. It generates independent value c_2 for relevance detection as $c_2=H_2(f\|bsn)$. Then it considers the c_2 as the public signature member value, and performs the zero-knowledge proof of possessing a legitimate DAA certification. It also generates c and s values, while c_2 is add into the Hash computation of c. The TPM sends c, s and the random number n_T all together to the host, the final signature σ generated by the host is constructed as $(R,S,T,E,c,s,n_2,n_T,n_V)$. Then, the operation of Verify is shown in **Fig. 9**.

The verifier performs the verify operation after receiving the signature σ. It firstly substitutes the blind certificate value $E=[f]S$ for the counterfeit f on RogueList to check if the f value used in the signature has been disclosed, then verifies whether the blind signature value (R,S,T,E) is correct or not, and judges whether the zero-knowledge proof of the legitimate DAA certification in signature is correct. If all these are correct, it indicates that the signer owns a legitimate secret value f and the legitimate DAA certificate based on the same f. If the verifier has been provided with a specific *bsn* in advance, the relevance detection of signature is also needed. Relevance detection can be performed by using the signature member value c_2 generated from the secret value f and the base name bsn of the verifier. The entire verify process is completed only if all these verification steps are completed.

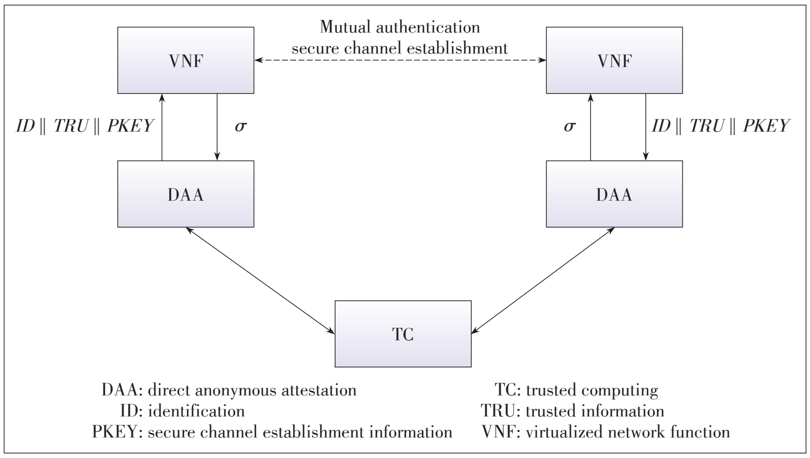
2.2 VNF Mutual Authentication and Secure Channel Establishment

In trusted NFV architecture, the VNF modules are able to mutually authenticate in a security and effective way as well as to establish secure a communication channel by the support of the NFV-DAA scheme. Mutual authentication refers to two parties authenticating each other. Under trusted NFV architecture, any two VNF should authenticating each other before communication in order to verify the identity and establish a security channel, i.e. exchanging the session key. The mutual authentication and security channel establishment is shown in **Fig. 10**.

As stated in Section 2.1, DAA's Sign protocol can generate signatures for any *msg* (message) accordingly. Therefore, during the mutual authentication, VNF sends 3 parameters (identification ID, trusted information TRU and secure channel establishment information PKEY) as the msg to NFV-DAA for the signature. The generated signature will be $\sigma_{ID\|TRU\|PKEY}=Sign(ID\|TRU\|PKEY)$. It will be sent to the counterpart VNF module in the other side. Then the recipient side will send the received signature to its own NFV-DAA scheme


$msg\|\sigma$
if there is a value f in the RogueList, and
$E=[f]S$, verification is failed
choose $e_1,e_2\leftarrow Z_q$
$\hat{t}([e_1]R,Y)\cdot\hat{t}([-e_1]S,P_2)\cdot\hat{t}([e_2](R+E),X)\cdot\hat{t}([-e_2]T,P_2)\overset{?}{=}1$
if the formula above does not hold, verifivation is failed
$Z'=[s]S-[c]E$
$c_1'=H_1(R\|S\|T\|E\|n_V)$
$c\overset{?}{=}H_4(c_1'\|c_2\|Z'\|bsn\|msg\|n_T)$
if the formula above holds, verification is successful

▲Figure 9. Flowchart of the verify operation.


Mutual authentication
secure channel establishment
VNF
VNF
ID ‖ TRU ‖ PKEY
σ
σ
ID ‖ TRU ‖ PKEY
DAA
DAA
TC
DAA: direct anonymous attestation
ID: identification
PKEY: secure channel establishment information
TC: trusted computing
TRU: trusted information
VNF: virtualized network function

▲Figure 10. NFV-DAA scheme application in trusted NFV system.

to verify its legality. After the verification, TRU is verified by the credibility verification service provided by NFV infrastructure. It finishes the entire process by establishing a secure communication channel under the exchange of keys through PKEY. ID and TRU are all security parameters coming from the two parties of mutual authentication, i.e. two VNFs.

3 Security and Performance Analysis of NFV-DAA Scheme

3.1 Security

The correctness and security of overall NFV-DAA protocol are analyzed in this section.

The CDH problem can be solved if the NFV-DAA scheme can be a breakthrough when the security parameters saved in TPM are not leaked.

At the time of verifying, (1) guarantees the correctness of NFV-DAA process.

$$\hat{\imath}([e_1]R,Y)\cdot\hat{\imath}([-e_1]S,P_2)\cdot\hat{\imath}([e_2](R+E),X)\cdot\hat{\imath}([-e_2]T,P_2)=1. \qquad (1)$$

To prove (1) true, $\hat{\imath}(R,Y)=\hat{\imath}(R,yP_2)=\hat{\imath}(yR,P_2)=\hat{\imath}(S,P_2)$ and $\hat{\imath}(R+fS,X)=\hat{\imath}(R+fS,xP_2)=\hat{\imath}(x(R+fS),P_2)=\hat{\imath}(T,P_2)$ are needed to be true. It indicates that the DAA certificate of the signature is generated in a correct way. The security of NFV-DAA is mainly reflected as follows: as long as the secret value of TPM *f* and DAA certification have not been disclosed, an attacker is unable to carry out any attacks, which ensure the security of NFV-DAA scheme.

The proof is as follows: in the context where no the TPM secret value or DAA certification is disclosed, without the verifier, an attacker should provide (R,S,T,E) alone to make (1) true, which expects $S=[y]R$ and $T=[x](R+E)$ to be true. Assume that attacker A selects $R=[\alpha]P_1$, $S=[\beta]P_1$, $T=[\gamma]P_1$ and $E=[\delta]P_1$, and the public key of the issuer is known as $X=[x]P_2$, $Y=[y]P_1$ and $S=[y\cdot\alpha]P_1$ which is needed to make $S=[y]R$ true. It means that given $[\alpha]P_1$ and $[y]P_1$, the attacker must be able to calculate $[y\cdot\alpha]P_1$, so as to solve the CDH problem in group G_1. However, it is obviously impossible for the attacker to solve the CDH problem. The non-symmetric bilinear pair is generally considered to be difficult.

In addition, different from the existing DAA schemes, NFV-DAA has the feature of mutual authentication. We assume that prior to any system setup, each issuer has its private endorsement private key SK_I and each TPM has the corresponding public key PK_I. The issuer generates a random number n_{I_2} and encrypts it with SK_I. The TPM admits the legitimacy of the issuer if the decrypting result is equal to the received n_{I_2}. In other words, the issuer sends its signature to the TPM for checking. Considering the issuer has checked the TPM by the endorsement key pair SK_T/PK_T in the Join protocol, mutual authentication is realized.

In order to prevent a corrupted host from taking advantages of an honest TPM to sign on an illegal message, it is necessary to bind the TPM and host when manufacturing the devices. A pair of pre-shared public/private key PK_H/SK_H is embedded into the TPM and host respectively. For the Join protocol and Sign protocol, the host needs to generate a random number $RAND$ and sends $RAND\|E_{SK_H}(RAND)$ to the TPM. The TPM checks the consistency of the decrypting result $RAND'$ and $RAND$ to verify the legitimacy of the host. Here, we assume that the embedded key cannot be extracted from the TPM and host due to hardware protection.

The existing DAA schemes prevent the change of *bsn* by the host to make signatures linkable, but they cannot prevent that a malicious message is delivered to TPM by the host to generate a legal signature. Since a valid TPM may be used in an illegal way in existing DAA schemes, verifying the identity of the host is necessary. In the NFV-DAA scheme, mutual authentication ensures the legitimacy of the host, hence the host will not deliver illegal messages to the TPM or disclose the identity of the TPM directly.

Based on all of the above results, it is easy to find out that the proposed NFV-DAA scheme is secure enough on the premise that the TPM is secure and credible. Compared with those DAA schemes based on LRSW and DDH assumptions, NFV-DAA has no security weakness and can improve overall protocol efficiency. It has the highest running efficiency among all the existing DAA schemes based on LRSW and DDH assumption at present. Both the Join and Sign protocols of the NFV-DAA scheme have been improved, the computation amount of TPM Join is reduced to $2G_1$, and that of TPM Sign is as low as $1G_1$. The detailed analysis of NFV-DAA efficiency will be presented in next section. The benefits of secured security, low cost, high efficiency, and being easy to implement help the NFV-DAA scheme meet the dual requirements for security and economic benefits of the trusted NFV system.

3.2 Performance

In this paper, we compare the performance of the scheme in [12] with that of the NFV-DAA scheme. In other words, we compare the time overhead of the Join protocol and Sign/verify protocol in both schemes. Here, the alternative simulation is used to make the experiments. It means that without considering the communication time between the TPM and the remote issuer, and between the TPM and the remote verifier, we just focus on the time overhead on the protocol operations of each protocol entity.

Based on the above consideration, we set the host, issuer and verifier to the same PC host. The software simulation scheme [14] is used to internally install the TPM to the same PC host, which communicates with the TPM via the hardware interface. Statistics on the time overhead of each protocol are provided in the stand-stone simulation environment. Besides, the protocol parameters are chosen the same as those in [15].

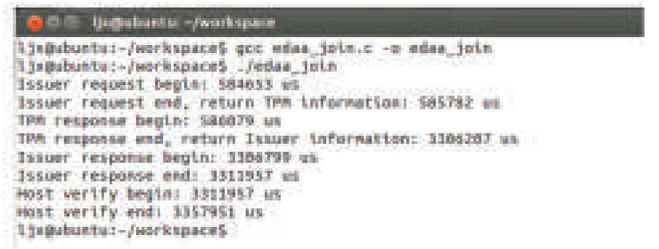
3.2.1 Join

In accordance with the NFV-DAA Join protocol process, with the cryptographic algorithm library package in OpenSSL software, we used C++ to write the client program edaa_join.c in ubuntu 9.10. Main parts of the edaa_join.c include the issuer requirement, issuer response and host verification. The TPM response is fulfilled by the TPM software. The program computed the time overhead of each protocol in microseconds and ran by calling the timing function gettimeofday( ) of the system. The experimental results after running edaa_join.c are shown in **Fig. 11**.

Furthermore, we did experiments to verify the scheme [12] and the experimental results are shown in **Fig. 12**.

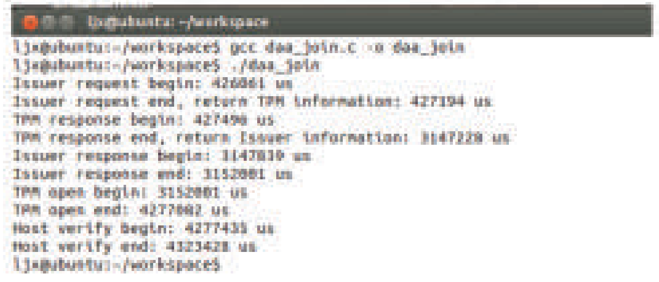
According to the results in Figs. 11 and 12, it is easy to compute the time overhead in the two schemes (**Table 1**).

In Table 1, the main differences between NFV-DAA and the scheme in [12] lie in two aspects. On one hand, the NFV-DAA scheme does not have the TPM Open process. On the other hand, the time overhead of the issuer response in the NFV-DAA scheme is 996 us larger than that in the scheme in [12]. The reason is that in NFV-DAA, the TPM does not have TPM Open operation, while the issuer makes the operation on behalf of the TPM. In this way, the issuer fulfills a group G_1 exponential operation originally conducted by the TPM. With the usage of Batch technology, the issuer response costs just 996 us more than the scheme in [12], which is much smaller than the TPM Open cost of 1,125,081 us. It is also found that the Join's total time of NFV-DAA is 2,770,411 us. Compared with the time overhead 3,896, 101us in the scheme [12], the performance of NFV-DAA Join is improved up to about 29%.



```
ljs@ubuntu: ~/workspace
ljs@ubuntu:~/workspace$ gcc edaa_join.c -o edaa_join
ljs@ubuntu:~/workspace$ ./edaa_join
Issuer request begin: 584653 us
Issuer request end, return TPM information: 585782 us
TPM response begin: 586079 us
TPM response end, return Issuer information: 3306207 us
Issuer response begin: 3306799 us
Issuer response end: 3311957 us
Host verify begin: 3311957 us
Host verify end: 3357951 us
ljs@ubuntu:~/workspace$
```

▲Figure 11. Time overhead of the Join protocol in NFV-DAA scheme.



```
ljs@ubuntu: ~/workspace
ljs@ubuntu:~/workspace$ gcc daa_join.c -o daa_join
ljs@ubuntu:~/workspace$ ./daa_join
Issuer request begin: 426061 us
Issuer request end, return TPM information: 427194 us
TPM response begin: 427496 us
TPM response end, return Issuer information: 3147228 us
Issuer response begin: 3147839 us
Issuer response end: 3152001 us
TPM open begin: 3152001 us
TPM open end: 4277082 us
Host verify begin: 4277435 us
Host verify end: 4323428 us
ljs@ubuntu:~/workspace$
```

▲Figure 12. Time overhead of the Join protocol in the scheme in [12].

3.2.2 Sign/Verify

Two more software programs edaa_sign.c and edaa_verify.c are written to test the performance of Sign/Verify protocol. **Table 2** provides the final statistics of comparing Sign/Verify time overhead in NFV-DAA and the scheme in [12].

From Table 2, the time of the scheme in [12] is different for the signature with correlation ($bsn = \perp$) and without correlation

▼Table 1. Comparison of NFV-DAA and the scheme in [12] on the time overhead of Join

Scheme	Join protocol					
	Issuer request (us)	TPM response (us)	Issuer response (us)	TPM open (us)	Host verifies (us)	Total time (us)
NFV-DAA	1129	2,720,128	5158	-	45,994	2,770,411
The scheme in [12]	1133	2,719,732	4162	1,125,081	45,993	3,896,101

DAA: direct anonymous attestation NFV: network function virtualization TPM: trusted platform module

▼Table 2. Comparison of NFV-DAA and the scheme in [12] on the time of Sign/Verify

Scheme		Sign/Verify procedure			
		Host Sign (us)	TPM Sign (us)	Verify (us)	Total time (us)
NFV-DAA		4088	1,149,213	47,301	1,200,602
The scheme in [12]	$bsn = \perp$	4146	1,148,901	47,232	1,200,279
	$bsn \neq \perp$	6312	2,292,876	48,310	2,347,498

DAA: direct anonymous attestation NFV: network function virtualization TPM: trusted platform module

($bsn \neq \perp$). The improvement of the NFV-DAA scheme is that no matter whether the signature has correlation, the computation time of each entity is the same as that of the scheme without correlation in the scheme in [12]. It is found that the NFV-DAA scheme costs 1,200,602 us that is similar to the time overhead in the scheme in [12] when $bsn = \perp$. However, compared to the time in the scheme in [12] when $bsn \neq \perp$, it is obvious that the performance of the NFV-DAA scheme's Sign/Verify is improved up to 49%. The reason is that the NFV-DAA scheme uses $c_2 = H_2\left(f \| bsn\right)$ instead of J and K in [12] for signature correlation detection so that TPM Sign gets less group G_1 exponential operation than in [12].

4 Conclusions

A secure and high efficient NFV-DAA scheme is proposed in this paper. The scheme is designed based on the architecture of trusted NFV system, taking advantages of existing security TPM in the architecture. Therefore, the scheme can be integrated into the architecture seamlessly. With a mutual authentication mechanism that the existing DAA schemes do not have and an efficient batch proof and verification scheme, the trusted NFV system has optimized performance. From the experiment results, we can find out that the proposed NFV-DAA scheme has higher security level and efficiency than those existing DAA schemes. The computation load in Join protocol is reduced from $3G_1$ to $2G_1$ exponential operation, while the time of NFV-DAA scheme's Sign/Verify protocol is improved up to 49%.

References

[1] ETSI. (2016 Jul.). *Network functions virtualization (NFV)* [Online]. Available: http://portal.etsi.org/NFV

[2] E. Brickel, J. Camenisch, and L. Q. Chen, "Direct anonymous attestation," in *Proc. 11th ACM Conference on Computer and Communications Security*, Washington DC, USA, 2004, pp. 132–145. doi:10.1145/1030083.1030103.

[3] *Information Technology Security Techniques-Trusted Platform Module*, ISO/IEC 11889, 2009.

[4] E. Brickell and J. Li, "A pairing-based DAA scheme further reducing TPM resource," in *International Conference on Trust and Trustworthy Computing*, Heidelberg, Germany, 2010, pp. 902–915. doi: 10.1007/978-3-642-13869-0_12.

[5] X. M. Wang, H. Y. Heyou, and R. H. Zhang, "One kind of cross-domain DAA scheme from bilinear mapping," in *IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, Beijing, China, 2014, pp. 237–243. doi:10.1109/TrustCom.2014.62.

[6] B. Zhu, H. H. Cui, L. Chen, and C. Tang, "Improvement of the DAA protocol based on TPM," in *3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, Chengdu, China, 2010, pp. 401–404. doi: 10.1109/ICCSIT.2010.5564832.

[7] L. Q. Chen, P. Morrissey, and N. P. Smart, "Pairings in trusted computing," in *International Conference on Pairing-Based Cryptography*, Heidelberg, Germany, 2008, pp. 1–17. doi: 10.1007/978-3-540-85538-5_1.

[8] L. Q. Chen, P. Morrissey, and N. P. Smart. (2016 Jul.). Fixing the pairing based protocols. Cryptology ePrint Archive Report 2009/198 [Online]. Available: http://eprint.iacr.org/2009/198

[9] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Advances in Cryptology-CRYPTO*, Berlin, Germany, 2004, pp. 56–72. doi: 10.1007/978-3-540-28628-8_4.

[10] L. Tan and M. T. Zhou, "A new process and framework for direct anonymous attestation based on asymmetric bilinear maps," *Chinese Journal of Electronics*, vol. 22, no. 4, pp. 695–701, 2013.

[11] L. Yang, J. F. Ma, and W. Wang, "Multi-domain direct anonymous attestation scheme from pairings," in *International Conference on Network and System Security*, Xi'an, China, 2014, pp. 566–573. doi: 10.1007/978-3-319-11698-3_47.

[12] L. Q. Chen, "A DAA scheme using batch proof and verification," in *International Conference on Trust and Trustworthy Computing*, Heidelberg, Germany, 2010, pp. 166–180. doi: 10.1007/978-3-642-13869-0_11.

[13] R. Granger, N. P. Smart. (2016 Jul.). On computing products of pairings. Cryptology ePrint Archive Report 2006/172 [Online]. Available: http://eprint.iacr.org/2006/172

[14] M. Strasser and H. Stamer, "A software-based trusted platform module emulator," in *International Conference on Trusted Computing*, Heidelberg, Germany, 2008, pp. 33–47. doi: 10.1007/978-3-540-68979-9_3.

[15] L. Q. Chen, D. Page, and N. P. Smart, "On the design and implementation of an efficient DAA scheme," in *International Conference on Smart Card Research and Advanced Applications*, Heidelberg, Germany, 2008, pp. 223–238. doi: 10.1007/978-3-642-12510-2_16.

Manuscript received: 2017-12-05

Biographies

CHEN Liquan (Lqchen@seu.edu.cn) received his B.Sc. degree in electronic engineering from Nanjing University, China in 1998, M.Sc. degree in radio engineering from the Purple Mountain Observatory, Chinese Academic of Sciences in 2001, and Ph.D. degree in signal processing from Southeast University, China in 2005. He is presently engaged in information processing and communication network research as a professor at Southeast University, China.

ZHU Zheng (zhuzheng@seu.edu.cn) is currently a master student at Southeast University, China. His research interests include information security and computer networks.

WANG Yansong (wang.yansong@zte.com.cn) is a principle product manager of ZTE Corporation. His research interests include communications technologies and computer networks.

LU Hua (lu.hua@zte.com.cn) is an engineer of ZTE Corporation. His research interests include 5G technologies, computer networks.

CHEN Yang (chenyang90@seu.edu.cn) is currently a master student at Southeast University, China. His research interests include information security and digital communications.

DOI: 10.19729/j.cnki.1673-5188.2018.04.006
http://kns.cnki.net/kcms/detail/34.1294.TN.20181203.0828.002.html, published online December 3, 2018

Antenna Mechanical Pose Measurement Based on Structure from Motion

XU Kun[1], FAN Guotian[1], ZHOU Yi[1], ZHAN Haisheng[2], and GUO Zongyi[2]
(1. ZTE Corporation, Shenzhen 518057, China;
2. Xidian University, Xi'an 710000, China)

Abstract

Antenna mechanical pose measurement has always been a crucial issue for radio frequency (RF) engineers, owning to the need for mechanical pose adjustment to satisfy the changing surroundings. Traditionally, the pose is estimated in the contact way with the help of many kinds of measuring equipment, but the measurement accuracy cannot be well assured in this way. We propose a non-contact measuring system based on Structure from Motion (SfM) in the field of photogrammetry. The accurate pose would be estimated by only taking several images of the antenna and after some easy interaction on the smartphone. Extensive experiments show that the error ranges of antenna's downtilt and heading are within 2 degrees and 5 degrees respectively, with the shooting distance in 25 m. The GPS error is also under 5 meters with this shooting distance. We develop the measuring applications both in PC and android smartphones and the results can be computed within 3 minutes on both platforms. The proposed system is quite safe, convenient and efficient for engineers to use in their daily work. To the best of our knowledge, this is the first pipeline that solves the antenna pose measuring problem by the photogrammetry method on the mobile platform.

Keywords

antenna mechanical pose measurement; SfM; photogrammetry; smartphone

1 Introduction

Due to the increase of mobile phone users, more and more GSM antennas need to be set up in populous regions. At the same time, the maintenance of a large number of antennas has been a difficult issue for radio frequency (RF) engineers.

Mechanical pose measurement is crucial for antenna management because any minor mechanical pose adjustment may cause big changes of antenna radiation patterns. Specifically, the mechanical pose of an antenna includes the heading, downtilt, Global Position System (GPS) location, and altitude. The heading is the horizontal angle of the antenna relative to true north and the downtilt is antenna's downward angle skews from the radial which is vertical to the ground in 3D space. RF engineers need to adjust these two angles according to surrounding changes, therefore, the two angles are the most vital parameters of the pose measurement. **Fig. 1** shows the mechanical parameters of an ordinary GSM sector antenna.

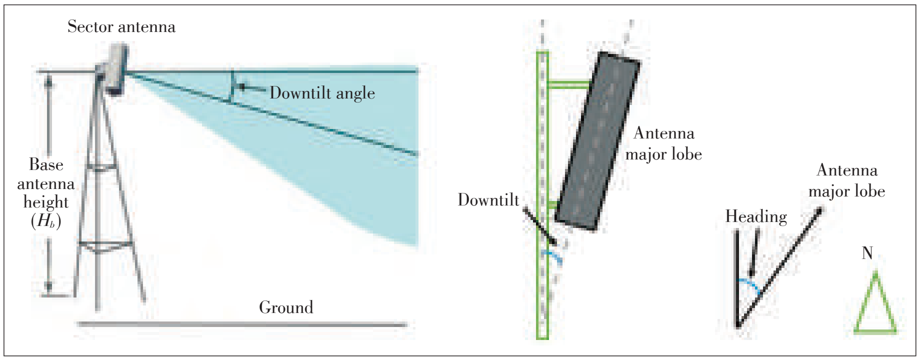
In traditional ways, RF engineers usually need to climb up towers to measure antenna poses, helped with many kinds of measuring tools like [1]. There are several drawbacks of this contact-type measuring way, which requires that engineers must contact antennas closely enough to get the parameters. The biggest risk is the security of engineers. Although wearing safety equipment, it is still dangerous for engineers to climb up tall towers with various structures. The measuring error is another problem, because the installation of measuring tools is easily affected by the human factor. Minor mounting displacement of the tools may lead to different measuring results. Moreover, equipment expenses, tools and personnel placement also make the measurement a costly procedure. Owing to all the disadvantages, it is not so plausible for engineers to measure antenna poses in a contact-type measuring way.

We want to solve the measuring problem in photogrammetry way. That is, engineers only need to use mobile phones to take several images of the antenna and well estimate the antenna pose well by easy interaction with the related applications. It is quite different from the traditional ways because engineers do not need to get close to the antenna anymore and remote measurement is available in this photogrammetry way.

We propose a measuring system based on Structure from Motion (SfM). SfM could estimate 3D structures from 2D image sequences. At the same time, the intrinsic and extrinsic parameters of each camera corresponding to each photo is calculated and we can reconstruct the object we want to obtain the antenna pose.

By the proposed way, engineers first take 5 to 10 images of an antenna and store the pose of the smart phone at the moment each image is taken. This pose is relative to the geodetic coordinate system. The quality of images should be guaranteed, which means there is not too much noise or motion blur in the images. Second, SfM is performed by these images. We can get the necessary information of every camera by this procedure. Third, we calculate the rotation, scale, translation transformation parameters from the SfM outputs and poses of the smart phone. These transformation parameters intend to transform the structures from SfM coordinate to geocentric coordinate. Fourth, engineers are guided to draw the bounding box of the antenna in each image and line extraction algorithm

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



▲**Figure 1. The downtilt, heading, location and height of an antenna.**

will be performed to show the ID of each line in the small selected image. Finally, by choosing the corresponding lines of the antenna in each image as inputs, the triangulation of the corresponding lines and points will provide the final pose of the antenna.

In Section 2, we give an overview of SfM and characteristics of various SfM algorithms. In Section 3, the whole photogrammetry-based measuring system is illustrated in detail. Section 4 validates our algorithms in indoor and outdoor datasets. We perform comprehensive experiments in different environments and analyze the experiment results. We conclude the paper in Section 5.

2 Structure from Motion

2.1 Motivation

In multi-view geometry, if we want to reconstruct the 3D shape of any object from 2D images, we have to know the intrinsic and extrinsic parameters of each camera. The phone that we use to capture the images provides some intrinsic parameters like focal length and pixel coordinates of principal points (could be computed from the image size). The extrinsic parameters, including 3D position and rotation matrix around the geodetic coordinate frame, can also be obtained by built-in sensors of the phone. However, the accuracy of these parameters cannot satisfy the need for reconstruction of the target. For example, the highest accuracy of GPS location of a smartphone is no better than 3 meters, even though corrected by over ten GPS satellites.

SfM can solve this problem because only images are required for the calculation of the parameters of each camera. We can use these parameters to triangulate the object we want and then transform the pose of the object from SfM coordinate system to geodetic coordinate system.

2.2 Pipeline of SfM

SfM for computer vision has received tremendous attention in the last decade. The proposed methods can be divided into two classes: sequential methods and global methods.

Sequential methods start from reconstruction of two or three views, then incrementally add new views into a merged representation. Bundler [2] is one of the most widely used sequential pipelines. However, there are several drawbacks of sequential methods. The quality of reconstruction is heavily affected by the choice of the initial images and the order of the subsequent image additions. Another disadvantage is that sequential methods are tend to suffer from the drift due to the accumulation of errors and cycle closures of the camera trajectory is hard to handle. The running speed of sequential methods is also a slow procedure, especially dealing with large image datasets.

Global methods have better performance than sequential ones. The classical pipeline of global methods can be summarized as following procedures.

1) Feature detection and matching

To find the correspondences between images, local corner features are detected and described. Scale-Invariant Feature Transform (SIFT) [3] is one of the most widely used feature detectors. These features are usually described as high-dimension vectors and can be matched by their differences. However, some of the matched features are incorrectly matched. These mismatches are called outliers and needed to be filtered. For example, Random Sample Consensus (RANSAC) [4] is often used to efficiently remove these outliers and keep the inliers in a certain probability.

2) Relative pose estimation

Given 2D-2D point correspondences between two images, we could recover the relative positions and orientation of the camera as well as the positions of the points (up to an unknown global scale factor) by the two-view geometry theory. Specifically, the essential matrix relating a pair of calibrated views can be estimated from eight or more point correspondences by solving a linear equation and the essential matrix could be decomposed to relative camera orientation and position. This issue is well illustrated by Hartley et al [5].

3) Absolute pose estimation

This procedure aims to robustly recover the absolute global pose of each camera from relative camera motions. Because of the fact that the relative rotation can be estimated much more precisely than relative translation even for small baselines, the global rotation averaging can be performed beforehand and then the translation averaging can compute the absolute translation with the orientations fixed. Essential matrices only determine camera positions in a parallel rigid graph, so essential matrix based methods [6], [7] are usually ill-posed at collinear camera motion. In another way, trifocal tensor based methods [8], [9] are robust to collinear motion because relative scales of translations are encoded in a trifocal tensor.

4) Bundle adjustment

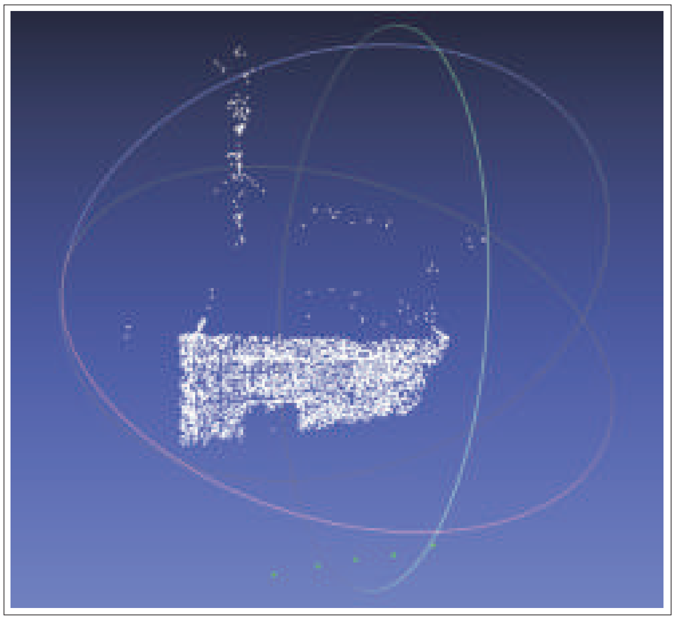
SfM gives an initial estimation of each camera's projection matrices and also the 3D points from images features. However, it is still necessary to refine this estimation using iterative non-linear optimization method. Bundle adjustment is defined as the problem of refining the 3D points of the scene and the intrinsic and extrinsic parameters of each camera, according to the optimal criterion involving the corresponding image projections of all points. The Levenberg-Marquardt (LM) based algorithm [10] is the most popular method for solving non-linear least squares problems and the choice for bundle adjustment.

3 Measurement System

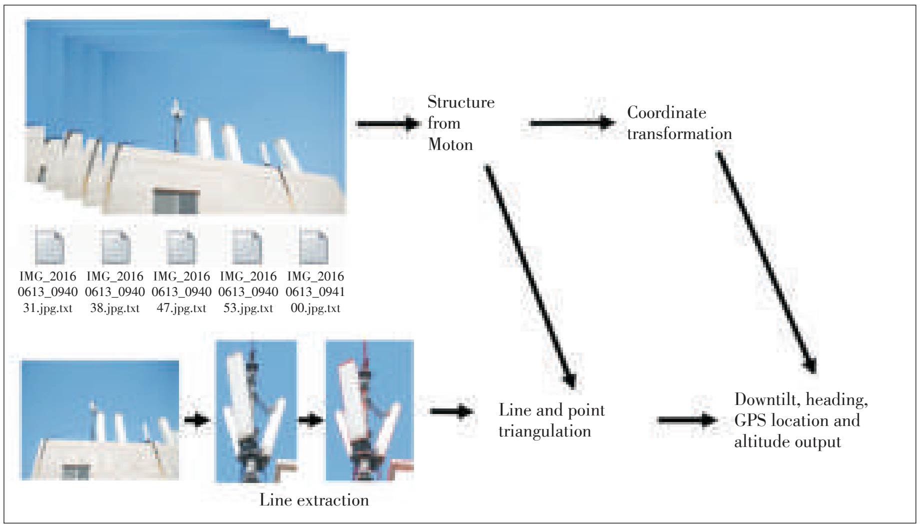
Our photography based measurement system can be implemented by the following steps: 1) taking 5 to 10 sequential images of an antenna and storing the poses of the phone relative to the geodetic coordinate system; 2) performing SfM on the images taken from the antenna; 3) estimating the rotation, scale and translation transformation parameters which convert the structures from SfM space to geodetic space; 4) selecting the small image of the antenna from each image, performing line extraction and choosing the corresponding lines of the antenna; 5) triangulating the line correspondences and estimating downtilt and heading; 6) triangulating the point correspondences and calculating the GPS and height. **Fig. 2** shows the whole pipeline of the measurment system.

3.1 Structure from Motion

Users need to take sequential images that include n different views of the antenna by the smartphone. Meanwhile the corresponding text file of each image is created, which stores camera poses, including rotation matrices, GPS, and height. We can get the intrinsic and extrinsic parameters of each camera by SfM. **Fig. 3** shows the output of SfM procedure. The points in white show the outline of the scene and the points in green are the cameras' positions.



▲**Figure 3. The output point clouds of Structure from Motion.**


Structure
from
Moton
Coordinate
transformation
IMG_2016 0613_0940 31.jpg.txt
IMG_2016 0613_0940 38.jpg.txt
IMG_2016 0613_0940 47.jpg.txt
IMG_2016 0613_0940 53.jpg.txt
IMG_2016 0613_0941 00.jpg.txt
Line and point
triangulation
Downtilt, heading,
GPS location and
altitude output
Line extraction

▲**Figure 2. The pipeline of the photography measurement system.**

3.2 Coordinate Transformation

3.2.1 SfM and Geodetic Coordinate System

Because cameras' parameters are estimated in the so-called SfM space, all the extrinsic parameters are relative to the SfM coordinate system. On the other hand, at the moment we take the images of the target, we can store the camera's parameters, including rotation matrices, GPS and height, which are relative to the geodetic coordinate system. **Fig. 4a** shows the geodetic coordinate system and **Fig. 4b** shows the device coordinate system. Android Application Program Interfaces (APIs) provide the access to get the camera pose of the device relative to the geodetic coordinate system.

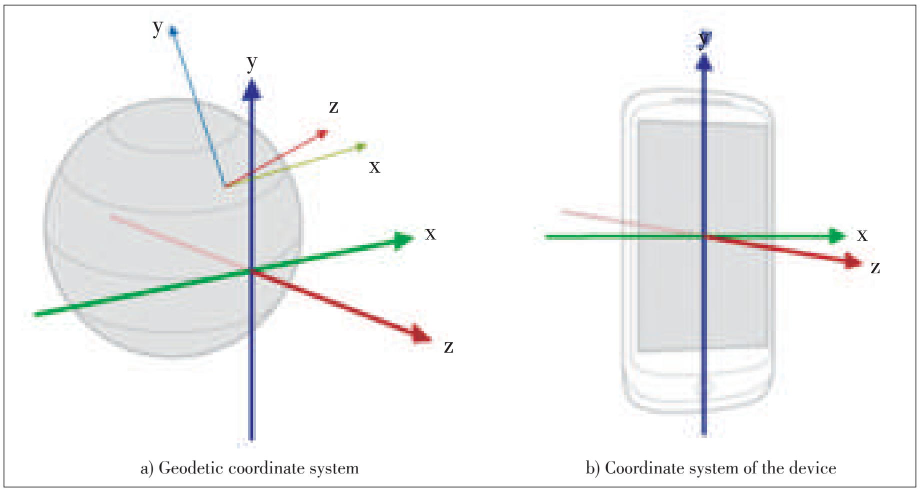
In order to transform the structures from SfM coordinate system to geodetic coordinate system as precisely as possible, we estimate the rotation transformation, scale transformation and tranlation transformation seperately.

3.2.2 Rotation, Scale and Translation Transformation

For each pair of cameras in SfM space and geodetic space, we can directly estimate the rotation transformation:

$$R_{trans} = R_{geo} * R_{SfM}^{-1}, \qquad (1)$$

where R_{geo} and R_{sfM} are the rotation matri-


y
y
z
x
x
z
a) Geodetic coordinate system
y
x
z
b) Coordinate system of the device

▲**Figure 4. Transformation between two coordinate systems.**

ces relative to the SfM coordinate system and geodetic coordinate system respectively. Because we take several images of the target, we can calculate several R_{trans} as the same number of images. We decompose each R_{trans} into three angles in the way $R_{trans}=R_z{*}R_y{*}R_x$ (each rotation matrix can be decomposed into three angles in any combination of R_x, R_y and R_z). Then all the R_x, R_y and R_z will be averaged respectively and the final R_{trans} is reconstructed.

GPS is a global navigation satellite system that provides geolocation in the form of degrees. Because both the scale and translation transformation should be calculated in Euclidean space, we need to convent each camera's GPS location into X_i, Y_i in the 2-D Cartesian coordinate system in the form of meters. Compared to latitude and longitude, X_i and Y_i is a horizontal position representation measured in meter. We leave alone the Z coordinate, thus the transformation equation is given by:

$$\begin{pmatrix} x_0 & 1 & 0 \\ y_0 & 0 & 1 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ x_i & 1 & 0 \\ y_i & 0 & 1 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ x_n & 1 & 0 \\ y_n & 0 & 1 \end{pmatrix} * \begin{pmatrix} S \\ T_x \\ T_y \end{pmatrix} = \begin{pmatrix} X_0 \\ Y_0 \\ \cdot \\ \cdot \\ X_i \\ Y_i \\ \cdot \\ \cdot \\ X_n \\ Y_n \end{pmatrix}, \tag{2}$$

where x_i and y_i are the values of the X and Y coordinates of each camera in SfM space, which have been rotated by the rotation transformation matrices and n is the number of images. S stands for the scale coefficient. T_x and T_y are the translation parameters. QR decomposition (a decomposition of a matrix A into a product $A=QR$ of an orthogonal matrix Q and an upper triangular matrix R) or Singular Value Decomposition (SVD) can easily solve this linear system to get the scale and translation parameters.

3.3 Line Extraction

In order to get the correspondences of the lines which stand for the same contour line in each image, we have to extract the lines from antenna images. There are too many lines of the whole image, but what we only need is several contour lines' parameters of the antenna. Therefore, it is reasonable for the users to select the bounding box of the antenna and perform line extraction on these small pictures.

Line segment detection in images has been extensively studied in computer vision. Traditional methods like Hough transform [11] or its variants [12], [13] cannot satisfy the robustness requirement under different circumstances. We use a latest algorithm named Line Segment Detector (LSD) [14], [15], which is a linear-time segment detector giving subpixel results without parameter tuning.

The LSD algorithm extracts line segments in three steps: 1) partitioning the images into line-support regions by grouping connected pixels that share the same gradient angle up to a certain tolerance; 2) finding the line segment that best approximates each line-support region; 3) validating or not each line segment based on the information in the line-support region. We exact and show the longest ten lines of each bounding box and manually input the ID of the corresponding contour line in the image. Finally the corresponding line data set is denoted as $\{l_0, l_1, \dots l_{n-1}\}$.

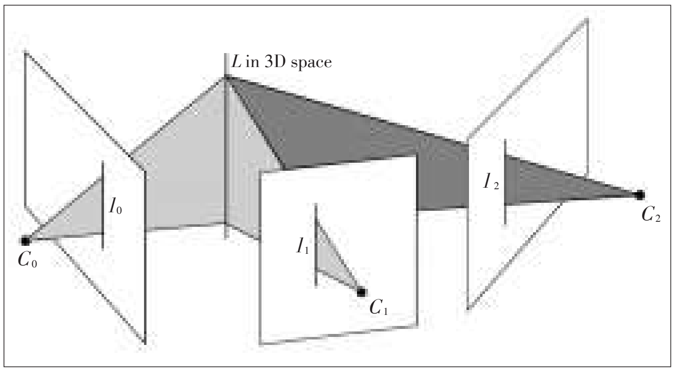
3.4 Line Triangulation and Angle Output

We suppose a set of n corresponding lines are all visible in n perspective images. Our goal is to recover the 3D pose of the antenna with known cameras' parameters and these line correspondences. We take three views of the images for explanation. As shown in **Fig. 5**, the planes back-projected from the lines in each view must all meet in a single line L in space and conversely the 3D line projects to corresponding lines l_0, l_1 and l_2 in these three images. This geometric property can be translated to an algebraic constraint, namely the trifocal tensor [16].

We use the method in [17] to perform line triangulation. The trifocal tensor matrix W is given by:

$$W=\begin{bmatrix} l_0{*}P_0 \\ l_1{*}P_1 \\ l_2{*}P_2 \end{bmatrix}, \tag{3}$$

where P_0, P_1 and P_2 are the projection matrices of these three images. Let $X_a=v(:,3)$ and $X_b=v(:,4)$, where $[u,\ \ s,\ \ v]=SVD(W)$. X_a and X_b can be regarded as two 3D



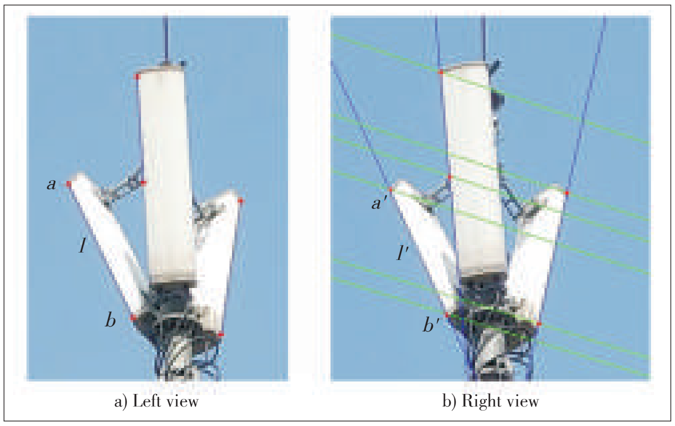
▲Figure 5. The line L in 3D space is triangulated as the corresponding triplet $l_0 \leftrightarrow l_1 \leftrightarrow l_2$ in three views indicated by their camera centers $\{C_0, C_1, C_2\}$ and image planes.

points. After transforming the two points by rotation transformation matrices, the downtilt and heading can be calculated. What's more, there are C_n^3 angles if we choose three arbitrary views of all the images. The final output could be averaged by these angles.

3.5 Point Triangulation and GPS Computation

Line triangulation cannot give the 3D coordinate of the antenna, so we try to triangulate the point correspondences to estimate the GPS and height. As shown in **Fig. 6**, the triangulation of points requires computing the intersection of two known rays in space and the point correspondence $x \leftrightarrow x'$ defines the rays. Fig. 6a shows the theory of epipolar constraint. If the projection point x is known, then the epipolar line l' is known and the point X projects into the right image on a point x' which must lie on this particular epipolar line. It can be formulated by the equation $x'^T Fx = 0$, where F is the fundamental matrix given by the two cameras' parameters.

To get point correspondence, we randomly choose two images to get the known matching line segments $l \leftrightarrow l'$ and the projection matrices of the two views. For the segment l's end point a in **Fig. 7a**, we can calculate the corresponding epipolar line in Fig. 7b. This epipolar line intersects with l' on the point a'. In this way, we can get two point correspondences $a \leftrightarrow a'$ and $b \leftrightarrow b'$, as shown in Fig. 7.

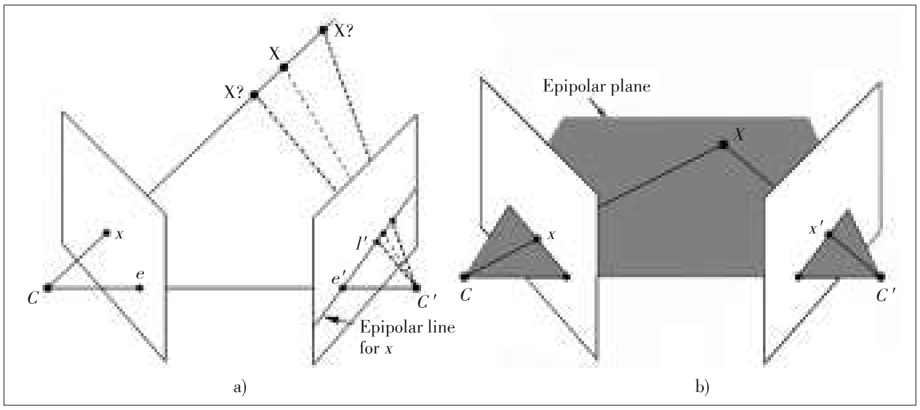


▲Figure 7. The method of computing point correspondence.

Fig. 6b tells the basic principle of point triangulation. There are many algorithms we can adopt for triangulation. We develop the iterative linear method in [18], which is efficient and accurate enough. The 3D coordinate of the antenna is defined as the middle point of the points A and B triangulated in 3D space. We use the scale and translation transformation parameters to transform the 3D coordinate and the GPS location can be calculated by re-projecting the values of meters to degrees. As for height, we assume all the pictures are taken on the same altitude and the height of the antenna can be directly given by the translation transformation.

4 Experiments

We implement the system on a PC and a smartphone. The PC has an Intel(R) Core(TM) i5-4590 3.30 GHz CPU with dual-core processors and 8 GB memory. The smartphone is ZTE A2017 which has a Qualcomm snapdragon 820 2.2 GHz CPU with quad-core processors and the 3GB RAM.



▲Figure 6. a) A ray in 3D space is defined by the first camera center C and x. This ray is imaged as an epipolar line l' in the second view. The point X in 3D space which projects to x must lie on this ray, so the corresponding point x' must lie on l'. b) Triangulation.

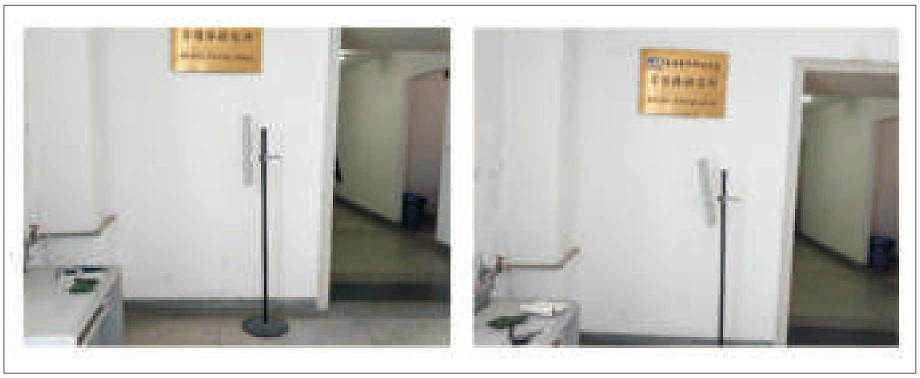
Two representative data sets are used to perform the experiments: an indoor antenna dataset and an outdoor antenna dataset. For each dataset we take 6 images of the antenna target. All the images have the resolution of 4160×3120 ppi. We develop multi-thread programs to speed up the feature extraction procedures, which makes the running time on the PC and smartphone can be within 2 minutes and 3 minutes respectively, including the time used for interaction.

4.1 Indoor Antenna Dataset

We set up an antenna for the experiments and put it in an indoor environment. In order to test the antenna with different poses, we take several images of the antenna with various downtilts from 3° to 12°, as shown in **Fig. 8**.

Only the downtilt and heading of the antenna could be estimated because there is no GPS signal in the indoor environment. On the other hand, we use different photography methods to take images of the antenna by the distance of 4 m. The poses of the smartphone relative to the geodetic coordinate system can be decomposed to three angles. In the proposed photography method, 3d representatives that we fix these three angles of each smartphone to (110°,0°,90°), with the help of a tripod standing; 2d means that only the latter two angles are fixed; 0d means that all the three angles could be different. **Tables 1** and **2** shows the experiment results based on the indoor datasets, where T and H stand for downtilt and heading respectively.

The experiment results based on the indoor antenna dataset show that different photography methods do not have an evident influence on the estimation accuracy. Downtilt errors are within 1° in different results, which are fairly accurate. However, heading errors are larger than the downtilt ones and the absolute error is within 5°. **Fig. 9** shows how the antenna’s downtilt affects the average measuring results, especially for the heading errors. We can clearly find that as the downtilt of the antenna increases, the accuracy of the heading also increases. It is because that the estimation error of the heading is inevitable; when the target is almost vertical to the ground, a minor displacement of the estimated pose will lead to a big error of the heading. In extreme cases, when the target is vertical to the ground, its heading is an almost random value.



▲Figure 8. Indoor antenna images.

▼Table 1. Experiment results based on the indoor antenna dataset: the values of downtilt

True value (°)		0d		2d		3d		Error average (°)
T	H	Result (°)	Error (°)	Result (°)	Error (°)	Result (°)	Error (°)	
3	316	2.824	0.176	2.643	0.357	2.689	0.311	0.281
6	317	5.982	0.018	5.574	0.426	5.774	0.226	0.223
9	318	8.550	0.450	8.489	0.511	8.484	0.516	0.492
12	318	11.975	0.025	12.188	0.188	12.507	0.507	0.240
15	320	14.872	0.129	14.663	0.337	14.157	0.844	0.436

▼Table 2. Experiment results based on the indoor antenna dataset: the values of heading

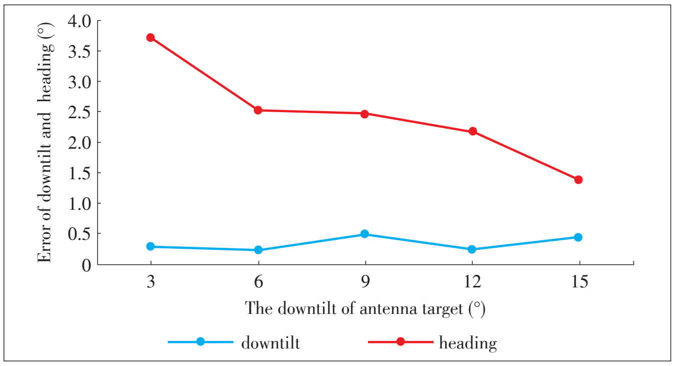
True value (°)		0d		2d		3d		Average error (°)
T	H	Result (°)	Error (°)	Result (°)	Error (°)	Result (°)	Error (°)	
3	316	314.513	1.487	311.253	4.747	311.090	4.910	3.715
6	317	314.542	2.458	314.007	2.993	314.907	2.093	2.514
9	318	314.060	3.940	316.146	1.854	316.404	1.596	2.463
12	318	316.468	1.532	318.755	0.755	322.213	4.213	2.167
15	320	319.239	0.761	319.947	0.053	316.660	3.340	1.385

4.2 Outdoor Antenna Dataset

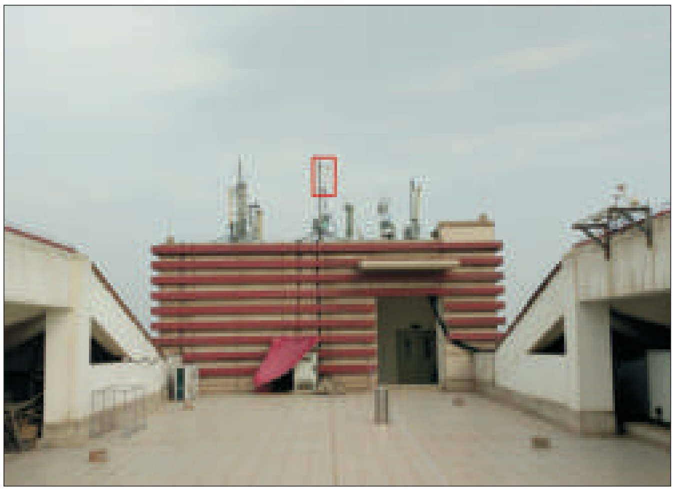
We also perform the experiments in an outdoor environment to testify the valid photographic distances and the stability in different environments. We take one of the environments as an example. The spot for photography is the rooftop of one of ZTE buildings (**Fig. 10**). The red box in the figure is the target antenna.

In this dataset, the photographic distance ranges from 4 m to 30 m. Because it is an outdoor environment, the GPS location and height of the smartphone can be stored and we can analyze the accuracy of these parameters. The antenna’s downtilt and heading is 11° and 180° respectively. The true value of the longitude is 108.827717 and the latitude value is 34.098142. The altitude is 415 m. **Table 3** shows the measuring results.

According to the results of Table 3 and **Fig. 11**, the system always gives an accurate downtilt at any distance within 30 m. Within this shooting distance, the heading error is below 5°. However, when the distance becomes farther, the accuracy of heading gets lower and more unstable too. This is because when the shooting distance is too far, the contour line of the antenna becomes smaller and the error of the line’s parameters is bigger. The GPS is also within 5 m in this photography distance range. However, the accuracy of the height is rather low because of the inaccuracy of the built-in sensors of the smartphone. For example, when we take 5 images for the antenna target on the platform at the same altitude, we find that the 5 altitude values stored by the phone vary a lot and are not consistent with the ground truth. This inaccuracy of the raw data leads to the error of the final altitude of



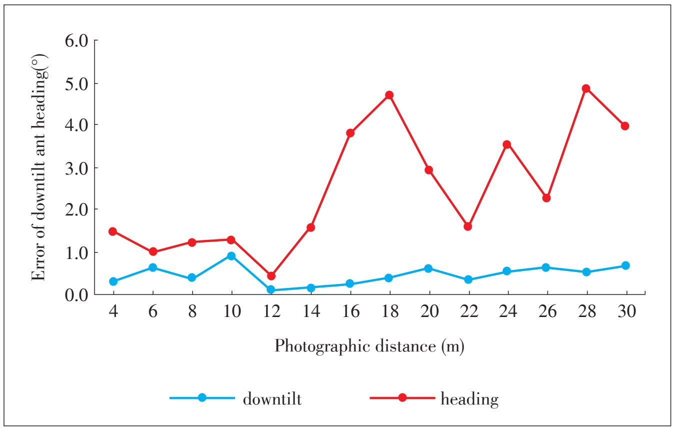
▲Figure 9. The relationship between the downtilt of antenna and average measuring error.



▲Figure 10. The rooftop of one ZTE building.

▼Table 3. Results of the outdoor antenna dataset

Short-distance								
Distance (m)	Downtilt		Heading		Longitude result	Latitude result	GPS error	Altitude
	Result (°)	Error (°)	Result (°)	Error (°)				
4	11.317	0.317	178.509	1.491	108.827724	34.098129	0.805	416.767
6	11.623	0.623	178.993	1.007	108.827728	34.098129	3.173	414.408
8	11.389	0.389	178.774	1.226	108.827733	34.098131	2.423	418.637
10	11.910	0.910	181.284	1.284	108.827735	34.098137	2.587	410.937
12	11.101	0.101	179.571	0.429	108.827734	34.098138	1.957	406.251
14	11.159	0.159	181.577	1.577	108.827746	34.098133	4.506	406.005
16	11.239	0.239	183.794	3.794	108.827755	34.098147	2.587	406.087
18	11.388	0.388	184.699	4.699	108.827742	34.098126	5.106	407.377
20	11.606	0.606	182.926	2.926	108.827710	34.098130	4.717	411.204
22	11.332	0.332	178.411	1.589	108.827735	34.098135	3.958	415.548
24	11.534	0.534	176.466	3.534	108.827697	34.098118	3.518	407.194
26	11.631	0.631	177.724	2.276	108.827716	34.098181	4.120	416.653
28	11.517	0.517	175.148	4.852	108.827734	34.098174	4.356	407.903
30	11.673	0.673	183.964	3.964	108.827747	34.098168	5.170	416.767



▲Figure 11. The relationship between photographic distance and angle measuring error.

the antenna.

We suggest that the photography distance is in the range of 3 m to 25 m and the number of the images should be more than 5 and less than 10 considering both the accuracy and efficiency. The image quality should be good. In particular, the contour lines of the antenna should be distinct and easy for extraction. The moving distance between two shooting spots should be from 0.3 m to 1 m, because too small or too big moving distances will increase the failure risk of Structure from Motion.

5 Conclusions

We propose a photogrammetry-based antenna pose measuring system, which only requires antenna engineers to take several images of the antenna and some easy interaction with the application on the smart phone. The experiment results show that within the distance of less than 30 m, the downtilt error is in the range of 2°. Owing to the physical property of the heading, within the distance of 25 m, its error is in the range of 5°, bigger than the downtilt error. The GPS error is within 5 m when the GPS information is well corrected by satellites after several minutes. The altitude results can just be regarded as a reference because the altitudes captured by the phone are too noisy.

The proposed system presents many advantages. With it, engineers do not have to wear the equipment, climb up the stairs and contact the antenna to measure the pose. What they only need is taking photos, touching the screen and waiting for about 3 minutes, and then the fairly accurate results will be estimated. It is safe, easy, economic and efficient. We believe that the proposed system can work in many measuring circumstances and help save many resources for the industry.

References

[1] 3Z. (2017, Apr. 10). *Antenna alignment tool* [Online]. Available: http://3ztelecom.com/antenna-alignment-tool
[2] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3D," *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, vol. 25, no. 3, pp. 835–846, Jul. 2006. doi: 10.1145/1141911.1141964.
[3] D. G. Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision (ICCV)*, Corfu, Greece, Sept. 1999, pp. 1150–1157. doi:10.1109/ICCV.1999.790410.
[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381– 395, Jun. 1981. doi: 10.1145/358669.358692.
[5] R. Hartley, J. Trumpf, Y. Dai, et al, "Rotation averaging," *International Journal of Computer Vision (IJCV)*, vol. 103, no. 3, pp. 267– 305, Jan. 2013. doi: 10.1007/s11263-012-0601-0.
[6] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, et al, "Global motion estimation from point matches," in *Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, Oct. 2012, Zurich, Switzerland, pp. 81–88. doi: 10.1109/3DIMPVT.2012.46.
[7] M. Brand, M. Antone, and S. Teller, "Spectral solution of large-scale extrinsic camera calibration as a graph embedding problem," in *European Conference on Computer Vision (ECCV)*, Berlin, Germany, Jul. 2004, pp. 262– 273. doi: 10.1007/978-3-540-24671-8_21.
[8] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *International Journal of Computer Vision* (IJCV), vol. 37, no. 2, pp. 151– 172, Jun. 2000. doi: 10.1023/A:1008199403446.
[9] J. Courchay, A. Dalalyan, R. Keriven, et al, "Exploiting loops in the graph of trifocal tensors for calibrating a network of cameras," in *European Conference on Computer Vision (ECCV)*, Heraklion, Greece, Sept. 2010, pp. 85– 99. doi: 10.1007/978-3-642-15552-9_7.
[10] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, no. 2, pp. 164–168, Jul. 1944.
[11] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, Dec. 1981. doi: 10.1016/0031-3203(81)90009-1.
[12] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probabilistic hough transform," *Computer Vision and Image Understanding (CVIU)*, vol. 78, no. 1, pp. 119– 137, Apr. 2000. doi: 10.1006/cviu.1999.0831.
[13] C. Galambos, J. Kittler, and J. Matas, "Gradient based progressive probabilistic Hough transform," *IEEE Proceedings-Vision, Image and Signal Processing*, vol. 148, no. 3, pp. 158–165, Aug. 2002. doi: 10.1049/ip-vis:20010354.
[14] R. G. V. Gioi, J. Jakubowicz, J. M. Morel, et al, "LSD: a fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis & Machine Intelligence (PAMI)*, vol. 32, no. 4, pp. 722– 732, Dec. 2010. doi: 10.1109/TPAMI.2008.300.
[15] R. G. V. Gioi, J. Jakubowicz, J. M. Morel, et al, "LSD: a line segment detector," *Image Processing on Line*, no. 2, pp. 35–55, Mar. 2012. doi: 10.5201/ipol.2012.gjmr-lsd.
[16] A. M. Andrew, "Multiple View Geometry in Computer Vision," *Kybernetes*, vol. 30, no. 9/10, pp. 1333–1341, Dec. 2001. 10.1108/k.2001.30.9_10.1333.2.
[17] D. Matinec and T. Pajdla, "Line reconstruction from many perspective images by factorization," in *IEEE Computer Society Conference on Computer Vision & Pattern Recognition (PAMI)*, Madison, USA, Jul. 2003, pp. 497– 502. doi: 10.1109/CVPR.2003.1211395.
[18] R. I. Hartley and P. Sturm, "Triangulation," in *International Conference on Computer Analysis of Images and Patterns*, Jun. 2001, Warsaw, Poland, pp. 146–157. doi: 10.1007/3-540-60268-2_296.

Manuscript received: 2017-12-31

Biographies

XU Kun (xu.kun7@zte.com.cn) received his master's degree from Xidian University, China. He is now the director of Big Data Department 4, ZTE Wireless Research Institute. From 2009 to 2012, he was the project manager of the UniPOS NetMAX product group at ZTE Corporation. He served as the product director and then the head of the Wireless Network Optimization Tool Department, ZTE Corporation from 2012 to 2014. He has rich experience in wireless network optimization.

FAN Guotian (fan.guotian@zte.com.cn) received his master's degree in engineering in 2008. Now he is a senior product manager at the Wireless Big Data Center of ZTE Corporation. His research interests include big data mining, wireless network planning/optimization, and data GIS positioning.

ZHOU Yi (zhou.yi5@zte.com.cn) received his master's degree in computer system architecture from Xidian University, China in 2008. Now he is a system engineer and project manager at the Wireless Big Data Center of ZTE Corporation. His research interests include big data and machine learning.

ZHAN Haisheng (zhan_haisheng@vip.163.com) received his doctor's degree in computer application technology from Xidian University, China in 2007. He is now an associate professor of the School of Network and Continuing Education, Xidian University. His main research interests include image processing and Chinese semantic processing.

GUO Zongyi (guozongyi75@126.com) received his bacherlor's degree in computer science and technology from the Department of Computer Science, Xidian Universidy, China in 2014. He is currently pursuing the master's degree at Computer Technology in Multimedia Technology Institute, Department of Computer Science, Xidian University. His research interests include image processing, machine vision, photogrammetry, and machine learning.

DOI: 10.19729/j.cnki.1673-5188.2018.04.007
http://kns.cnki.net/kcms/detail/34.1294.TN.20181210.0836.002.html, published online December 10, 2018

Energy Efficiency for NPUSCH in NB-IoT with Guard Band

ZHANG Shuang[1, 2], **ZHANG Ningbo**[1, 2], **and KANG Guixia**[1, 2]

(1. Key Laboratory of Universal Wireless Communications (BUPT), Ministry of Education, Beijing 100876, China;
2. Science and Technology on Information Transmission and Dissemination in Communication Networks Lab, Beijing 100876, China)

Abstract

Narrowband Internet of Things (NB-IoT) has been proposed to support deep coverage (in building) and extended geographic coverage of IoT. In this paper, a power control scheme for maximizing energy efficiency (EE) of narrowband physical uplink shared channel (NPUSCH) with the guard band is proposed. First, we form the optimization problem based on the signal model with the interferences of narrowband physical random access channel (NPRACH) which are caused by the non-orthogonality of NPUSCH and NPRACH. Then, a method of reserving guard bands is proposed to reduce these interferences. Based on it, an efficient iterative power control algorithm is derived to solve the optimization problem, which adopts fractional programming. Numerical simulation results show that NPUSCH with the guard band has better performance in EE than that without the guard band.

Keywords

NB-IoT; energy efficiency; NPUSCH; NPRACH; interference; guard band

1 Introduction

To support some specific scenarios of Internet of Things (IoTs), e.g. deep coverage (in building) and extended geographic coverage, the Narrowband Internet of Things (NB-IoT) was standardized at the Third-Generation Partnership Projects (3GPPs) Radio Access Network Plenary Meeting 69 [1]. This new technology can improve network coverage, support massive number of low throughput devices, and provide low delay sensitivity and high energy efficiency (EE). This technology includes three mode operations: 1) stand-alone as a dedicated carrier, 2) in-band within a normal Long Term Evolution (LTE) carrier, and 3) guard bands within a LTE carrier [1].

The uplink of NB-IoT mainly includes two physical channels: narrowband physical uplink shared channel (NPUSCH) and narrowband physical random access channel (NPRACH). They support different subcarriers, i.e., 15 kHz and 3.75 kHz subcarriers for NPUSCH and only single-tone with 3.75 kHz subcarrier for NPRACH. Once the 15 kHz subcarrier for NPUSCH is deployed close to NPRACH, there will be interference between them due to the non-orthogonal signal. If there is no filters in the receivers of NPUSCH, the symbol of NPRACH will interfere with the subcarriers of NPUSCH after discrete Fourier transform (DFT). Such phenomenon can be regarded as narrow-band interference (NBI) [2], which will increase the transmission power cost of NPUSCH and reduce the EE of NPUSCH [3]. We can also adopt the method of reserving guard bands in LTE to ease the interference from NPRACH and improve EE of NPUSCH. Reserving guard bands is an efficient method to avoid the interference from the non-orthogonal subcarriers, which means reserving an idle subcarrier between NPUSCH and NPRACH. At the same time reserving guard bands also means the loss of frequency spectrum of the NB-IoT system. Therefore, researching on the relationship between the guard band and interference is an important issue for adopting this method.

In the last decade, due to economic, operational, and environmental concerns [4], EE has emerged as a new prominent figure of merit for communication networks design. As a result, the research on EE (bit/Joule) has drawn much attention. In [5], the authors studied the energy-efficient resource scheduling with quality of service (QoS) guarantees in multi-user orthogonal frequency division multiple access network. In [6], an energy-efficient resource control problem which is subject to constraints in service quality requirements, total power, and probabilistic interference was modeled as a chance-constrained programming for multicast cognitive orthogonal frequency division multiplexing (OFDM) network. An energy-efficient resource control problem for device-to-device (D2D) links was studied in [7], which aims to maximize the minimum weighed EE of D2D links while guaranteeing minimum data rates for cellular links. Using a two-level dynamic scheme, energy-efficient resource control and inter-cell interference management were both considered in heterogeneous networks [8]. The authors in [9] investigated the fundamental tradeoff between EE and spectral efficiency for interference-limited wireless networks. All of the works mentioned above focus on EE with considering the constraints of QoS, power, interference, etc. However, to the best of our knowledge, the guard band as a factor that influences EE in some certain scenes has not been

This work was supported by the National Natural Science Foundation of China under Grant No. 61501056, National Science and Technology Major Project of China under Grant No. 2016ZX03001012, and ZTE Industry-Academia-Research Cooperation Funds.

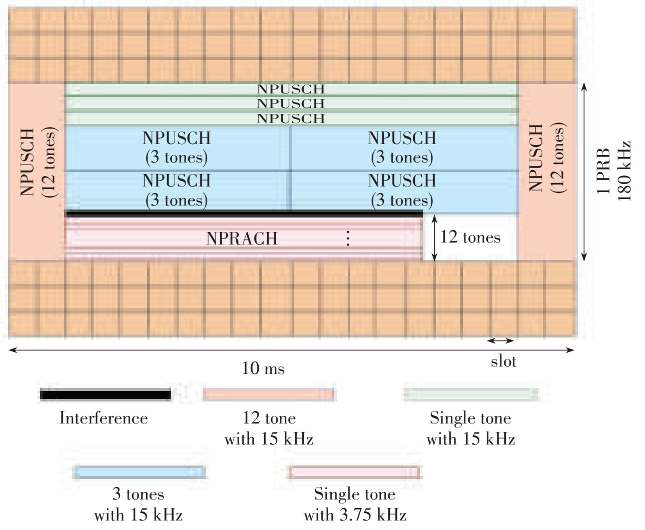
studied yet.

Motivated by the aforementioned observation, we have proposed a power control algorithm optimized for maximizing EE of NPUSCH in NB-IoT with the guard band. The main contributions of this study are as follows: 1) A signal model with NPRACH interference is introduced; 2) A method of reserving guard bands is designed to reduce the interference from NPRACH; 3) Based on the above two points, an energy efficient power control using iterative algorithm with the guard band is proposed.

The rest of the paper is organized as follows. Section 2 describes the system model and focuses on problem formulation. Section 3 introduces a method of reserving guard bands for easing the interference from NPRACH and elaborates the power control algorithm for EE maximization. The simulation results are presented in Section 4, while a conclusion is drawn in Section 5.

2 System Model and Problem Formulation

In this paper, we discuss the situation where the 15 kHz subcarrier for NPUSCH is deployed adjacently to NPRACH. **Fig. 1** is an example for this deployment in the mode of in-band operation which utilizes one resource block as its bandwidth. In the figure, the black long rectangle is the interference from NPRACH to NPUSCH; the rectangles of different colors are used to describe the mapping of the NPUSCH to resource elements, which is called resource unit (Ru) including consecutive Single-Carrier Frequency Division Multiple Access (SC-FDMA) symbols in the time domain and consecutive orthogonal subcarriers in the frequency domain. Different from the PRACH in LTE, NPRACH transmits preambles based on single-subcarrier frequency-hopping symbol groups and the frequency location of its transmission is constrained within 12 subcarriers, i.e., frequency hopping shall be used within the 12 subcarriers [10].



NPRACH: narrowband physical random access channel
NPUSCH: narrowband physical uplink shared channel
PRB: physical resource block

▲Figure 1. Example for NB-IoT design of uplink (in-band).

2.1 Signal Model

Suppose that a NPUSCH contains M consecutive orthogonal subcarriers, while a NPRACH only contains one subcarrier n. Let p_m denote the transmitted power of subcarrier $m \in \{1, \cdots M\}$. The received symbol R_m at the Fast Fourier Transform Algorithm (FFT) output can be described by

$$R_m = X_m H_m \sqrt{p_m} + N_m + I_m, \tag{1}$$

where X_m and H_m are the transmitted symbol and transfer factor of subcarrier m, respectively. N_m is additive while Gaussian noise at $m-$th subcarrier and I_m is the interference from NPRACH on the subcarrier m.

The interference from NPRACH can be regarded as a narrow band interference (NBI) [2], [11]. NPRACH with single tone can be described by

$$x(k) = a \bullet e^{j(2\pi f_c k + \phi)}, \tag{2}$$

where a, f_c and ϕ are the corresponding amplitudes, frequencies and a random phase, respectively. At the receiver, the interferer goes through the N point DFT and appears on the OFDM spectrum when $f_c \in \{m < f < m+1, m = 1, \dots M\}$. The result of this operation on the subcarrier m is described by

$$I_m = \sum_{k=0}^{N-1} x(k) e^{-j2\pi km/N}, k = 0, \dots, N-1. \tag{3}$$

Then, the amplitude spectrum of the interference on each subcarrier m applying a rectangular window is given by

$$I_m = a.e^{j\phi} e^{j(N-1)(\pi f_c - \pi m/N)} \frac{\sin N(\pi f_c - \pi m/N)}{\sin(\pi f_c - \pi m/N)}. \tag{4}$$

The derivation of the equation can be found in Appendix A. The received Signal to Interference plus Noise Ratio (SINR) of the subcarrier m is γ_m:

$$\gamma_m = \frac{p_m |H_m|^2}{|I_m|^2 + \sigma_m^2}, \tag{5}$$

where $|I_m| = \left| a \bullet \frac{\sin N(\pi f_c - \pi m/N)}{\sin(\pi f_c - \pi m/N)} \right|$ and $\sigma_m^2 = E|W_m|^2$.

2.2 Problem Formulation

Using (5), the data rate of the subcarrier m can be written as follow:

$$R_m = \log_2(1 + \gamma_m). \tag{6}$$

Correspondingly, the sum rate is

$$R=\sum_{m=1}^{M}R_m, \tag{7}$$

and the total power consumption of NPUSCH is

$$P=\varepsilon\cdot\sum_{m=1}^{M}p_m+P_0, \tag{8}$$

where the coefficient ε is a constant power-amplifier inefficiency factor of NPUSCH, and P_0 is the circuit power consumption [4].

We formulate the optimization problem as maximizing the EE of one NPUSCH, in which "bit per Joule" is defined as the metric subject to the constraints of the target rate and total transmit power. The problem can be written as:

$$\max \eta_{EE}=\frac{R}{P}, \tag{9a}$$

$$s.t.\ R_m>r_m, \tag{9b}$$

$$\sum_{m=1}^{M}p_m\leqslant P_{\max}, \tag{9c}$$

$$p_m\geqslant 0, \tag{9d}$$

where r_m and $P_{\max}$ are the minimum rate requirements for the subcarrier m and the total transmit power of the NPUSCH, respectively. Here we assume $r_1=r_2...=r_M=r_{0.}$ The constraint (9d) guarantees the feasible sets of p_m .

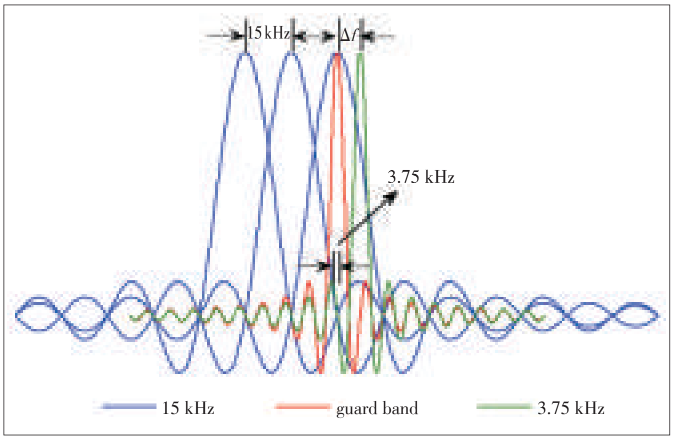
3 Design for Guard Band and Power Control

In this section, we will elaborate the method of reserving guard bands, based on which a power control algorithm is also proposed to solve the optimization problem.

3.1 Guard Band

In LTE, the physical random access channel (PRACH) transmits preambles generated from Zadoff-Chu sequences using a certain amount of subcarriers. By designing the number of subcarriers, one physical uplink shared channel (PUSCH) is set respectively at the two edges of PRACH as guard bands to avoid the interference between PUSCH and PRACH [10]. We can adopt the method of reserving guard bands to reduce the inference from NPRACH, like the method for LTE.

In this paper, the size of a guard band is depend on the interference from NPRACH. From (3), we know that the signal transmitted on NPRACH appears on the OFDM spectrum of the subcarrier m of NPUSCH due to their overlapped frequency band. **Fig. 2** is an example of the guard band between NPRACH and NPUSCH. In the figure, the three blue ones are the subcarriers of NPUSCH, i.e., $M=3$; the red waveform represents the subcarrier of NPRACH, i.e., $n=1$; the green one denotes the subcarrier n when the guard band Δf is deployed between the NPUSCH and NPRACH. As we can see, before reserving the guard band, the crest of red waveform overlaps the blue ones, i.e., there is interference between NPUSCH and NPRACH due to the non-orthogonal signal. Compared with the primary NPRACH, reserving Δf means the center frequency of the subcarrier n is varied from the red one to the green one with Δf . As we mentioned above, NPRACH only supports single tone and (2) can be rewritten with Δf as follow



▲ **Figure 2. Guard band for the narrowband physical random access channel (NPRACH) and narrowband physical uplink shared channel (NPUSCH).**

$$x(k)=a\cdot e^{j(2\pi(f_c+\Delta f/f_s)k+\phi)}. \tag{10}$$

When $\left(f_c+\Delta f/f_s\right)\in\left\{f\middle|m<f<m+1,m=1,...,M\right\}$, the interference on each subcarrier m can be written by

$$I'_m=a\cdot e^{j\phi}e^{j(N-1)(\pi m/N)}\frac{\sin N(\pi f'-\pi m/N)}{\sin(\pi f'-\pi m/N)}, \tag{11}$$

where $f'=f_c+\Delta f/f_s$ and f_s is the sample rate. Then, SINR with guard band is

$$\gamma'_m=\frac{p_m\left|H_m\right|^2}{\left|I'_m\right|^2+\sigma_m^2}. \tag{12}$$

As defined in (6), we will get a new data rate of the subcarrier m and sum rate R' with guard band, i.e., we can also form the EE problem with η'_{EE} as defined in (9), which can be solved by power control.

3.2 Power Control for EE

Fractional programming [12] and Convex (CVX) will be used for problem transformation and obtaining optimal power, respectively.

We first change the fractional formula (9) with η'_{EE} and

$R^{'}(\mathbf{P})$ into a compact form

$$\max_{P\in\Omega} \eta^{'}_{EE} = \frac{R^{'}(\mathbf{P})}{P(\mathbf{P})}, \quad s.t.\ R^{'}_{m} > r_0,\ (9c),\ (9d), \tag{13}$$

where $\mathbf{P}$ is a $1\times M$ matrices with elements p_m, $R^{'}(\mathbf{P})$ and $P(\mathbf{P})$ are the numerator and denominator in $(9a)$ with $\eta^{'}_{EE}$, respectively. While Ω is the feasible domain defined by the constraints in (13).

Moreover, (13) can be written in the following form:

$$\max_{P\in\Omega} \left\{R^{'}(\mathbf{P}) - \eta^{'}_{EE}\cdot P(\mathbf{P})\right\}, \quad s.t.\ R^{'}_{m} > r_0,\ (9c),\ (9d). \tag{14}$$

Many previous works [8], [11] have proven the following equivalence

$$\max_{P\in\Omega}\left\{R^{'}(\mathbf{P}) - \eta^{'}_{EE^*}\cdot P(\mathbf{P})\right\} = R^{'}\left(\mathbf{P}^{*}\right) - \eta^{'}_{EE^*}\cdot P(\mathbf{P}^{*}) = 0, \tag{15}$$

where $\eta^{'}_{EE^*}$ and $\mathbf{P}^{*}$ are the optimal value and the optimal solution of (14), respectively. That means if and only if (15) is satisfied, the maximum EE $\eta^{'}_{EE^*}$ can be achieved.

$T(\eta^{'}_{EE}) = \max\left\{R^{'}(\mathbf{P}) - \eta^{'}_{EE}\cdot P(\mathbf{P})\right\}$ is strictly monotonic decreasing and continuously proved in [11]. Besides, $R^{'}(\mathbf{P})$ and $P(\mathbf{P})$ are concave function and convex function, which are judged by their second-order conditions. With all these properties, an iterative algorithm can be designed to get $\eta^{'}_{EE^*}$, which is described in **Algorithm 1**. In the algorithm, $\mathbf{P}^{(t)}$ and $\eta^{'(t)}_{EE}$ are the t-th iteration value of $\mathbf{P}$ and $\eta^{'}_{EE^*}$, respectively. The optimization problem is concluded in the following

$$T(\eta^{'(t)}_{EE}) = \max_{P\in\Omega}\left\{R^{'}(\mathbf{P}) - \eta^{'(t)}_{EE}\cdot P(\mathbf{P})\right\}. \tag{16}$$

Algorithm 1: Energy-efficient power scheduling with guard band (EP-GD)

Initialization:

Set any feasible $\mathbf{P}^{(0)}$ with $R^{'}(\mathbf{P}^{(0)})/P(\mathbf{P}^{(0)})$, maximum iteration $T_{\max}$ and maximum tolerance $\tau > 0$.

```
1: $\eta^{'(t)}_{EE} = R^{'}(\mathbf{P}^{(0)})/P(\mathbf{P}^{(0)})$ proceed to 2 with $t = 1$ .
2: CVX to obtain $\mathbf{P}^{(t)}$ by solving equation (16).
3: **if** $T(\eta^{'(t)}_{EE^{'}}) \leqslant \tau$ then
4: $\eta^{'}_{EE^*} = \eta^{'(t)}_{EE^{'}}$ Break.
5: **else if** $T(\eta^{'(t)}_{EE^{'}}) > \tau$ then
6: Evaluate $\eta^{'(t)}_{EE} = R^{'}(\mathbf{P}^{(0)})/P(\mathbf{P}^{(0)})$ and go to 3 replacing $\eta^{'(t)}_{EE}$ by $\eta^{'(t+1)}_{EE}$, $t = t + 1$ .
7: **end if**
```

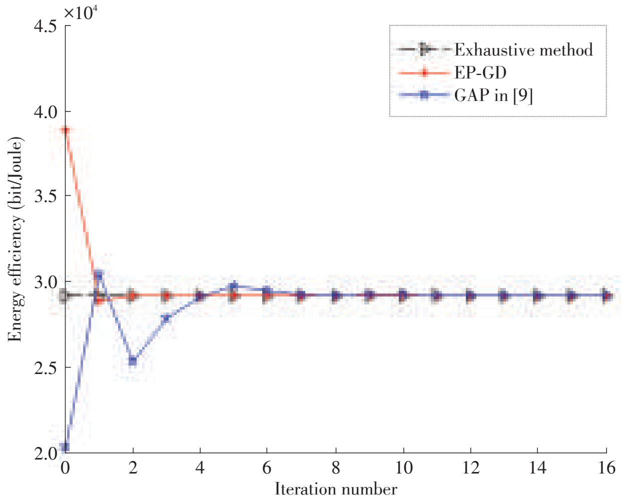
The convergence of EP-GD has been proven in Appendix B.

4 Numerical Simulation

This section presents several numerical results to evaluate EE of NPUSCH using power control with the guard band. The total number of subcarriers for NPUSCH and points for DFT are set 3 and 4, respectively. The amplitude value and phases for NPRACH are set 1 and 0, respectively. Without loss of generality, we assume the value of sample rate is $1/(15\ \text{kHz}\times 4)$. The channel gains for subcarriers of NPUSCH are assumed to be rayleigh fading and noise power $\sigma_m^2 = 0.1\,\mu W$.

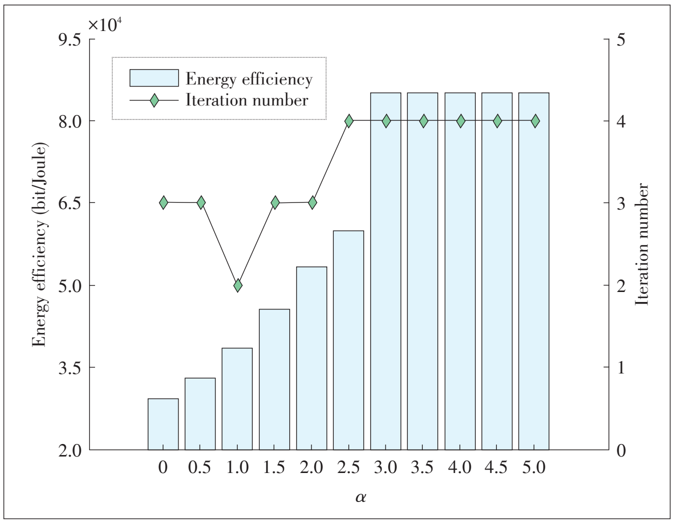
Fig. 3 verifies the correctness of EP-GD without guard band by exhaustive method, which is denoted by black line. It was also compared with the general algorithm procedure (GAP) [9] which is denoted by blue line. In this comparison, $P_{\max} = 0.5$ mW , $r_0 = 0.5$ bit/s/Hz , $P^{(0)} = [0.3\ 0.2\ 0.3]\times P_{\max}$, and $P_0 = 0.1$ mW . From Fig. 3, we can see EP-GD converges to the exhaustive method which can be regarded as the theoretical optimality after a few iterations, so does GAP which adopts a bisection method. The overlap line also demonstrates the convergence of EP-GD. Moreover, approaching to the exhaustive method after a few iterations, EP-GD has better performance than GAP.

Fig. 4 gives the relationship between EE and Δf with an iteration number. Since NB-IoT supports the single-tone with 3.75 kHz subcarrier and we can reserve 3.75 kHz subcarrier to avoid the interference, we define $\Delta f = \alpha\cdot 3.75$ kHz . We then choose different values of α to show the variation of EE. As the blue bar in Fig. 4 shows, EE increases greatly at the beginning, but then it holds steady. The reason is that more Δf will result in less interference, but when Δf is large enough, the frequency bands of NPRACH and NPUSCH will not overlap,



EP-GD: energy-efficient power scheduling with guard band
GAP: general algorithm procedure

▲**Figure 3. Comparison of energy efficiency of EP-GD, the exhaustive method, and GAP.**



▲**Figure 4. Energy efficiency of different Δf with the iteration number.**

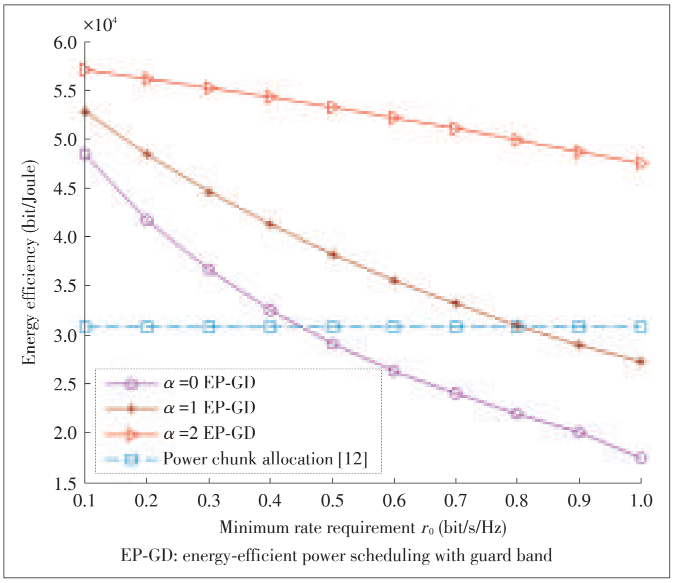
i.e., the interference is inexistence. The black line in Fig. 4 represents the iteration number which still holds a lower level when Δf increases.

Limited by space, we choose three values of Δf that is defined as $\Delta f = \alpha \cdot 3.75$ kHz and illustrate their variations under different r_0 in **Fig. 5**. For comparison, we bring our interference expression into the power chunk allocation in [12] which obtained the optimal sum throughput. We then calculate the EE with the same simulation parameters. As Fig. 5 shows, the EE of our EP-GD decreases monotonically as r_0 increases on the whole trend. The reason is that the larger r_0 is, the more power subcarrier m needs, which results in EE performance degradation. Particularly, when $\alpha = 0$, EE decreases more quickly than the other two cases. The reason is that no guard band means more interference and more power is allocated to subcarriers to satisfy their QoS requirements. This implies that if we need higher data rate communications, we can choose more guard bands to reduce the interference from NPRACH. Compared with the power chunk allocation, EP-GD has a better performance at the beginning, but it descends obviously after $r_0 = 0.4$ bit/s/Hz, when $\alpha = 0$. However, this situation will be improved when $\alpha = 1$, and the descended point will be pushed to $r_0 = 0.8$ bit/s/Hz. EE of EP-GD will greatly increase and outperform the power chunk allocation for any minimum data rate when $\alpha = 0$. In summary, EP-GD has a higher EE at lower data rate communications, which is suitable for NB-IoT. Moreover, higher data rate communications will be realized at the cost of bandwidth.

Fig. 6 shows the effect of $P_{\max}$ on EE, and we also choose three values of Δf to see the variations of EE. It can be seen EE increases greatly with more guard bands in the same $P_{\max}$. The beginning of EE with different Δf is also worth observing, which can be explained that the more guard bands are chosen, the less interference will be suffered and the least $P_{\max}$ can be obtained. Besides, when $P_{\max}$ is large enough, EE approaches a constant value since the algorithm will not consume more power. The power chunk allocation outperforms the proposed EP-GD at the beginning when $\alpha = 1$ and $\alpha = 2$. Then, EE of the power chunk allocation has a sharp descent with the increasing of $P_{\max}$, while EP-GD holds a higher steady level after a modest rise. At last, EP-GD outperforms the power chunk allocation after two intersections.

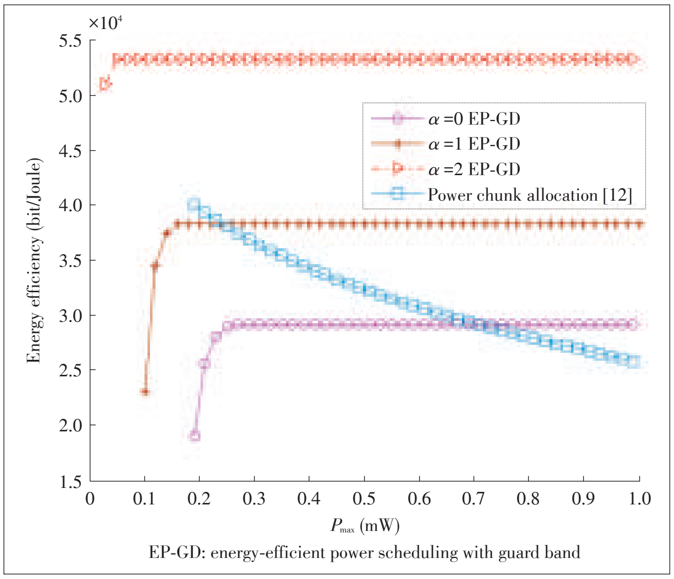
5 Conclusions

In this paper, based on the signal model with NPRACH interference, we formulate the EE of NPUSCH in NB-IoT as an optimization problem, in which the circuit power consumption



EP-GD: energy-efficient power scheduling with guard band

▲**Figure 5. Energy efficiency vs. minimum rate requirement r_0 under different Δf.**



EP-GD: energy-efficient power scheduling with guard band

▲**Figure 6. Energy efficiency of different Δf under different $P_{\max}$.**

and minimum data rate requirement were taken into consideration. A method of reserving guard bands is proposed to reduce the interference from NPRACH, based on which an efficient iterative power control algorithm is derived for maximization of the optimization issue. The simulation results show that the asymptotically optimal power solutions could be obtained after a few iterations by using the proposed algorithm. Moreover, we find EP-GD has a higher EE for lower data rate communications, which is suitable for NB-IoT. Higher data rate communications will be realized at the cost of bandwidth. Since reserving the guard band means losing the frequency spectrum, and the relation between EE and spectral efficiency (SE) of the whole NB-IoT is our further study.

Appendix A

From (3),

$$I_m = \sum_{k=0}^{N-1} x(k) e^{-j2\pi km/N} =$$

$$a \cdot e^{j\phi} \sum_{k=0}^{N-1} e^{2\pi jk(f_c - m/N)} =$$

$$a \cdot e^{j\phi} \frac{1 - e^{2\pi jk(f_c - m/N)}}{1 - e^{j(2\pi f_c - 2\pi m/N)}} =$$

$$a \cdot e^{j\phi} \frac{1 - \cos(2\pi N(f_c - m/N) - \sin(2\pi N(f_c - m/N))}{1 - \cos(2\pi(f_c - m/N)) - \sin(2\pi(f_c - m/N))} =$$

$$a \cdot e^{j\phi} e^{j(N-1)(\pi f_c - \pi m/N)} \frac{\sin N(\pi f_c - \pi m/N)}{\sin(\pi f_c - \pi m/N)}. \tag{17}$$

Appendix B

We follow a similar approach with [11] for proving the convergence of the proposed algorithm. The proof is divided into two steps:

Step 1: Prove $\eta_{EE}^{'(t+1)} > \eta_{EE}^{'(t)}$ for all t with $T(\eta_{EE}^{'(t)}) > \tau$.

Let $P^{(t)}$ be an arbitrary feasible solution and $\eta_{EE}^{'(t)} = R^{'}(P^{(t)}) / P(P^{(t)})$, and then $T(\eta_{EE}^{'(t+1)}) = \max\left\{R^{'}(P) - \eta_{EE}^{'(t)} P(P)\right\} \geqslant R^{'}(P) - \eta_{EE}^{'(t)} P(P) = 0$. By the definition in the algorithm, we have $\eta_{EE}^{'(t+1)} = R^{'}(P^{(t)}) / P(P^{(t)})$, hence $T(\eta_{EE}^{'(t)}) = R^{'}(P^{(t)}) - \eta_{EE}^{'(t+1)} P(P^{(t)}) - \eta_{EE}^{'(t)} P(P^{(t)})$, since $P(P^{(t)}) > 0, T(\eta_{EE}^{'(t)}) > 0$, and $\eta_{EE}^{'(t+1)} > \eta_{EE}^{'(t)}$.

Step 2: $\lim_{x \to \infty} \eta_{EE}^{'(t)} = \eta_{EE^*}^{'}$

As we mentioned above, $T(\eta_{EE}^{'}) = \max_{P \in \Omega}\left\{R^{'}(P) - \eta_{EE}^{'} \cdot P(P)\right\}$ is strictly monotonic decreasing and $\max_{P \in \Omega}\left\{R^{'}(P) - \eta_{EE^*}^{'} \cdot P(P)\right\} = R^{'}(P^*) - \eta_{EE^*}^{'} \cdot P(P^*) = 0$, i.e., if $T(\eta_{EE}^{'(t+1)}) < T(\eta_{EE}^{'(t)})$, combined with Step 1, we have $T(\eta_{EE}^{'(t)}) < T(\eta_{EE^*}^{'})$. The elementary function and their four operations are continuous on the domain of $\eta_{EE}^{'(t)}$, and we then have $\lim_{x \to \infty} \eta_{EE}^{'(t)} = \eta_{EE^*}^{'}$.

References

[1] 3GPP, "Cellular system support for ultra-low complexity and low throughput internet of things (CIoT)," 3GPP TR45.820, Aug. 2015.

[2] D. Galda and H. Rohling, "Narrow band interference reduction in OFDM based power line communication systems," in *Proc. IEEE International Symposium on Power Line Communications and Its Applications (ISPLC)*, Malmö, Sweden, Apr. 2001, pp. 345–351. doi: 10.1109/ISPLC.2016.7476279.

[3] R. Mahapatra, Y. Nijsureet, G. Kaddoum, et al., "Energy efficiency tradeoff mechanism towards wireless green communication: a survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 686–705, 1st Quart. 2016. doi: 10.1109/COMST.2015.2490540.

[4] S. Buuzzi, I. Chih-Lin, E. Klein, et al., "A survey of energy-efficient techniques for 5G networks and challenges ahead," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 4, pp. 697–709, Apr. 2016. doi: 10.1109/JSAC.2016.2550338.

[5] X. Xiao, M. Tao, and J. Lu, "QoS-aware energy-efficient radio resource scheduling in multi-user OFDMA systems," *IEEE Communications Letters*, vol. 51, no. 6, pp. 86–93, Jun. 2013. doi: 10.1109/LCOMM.2012.112012.121910.

[6] L. Xu and A. Nallanathan, "Energy-efficient chance-constrained resource allocation for multicast cognitive OFDM network," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1298–1306, May 2016. doi: 10.1109/JSAC.2016.2520180.

[7] T. D. Hoang and L. B. Le, "Energy-efficient resource allocation for D2D communication in cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 6972–6986, Sept. 2016. doi: 10.1109/TVT.2015.2482388.

[8] A. Y. Al-Zahrani and F. R. Yu, "An energy-efficient resource allocation and interference management scheme in green heterogeneous networks using game theory," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 7, pp. 5384–5396, Jul. 2016. doi: 10.1109/TVT.2015.2464322.

[9] Y. Z. Li, M. Sheng, and C. G. Yang, "Energy efficiency and spectral efficiency tradeoff in interference-limited wireless networks," *IEEE Communications Letters*, vol. 17, no. 10, pp. 1924–1927, Oct. 2013. doi: 10.1109/LCOMM.2013.082613.131286.

[10] 3GPP, "Physical channels and modulation," 3GPP TR 36.211, V12.7.0, Jun. 2016.

[11] T. Shongwe, V. Papilaya, and A. Vinck, "Narrow-band interference model for OFDM systems for powerline communications," *in Proc. IEEE International Symposium on Power Line Communications and Its Applications (ISPLC)*, Johannesburg, South Africa, Mar. 2013, pp. 268–272. doi: 10.1109/ISPLC.2013.6525862.

[12] W. Dinkelbach, "On nonlinear fractional programming," *Management Science*, vol. 13, no. 7, pp. 492–498, Mar. 1967. doi: 10.1287/mnsc.13.7.492.

Manuscript received: 2017-11-27

Biographies

ZHANG Shuang (zhangshuang2015@bupt.edu.cn) received the M.S. degree from Hebei Normal University, China. She is a Ph.D. candidate at Beijing University of Posts and Telecommunications, China. Her research interests include heterogeneous networks, energy efficient transmission and non-orthogonal multiple access.

ZHANG Ningbo (nbzhang@bupt.edu.cn) received the Ph.D. degrees at Beijing University of Posts and Telecommunications (BUPT), China in 2010. He worked in Huawei Technology from 2010 to 2014. He is currently an assistant professor of BUPT. His major research interests include wireless communication theory, machine to machine communications, multiple access, cognitive radio, and signal processing.

KANG Guixia (gxkang@bupt.edu.cn) is a professor and Ph.D. tutor at Beijing University of Posts and Telecommunications, China. She is currently the group leader of the Ubiquitous Healthcare Working Group of China's Ubiquitous Network Technologies and Development Forum, the executive vice director of E-Health Professional Commission of China Apparatus and Instrument Association, one of the four domestic smart medicine specialists of China Smart City Forum, and a member of ITU-R 8F-China Working Group. Her research interests include wireless transmission technologies in PHY and MAC layers. Besides, she is the founder of wireless eHealth and works on its technology, standardization and application.

DOI: 10.19729/j.cnki.1673-5188.2018.04.008
http://kns.cnki.net/kcms/detail/34.1294.TN.20181206.0754.002.html, published online December 6, 2018

Portable Atmospheric Transfer of Microwave Signal Using Diode Laser with Timing Fluctuation Suppression

CHEN Shijun [1], BAI Qingsong [2], CHEN Dawei [1], SUN Fuyu [2], and HOU Dong [2]
(1. ZTE Corporation, Shenzhen 518057, China;
2. University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract

We demonstrate an atmospheric transfer of microwave signal over a 120 m outdoor free-space link using a compact diode laser with a timing fluctuation suppression technique. Timing fluctuation and Allan Deviation are both measured to characterize the instability of transferred frequency incurred during the transfer process. By transferring a 100 MHz microwave signal within 4500 s, the total root-mean-square (RMS) timing fluctuation was measured to be about 6 ps, with a fractional frequency instability on the order of 1 × 10—12 at 1 s, and order of 7 × 10—15 at 1000 s. This portable atmospheric frequency transfer scheme with timing fluctuation suppression can be used to distribute an atomic clock-based frequency over a free-space link.

Keywords

atmospheric communication; frequency transfer; diode laser; timing fluctuation suppression

1 Introduction

Timing and frequency transfer is important to precision scientific and engineering applications, such as frequency standards, optical communication, radar, and navigation [1]–[4]. Over the past decades, many studies of highly stable frequency distribution were focused on the transfer technique via fiber link [5]–[10]. Recently, timing and frequency transfer based on free-space links has begun to attract a remarkable attention as it can provide higher flexibility than fiber links [11]. This free-space frequency transfer can benefit the application for high-fidelity optical links in the future space-terrestrial networks [12] and alternative navigating schemes independent of the global positioning system [13]. In the last few years, there have been several important works in free-space transfer of optical and microwave frequency information. Sprenger et al. studied the frequency transmission of both optical-frequency and radio-frequency (RF) clock signals over 100 m atmospheric link using a continuous wave (CW) laser [14]. Gollapalli and Duan used a pulsed laser to achieve an atmospheric transfer of both RF and optical clock signals over 60 m free-space link [15], [16]. With two cavity stabilized optical frequency combs (OFC), Giorgetta et al. demonstrated an optical time-frequency transfer over 2 km free-space link via two-way exchange between the coherent OFCs with the femtosecond-level resolution of [17]. Furthermore, they improved their experimental setup and achieved a highly precision timing-frequency transfer over a 10 km free-space link in a city environment [18]. Recently, Kang et al. reported a technique of timing jitter suppression for indoor atmospheric frequency comb transfer, which achieved a few femtoseconds timing fluctuation [19].

Although these experiments have demonstrated that the current atmospheric frequency transfers achieved synchronizations between two sites over free-space links, some of them did not suppress the timing fluctuations affected by air turbulence [14]–[16]. In this case, the extra timing fluctuation limits the applications of laser-based atmospheric frequency transfer in areas where sub-picosecond synchronization systems should be constructed. However, the experimental systems for suppressing the timing fluctuations were not concise and robust. For example, the two-way time and frequency transfer (TWTFT) technique used two cavity-stabilized frequency comb to bi-directionally transfer timing signals, which increased the difficulty of some portable applications [17], [18]. The balanced optical cross-correlators (BOC) technique [19] used a crystal to generate optical harmonics, which could result in the difficulty of collimation and focus in the outdoor use. Therefore, it is a big challenge to build a simple and portable sub-picosecond frequency transfer system in outdoor environment.

In this paper, we demonstrate an outdoor atmospheric transfer of microwave signals over free-space link using a compact diode laser with a timing fluctuation suppression technique.

2 Schematic of Timing Fluctuation Suppression in Frequency Transfer

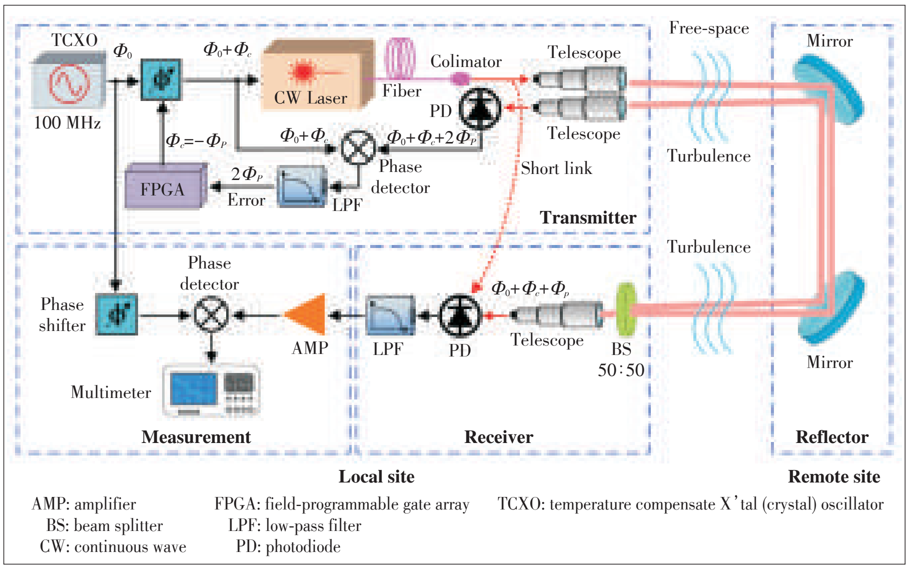
In order to transfer a microwave signal from a transmitter to a receiver via an optical carrier, the most convenient scheme has three steps: directly loading the microwave signal onto the

This work was supported by ZTE Industry-Academia-Research Cooperation Funds, the National Natural Science Foundation of China under Grant Nos. 61871084 and 61601084, the National Key Research and Development Program of China under Grant No. 2016YFB0502003, and the State Key Laboratory of Advanced Optical Communication Systems and Networks, China.

optical carrier; transferring the light to remote receiver via an optical link; recovering this microwave signal with a photo-detection scheme [20]–[22]. However, in an actual frequency transfer system, the three steps introduce excess phase noise or timing jitter into the signal, and result in the degradation of stability of the original microwave signal. On the transmitter, the intense noise of the optical light, primarily introduced by the instability of the laser's current driver, determines the quality of the local modulated optical carrier. Over the transmission link, the air turbulence and temperature drift may introduce excess strong timing jitter and drift into the transmitted optical light in the atmosphere. On the receiver, the photodetection electronics, including photodiode, amplifier, and filter, may also introduce electronic excess noise in the recovered signal. Altogether, these noise sources contribute to the total timing fluctuations in the microwave transmission system. For an atmospheric microwave transfer over a long distance transmission link, the timing fluctuations and frequency instability are mainly caused by the air turbulence and temperature drift. Literatures reveal that the air turbulence induces fluctuations of the refractive index [23], [24], which could directly lead to excess phase noise in the transmitted frequency signal. Therefore, in order to improve the stability of the frequency transfer significantly, the timing fluctuation should be suppressed by a phase compensation technique. Here, based on the three-step scheme mentioned above, we use an upgraded atmospheric frequency transfer with phase compensation.

Fig. 1 shows the schematic of the frequency transfer with a timing fluctuation suppression technique. At the local site, an optical carrier is provided by a commercial distributed feedback laser (DFB) diode which has a center wavelength of 1550 nm. A 100 MHz microwave signal generated from a temperature compensate X'tal oscillator (TCXO) is phase shifted first, and is then loaded onto the optical carrier via a direct current amplitude modulation (AM) scheme. The modulated laser beam is coupled to a telescope via an optical collimator, and is then launched into the free-space transmission link. At the remote site, the beam is sent back to the receiver by a golden-coated reflector. At the receiver, half of the transmitted beam is reflected to the transmitter along the same optical path by a 50:50 beam splitter. The reflected beam is collected by another telescope and is tightly focused onto a high-speed photodiode at the transmitter. The detected microwave signal with twice timing fluctuations is amplified and then mixed with the phase-shifted microwave signal to generate a phase error signal. This error signal is filtered to eliminate the harmonic components and fed back to a digital field-programmable gate array (FPGA) processor, to calculate out the one-way timing fluctuation. Based on the calculation, the FPGA processor produces a controlling signal to adjust the phase shifter, to compensate the timing fluctuation (Fig. 1). At the receiver, with another high-speed photodiode, the remaining beam is also converted to a microwave signal which is used to compare with the RF reference source, to estimating the performance of the frequency transmission.

The mechanism of the timing suppression is explained below. As shown in Fig. 1, we assume that the microwave signal has an initial phase. At the transmitter, this signal is phase-shifted with Φ_c, and then delivered to the receiver over a free-space transmission link. Assuming the air turbulence introduces a phase fluctuation Φ_p to the transmitted signal over the one-trip free-space link, the total phase delay of the recovered microwave signal at the receiver is given by $\Phi_{\text{total}}=\Phi_0+\Phi_c+\Phi_p$. With a 50:50 beam splitter, half of beam is reflected to the transmitter along the almost same optical path, which will introduce a same turbulence-affected phase fluctuation Φ_p. In this case, the round-trip returned microwave signal detected by the photodiode has a phase delay $\Phi_0+\Phi_c+2\Phi_p$ due to the twice turbulence effect. After eliminating the fixed phase $\Phi_0+\Phi_c$ by comparing to the phase-shifted reference signal, we get an error signal with the phase error information $2\Phi_p$. Here, this error signal is fed back to the FPGA processor to calculate out the one-way timing fluctuations Φ_p, and the processor also produces a controlling signal to adjust the phase shifter as the following equation $\Phi_c=-\Phi_p$, to compensate the turbulence-affected phase fluctuation. When the FPGA is active, the phase error Φ_p will be compensated, and consequently, the timing fluctuation affected by air turbulence for the recovered microwave signal on the receiver will be corrected. Based on the schematic and analysis of the timing fluctuation suppression above, we built up an actual atmospheric frequency transfer experiment,



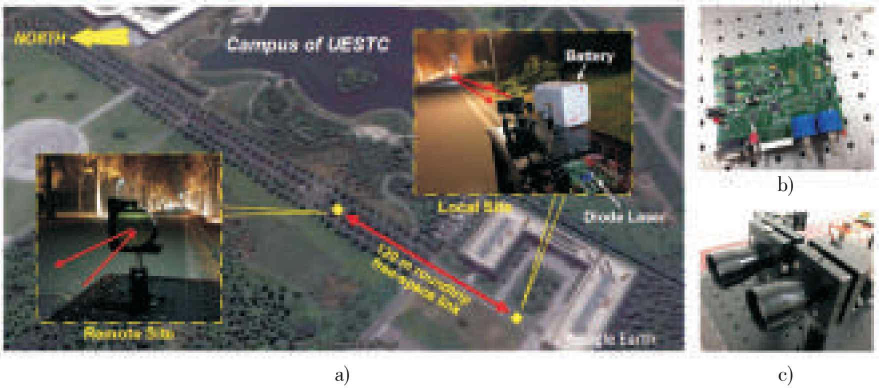
▲**Figure 1. Schematic of atmospheric frequency transfer with timing fluctuation suppression.**

which is shown in **Fig. 2**.

3 Experimental Setup of Frequency Transfer

The atmospheric frequency transmission link was located on a long avenue in the campus of the University of Electronic Science and Technology of China (UESTC) (Fig. 2a). The local site included a transmitter and a receiver, and the remote site included a mirror as beam reflector. The distance between local and remote sites was 60 m. On the local site, a modulated laser beam generated from a diode laser with 1550 nm center wavelength, 3 MHz linewidth, and 18 mW output power, was launched from the transmitter via a 1550 nm AR-coated telescope, and the beam size over free-space was about 20 mm. On the remote site, a golden-coated 2 inch mirror was mounted on a sturdy mount which was anchored on a platform along the avenue. The beam was sent back to the receiver's telescope on the local site, and collected by a fast photodetector to recovery a microwave signal. Here, the telescopes on local site were anchored on another platform along the same avenue (Fig. 2c). The forward and backward transfer formed a total 120 m roundtrip transmission link. Note that, our experimental setup was in an open-air environment and far from the laboratory rooms. This was attributed to a UPS-based battery which supported all electronic components in our system.

Our experiment was conducted in our campus at a normal night. In this experiment, we measured the timing fluctuations and frequency instability of the transferred microwave signal caused by air turbulence. In this case, a 100 MHz microwave signal with a power of 20 mW generated from the TCXO was loaded onto the DFB laser. In our experiment, we launched the laser beam with 18 mW output power, and detected 2 mW round-trip returned beam on the transmitter's photodetector. The great optical power loss is mainly due to the bad air quality in our city. Here, the photo-detection of the retro-reflected signal can introduce additional phase-error due to limited optical power as well as photo-detection nonlinearity. To minimize the residual timing error, the beam must be focused on the center of the photodiode (PD)'s detection area to obtain the best signal to noise ratio (SNR). In addition, the collected 2 mW optical beam is enough to produce the electronic signal for the next stage. In this case, the residual timing error can be ignored since it is far less than the timing fluctuation affected by air turbulence. We extracted and amplified the round-trip returned microwave signal by a band-pass filter and RF amplifier, to obtain a 7 dBm microwave signal. This signal was compared to the local reference signal to produce an error signal. By sending the error signal into the FPGA processor, a controlling signal was produced to drive the phase shifter, so that the timing fluctuation caused by the air turbulence could be compensated. In this servo loop, the compensation bandwidth of the FPGA processor is about 10 kHz. Therefore, we believe the most of fluctuations affected by air turbulence can be suppressed in this bandwidth. To evaluate the quality of the transmitted signal with the proposed timing fluctuation suppression technique, we collected the transmitted beam on the receiver, converted it to a 100 MHz microwave signal on the photodiode, and amplified it with a high-gain low-noise RF amplifier. The amplified 7 dBm microwave signal was mixed with the reference signal to produce a DC output. After low pass filtering, the DC signal was recorded by a high-resolution voltage meter. Our transfer experiment started at 1 a.m. and ended at 5:30 a.m. roughly. Since the wind was not very strong during the measuring time, the beam sway caused by the amplitude noise was not very significant in this case.



UESTC: University of Electronic Science and Technology of China

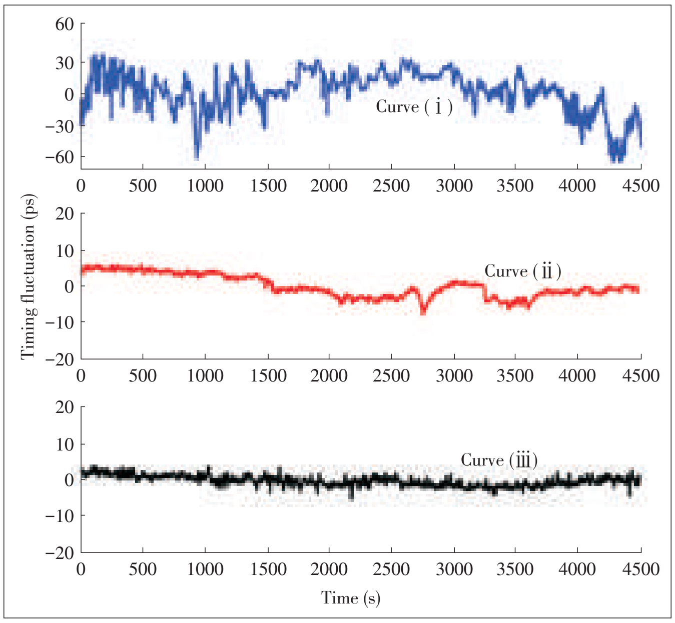
▲Figure 2. a) The actual experimental setup for portable atmospheric frequency transfer with the timing fluctuation suppression. The local and remote sites are located on a long avenue in UESTC. The distance between them is 60 m and the total free-space transmission distance is 120 m; b) the diode laser with a low-power current driver; c) the telescopes for launching and receiving beams.

4 Experimental Results and Discussion

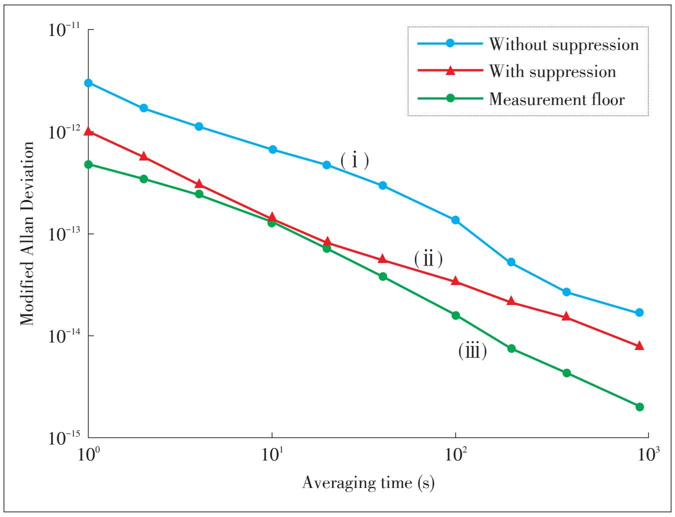
In this experiment, the two telescopes were put as close as possible at the transmitter side, to get the identical turbulence effect over the bidirectional transmission links. Since the phase compensation can suppress the extra timing fluctuations affected by turbulence, we believe the quality of the frequency transfer could be improved distinctly, compared to the direct link. Here, we measured the timing fluctuations and frequency instabilities of the transferred microwave signals with and without the timing suppression, respectively. The timing fluctuation results are shown in **Fig. 3**. Curve (i) shows the timing fluctuation of the transmitted 100 MHz microwave signal without timing fluctuation suppression, and its calculated RMS timing fluctuation is about 22 ps within 4500 s.

Curve (ii) shows the timing fluctuation of transmitted microwave signal with timing fluctuation suppression, and the RMS timing fluctuation is reduced to about 6 ps within 4500 s. Here, we also measured the timing fluctuations of frequency transfer with a short link as the measurement floor (Fig. 1), which is just attributed by the electronic noise of our photonic system. For this short link, its timing fluctuation is shown as Curve (iii) and the RMS timing fluctuation is calculated about 1.3 ps within 4500 s. By comparing the transmission links with and without timing fluctuation suppression, we believe that the phase compensation technique could suppress the timing fluctuation effectively.

Fig. 4 demonstrates the instability results of the transferred microwave signal. Curve (i) is the relative Allan Deviation result of the transferred signal without phase compensation, which is calculated from the sampled data in Fig. 3. It shows the 120 m free-space frequency transmission without timing fluctuation suppression has a instability of 3×10^{-12} for 1 s and 2×10^{-14} for 1000 s. Curve (ii) is the relative Allan Deviation result for the transferred signal with timing fluctuation suppression, and it shows the 120 m free-space frequency transmission with timing fluctuation suppression has a instability of 1×10^{-12} for 1 s and 7×10^{-15} for 1000 s. Curve (iii) shows the measurement floor, which was obtained via a short link (Fig. 1). With the comparison of these curves, we can find that the instability of the free-space transmission link with timing fluctuation suppression is improved distinctly. Note that, curve (iii) is merely the lower bound of the instability incurred during atmospheric transfer of microwave signals. This is because it was measured only with the short link, and most of the turbulence effect was cancelled. This Allan Deviation measurement floor in our case is limited by the stability of the frequency source and the electronic noise on local site. The accuracy achieved by our atmospheric frequency transfer system with phase compensation may be quite adequate with some short distance free-space applications. With comparison with instability of our transfer results and a commercial Cs clock (5071A) [25], we can find that the instability of our transmission link is lower. Therefore, we believe that disseminating a Cs or Rb clock signal over free-space link by using the proposed portable atmospheric frequency transfer scheme in this paper is feasible. The compensation bandwidth of our loop is about 10 kHz, and most of the timing fluctuation with the frequency below 10 kHz can be suppressed. However, this also limits the distance of the transmission link, because the short compensation time (100 μs, corresponding to 10 kHz) limits the round-trip travel time of optical beam. In this case, the distance will be limited to tens km (by multiplying compensation time and light velocity). Therefore, our technique for atmosphere transfer of microwave can be used in the application of short distance synchronization between two or more stations.



▲ **Figure 3. Timing fluctuation results for the atmospheric microwave transfer. Curve (i) is the result for 120 m free-space transmission link without timing fluctuation suppression; Curve (ii) is the result for 120 m free-space transmission link with timing fluctuation suppression; Curve (iii) is the result for a short link at local site as a measurement floor.**



▲ **Figure 4. Instability results for the atmospheric microwave transfer: (i) relative Allan Deviation between the transferred microwave and reference signal without timing fluctuation suppression; (ii) relative Allan Deviation with timing fluctuation suppression; (iii) Allan Deviation for a short link as the measurement floor.**

5 Conclusions

We demonstrate an outdoor atmospheric frequency transfer technique using a compact diode laser with a timing fluctuation suppression. The RMS timing fluctuation for 120 m transmission of a 100 MHz clock frequency was measured to be approximately 6 ps within 4500 s, with fractional frequency instability on the order of 1×10^{-12} at 1 s and the order of 7×10^{-15} at 1000 s. Comparing the instability of transfer results of the proposed system and a commercial Cs clock (5071A), we find that the instability of our transmission link is lower than the Cs clock. We believe that disseminating a Cs or Rb clock signal over free-space link by using the proposed atmospheric frequency transfer scheme is feasible. We will challenge in building a femtosecond portable atmospheric frequency transmission link with longer distance. There will be some improvement to achieve this. For example, we will use higher frequency microwave to increase the resolution of phase discrimination and higher power laser to increase the SNR on the photodetection. In addition, a fast steering mirror will be used to cancel the beam vibration.

References

[1] J. Levine, "A review of time and frequency transfer methods," *Metrologia*, vol. 45, no. 6, pp. S162–S174, Nov. 2008. doi: 10.1088/0026-1394/45/6/S22.

[2] B. H. Li, C. Rizos, H. K. Lee, and H. K. Lee, "A GPS-slaved time synchronization system for hybrid navigation," *GPS Solutions*, vol. 10, no. 3, pp. 207–217, Jul. 2006. doi: 10.1007/s10291-006-0022-z.

[3] W. Wang, C. Ding, and X. Liang, "Time and phase synchronisation via direct-path signal for bistatic synthetic aperture radar systems," *IET Radar, Sonar & Navigation*, vol. 2, no. 1, pp. 1–11, Jan. 2008. doi: 10.1049/iet-rsn:20060097.

[4] S. Bregni, "A historical perspective on telecommunications network synchronization," *IEEE Communications Magazine*, vol. 36, no. 6, pp. 158–166, Jun. 1998. doi: 10.1109/35.685385.

[5] S. M. Foreman, K. W. Holman, D. D. Hudson, D. J. Jones, and J. Ye, "Remote transfer of ultrastable frequency references via fiber networks," *Review of Scientific Instruments*, vol. 78, article 021101, Feb. 2007. doi: http://dx.doi.org/10.1063/1.2437069.

[6] J. Kim, J. A. Cox, J. Chen, and F. X. Kärtner, "Drift-free femtosecond timing synchronization of remote optical and microwave sources," *Nature Photonics*, vol. 2, no. 12, pp. 733–736, Dec. 2008. doi: 10.1038/nphoton.2008.225.

[7] D. Hou, P. Li, P. Xi, J. Zhao, and Z. Zhang, "Timing jitter reduction over 20-km urban fiber by compensating harmonic phase difference of locked femtosecond comb," *Chinese Optics Letters*, vol. 8, no. 10, pp. 993–995, Oct. 2010. doi: 10.3788/COL20100810.0993.

[8] D. Hou, P. Li, J. Zhao, and Z. Zhang, "Long-term stable frequency transfer over an urban fiber link using microwave phase stabilization," *Optics Express*, vol. 19, no. 2, pp. 506–511, Jan. 2011. doi: 10.1364/OE.19.000506.

[9] K. Predehl, G. Grosche, S. M. Raupach, et al., "920-kilometer optical fiber link for frequency metrology at the 19th decimal place," *Science*, vol. 335, no. 6080, pp. 441–444, Apr. 2012. doi: 10.1126/science.1218442.

[10] B. Wang, X. Zhu, C. Gao, et al., "Square kilometre array telescope–precision reference frequency synchronisation via 1f-2f dissemination," *Scientific Reports*, vol. 5, article ID 13851, Sept. 2015. doi: 10.1038/srep13851.

[11] T. Mao, Q. Chen, W. He, et al., "Free-space optical communication using patterned modulation and bucket detection," *Chinese Optics Letters*, vol. 14, no. 11, pp. 110607–110611, Nov. 2016. doi: 10.3788/COL201614.110607.

[12] V. W. S. Chan, "Free-space optical communications," *Journal of Lightwave Technology*, vol. 24, no. 12, pp. 4750–4762, Dec. 2006.

[13] F. Pappalardi, S. J. Dunham, M. E. LeBlang, et al., "Alternatives to GPS," in *Proc. OCEANS*, Honolulu, USA, Nov. 2001, pp. 1452–1459. doi: 10.1109/OCEANS.2001.968047.

[14] B. Sprenger, J. Zhang, Z. Lu, and L. Wang, "Atmospheric transfer of optical and radio frequency clock signals," *Optics Letters*, vol. 34, no. 7, pp. 965–967, Apr. 2009. doi: 10.1364/OL.34.000965.

[15] R. P. Gollapalli and L. Duan, "Atmospheric timing transfer using a femtosecond frequency comb," *IEEE Photonics Journal*, vol. 2, no. 6, pp. 904–910, Sept. 2010. doi: 10.1109/JPHOT.2010.2080315.

[16] R. P. Gollapalli and L. Duan, "Multiheterodyne characterization of excess phase noise in atmospheric transfer of a femtosecond-laser frequency comb," *Journal of Lightwave Technology*, vol. 29, no. 22, pp. 3401–3407, Nov. 2011. doi: 10.1109/JLT.2011.2169449.

[17] F. R. Giorgetta, W. C. Swann, L. C. Sinclair, et al., "Optical two-way time and frequency transfer over free space," *Nature Photonics*, vol. 7, pp. 435–438, Apr. 2013. doi: 10.1038/nphoton.2013.69.

[18] L. C. Sinclair, W. C. Swann, H. Bergeron, et al., "Synchronization of clocks through 12 km of strongly turbulent air over a city," *Applied Physics Letters*, vol. 109, article 151104, Oct. 2016. doi: http://dx.doi.org/10.1063/1.4963130.

[19] J. Kang, J. Shin, C. Kim, et al., "Few-femtosecond-resolution characterization and suppression of excess timing jitter and drift in indoor atmospheric frequency comb transfer," *Optics Express*, vol. 22, no. 21, pp. 26023–26031, Oct. 2014. doi: 10.1364/OE.22.026023.

[20] K. Djerroud, O. Acef, A. Clairon, et al., "Coherent optical link through the turbulent atmosphere," *Optics Letters*, vol. 35, no. 9, pp. 1479–1481, May 2010. doi: 10.1364/OL.35.001479.

[21] H. Bergeron, L. C. Sinclair, W. C. Swann, et al., "Tight real-time synchronization of a microwave clock to an optical clock across a turbulent air path," *Optica*, vol. 3, no. 4, pp. 441–447, Apr. 2016. doi: https://doi.org/10.1364/OPTICA.3.000441.

[22] J. Nie, L. Yang, and L. Duan, "Atmospheric transfer of a radio-frequency clock signal with a diode laser," *Appl. Opt.*, vol. 51, no. 34, pp. 8190–8194, Dec. 2012. doi: 10.1364/AO.51.008190.

[23] L. C. Sinclair, F. R. Giorgetta, W. C. Swann, et al., "Optical phase noise from atmospheric fluctuations and its impact on optical time-frequency transfer," *Phys. Rev. A*, vol. 89, no. 2, article 023805, Feb. 2014. doi: 10.1103/PhysRevA.89.023805.

[24] C. Robert, J. M. Conan, and P. Wolf, "Impact of turbulence on high-precision ground-satellite frequency transfer with two-way coherent optical links," *Phys. Rev. A*, vol. 93, no. 3, article 033860, Mar. 2016. doi: 10.1103/PhysRevA.93.033860.

[25] Microsemi. (2017). *DS-5071a* [Online]. Avaliable: http://www.microsemi.com/products/timing-synchronizationsystems/time-frequency-references/cesium-frequency-standards/5071a

Manuscript received: 2017-12-11

Biographies

CHEN Shijun (Chen.shijun@zte.com.cn) received his B.Sc. and master's degree from Harbin Engineering University, China. He currently works at Algorithm Department of ZTE. His research interests include MIMO, COMP and high-precision orientation. He has been in charge of and participated in 23 projects, some of which are supported by Chinese "863" Program, National Science and Technology Major Project, and National Key R&D Plan, and won 10 achievement awards. He has published more than 20 papers and holds more than 60 patents.

BAI Qingsong (baiqingsong@std.uestc.edu.cn) received his B.Sc. degree from Changchun University of Science and Technology, China in 2013. Since 2013, he has been studying at University of Electronic Science and Technology of China for his Ph.D. degree. His current research interests include femtosecond optical combs and highly stable frequency transfer on free-space link. He has published more than 10 papers.

CHEN Dawei (chen.dawei2@zte.com.cn) received his B.Sc. and master's degree from Harbin Institute of Technology, China. He is an algorithm engineer at ZTE Corporation. His research interest is indoor orientation. He has been in charge of and participated in two National Science and Technology Major Projects. He has published three papers.

SUN Fuyu (fysun@uestc.edu.cn) received his B.Sc. degree from Chengdu University of Technology, China and M. S. degree from University of Electronic Science and Technology of China. He is currently studying at University of Electronic Science and Technology of China for his Ph.D. degree. His current research interests include femtosecond optical combs and atom-based measurement technology. He has published more than 10 papers.

HOU Dong (houdong@uestc.edu.cn) received his B.Sc. degree in electronic engineering from North China University of Technology, China in 2004 and Ph. D. degree in electronic engineering from Peking University, China in 2012. From 2004 to 2007, he worked in the E-world and Lenovo Corporations respectively, as a senior electronic engineer. From 2012 to 2014, He worked at Peking University as a postdoctoral fellow. From 2014 to 2016, He worked at University of Colorado Boulder, USA, as a postdoctoral fellow. Now, he is an associate professor at University of Electronic Science and Technology of China. His current research interests include stabilization techniques for mode-locked laser/optical combs with high repetition frequency, highly stable frequency transfer on fiber link, and radio frequency circuit design. He has published more than 30 journal papers and 20 conference papers.

DOI: 10.19729/j.cnki.1673-5188.2018.04.009
http://kns.cnki.net/kcms/detail/34.1294.TN.20181121.0808.002.html, published online November 21, 2018

Time Sensitive Networking Technology Overview and Performance Analysis

FU Shousai, ZHANG Hesheng, and CHEN Jinghe
(Beijing Jiaotong University, Beijing 100044, China)

Abstract

Time sensitive networking (TSN) is a set of standards developed on the basis of audio video bridging (AVB). It has a promising future in the Industrial Internet of Things and vehicle-mounted multimedia, with such advantages as high bandwidth, interoperability and low cost. In this paper, the TSN protocol stack is described and key technologies of network operation are summarized, including time synchronization, scheduling and flow shaping, flow management and fault tolerant mechanism. The TSN network model is then established. Its performance is illustrated to show how the frame priority works and also show the influence of IEEE802.1Qbv time-aware shaper and IEEE802.1Qbu frame preemption on network and time-sensitive data. Finally, we briefly discuss the challenges faced by TSN and the focus of future research.

Keywords

TSN; AVB; the industrial internet of things (IIoT)

1 Introduction

With the development of the Industrial Internet of Things (IIoT), widely distributed network information needs to be shared and transmitted in time [1], [2]. At present, the industrial system is being extended from small closed networks to the IIoT, which requires reliable, integrated, remote and secure access to all network components. Existing Ethernet will no longer apply to customers who want to integrate Internet of Things (IoT) concepts into their industrial systems to improve productivity, increase normal running time or reduce maintenance.

In addition, Ethernet for Control Automation Technology (EtherCAT), Process Field Net (PROFINET) and other industrial Ethernet protocols are typically developed for specific tasks or domains, and are formed by modifying or adding specific functions based on standard Ethernet protocols. These protocols meet the real-time and deterministic requirements of industrial control systems that can perform their specific tasks well in specific areas, but they are limited when combined with standard Ethernet networks and devices. Lack of bandwidth, lack of interoperability and high cost make it difficult to meet data transmission requirements of Industrial 4.0 presently. As a result, NI, Broadcom, Cisco, Harman, Intel, Xilinx and other well-known companies have jointly founded the Avnu Alliance. It aims at the promotion of the new Ethernet standard for time sensitive networking (TSN) applied to the IIoT [3]. The core technology of TSN includes network bandwidth reservation, precise clock synchronization, and traffic shaping, which ensures low latency, high reliability and other needs.

As it is a new field of study, there are few contributions in the domain of TSN. Related to the TSN research topics, [4] presents a model, called audio video bridging (AVB) scheduled traffic (AVB ST). The main difference between TSN and AVB ST lies in the way protected windows are created for time-sensitive traffic. Paper [5] presents a delay analysis of AVB frames under hierarchical scheduling of credit-based shaping and time-aware shaping on TSN switches. Considering TSN's time-aware and peristaltic shapers and evaluating whether these shapers are able to fulfill these strict timing requirements, a formal timing analysis is presented in [6], which is a key requirement for the adoption of Ethernet in safety-critical real-time systems, to derive worst-case latency bounds for each shaper. In [7], the equations are derived to perform worst-case response time analysis on Ethernet AVB switches by considering its credit-based shaping algorithm. Moreover, [8], [9] and [10] analyze the transmission delay of TSN network. Recently, an analysis is proposed to compute fully deterministic schedules (i.e. time aware shaper (TAS) scheduling tables) for TSN multi-hop switched networks, while identifying functional parameters that affect communication behavior [11].

Our work is based on the IEEE 802.1Qbv TSN standard, which enhances Ethernet networks to support time sensitive applications in the automotive and industrial control domains. A key feature of TSN is the new traffic shaping mechanism TAS, which is capable of accommodating hard real-time streams with deterministic end-to-end delays. This paper describes the TSN protocol stack and summarizes the key technologies used

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

in the process of network operation. A specific example is given to analyze how frame priority works and the impact of IEEE802.1Qbv time aware shaping and IEEE802.1Qbu frame preemption on network and time sensitive data. The current research status and future trend of TSN are also analyzed.

2 AVB and TSN

TSN is an extension of IEEE 802.1 Ethernet, which is a set of new standards developed by the Time Sensitive Networking Task Group of the IEEE 802.1 Working Group on the basis of existing standards. The TSN task force was established in November 2012, renaming the existing AVB task set, extending the scope of the AVB's work and supporting all standard AVB devices. In other words, TSN is actually an enhancement and improvement of AVB [12], [13].

2.1 Overview of AVB/TSN

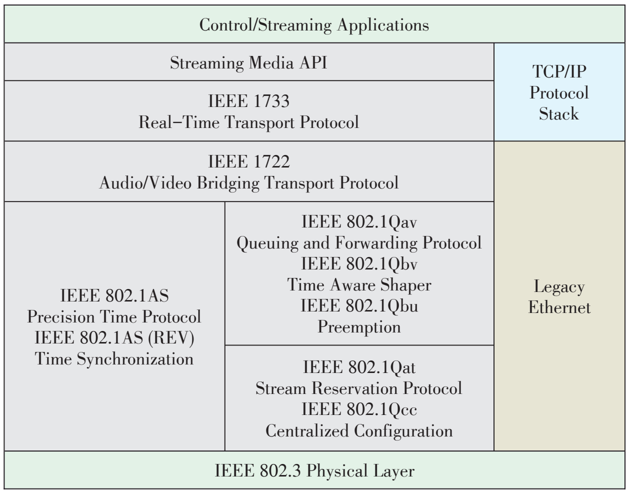
AVB is a set of protocol sets for real-time audio and video transmission, which was developed in 2005 by the IEEE 802.1 Task Group [13], [14]. It can effectively solve the problem of timing, low latency and traffic shaping in network transmission. At the same time, it guarantees 100% backward compatibility with traditional Ethernet. AVB is the potential next-generation audio/video transmission technology, including:

- 802.1AS: Precision Time Protocol (PTP)
- 802.1Qat: Stream Reservation Protocol (SRP)
- 802.1Qav: Queuing and Forwarding Protocol (Qav)
- 802.1BA: Audio Video Bridging Systems
- 1722: Audio/Video Bridging Transport Protocol (AVBTP)
- 1733: Real-Time Transport Protocol (RTP)
- 1722.1: Device Discovery, Enumeration, Connection Management and Control Protocol for 1722-Based Devices.

As shown in **Fig. 1**, TSN is primarily aimed at the data link layer of ISO model, and it is a new standard that seeks to make Ethernet real-time (low latency) and deterministic (high reliability) [7], [15], [16]. It mainly includes:

- 802.1AS (REV): Time Synchronization (Update timing and synchronization based on 802.1AS-2011)
- 802.1Qbv: Time Aware Shaper (Traffic scheduling was enhanced based on 802.1Qav)
- 802.1Qbu: Preemption (Update frame preemption based on 802.1Qav)
- 802.1Qci: Ingress Policing
- 802.1CB: Seamless Redundancy
- 802.1Qcc: Centralized Configuration (Enhancements and performance improvements for Stream Reservation Protocol).

The protocols for TSN, such as OLE for Process Control (OPC) Unified Architecture (UA), Object Management Group (OMG) Data Distribution Service for Real - Time Systems (DDS), IEEE1722, PROFINET, IEC61850 and Ethernet/IP, and other industrial Ethernet protocols provide a common layer 2 (data link layer) [17], which makes Ethernet transmission more reliable, jitter lower and delay shorter and meets all the requirements of real-time Ethernet. However, TSN will not and cannot replace the existing real-time protocols, because TSN is just a set of second level protocols, which provides all real-time features that are close to hardware, but does not provide a whole stack of second layers. In the long run, it may replace all second layers of extensions for PROFINET, EtherCAT, etc., but there will still be PROFINET or Ethernet/IP.

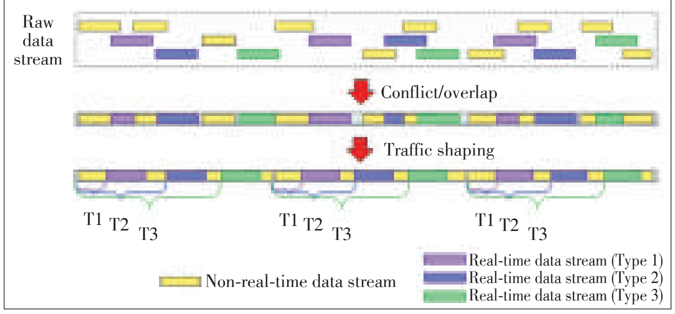


API: Application Programming Interface
REV: revision
TCP/IP: Transmission Control Protocol/Internet Protocol

▲Figure 1. The audio video bridging/ time sensitive networking (AVB/TSN) protocol set in the open systems interconnection (OSI) model hierarchy.

2.2 Basic Principles of AVB/TSN

In the case of insufficient bandwidth, the data from different devices overlap with each other. As shown in **Fig. 2**, the conflicting parts of all the data flows in this case are forwarded by the QoS priority mechanism. On the one hand, network devices cannot tolerate too much delay forwarding. On the other hand, the physical port cache of switches is very small, which cannot effectively solve a large number of data packets arriving at the same time, so some data packets will be discarded. Typically, when a switch has a bandwidth occupancy rate of over 40%, it


Raw
data
stream
Conflict/overlap
Traffic shaping
T1 T2 T3
T1 T2 T3
T1 T2 T3
Non-real-time data stream
Real-time data stream (Type 1)
Real-time data stream (Type 2)
Real-time data stream (Type 3)

▲Figure 2. Traffic shaping schematic.

has to be expanded. The goal is to avoid congestion by increasing the bandwidth of the network.

In order to avoid the overlap of bandwidth, traffic shaping is carried out for different data streams to achieve the purpose of improving reliable delivery. As shown in Fig. 2, after traffic shaping, the bandwidth occupied by each real-time stream is at the same time node, and all non-real-time streams consume other bandwidth, which will ensure reliable delivery of real-time data.

The AVB/TSN can make traffic shaping for the sender, such as the network ports of different audio/video equipment, and also make the reshaping of each forwarding node in the switch. The basic principle of AVB/TSN is that each audio/video stream only occupies its own bandwidth and does not affect the transmission of other data streams [18], [19].

2.3 Main Features and Typical Application of TSN

TSN has a lot of characteristics compared to existing standards and proprietary industrial Ethernet protocols [20]:

1) Bandwidth: Advanced sensing applications generate large amounts of data, resulting in network bandwidth resource constraints. At present, the dedicated Ethernet protocol commonly used in industrial control is generally limited to 100 MB bandwidth and half duplex communication. TSN incorporates a variety of standard Ethernet rates (including 1 GB, 10 GB and 400 GB currently in use) and support full-duplex transmissions.
2) Security: TSN extends the security of infrastructure for underlying control and integrates IT security rules.
3) Interoperability: By using standard Ethernet components, TSN seamlessly integrates existing applications and standard IT networks to improve usability, such as HTTP interfaces and Web service, and realize the IIoT system for remote diagnosis, visualization, repair, and other functions.
4) Low cost: TSN uses standard Ethernet chipsets as mass-produced commercial silicon chips. This reduces component costs, which is particularly evident compared with the use of a special Ethernet protocol based on ASIC chips.
5) Delay and synchronization: Fast response systems and closed-loop control applications require low latency communications. TSN achieves deterministic transmission of tens of microseconds and time synchronization at dozens of nanoseconds between nodes, and provides automated configuration for high-reliability data transmission paths to provide lossless path redundancy by copying and merging packets.

TSN is a deterministic network with the following typical applications [1], [21]:

1) Professional audio and video (Pro AV): The main clock frequency is emphasized in the field of professional audio and video. In other words, all video network nodes must keep to the time synchronization mechanism.
2) Automotive control: Most automobile control systems are very complicated at present. In fact, all systems can be managed with TSN that support low latency and real-time transport mechanisms, reducing the cost and complexity of adding network capabilities to automotive and professional audio/video devices [22].
3) Industrial areas: TSN networks can be applied in the industrial areas that require real-time monitoring or real-time feedback, such as robotics industry, deepwater oil drilling, and banking. In addition, TSN can also be used to support large data transfer between servers. At present, the global industry has entered the IoT era. TSN is an effective way to improve the efficiency of the IoT.

3 Key Technologies of TSN

In order to provide a complete real-time communication solution, IEEE 802.1 has developed a TSN standard file, which can be divided into the following three basic components: time synchronization, scheduling and traffic shaping, and stream management and fault tolerance.

3.1 Time Synchronization

Time plays an important role in the TSN network compared with the IEEE 802.3 and IEEE 802.1Q standard Ethernet. By clock synchronization, the network devices can run consistently and perform the required operations at the specified point in time.

Time synchronization in TSN networks can be achieved with different technologies. Time in a TSN network is typically distributed from a central time source through the network itself. In most cases, this is done using the IEEE 1588 Precision Time Protocol, which utilizes Ethernet frames to distribute time synchronization information. In addition to the IEEE 1588 standard, the Time-Sensitive Task Group of the IEEE 802.1 Working Group has also developed a brief for IEEE 1588, called IEEE 802.1AS-2011, which is mainly applicable to Internet environments such as home, automotive, and industrial automation.

3.2 Scheduling and Traffic Shaping

The purpose of scheduling and traffic shaping is to allow different traffic classes to coexist in the same network. These traffic classes have different priorities and have different requirements for available bandwidth and end-to-end delay. In the field of industrial automation and cars, as a result of the existence of closed loop control and safety application, reliable and timely information delivery plays an important role, so the IEEE 802.1Q strict priority scheduling mechanism needs to be strengthened.

3.2.1 TAS

TSN enhances standard Ethernet traffic by adding mechanisms. In order to maintain backward compatibility, TSN still maintains eight different virtual local area network (VLAN) priorities, which keeps the interoperability with existing infrastructure well and allows seamless migration to new technologies. IEEE 802.1Qbv Time-Aware Shaper separates the communication time on the Ethernet network into a fixed length and a repetitive cycle. In the cycle, different time slices may be configured, each of which is assigned with one or more of eight priorities. It is a time division multiple access (TDMA) scheme that separates time-critical traffic from non-critical background services by establishing a virtual channel for a specific time period [22].

1) IEEE 802.1Qbv Guard Bands Mechanism

Since the transmission of frames cannot be interrupted, this presents a challenge to the TDMA approach of IEEE 802.1Qbv scheduling. As shown in **Fig. 3**, if a new frame is transmitted before the end of the time slice 2 of cycle *n*, since the frame is too large and the transmission process cannot be interrupted, the frame invokes the subsequent time slice 1 of the next cycle *n*+1. Through partial or complete blocking, real-time frames in the subsequent time slices will be delayed, which cannot meet the needs of applications. What it has to do for the actual buffer effect is very similar with ordinary Ethernet switches [11].

As shown in **Fig. 4**, the guard band in TAS can prevent this from happening. During this guard band, no new Ethernet frame transmission may be started, only already ongoing transmissions may be finished and the duration of this guard band has to be as long as it takes the maximum frame size to be safely transmitted. For an Ethernet frame, the maximum length is: 1518 bytes (frames) + 4 bytes (VLAN tag) +12 bytes (frame spacing) = 1534 bytes.

While the guard bands manage to protect the time slices with high priority and critical traffic, they also have some significant drawbacks: The guard band can cause loss of bandwidth; the guard band affects the minimum achievable time slice length and cycle time. The standard IEEE 802.1Qbv contains a length-aware scheduling mechanism in order to partially reduce the bandwidth loss due to the presence of the guard band. Therefore, length-aware scheduling is an improvement, but cannot mitigate all drawbacks that are introduced by the guard band.

2) IEEE 802.1Qbu Frame Preemption for Minimizing Guard Band

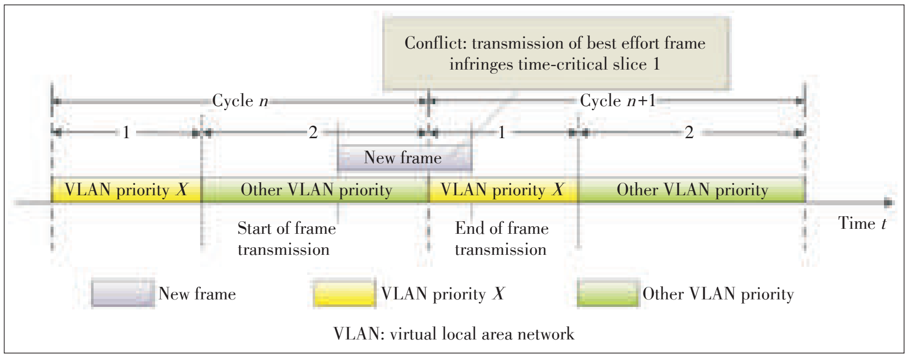
To further mitigate the negative effects from the guard bands, the IEEE 802.1 and 802.3 Working Groups have specified the frame preemption technology. IEEE 802.1Qbu is used for bridging management components [24] and IEEE 802.3br for Ethernet MAC components [25]. Fig. 4 uses a basic example to show how frame preemption works. During the process of sending one best effort Ethernet frame, the MAC interrupts the frame transmission before the guard band. After the high priority traffic of time slice 1 passes, the period is switched back to time slice 2, and the interrupted frame transmission is resumed.

Frame preemption allows for a significant reduction of the guard band. The length of the guard band is dependent on the precision of the frame pre-emption mechanism. IEEE 802.3br specifies the best accuracy of 64 bytes for this mechanism, since this is the minimum size of a still valid Ethernet frame. In this case, the guard band can be reduced to the total of 127 byte: 64 byte (the minimum frame) plus 63 byte (the minimum length that cannot be preempted).

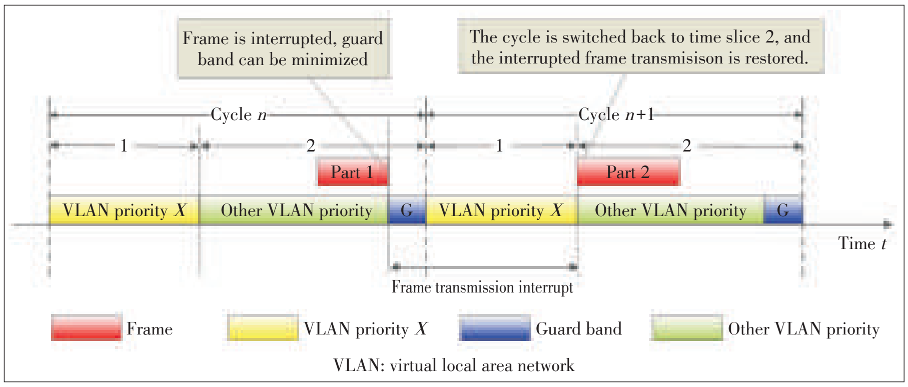
3.2.2 Credit-Based Shaper (CBS)

We can achieve low latency for data transfer by configuring protected windows and prioritizing, but in some cases it is also important to eliminate jitter. The reason is that a certain time delay may be acceptable, but larger jitter will degrade the quality of communication service. For example, some delays do not make a difference when watching a video, but they may lead to frame skipping if the jitter is too high. Credit-based shaping can be used to transmit data more evenly in order to provide a continuous stream. **Fig. 5** shows the basic idea [26], [27].

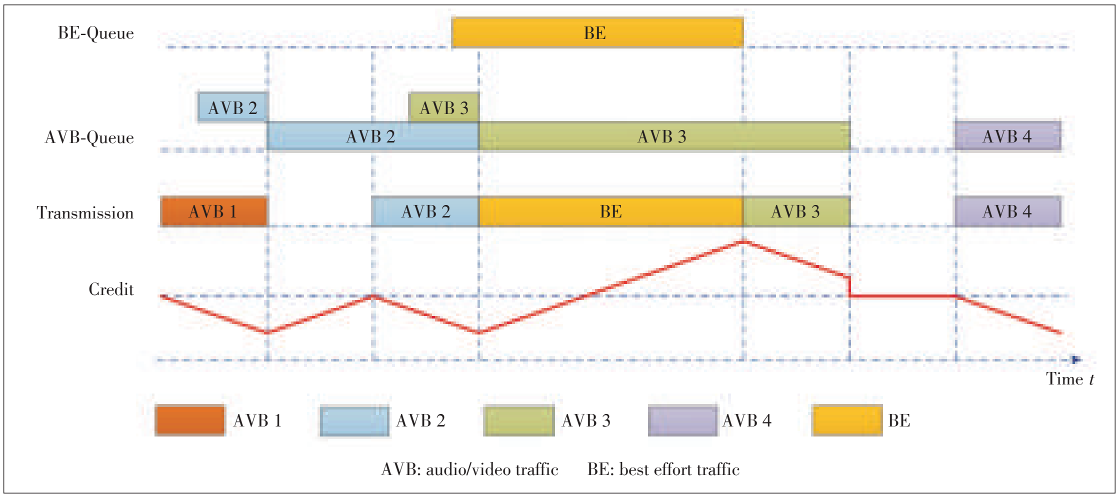
The transmission of Audio/Video traffic (AVB) is analyzed


Conflict: transmission of best effort frame infringes time-critical slice 1
Cycle *n*
Cycle *n*+1
1
2
1
2
New frame
VLAN priority *X*
Other VLAN priority
VLAN priority *X*
Other VLAN priority
Time *t*
Start of frame transmission
End of frame transmission
New frame
VLAN priority *X*
Other VLAN priority
VLAN: virtual local area network

▲**Figure 3. The frame that is sent too late in the best effort time slice infringes the high-priority time slice.**


Frame is interrupted, guard band can be minimized
The cycle is switched back to time slice 2, and the interrupted frame transmisison is restored.
Cycle *n*
Cycle *n*+1
1
2
1
2
Part 1
Part 2
VLAN priority *X*
Other VLAN priority
G
VLAN priority *X*
Other VLAN priority
G
Time *t*
Frame transmission interrupt
Frame
VLAN priority *X*
Guard band
Other VLAN priority
VLAN: virtual local area network

▲**Figure 4. Example of frame preemption.**



◀**Figure 5.**
Credit-based shaper.

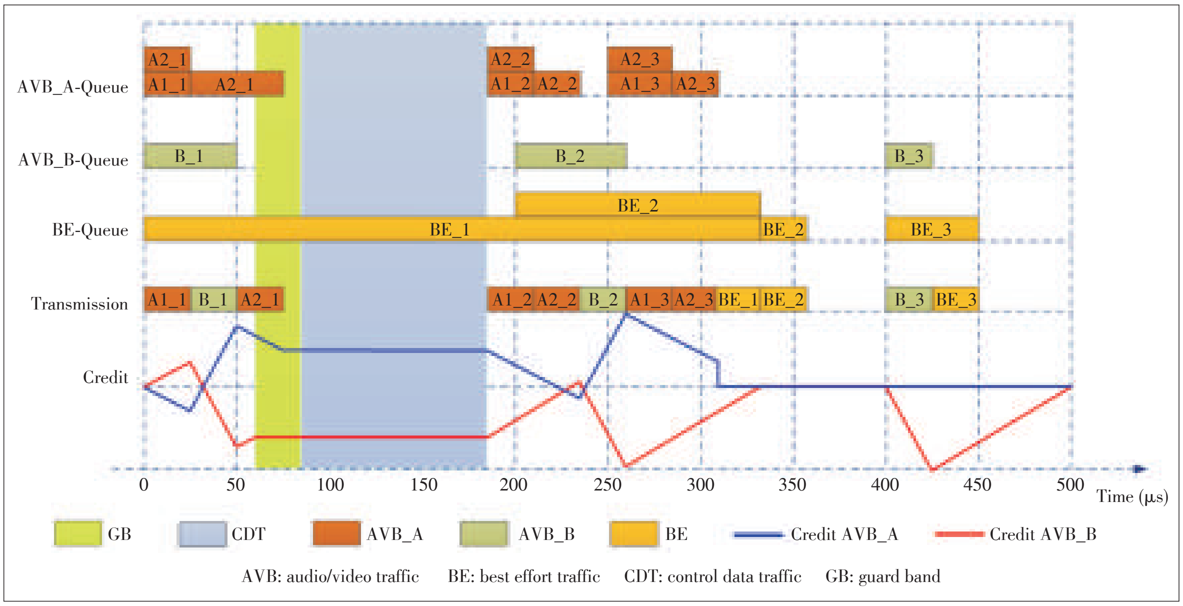
as an example: the queue starts to transmit when the credit is positive or zero and the credit decreases with a fixed slope during the transmission; when there is a frame waiting, the credit will accumulate with another fixed slope; the slope of the decrease or accumulation of credit can be configured according to the actual situation or experience. In addition, the credit will be set to zero when the credit is positive but the queue is emptied. It can be seen that the shaping mechanism of CBS can make the transmission of AVB data flow more uniform, thus reducing the jitter. It also makes the lower priority data streams have the opportunity to be transmitted.

3.2.3 TAS+CBS Scheduling Mechanism

There are many types of data streams in a communication system, such as control data traffic (CDT), audio/video traffic (AVB), and other background services traffic (best effort (BE) traffic). They have different characterization requirements: the real-time requirement of CDT transmission is higher; AVB transmission requires less jitter; BE traffic has lower priority. In view of the situation, we adopt TAS + CBS scheduling mechanism [28], [29].

The following example illustrates the scheduling mechanism of TAS + CBS. The IEEE TSN standard states that the credit will no longer increase for the duration of a class of data flows that is forbidden to be transmitted. To make it easier to understand, we only analyze the traffic transmission in a time period T = 500 μs. As shown in **Fig. 6**, there are two AVB_A streams, one AVB_B stream and one BE stream for transmission in the TSN switch. The credits for AVB_A data streams are shown in blue, while the credits for AVB_B data streams are shown in red. The frames of AVB_A begin transmission at t = [0 μs, 125 μs, 250 μs], the frames of AVB_B and BE begin transmission at t = [0 μs, 200 μs, 400 μs], the guard band is activated at t = 60 μs, and the CDT slot is activated at t = 85 μs. We assume that all frames have the same size of 25 μs, so the guard band is also defined to be 25 μs and the size of the CDT slot is spec-

Figure 6.▶
Time Aware Shaper + Credit-Based Shaper (TAS+CBS) scheduling mechanism.



ified as 100 μs.

It can be seen that at t = 50 μs, stream AVB_A2 starts transmitting a frame, and it continues transmitting until t = 75 μs. Between t = 60 μs and t = 75 μs, the number of credits of the AVB_A class flow continues to decline because the frame begins to transmit before the guard band. As opposed to class AVB_A, class AVB_B has a negative credit at t = 50 μs, hence its credits are incremented according to its idle slope until t = 60 μs, at which moment its gate is closed due to the activation of the guard band. The number of credits of class AVB_B stays constant during the guard band and the CDT-slot, even though it has a negative value. It is worth noting that at t = 310 μs, the number of credits of AVB_A drops from positive to zero because there is no AVB_A frame waiting for sending at this time. From the example analysis, the TAS + CBS scheduling mechanism can effectively reduce the jitter of the audio/video traffic while ensuring the reliable and real-time transmission of the control data traffic.

3.3 Stream Management and Fault Tolerance

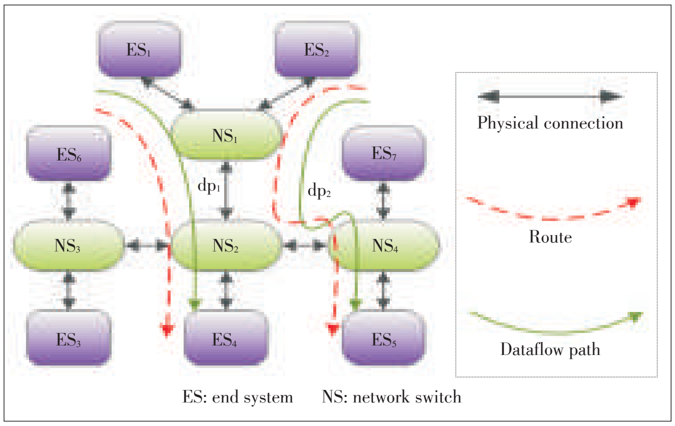
The core of TSN is its meeting the requirements of individual applications regarding timing behavior and reliability. In order to achieve the TSN characteristics, applications must register the corresponding data flows before their transmission. The identification, registration, and management of suitable paths can be a challenge, especially in larger networks and in conjunction with fixed transmission windows for different streams. To support the identification, registration, and management of suitable paths, TSN defines a set of mechanisms and interfaces in IEEE P802.1Qcc.

The reliability of data flows, especially in the event of errors, is also of great importance for many TSN application scenarios. For example, safety-related control loops or vehicle autonomous driving networks must be protected against failure in hardware or network media. Therefore, the TSN Task Group is currently providing the fault tolerance protocol IEEE 802.1CB for this purpose [30], and the mechanisms defined in IEEE P802.1CB and IEEE P802.1Qca allow replication and redundant transmission of data over several disjunctive paths. In addition to this agreement, existing high reliability protocols, such as Hierarchical State Routing (HSR) and Probabilistic Routing Protocol (PRP), as specified in IEC 62439-3, may also be used.

4 Modeling and Performance Analysis of Time Sensitive Networking

4.1 Modeling

The establishment of a simple network model is shown in **Fig. 7** [31], [32]. A dataflow path dp_i is an ordered sequence of links connecting one sender $ES_i \in ES$ to one receiver $ES_j \in ES$. In Fig. 7, we have:



▲Figure 7. A TSN topology model.

$$dp_1 = (ES_1 \rightarrow NS_1, NS_1 \rightarrow NS_2, NS_2 \rightarrow ES_4), \qquad (1)$$

$$dp_2 = (ES_2 \rightarrow NS_1, NS_1 \rightarrow NS_2, NS_2 \rightarrow NS_4, NS_4 \rightarrow ES_5). \qquad (2)$$

4.2 Network Performance Analysis

The TSN performance is analyzed by a specific example. We define a time unit as a step. The characteristics of the example frame are shown in **Table 1**, which explains why the time criticality problem may still occur in the case of defining a priority. Moreover, for easy understanding, we do not consider the existence of the guard band in the following analysis.

4.2.1 IEEE 802.1Q Strict Priority Scheduling Mechanism

Based on the network topology model in Fig. 7, we using the IEEE 802.1Q priority scheduling mode is used and the results are shown in **Fig. 8**. It is not difficult to see that even if these frames have different priorities, the frame delay time with high priority is not necessarily more stable. Frame 1 with the lowest priority has a stable delay of 9 steps, Frame 2 has a delay of 14 or 13 steps, and Frames 3 and 4 have 8 steps of delay. The total delay time in the example is 77.

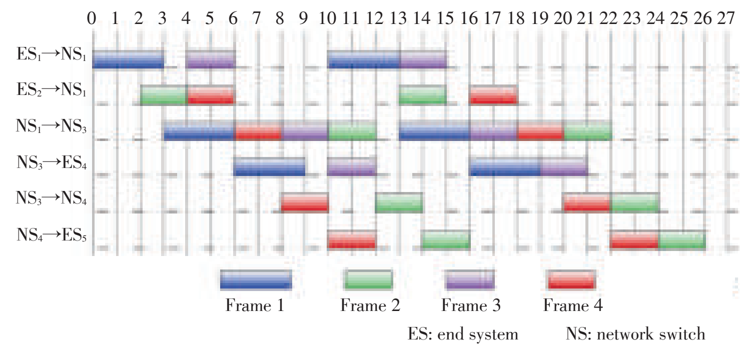
4.2.2 IEEE 802.1Qbv Time-Aware Shaper

If Frame 2 is considered to be a time-critical key frame, other features remain the same. In order to guarantee the low delay of Frame 2, the IEEE 802.1Qbv time-aware shaper mechanism is adopted and the results are shown in **Fig. 9**.

By comparison, it can be found that the total time of transmission of all frames increases from 77 to 82 steps. However,

▼Table 1. Example frame features

Frame	Priority	First arrival	Interval	Transmission time/length
Frame 1	1	0	10	3
Frame 2	3	2	11	2
Frame 3	5	4	9	2
Frame 4	7	4	12	2



Scheduling results

Frame	ID	Arrival	Finish	Latency
Frame 1	1	0	9	9
	2	10	19	9
Frame 2	1	2	16	14
	2	13	26	13
Frame 3	1	4	12	8
	2	13	21	8
Frame 4	1	4	12	8
	2	16	24	8

◀**Figure 8.**
IEEE 802.1Q priority scheduling and results.

for the time-sensitive Frame 2, the delay is stable to 8 steps, which is ES_2 to ES_5 transmission of the best time.

4.2.3 IEEE 802.1Qbu Frame Preemption Scheduling Mechanism

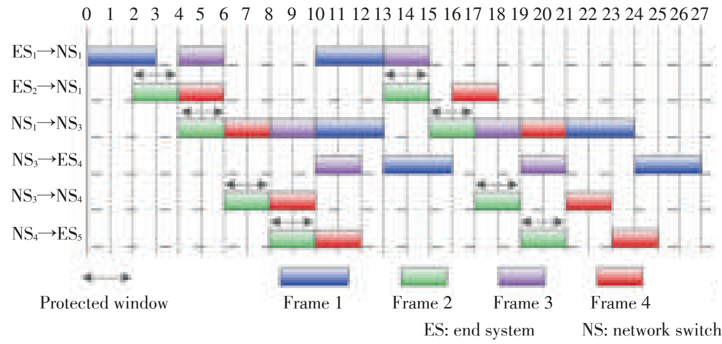
Frame 2 is a critical frame for time requirement. The IEEE 802.1Qbu frame preemption mechanism is adopted here and **Fig. 10** shows the results. It can be found that this scheduling mechanism can guarantee the timely transmission of Frame 2 and overcome the shortcomings of IEEE 802.1Qbv Time-Aware Scheduling. It reduces the total time of transmission frame from 82 steps to 78 steps.

5 Conclusions

In this paper, the time sensitive networking is introduced. The key technologies of TSN are summarized. A concrete example is given to illustrate the unique performance of the time sensitive networking. We obtain the following results:

1) IEEE802.1Q strict priority scheduling is very easy to generate data conflict in the case of insufficient bandwidth, and cannot guarantee the timely transmission of data frames.
2) IEEE802.1Qbv Time-Aware Shaper can ensure the timely transmission of key frames, but the protected window and guard band mechanism will cause a certain degree of bandwidth waste at the downside.
3) IEEE802.1Qbu guarantees the timely transmission of data, and at the same time, uses the frame preemption mechanism to minimize the guard band to solve the problem of bandwidth waste in IEEE802.1Qbv Time-Aware Shaper.

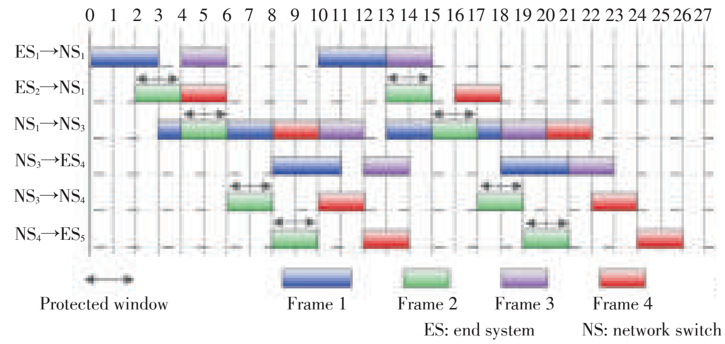
Some TSN standards have not been formally released, and the related applications in automotive electronics and industries have not been widely promoted. Basically, TSN is still in the testing platform stage. The development of chips and prod-



Scheduling results

Frame	ID	Arrival	Finish	Latency	Difference
Frame 1	1	0	16	16	+7
	2	10	27	17	+8
Frame 2	1	2	10	8	−6
	2	13	21	8	−5
Frame 3	1	4	12	8	0
	2	13	21	8	0
Frame 4	1	4	12	8	0
	2	16	25	9	+1

Figure 9.▶
IEEE 802.1Qbv time-aware scheduling and results.



Scheduling results

Frame	ID	Arrival	Finish	Latency	Difference
Frame 1	1	0	11	11	−5
	2	10	21	11	−6
Frame 2	1	2	10	8	0
	2	13	21	8	0
Frame 3	1	4	14	10	+2
	2	13	23	10	+2
Frame 4	1	4	14	10	+2
	2	16	26	10	+1

Figure 10.▶
IEEE802.1Qbu frame preemption scheduling and results.

ucts for TSN will be the next focus of major manufacturers and related research institutes.

References

[1] Y. Zhong, "I^2oT: advanced direction for internet of things," *ZTE Communications*, vol. 13, no. 2, pp. 3–6, Jun. 2015. doi: 10.3969/j.issn.1673-5188.2015.02.001

[2] R. Y. Yu, Y. Liu, and D. Liu, "Application of time-sensitive network in industrial internet," *Automation Expo*, no. 6, pp. 58–60, 2017. doi: 10.3969/j.issn.1003-0492.2017.06.030.

[3] TTTech. (2015). *IEEE TSN (Time-Sensitive Networking): A Deterministic Ethernet Standard* [Online]. Available: https://www.tttech.com/technologies/deterministic-ethernet/time-sensitive-networking

[4] M. Ashjaei, G. Patti, M. Behnam, et al., "Schedulability analysis of Ethernet audio video bridging networks with scheduled traffic support," *Real-Time Systems*, vol. 53, no. 4, pp. 526–577, Jul. 2017. doi: 10.1007/s11241-017-9268-5.

[5] D. Maxim and Y.-Q. Song, "Delay analysis of AVB traffic in time-sensitive networks (TSN)," in *International Conference on Real-Time Networks and Systems (RTNS)*, Grenoble, France, Oct. 2017, pp. 10 - 10. doi: 10.1145/3139258.3139283.

[6] D. Thiele, R. Ernst, and J. Diemer, "Formal worst-case timing analysis of ethernet TSN's time-aware and peristaltic shapers," in *IEEE Vehicular Networking Conference (VNC)*, Kyoto, Japan, Jan. 2015, pp. 251- 258. doi: 10.1109/VNC.2015.7385584.

[7] Simon, Csaba, et al., "Ethernet with time sensitive networking tools for industrial networks," *Infocommunications Journal IX*.2, pp. 6-14, 2017.

[8] F. Smirnov and F. Reimann, "Formal timing analysis of non-scheduled traffic in automotive scheduled TSN networks," in *Design, Automation & Test in Europe Conference & Exhibition*, Lausanne, Switzerland, Mar. 2017, pp. 1647- 1650. doi: 10.23919/DATE.2017.7927256.

[9] Y. Baga, F. Ghaffari, E. Zante, M. Nahmiyace, and D. Declercq, "Worst frame backlog estimation in an avionics full-duplex switched ethernet end-system," in *Digital Avionics Systems Conference (DASC)*, Sacramento, USA, Dec. 2016. doi: 10.1109/DASC.2016.7777990.

[10] H. Bauer, J. L. Scharbarg, and C. Fraboul, "Worst-case backlog evaluation of avionics switched ethernet networks with the trajectory approach," in *Real-Time Systems (ECRTS)*, Pisa, Italy, Jul. 2012, pp. 78- 87. doi: 10.1109/ECRTS.2012.12.

[11] S. S. Craciunas, R. S. Oliver, M. Chmelík, et al., "Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks," in *24th International Conference on Real-Time Networks and Systems*, Brest, France, Oct. 2016, pp. 183–192. doi: 10.1145/2997465.2997470.

[12] H. Lee, J. Lee, C. Park, and S. Park, "Time-aware preemption to enhance the performance of audio/video bridging (AVB) in IEEE 802.1 TSN," in *IEEE International Conference on Computer Communication and the Internet*, Wuhan, China, Oct. 2016, pp. 80–84. doi: 10.1109/CCI.2016.7778882.

[13] H. T. Lim, D. Herrscher, M. J. Waltl, et al., "Performance analysis of the IEEE 802.1 Ethernet audio/video bridging standard," in *International ICST Conference on Simulation TOOLS and Techniques*, Desenzano del Garda, Italy, Mar. 2012, pp. 27–36. doi: 10.4108/icst.simutools.2012.247747.

[14] H. Yang, G. H. Qin, H. Yu, et al., "Review of vehicle time-sensitive network technology," *Computer Applications and Software*, no. 8, pp. 1–5, 2015. doi: 10.3969/j.issn.1000-386x.2015.08.001.

[15] D. Pannell. (2015, Oct). *IEEE TSN standards overview & update* [Online]. Available: http://standards.ieee.org/events/automotive/2015/03_IEEE_TSN_Standards_ Overview_and_Update_v4.pdf

[16] J. Y. Cao, P. J. L. Cuijpers, R. J. Bril, and J. J. Lukkien, "Tight worst-case response-time analysis for ethernet AVB using eligible intervals," in *IEEE World Conference on Factory Communication Systems*, Aveiro, Portugal, May 2016, pp. 1–8. doi: 10.1109/WFCS.2016.7496507.

[17] V. Goller. (2016). *Time sensitive networks for industrial automation systems* [Online]. Available: http://www.analog.com/media/en/Other/electronica/Time-Sensitive-Networking-for-Industrial-Applications-Volker-Goller.pdf

[18] K. Zhang. (2017, Jan.). *The difference between traditional Ethernet and time-sensitive network TSN* [Online]. Available: http://www.proav - china.com/News/16800.html

[19] U. D. Bordoloi, A. Aminifar, P. Eles, et al., "Schedulability analysis of ethernet AVB switches," in *IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, Chongqing, China, Aug. 2014, pp. 1–10. doi: 10.1109/RTCSA.2014.6910530.

[20] NI. (2016). *Network standard evolution of industrial Internet of things* [Online]. Available: ftp://ftp.ni.com/pub/branches/china/Trend_Watch_2016_IIOT.pdf

[21] Industrial Ethernet Book. (2015, Jul.). *Deterministic Ethernet & TSN: automotive and industrial IoT* [Online]. Available: https://www.tttech.com/technologies/deterministic-ethernet/time-sensitive-networking

[22] A. Neumann, M. J. Mytych, D. Wesemann, L. Wisniewski, and J. Jasperneite, "Approaches for in-vehicle communication—an analysis and outlook," in *International Conference on Computer Networks (CN)*, 2017, pp. 395- 411. doi: 10.1007/978-3-319-59767-6_31.

[23] *IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks—Amendment 25: Enhancements for Scheduled Traffic*, 802.1Qbv-2015, Mar. 2016.

[24] *IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks—Amendment 26: Frame Preemption*, IEEE 802.1Qbu-2016, Aug. 2016.

[25] *IEEE Standard for Ethernet Amendment 5: Specification and Management Parameters for Interspersing Express Traffic*, IEEE 802.3br-2016, Oct. 2016.

[26] P. Meyer, T. Steinbach, F. Korf, and T. C. Schmidt, "Extending IEEE 802.1 AVB with time-triggered scheduling: a simulation study of the coexistence of synchronous and asynchronous traffic," in *IEEE Vehicular NETWORKING Conference*, Boston, USA, Feb. 2013, pp. 47–54. doi: 10.1109/VNC.2013.6737589.

[27] J. Y. Cao, P. J. L. Cuijpers, R. J. Bril, and J. J. Lukkien, "Independent yet Tight WCRT Analysis for Individual Priority Classes in Ethernet AVB," in *International Conference on Real-Time Networks and Systems*, New York, USA, Oct. 2016, pp. 55–64. doi: 10.1145/2997465.2997493.

[28] D. Maxim and Y. Q. Song, "Delay analysis of AVB traffic in time-sensitive networks (TSN)," in *International Conference on Real-Time Networks and Systems*, Grenoble, France, Oct. 2017, pp. 18–27. doi: 10.1145/3139258.3139283.

[29] F. Dürr and N. G. Nayak, "No-wait packet scheduling for IEEE time-sensitive networks (TSN)," in *International Conference on Real-Time Networks and Systems*, Brest, France, Oct. 2016, pp. 203–212. doi: 10.1145/2997465.2997494.

[30] S. Kehrer, O. Kleineberg, and D. Heffernan, "A comparison of fault-tolerance concepts for IEEE 802.1 time sensitive networks (TSN)," in *IEEE Emerging Technology and Factory Automation (ETFA)*, Barcelona, Spain, Sept. 2014, pp. 1–8. doi: 10.1109/ETFA.2014.7005200.

[31] P. Pop, M. L. Raagaard, S. S. Craciunas, et al., "Design optimisation of cyber-physical distributed systems using IEEE time-sensitive networks," *IET Cyber-Physical Systems: Theory & Applications*, vol. 1, no. 1, pp. 86–94, Dec. 2017. doi: 10.1049/iet-cps.2016.0021.

[32] S. M. Laursen, P. Pop, and W. Steiner, "Routing optimization of AVB streams in TSN networks," *Acm Sigbed Review*, vol. 13, no. 4, pp. 43–48, Nov. 2016. doi: 10.1145/3015037.3015044.

Manuscript received: 2018-01-14

Biographies

FU Shousai (16121438@bjtu.edu.cn) received his bachelor's degree in electrical engineering from Inner Mongolia University of Technology, China. Now he is a master candidate in electrical engineering at Beijing Jiaotong University, China, and he engages in research of fieldbus and industrial Ethernet.

ZHANG Hesheng (hszhang@bjtu.edu.cn) received his B.S.E.E. and M.S. degrees in electrical engineering from Northern Jiaotong University of China in 1992 and 1995, respectively. He received his PH.D. degree in automation science and technology from Tsinghua University, China in 2006. He is now a professor with School of Electrical Engineering, Beijing Jiaotong University, China. His research interests include fieldbus, sensor network, network communication performance, and intelligent control. He is a senior member of the CES and CCF. Dr. ZHANG has published more than 60 papers in the journals and conferences as the first or second author, and has applied for nine national invention patents as the first inventor.

CHEN Jinghe (16121422@bjtu.edu.cn) received her bachelor's degree in electrical engineering from Jinan University, China. Now she is a master candidate in electrical engineering at Beijing Jiaotong University, China, and she engages in research of fieldbus and industrial Ethernet.

ZTE Communications

Table of Contents, Volume 16, 2018

Volume-Number-Page

SPECIAL TOPIC

Wireless Data and Energy Integrated Communication Networks

Ultra-Dense Networking Architectures and Technologies for 5G

Next Generation Mobile Video Networking

Security and Availability of SDN and NFV

ZTE Communications

Table of Contents, Volume 16, 2018

RESEARCH PAPER

REVIEW

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

ZTE COMMUNICATIONS

中兴通讯技术（英文版）

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

ZTE COMMUNICATIONS
Vol. 16 No. 4 (Issue 64)
Quarterly
First English Issue Published in 2003

Supervised by:
Anhui Science and Technology Department

Sponsored by:
Anhui Science and Technology Information Research Institute;
Magazine House of ZTE Communications

Published and Circulated (Home and Abroad) by:
Magazine House of ZTE Communications

Staff Members:
Editor-in-Chief: WANG Xiang
Executive Associate Editor-in-Chief: HUANG Xinming
Editor-in-Charge: ZHU Li
Editors: XU Ye and LU Dan
Producer: YU Gang
Circulation Executive: WANG Pingping
Assistant: WANG Kun

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

Annual Subscription: RMB 80

Printed by:
Hefei Tiancai Color Printing Company

Publication Date: December 25, 2018

Publication Licenses:

ISSN 1673-5188
CN 34-1294/ TN