Detection and Prevention of ARP Spoofing using Dynamic Port and State Allocation.

Anika Akhtar Lima
Dept. of Computer Science and Engineering
Chittagong University of Engineering and Technology
Chittagong, Bangladesh
anikalima78@gmail.com

Asaduzzaman
Dept. of Computer Science and Engineering
Chittagong University of Engineering and Technology
Chittagong, Bangladesh
asad@cuet.ac.bd

***Abstract*—Providing a secure internet for dedicated users sharing a communication link has become an important issue now-a-days. Among several kinds of spoofing activities spread all over the communication system , ARP poisoning is referable. SDN security module Provides protection against ARP spoofing. For detecting and preventing ARP attacks, dynamic port allocation has been used.SDN deploys a controller which can be a pox controller or a ryu controller or NOX controller.Controller serves as the brain of software defined networking.Fixed value assignment for states may make it easier for the attacker to drudge the network.Hence random value assignment for states may solve this problem.An attacker may also approach with Multiple arrival request with a hope to verify himself as legitimate host.So arrival request should be rejected if DHCP packet arrives more than a specific number of time. Mininet is used for setting up NFG implementation. Pyretic query was used by NFG. Moreover our framework's key contribution is that it affirms secure ARP with nominal involvement by network operators while assisting both static and dynamic port allotment demanding no changes to the current network topology or protocols, and demanding no consumer software installation.**

***Index Terms*—ARP Poisoning,Security,Software Defined Networks,Network Topology, DHCP**

I. Introduction

Address resolution protocol (ARP) request is a kind of request when a host wants to learn the MAC address of a destination host.IP address is known to the host but MAC address is unknown. So a request is delivered with the source IP and source MAC address while the destination IP address is known but MAC address is not known. Thus an ARP request appears [1]. As the source IP is known, but destination IP is not known, so anyone can claim the destination IP as its own IP. ARP attacks occur by transpiring several spoofing enterprises [2]. That stipulates when the attacker has changed its own IP with the desired destination IP, then the attack initializes. ARP spoofing activities can be minimized by clasping some ARP preventing steps. Several ARP preventing steps are enduring. They are apprised as dynamic ARP inspection, intrusion detection system, port security, etc.The current states for detecting and preventing ARP spoofing are slow to detect or the technologies need skilled professionals to maintain. SDN based preventing methods ensure more efficiency by enforcing NFG (network flow guard) for ARP [3]. Network flow guard is represented as a modular application which accelarates an SDN controller to detect and prevent ARP replies from unauthorized hosts. NFG verifies Dynamic host configuration protocol (DHCP) offers , requests, and acknowledgements from valid DHCP servers to assemble a dynamic table. NFG's key contribution is that it requires no additional software installation and no additional equipment to the current network protocol. Doing so permits NFG to enhance the competence of SDN to restrain ARP poisoning attempts as they occur. SDN exertions by decoupling data plane from control plane. In this paper algorithms are proposed for preventing ARP spoofing activities. In general SDN controllers detects replies from unauthorized sources and prevents ARP spoofing. DHCP (dynamic host configuration protocol) offers, requests and acknowledgement need verification to justify whether these are from valid DHCP server or not. A MAC:IP:port:fixed association is constructed in dynamic table. Thus SDN is capable to detect ARP poisoning attempts [4].

This paper is organized as follows. Section II furnishes background information concerning shortcomings of the existing method and assessments taken against the hindrances of existing method. Section III discusses the current state of the art for detecting and preventing ARP poisoning. In Section IV, we first provide a review of network flow guard module design. Here we discuss our test environment comprising controller, openflow switch, DHCP server network address translator and other devices. Section V discusses implementation including algorithm and a brief of packet loss, link loss and delay. Section VI contains test environment and results. Then, in Section VII, we discuss future work, and finally conclude in Section VIII.

II. Background

Network security is a growing concern. And hackers continuously test the limits of tools available to network operators. Several mechanisms are followed to prevent ARP spoofing. One of them is dynamic ARP inspection [5]. Intrusion detection system, port security, etc. are various types of anti-spoofing skills introduced by developers [6]. These solutions need more and more guidance for restricting the attacker from committing spoofing terminologies. Port security has the problem that if the host is not connected with switch port then the host will not be able to establish a connection with those

hosts [7]. Intrusion detection system mostly depend on the network administrator [8]. Network administrator's presence is a must for preventing a source from unauthorised access. Manual effort is compulsory here. The puzzling problems about the existing method is that the update of state values are fixed [9].

Using a fixed value for state update provides a risk for the users that one may guess the correct state value and then he may attack the network. Thus using a constant update value may lead to a problem.

The another problem is that checking whether DHCP offer or acknowledgement is true or false. This checking is based on the request arrival. When a request arrives the previously assigned value ensures whether it is an offer or an acknowledgement. After that if we find that the request is from a suspicious IP, then the packet is ignored . Here if the attacker tries several times, then he will be succeeded to guess a correct state value. So what can be done is to set a limited number of time and this time indicates how many time an IP can request for offer or acknowledgement. Finally , if request from an IP exceeds that limited number of time, then we will ignore the packet [10]. Otherwise attacker will send more and more requests by guessing a correct state value and the network will be spoofed. We propose to set a counter like variable that can count how many time a request has arrived. Next if a request arrives more than the counters value ; then the request is ignored . Thus this limitation of the existing algorithm can be solved. The another problem is, if the state value is constant for all the request then one time the attacker will somehow manage to guess the state value. So we should consider such methods that will prevent the attacker from guessing a correct value for state.

As a solution of this problem , We assign the state with random values. When a request arrives ,its state value is updated with a random value. As random values are difficult to guess, so security is assigned to the ARP protocol.

Large network needs huge resources for running a testbed. Mininet can implement a large network with limited resources. As hardware testbeds are very expensive, to avoid expenses initially, mininet is the best testbed alternative to hardware. Accuracy of performance can also be measured using a mininet testbed [11]. Mininet contains Minimal, single, tree, reversed or linear topologies. Mininet holds the feature of creating custom topologies. We can add as many hosts, switches or other devices as much as we wish [12]. Lots of numbers , classes, functions are accessible in mininet. A list of basic commands are used for configuring mininet. Mininet topologies need a base class. Network can be started or stopped using start() and stop() command [13] in mininet. Connectivity among hosts can also be checked. User can dump any connection or link if the user wants to. Topology needs an application of appropriate parameter. An expiration of flow entries are available. That means after a certain time period, flow entries will be added again. Previous entries are denied. Reconfiguration of a real system is difficult. So one can use a virtual machine installed with mininet for configuring a system again and again. Its free of cost and other technical issues have been ignored here [14]. Very large scale network which is hard to implement on real system , can be implemented using mininet [15].

Fixed value assignment for states may make it easier for the attacker to drudge the network.Hence random value assignment for states may solve this problem.An attacker may also approach with Multiple arrival request with a hope to verify himself as legitimate host.So arrival request should be rejected if DHCP packet arrives more than a specific number of time.

III. Method Overview

In this portion, we are presenting the proposed method. The topology uses a pox controller, dhcp server, openflow switch, hosts. Attacker and the spoofer are two hosts . Initially when the request arrives, it is checked whether it is an offer or acknowledgement. If the result is false then the MAC is checked on the table. If the MAC is found, then port is checked. If port is static then packet is ignored. If port is not static then sate is checked ; meanwhile state has been updated in another state. After that a new state value is assigned.Again if the offer and acknowledgement results in a true value then MAC is checked on table. If MAC is not found then an entry will be added to the dynamic table. It indicates that the request has arrived for the first time. Then it also ensures that offer has been resulted in a true value.

Another part is when acknowledgement is true then the offer or acknowledgement results in a true value, then MAC is checked on the table. If MAC is found then IP is matched and the state is also matched. If both of them results in a true value then the verification state is achieved. Thus the process works. Till this part it is similar as the existing one. A counter has been set for counting the number of time an IP request arrives. This is basically done for limiting the number of time an IP can send request to the system. We are using random numbers for state assignment.

A. Review of NFG Implementation

A representation of NFG module [16] is presented in brief . Pyretic queries are used by NFG [17] where Different types of controllers exist. Among them pox controller, ryu controller and frenetic controller , open day light controllers are commonly known [18]. Another existing controller is NOX controller.But NOX controller is specified using c++ language. Python based controllers are ryu and pox. These controllers are used for different purposes [19]. For implementing a simple topology, frenetic controllers are the best. Every controllers have different throughput and latency performances [20].

Controller is the core part of the network. When a poisoning activity is defined, controller tells the switch to disconnect the port. A dynamic table is used for showing which ports have been disconnected. The ports are shown separately from other table entries. After detecting an active attack, a prevention is also needed. Otherwise only detecting a poisonous activity does not ensure a networks security. So , the poisonous entry

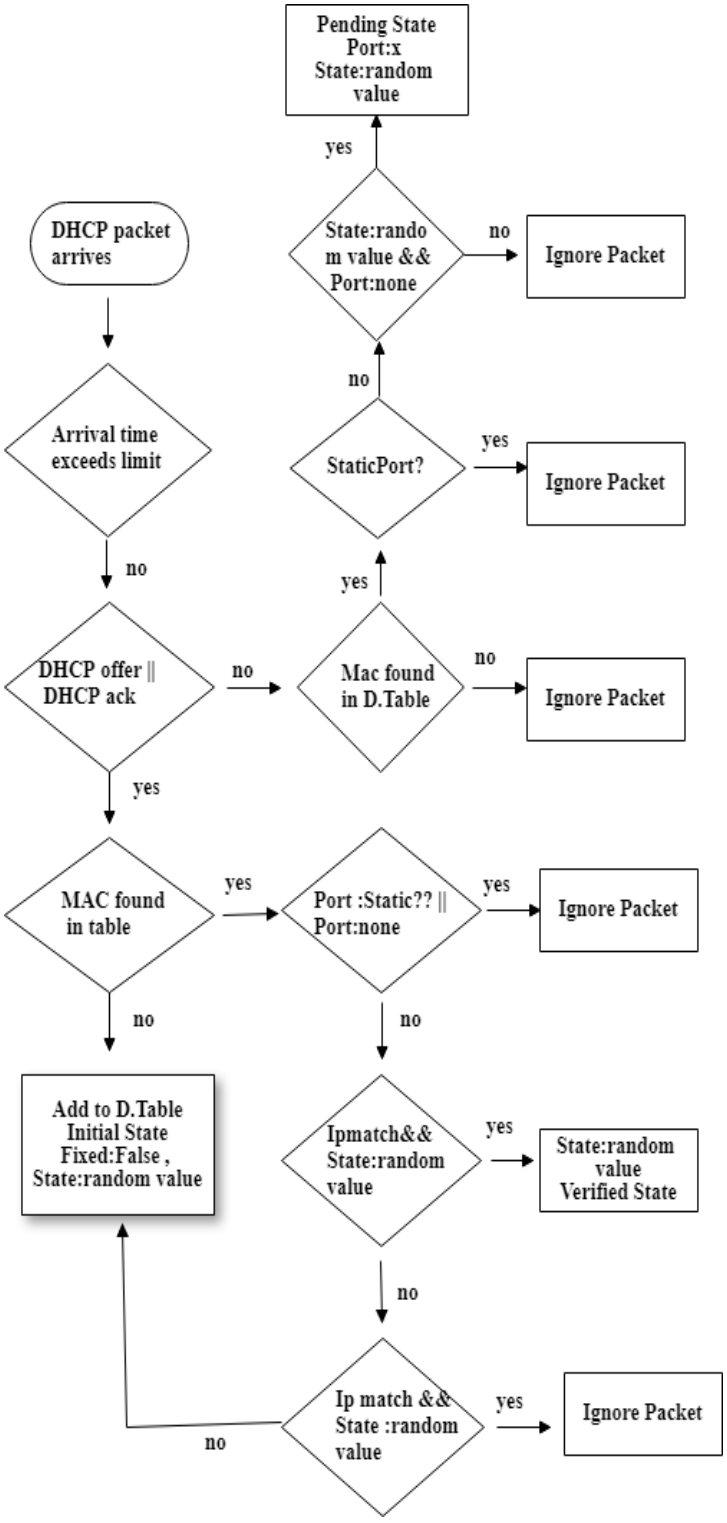

Fig. 1. Packet flow graph for ARP spoofing detection.

has been removed from the table and then all the entries are updated as before [21]. Dynamic port allocation means that the ports can be updated using different states. Table has different entries [22]. And this entries can be modified at any time. Entries that are old can be replaced with a new one. That means entries can be updated at any time. If any saved entry is not updated for a long time then this entry will be removed from the table and a new entry will be added to the table.

Packet parse means for parsing or resolving into components. Different types of packets are declared. TCP, UDP, ARP etc. have fixed amount of parameters and are briefly initialized while declaring each packets.By creating a dhcp configuration file, finally dhcp server is activated and deactivated again and again. Hosts links are configured with necessary parameter and used for assigning gateway, spoofer and target IP.

IV. Experimental Setup and Implementation

An adaptable solution for current network includes a NFGC (NFG coupler) module. The NFG coupler couples both a MAC learning switch and NFGA (NFG ARP).Rules are generated by the NFGA and NFG coupler allows to generate these rules [23]. These rules are pushed through the northbound interface of POX controller along with the correspondence of switch. POX controller generates and interprets these instructions and then pass them via southbound interface of the openflow switch [24].

NFGC have been modified using pyretic, which is a modular programming language. Through this part, the controller is told to handle almost all the packets. After identifying destination IP, source IP, destination MAC, source MAC the controller tells the switch port to disconnect a port, or maintain connectivity with a host. Controller has some basic rules that contain all the necessary conditions for checking whether an attack exists or not. Controller handles the switches flow table. The flow table entries are updated using the commands that will be passed from the controller via its southbound interface [25].

A flow chart named packet flow graph for ARP spoofing detection is shown in method overview section. The modification includes checking arrival time of a packet. In this part number of time an IP arrives is checked and a counter is initialized for calculating total arrival request for an IP. In case counter value exceeds limit then the request from that particular IP won't be approved at all.

The another modification includes assigning random values to the state.

A. Topology for POX Controller

At first devices are initialized in a topology for securing ARP protocol.A controller is the core part of a topology designed for ARP spoofing detection. In this case a POX controller is used. A controller checks for a legitimate ARP reply. When a controller detects a false ARP reply , openflow switch is told to ignore that packet. Various forms of controllers are available. As various types of controllers perform distinctly, two types of controller is used here for experiment.

B. Modified Topology for RYU Controller

In previous subsection, pox controller is used in the topology. Here, a ryu controller is used as a replacement of pox controller in the topology for detection method. Considering that the other settings are the same.

As various controllers shows a variety of performances. So pox controller is replaced with ryu controller so that variation in performance can be observed. Again the best performing controller can also be discovered.

C. Modified Topology with Lighttpd Web Server

In this subsection , a lighttpd web server has been used, basically it is done for providing a comparison among the testbed [26]. Comparison of packet loss rate, link loss rate and delay have been shown in another section. Hence addition of one host or device to the topology generates a different results from other topologies. This is one of the reason why a lighttpd server is added to the topology. Another reason for adding a lighttpd server is when a network faces an unauthorized access , a lighttpd web server shows the practical situation. A topology is shown including NAT, an openflow switch, a dhcp server, hosts, lighttpd web server and a controller.

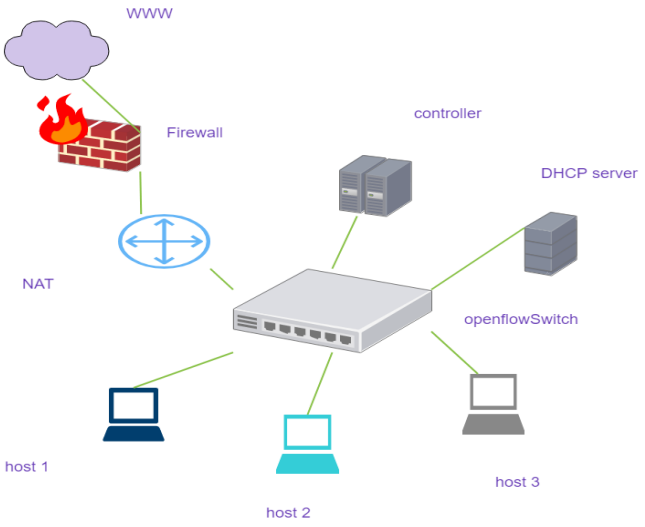

Fig. 2. NFG Implementation module setup.

When a host responses to a false ARP reply, the network permits an unauthorised access. That spoofer is now a part of the network. if the network topology contains a lighttpd web server, then with the association of a network address translator a host is able to connect to the internet. A random host may log into the web server using its own password and username [27]. This random host will perform the role of a target for the attacker or spoofer. A spoofer is now able to see the log in information. Thus the target host is now at risk. A web server is added to the topology to show whether a spoofing activity is present or not. As it is essential to perform a spoofing activity first and then preventive measures can be undertaken followed by detecting a spoofing activity. So in this topology a web server is used for poisoning the network first and then security of ARP protocol is ensured.

D. Modified Topology with Additional Host

Hosts have been added to the proposed topology.Previously only two hosts have been used, they are the target and the attacker.

This methodology follows the same algorithmic processes as shown in the above flow chart before. The same processes are running on the switch, dhcp and controller [28].

Detecting a false ARP response requires basically three hosts , one is the sender and another one is the receiver and the last one is the spoofer. The scenario is that as soon as request arrives for a particular host (receiver) , the spoofer sends a reply with the intended IP address though spoofer actually does not belong to that particular IP address. Adding more hosts to the topology may show variances in result. For observing variations in performances , number of hosts have been inserted to the topology.

E. Finding Delay

A controller, switches , hosts comprise a topology. Delay from one switch to another may vary for traffic load or other reason. The delay is measured by the controller and controller also decides which route to follow. Controller also sends instruction to switch to send packet to other host or device. Then switch forwards the packet back to controller. The time taken for sending a packet and the time taken for receiving a packet is essential for measuring delay.

Initially one time delay is calculated for both sender and receiver. Then the difference is calculated to measure delay.Thus number of delay for a number of individual packets has been calculated. To calculate average delay, we have to divide the total number of delay with the number of received packet [29].

average delay for pox controller is 69.997 ms. The process for finding delay is same as described above. Like before, the delays for individual methods have been calculated and then the average delay has been calculated. Thus the delay has been calculated [30]. In this case we made a change to the current topology and that is we have used ryu controller instead of the pox controller. Here the topology now contains an openflow switch, a dhcp server, a ryu controller, number of hosts. a change is also made to the current topology with a number of added hosts.

in order to make changes, lighttpd web server has also been added to the current topology.

F. Finding Packet Loss

Due to the link loss rate setting in the mininet , the packets transmitted over the link between switches and other subordinates may lost. The difference between inputPkt and outputPkt measures the packet loss. When the controller gains no response for a particular request , controller considers it as a loss.

Thus finding the difference between inputPkt and outputPkt, controller finds the packet loss rate.

We have considered the topology containing a pox controller.

Here the number of packets transmitted were 48 and the received packets were 44. Formula for finding packet loss is, packet loss= (transmitted packets-received packet)*100/transmitted packet.[30] packet loss=(48-44)*100/48

packet loss for pox controller is 8.3333333333

G. Link Loss Rate

Host H0 sends packets to the desired host through different links. Thus this links start from the host1 and passes through the switches or controller and other existing devices in the topology and finally reaches to the other end host. The link loss rate may be calculated by following mentioned formula. Link Loss Rate = (InputPkts-OutputPkts)*1.0/InputPkts*100

A function has been created for timestamp which is basically used for getting the time.

def getTheTime():

Timer function has a handler which sends requests to all the switches and here the switches are needed to be connected to the controller. Flow statistics are shown using the handler function. Port statistics are also shown using the handler function. If input packets are not zero, then the link loss rate is calculated.

V. Test Environment and Results

Fig. 3. Pox controller set up using mininet .

The above terminal shows the setting up process of POX controller on mininet [30].Switch holds a port for connecting hosts to it. Ports are disconnected when a poisonous entry is found. So it is necessary to verify whether a host connected to the port is valid or not. Switch drops a packet by disconnecting a port when the controller tells to do so. Otherwise switch works as a learning switch.

wireshark is used to detect duplicate use of IP address. As the target thinks that the attackers IP address is its gateway when the gateway has its own IP address. Thus in a network there exists two similar IP addresses. Wireshark uses a warning message which is duplicate use of IP [31].

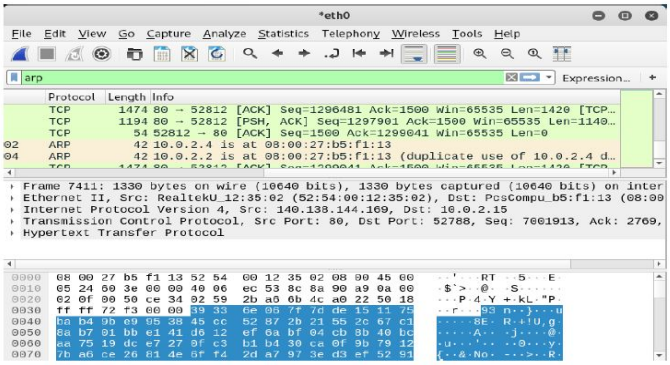

Fig. 4. attack detection using wireshark.

A. Comparison of Packet Loss

Ryu controller passes more transmitted packet than the pox controller at a given time. On the other hand, pox controller passes less number of packet than a ryu controller. So naturally the packet loss for ryu controller is less than pox controller.

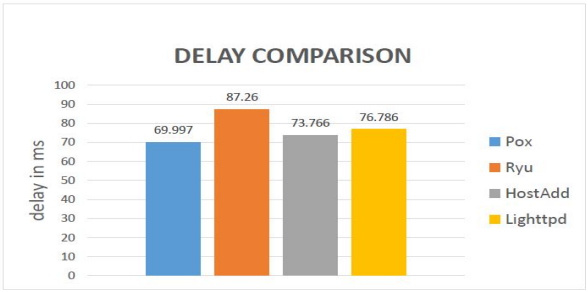

Fig. 5. comparison of packet loss.

Using a lighttpd server also generates a number of packet loss, because a number of packet received is less than a number of packet transmitted considering a given time period. Similarly, adding a number of host to the topology also generate a number of packet loss. In this case, the number of received packet varies in different topologies, variation in number of received packets provides a packet loss. Ryu has the less number of packet loss rate.

B. Comparison of Link Loss Rate

Ryu has the less link loss rate. This is because the ryu controller has the most latency and throughput. As ryu gains the maximum number of transmitted packets as received packets,a topology using a ryu controller has the less number of link loss rate.

Topology containing a pox controller has a packet loss rate of 8.3. Ryu has less packet drop rate than the pox controller. Hence ryu has less link loss rate than pox controller. In fact ryu has the least number of link loss rate comparing with all other topologies.

Calculated link loss rate of ryu controller is 6.435. Applying additional host to the topology, packet loss is calculated as 12.

Hence link loss rate is more here. As more packets have not been sent to the receiver, more links have not been established in this case. Packets are dropped after they have been detected as suspicious by the controller. Topology containing Lighttpd server has less packet drop rate than the topology containing additional host. So link loss rate is less in topology containing lighttpd server than topology containing additional host.

C. Comparison of Delay

Ryu controller has more latency and throughput than pox controller. For this reason, for a given period of time, ryu receives the maximum number of packets than a pox controller. As ryu receives more packets, it also has more transmission and reception delay that pox controller. Again pox has less latency than ryu controller. A pox controller has more throughput than a ryu controller. That is why it has less transmission delay and reception delay.

VI. Future Work

As providing security to network has become a growing concern, better cautionary measures may contribute to a successful ARP poisoning detection. Following is a series of actions for future improvement:

Number of hosts detected need to be increased. Number of packet loss needs to be reduced. Although when ryu controller is used, the loss is minimized to an extent.

Depending on the controller, the time required for detecting a poisonous host may become faster. So various controller need to be tested for finding a better performance.

For minimizing delay, necessary measures need to be taken. Then detection and prevention of ARP based attacks will be performed in a faster time.

VII. Conclusion

A detection measure have been represented that finds the malicious ARP reply in response of valid ARP request. Implementation process uses a mininet platform. The core idea is we need to make the port dynamic so that state values can be updated step by step. Here the checking is done using the fixed constraint. If the constraint returns a true value that indicates that the port is static, then the entry is dropped from the table. For preventing ARP attacks we use a controller in our mininet topology, where the detection algorithm is being continued. Then the controller tells the switch to disconnect the port association. The state value is updated in every step until it reaches to the verification step. The verification step also assigns a value to the state. This process supports dynamic state allocation. Moreover, we used some test criteria for comparing results obtained. We calculated packet loss, link loss rate, delay for various combination. We implemented a test case using pox controller, then we considered a ryu controller. We added some hosts to the topology mentioned above. Again, a lighttpd web server was initiated.

Following is a table showing comparison among packet loss rate, link loss rate and delay(in ms) for different topologies:

Tables

	POXController	RYUController	AdditionalHost	Lighttpd
Packet Loss	8.3333333333	8	12	10
Link Loss Rate	10.978	7.8404	8.527	8.785
Delay	69.997ms	87.260ms	73.766ms	76.786ms

TABLE I
PACKET LOSS, LINK LOSS RATE, DELAY COMPARISON

References

[1] S. Hijazi and M. S. Obaidat , "A New Detection and Prevention System for ARP Attacks Using Static Entry,"in IEEE Systems Journal, IEEE, vol. 13, pp. 2732 - 2738, 2018 .

[2] J. Clerk Maxwell, "An efficient and feasible solution to ARP Spoof problem," in 2009 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, vol. 2,Pattaya, Chonburi, 2009.

[3] N. Tripathia, B. Mehtre, "Analysis of various ARP Poisoning mitigation techniques : A comparison," in International Conference on Control, Instrumentation, Communication Computational Technologies, 2014.

[4] W. Xia , Y. Wen , C. H. Foh , D. Niyato , H. Xie, "A Survey on Software-Defined Networking," in IEEE Communications Surveys Tutorials, vol. 17, pp. 27 - 51, 2015 .

[5] L. Senecal, "Understanding and preventing attacks at layer 2 of the OSI reference model," in Published in: 4th Annual Communication Networks and Services Research Conference (CNSR'06), IEEE, Moncton, NB, Canada, 2006.

[6] L. Senecal, "Understanding and preventing attacks at layer 2 of the OSI reference model," in Published in: 4th Annual Communication Networks and Services Research Conference (CNSR'06), IEEE, Moncton, NB, Canada, 2006.

[7] H. S. Kang , J. H. Son , C. S. Hong, "Defense technique against spoofing attacks using reliable ARP table in cloud computing environment," in 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), IEEE, Busan, South Korea, 2015.

[8] B. Mukherjee , L.T. Heberlein and K.N. Levitt,"Network intrusion detection," in IEEE Network, vol. 8, pp. 26 - 41, 1994.

[9] S. Venkatramulu, C. V. G. Rao,"Various Solutions for Address Resolution Protocol Spoofing Attacks," in International Journal of Scientific and Research Publications, vol. 3,2013.

[10] J. H. Cox , R. J. Clark , H. L. Owen,"Leveraging SDN for ARP security," in SoutheastCon 2016, Norfolk, VA, USA, 2016.

[11] F. Keti , S. Askar,"Emulation of Software Defined Networks Using Mininet in Different Simulation Environments," in 2015 6th International Conference on Intelligent Systems, Modelling and Simulation, IEEE, Kuala Lumpur, Malaysia, 2015 .

[12] C. Pal ,S. Veena ,R. P. Rustagi , K. N. B. Murthy,"Implementation of simplified custom topology framework in Mininet," in 2014 Asia-Pacific Conference on Computer Aided System Engineering (APCASE), IEEE, South Kuta, Indonesia,2014.

[13] Mininet homepage,http://www.mininet.org/pyretic, 2019.

[14] R. L. S. de Oliveira , C. M. Schweitzer , A. A. Shinoda , L. R. Prete,"Using Mininet for emulation and prototyping Software-Defined Networks," in 2014 IEEE Colombian Conference on Communications and Computing (COLCOM), IEEE, Bogota, Colombia, 2014.

[15] Introduction to Mininet,https://www.costiser.ro, 2019.

[16] J. H. Cox , R. Clark , H. Owen, "Leveraging SDN and WebRTC for Rogue Access Point Security," IEEE Transactions on Network and Service Management, IEEE, vol. 14, pp. 756 - 770,2017 .

[17] Pyretic homepage,http://www.frenetic-lang.org/pyretic, 2013.

[18] M. Paliwal ; D. Shrimankar ,O.Tembhurne,"Controllers in SDN: A Review Report," in IEEE Access , IEEE, vol. 6, pp. 36256 - 36270, 2018 .

[19] F. Bannour , S. Souihi , A. Mellouk,"Distributed SDN Control: Survey, Taxonomy, and Challenges," in IEEE Communications Surveys Tutorials , IEEE, vol. 20, pp. 333 - 354, 2017 .

[20] F. Yamei , L. Qing , H. Qi,"Research and comparative analysis of performance test on SDN controller," in 2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI) , IEEE, Wuhan, China,2016 .

[21] X. Jia , Y. Jiang , Z. Guo , Z. Wu,"Reducing and Balancing Flow Table Entries in Software-Defined Networks," in 2016 IEEE 41st Conference

on Local Computer Networks (LCN) , IEEE,Dubai, United Arab Emirates,2016 .

[22] A. A. Kassem , N. Mitten,"Adapting dynamically neighbourhood table entry lifetime in wireless sensor networks," in 2010 International Conference on Wireless Communications Signal Processing (WCSP), IEEE, Suzhou, China, 2010 .

[23] M. Wang, J. Liu , J. Chen , X. liu , J. Mao,"PERM-GUARD: Authenticating the Validity of Flow Rules in Software Defined Networking," in 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing, IEEE,New York, NY, USA,2016.

[24] A. Lara ,A. Kolasani, B. Ramamurthy,"Network Innovation using OpenFlow: A Survey," in IEEE Communications Surveys Tutorials, IEEE, vol. 16, pp. 493 - 512,2013.

[25] D. Kreutz ,F. M. V. Ramos , P. E. Verssimo,C. E. Rothenberg,S. Azodolmolky,S. Uhlig,"Software-Defined Networking: A Comprehensive Survey," in Proceedings of the IEEE, IEEE, vol. 103, pp. 14 - 76, 2014 .

[26] Home - Lighttpd - fly light,https://www.lighttpd.net/.

[27] Use lighttpd Web Server on Ubuntu 16.04 (Xenial Xerus),https://www.linode.com/.

[28] C. Fancy , M. Pushpalatha,"Performance evaluation of SDN controllers POX and floodlight in mininet emulation environment," in 2017 International Conference on Intelligent Sustainable Systems (ICISS), IEEE, Palladam, India, 2018.

[29] D. Ramirez , B. Aazhang ,"Optimal Wireless Service Within Average Delay," in IEEE Transactions on Wireless Communications, IEEE, vol. 17, pp.5494 - 5505,2018.

[30] P. Puri ; M. P. Singh,"A survey paper on routing in delay-tolerant networks," in 2013 International Conference on Information Systems and Computer Networks, IEEE,Mathura, India,2013.

[31] G. Munz , G. Carle ,"Distributed Network Analysis Using TOPAS and Wireshark," in NOMS Workshops 2008 - IEEE Network Operations and Management Symposium Workshops, IEEE,Salvador da Bahia, Brazil,2008.