A Novel Central Arbiter to Mitigate Denial of Service Attacks on Duplicate Address Detection in IPv6 Networks

Shailendra S.Tomar, Anil Rawat
Computer Division
Raja Ramanna Centre for Advanced Technology
Indore, Madhya Pradesh, India
e-mail: tomar@rrcat.gov.in, rawat@rrcat.gov.in

Prakash D. Vyavahare
Dept. of Electronics & Telecommunication Eng.
Shri G.S.Institute of Technology & Science
Indore, Madhya Pradesh, India
e-mail:prakash.vyavahare@gmail.com

Sanjiv Tokekar
Department of Electronics & Telecommunication Engineering, Institute of Engineering & Technology,
Devi AhilyaVishwa Vidyalaya
Indore, Madhya Pradesh, India
email: stokekar@ietdavv.edu.in

***Abstract*—A node joining any Internet Protocol version 6 (IPv6) network is susceptible to Denial of Service (DoS) attack in the Duplicate Address Detection (DAD) phase of the IP address assignment process. A lot of research work is being carried out to mitigate this form of DoS attack. However, available approaches require changes in the Neighbor Discovery Protocol (NDP) and/or lead to increased computational and configuration overheads/complexity on each client. In this paper, we present a central arbiter approach to detect and mitigate DoS attacks on DAD in Software Defined Network (SDN) controlled wired IPv6 networks. Advantages of this approach over other approaches are its simplicity and zero modification requirements to the NDP. The proposed approach has been simulated on a Mininet emulator configured for SDN using RYU controller and is observed to achieve the desired results. The effectiveness of the proposed scheme in handling DAD DoS attacks is also presented in the paper. The results show that this scheme introduces a delay of the order of 0.34 seconds in the DAD process which is a good trade-off for providing DoS attack protection.**

Keywords - IPv6; DAD; DoS Attack; Central Arbiter Approach; SDN; NDP.

I. INTRODUCTION

IPv6 [1] networks use NDP [2] with State-Less Address Auto-Configuration (SLAAC) [3] feature for zero configuration. NDP has many known vulnerabilities [4], which may be exploited to perform DoS attacks on network nodes. One of them is the DAD vulnerability, which can be exploited to perform DoS attacks during the address initialization phase of a node. The Hackers Choice (THC) toolkit [5] provides simple tools to perform such attacks. Various solutions to mitigate this problem have been proposed by researchers. The best known and accepted approach to mitigate NDP related attacks is provided by the Secure Neighbor Discovery (SeND) protocol [6] [7]. However, the lack of mature implementations of the SeND protocol, and the computational and configuration complexities involved in the approach makes it less practical in the real world. Other approaches like Simple Secure Addressing Scheme (SSAS) [8], Trust-ND [9], and Secure-DAD [10] have also been proposed in the literature, but all of them have the same drawbacks. All of these approaches require modifications in the NDP messages. Hence, they require changes in the NDP implementation in various Operating Systems (OS). Network access control (IEEE 802.1x) based solutions have also evolved for mitigating layer-2 related attacks. But the complexity involved in configurations of intermediate switches and end nodes, have led researchers to think of alternate solutions.

SDN [11] [12] technology has matured in recent years. The programmable controller in SDN can be utilized to view and control the flow of NDP packets in a network. The controller can, thus, become an arbiter settling DAD disputes without requiring changes in the NDP messages and hence, no changes and complex configurations are needed in the client OS. This is the basis for the motivation of the present work.

In this paper, a central arbiter approach to mitigate DoS attacks on DAD is proposed. The central arbiter acts as a big brother and NDP related Internet Control Message Protocol version 6 (ICMPv6) messages flow controller. It sends DAD replies on behalf of genuine nodes only, thus blocking the replies of rogue nodes. The proposed solution has been tested using the Mininet network emulation software configured for SDN using RYU [13] controller. The results show that DAD attacks can be mitigated without making modifications to the NDP. This approach has an added advantage of zero computation and configuration overheads on the client side, which is a major drawback in other approaches.

 ISBN: 978-1-61208-582-1

The rest of this paper is organized as follows. Section II is the study of literature and the technology background section. Section III presents the review of already reported approaches. Section IV presents the hypothesis for the research work. Section V describes the methodology of the proposed approach in detail. Section VI describes the methods, tools and techniques used to implement and test the proposed approach. Section VII is the results section. The conclusions and future works section is at the end of the article in Section VIII.

II. STUDY OF LITERATURE

IPv6 security issues have been studied by a number of researchers [14][15]. One Hop Security or First Hop Security are common terms used to refer to NDP related security in IPv6 networks. NDP vulnerabilities are well known and many researchers demonstrated it to be easily exploitable [16]-[19]. DAD implementation in NDP is also vulnerable to DoS attacks and is easily exploitable [20]-[22] .

DAD ensures uniqueness of the IPv6 address in the network. According to the specification, DAD in IPv6 networks works during the IP address assignment phase only, if the following two conditions are satisfied:

- A node which is the genuine owner of an IP address must also listen to the Solicited Node Multicast Address (FF02::1:FFXX:YYZZ) of its corresponding unicast IP address and respond to queries, whenever requested, for DAD. Here, XXYYZZ are the last 24 bits of the unicast IPv6 address.
- The DAD reply, in the form of Neighbor Advertisement (NA) packet, to the all node multicast group (FF02::1) must be sent by the node in possession of the duplicate IP address.

IP address assignment of a node in an IPv6 network is complete after the following steps are successfully completed:

- IP Address Generation: A node generates an IP address for itself by using any one of the following techniques: Static random, Extended Unique Identifier (EUI) formatted, Cryptographically Generated Address (CGA) and Hashed.
- DAD: The node attempting to connect to the network, sends a DAD request in the form of Neighbor Solicitation (NS) request to the solicited node multicast address of the corresponding unicast address and if no reply (NA) is received within a specified timeout period, then it assigns that address to itself.

DoS attack on the DAD vulnerability works as follows: Rogue nodes in the network falsely claim to possess any IP address requested by any new node, which is attempting to join the network or may claim all IP addresses of the network prefix. This causes DoS attack on all new nodes that are attempting to connect to the network. Thereafter, no new node can connect to the network, if this situation persists. The characteristics of possible forms of DoS attacks on the DAD can be categorized as follows:

a) **Reactive:** In a reactive attack, the attacker listens to the DAD requests and gets to know the target IP address being assigned to the new node. It then reacts to such requests and sends DAD replies, thereby forcing DAD failure.

b) **Guessing:** In this case, the attacker does not know the target IP address. It only guesses the target address as to be in a particular pattern as predicted from IP addresses of other nodes. It then sends DAD replies for next in pattern target IP addresses.

c) **Flooding:** In this type of attack, the attacker floods the network with DAD replies claiming all IPv6 addresses related to a network prefix.

Research work has been carried out to mitigate all types of NDP attacks. The root cause of the problem has been identified as lack of adequate security measures in NDP message exchanges in the IPv6 network. Hence, the entire research work focuses on a) incorporating some authentication mechanism in message exchanges and b) securing the exchange of NDP messages by encrypting them. All this adds to computational and configuration complexity to the client nodes and requires changes in the NDP. Computational complexity in the client nodes is introduced in the form of additional computation required for encrypting and decrypting NDP packets. Configuration complexity is introduced in the client nodes in the form of loading of additional OS patches and their configuration for the network.

Network access control based approaches for one hop security solutions in intelligent network switches, like the 802.1x [23] and IP-MAC binding, address the DAD related DoS attack problem by registering, rate limiting and blocking rogue nodes. The configuration complexity, in the form of additional configurations of (Internet Protocol – Media Access Control) IP-MAC binding, setting rate limits at the switch level and loading and configuring necessary software agent at the client level involved in the process, makes these approaches less popular and are rarely used. Hence, there is a need for an alternative and simpler solution.

SDN is emerging as a promising network architecture. It is worthwhile to explore the possibility of detecting and mitigating NDP related attacks in SDN. SDN controller can be programmed to become a central arbiter for NDP traffic flow. In this paper, we are focusing on making the SDN controller to act as a central arbiter for only the DAD related NDP traffic.

III. ONE HOP SECURITY IN IPv6

One hop security in IPv6 networks can be divided into two main categories. In the first category, researchers secure NDP messages by using cryptographic techniques. In the second category, researchers try to control the access to the network and thereby, minimize/check the flow of NDP messages from rogue devices into the network. This section describes the features and limitations of these mechanisms in brief.

SeND uses CGA and certificate distribution framework to securely transmit NDP messages. Although SeND is able to prevent DAD related attacks, it is observed that [8] [9]

 ISBN: 978-1-61208-582-1

SeND has a drawback like high computation to generate the options, especially the CGA option and Rivest, Shamir, and Adelman (RSA) signature. Thus, it requires higher computation time. SeND mechanism adds significant processing time, of the order of 300-400 milliseconds, to perform the message verification [9]. Hence, the usage of SeND adds to delay and increased complexity during the DAD process, as highlighted by researchers [7]. These delays are unacceptable for some real-time mobile applications. Further, DoS attacks can also be performed on SeND [7].

SSAS [8] was proposed as an improved version of the SeND mechanism. SSAS introduces alternative addressing scheme by employing Elliptic Curve Cryptography (ECC) algorithm as compared to RSA which is used by SeND mechanism for address configuration. Although SSAS has reduced complexity and decreased message processing time as compared to SeND mechanism, this method depends on signature and key exchanges. Hence, the time complexity issue still exists [9]. Based on research conducted by Praptodiyono et al. in 2015 [9], SSAS mechanism takes 223.1 milliseconds to generate an interface identifier, which is still a substantial amount of processing time.

Another research work, proposed as Trust-ND [9], is a lightweight mechanism for the DAD process. The main approach of this mechanism is to reduce the processing time of ND messages during the DAD process, as compared to the SeND and SSAS. In Trust-ND message, authentication is done by employing Secure Hash Algorithm 1 (SHA-1) operation as message integrity check. Researchers [24] [25] have shown that SHA-1 and Message Digest 5 (MD5) hash functions are susceptible to hash collision attacks.

Yet another research work proposed as Secure-DAD [10] states to use Message Authentication Code using Universal hashing (UMAC) for hashing and authentication of the messages. It is argued that UMAC is a more efficient algorithm and more secure algorithm than SHA-1/MD5 [26] [27]. Their work is similar to that of Trust-ND but with a different hashing algorithm. This approach also suggests to make changes to the original NDP message exchanges.

Another research work [28] proposes a novel duplicate address detection with a hash function. It exchanges hash values of the IPv6 addresses between all the nodes. Only the node owning the real IPv6 address can generate the equivalent hash and thus, claim to be the real owner of the address. This work also requires modification to the NDP protocol.

An SDN based authentication mechanism for securing neighbor discovery protocol in IPv6 is proposed in [29]. It basically provides a solution to counter IP spoofing attacks in IPv6 networks using the SDN architecture. It utilizes a table on the controller to learn MAC addresses and binds them to ports, thus ensuring MAC spoofing protection from other network ports.

Recent research work [30] proposes to address the one hop security concerns from ground up, by using the Trusted Platform Module (TPM) for ensuring trusted endpoints on the network. The required restrictions on clients, to be TPM enabled for ensuring one hop security, makes this solution less practical in real world.

IV. HYPOTHESIS

In this section, we state the hypothesis on which our approach is based. We also discuss the mechanism which we have used to prove our hypothesis.

Assume a central arbiter which acts as a gateway for all IPv6 related NDP traffic, especially DAD requests (NS) and DAD replies (NA). We know that every DAD process is initiated by a DAD request in the form of a NS packet from a new node. If all DAD requests are blocked by the central arbiter then no other node will get DAD requests, from which it can extract the target IP address to attack. Thus, rogue nodes cannot generate DAD replies. The central arbiter can selectively generate DAD replies on behalf of the genuine target node which is definitely present in the network. Thus, the DAD process can be completed securely without changing the NDP message formats and without configuration of the intermediate devices, if some alternate mechanism to search "already in use IP addresses" for the network is present.

Thus, our hypothesis states that "*using a central arbiter approach, the security of the DAD process can be successfully accomplished without:*

a) change in the NDP message structures,

b) change in client configurations,

c) additional computational overheads at clients,

d) additions in network access control configurations in intermediate switches".

This hypothesis can be tested using the emerging SDN technology. The SDN managed network must be configured with a modified controller as per our approach. The hypothesis for providing a solution to the DoS attack on DAD process can be tested if the following conditions are met:

1) SDN controller must have a global view of all nodes connected to the network. It should also be able to intercept all NDP traffic related to the network.

2) The controller must be enabled as a central arbiter for analyzing all DAD requests (NS).

3) The controller should be able to fabricate DAD replies for duplicate requests and dispatch them to the node from where DAD request was received.

4) The SDN controller should be able to distinguish between genuine nodes and rogue nodes with respect to DAD processing based on the following:

- Searching a persistent table called "IP_MAC_Port_Time" table which contains the list of all nodes and the IP addresses presently assigned on the network.

The management of the IP_MAC_Port_Time table is done as follows:

- The table gets populated on the attachment of every node to the network followed by its subsequent IP address assignment.
- The table entries are pruned after the expiry of the configured IP address lifetime for the network or if a reply to heartbeat packet is not received within a specified time.
- Static IP address assignments can be manually inserted into the table with an infinite lifetime. These can be manually pruned by the administrator to reflect topology changes.
- Limits on the number of entries on per port basis should be implemented to counter table flooding attacks.

V. PROPOSED SOLUTION

This section describes the proposed mechanism. It describes the workflow and explains the design of various modules of the proposed solution.

A. *Workflow*

Figure 1 depicts the workflow of the proposed approach. All NDP related ICMPv6 packets are fed into the SDN controller which is configured with central arbiter module. This allows the controller to learn all IP MAC Data Path Identifier (DPID) Port associations that are existing in the network. The Collector sub-module is responsible for populating the IP_MAC_PORT_Time table which is a table with IP, MAC address, DPID, Port Number and timestamp fields, as shown in Table I. DPID is the identifier of the switch connected in the network. Other fields are self-explanatory. Only DAD request packets are fed into the Verification sub-module.

The Verification sub-module extracts the target IP from the NDP packet and searches for the target IP. If the target IP is found in the IP_MAC_Port_Time table, then the DADReplyGenerator sub-module is invoked. The DADReplyGenerator sub-module fabricates a DAD reply based on the DAD Request packet and then dispatches the DAD reply to the corresponding switch port of node from where the DAD request has originated. If the target IP search fails, then no DAD reply packet is sent by the controller and DAD request packet is blocked at the controller level itself. A node, thus, completes the IP address assignment process.

TABLE I. IP_MAC_PORT_TIME TABLE FIELDS

IP	MAC	DPID	Port	Time stamp

Figure 2 depicts the logic for deleting entries from the table used in the central arbiter. The Prune_IP_MAC_PORT_TIME sub-module extracts the target IP from a new DAD request, and checks whether the IP address exists. If so, it generates an ICMPv6 ECHO REQUEST packet and dispatches it to the switch port with which the IP is associated. If the ECHO REPLY is not received within a time period, then the associated IP entry is deleted from the table. The entries in the table are also deleted on expiry of the IP lifetime for the network, which can be defined by the administrator.

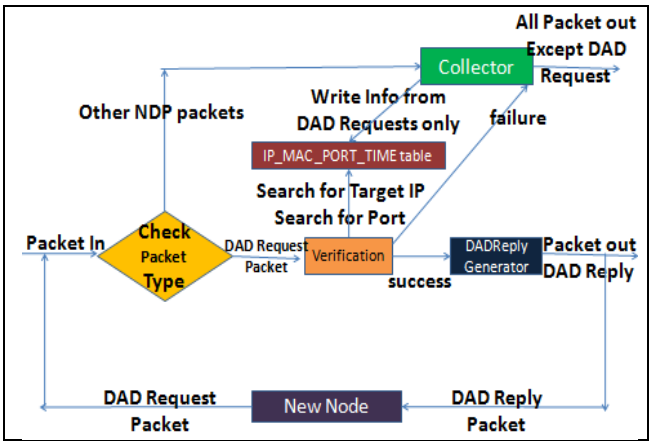

Figure 1. Workflow inside the Central Arbiter enabled SDN controller

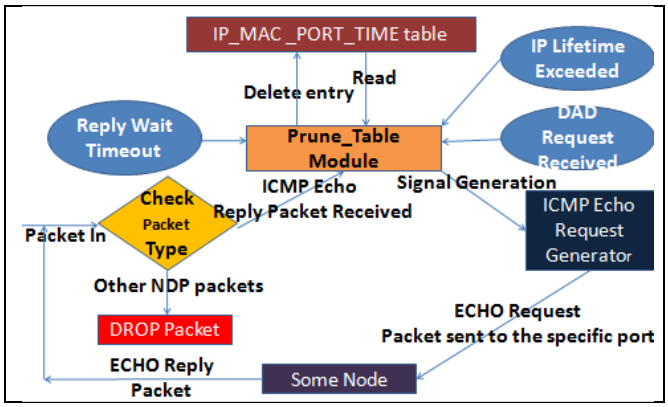

Figure 2. Pruning logic for the IP_MAC_PORT_TIME table

B. *Modules*

The proposed solution consists of four modules, which are as follows:

i) Collector Module

The collector module is designed to read all ICMPv6 packets. It populates the IP_MAC_PORT_TIME table on new DAD requests only. A limit is also imposed on the number of MAC addresses and IP addresses that can be associated with one switch port. Every new IP address to the port association, which does not exceed the allowed per port level MAC and IP limits is inserted in this table. The input to the module is the DAD request packet. The module connects to a database and inserts new entries into the table. The table will be populated just after a DAD request is received and it is verified that no such IP address is present in the network, as well as the switch port, and it does not exceed the maximum IP/MAC address associations. The module can also make permanent entries with value zero (0) in the

 ISBN: 978-1-61208-582-1

timestamp field. This is required for static IP addresses. This table is made persistent across reboots.

ii) Verification Module

The verification module receives all DAD request packets. It then parses the packet and extracts the target IP of the packet. Then, it searches for that IP in the IP_MAC_PORT_TIME table. If an entry for the target IP is present in the table, then success is reported by the module, otherwise failure is reported. In the case of success, the packet is passed on to the DADReplyGenerator module for further action. In the case of failure, the packet is passed on to the collector module.

iii) DAD Reply Generator Module

This module is responsible for fabricating a DAD reply packet. The input to the module is the DAD request packet. The target IP and the switch DPID, along with the port number, are extracted from the packet by the module. The controller fabricates a Neighbor Advertisement (NA) packet on behalf of the node which owns the target IP. This fabricated DAD reply is then dispatched to the node on the switch port from where the DAD request was received.

iv) Prune_IP_MAC_PORT_TIME table module

This module is responsible for pruning the table entries after a configured time (one day as per RFC 4941) has elapsed and/or the node with an IP-MAC-PORT association in the table is no longer active. The state of a node is confirmed at the time when a new DAD REQUEST packet is received for an existing IP address in the table. The state is confirmed by sending a heartbeat packet to the port on which the IP address was last associated and upon reception of ICMP ECHO REPLY packet. The timestamps of the entries for IPv6 addresses from which a reply is received within a specified period are updated. If the entry is older than a configured time, then it is pruned. The table can also be pruned manually by the administrator.

VI. TEST SETUP

A laptop with a single Intel core i5-4200U processor (1.6 GHz), 8 GB RAM and 200 GB free hard disk space has been used as the physical machine for the simulation setup. Oracle Virtualbox version 5.1.22 has been used as virtual machine manager to load two Virtual Machines (VM) on the laptop. The test setup is based on Mininet emulator version 2.2.1, SDN RYU controller version 4.13, THC toolkit version 3.2 and Wireshark version 1.10.6. The freely available "simple_switch_13.py" python application of the RYU controller has been extended with the proposed central arbiter module.

The Mininet emulator and SDN RYU controller are run on two different virtual machines. Both the virtual machines use one core of the processor (1.6 GHz), 1 GB RAM and 16GB of storage. The Mininet emulator VM and the RYU controller VM are on the same network.

The network topology is as depicted in Figure 3. The Mininet topology consists of two OpenFlow version 1.3 compatible Openvswitch virtual switches. Each switch is further connected to three nodes. The topology is a flat network with no routing node since we want to test DAD behavior only. One of the hosts, h3, is configured to act as a rogue node generating DAD DoS traffic. This node is capable of performing all three types of DAD DoS attacks, as mentioned in Section II. The response of the central arbiter configured SDN controller, to all three types of attacks was observed separately in three different test cases. In the first test case, this rogue node is used for generating DAD NA replies for every DAD NS request that is generated by new nodes joining the network, thus performing reactive attacks. In the second test case, the node h3 was programmed to perform guessing attacks by generating DAD NA replies for random IPv6 addresses. The guess is done using the information about earlier assigned addresses on the network. In the third test case, h3 was programmed to generate DAD DoS flooding attack by claiming all IPv6 addresses for the network prefix.

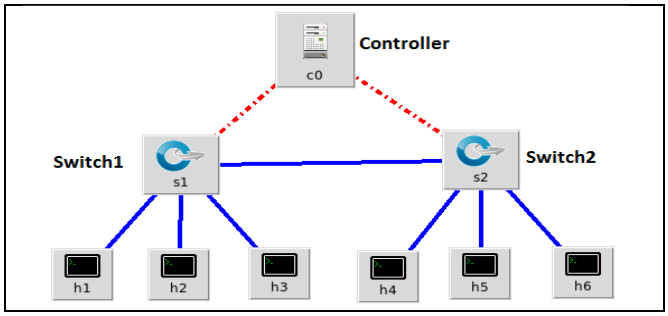

Figure 3. Topology of the test setup with 6 nodes

VII. RESULTS

A) Effectiveness of central arbiter approach in handling DAD DoS attacks:

In the test setup, firstly, we performed a test by disabling the central arbiter module in the SDN RYU controller, for which the RYU controller with the unmodified "*simple_switch_13.py*" script was invoked. For achieving this "*ryu-manager ryu/ryu/app/simple_switch_13.py*" command is executed on the VM configured as controller. On the attacker host, named h3, the command "*dos-new-ip6 h3-eth0*" which is available in the THC toolkit is executed. The output of the command is as shown in Figure 4.

Figure 4: Screenshot of launch of DAD Denial of Service attack in h3

This command configures host h3 to generate DAD NA (ICMPv6 Type 136) reply packets for every DAD NS (ICMPv6 Type 135) request packet received on the network. Next, the addition of a new node is simulated by manual assignment of a new IPv6 address to the host named h6,

using the "*ifconfig h6-eth0 inet6 add fec0::6/64*" command. It was observed that the host named h3 which is configured as the attacker, received the multicasted DAD NS packet and responded by spoofing the DAD NA reply packet, claiming to be the owner of requested IPv6 address. The screenshot of the output, as generated on the attacker host h3, is shown in Figure 5.

Figure 5: Screenshot showing the spoofed packet log on h3

This was further confirmed by executing the "ip addr sh" command on host h6. The output containing the highlighted "tentative dadfailed" line on host h6 is shown in Figure 6.

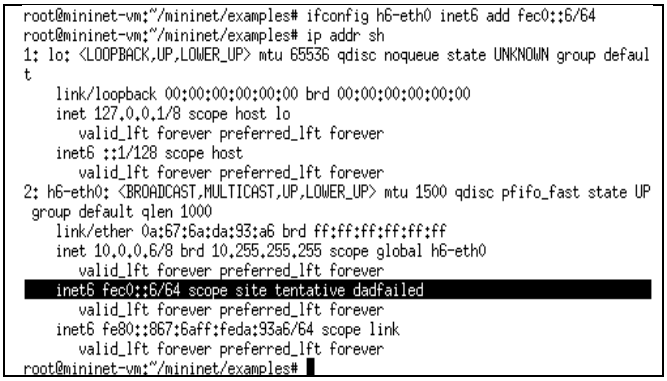

Figure 6. Screenshot of commands and output generated in h6 while testing non duplicate IPv6 address assignment with central arbiter module disabled on the controller and attacker active in h3

Next, we performed the IP address assignment by disconnecting the attacker node. In case of an unused IPv6 address assignment and in the absence of the attacker host named h3, the host named h6 could complete the IPv6 address assignment process and get connected to the network as expected. It is shown in Figure 7.

Figure 7. Screenshot of commands and output generated in h6 while testing non duplicate IPv6 address assignment with central arbiter module disabled on the controller and in the absence of the attacker

After this, we enabled the attacker host h3 and the central arbiter module in the controller. Then, we assigned an IPv6 address to the host named h6 using the "*ifconfig h6-eth0 inet6 add fec0::6/64*" command. This time the attacker host h3 did not receive the DAD NS request packet and hence could not perform DAD DoS attack. Thus, this address assignment was successful. The combined screenshots of host h3 showing active attacker in h3 and results of the IP address assignment and displaying commands in host h6 are shown in Figure 8. This shows that the central arbiter module in the controller effectively blocked the DAD related NS packets from reaching other hosts of the network. This is further confirmed by checking the log messages on the controller enabled with central arbiter module. The screenshot on the controller is as shown in Figure 9.

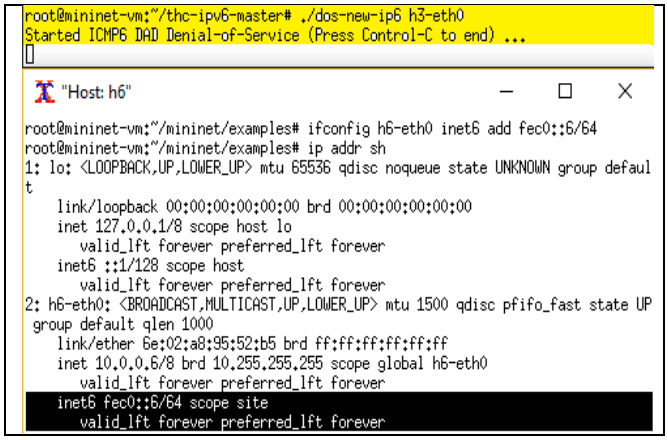

Figure 8. Screenshot of commands and output generated in h3 and h6 while testing non duplicate IPv6 address assignment with central arbiter module enabled on the controller and attacker active in h3

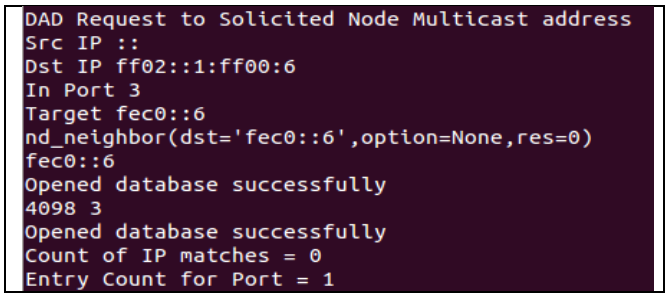

Figure 9. Screenshot displaying log information on the controller while testing non duplicate IPv6 address assignment with central arbiter module enabled on the controller and attacker active in h3

Finally, to check whether the central arbiter correctly sends DAD replies in case of true duplicate addresses on the network, the IPv6 address of host h6 was duplicated by manually assigning the address to host named h1 using the "*ifconfig h1-eth0 inet6 add fec0::6/64*" command.

 ISBN: 978-1-61208-582-1

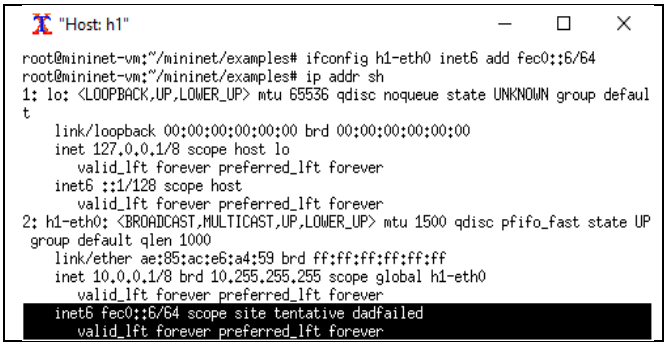

Figure 10. Screenshot displaying dadfailed message on h1 while testing duplicate IPv6 address assignment with central arbiter module enabled on the controller and attacker active in h3

It was observed that in this case, the central arbiter module correctly identified the duplicate IPv6 address and sent the NDP DAD NA reply to host h1, as shown in Figure 11. This caused the IP address assignment process to end without permanent IP address assignment on node h1, as shown in Figure 10.

Figure 11. Screenshot displaying log information on the controller while testing duplicate IPv6 address assignment with central arbiter module enabled on the controller and attacker active in h3

It is verified that the Central Arbiter approach presented in the paper is able to effectively detect and mitigate DoS attacks on DAD in IPv6 networks. The design goals of not introducing any change in NDP, and not increasing client configuration and computation complexity in the proposed solution, are fully met as explained in Table II.

TABLE II. CENTRAL ARBITER EFFECTIVENESS

Type of DoS Attack on DAD	Additional Client Configuration Complexity for protection	Additional Client Computation Complexity for protection	Results with Logic in Central Arbiter providing protection
Reactive	None	None	**PROTECTED** All DAD Requests were blocked by central arbiter from reaching other nodes and DAD replies were sent by central arbiter only for already in use IPv6 address.
Guessing	None	None	**PROTECTED** The maximum limits defined on Per Switch Port IP and MAC address associations did not permit more than the allowed number of IPv6 address requests from a node attached to a switch port.
Flooding	None	None	**PROTECTED** The maximum limits defined on Per Switch Port IP and MAC address associations did not permit more than the allowed number of IPv6 address requests from a node attached to a switch port.

B) *DAD process timing comparison with and without central arbiter module:*

DAD timing tests were performed in the network with the topology as shown in Figure 3. The time taken to complete DAD process was observed in 5 cases that are mentioned in Table III.

TABLE III. DAD PROCESS TIMING COMPARISON

S.No.	Time taken in DAD process With Central Arbiter (sec)	Time taken in DAD process Without Central Arbiter (sec)	Delay introduced by central arbiter scheme (sec)
1.	0.888826	0.14075	0.748076
2.	0.939575	0.593365	0.34621
3.	1.018122	0.67268	0.345442
4.	0.474215	0.311938	0.1662277
5.	0.655199	0.54579	0.109409

The results indicate that, on an average, a delay of about 0.34 seconds is introduced in the central arbiter scheme. This is because data processing and searching on sqlite database (used for persistent storage of all IPv6 addresses currently in use on the network) is involved in the process.

VIII. CONCLUSIONS & FUTURE WORK

In any IP network, IP address assignment is the first step that needs to be completed before a node can start communicating with other nodes. Duplicate Address Detection phase in IPv6 address assignment step needs to be completed for successful assignment of an IPv6 address. IPv6 uses NDP control messages for updating and checking the status of the network. NDP security vulnerabilities lead to security loopholes in the network. All existing mechanisms are complex to implement.

IPv6 networks can be controlled efficiently by central arbitration of NDP messages. The central arbiter approach to

 ISBN: 978-1-61208-582-1

mitigate DoS attacks on DAD in IPv6 networks is proposed in this paper. It achieves the desired goal by intelligently filtering DAD requests and corresponding replies. The simulation results have shown that the DAD process can be completed with additional delay of the order of 0.34 seconds, using the approach presented in this paper. This approach has been demonstrated to work in Software Defined Networks.

The management (purging/updating of stale entries) of IP_MAC_PORT_TIME table, introduced in the presented approach is critical for the functioning of the central arbiter.

The central arbiter approach presented in the paper may seem to introduce single point of failure by having dependency on a SDN controller for controlling the DoS attacks. With Active-Active failover mode of operation of SDN controllers becoming popular, this concern can be addressed effectively. Further, since no changes in the NDP messages are suggested in this approach, the failure of a SDN controller will only lead to a network without DAD DoS protection and the network will continue to work under the usual threat of DAD DoS attacks.

The scalability of the proposed approach depends on the maximum permissible size of the IP_MAC_PORT_TIME table in the SDN controller, which in turn will be governed by the amount of the primary memory availability in the controller. With the usage of fast data insertion and search algorithms, the proposed solution can scale to work in the largest of the IPv6 networks with 2^64 nodes. Since, practically such large networks are not foreseen in near future, it can safely be assumed that the proposed approach will scale to work in all practical IPv6 networks.

Further, the work presented here programs SDN controller as a central arbiter in such a way that it can emphatically and proactively confirm whether an IP address is already in use in that network without completing the DAD process, which involves timeout. This concept can be further extended to achieve fast IP address assignments by making minor changes in the NDP. The reduction of DAD timeout is a major requirement of fast handovers in mobile networks. Further, the heartbeat mechanism presented in the paper for the management of this table can be improved.

REFERENCES

[1] S. Deering, and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, [retrieved: July, 2017].

[2] T. Narten, E. Nordmark, E, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, [retrieved: July, 2017].

[3] S. Thomson, T. Narten, and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, [retrieved: July, 2017].

[4] P. Nikander, J. Kempf, and E. Nordmark, “IPv6 Neighbor Discovery (ND) Trust Models and Threats,” Internet rfc 3756 edn. 2004, [retrieved: July, 2017].

[5] THC-IPv6, https://www.thc.org/thc-ipv6/, [retrieved: July, 2017]

[6] J. Arkko, J. Kempf, B. Zill, and P. Nikander, “Secure Neighbor Discover (SEND),” Internet rfc 3971 edn. 2005, [retrieved: July, 2017].

[7] A. AlSa'deh, and C. Meinel, “Secure neighbor discovery: Review, challenges, perspectives, and recommendations”. Security & Privacy, IEEE, 10(4), 26-34, 2012.

[8] H. Rafiee and C. Meinel, "SSAS: A simple secure addressing scheme for IPv6 autoconfiguration," 2013 Eleventh Annual Conference on Privacy, Security and Trust, Tarragona, 2013, pp. 275-282.doi: 10.1109/PST.2013.6596063.

[9] S. Praptodiyono, R. K. Murugesan, I. H. Hasbullah, C. Y. Wey, M. M. Kadhum and A. Osman, "Security mechanism for IPv6 stateless address autoconfiguration," 2015 International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT), Bandung, 2015, pp. 31-36.

[10] S. U. Rehman and S. Manickam, “Improved Mechanism to Prevent Denial of Service Attack in IPv6 Duplicate Address Detection Process” International Journal of Advanced Computer Science and Applications(IJACSA), 8(2), 2017.

[11] E. Haleplidis, K. Pentikousis, S. Denazis, H. Salim, J. Meyer, and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, [retrieved: July, 2017].

[12] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, “Software defined networking: state of the art and research challenges,” Computer Networks, vol. 72, pp. 74–98, 2014.

[13] RYU, https://osrg.github.io/ryu/, [retrieved: July, 2017].

[14] V. Nicolls, N. A. Le-Khac, L. Chen and M. Scanlon, "IPv6 security and forensics," *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, Dublin, 2016, pp. 743-748.

[15] R. Radhakrishnan, M. Jamil, S. Mehfuz and M. Moinuddin, "Security issues in IPv6," *Networking and Services, 2007. ICNS. Third International Conference on*, Athens, 2007, pp. 110-110.

[16] A. S. Ahmed, R. Hassan and N. E. Othman, "Improving security for IPv6 neighbor discovery," 2015 International Conference on Electrical Engineering and Informatics (ICEEI), Denpasar, pp. 271-274, 2015.

[17] R. Hassan, A. S. Ahmed, and N. E. Osman, “Enhancing security for ipv6 neighbor discovery protocol using cryptography”. American Journal of Applied Sciences, 11(9), 1472-1479, 2014.

[18] F. Xiaorong, L. Jun, and J. Shizhun, “Security analysis for IPv6 neighbor discovery protocol,” in Proceedings of the 2nd International Symposium on Instrumentation and Measurement (IMSNA '13), pp. 303–307, Toronto, Canada, December 2013.

[19] A. S. Ahmed, R. Hassan and N. E. Othman, "Improving security for IPv6 neighbor discovery," 2015 International Conference on Electrical Engineering and Informatics (ICEEI), Denpasar, pp. 271-274, 2015.

[20] S. U. Rehman and S. Manickam, “Significance of Duplicate Address Detection Mechanism in Ipv6 and its Security Issues:A Survey,” Indian Journal of Science and Technology, vol. 8(30), 2015.

[21] S. Praptodiyono, I. H. Hasbullah, M. M. Kadhum, R. K. Murugesan, C. Y. Wey and A. Osman, "Improving Security of Duplicate Address Detection on IPv6 Local Network in Public Area," 2015 9th Asia Modelling Symposium (AMS), Kuala Lumpur, pp. 123-128, 2015.

[22] C. Zhang, J. Xiong and Q. Wu, "An efficient CGA algorithm against DoS attack on duplicate address detection process," 2016 IEEE Wireless Communications and Networking Conference, Doha, pp. 1-6, 2016.

[23] IEEE 802.1x, http://www.ieee802.org/1/pages/802.1x.html, [retrieved: July, 2017].

 ISBN: 978-1-61208-582-1

[24] E. Andreeva, B. Mennink, B. Preneel, “Open problems in hash function security. Designs, Codes and Cryptography,” vol. 77, pp. 611-631, 2015.

[25] K. Bhargavan, and G. Leurent “Transcript collision attacks: Breaking authentication in TLS, IKE, and SSH,” NDSS, 2016.

[26] V. Shoup, “Fast and provably secure message authentication based on universal hashing”. In Advances in Cryptology—CRYPTO’96, pp. 313-328, 1996.

[27] T. Krovetz, “UMAC: Message authentication code using universal hashing,” Internet RFC 4418, 2006, [retrieved: July, 2017].

[28] G. Song, and Z. Ji, “Novel Duplicate Address Detection with Hash Function,” PLoS ONE 11(3): e0151612, 2014, https://doi.org/10.1371/journal.pone.0151612.

[29] Y. Lu, M. Wang, and P. Huang, “An SDN-Based Authentication Mechanism for Securing Neighbor Discovery Protocol in IPv6,” Security and Communication Networks, vol. 2017, Article ID 5838657, 9 pages, 2017. doi:10.1155/2017/5838657.

[30] Tian, K. Butler, J. I. Choi, P. D. McDaniel, P. Krishnaswamy, "Securing ARP/NDP From the Ground Up," in IEEE Transactions on Information Forensics and Security , vol.PP, no.99, pp.1-1doi: 10.1109/TIFS.2017.269598.

 ISBN: 978-1-61208-582-1