Advances in Intelligent Systems Research, volume 154
6th International Conference on Measurement, Instrumentation and Automation (ICMIA 2017)

Review about Software Defined Networking

HAN Wei-Jie[1,2, a], Xue Jing-Feng[1,b]
[1] School of Software, Beijing Institute of Technology, China
[2]Department of Information Equipment, Equipment Academy, China
[a]bit_hwj2016@126.com, [b]xuejf@bit.edu.cn

Keywords: Software defined networking; OpenFlow; Security challenges; Functions decoupling
Abstract. Software defined networking (SDN) has pushed the rapid progress of the network by decoupling the control plane from the data plane thus enabling the dynamism and flexibility of the network compared with the traditional static network architecture. This paper summarizes the theories and the techniques of SDN. Firstly, the concept and architecture of SDN are reviewed. Secondly, the key protocol and some practical applications of SDN are introduced. Finally, we discuss about the main security challenges faced by SDN. The current and future research trends about SDN mainly include the security enhancement and standardization of the key components of SDN.

Introduction

Software-defined networking (SDN) is a newly approach to build a computer network that enables the administrators to manage network services through the abstraction of low-level functionality[1]. Compared with the traditional static network architecture, SDN support the dynamic, scalable computing and storage needs of modern computing environments by decoupling the traffic sending system (the control plane) from the traffic forwarding system (the data plane). The above dynamic architecture is the most attractive advantage that we can depend on so as to solve the current networking problems.

SDN has been applied in many various domains successfully due to the OpenFlow protocol which was proposed in 2011[2]. This protocol realizes remote communication between the network plane elements to determine the path of network packets across network switches. However, many different techniques have been put forward since 2012 which include Cisco Systems' Open Network Environment[3] and Nicira's network virtualization platform[4]. SDN can be constructed and implemented in various ways based on the above different techniques.

WHY WE NEED A NEW NETWORK ARCHITECTURE?

With the developing trends of modern computing services such as mobile devices and content, server virtualization, and cloud services, the traditional network architecture should be re-examined and modified simultaneously. The conventional network architecture is hierarchical usually in which the Ethernet switches are arranged in a tree structure. This architecture is static in essence which could satisfy the needs of the traditional client-server computing, but not the dynamic computing and storage needs of the modern enterprise data centers.[5]

The need of SDN is driven by the some of the following key computing developments which include:

(1) The traffic patterns have been changed significantly.

In modern network environments, the traffic patterns have changed significantly especially within the enterprise data center. In contrast to the traditional client-server applications where the bulk of the communication often occurs between the clients and the servers, the current applications access different databases and servers which creates massive machine-to-machine traffic before returning the data to the end user devices. At the same time, the network traffic patterns on the user side have been changed obviously because the users may access the network contents and applications from any type of devices, anywhere, and at any time. Furthermore, the traffic patterns on the managers' side

have also been changed because many enterprise data center managers are driven to apply some novel computing models due to the emergence of cloud computing and the massive traffic across the network.

(2) A more efficient network service strategy should be developed.

With the increasing and wide use of personal electric devices, the IT enterprises are under more heavy pressure than before because they have to provide necessary network services for these devices in a fine-grained manner and ensure the security of the service process. In order to satisfy the network requirements, many enterprises have deployed different kinds of cloud services such as public and private cloud services, and hybrid cloud services. In this paradigm, the complexity of the service strategy will be increased dramatically because the enterprises should supply services dynamically and securely on demand. It means that a more efficient network service strategy should be developed for the above service requirements.

(3) The massive network resources need a more efficient management strategy.

In the cloud computing environment, the complexity of managing the computing and storage resources will be increased for the following reasons: 1) The requirement of elastic scaling of computing, storage, and network resources; 2) Handling the large-scale datasets requires massive parallel processing on thousands of servers; 3) The network capacity is rising constantly due to the rise of mega datasets; 4) The capacity for supporting the daunting network processing tasks by the data center maybe exceed the ultimate limit. Thus, a more efficient management strategy should be developed to solve the above problems aiming to manipulate the network resources efficiently and effectively.

CONCEPT AND ARCHITECTURE OF SDN

SDN is an architecture which is suitable for the high-bandwidth, dynamic nature of the modern applications and characterized by the dynamic, manageable, cost-effective, and adaptable feature. SDN architecture reaches the demands of the massive network services by decoupling the network controlling and forwarding functions, and makes the network controlling process directly programmable.[6,7]

The key characteristics of SDN architecture consists of:

(1) Directly programmable

The process of network control can be programmed directly because it is decoupled from the forwarding functions.

(2) Agile

The administrators can adjust the network-wide flows on changing demand because network control is abstracted from the forwarding functions.

(3) Centrally managed

The SDN network can be managed centrally because there exists a centralized logical controller which maintains a global view of the network. Therefore, the functions can be managed centrally through the single and logical management switch.

(4) Programmatically configured

The network managers can configure and manage the network resources dynamically and automatically because the process can be implemented by writing SDN programs without proprietary software support.

(5) Simply implementation

The design and operation of SDN is simply because it is implemented by open standards and the instructions are provided by SDN controller directly instead of proprietary devices and protocols.

The architecture of SDN mainly consists of the following key components[8]:

(1) The data plane

The data plane consists of network elements such as router and switch. The network elements are linked by the SDN datapath which is constructed according with various disciplines.

(2) The control plane
The control plane consists of the logical centralized controller in charge of running the control logic strategies and maintaining a global view of the network. The network services are abstracted from the global view of the network by the controller. The network datapath can be called by accessing the agent of the Control to Data-Plane Interface. Furthermore, the controller provides easy-to-use Northbound Interfaces for the operators and researchers in order to enable them to realize customized applications and manage the network logically.
(3) The application plane
The application plane consists of various kinds of SDN-based applications. The user can deploy new applications quickly by programming simply without worrying about the technique details of the underlying infrastructures.
(4) The Control to Data-Plane Interface (CDPI)
The SDN CDPI is the interface defined between an SDN Controller and an SDN Datapath, which delivers the forwarding rules from the network operating system to the network devices without influence on the control plane and its belonged logics. One unique advantage of CDPI lies in that the CDPI is implemented in an open, vendor-neutral and interoperable way.
(5) SDN Northbound Interfaces (NBI)
SDN NBIs are interfaces between SDN Applications and SDN Controllers and typically provide abstract network views and enable direct expression of network behavior and requirements. The NBI enables a third party to develop management software and application on demand and provides much more choices for the managers. Additionally, the characteristic of network abstraction enables the users to choose different network operating systems on demand without influence on the normal running of the physical devices.

The key protocols for SDN--OpenFlow

There exist a number of protocols for constructing a SDN architecture among which OpenFlow is the first standardized protocol accepted by the network industry. OpenFlow is a communication protocol that gives access to the forwarding plane of a network switch or router over the network which enables network controllers to determine the path of network packets across a network of switches[9]. The unique characteristic lies in that the controllers are distinct from the switches. This separation of the control from the forwarding allows for more sophisticated traffic management than the access control lists (ACLs) and routing protocols. Also, OpenFlow allows switches to be separated from different vendors in order to be managed remotely using a single, open protocol. The protocol's inventors consider OpenFlow an enabler of software defined networking (SDN).

Key components of OpenFlow network

The OpenFlow network consists of three parts: the OpenFlow switch, the FlowVisor and the Controller. The OpenFlow switch realizes the forwarding function of the traffic flow; the FlowVisor realizes the virtualization of the network; the Controller realizes the controlling function by managing the network centrally.

（1）The OpenFlow switch[10]

The OpenFlow switch is the core of an OpenFlow network which realizing the forwarding functions of the traffic flow. After receiving a data packet, the switch looks up for the forwarding destination port in its local flow table firstly. If no matches are found, the packet will be sent to the Controller which will decide the forwarding port.
A typical OpenFlow switch is composed of the flow table, the secure channel and the OpenFlow protocol.
1) the secure channel
The secure channel is the interface between the OpenFlow switch and the controller. The controller not only controls and manages the switches, but also receives the event notifications from the

switches and sends packets to them by the interface. The switch communicates with the controller through the channel according with the specification of the OpenFlow protocol.

2) the OpenFlow protocol

The OpenFlow protocol plays the role of setting standards for information exchange and interface between the switch and the controller. The core part of the protocol is a set of information structure for the OpenFlow protocol.

The OpenFlow protocol supports three kinds of information type: Controller-to-Switch, Asynchronous and Symmetric which of them consists of multiple subtypes. The Controller-to-Switch information is initiated from the controller and used to detecting the status of the switch; the Asynchronous information is initiated from the switch and used to update and modify the status of the controller; the Symmetric information is initiated from the controller or the switch without any network request.

(2) the FlowVisor[11]

In contrast to the computer virtualization, the FlowVisor plays the role of a network virtualization layer between the hardware and the software. It can be realized that a number of controllers control one OpenFlow switch by the OpenFlow, but each of them can only control one slice which passes the OpenFlow switch. Therefore, multiple network experiments can run in different virtualized network environments at the same time without any influence on the forwarding speed of the business traffic by FlowVisor. One of the advantages of FlowVisor lies in its compatibility with common commercial switches without any proprietary programmable hardware.

(3) the Controller[12]

The OpenFlow realizes the separation the data plane from the control plane because that the forwarding function of the data plane is realized by the OpenFlow switch and the controlling function of the control plane is realized by the Controller. The Controller controls the flow tables within the OpenFlow switch by the standard interface provided by the OpenFlow protocol so as to control the overall network centrally. All the above functions of the Controller are realized by NOX which are just like the operating system of the OpenFlow network. Moreover, some other applications such as Plug-n-server, OpenRoads and OpenPipes can also run on NOX.

Structure of the flow table

The OpenFlow pipeline of every OpenFlow switch contains multiple flow tables, each flow table containing multiple flow entries. The OpenFlow pipeline processing defines how packets interact with those flow tables. An OpenFlow switch with only a single flow table is valid, in this case pipeline processing is greatly simplified.

A flow table consist multiple flow table items which are the forwarding rules. Whenever a switch receives a packet, it will look up for the destination port in the flow table. A flow table item is composed of the header field, the counter and the operators. The header field is a ten-element set which is the label of the flow table item; the counter is meant to count the number of the flow table items; the operators consist of the next operation of the matched packet. The process of packet forwarding by the flow table is shown as Fig. 1.

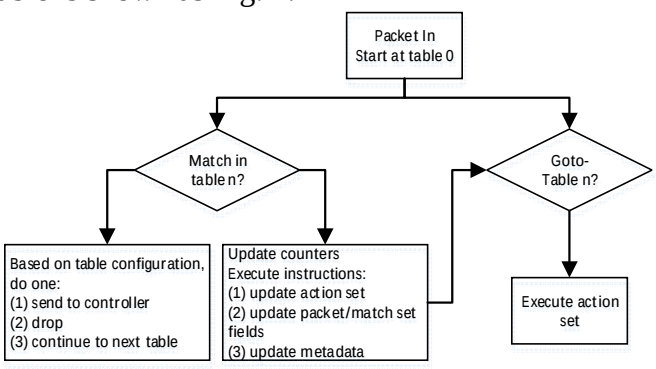

Fig. 1 The process of packet forwarding by the flow table

APPLICATIONS OF SDN

SDN has been applied in a number of various fields practically because of its advantages described in the above chapters. The typical applications of SDN include:

（1）SDMN

Software-defined mobile networking (SDMN)[13] is an approach by which the SDN techniques are utilized into the design of mobile networks. SDMN is characterized by the feature that it is implemented in software so as to maximize the use of commodity hardware and software in the realization of the network. SDMN is considered to be an extension of SDN paradigm by incorporating mobile network specific functions based on the SDN techniques.

（2）SD-WAN

A software defined-wide area network (SD-WAN)[14] is constructed by the principles and techniques of software-defined networking. The management efficiency of SD-WAN could be improved by the centralized control emphasis and the cost be lowered by replacing the expensive parts by the SDN virtualization approach. In addition, the control and management process is separated from the hardware by configuring and administrating the process easily based on its central controller.

（3）Security ensured by SDN

The network-related security of the SDN-based applications may be enhanced or enabled because the advantageous features of the SDN architecture which include the global view of the network, reprogram the data forwarding process and manage the control centrally. The advantages provided by SDN include:

1) Detection of Distributed Denial of Service, botnet and worm propagation can be realized more easily than before for the security applications built upon the SDN controller, because we can collect network statistics from the forwarding plane of the SDN network and apply classification algorithms on these statistics so as to detect any network anomalies[15].

2) It is easier to protect the key properties in the network from being attacked by hiding or changing these software and hardware based on some efficient moving target defense algorithms. The algorithms are implemented based on the centrality of the SDN controller. For instance, we can assign some virtual IPs or ports to the hosts within the network in order to make the attackers confused.

3) There still exist several additional security advantages enabled by the SDN architecture. For example, FlowVisor enables to use a single hardware forwarding plane while sharing multiple separated logical networks. The hardware resources can be utilized for multiple purposes simultaneously in the same way. Furthermore, the network resources can be divided into multiple slices which can be delivered to different users. Then the users can carry out a larger number of tasks in parallel by using their own slice.

Security Challenges faced by SDN

The security challenges in the SDN network can be categorized into five kinds of security problems according to the logic separation structure of the control function from the forwarding function[16,17,18]. The topic is illustrated as Fig. 2.

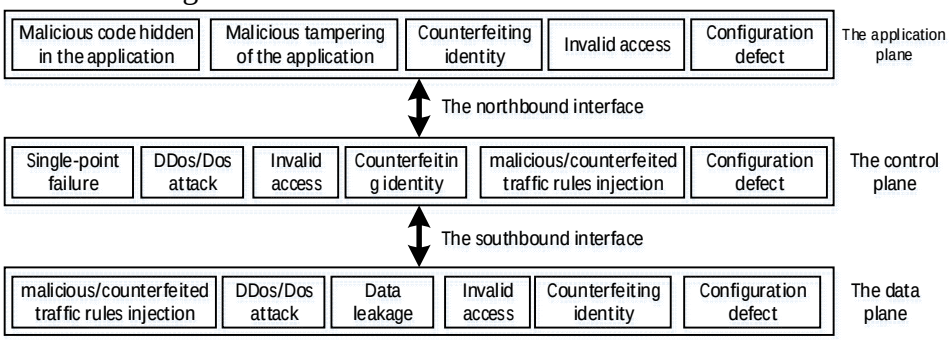

Fig. 2 Security challenges faced by SDN

(1) the security challenges of the application plane

The security protection mechanisms for the application plane are still imperfect because the switches and other network devices in the data plane receive and forward the packets from the control plane without any doubt. However, the application plane maybe encounters some unexpected security damages once the applications are attacked and tampered which have involved in setting the forwarding rules. In this way, the security threats faced by the application plane consist of the following aspects: the malicious code hidden in the application, the malicious tampering of the application, counterfeiting identity, unauthorized access and the configuration defect of the application itself, et al.

(2) the security challenges of the control plane

The control plane plays the key role of being the bridge between the application plane and the data plane. However, the control plane maybe the weakest section of the whole SDN security chain. The most typical security problem is the single-point failure due to the centralized management and control which lies in the following two aspects: ① the DoS/DDos attack: If encountered with massive invalid accesses from the attackers beyond its load-balance capacity, the controller will spare no time to deal with the normal accesses and looks like unavailable anymore for the valid users. ② the controller is damaged logically or physically which means the key controllers are damaged logically or physically resulting that the valid network requests maybe denied by the controller. The additional security problems faced by the control plane consist of invalid access, counterfeiting identity, malicious/counterfeited traffic rules injection and the configuration defect of the controller itself so on.

(3) the security challenges of the data plane

The data plane is composed of the basis devices such as switch and router so on which is in charge of handling and forwarding data, and collecting the status. There exists a security defect in its function that receives the flow rules from the controller without any doubt. Consequently, the plane will be faced with the following security threats: malicious/counterfeiting flow injection, DDoS/Dos attack, data leakage, invalid access, counterfeiting identity and the configuration defect of the switch itself so on. Furthermore, this plane maybe faced with some additional security threats such as the flow table chaos due to disordered control commanders from the counterfeited controller.

(4) the security challenges of the southbound interface

The security challenges of the southbound interface are essentially caused by the vulnerability within the OpenFlow protocol. The flow data in the OpenFlow channel is usually encrypted based on SSL/TLS. However, the southbound interface will be faced with some security threats for instance eavesdropping and counterfeiting the controller due to the imperfect security property of SSL/TLS and taking no safeguard of the channel.

(5) the security challenges of the northbound interface

The standardization of the northbound application programming interface has become a hot topic currently. The authentication method and granularity has not yet been unified because of the various types and constantly updating of the applications. Moreover, the trustworthy relationship between the northbound applications and the controller is more fragile than the southbound interface. At present, the security threats faced by the northbound interface mainly include: invalid access, data leakage, message tampering, counterfeiting identity and the internal and external vulnerabilities of the applications themselves so on.

Conclusions

In order to satisfy the requirements of managing large-scale and complex computation and storage, SDN has become one of the most important approaches by decoupling the control plane from the data plane. We have discussed about not only the architecture of SDN and some key implementation technologies, but also the security challenges faced by SDN. Currently, the research trends about SDN mainly include[19, 20]: (1) Design and development of the new-type security-oriented

controller/network operating system; (2) Standardization of the security protocol for the northbound interface; (3) Detection and defense of the Dos/DDos attack for the controller.

Acknowledgements

This work was financially supported by the Key Technology Research Project of the General Armaments Department (Grant No. 2015ZD004025) and National Key Research & Development Program 2016(2016YFB0801304).

References

[1] D.L. Tennenhous and D.J. Wetherall. Towards an active network architecture. Proceedings of IEEE DARPA Active networks conference and exposition, 2002, p 2-15.
[2] Tomas A. Limoncelli. Openflow: a radical new idea in networking, ACM Communications. 2012, 55(8):42-47.
[3] Cisco Open Network Environment: Bring the Network Closer to Applications. White Paper, Cisco, 2015.
[4] Network virtualization platform.https://en.wikipedia.org/wiki/Network_virtualization_platform.
[5] Idris Z. Bholebawa, Bakesh Kumar Jha, Upena D. Performance analysis of proposed network architecture: OpenFlow vs. traditional network, International Journal of Computer Science and Information Security. 2016, 14(3):30-39.
[6] Fei Hu, Qi Hao, Ke Bao. A Survey on Software-Defined Network (SDN) and OpenFlow: From Concept to Implementation, IEEE Communications Surveys and Tutorials, 2014, 16(4):2181-2206.
[7] Nunes BAA, Mendonca M, Nguyen XN, et al. A survey of software-defined networking: past, present, and future of programmable networks, IEEE Communications Surveys and Tutorials. 2014, 16(3):1617-1634.
[8] SDN architecture overview. Open Networking Foundation, 2014.
[9] Adrian Lara, Anisha Kolasani, Byrav Ramamurthy. Network Innovation using OpenFlow: A Survey, IEEE Communications Surveys & Tutorials. 2014, 16(1):493-512.
[10] OpenFlow Switch Specification. Open Networking Foundation. 2014.
[11] Rob Sherwood, Glen Gibb. FlowVisor: a network virtualization layer, CiteSeer, 2009.
[12] Alexander Shalimov, Dmitry Zuikov, Daria Zimarina et al. Advanced study of SDN/OpenFlow controllers, Proceedings of the 9th Central & Eastern European Software Engineering Conference, 2013, p 1-6.
[13] K Pentikousis，Y Wang，W Hu. Mobileflow: Toward software-defined mobile networks, IEEE Communications. 2013, 51(7):44-53.
[14] R Ahmed，R Boutaba. Design considerations for managing wide area software defined networks, IEEE Communications. 2014, 52(7):116-123.
[15] SA Mehdi，J Khalid，SA Khayam. Revisiting Traffic Anomaly Detection Using Software Defined Networking, Proceedings of the 14th International conference on Recent Advances in Intrusion Detection. 2011, p 161-180.
[16] Wang Meng-Meng, Liu Jian-Wei, Chen Jie et al. Software-Defined Networking: Security Model, Threats and Mechanism, Journal of Software, 2016, 27(4):969-992.
[17] Threat Analysis for the SDN Architecture. Open Networking Foundation. 2016.
[18] Principles and Practices for Securing Software-Defined Networks. Open Networking Foundation. 2015.
[19] Shin S, Song Y, Lee T, et al. Rosemary: A robust, secure, and high-performance network operating system, Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. 2014, p 78-89.
[20] Cui JS, Guo C, Chen L, et al. Establishing process-level defense-in-depth framework for software-defined networks, Journal of Software, 2014, 25(10):2251-2265.