Efficient Domain Name Resolution using SDN Architecture

Aswathy A, Arjun D Shenoy, Leena Vishnu Namboothiri

ABSTRACT---SDN is a dynamic, programmable network that follows layered architecture. The Domain Name System can be implemented in an SDN network, due to which the structural efficiency of SDN architecture can be employed on to Domain name system. For this purpose, we propose a new framework which works with four algorithms InEx (Internal/ External Check), PlyIn (for internal query), PlyOut (for external query) and FiSt (Find the Stored).

Index Terms - SDN (Software Defined Network), DNS (Domain Name System), OpenFlow Protocol.

I. INTRODUCTION

The significance of cloud computing has increased in such a way that traditional networking architectures are to be changed to employ highly scalable, efficient and cost-effective networks. This led to the development of a new architecture called the Software Defined Networking (SDN). SDN has a highly decoupled design which makes it flexible and easy to program which has created revolutionary impacts and gained high popularity.

A. SDN Architecture

SDN has a scalable, highly dynamic, de-coupled architecture with a centralized controller and has an exposure of abstract network resources. Due to the hierarchical levels Scalability, Modularity and Security can be ensured [9]. The control plane is called the "Brain" of the network. The north bound interface contains the user applications and it collects/requests resources from/to the controller whereas the southbound consists of hardware devices that perform forwarding functions [2][1][3]. Out of many protocols SDN uses OpenFlow protocol [2]. The SDN architecture is depicted in Fig 1.

B. DNS Architecture

DNS is a system for converting Domain Names to IP address. It is a distributed database implemented in a hierarchy of DNS servers, and an application-layer protocol that allows hosts to query the distributed database. It is a hierarchical decentralized naming system for computers,services, or other resources connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities.

In this paper, we suggest a new framework for DNS using SDN which will make the working of DNS more efficient and provide structural efficiency.

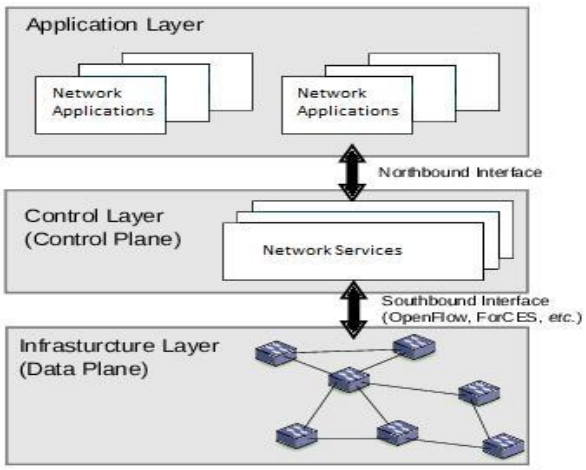

Fig 1: SDN Architecture

II. LITERATURE REVIEW

SDN network is programmable and centralized. But traditional network is hardware-centric. Any small change in a traditional framework was found costly. We observed that, SDN is an approach that can enable research innovations and simplified network management [1] [5]. We studied that SDN is an emerging technology where control plane and data plane are differentiated [9] [2] [5]. We noticed that SDN has three main components; Control Plane, Data Plane and Application Plane. The flow table in Switches is defined by the OpenFlow protocol [10] [8]. SDN is highly scalable, private and incremental deployment is possible [10].

We understood that SDN provides Quality of service for applications since they bring centralized view, and more fine-granular flow management opportunities [3]. DNS amplification attacks can be mitigated by using SDN architecture which led to the development of idea of this paper [4].

Currently, SDN is used in government organizations, industry and campus networks [5].

We also noted that switches contain flow table and flow table contains three parts named as Rule Field, Action Field, Stats Field and switches forward packets based on flow table [1]. Due to the decoupled design, it was found that SDN makesthe network system programmable and provides unified control to the applications which makes it easy for researchers, system designers and administrators to design new network functions and protocols [6]. We further studied that by using SDN, it is possible to enable on-

Revised Manuscript Received on May 29, 2019.

Aswathy A, PG Student, Dept of Computer Applications, Amrita VishwaVidyapeetham, Kochi, Kerala

Arjun D Shenoy,PG Student, Dept of Computer Applications, Amrita VishwaVidyapeetham, Kochi, Kerala.

Leena Vishnu Namboothiri, Assistant Professor, Dept of Computer Applications, Amrita VishwaVidyapeetham, Kochi, Kerala.

demand resource allocation, self-service provisioning, virtualized networking and secure services [7]. Since, it was difficult for routers to process domain names, every domain name was identified in network with an IP address, and therefore, mapping was mandatory. Domain Name System is responsible for mapping domain names to IP address [12]. Port 53 is used by DNS which runs over UDP protocol [11]. Other services provided by DNS include Host Aliasing, Mail server Aliasing, Load Distribution [13].

By using SDN in traditional network, QoS can be enhanced. QoS parameters include bandwidth, delay, jitter, loss etc [3]. Considering the things mentioned above, we propose an efficient framework for DNS using SDN which makes DNS work with all the qualities of SDN.

III. PROPOSED SYSTEM

In a conventional system, when a domain name mapping is required, the client gives request to firewall and firewall checks local DNS cache, if local DNS server does not have the response to the query, the local DNS server will give an iterative request to the root DNS server. But this is very time consuming. So, we are proposing a new framework for domain mapping using SDN architecture which is highly scalable, efficient and cost-effective network. The proposed architecture is depicted in Figure 2.

The components in SDN network act as DNS server, firewall and routers. The SDN controller contains local DNS server and firewall. The client machines are connected to openflow enabled switches which have two tables. The flow table and the Switch_Table.

The flow table contains the actions associated with processing packets. The proposed Switch_Table contains the frequently accessed sites by different users.

Table 1: Switch_Table

AFS	DOMAIN NAME	IP ADDRESS	TIMELEFT
111	www.google.com	172.217.26.206	10
110	www.youtube.com	172.217.160.142	7
100	www.facebook.com	157.240.7.35	3

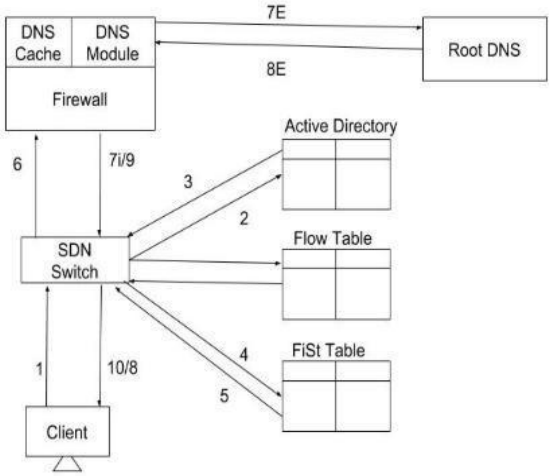

Fig 2: Proposed DNS Framework

Table 1 contains four columns; The first column is AFS which is abbreviated as Admin, Faculty and Student.

If AFS= 111 then Admin, Student and Faculty can access.

If AFS= 110 then Admin and Faculty can access.

If AFS= 100 then only Admin can access.

The fourth column contains the time left, which for every entry will be 20 by default. An algorithm will work in regular intervals which will reduce the time left by 2 for every 5 minutes. As the time left is less than or equal to 2, the entry will be removed from the table. Every time a particular domain name is accessed, the time left is changed to 20.

When a request arrives the SDN switch, it will first trigger the InEx Algorithm which works to identify whether a request is Internal or external to the network. From the InEx algorithm, according to the condition, PlyIn algorithm or PlyOut algorithm is triggered. The PlyIn algorithm works for the internal network, whereas the PlyOut algorithm works for the external network.

A. Algorithm 1: InEx Algorithm

1. Start
2. If port number of request=53then,
   2.1. Ifauthoritativeserverpartofrequest=asas.kh.amrita.edu
      2.1.1. Trigger PlyIn Algorithm
   2.2. Else
      2.2.1. Trigger PlyOut Algorithm
   2.3. End IF
3. Stop

B.Working of InEx Algorithm

When a request from a client arrives the Switch, it checks whether the request is a DNS request. If it is a DNS request, it will check whether it is an internal or an external query.

If internal, the request will contain asas.kh.amrita.edu, and PlyIn is triggered. Otherwise, PlyOut is triggered. The flowchart of InEx algorithm is shown in Figure 3.

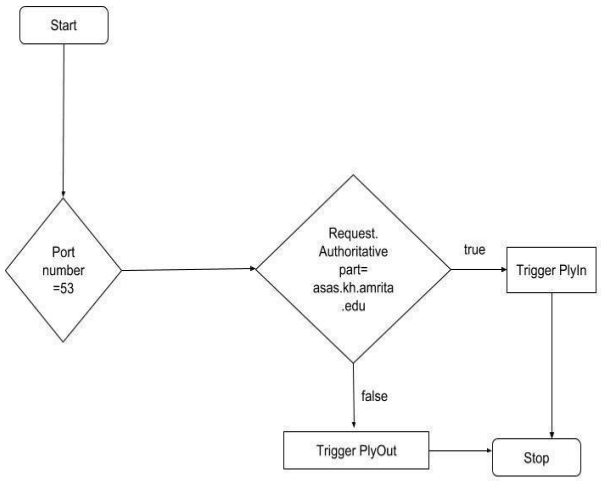

Figure 3: InEx Algorithm

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

C. Algorithm 2: PlyIn Algorithm

1. Start
2. Trigger FiSt
   - 2.1. If (FiSt(Request.DomainName) = Switch_Table.Domain Name)
     - 2.1.1. If (FiSt(Request.Domain Name.Access) = denied)
       - 2.1.1.1. Deny the Request
     - 2.1.2. Else
       - 2.1.2.1. Passthecorresponding IP address to the client.
   - 2.2. Else
     - 2.2.1. Forward to DNS Module inthe controller. DNS module responds to the client via switch. An entry is made in the Switch_Table.
3. Stop

D.Working of PlyIn Algorithm

When PlyIn algorithm is triggered, it first checks whether the Domain Name of Request is present in Switch_Table or not. If present, it will check for access. If Domain Name is not present in table, it is forwarded to the local DNS module in the controller. The flowchart of PlyIn algorithm is depicted in Figure 4.

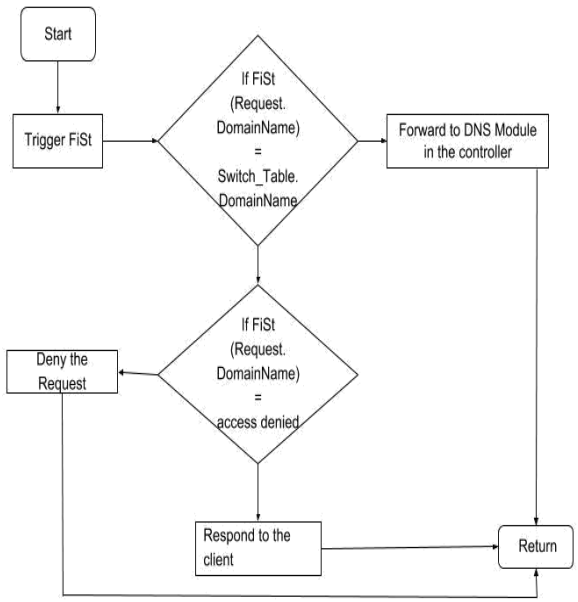

Figure 4: PlyIn Algorithm

E. Algorithm 3: PlyOut Algorithm

1. Start
2. Trigger FiSt
   - 2.1. If (FiSt(Request.DomainName) = Switch_Table.Domain Name)
     - 2.1.1. If(FiSt(Request.DomainName.Access)= denied)
       - 2.1.1.1. Deny the Request
     - 2.1.2. Else
       - 2.1.2.1. Pass the corresponding IP address to the client.
   - 2.2. Else
     - 2.2.1. Forward to DNS Module in the controller.
       - 2.2.1.1. If (DNScache.DomainName = Request.Domain Name)
         - 2.2.1.1.1. Respond to client via switch. An entry is made in the Switch_Table.
       - 2.2.1.2. Else
         - 2.2.1.2.1. Query forwarded to the root server as an iterative query. When Response comes to SDN controller, it is forwarded to client via switch. An entry is made in the Switch_Table.
3. Stop

F.Working of PlyOut Algorithm

When PlyOut algorithm is triggered, it first checks whether the Domain Name of Request is present in Switch_Table or not. If present, it will check for access. If Domain Name is not present in table, it is forwarded to the local DNS module in the controller. If the DNSCache has the Domain Name of Request, it will respond back to the client. Otherwise, it will forward the request to the root DNS server. The flowchart of PlyOut algorithm is depicted in Figure 5.

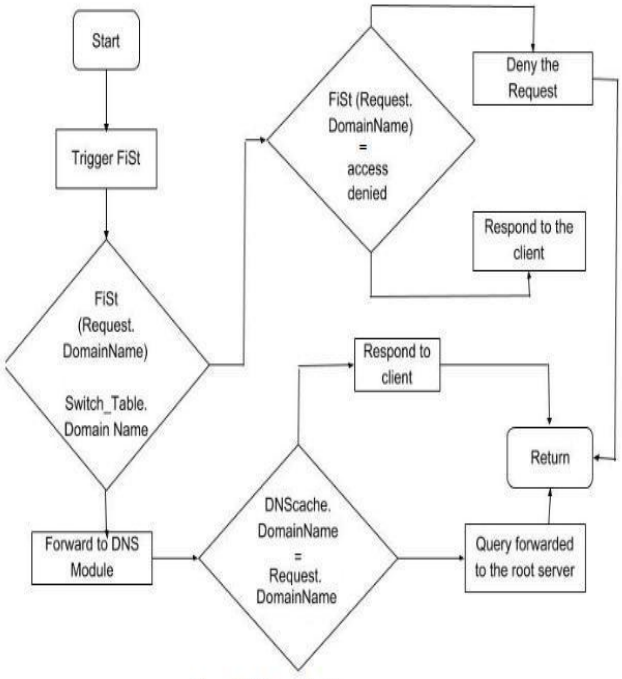

Figure 5: PlyOut Algorithm

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

G. Algorithm 4: FiStAlgorithm(Domain Name)

1. Start
2. If Switch_Table.Domain Name= Request.Domain Name
   - 2.1. If Active Directory.Role=Admin
     - 2.1.1. Access=granted
   - 2.2. If Active Directory.Role=Faculty and (AFS=111 or AFS=110)
     - 2.2.1. Access=granted
   - 2.3. If Active Directory.Role=Student and (AFS=111)
     - 2.3.1. Access=granted
   - 2.4. Else
     - 2.4.1. Access=Denied
   - 2.5. EndIf
3. Stop

H. Working of FiSt Algorithm

The algorithm FiSt(Find the Stored) finds the entries in the Switch_Table. There are mainly three users; Admin, Faculty and Student. When a user logs in, the role of the user is found from active directory. According to the Domain Name and Role, the algorithm returns access granted /denied.

IV. CONCLUSION

In this paper, we describe the implementation of local DNS using SDN architecture and how the structural efficiency of SDN is employed in DNS. Further, we have devised four algorithms that that are implemented in the proposed framework. These algorithms work together to process the domain name queries from different users in the local network. Since the frequently accessed domain names are cached, it reduces time delay.

REFERENCES

1. Neeraja U Gautham, M.A. Saimi Thresia and Leena Vishnu Namboothiri, Efficient Domain Name Resolving in SDN Using ReSt and HDNM.
2. Akhil Raj, Anjali S Bhat and Leena Vishnu Namboothiri, Effective Threshold Defence Against DOS Attack on SDN Controller.
3. Murat Karakus, Arjan Durresi, Quality of Service (QoS) in Software Defined Networking (SDN): A survey, Journal of Network and Computer Applications 80 (2017), 200–218.
4. Soyoung Kim, Sora Lee, Geumhwan Cho, Muhammad Ejaz Ahmed, Jaehoon (Paul) Jeong, and Hyoungshick Kim(B), Preventing DNS Amplification Attacks Using the History of DNS Queries with SDN.
5. Jacob H. Cox , Jr.1, Joaquin Chung 2, Sean Donovan 2, Jared Ivey 2, Russell J. Clark 3, (Member, Ieee), George Riley2 and Henry L. Owen, Advancing Software-Defined Networks: A Survey.
6. Akram Hakiri a, Aniruddha Gokhale c, Pascal Berthou a,Douglas C. Schmidt c, Thierry Gayraud a, Software-Defined Networking: Challenges and research opportunities for Future Internet.
7. Wenjuan Li a, Weizhi Meng, Lam For Kwoka, A survey on Open Flow-based Software Defined Networks:Security challenges and countermeasures.
8. Syed Taha Ali, Member, IEEE, Vijay Sivaraman, Member, IEEE, Adam Radford and Sanjay Jha, A Survey of Securing Networks using Software Defined Networking
9. Open Network Foundation, SDN Architecture, issue 1,june2014https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf
10. Cisco, a platform to understand hardware and networkinghttps://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-59/161-sdn.html
11. Jaeyeon Jung, Emil Sit, Hari Balakrishnan, Robert Morris, DNS Performance and the Effectiveness of Caching, IEEE/ACM Transactions on Networking 10(5) (2002).
12. Anees Shaikh, RenuTewari, Mukesh Agrawal, On the Effectiveness of DNS-based Server Selection, Proceedings. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies 3 (2001), 1801-1810.
13. Florian Weimer, Passive DNS Replication, FIRST conference on computer security incident (2005).

AUTHORS PROFILE

ASWATHY A
PG Student, Master of Computer Application,Department of Computer Applications, AmritaVishwaVidyapeethamSchoolofArtsandSciences,Brahmasthanam,EdappallyNorthP.O. Kochi - 682 024, Kerala.

ARJUN D SHENOY
PG Student, Master of Computer Application,Department of Computer Applications, AmritaVishwaVidyapeethamSchoolofArtsandSciences,Brahmasthanam,EdappallyNorthP.O. Kochi - 682 024, Kerala.

LEENA VISHNU NAMBOOTHIRI
Assistant Professor, Department of ComputerApplications, , Amrita VishwaVidyapeethamSchool of Arts and Sciences, Brahmasthanam,Edappally North P.O. Kochi - 682 024, Kerala.Qualifications: Master of Computerapplications, CCNA. Paper Publications:Detection of incongruent firewall rules andflow rules in SDN, Advances in Intelligent Systems and Computing Volume 517, 2017, Pages 13-21, ISSN- 2194-5357 and Prevention of black hole attack in MANETs using enhanced AODV protocol, International Journal of Applied Engineering Research Volume 10, Issue 55, 2015, Pages 2037-2042, ISSN- 0973-4562