Research on network traffic acquisition based on in-band telemetry technology

Lili Cheng[a], Pengbo Xiao[b], Lisha Gao[a], Pan Zhou[a], Yanqi Liu*[a]
[a]Department of Mechanical and Electrical Engineering, Wenhua College, Wuhan, Hubei, China;
[b]Agricultural Bank of China R&D Center Wuhan R&D Department, Wuhan, Hubei, China

ABSTRACT

Network measurement technology is the basis of the monitoring network devices, link and flow running state and flow behavior model. The current network measurement mainly adopts the method of active measurement, in other words, managers use injection flow probe to measure the network actively, but this method can increase the network flow and influence the state of the network and the network load. To solve this problem, this paper proposes a method that is fine-grained network traffic parameter measurement based on in-band network telemetry technology. Software defined network architecture and the principle of data plane programmable technology is introduced first, and then the process of using in-band network telemetry to obtain network traffic parameters is described in detail. Finally, this paper simulates the network environment and attack environment in the simulation environment to obtain the data set required for the experiment, and the results of the data set are analyzed to verify the effectiveness of this method based on in-band network telemetry technology.

Keywords: Software defined network, in-band network telecommunication, DDoS attack detection, DDoS attack prevention

1. INTRODUCTION

Software defined network is a new type of network architecture. The core idea of this architecture is to decouple the control plane and data plane in network equipment[1]. The SDN architecture is shown in Figure 1.

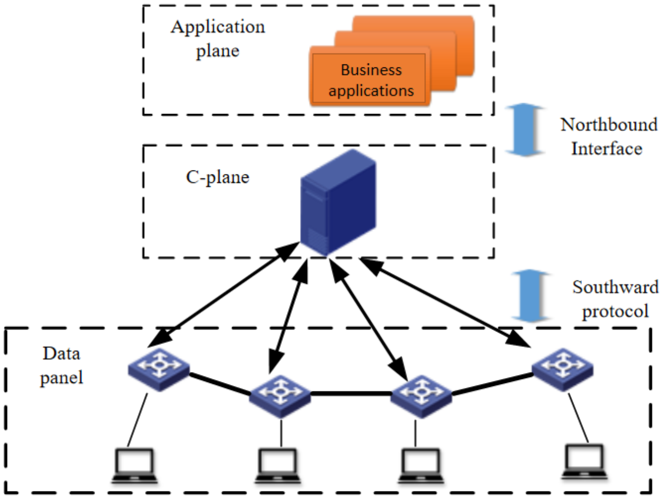


Figure 1. Software defined network architecture.

The top layer of the software defined network architecture is the application plane layer, which includes various SDN applications and services. These applications and services can interact with the controller of the control plane through the

[*] chenglilly@163.com

Third International Conference on Computer Science and Communication Technology (ICCSCT 2022)
edited by Yingfa Lu, Changbo Cheng, Proc. of SPIE Vol. 12506, 125061K
© 2022 SPIE · 0277-786X · doi: 10.1117/12.2661968

northbound API, and communicate the network behavior request to the controller in a programmable way; The control plane is composed of a controller written in software. The control plane provides the state and event model of the underlying network for the SDN application of the application plane. When the device state of the data plane changes, the controller quickly updates the State Library[2] through the southbound protocol, and converts the strategy of the application plane into flow rules and generates a flow table item to be sent to the data plane. The data plane is composed of the underlying general switch, which is responsible for forwarding, modifying, discarding and other operations of data packets. SDN decouples the control function from the forwarding function, and greatly improves the degree of freedom of network operation control by separating the control plane and the data plane. With the further research of software defined network, the programmable ability is extended to the data plane[3], which allows the nodes in the network to perform lightweight operations.

1.1 Research on programmable principle of data plane

The traditional network switch equipment is limited by the manufacturers' consideration of commercial interests. Without an open network telemetry interface, it can only be treated as a black box. The network administrator can only indirectly obtain the network telemetry information from the network edge switch by means of bypass image or network probe. The data plane programmable technology makes the switch customizable, so that programmable switches are not limited to manufacturers and can freely define the data processing logic of switches according to specific services.

The realization of data plane programmable technology is closely related to the programmable language P4. P4 is a cross platform language which can be ported to different platforms through compilers. Its reconfigurable features enable developers to dynamically modify the code after it is compiled and deployed to the platform. P4 is not bound to a specific network protocol. Developers can write protocols that conform to their own rules according to the syntax rules, or delete redundant protocols[4]. The abstract forwarding model of P4 language is shown below in Figure 2.

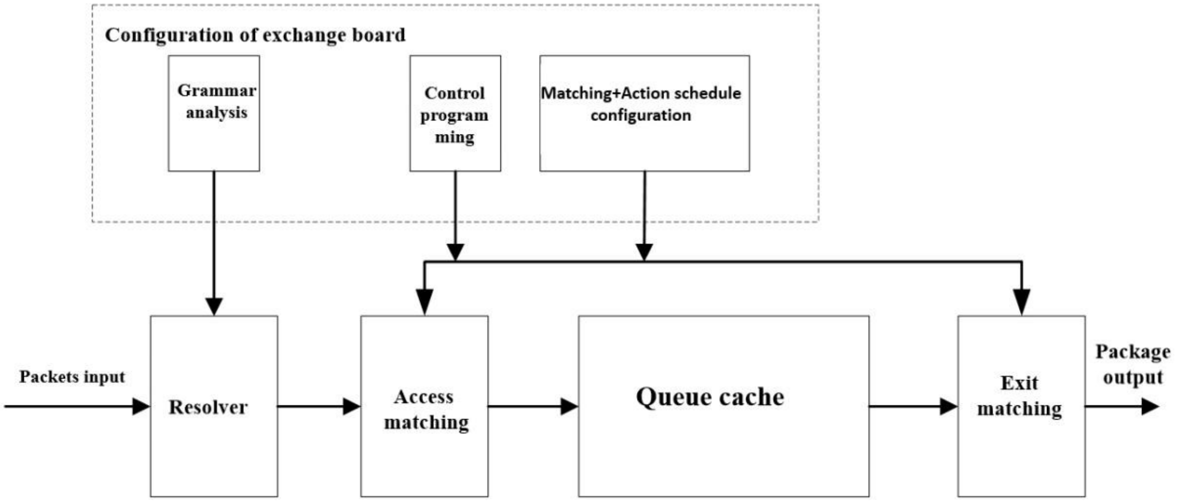


Figure 2. P4 programming abstract forwarding model.

The first forwarding model is the data packet header parser[5]. With the help of the characteristics of P4 language, the parsing rules are customized according to the requirements of different message formats. After compilation, the parsing state transition diagram parsed from top to bottom according to the protocol types of each layer of Ethernet will be generated, as shown in Figure 3. The generated analytic state transition diagram to the data message header parser is deployed first in the forwarding model, so that when the data message enters the switch from the input end, the parser will parse the message header and allocate the message header to the corresponding custom message field according to the rules of analytic state transition diagram. The parsed message segment will enter the subsequent matching action part.

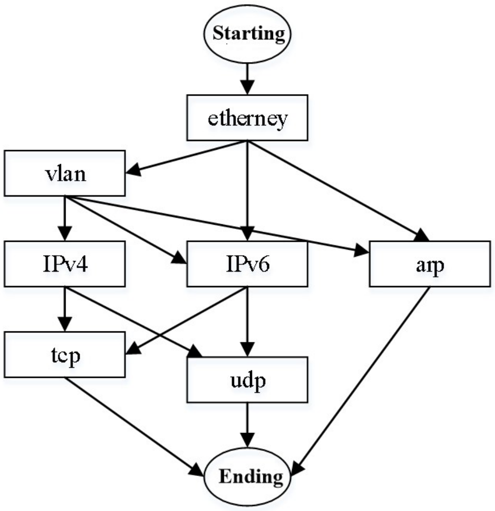


Figure 3. Analytic state transition diagram.

The matching part is the access matching action and exit matching action in Figure 2. After compiling P4 code, the matching action table will be generated. Entry matching modifies the packet and determines the exit and queue of the packet. Export matching has fewer function of modifying data packets than the former, and is only responsible for modifying data packets. The last part is the control program, which sends or installs the control flow table and other configuration information to the switch through the code interface.

1.2 Research on the principle of in-band network telemetry technology

Based on the data plane programmable technology, this paper will redesign the programmable multi-stage pipeline and add In-band network telemetry module in the data plane layer. This passive network telemetry technology can directly collect and report network telemetry information to the monitor in the data plane, without additional communication overhead with the control plane.

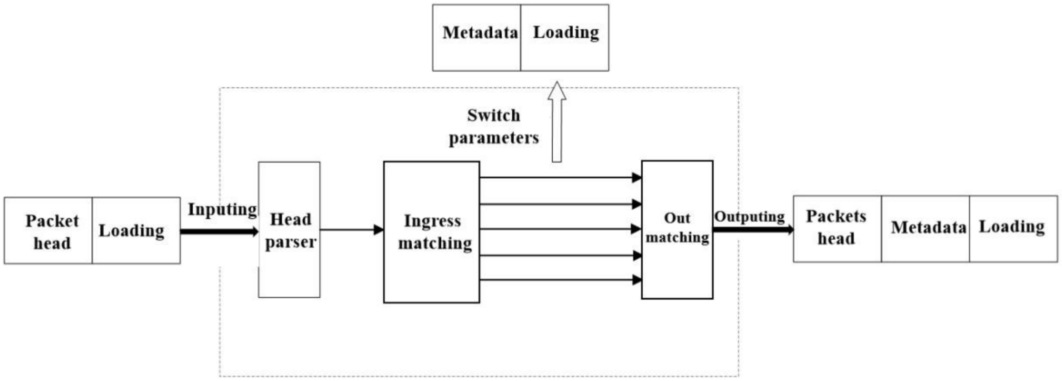


Figure 4. Switch metadata parsing process.

In-band network telemetry technology is based on data plane programmable technology. Developers can customize the process of message forwarding on the data plane by using the characteristics of P4 language. After the data message is sent to the programmable switch, the header parser will parse the message header[6], and the message header will be parsed into metadata. The parsed metadata will determine the queue by comparing the control flow table issued by the control application through the interface by means of the access matching process, and then the exchange opportunity updates the packet header of the data message. The exchange opportunity writes the current status parameter information obtained internally into the packet header. Finally, the repackaged packet header will be forwarded from the specified port through the exit matching. Therefore, the data message using In-band measurement technology will have its real-time status parameters passing through the switch. The process of writing metadata for In-band measurement is shown in Figure 4.

When designing the message parsing process, the use of In-band network telemetry technology enables each packet to bring out the current status parameter information of the switch which it passes through. Each switch in the network uses P4 language to turn on the In-band telemetry function, and the packet header after the packet passes through these switches will contain the switch status parameter data on the whole path. When reaching the last hop switch, the header of the data packet will pop up and be restored to the original message. The pop-up metadata will be re-encapsulated into a message, which will be sent and output through the designated port after that the matching flow table is generated. In the process, the data packet acts as a carrier. The metadata is added from the first switch to the last hop switch. The whole process is shown in Figure 5 below. Switches S1 to SN are configured with In-band network telemetry function.

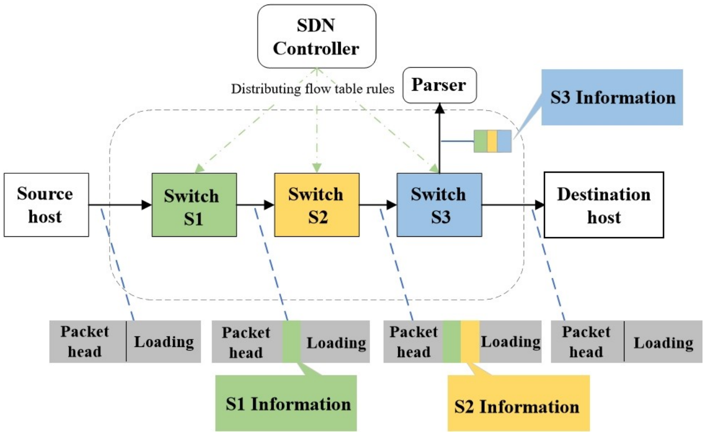


Figure 5. Measurement process of in-band network telemetry technology.

The principle and relationship of data plane programmable technology and In-band network telemetry technology are described above. The type of metadata of key switch state parameters in In-band network telemetry technology is affected by manufacturers. The metadata that can be extracted by the experimental software switch in this paper is shown in Table 1 below. In addition to the switch state parameters, it can also obtain traditional “five tuples” data (including source IP address, destination IP address, source port, destination port and transport layer protocol). The switch state parameters obtained by In-band network telemetry technology can reflect the network state in real time, which is suitable for obtaining DDoS attack detection data in this paper. This paper will use the metadata obtained by in-band network telemetry and five tuples to extract features for the subsequent detection process. The specific feature extraction method is described in the next chapter.

Table 1. Metadata types.

Metadata type	Metadata	Describe	Remarks
1	Ingresport	Inbound slogan	Device node
2	Egressport	Outbound slogan	Device node
3	Ingresstimestamp	Time tag of incoming switch	Device node
4	Egresstimestamp	Outgoing switch time tag	Device node
5	Switchid	Switch identifier	Device node
6	Hoplatency	Dwell time in switch	Device node
7	Linklatency	Link delay	link
8	Ingressqueuedepth	Number of packets queued when entering the queue	Device node
9	Egress queue depth	Number of packets queued when leaving the queue	Device node

In addition, the parameters measured in the band and written to the packet header will not be ejected until the destination switch. For arranging the ejected metadata in order, it is required to record the parameter information on a path in packets and output them in the order of time when they arrive at the destination switch, so that all status parameter data will be arranged in order of time. The pop-up metadata output form is shown in Table 2 below.

Table 2. Metadata output form.

Packet ID	Upstream switch ID	Downstream switch ID	Metadata	Quintet
0	Switch 7	Switch 0	……	……
0	Switch 5	Switch 7	……	……
0	Switch 2	Switch 5	……	……
0	Switch 1	Switch 2	……	……
1	Switch 7	Switch 0	……	……
1	Switch 6	Switch 7	……	……
1	Switch 3	Switch 6	……	……
1	Switch 1	Switch 3	……	……
……				

The in-band network telemetry technology is realized on the programmable data plane, which does not need the resources of the control plane, and is realized by the control plane in the whole process, so that the data plane has the end-to-end network parameter measurement ability, and the real-time fine-grained network metadata and traffic path information obtained can effectively support the detection and defense of DDoS attacks. This paper will further explain the data acquisition method of In-band network telemetry technology and its application in attack detection and defense in the following chapters.

2. RESEARCH ON THE PRINCIPLE AND CHARACTERISTICS OF DDOS ATTACK

Before DDoS attack detection, it is necessary to study the principle and characteristics of DDoS attack firstly, and provide a theoretical basis for subsequent detection algorithms to extract the corresponding features.

2.1 DDoS attack principle

DDoS attack is one of the most frequently used attack methods among all network attack methods[7]. The attacker uses the controlled "broiler"[5] to attack the target of the victim host, to consume the cup resources of the victim host and the bandwidth resources of the network link, to block the normal transmission process of legal packets. The principle of DDoS attack is shown in Figure 6.

According to the results of the literature survey, the most common types of DDoS attacks are divided into three categories[8,9], namely SYN Flood attack, UDP flood attack and ICMP flood attack:

- SYN Flood Attack. SYN Flood mainly uses the defects existing in the traditional TCP/IP three-time handshake process to send false SYN messages by constructing the original socket. The false messages in the message mainly replace the source address in the message, so that the target server will not complete the three-time handshake with the client, because the false messages will fill the queue with the protocol of the target server. At this time, the server resources cannot be released[10], resulting in the failure of normal users to effectively transmit information with the target server.

- UDP Flood Attack. UDP flood attack uses the connectionless feature of UDP protocol[11] to consume the resources of the target host by sending UDP packets with random source IP. The data packets sent by UDP flood are smaller, and the number of small packets will be more under the same traffic. Because the network link equipment needs verify the data packets in the forwarding process, a large number of small packets can effectively consume the resources of network equipment, resulting in service congestion.

- ICMP Flood Attack. ICMP attacks were widely used in the early years. Its principle is to continuously send a large number of ICMP type data packets to the target host, and to consume the resources of the target host, in the meantime attackers use forged IP addresses to launch attacks on the victim host[12].

According to the statistics of several famous large-scale DDoS attacks[13], the attack types contain 47% of SYN Flood, 31% of UDP flood, 12% of ICMP flood, and 10% of other attack types.

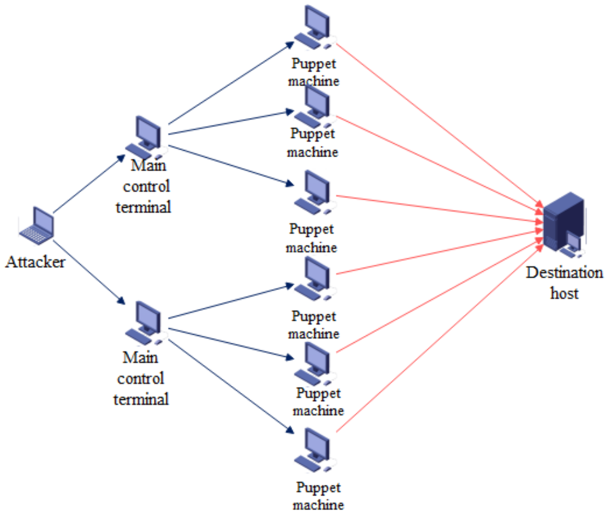


Figure 6. DDoS attack principle.

2.2 DDoS attack characteristics

DDoS attacks have characteristics that are obviously different from normal traffic:

- The target addresses of DDoS attacks are very centralized. Due to the increasing CPU computing power and network bandwidth of the device, it is difficult for a general DDoS attack to pose a threat to the network host. In addition, DDoS attacks not only consume the resources of the target host, but also consume a lot of resources of the attacker. The attacker must focus on the attack purpose to achieve the attack effect[13].

- The source address of DDoS attack is quite discrete. When launching a SYN attack, in order to prevent the attacking host from receiving a large number of response messages returned by the target host, the attacker will forge the attack source IP to avoid attack reflection. In addition, in order to evade tracing, attackers will also use forged IP addresses.

- The packet size of DDoS attacks is relatively small and evenly distributed. Attackers use smaller packets to send a large number of attack packets at high speed in a short time, increasing the processing pressure of network devices.

3. FLOW SIMULATION EXPERIMENT AND RESULT VERIFICATION

3.1 Experimental environment

The data acquisition experiment in this paper is carried out by simulation. The host CPU of the experimental server is Intel Xeon e5-2620 V3, with 64G memory and the operating system is Ubuntu 16.04[14]. The software defined network environment is simulated by mininet simulator. The data plane is composed of seven bmv2 programmable switches S1 to S7 interconnected. The controller version is onos113. The data plane communicates with the controller through P4 runtime protocol. The generation of background traffic is based on Python's Python library. DDoS attack traffic is generated through the hping3 tool. The attack host is A1 and the path is S1-S2-S5-S7. The topology generated by traffic is shown in Figure 7.

Since there is no public In-band network telemetry data set yet, this paper simulates the network environment and attack environment in the simulation environment to obtain the data set required for the experiment. The hosts on S1, S2 and S3 in the topology is defined as the traffic sending end, and packets is randomly sent at an average rate of 100 packets per second. The average packet size is about 400 bytes. In the packet data, TCP packets account for 80%, UDP packets account for 20%, and the destination host is the host on S7. An attack host A1 is added to S1, and the host console is accessed through xterm, then the attack traffic in flood mode is sent by using the hping3 tool (greater than the rate of 100 packets per second). The attack target is the host on S7, and two high-intensity attacks have been carried out in a period of time.

This paper uses bmv2 software switch to simulate the programmable P4 switch. First, the simulation setting step is to configure the Jason file of P4, and to start the in-band network telemetry on the link switch. When the flow of the host targeted at S7 reaches S7, the in-band network telemetry data will automatically pop up. Finally, the metadata obtained will be parsed through the parsing script. The metadata forwarded can be obtained in real time at the listening end.

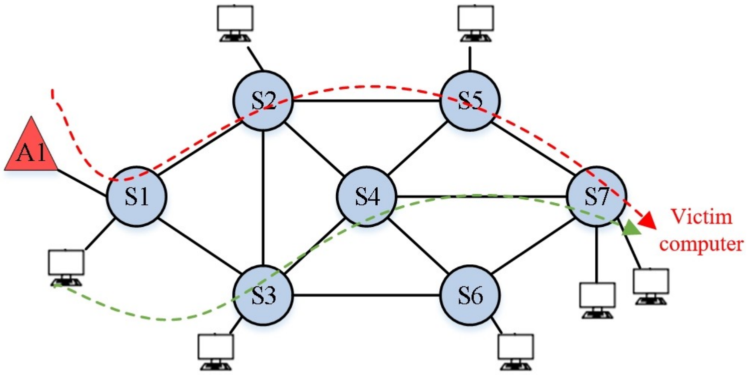
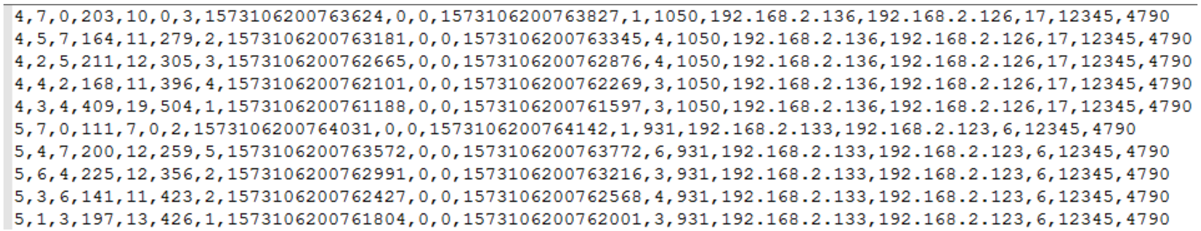


Figure 7. Topology of flow generation experiment.

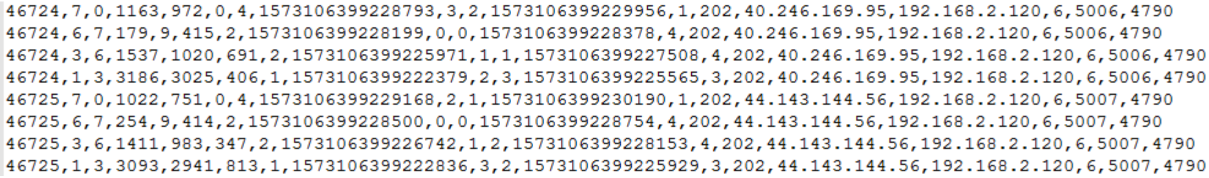
3.2 Results and analysis

The results of In-band network telemetry are stored in the CSV file after parsing the script. The normal traffic data and attack traffic data are shown in Figures 8 and 9 respectively.



```
4,7,0,203,10,0,3,1573106200763624,0,0,1573106200763827,1,1050,192.168.2.136,192.168.2.126,17,12345,4790
4,5,7,164,11,279,2,1573106200763181,0,0,1573106200763345,4,1050,192.168.2.136,192.168.2.126,17,12345,4790
4,2,5,211,12,305,3,1573106200762665,0,0,1573106200762876,4,1050,192.168.2.136,192.168.2.126,17,12345,4790
4,4,2,168,11,396,4,1573106200762101,0,0,1573106200762269,3,1050,192.168.2.136,192.168.2.126,17,12345,4790
4,3,4,409,19,504,1,1573106200761188,0,0,1573106200761597,3,1050,192.168.2.136,192.168.2.126,17,12345,4790
5,7,0,111,7,0,2,1573106200764031,0,0,1573106200764142,1,931,192.168.2.133,192.168.2.123,6,12345,4790
5,4,7,200,12,259,5,1573106200763572,0,0,1573106200763772,6,931,192.168.2.133,192.168.2.123,6,12345,4790
5,6,4,225,12,356,2,1573106200762991,0,0,1573106200763216,3,931,192.168.2.133,192.168.2.123,6,12345,4790
5,3,6,141,11,423,2,1573106200762427,0,0,1573106200762568,4,931,192.168.2.133,192.168.2.123,6,12345,4790
5,1,3,197,13,426,1,1573106200761804,0,0,1573106200762001,3,931,192.168.2.133,192.168.2.123,6,12345,4790
```

Figure 8. Normal flow data obtained by int.



```
46724,7,0,1163,972,0,4,1573106399228793,3,2,1573106399229956,1,202,40.246.169.95,192.168.2.120,6,5006,4790
46724,6,7,179,9,415,2,1573106399228199,0,0,1573106399228378,4,202,40.246.169.95,192.168.2.120,6,5006,4790
46724,3,6,1537,1020,691,2,1573106399225971,1,1,1573106399227508,4,202,40.246.169.95,192.168.2.120,6,5006,4790
46724,1,3,3186,3025,406,1,1573106399222379,2,3,1573106399225565,3,202,40.246.169.95,192.168.2.120,6,5006,4790
46725,7,0,1022,751,0,4,1573106399229168,2,1,1573106399230190,1,202,44.143.144.56,192.168.2.120,6,5007,4790
46725,6,7,254,9,414,2,1573106399228500,0,0,1573106399228754,4,202,44.143.144.56,192.168.2.120,6,5007,4790
46725,3,6,1411,983,347,2,1573106399226742,1,2,1573106399228153,4,202,44.143.144.56,192.168.2.120,6,5007,4790
46725,1,3,3093,2941,813,1,1573106399222836,3,2,1573106399225929,3,202,44.143.144.56,192.168.2.120,6,5007,4790
```

Figure 9. Attack traffic data obtained by int.

Figures 8 and 9 respectively show the normal traffic data and attack traffic data obtained by int. the first item is the packet number, and the data with the same number is represented as the same stream, followed by the incoming switch number, outgoing switch number, switch processing time, instantaneous queuing delay, link delay, incoming switch port number, incoming switch timestamp, incoming queue number, outgoing switch timestamp Output the switch port number, packet length (in bytes), source IP, destination IP, transport layer protocol, source port, and destination port.

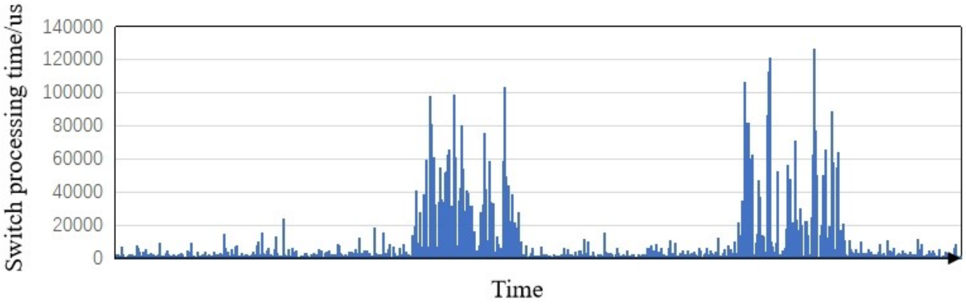


Figure 10. Change diagram of switch processing time after switch S5.

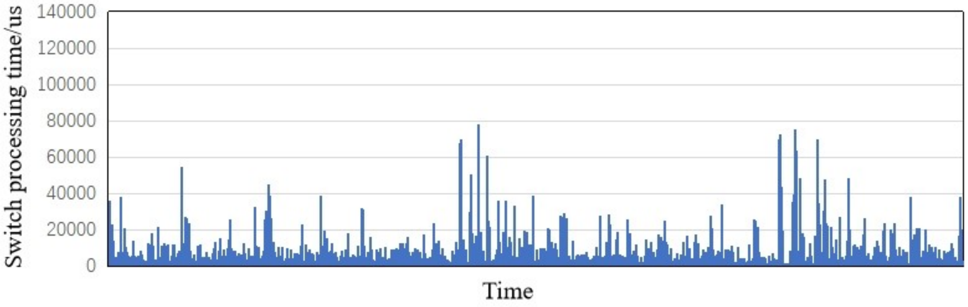


Figure 11. Change diagram of switch processing time after switch S4.

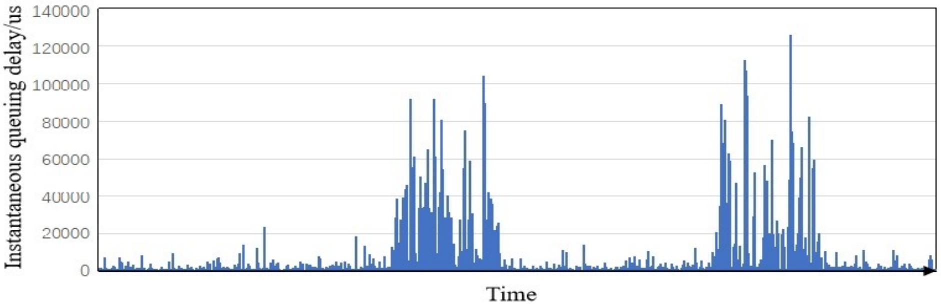


Figure 12. Instantaneous queue delay variation through switch S5.

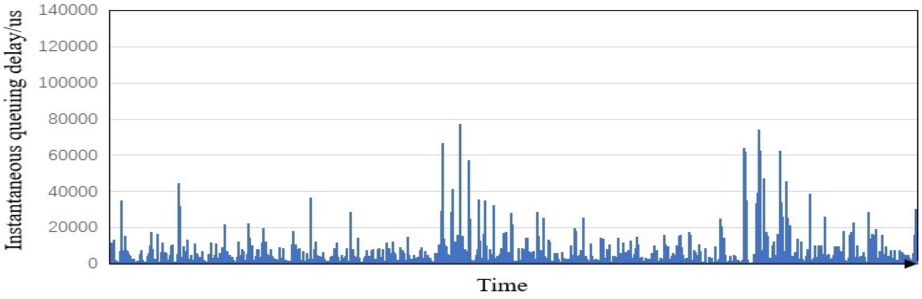


Figure 13. Instantaneous queue delay variation through switch S4.

Figures 10 to 13 show the change of processing time and instantaneous queue delay of switch S5 and switch S4 respectively. From the figure, it can be seen that under normal circumstances, about 85% of the processing time of switch S5 is within 1000US, that is, 1ms, while the processing time of switch S4 is within 6ms. This is because in the set analog traffic path, the packet path through switch S4 is more than that through switch S5, which also shows that the performance of the switch is affected by the number of packets to be processed. When the attack occurs, because the attack path passes through S5, the switch processing time and instantaneous queuing delay of S5 increase sharply, while the processing time and instantaneous queuing delay of S4 do not change significantly, which again shows that the performance of the switch is affected by the number of packets. At the same time, it can be seen that when the attack occurs, the processing time and instantaneous queue delay of the switch increase in a stepped manner, which is due to the gradual increase of processing time caused by a large number of packet congestion.

The impact of DDoS attacks on the acquired In-band network telemetry metadata shows that it is highly feasible to detect DDoS attacks by analyzing these metadata[15].

4. CONCLUSION

This paper first introduces the SDN network architecture and studies the principle of data plane programmable technology, then studies the principle and process of in-band network telemetry technology, and then studies the principle and characteristics of DDoS attack. Finally, the background traffic and high-intensity DDoS attacks are simulated through experiments, and the impact of attack traffic on network performance is analyzed by comparison. The attack will increase the switch processing time and instantaneous queue delay on the attack path, but it has no impact on the switch that is not on the attack path. It proves that it is feasible to detect and defend DDoS attacks through metadata.

ACKNOWLEDGMENTS

This project is supported by the digital twin of intelligent logistics equipment of the outstanding young and middle-aged scientific and technological innovation team plan project (T2021043) of colleges and universities in Hubei Province; The Doctoral Fund Project of Wenhua College (J0200540506) studies the security defense strategy of intelligent recognition of autopilot images.

REFERENCES

[1] Nunes, B., Mendonca, M., Nguyen, X., et al., “A survey of software defined networking: past, present, and future of programmable networks,” IEEE Communications Surveys & Tutorials 16(3), 1617-34 (2014).

[2] Sezer, S., Scott Hayward, S., Chouhan, P. K., et al., “Are we ready for SDN implementation challenges for software defined networks,” IEEE Communications Magazine 51(7), 36-43 (2013).

[3] Koponen, T., Casado, M., Gude, N, et al. “Onix: a distributed control platform for large-scale production networks,” Proc of the Osdi, 1-6 (2010).

[4] Da Silva, J. S., Boyer, F. R., Chiquette, L. O., et al., “External objects in p4: an ROHC header compression scheme case study,” 2018 4th IEEE Conf. on Network Software and Workshops (Netsoft), 517-522 (2018).

[5] He, C. H., Chang, B. Y., Chakraborty, S., et al., “A zero flow entry expiration timeout P4 switch,” Proc. of the Symposium on SDN Research ACM, 19 (2018).

[6] Li, Y., Miao, R., Kim, C., et al., “Lossradar: fast detection of lost packets in data center networks,” Proc. of the 12th International on Conf. on Emerging Networking Experts and Technologies ACM, 481-495 (2016).

[7] Li, H. F., Huang, X. L. and Zheng, Z. Q., “DDoS attack detection method based on software defined network and its application,” Computer Engineering (02), 118-123 (2016).

[8] Zhang, C. K., Cui, Y., Tang Y. I. and Wu, J. P., “Research progress of software defined network (SDN),” Journal of Software 26(1), 62-81 (2015).

[9] Lantz, B., Heller, B. and Mckeown, N., “A network in a laptop: rapid prototyping for software defined networks,” Proc. of the 10th ACM Workshop on Hot Topics in Networks Hotnets, 20-21 (2010).

[10] Gil, T. M. and Polletto, M., “Multipops: a data structure for bandwidth attack detection,” USENIX Security Symposium, 23-38 (2001).

[11] Xu, Y. and Liu, Y., “DDoS attack detection under SDN context,” 35th Annual IEEE International Conf. on Computer Communications, 1-9 (2016).

[12] Wang, R., Jia, Z., Ju, L., et al., “An entropy based distributed DDoS detection mechanism in software defined networking,” IEEE Trustcom/Bigdatase/IspaIEEE 1, 310-317 (2015).

[13] Dai, M., Cheng, G. and Zhou, Y. Y., “Research on measurement method of software defined network,” Journal of Software 30(6), 1853-74 (2019).

[14] Mohan, V., Reddy, Y. R. J. and Kalpana, K., “Active and passive network measurements: a survey,” Intel. Journal of Computer Science and Information Technologies 2(4), 1372-85 (2011).

[15] Xiao, P. B., [Research on SDN detection and defense against DDoS attacks based on In-band telemetry technology], China Naval Research Institute, Master’s Thesis (2020).