A Novel Method to Detect and Defend the MITM attack in Software Defined Networks.

N. Narendra[1], Y. Naveen[2], K. Hari sairam[3], S. Keerthi[4], E. Suresh Babu[5]

Abstract- Maintenance and Management of the networks has become a highly complex activity. Traditional networking is highly rigid because of the non-programmable networks. The concept of Programmable networks has recently gained interest because of the emerging of Software-Defined networks. With this SDN the complexity of the networking has been reduced. For wide scale electronic communications which involve activities such as sharing of sensitive information and financial transactions Internet has emerged as a medium. In this process the ease of occurrence of the attacks will be very high. One such attack is the MITM attack. This paper outlines the necessity for the development of the intrusion detection mechanism. This paper explains the various steps that are to be taken to implement the detecting and defending mechanisms and the necessary components that are required.

Keywords— Traditional networking, Software-Defined networking (SDN), MITM attack.

I. INTRODUCTION

Internet is a most prominent innovation that has empowered more essential access to data by giving new modes of correspondence among individuals and organizations and on a very basic level changed the way we work. Unfortunately, the Internet is currently making obstructions to innovations in the network administration. While some created innovative solutions for an extensive variety of network administration challenges, resultantly there has been a little progress in deploying those solutions in the Internet. The current Internet structural planning created among the different stakeholders have turned into a genuine obstacle for its proceeding with growth and development. Here comes the time for the proposal of Programmable networks to facilitate the network evolution.

In particular Software-Defined Networking is a rising innovation that helps to change the present day issues by differentiating the system's control from the network devices such as routers and switches and making a centralized control of the system. In SDN architecture the controller is implemented in a server, separated from switches that forward traffic. The separation of the forwarding hardware from the control plane simplifies the network visualization and makes it easy to deploy new applications and protocols. This architecture includes the entire network platform.

The bottom level of the architecture involves the physical network equipped with routers and Ethernet switches. This gives the data plane. The middle level of the architecture contains the controller which helps to set up and tear down the flows and paths in the network. The controller uses the information about the availability and the requirement obtained from the network devices through which the traffic flows. Application Programming Interface (API) known as southbound API acts as a link between the central level and bottom level. The connections in the controller are through east and west bound APIs and controller application interface is through northbound API. The upper level indicates the various applications which network operators can provide.

Computer Networks are typically built from various network devices such as routers and switches with routing protocols implemented on them. It becomes difficult for the network operators to handle the configuring issues with these Non-Programmable networks. They need to handle these issues manually and transform them from high level issues to low level issues while adapting the changes to the network conditions. They need to solve these issues by having access only to limited number of tools. As a result the network management and performance modification will become challenging.

The main problem the network operators and the administrators face is due to the integration of Data plane and Control plane in a network node. The control plane looks after the configuration of the node and how to program the paths that are to be used for the data flow. Once the paths are defined, they are pushed to the Data plane. Data forwarding at the hardware level is based on the controlled information. In this approach once the forwarding policy has been defined, the only way to modify the policy is to change the configuration of the network devices. This becomes hard for the network operators who are keen to improve their network with respect to the changing traffic demands and impact of big data.

In SDN the Control plane and Data plane are decoupled and control is moved out of the network node and a centralized control is made. Network Operating system (NOS) [1] gets the information from the API through which the SDN switches and Data plane are controlled. SDN gets the abstract view of the network topology and hosts the applications. Furthermore it provides the facility of Programmable networking at all the levels of the architecture. This helps the applications and network to be aware of each other which improves the potential of the new applications.

In this Section the concept of SDN is explained and Section-2 explains the modelling of Man in the middle attack and Section-3 gives the implementation of the MITM attack in SDN environment using Mininet. Section-4 discusses about detecting and defending mechanisms for MITM attack. Section-5 discusses about the Mininet which is a latest network emulator tool. Section-6 discusses about the results obtained. Finally section-7 concludes this work.

II. Modelling the Man-in-the Middle Attack

Internet connections are prone to different types of attacks. “Man in the Middle” attack is the most general type of attack. The main theme of this attack is to get in between the recipient and the sender, access the traffic, modify it and forward it to the recipient.

The name “Man-in-the-middle” has been in the lime lights of computer security and cryptography since 1994[2].There are so many attacks which resembles the Man in middle attack but the general definition for man in the middle attack is “ Computer security breach in which a malicious user intercepts and possibly alters the data traveling along a network.” [3]. Man-in-the-middle attack is a general type of attack which is generally discussed under Network security and Cryptography and it can be abbreviated in many ways, including MITM, MIM. In this type of attack the attacker gets himself inserted in between the communication of any two systems and he gains access to all the information that the two systems are sharing .The victim systems may not even know that someone is getting access to their information before it is too late. The attacker can do anything with the data like rerouting that data to another system and also knowing the usernames and passwords the two victims are sharing. It is clearly shown in the below figure:

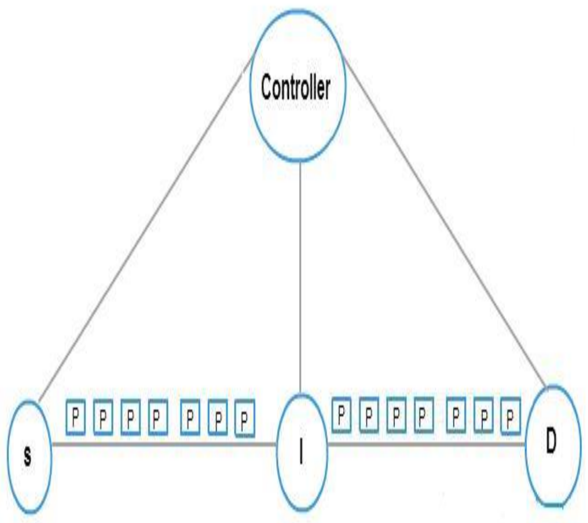


Fig.1. Without Man in the middle attack

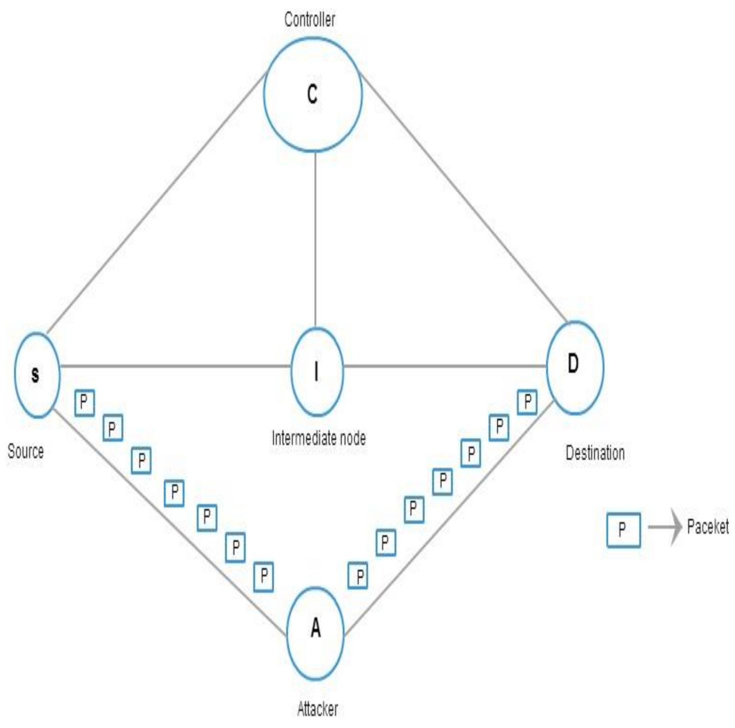


Fig.2.With Man in the middle attack

As the name itself suggests that the attacker impersonates himself in between the two parties who want to communicate with each other .The two parties may be two individual computers or a computer and a web server [Fig. 1].or two web servers. In the above figure before the attack occurs the victim and the server communicates through original communication path. When the attacker starts Man in the Middle attack the communication between them happens via the attacker so that he can observe all the data that is sent through him.

Three-way hand shake mechanism is used by TCP/IP to establish a connection. To send and receive data it uses the data acknowledgments and packet sequencing. Data is shared between the physical layer and application layer. Ethernet is formed at Layer 2 and IP packet datagrams are formed at Layer 3 where as a cryptographic SSL session is established at presentation layer assuming the applications would use it. The major thing to note is that all these layers can effectively participate in MITM attack. Attackers use the packet sniffer initially to steal the session. Once the packets are gathered they choose the layer to attack whether ARP protocol or IP or SSL. Later the attackers inject the packets with the help of some advanced tools to steal the session without any idea of which victim session has been hijacked. Although IP spoofing is used in MITM at its base, it goes further beyond to gain access by selecting sessions from one or many layers to be hijacked.

III. Implementation of Man-in-the-middle attack using Arp-flooding Technique in Mininet

In order to implement Man in the middle attack an emulation tool like Mininet is to be used. First in Mininet the required network topology (Say linear with 4 hosts) must be created and separate terminals for each host must be opened. Then in order to know the devices or hosts which are active in the network at that point of time a command called "arp-scan" is implemented in one of the terminals of that host (Say host 4 i.e h4 is the attacker). Then a list of devices which are connected to that network are shown in the terminal of attacker. The attacker then selects the victims, the conversation between whom he wants to eavesdrop. Let the victims be h1 and h2.In the arp-scan command the attacker keeps the interface as h4-eth0. Here interface means the port of the host which is connected to the LAN network (i.e h4-eth0). Local net is to be mentioned in the arp-scan command because it will automatically generate addresses from network interface configuration and checks whether they are alive at that moment or not.

The attacker (h4) then runs a command "arpspoof" in order to do a Man in the middle attack. But before doing that he should run another command to make the ip_forward file in his system to successfully routes the traffic via him.

In the "arp spoof" command the attacker should specify the target hosts IP addresses and the interface where he has to look on that device.

In the "Arp spoof" command the attacker should mention interface as h4-eth0. At the place of target 1, the IP address of the victim- 1 should be mentioned (i.e 10.0.0.1) and at the target 2 IP address of second target should be mentioned (i.e 10.0.0.2).

When the Arp spoof command is run in the attacker terminal the traffic which is sent from host 1 to host 2 goes through the attacker. But the traffic from host 2 to host 1 is not sent through attacker if the attacker wants that traffic also to pass via him, he should run the above command again but by replacing target 1's IP address with that of target 2 and vice-versa. Now the whole conversation between host 1 and host 2 goes via the attacker i.e host 4.

In this manner the attacker can successfully do a Man in the middle attack and can steal the information between any two hosts and the host will have no idea about the attack that is held on them.

IV. Detecting and defending Man-in-the-middle attack in SDN

Man-in-the-middle attack is a very dangerous attack often under estimated by many people .It is a part of packet sniffing and spoofing attacks. When carried out properly this attack can cause serious damage to the victims. Man-in-the Middle attack can steal important data like usernames and passwords of any accounts the victims are communicating with each other. For example, a person sharing his bank accounts information to his brother via chat application. The attacker can also cause Denial-of-service to the victims by using this attack.

As present day networks are having serious drawbacks one day or the other all networks will migrate to Software Defined Networks. Then also there will be several types of attacks which try to disturb normal behavior of the network. So in this paper we want to present how to detect and defend a Man-in-the-middle attack done on a Software Defined Network. We have already explained how a Man-in-the-middle attack is done in the Section-3. In this section we are going to explain detecting and defending mechanism for Man-in-the-middle attack in SDN.

A. Mechanism:

A Man-in-the-middle attack normally happens when two or more persons communicate with each other using the protocol TCP/IP. As said in Section-2 normally when an attacker want to do a Man-in-the-middle attack (using Arp spoofing).He normally sends spoofed ARP packets into the network by associating his MAC ID with the IP address of the Victim. So that all devices in the network updates their cache with the information in spoofed ARP packets and starts sending their messages to the attacker rather than sending them to the actual destination. So MAC ID and IP address pair of a host should change in order to do a Man-in-the-middle attack. Taking this point into consideration we have designed our detecting and defending mechanism

We have assumed that our network contains systems with the static IP's i.e. Static Network and the MAC ID's of all the systems is also known. As this is a SDN environment we can get all the above information by running some commands in the controller. Then we will make a file with MAC ID of each device, IP address associated with it. Let this file be Standard Hosts file. This file should be manually prepared at the time of network installation and should be updated whenever a new authorized device enters or leaves the network. Then we should find the number of switches in the network by running some commands in the controller and doing some file processing to finally get the number of switches in the network. For every certain span of time we will check the active hosts in the network by running a specific command in each switch present in the network to catch all the ARP flows running through it and then doing some file processing to finally get the MAC ID and the IP information fof each host connected to that switch from those ARP flows. Then the comparison process happens in the following steps:

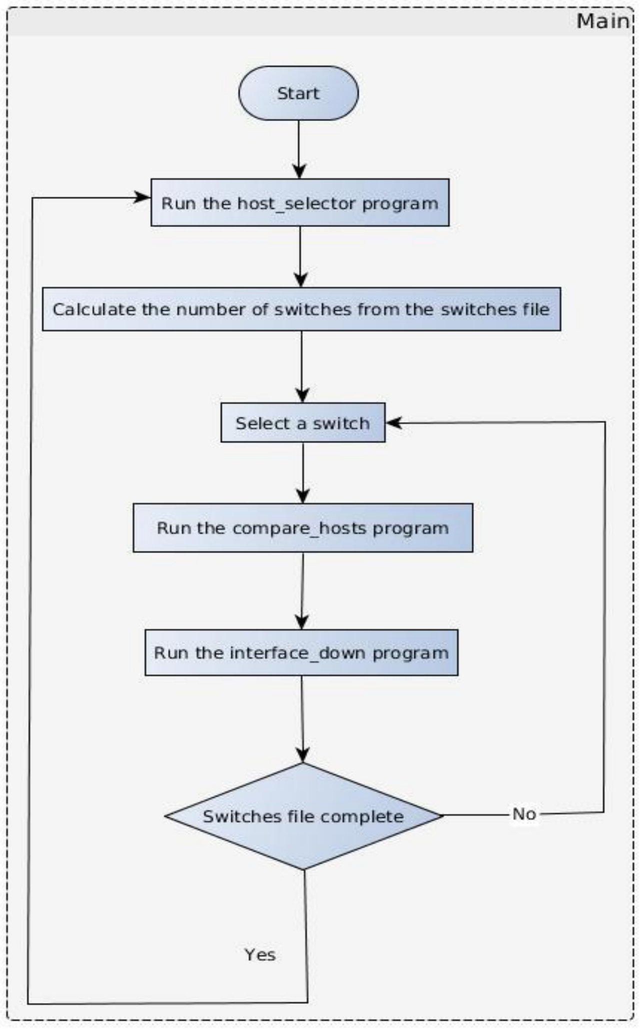
Step 1: Take a MAC ID from an ARP flow of a switch and check whether it is present in the Standard Hosts file.

Step 2: If the MAC ID exists then take the IP address associated with that MAC ID from ARP flow of that Particular switch and check whether that IP is associated to that specific MAC ID or not in the Standard Hosts file.

If the MAC and IP pair matches then the above procedure continues until all the Arp flows of all switches present in the network are checked. Suppose if any of the above mentioned steps fails then below steps are followed.

If Step 1 fails: If a specific MAC ID of Active host's file is not available in Standard Hosts file that means some new device has intruded into the network. Then we can disconnect

that device from the network by checking the port of the switch from where that ARP flow has come and disconnecting (Port Down) that port from the network.
If Step2 fails: The device with that particular MAC ID (from step 1) is doing Man-in-the-middle attack and we can disconnect that device from the network as said in the above step by checking the port of the switch from where that ARP flow has come and disconnecting (Port Down) that port from the network.
The above procedure is illustrated in the below flowchart.


Main
Start
Run the host_selector program
Calculate the number of switches from the switches file
Select a switch
Run the compare_hosts program
Run the interface_down program
Switches file complete
No
Yes

In the above flow chart the host selector program reads all the available switches in the network and writes them into switches file, accesses the Arp flows of each switch and some file processing is done to sort MAC ID and corresponding IP address of each host connected to that switch and writes them into a file with that switch name.
compare_hosts program compares the MAC ID and IP addresses present in each switch with the standard hosts file and if any mismatch happens it writes that interface to the interface_down file. interface_down file reads the interface from the interface_down file and power downs that interface.

All the above illustrated procedure can be written in a shell file and that shell file is made to run in the controller of the network recursively to continuously monitor the network from the Man-in-the middle attack.

V. Mininet

Mininet is a latest network emulator tool. A collection of hosts, switches, links and routers on a single Linux kernel can be run using Mininet. By using the concept of lightweight virtualization with the same kernel, a single system can be made to look as a complete network. A Mininet host can run programs arbitrarily and behaves like a real machine. Packets can be sent by the programs we run through an interface which seems to be like an Ethernet interface. Packets can be processed by the one which looks like an Ethernet switch, router with some Queuing.
In brief, Mininet’s virtual hosts and interfaces and controllers are the real thing, they are created just by using software rather than hardware but their behavior seems to be similar to distinct hardware elements. It is possible to create a network that resembles same in both hardware platform and in Mininet and to run applications and binary code on either of these platforms.
The evaluation of the network can be done by two common methods, simulation and emulation. The use of simulator is inexpensive, controllable and flexible but the results of the simulator will deviate with the real time results of the model. To avoid this problem, use of emulators came into existence. Emulation uses real devices which can run real time applications. Emulator must use a real time clock whereas the simulator uses a clock which may be slower or faster than the real time clock. Some other major facts that makes the users attract to use Mininet are:

- It is very fast starting and a simple network can be created within seconds.
- It provides the ease of creating custom topologies like a single switch, datacenter etc.
- It provides flexibility to run the real time programs. Anything that runs on Linux can be run using this, from servers to TCP monitoring tools to Wireshark.
- It provides the flexibility for the customized packet forwarding.
- It can be run on any Laptop or PC, on a server, on a Virtual Machine and on a Linux box etc.
- Once the code is packaged, the results can be shared and replicated. Any user with computer can run this code.
- Mininet is easy to use. By writing codes using simple python scripts it is easy to create and run experiments.
- Mininet is an open source project and one can examine the codes and can find errors, debug them and can edit the code using some discussion forums.

VI. Results

The results for Packet delivery ratio, Throughput and End to -end delay are evaluated.

1. End to end delay

The average time taken to reach the destination by a data packet. This includes all the delays due to buffering during the discovery of latency for a route and queuing at the interface queue. This is calculated by the subtraction of the transmission time between the first data packet that has been sent and arrived at the destination. It can be defined mathematically as:

Avg. EED= S/N

Where S is the time taken to deliver the packets to destination and N is the number of packets received by the destination.

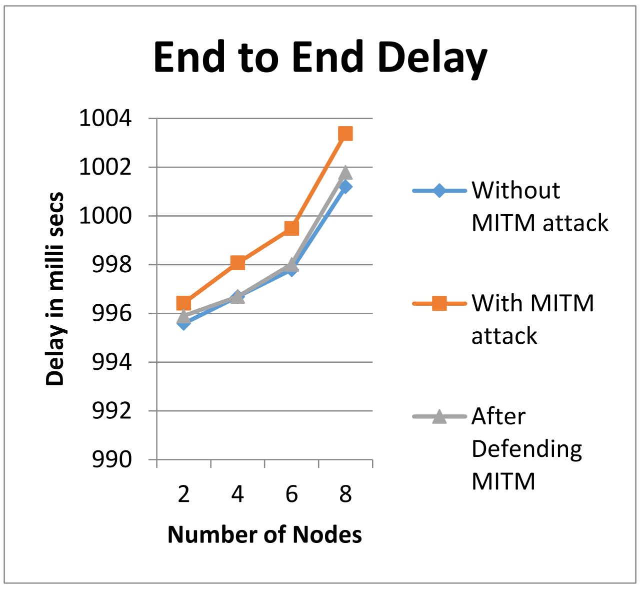


Fig. End to End delay

2. Packet Delivery Ratio

The ratio of the total number of packets received to the total number of packets sent is termed as the packet delivery ratio. Mathematically it can be defined as:

PDR= R/S

Where R is the number of packets received by the receiver
S is the number of packets sent by the sender
The above graph shows the variation of the packet delivery ratio when MITM attack happens and when it is defended using our defending mechanism. The graphs clearly shows that our mechanism improves packet delivery ratio when MITM attack happens in the network.

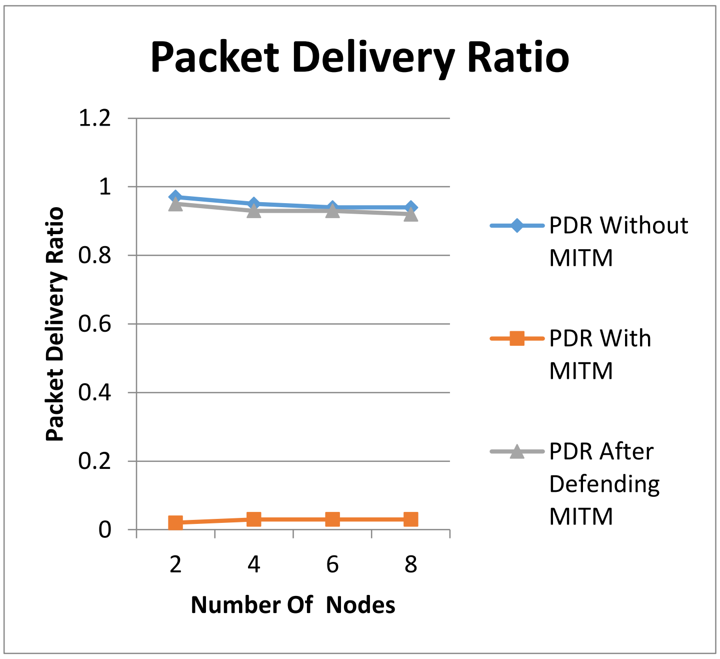


Fig. Packet delivery ratio

References

[1] Are We Ready for SDN? Implementation Challenges for Software-Defined Network, http://ieeeexplore.ieee.org/stamp/stamp.jsp?tp-=&arnumber=6553676
[2] How string is Clipper? Computer Fraud & Security Bulletin, May, 1994
[3] Definition of man-in-the-middle, Webpage2002-03-26, Retrieved 2002-09, http://www.wordspy.com/words/maninthemiddleattack.asp.
Review : Software Defined Networking and Open Flow,https://www.academia.edu/5041216/Review_Software_Defined_Networking_and_OpenFlow
5.white box switcehs,https://www.sdxcentral.com/resources/white-box/what-is-white-box-networking/.“Interface to the Routing System,” IRTF Working Group,https://datatracker.ietf.org/wg/i2rs/charter/
6.“Interface to the Routing System,” IRTF Working Group, available: https://datatracker.ietf.org/wg/irs/charter/.
7.ETSI Industry Spec. Group, “Network Function Virtualisation,” http://portal.etsi.org/portal/server.pt/community/NFV/367.
8.R. Ozdag, “Intel Ethernet Switch FM6000 Series — Software Defined Networking,” http://www.intel.co.uk/content/dam/www/public/us/en/documents/white-papers/ethernet-switch-fm6000-sdn-paper.pdf
9.Software-Defined Networking Using OpenFlow :Protocols,Applicatons and Architectural Design choices,http://swww.mdpi.com/1999-5903/6/2/302/pdf

10.Use Cases for ALTO with Software Defined Networks,http://tools.ietf.org/pdf/draft-xie-alto-sdn-use-cases-01.pdf
11.HyperFlow: A Distributed Control Plane for OpenFlow ,http://www.cse.iitd.ac.in/~siy107537/csl374/a5/files/Tootoonchian.pdf
12.IETF Network WG, "Security Requirements in the Software Defined Networking Model," Internet drafthttp://tools.ietf.org/pdf/draft-hartman-sdnsec-requirements-01.pdf
13.A Security Enforcement Kernel for OpenFlow Networks,http://www.cs.columbia.edu/~lierranli/coms6998-8SDNFall2013/papers/FortNox-HotSDN2012.pdf
14.Path Computation Element(PCE),http://datatracker.ietf.org/wg/pce/charter/
15.Load balancing web server using OpenFlow,http://conferences.sigcomm.org/sigcomm/2009/demos/sigcomm-pd-2009-final26.pdf
16.K.K.Yap,R.Sherwood,M.Kobayashi,T.Y.Huang,M.Chan, N.Handigol,N. McKeown,and G.Parulkar.Blueprint for introducing innovation into wireless mobile networks. In Proc. 2nd ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures, pages 25–32. ACM, 2010.
17.K.K.Yap,M.Kobayashi,R.Sherwood,T.Y.Huang,M.Chan, N.Handigol,and N.McKeown.Openroads:Empowering research in mobile networks. ACM SIGCOMM Computer Commun. Review, 40(1):125–126, 2010.
18.L.E.Li,Z.M.Mao,and J.Rexford.Toward software-defined cellular networks.In Software Defined Networking (EWSDN),2012 European Workshop on,pages 7–12,2012.
19.B.Heller,S.Seetharaman,P.Mahadevan,Y.Yiakoumis,P.Sharma,S.Banerjee,and N.McKeown.Elastictree:Saving energy in data center networks. In Proc. 7th USENIX conf. on Networked systems design and implementation, pages 17–17. USENIX Association, 2010.
20.Inter-datacenter wan with centralized te using sdn and openflow.In Open Networking Summit,April 2012.
21.Sushant Jain,Alok Kumar,Subhasree Mandal,Joon Ong,Leon Poutievski,Arjun Singh,Subbaiah Venkata,Jim Wanderer,Junlan Zhou, Min Zhu, et al. B4: Experience with a globally-deployed software defined wan. In Proc. ACM SIGCOMM 2013 conf.on SIGCOMM, pages 3–14. ACM, 2013.
22.V.Gudla,S.Das,A.Shastri,G.Parulkar,N.McKeown,L.Kazovsky,and S.Yamashita.Experimental demonstration of openflow control of packet and circuit switches.In Optical Fiber Commun. (OFC), collocated National Fiber Optic Engineers Conf., 2010 Conf. on (OFC/NFOEC), pages 1–3, 2010.
23.Netfpga platform. http://netfpga.org
24.Dimitra E.Simeonidou,Reza Nejabati,and Mayur Channegowda.Soft-ware defined optical networks technology and infrastructure: Enabling software-defined optical network operations. In Optical Fiber Commun. Conf./National Fiber Optic Engineers Conf. 2013, page OTh1H.3. Optical Society of America, 2013.
25.A.N. Patel,P.N.Ji,and Ting Wang.Qos-aware optical burst switching in openflow based software-defined optical networks.In Optical Netw.Design and Modeling (ONDM),2013 17th Int. Conf. on, pages 275– 280, 2013.
26.Generalized Multiple Protocol Label Switching,http://www.hit.bme.hu/~jakab/edu/litr/GMPLS/GMPLS_Tutorial.pdf
27. K .L. Calvert, W. K. Edwards, N. Feamster, R.E. Grinter, Y. Deng, and X. Zhou. Instrumenting home networks. ACM SIGCOMM Computer Commun.Review,41(1):84–89, 2011.
28. N. Feamster. Outsourcing home network security. In Proc.2010 ACM SIGCOMM workshop on Home networks, pages 37–42. ACM, 2010.
29.R. Mortier,T. Rodden,T. Lodge, D.McAuley,C. Rotsos, AW Moore, A. Koliousis, and J.Sventek.Control and understanding:Owning your home network.In 2012 4th Int.Conf.on Commun.Syst. and Netw. (COMSNETS),pages 1–10.IEEE, 2012.

Author's Profile

Narendra Neerukonda is currently pursuing his B.Tech Degree in Electronics and communication engineering at K L University. He is getting specialization in Computer Networks. He is a member of IETE. His interests in research include Software Defined Networks.

Naveen Yamparala is currently pursuing his B.Tech Degree in Electronics and communication engineering at K L University. He is getting specialization in Computer Networks. He is a member of IETE. His interests inresearch include Software Defined Networks.

Hari Sairam Kappagantula is currently pursuing his B.Tech Degree in Electronics and communication engineering at K L University. He is getting specialization in Computer Networks. He is a member of IETE. His interests in research include Software Defined Networks.

E. Suresh Babu received his B.Tech degree in Computer Science from RGM College of Engineering, Nandyal, M.Tech degree in Computer Science from V.T.University Belgaum and pursuing PhD in Computer Science & Engineering from J.N.T.University Kakinada. Currently, he is working as an Associate Professor in the Department of CSE in PACE Institute of Technology & Sciences; Ongole He has got 11 years of teaching experience. He has published 6 research papers in various International Journal and 7 research papers in various National and International Conferences. He has attended 20 seminars and workshops. His areas of interests are wireless communication and Mobile Computing