

An Approach to Effective Bandwidth Utilization using Software Define Networking

Deepika M S¹, K N Rama Mohan babu²

*Dept. of I.S.E, Dyananda Sagar college of Engineering^{1,2}
Kumarswamy layout, Banglore, India*

Abstract— Bandwidth utilization is an important factor to improve the network performance. For the effective utilization of bandwidth, pathloss rate and latency should be measured. So that the application can adapt its send rate to the estimated bandwidth. To measure the path loss rate and latency between two nodes in a college campus network active measurement techniques are used. These techniques do not require prior knowledge about the network topology. The drawback of these techniques is that every device in the network must be configured to estimate the bandwidth measures like path loss rate and latency. In this paper, we propose an SDN framework for estimating the pathloss rate and latency. This framework includes features such as detecting path loss rate and measuring the latency in a college campus network. By deploying this method the loss rate and latency can be calculated using the initial packet separation and the time stamps as input. In SDN, for a single change in networks, the network configurations are changed centrally on specific controller(s) rather than configuring individual network devices. As a result, we can improve bandwidth utilization and network performance in a college campus network. Comparison of bandwidth measurements is done using iperf between traditional network and SDN, results obtained from simulation by using mininet shows better performance in SDN than traditional network.

Keywords— Software Define Networking(SDN), Bandwidth utilization, Probing methods.

I. INTRODUCTION

In traditional network architectures such as college campus networks use network devices that combine both control plane and data plane functions in a single device such as router or switch. The control plane is an element of a router or switch that determines how an individual device within a network interacts with its neighbours. Routing protocols, such as Open Shortest Path First (OSPF), Border Gateway Proto-col (BGP), and Spanning Tree Protocol (STP) are the examples of control plane protocols. These protocols determine how packets are forwarded to optimal port or interface in the data plane. While these control plane protocols provide scalability, network resiliency, they pose limitations. For example, based on static metrics such as interface bandwidth or hop count the routing protocols may only be able to determine the best path through a network. Similarly, control plane protocols typically do not have any visibility into the applications running on the network, or how the network affects application performance.

Data plane functionality contains features such as quality of service (QoS), bandwidth measures, encryption, Network Address Translation (NAT), and access control lists (ACLs). These features directly affect how a packet is forwarded or

being dropped. All though, many of these features are static in nature and are determined by the fixed configuration of the network device. But still there is no mechanism to modify the configuration of these features dynamically based on the changing conditions of the network or its applications. So the configuration of these features is done on each and every device in a network, thus limiting the scalability of applying the required functionality to the network. To overcome these limitations and to ensure programmability in network, Software define networking (SDN) came into existence. In SDN the control plane is decoupled from data plane, there by simplifying the network management and enabling innovation through network programmability. Hence SDN have been growing in popularity. Figure 1 shows the SDN architecture.

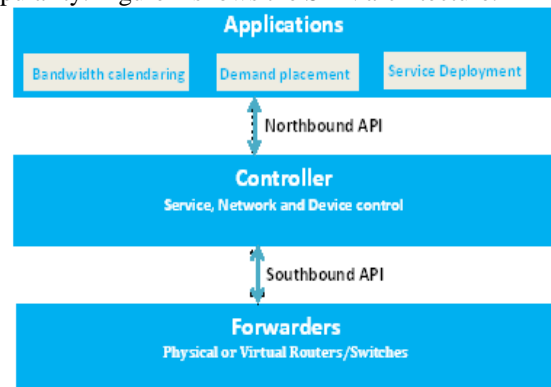


Figure 1: SDN architecture [11]

In a college campus network, since many users must share the common bandwidth capacity on the network, there will be a situation in the network where the demand is beyond the capacity. This causes network congestion and has negative effect on the data transmission rate, bandwidth utilization and quality. The most important part of the network is to plan forwarding of data with equal priority, independent on the source and destination. Hence, there is no trivial technique to assure a certain transmission rate.

In this paper we propose an SDN framework to measure the bandwidth, calculate pathloss rate and latency. The features such as type of service forwarding, measuring pathloss rate and latency is placed in the controller, so that the provider can program the network, according to the user needs and modify the configuration of these features based on the dynamic conditions of the network or its applications. The measurement data used for detecting pathloss rate and latency are divided into probe packets, these probe packets are injected into the network and are time stamped on the receiver side. In this paper we discuss the concept of having centralized routing control plane separated from forwarding

elements for more flexible and effective bandwidth utilization which in turn improves the network performance.

II. LITERATURE SURVEY

The two common types of techniques to measure available bandwidth and latency are passive and active techniques [5]. C. Fraleigh et.al explains how enormous applications can benefit from bandwidth estimation given by either passive or active measurement techniques [2].

A. Passive measurements

Constantinos Dovrolis et.al explains passive measurement techniques may be an attainable strategy to measure network characteristics, like the available bandwidth and latency [3]. Passive measurement techniques and tools act as observers within a network and generally they will not interfere with any other traffic [4]. These techniques most frequently need control and administrative privileges of the underlying network infrastructure for accessing routers and servers within the network.

Eg: MRTG, IPMON.

B. Active measurements

Instead of using above techniques, we can implement active measurement techniques. In active measurement technique probe traffic is injected into the network at a traffic source and the network's impact is measured at a probe traffic receiver [2]. Therefore, active measurement techniques have an effect on the network traffic itself, conflicting to the passive measurement technique. The active measurement access two hosts, one is the traffic source and other is a traffic receiver. Such techniques are referred to as end-to-end measurement techniques [5]. Probing is the basic component of all active measurement techniques, including measurements to acquire information about link capacity and bandwidth measures of a network. There are many probing schemes. Out of that, two basic probing schemes are explained: the packet-pair and the packet-train probing schemes.

C. Probing schemes for Bandwidth measurements

Manish Jain et.al explains, how to estimate end-to-end available bandwidth and pathloss rate or latency in active measurement technique[2]. The primary step is to probe the network path. This can be done by injecting a set of probe packets with a pre-defined separation or distribution. The distribution is inversely proportional to the probe rate(measured in bits per second). Hence, the smaller is the distribution between probe packets the higher is the probe rate.

The most fundamental probing scheme is to divide the probe packets in pairs, where each pair has a pre-defined distribution that corresponds to the probe rate [3]. The each pair of probe packets is sent through the network to the receiver and the packets are time stamped usually at the application layer. The arrival timestamps are used to calculate the bandwidth measures. Instead of using pairs of probe packets many techniques use trains of probe packets [7]. The distribution between the probe packets inside the train can be equal or exponentially decreasing. There is a

basic difference between using packet-pair and packet-train probing schemes. In packet-train probing schemes numerous probe packet delays might depend on each other, which is not in the case of packet-pair probing schemes[6].

The arrival rates of probe packets can be used as an input to the analysis to estimate the available bandwidth. One of the inherent problems in bandwidth measurement is that the bandwidth measures vary over time. Estimation of band-width measure has been an emphasis over some time and the area is still emerging. However, there are still several problems existing that are unsolved or needs to be addressed more carefully. The existing techniques do not centrally manage the routing of probe packets. Every device in network must be configured to estimate the bandwidth measure. This paper focus on estimating the bandwidth measure such as pathloss rate and latency in the proposed SDN framework.

III. THE SDN FRAMEWORK

This section gives a brief explanation about proposed SDN framework for measuring the pathloss rate and Latency and to improve bandwidth utilization.

D. Architecture Overview

The proposed SDN framework for measuring bandwidth, pathloss rate and latency are shown in figure 2. The routing service module in SDN framework is an intelligent application based on OpenFlow architecture. This data store centric design helps in the implementation of enhanced routing services, such as calculating the shortest path based on a consistent view of network state, calculating pathloss rate and latency. This implementation reduces complexity and controls interactions of a probe packet. The Routing Service application contains three modules: Link Discovery, Topology Manager and Virtual Routing Engine.

1) *Link Discovery module*: This is responsible for detecting and maintaining the state of the physical links in the network.

2) *Topology Manager Module*: This module uses information from the link discovery module to build and maintain the network topology information in the controller, and calculates the routes in the network using the neighbor database. It creates the logical view of the network, determining the shortest path between two nodes or host. It also stores information in Database at the controller, which contains the shortest path information to any node or host.

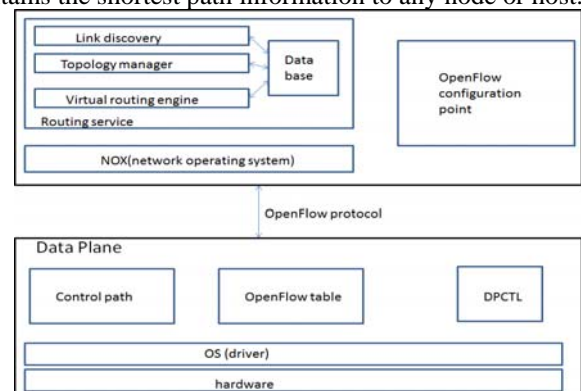


Figure 2: SDN framework for measuring bandwidth,pathloss rate and latency

3) *Virtual Routing Engine*: It creates a virtual networking topology which consist virtual machines that run traditional routing protocol. It also updates the Database to store shortest paths from one node to the next hop.

4) *OpenFlow configuration point*: It is used to centrally configure all OpenFlow switches in a data plane. The pox network OS controls and co-ordinates the operation of the controller. The DPCTL command ensures interaction between OpenFlow switch and the controller.

E. The Route Calculation process

Figure 3 explains the route calculation process in the proposed SDN framework. The Topology Manager module computes the shortest path between the nodes in a given network.

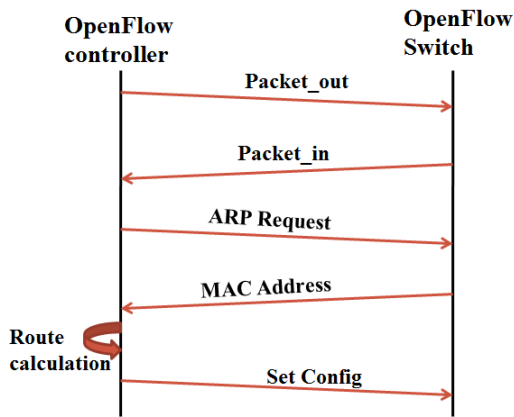


Figure 3: Route calculation process

This module calculates the destination-rooted trees to identify the shortest path between any two nodes and using the information from the database it develops routes based on the given network topology. The controller calculates the end to end path from the destination rooted trees when it knows the location of both sender and receiver on the network. It also configures the same information at all switches along the path. The Topology Manager module keeps track of the MAC addresses and the switch ports of the network. Once the MAC address is known a point-to-point route is calculated and a flow entry is added to each switch along the shortest path through DPCTL command.

F. Calculation of available bandwidth

To calculate available bandwidth iperf command is used. Iperf is a simple and very powerful network tool used for measuring TCP and UDP bandwidth performance. By modifying various parameters and characteristics of the TCP protocol, the network’s bandwidth availability, latency, jitter and pathloss rate are found.

IV. ALGORITHMS AND EQUATIONS

This section describes the algorithms and equations used in calculation of pathloss rate and latency.

A. Measuring pathloss rate:

Path loss rate plays a very important role in wired network. Path loss is the calculation of unwanted introduction of energy which interferes with the proper reception and

reproduction of the signals during its journey from source to destination.

The equation used to calculate pathloss rate is as follows.

$$\text{PathLoss rate} = (I) \frac{\text{Input}_{pkts} - \text{output}_{pkts}}{\text{Input}_{pkts}} * 100 \text{ -----eq 1}$$

Where,

Assumption: $\text{Input}_{pkts} > \text{output}_{pkts}$

Input_{pkts} – are the packets received at the destination.

output_{pkts} – are the packet sent out of the host.

I is interval between probe packets.

The following algorithm explains how the pathloss rate is calculated in the controller. This algorithm uses equation 1 to calculate the pathloss rate.

Algorithm: calculation of the pathloss rate

Step 1: controller sends the probe packet to all switches in the network and creates the time stamp of the network.

Step 2: controller sends the timer request to all switches and collect the flow statistics from switches and stores in the database.

Step 3: controller keeps track of the counter to count the number of probe packets sent out of the host and stores it in the database as output_{pkts} .

Step 4: controller keeps track of the counter to count the number of probe packets received at the destination and stores it in the database as Input_{pkts} .

Step 5: controller retrieves the stored information from data-base and calculate the pathloss rate using equation 1.

B. Measuring latency:

Latency is defined as the time taken for a message to travel from source to the destination in the network.

The equation to calculate latency is as shown below

$$\text{Latency} = (I) * (Tq - Tp) \text{ ----- eq 2}$$

Where,

Tp - is the time when sending out $\text{port_stats_request}$ packet.

Tq - is the time when receiving $\text{port_stats_received}$ packet

I – is the interval between probe packets.

The following algorithm describes how to measure latency using probe packets. The algorithm below uses equation 2 to measure latency.

Algorithm: measuring latency

Step 1: controller sends the probe packets from one switch to another and switch will forward back the probe packet to the controller.

Step 2: When the controller knows src_dpid and dest_dpid send the probe packets every two seconds.

Step 3: controller Measures $T1, T2, T3$ using

If $\text{dpid} = \text{src_dpid}$

$$T1 = 0.5 * (\text{received_time} - \text{sent_time})$$

Else if $\text{dpid} = \text{dest_dpid}$

$$T3 = 0.5 * (\text{received_time} - \text{sent_time})$$

Else if $\text{dpid} = \text{dst_dpid} \&\& \text{pkt_type} = 0x5577$

$$T2 = \text{received_time} - T1 - T3$$

Step 4: controller calculates Latency using equation 2

V. SIMULATION

Mininet network simulator is used to simulate the proposed method. The table 1 shown below represents the bandwidth performance is increased in software define network than traditional network.

Use Cases	Traditional Network		Software Define Network	
	Switch count	Bandwidth measured	Switch count	Bandwidth measured
1	1	108 Mbits/sec	1	1.07 Gbits/sec
2	10	15.2 Mbits/sec	10	457 Mbits/sec
3	20	7.80 Mbits/sec	20	287 Mbits/sec
4	40	3.66 Mbits/sec	40	153 Mbits/sec
5	60	2.46 Mbits/sec	60	105 Mbits/sec
6	80	1.60 Mbits/sec	80	81.2 Mbits/sec
7	100	1.38 Mbits/sec	100	64.7 Mbits/sec

TABLE 1: COMPARISON OF BANDWIDTH PERFORMANCE BETWEEN TRADITIONAL NETWORK AND SDN.

The simulation is done for 7 use cases. Bandwidth performance is measured by increasing the number of switches and hosts. As the number of switches increases the bandwidth performance decreases

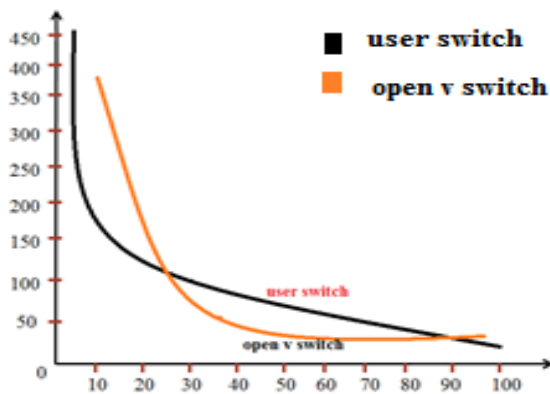


Figure 6: comparison of bandwidth performance in SDN and traditional network

Figure 6 illustrates how the bandwidth performance is increased in software define network than in traditional network. The graph is plotted using data in table 1. The black line is the bandwidth measurement of user switch which is referred to as traditional switch. The orange line is the bandwidth measurement of open v switch. Results obtained show that bandwidth utilization and network performance is more in SDN than in traditional network.

VI. CONCLUSIONS AND FUTURE WORK

Since the network parameters dynamically change SDN framework is very appealing method for calculating path-loss rate and latency. In SDN, for a single change in network, the network configurations are changed centrally on specific controller(s) rather than configuring individual network devices. As a result, by measuring the pathloss rate and latency we can improve bandwidth utilization and network performance in a college campus network. Comparison of bandwidth performance is done using iperf between tradi-tional network and SDN, results obtained from simulation by using mininet shows better performance in SDN than traditional network. By detecting the path loss

rate and latency the bandwidth can be utilized effectively. When compared to traditional network, in SDN there is 20% reduction in latency, and 30% increase in bandwidth utilization. And therefore overall increase in network performance.

The drawback in proposed SDN framework is as the number of switches increase the performance decreases. The future work is to implement the shared multiple table lookup in OpenFlow.1.1, using Smart caching and hybrid software-hardware flow state in switch to overcome the drawback.

REFERENCES

- [1] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot. Packet-level traffic measurement from the sprint ip backbone. In IEEE Network Magazine, November 2003.
- [2] ConstantinosDovrolis, ParameswaranRamanathan, and David Moore. Packet distribution techniques and ca-pacity estimation. In IEEE/ACM Transactions on Networking, December 2004.
- [3] ConstantinosDovrolis, ParameswaranRamanathan, and David Moore. What do packet distribution techniques measure? In Proceedings of IEEE INFOCOM, pages 905–914, Anchorage, AK, USA, April 2001.
- [4] Manish Jain and ConstantinosDovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. In Pro-ceedings of ACM SIGCOMM, Pittsburg, PA, USA, August 2002.
- [5] Bob Melander, Mats Björkman, and Per Gunningberg. Regression-based available bandwidth measurements. In Proceedings of the 2002 International Symposium on Performance Evaluation of Computer and tele-communications Systems, San Diego, CA, USA, July 2002.
- [6] Vinay Ribeiro, Rudolf Riedi, Richard Baraniuk, Jiri Navratil, and Les Cottrell. pathchirp: Efficient available bandwidth estimation for network paths. In Passive and Active Measurement Workshop, 2003.
- [7] Andreas Johnsson, Bob Melander, and Mats Björkman. Modeling of packet interactions in distribution-based network probing schemes. Technical report, Mälardalen University, 2004.
- [8] C. E. Rothenberg, M. R. Nascimento, M. R. Salvador, C. N. A. Correa, S. Cunha de Lucena, and R. Raszuk. “Revisiting routing control platforms with the eyes and muscles of software-defined networking”. In Proc. of ACM SIGCOMM workshop on HotSDN, 2012, pp.13-18.
- [9] VasileiosKotronis, XenofontasDimitropoulos. “Out-sourcing the Routing Control Logic: Better Internet Routing Based on SDN Principles”. Hotnets ’12, October 29–30, 2012, Seattle, WA, USA. pp. 1-6.
- [10] M. Ribeiro Nascimento, C. Esteve Rothenberg, M. R. Salvador, Carlos Corrêa, Sidney Lucena and M. F. Magalhães. “Virtual Routers as a Service: The RouteFlow Approach Leveraging Software-Defined Networks”. In 6th International Conference on Future Internet Technologies 2011 (CFI 11), Seoul, Korea. pp.1-4.
- [11] Saro Velrajan. white paper on “Application-aware routing in SDN”. Aricent Technology. aricent.com © 2013. pp. 1-6.
- [12] O. M. E. Committee. White paper on “Software-defined Networking: The New Norm for Networks”. Open Networking Foundation, 2012. pp. 1-17.
- [13] Nick McKeown, et al., “OpenFlow: enabling innovation in campus networks”, ACM SIGCOMM CCR, vol-ume 38 n.2, April 2009. pg17-pg29.
- [14] A Survey of Software-Defined Networking: Past, Pre-sent, and Future of Programmable Networks. Marc Mendonca, Bruno Astuto A. Nunes, Xuan-Nam Ngu-yen, Katia Obraczka, and Thierry Turletti, hal-00825087, version 2 - 17 Jun 2013.
- [15] S. Ghorbani and M. Caesar. Walk the line: consistent network updates with bandwidth guarantees. In Pro-ceedings of the first workshop on Hot topics in soft-ware defined networks, HotSDN ’12, pages 67–72, New York, NY, USA, 2012. ACM.