Simulation of an SDN Network Using fognetsimpp and Performance Comparison of SDN Network with a Conventional Network

*** D. Naveen**
**** K. Khavya**

Abstract

Software Defined Network (SDN) is becoming popular and many companies are trying to switch from traditional network to other sources such as SDN, cloud etc. SDN is the best alternative to traditional network because it can be automated and preprogrammed as per needs. This paper simulates an SDN network with the help of an opensource software. The results show that it performs better than conventional network. The metrics taken for comparison are latency, link failure, packet loss, and the speed of transmission.

Keywords : **Latency, link failure, packet loss, preprogrammed, SDN, speed of transmission, traditional network**

I. INTRODUCTION

This paper deals with simulating the SDN environment with Fognettsimpp tool. The basic network is created with SDN architecture and it is compared with the functionality of the traditional network. The theoretical analysis depicts that SDN performs good than traditional network [1].

II. SOFTWARES / TOOLS REQUIRED TO PERFORM SMULATON

The list of tools required to perform the simulation is as follows [2].

A. Fognetsimpp

Fognetsimpp is a toolkit used to model and simulate distributed fog surroundings. It is being proposed as a new fog simulator called fognetsimpp which provides detailed configuration options for users to simulate a large fog network. This helps researchers to implement custom mobility models and algorithms for the scheduling of fog nodes, and to control the mechanisms for the transfer.

B. OMNeT++

OMNeT++ (Objective Modular Network Testbed in C++) is a modular C++ simulation library and system based on modules, specifically for network simulators. OMNeT++ can be used for non-commercial applications, such as at academic institutions and teaching, for free. OMNEST is an expanded OMNeT++ version for commercial use cases.

OMNeT++ is itself a simulation system for network protocols such as IP or HTTP without templates. Models for simulation of the global computer network are available in many external frameworks. The most frequently used one is INET which offers a variety of models for all kinds of network protocols and technologies such as IPv6, BGP etc.

Manuscript Received: April 5, 2020; Revised: May 2, 2020; Accepted: May 10, 2020.
*D. Naveen is Senior Executive (IT Infrastructure) at K12 Techno Services Pvt. Ltd., Mathikere, Bengaluru, Karnataka - 560 054, India. (email : naveenram2207@gmail.com)
**K. Khavya, Intern, Intel Technology India Pvt. Ltd., Bangalore., 3/31 Sivanandapuram, Saravanampatti post, Saravanampatti, Coimbatore, Tamil Nadu - 641035, India. (email : khavyakannappan97@gmail.com)
DOI : 10.17010/ijcs/2020/v5/i2&3/152554

C. MQTT

Message Queuing Telemetry Transport (MQTT) is a Publish-Subscribe-based Messaging Protocol (ISO / IEC PRF 20922) standard. This operates alongside the TCP / IP protocol series. It is designed for remote location connections where a "small code footprint" is necessary or where the network bandwidth is limited. The messaging pattern for the publish-subscribe includes a message broker.

D. INET

INET Platform is an open source modeling library for the simulation environment OMNeT++. It provides researchers and students working with the communication networks with protocols, agents, and other models. INET is particularly useful to design and validate new protocols or to explore new or exotic scenarios.

INET includes Internet stack models (TCP, UDP, IPv4, IPv6, OSPF, BGP, etc.), wired and wireless connection layer protocols (Ethernet, PPP, IEEE 802.11, etc.), mobility support, MANET protocols, DiffServ, MPLS with LDP and RSVP-TE signaling, various application models, and many other protocols and components.

III. STEPS TO BE FOLLOWED TO CREATE SIMULATION ENVIRONMENT

This experiment is carried out in ubuntu operating system with basic hardware and software requirements [2][3].

Hardware requirements include 4GB RAM, i3 processor, PC/ laptop, mouse and a keyboard.

Software requirements include all the above listed latest version of the software to be downloaded in the machine.

Here are the steps to be followed while simulating the SDN environment.

- **Step 1 :** Download and install omnet++ 4.6.
- **Step 2 :** Download Inet 3.3.0 and import it as a project in omnet++
- **Step 3 :** Create a directory like Inet /inet/applications/ and place the source file in this directory. It is to be hardcoded which has the MQTT functionality.
- **Step 4 :** Then code the simulation functionality as an omnet++ project and place it into omnet++.
- **Step 5 :** Add Inet as reference in Fognetsimpp project.
- **Step 6 :** Configure the environment and make the application to run. Use ". /configure" command to configure the environment.
- **Step 7 :** Then type make command in order to setup variables and path.
- **Step 8 :** Now open omnet++ using the command omnetpp.
- **Step 9 :** Then wait until the fognetsimpp modules are loaded into the library. After it is been installed, run the code.

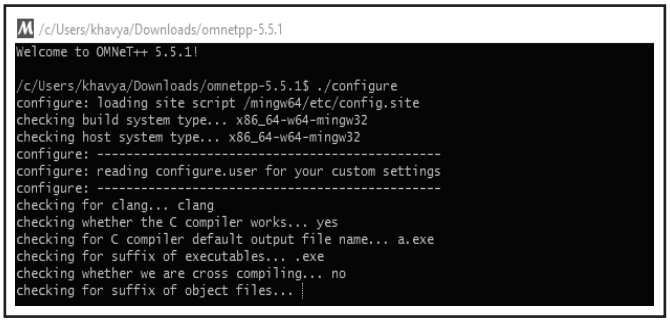

Fig. 1. Configuring the Environment

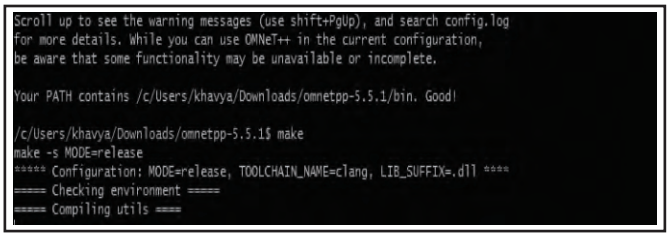

Fig. 2. Using Make Command to Setup the Path for the Environment

Fig. 3. Starting the Omnet IDE

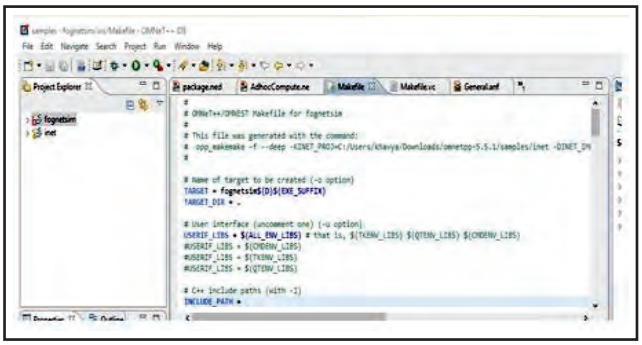

Fig. 4. Code Snippets

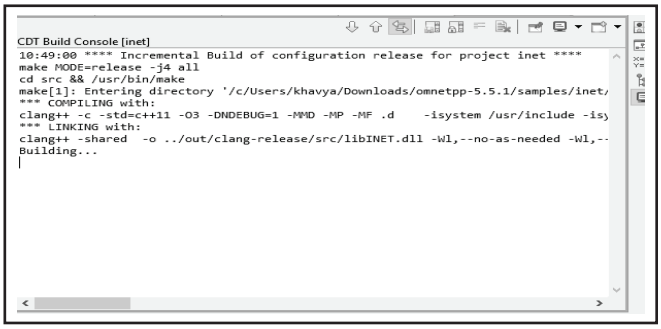

Fig. 5. Code Building Status

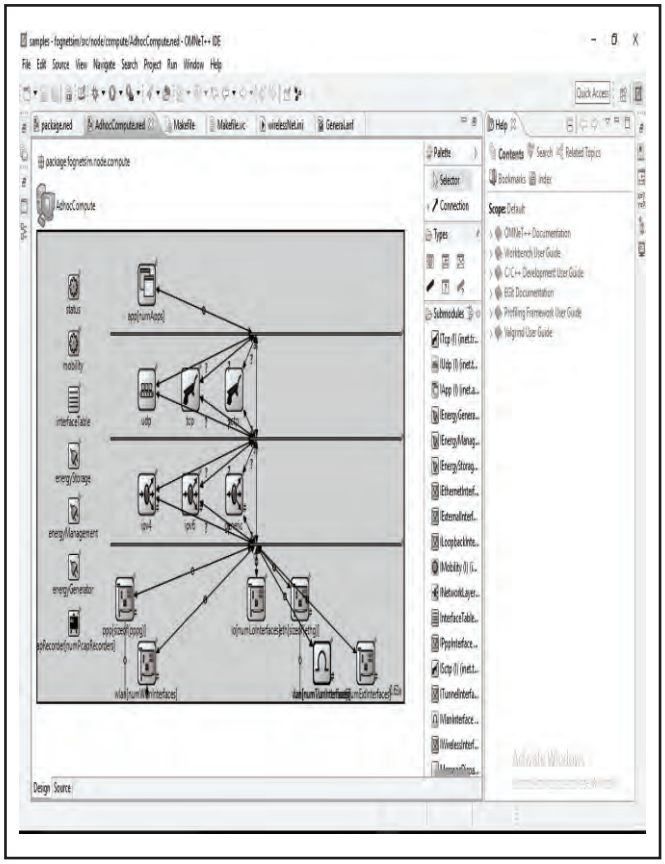

Fig. 6. SDN Network Created Using Simulator

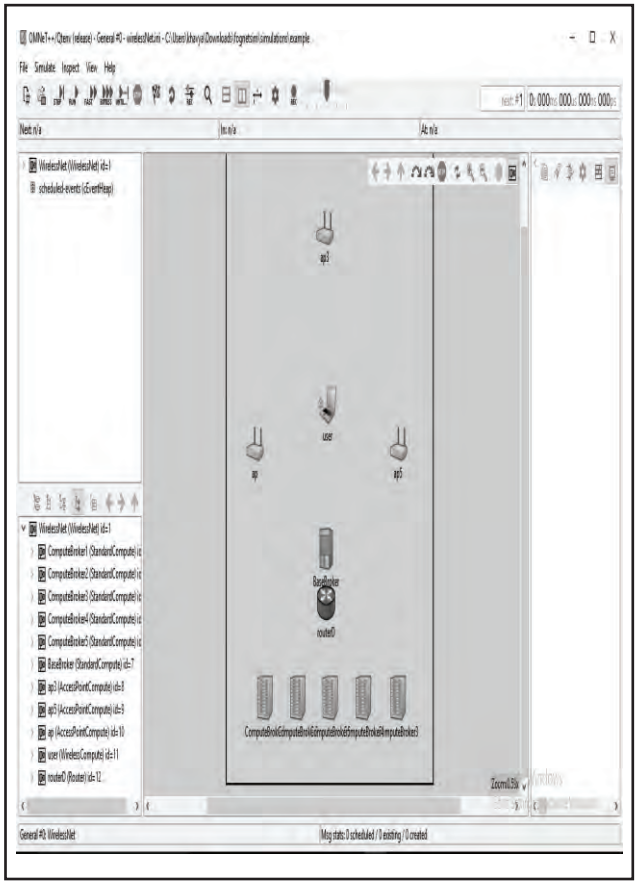

Fig. 7. The Wireless Net Implementation Screen Which has User, Access Points, Router, base Broker and Compute Brokers

IV. COMPARISION BETWEEN SDN NETWORK AND A CONVENTIONAL NETWORK

The following study gives the comparison between the simulated SDN and a conventional network. The conventional network is taken with 10 nodes with LAN connectivity, and the results are discussed. The conventional network uses Open Source Shortest Path (OSPF) algorithm and few hops are disconnected to check its functionality.

Thus, the simulation of an SDN network is done using fognetsimpp software. It includes control plane, management plane, and data plane. Total nodes in the sample simulated network are 10.

V. CONCLUSION

Thus, this work simulates the SDN environment using an

TABLE I.
COMPARISON OF SDN AND CONVENTIONAL NETWORK

FEATURES	SDN	CONVENTIONAL NETWORK
Number of nodes	10	10
Hop count	Used wireless LAN architecture	20
Latency	Nil	In milliseconds
Topology	Mesh topology	Ring topology
Link Failure	No failure. If a node is made to fail, the network opts as per the policy and the packet gets delivered to the destination	The network uses OSPF protocol. If a link is made to fail, then the packet is lost since it reached Time to Live (TTL)
Packet loss	No packet loss since it has no link failure and even it happens algorithm will reroute and delivers the packet within TTL	Packet loss happens since the packet reaches TTL if the link failure happens.
Speed of transmission	The wireless LAN 802.11 b transmits around 5.5 Mbps	The transmission speed100 Mbit/s. It uses fiber optic cable for transmission

opensource software named fognetsimpp along with few other supporting tools and compares the performance of the simulated SDN network with the conventional network and found that SDN works best compared to the conventional networks [4].

REFERENCES

[1] K. Khavya and T. Vairam, "A comparative study on conventional network and software defined network," *Indian J. of Comput. Sci*., vol. 4, no. 6, 2019.

[2] Github. [Online]. Available: https://github.com/rtqayyum/fognetsimpp/blob/master/README.md

[3] T. Qayyum, A. W. Malik, M. A. K. Khattak, O. Khalid, and S. U. Khan, "FogNetSim++: A toolkit for modelling and simulation of distributed fog environment," IEEE Access, 6, 63570-63583, 2018.

[4] D. Gopi, S., Cheng, and R. C. Huck, "Comparative analysis of SDN and conventional networks using routing protocols," In *2017 Int. Conf. on Comput., Inform. and Telecommunication Syst.* (CITS) (pp. 108-112). IEEE, July, 2017. Doi: 10.1109/CITS.2017.8035305

About the Authors

D. Naveen is currently working as a Senior Executive IT Infrastructure at K12 Techno Services Pvt. Ltd. He is passionate towards cloud computing, software defined networking and networking.

K. Khavya completed M. Tech (Information Technology) in 2020. She is currently an intern at Intel Technology India Pvt. Ltd., Bangalore. She completed B.E. (Computer Science and Engineering) in 2018. Her areas of interest include cloud computing and networking.