International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 4

A Study and Designing of Custom Network Configuration using Integrated Protocols in SDN Architecture

Farheen Fatma Ansari, Prof. D. P. Mishra, Prof. Sumit Kumar Sar
Department of Computer Science and Engineering, Bhilai Institute of Technology,
Durg, Chhattisgarh, India

ABSTRACT

Today's world emphasizes a lot on speed, performance and ease of use. Taking these demands into consideration it has become mandatory to change all existing systems and transform them into ones that can deliver these characteristics. One such system has been network architecture. Since many years' vendors have been creating and developing their own Network Infrastructure, with different policies and different instruction sets. But communication between networks has taken up importance so as to reduce the timing taken by the data packet to pass through the security settings of another network and vice versa. Hence here is where Interoperability comes into play. If we can take all the networks and have a common controller for all the networks, it would reduce the time and complexity for communicating between them. A simple way to achieve it is through Software Defined Networking, where we can use the software to control the flow of data packets as well as monitor the network. Software Defined Networking is a networking prototype which allows network operators to manage networking elements using software running on an external server. This is accomplished by a split in the architecture between the forwarding element and the control element. The traditional network tends to be rigid. Once the forwarding policy has been defined, the only way to change it is by changing the configuration of all the affected devices.This is time consuming and puts a limit on scalability and meeting challenges of mobility and big data. In this context software defined networking (SDN) is being looked upon as a promising paradigm that has the power to change the way networking is done. By centralizing control, and making forwarding nodes simple, SDN offers flexible control over the traffic flows and the policies networks use to manage these flows. Along with the excitement, there have been apprehensions regarding SDN. In any organization, actual network is not confined to simple configuration. The real network configurations in these organizations are complex and represent more like bulky connectivity. The designing of custom network is done on the GNS3 platform. It is python based and the configurations on GNS3 can be moved to real world networks with minimal changes.

The custom network refers to the configuration of network according to ones need and requirement. That means that number of router, switches and links are not defined, they may change as per the need. The various components of the SDN network is then studied in both the traditional and custom network. The adoption of SDN has still not done in physical network. There have been a number of trial deployments but very few production networks. In this paper, we would discuss the important implementation issues in SDN, the requirements it would fulfill in two fast growing areas wireless networks and clouds and the research that is taking place in these areas. We discuss, in some detail, the enrichments required in the SDN framework as far as security is concerned. Finally, we explore the future directions that SDN research is expected to take. The biggest benefit of Open Flow is to decouple the control plane from the data plane, allowing the centralized forwarding decisions in comparison to traditional distributed control network. As a result, we propose a Software-Defined coding network and address key technical challenges in practice.

Keywords: *Planes, Controllers, Forwarding*

1. INTRODUCTION

Due to blistering evolution of the internet, nowadays there is a tremendous increase in the complication of the network architecture. The network operators are struggling and working hard to manage the challenges of the network with the merging of various types of networks. Networks typically consist of large number of network devices like switch, routers and many middle-boxes.

The traditional network tends to be inflexible. The only way to change the configuration in case of failure is by changing the devices. This is a tedious time consuming task and the traditional network is not able to meet the requirements of the network. As an optimized solution, there comes the Software Defined Network(SDN).

Software-defined networking (SDN) is an ever-growing technology that allows the design and implementation of more flexible and programmable networks. In the past several years, both researchers and vendors have shown tremendous interest on SDN technology.SDN is a disruptive technology that completely redefines the management of computer networks. It is an emerging network technology that supports the separation of control and data plane allowing direct programmability. In SDN, the intelligence of the network is centralized in the software based controllers and the other networking entities acts as normal forwarding devices. Controller has the global view of the network topology and has absolute control over the forwarding elements.

Typical network devices such as switches contain two main components: control plane and data plane. Control plane computes forwarding paths of incoming packets with prebuilt network protocols. Data plane sends those packets to the next hop defined in the forwarding paths. Modification of the network configuration has to be done via command line tools or web console manually. It is difficult to obtain a global view of the whole network. Furthermore, it is inefficient to change the configuration of switches at runtime.

The core and most attractive idea of SDN is to decouple control planes from physical switches. Software-defined control planes such as SDN controllers run on servers and centralize the management of network devices and data flows.

2. PROBLEM IDENTIFICATION

There are various studies done on the concept and implementation of SDN. Many researchers have spent time on its simulation and emulation platform too. Tests are being conducted to study the SDN emulation tool called Mininet. Many topologies are tested with varying number of routers, switches and nodes. But the research work is not done in the area to provide best network configuration with the best suited topology for the implementation of SDN network. Due to stepping in the compacted approach, the reason behind the influence of topology and number of connected switches is yet to be clarified. Network configurations has also effected the processing time of SDN controllers, so it becomes a critical need of the day to find or generate such network that supports the SDN's concept and gives an outstanding performance without delay.

The emulation tool called GNS3 is a network software emulator first released in 2008.It allows combination of virtual and real devices. The main specialty is to simulate complex networks. It uses Dynamips emulation software to simulate Cisco IOS. It is written in Python Language. It is used by many large companies including Exxon, Walmart, At &T & NASA. It is also popular for preparation of Network Professionals certification exams. GNS3 allows to visualize, plan, test and troubleshoot network environments across any vendor platform at scale-without the need to directly interact with the network hardware. But here with the intuitive graphical interface, we can seamlessly connect all types of virtual interfaces to compose a real representation of networks.

With the advancement of GNS3, we are going to study and configure custom network with the help of various types of protocols. This custom network can be changed according to the need of the operator and the network. The custom network is not defined by any standard design or certainty. There has been terrific effect of configuration in the traffic, variance, cost of the network and security concerns. Thus here we provide the configuration of custom network and also give an idea about the better performance of the SDN network by comparison with the traditional network. To create custom network, we have to study about the existing GNS3 or we must have good knowledge of python language.

Here in this paper, the creation of custom network is done on the existing GNS3 tool by dragging and dropping network devices, and then ensuring the best configuration for implementing SDN by comparing this custom network with the traditional network on the bounds like cost, variance, traffic analysis and time taken for creation of network (start/stop) and security concerns. The name custom network defines the charter which enables the creation of any customized network according to the need and requirement.

This paper lays the emphasis on the creation of custom network and the comparative study with traditional network to find the best network platform for the application of SDN. The simulation tool here used for the creation of network is GNS3. This research paper enables the users to implement SDN on the custom network configuration to get the best performance and thus reduce the traffic.

3. METHODOLOGY

The paper speaks of the configuration of Custom Network and then having a comparative study with the traditional network to find the better performance of the SDN network. The main emphasis lay on the designing of the network in the GNS3 tool. There are some methods which need to be considered before configuring the network. So, firstly we should take a look into this things: -

1) Installing GNS3 Tool
2) Dragging & Dropping components
3) Configuring
4) Consoling
5) Running
6) Installing Wireshark
7) Analyzing Packet traffic

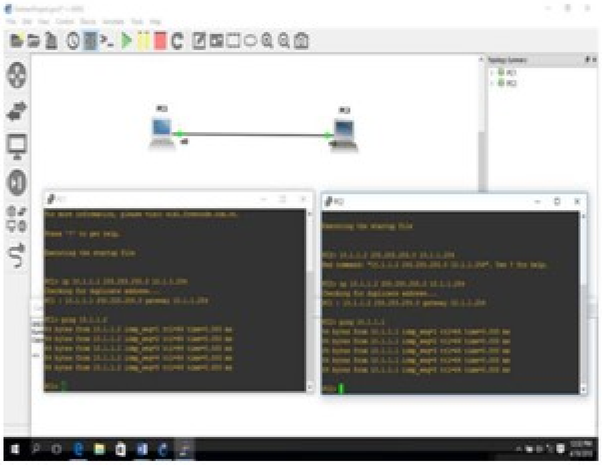
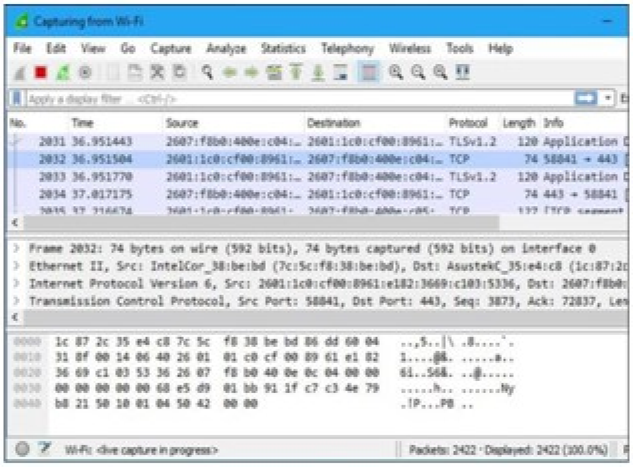


Fig. 3.1 Network Configuration in GNS3 Workspace


Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time        Source                Destination            Protocol  Length Info
 2031 36.951443   2607:f8b0:400e:c04:…  2601:1c0:cf00:8961:…   TLSv1.2     120 Application
 2032 36.951504   2601:1c0:cf00:8961:…  2607:f8b0:400e:c04:…   TCP          74 58841 → 443
 2033 36.951770   2601:1c0:cf00:8961:…  2607:f8b0:400e:c04:…   TLSv1.2     120 Application
 2034 37.017175   2607:f8b0:400e:c04:…  2601:1c0:cf00:8961:…   TCP          74 443 → 58841

> Frame 2032: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
> Ethernet II, Src: IntelCor_38:be:bd (7c:5c:f8:38:be:bd), Dst: AsustekC_35:e4:c8 (1c:87:2c
> Internet Protocol Version 6, Src: 2601:1c0:cf00:8961:e182:3669:c103:5336, Dst: 2607:f8b0:
> Transmission Control Protocol, Src Port: 58841, Dst Port: 443, Seq: 3873, Ack: 72837, Len

0000  1c 87 2c 35 e4 c8 7c 5c  f8 38 be bd 86 dd 60 04   ..,5..|\ .8....`.
0010  31 8f 00 14 06 40 26 01  01 c0 cf 00 89 61 e1 82   1....@&. .....a..
0020  36 69 c1 03 53 36 26 07  f8 b0 40 0e 0c 04 00 00   6i..S6&. ..@.....
0030  00 00 00 00 00 68 e5 d9  01 bb 91 1f c7 c3 4e 79   .....h.. ......Ny
0040  b8 21 50 10 01 04 50 42  00 00                     .!P...PB ..

Wi-Fi: <live capture in progress>        Packets: 2422 · Displayed: 2422 (100.0%)
```

Fig.3.2 Packet capturing in Wireshark

The paper proceeds by the configuration of Custom Network. Then having a comparative study with the traditional network to find the better performance of the SDN network.

The main emphasis lay on the designing and configuration of the network in the GNS3 tool. Moreover, firstly we create a small network and ping IP address to check whether tool is working on a local server or not. If no other option could be choosed as GNS3 VM too.

4. COMPARITIVE STUDY WITH TRADITIONAL NETWORK

In the projected framework, any network can be configured according the need of user and requirement of network that resembles the real network where the implementation has to be done. GNS3, by default, with the intuitive graphical interface users can seamlessly connect all types of virtual interfaces to compose a real representation of networks.

The Customized Network can be configured either by configuring on the existing GNS3 tool or writing command in console or code in python language in the existing GNS3. This work has done in the area of enhancing the existing Mininet for configuration of custom network. Here we propose the creation of custom network in the existing GNS3 so that there shouldn't be the need of upgrading.

All the links can be specified as configuration data and this file is captured as command line file in Wireshark by invoking the GNS3. The work also allows the operator to provide parameter details like delay, bandwidth and packet loss. These values can be specified and even changed according to ones need in custom network. At the time of network configuration, the link details are taken from the configuration file and network is created corresponding to these specified values. After the creation of custom network, the comparative study is done with the traditional network configuration to find the betterment of the SDN. The study is done to find the amount of traffic analysis, capturing in wireshark, processing time and throughput of the custom network.

5. RESULT AND DISCUSSION

The main objective of our project was to create Custom Network. The configuration of any of the networks according to the need of the customers or the users is termed as Custom Network. The GNS3 GUI tool by default supports this type of configuration of the networks. And the creation of Custom Network can be done by little changes in the configuration file or by using python commands. Moreover, by writing certain command on the console we can configure such networks. One advantage of using GNS3 tool was real and virtual part in terms of network components. The custom network supports the change in topology according to the need of user and demand of the network. The number of switches, nodes and controller once deployed could be changed as per the need after implementation. In this type of networks, the architecture does not remain defined and can be altered if one needs to configure more no of elements according to need. Thus the creation of Network that can be changed in variance with the demand is the need of today's network.

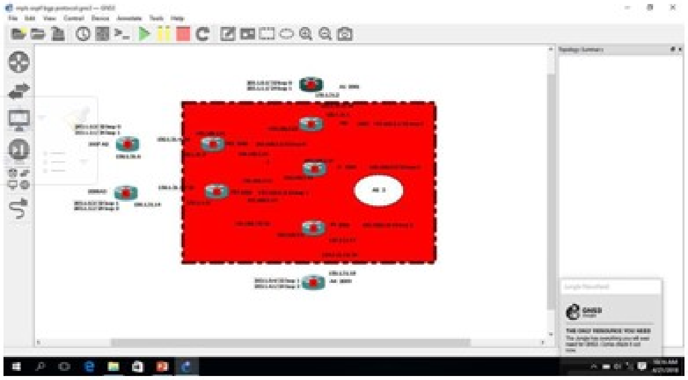


Fig. 5.1 Custom Network

The main focus of discussion is why are we designing Custom Networks? So, the best answer to this is that we are required with such network for Customers with which they can easily fulfill their needs without any worries. Therefore, after creation and configuration of Custom Network, we were required to capture packets from the configured Network. For this purpose, the Display Monitor used here is Wireshark. Hence we can see all the parameters like traffic, delay and time easily in this.

6. CONCLUSION AND FUTURE SCOPE

SDN is presented as one of the effective technology for controlling and managing the complex network by configuring the program in the external server. The SDN supports the separation of data plane and control plane. The data plane includes the components like switches, routers and other middle-boxes. These all devices just act as normal forwarding elements. The decision making capacity of device is being extracted and shifted to a centralized server called controller. The centralized server has the complete view of the topology and thus it monitors and manages the network flow. The emulation tool called GNS3 is used for the simulation of SDN. This tool directly runs on your laptop or desktop. GNS3 helps to configure a network composed of hosts, switches, and a router; being intuitive graphical interface software makes the response time faster and easier to use.

GNS3 is free to download and is open source software that emulates Open Flow devices and Software Defined network controllers. By GNS3 being an emulator it is able to visualize, plan, test and troubleshoot network environments across any vendor platform at scale-without the need to directly interact with the network hardware. With the attractive user friendly interface, users can seamlessly connect all types of virtual interfaces to compose a real representation of networks. GNS3 runs on traditional PC hardware and may be used on multiple operating systems, including Windows, Linux & MacOSx.

This paper summarizes the various aspects of the SDN and creation of custom network. The comparative study shows that the custom network is best for the implementation of SDN network. The result showed that the traffic analyzed in the created network is reduced.

SDN deployment in the physical network provides great platform for research area related to provide security in the controller level. This paper presented that the configuration of custom network is suitable for the application of the SDN. But various algorithms like spanning tree algorithm to solve the loop problem is not discussed here.

The main focus of the research work is done on finding the best configuration for SDN, so that best performance can be obtained after implementing SDN concerned with traffic analysis. This works motivates us to adopt SDN network with supporting custom network configuration in our college campus as well as in hospitals for surveillance purpose.

Thus we can conclude the work by saying that the parameter of SDN gives outstanding performance in the custom network as well as the GNS3 is the user friendly tool for the emulation of SDN network.

7. REFERENCES

1) M. Jarschel, T. Zinner, T. Hoßfeld, P. Tran-Gia, and W. Kellerer, "Interfaces, Attributes, and Use Cases: A Compass for SDN," IEEE Communications Magazine, 2014.

2) N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," SIGCOMM CCR, 2008.

3) A. Tootoonchian and Y. Ganjali, "HyperFlow: a Distributed Control Plane for OpenFlow," in INM/WREN'10, Berkeley, CA, USA, 2010.

4) M. F. Bari, A. R. Roy, S. R. Chowdhury, Q. Zhang, M. F. Zhani, R. Ahmed, and R. Boutaba, "Dynamic Controller Provisioning in Software Defined Networks," in International Conference on Network and Services Management (CNSM), Z¨urich, Switzerland, 2013.

5) B. Heller, R. Sherwood, and N. McKeown, "The Controller Placement Problem," in HotSDN '12, New York, NY, USA, 2012.

6) D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, and P. Tran- Gia, "ParetoOptimal Resilient Controller Placement in SDN-based Core Networks," in 25th International Teletraffic Congress (ITC), 2013.

7) D. Hock, M. Hartmann, S. Gebert, T. Zinner, and P. Tran-Gia, "POCOPLC: Enabling Dynamic Pareto-Optimal Resilient Controller Placement in SDN Networks," INFOCOM, Toronto, Canada, 2014.

8) Stanislav Lange, Steffen Gebert, Thomas Zinner, Phuoc Tran-Gia, David Hocky, Michael Jarschelz, and Marco Hoffmannz,'' Heuristic Approaches to the Controller Placement Problem in Large Scale SDN Networks,'' University of Wurzburg, Institute of Computer Science, Chair of Communication Networks, Wurzburg, Germany,2015.

9) P. Czyazak and A. Jaszkiewicz, "Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization," Journal of Multi-Criteria Decision Analysis, 1998.

10) Karamjeet Kaur, Japinder Singh and Navtej Singh Ghumman, "Mininet as Software Defined Networking Testing Platform", International Conference on Communication, Computing Systems (ICCCS2014).