UPLB Network Queue Simulator (UNQS): Analyzing Network Performance For Internet Bandwidth Management

Leensey M. Lawas and Danilo J. Mercado

***Abstract*— Internet bandwidth is an expensive resource that is increasing in demand. As an alternative to meet those demands, UNQS was developed to simulate real traffic data and to help identify the optimal bandwidth setting, which can then minimize cost. Traffic data was collected using *ntopng* and had a bandwidth of 3,448.96 Kbps, a total of 554,392 flows and 432,616,056 packets with a total size of 273,715,167,385 bytes over a span of 8 days. The data was processed using the FIFO scheduling algorithm with different bandwidth constraints and a TTL of 60 seconds. Results showed that increasing the bandwidth reduced the amount of unserviced flows until eventually, all flows are serviced and that a minimum recommended bandwidth can be determined by observing the shift in the trend of the results. In conclusion, UNQS can be used to find the optimal bandwidth setting for a network given existing traffic data.**

***Index Terms*— bandwidth management, network performance, queueing algorithm, scheduling algorithm, traffic engineering**

I. Introduction

With the prevalence of internet usage in this digital age, the rise of demand for fast and reliable execution of online services is inevitable. Whether it is for personal use, like video chatting with friends and family from abroad, or for commercial and business transactions, customers want to make sure their services are done efficiently without slowing down or timing out. Client requests are sent simultaneously that when the server responses (or traffic) are returned, the routers are unable to inspect long fields in Internet Protocol (IP) packet headers quickly and are unable to reassemble and segment packets fast enough, causing performance bottleneck [1]. A quick solution to the problem would be increasing bandwidth size, because as the demand for services increases, the bandwidth must also be increased [2]. However, this method is costly and inefficient, which is why traffic engineering (TE) takes place.

Awduche, Chiu, Elwalid, Widjaja, and Xiao of The Internet Society (2002) [3] defined that internet traffic engineering deals with evaluating network performance and optimizing it. Bandwidth is the unit of measurement, usually in Kbps or Mbps, used to monitor network performance for quantifying how much information a communication channel can handle [4].

Presented to the Faculty of the Institute of Computer Science, University of the Philippines Los Baños in partial fulfillment of the requirements for the Degree of Bachelor of Science in Computer Science

A. Background of the study

Tong and Yang (2007) [5] cited that there have been many studies on TE, but most of them dealt with route selection algorithms, and few tackled bandwidth management techniques. For this study, bandwidth management is the TE method chosen. Kanu, Kuyoro, Ogunlere, & Adegbenjo [6] define bandwidth management as an optimization technique that helps differentiate the types of network traffic from each other and determine which client or service should be prioritized. In short, bandwidth management allocates the available bandwidth depending on network traffic and client/service priority.

In an article named *Bandwidth management pays off* (2002) [2], two key devices were identified to help in bandwidth management: traffic shaping or congestion avoidance mechanisms and queueing techniques. *Congestion avoidance mechanisms* or *congestion control* locates where in the router the packets do not enter the system, and finds an alternative route so the packets do not block the way and cause timeout [7]. *Queueing techniques*, on the other hand, help predict and direct the traffic flow by implementing a constraint or constraints to provide the services as demanded [8]). In addition, queueing network models are known for accuracy and efficiency [9]. By implementing these techniques, issues with cost and congestion will be dealt with while providing Quality of Service (QoS), thus increasing performance and reducing delay [10].

B. Significance of the study

Inefficient internet bandwidth management can lead to dissatisfied customers and reduced productivity. As long there is a need for âĂIJhigh quality of corporate customer satisfactionâĂİ, bandwidth management will continue to grow as a body of knowledge [11]. This study simulated actual traffic data within the University of the Philippines Los Banos (UPLB) Network in order to determine whether the existing bandwidth is optimal or not. Additionally, the results can also be used for planning that can entail significant cost reductions, thus optimizing both bandwidth and budget to provide best quality service.

C. Objectives of the study

The general objective of the study is to efficiently simulate the UPLB network traffic by identifying the most optimal bandwidth setting. Specifically, the study was able to:

1) Collect traffic data from the UPLB network;
2) Simulate the traffic data using various bandwidth constraints;
3) Observe the result of the simulation by noting the duration, throughput, and flow loss for each constraint; and
4) Determine the most optimal network setting.

D. Time and Place of the study

The study was conducted from January 2017 until June 2018, at the Institute of Computer Science, UPLB.

E. Scope and Limitation of the study

The study is limited to monitoring and simulating a portion of the UPLB network. Also, it focused on the traditional queueing technique known as First In-First Out Queueing (FIFO).

In determining the most optimal bandwidth, the throughput, latency, and flow loss values was measured, noted, and compared. Throughput is the number of tasks accomplished over a period of time, latency is the time it takes for a fixed task to be finished, and flow loss is the percentage of dropped flows over the total number of flows.

II. Review of Related Literature

Several studies have been conducted which attempted to manage internet bandwidth as efficiently as possible. With a goal to provide speedy transaction of certain services such as e-mailing, video streaming, downloading, and many more, internet bandwidth management plays an important role not just for business and commerce applications, but as well as personal usage, for customer satisfaction and improved network performance. Because of the many details enumerated, the demand for internet bandwidth management has never been greater.

Internet Live Stats (2016) [12] records that internet users from 2015 to 2016 had increased with an estimated 43.4% of the world population to 46.1%. It has been a dramatic increase since 1995, with users all over the world amounting to less than 1%. The implications of this statistics to the UPLB academe can also be applicable, as more students and workers enter the university to make use of the campus network. The increase entails a growth in demand for larger internet bandwidth. With a number of users sending multiple requests for different services with varying sizes, data traffic becomes congested. No end-user wants delays, slow downs, or timeouts, in accomplishing the services they requested. Instead, end-users want fast execution of their requests so they can proceed to doing other tasks.

To fix the problem, bandwidth management takes place. Instead of paying for an increase in bandwidth for a temporary fix, as mentioned in an article [2], bandwidth management aims to properly allocate the already existing bandwidth size as effectively and as efficiently as possible.

Before packets are received by the destination address, they first arrive in packet switches. These packet switches are in charge of queueing the packets and forwarding them eventually to the destination [13]. Thus enters the scheduling algorithms used to identify which packets must be distrbuted first to the computers in the network.

An important aspect in network scheduling is the data that will be subjected to it. Jerkins and Wang (1999) [14] emphasized the importance of characterizing these data in order to extend its applications to traffic management, and more specifically, for the purpose of this paper, bandwidth management.

Traffic classification is an initial stage that plays a key role in some scheduling algorithms, such as Priority Queueing (PQ). Because of bandwidth contraints, traffic classification helps in managing the fixed, limited, and available bandwidth. Classification can be payload-based, meaning a field of the payload is examined and used for classification [15, Chapter 5]. An early traffic classification technique [16] made use of port numbers, which worked best for well-known or reserved ports. The other method for classifcation uses statistical analysis of traffic behavior [15, Chapter 5].

Karim, Nasser, Taleb, and Alqallaf (2012) proposed a priority packet scheduling algorithm that made use of three priority queues that gave importance to real-time traffic (priority 1) over non-real time traffic (priorities 2 and 3). Its result suggested of a better performance opposed to FCFS and multi-level queue scheduler algorithms [17].

From the study of Muhilan, Arulselvi and Kiran Kumar (2013), packet scheduling algorithms using fair queueing and two additional variants were simulated to compare the delay. It showed that WFQ and Self Clock Fair Queueing (SCFQ) experienced a linear delay, whereas the Worst Case Weighted Fair Queueing (WF2Q) share the output link [18].

The aforementioned studies inspired and influenced this study, which used traffic data as input for the simulations of FIFO.

III. Methodology

A. Traffic Collection

A mirror port was setup and connected to a 64-bit Ubuntu server. A community version of the traffic monitoring application, `ntopng`, was installed to the server. `MySQL` database management system was also installed. It shall contain the database where traffic flow data from `ntopng` will be dumped. After collecting the data, `mysqldump` was used to make a copy of the database to be used for the simulation.

`ntopng` uses the Unix timestamp to label the arrival time of packets into the switch (known as `FIRST_SWITCHED`) and their exit time from switch to their destination hosts (`LAST_SWITCHED`). This timestamp counts the seconds that have passed since January 1, 1970 (Coordinated Universal Time/UTC).

B. Simulation

A class diagram was constructed to get an overview of how UNQS was implemented in the Java programming language. The following items give a brief description of each class/interface.

1) **Cofiguration.java** is the class responsible for setting, updating, validating, and displaying the configuration for the database connection and simulation settings.
2) **Flow.java** is a class that represents a flow, which is a network link between a source host and a destination host. For the purpose of this program, the IP addresses of both hosts are not defined as this class' attribute.
3) **NetworkBuffer.java** is a class that defines the queue or cache that will contain all arriving flows that need to be switched.
4) **Schedule.java** is an interface that has final-static-defined variables FIFO (0), PQ (1), and WFQ (2). All of its methods are abstract and therefore must be overriden by the classes that will implement it.
5) **FirstInFirstOut.java** implements the Schedule interface and overrides its methods in order to perform FIFO's logic.
6) **UNQS.java** contains the main function where the connection to the database is established. Time is counted from the configuration's defined range of *start_time* to *end_time* and shall continue to iterate until all flows within the defined network buffer(s) are scheduled from the switch to their destination.

The implementation is summarized by the figure in section III-B, pg. 4.

IV. Results and Discussion

A. Traffic Description

Network traffic was collected from November 13-20. There were a total of 554,392 flows, 432,616,056 packets, and 2,297,393,747,224 bits observed in a period of 7.71 days (computed as the difference between the maximum `FIRST_SWITCHED` minus the minimum `FIRST_SWITCHED`, both written in Unix timestamp). The overall bandwidth of the observed traffic is tabulated below.

The top destination IP addresses for out-flow traffic are shown in Table I and the top source IP addresses for in-flow traffic are seen in Table II.

Listed are brief descriptions for each company that is associated with the identified IP addresses.

1) **KDDI Corporation** is a telecommunications business based in Japan. It provides content hosting over optimized networks, and ICT and business services and solutions.
2) **Google LLC** is an American multinational technology company that specializes in Internet-related services and products such as online advertising technologies, search engine, cloud computing, software, and hardware.
3) **WorldStream B.V.** is a popular Internet Service Provider based in the Netherlands and is used by customers from all over the world. It provides cost-effective services to secure hosting environment, and offers hardware and Operating System technologies.
4) **M247 Ltd** is a UK-based technology company that offers services and tools to secure network and data while providing connectivity and internet infastructure that expands to a global scale.
5) **Converge ICT** is a Philippine technology company with the fastest growing fiber internet and offers services to ensure pure end-to-end fiber internet connection, thus reducing data loss, faster speed and bandwidth.
6) **Facebook** is an American online social media and social networking service company.
7) **Apple Inc.** is an American multinational technology company that designs, develops, and sells consumer electronics, computer software, and online services. Multicast IPs allow group communication to be sent simultaneously to multiple computers.
8) **Multicast** * is not a company, but rather a form of data transmission that allows multiple hosts to receive data simultaneously.

The Internet Assigned Numbers Authority (IANA) is responsible for associating port numbers with certain internet protocols used by network applications. These identifications can be found in IANA's *Service Name and Transport Protocol Port Number Registry*. Table III listed the top destination port numbers as recorded in the data. Each port number has a corresponding application protocol, which are as follows:

1) **Hypertext Transfer Protocol Secure (HTTPS)** - secures communication over a network
2) **Domain Name System (DNS)** - matches names to IP addresses and vice versa to facilitate network communications
3) **Simple Service Discovery Protocol (SSDP)** - advertises presence information to locate available network services
4) **Session Inititation Protocol (SIP)** - signals and controls multimedia communication sessions in Voice over IP (VoIP) applications like online voice and video calls
5) **Microsoft-Directory Services (MS-DS)** - follows the Server Message Block (SMB) protocol for shared access to files, printers, and serial ports and other communications between nodes on a network
6) **Hypertext Transfer Protocol (HTTP)** - facilitates data communication for the World Wide Web
7) **Link Local Multicast Name Resolution** - is based on DNS and allows both IPv4 and IPv6 hosts to perform name resolution for hosts on the same local links
8) **0 (Reserved)** and **7437 (Faximum)** - have no specifics protocols linked to them, and therefore are likely abused ports to send computer attacks and harmful computer and network content like viruses
9) **Simple Network Management Protocol (SNMP)** - collects and organizes information about managed devices on IP networks, and can modify information to change device behavior
10) **Bootstrap Protocol Server (BOOTP)** - exclusive pro-

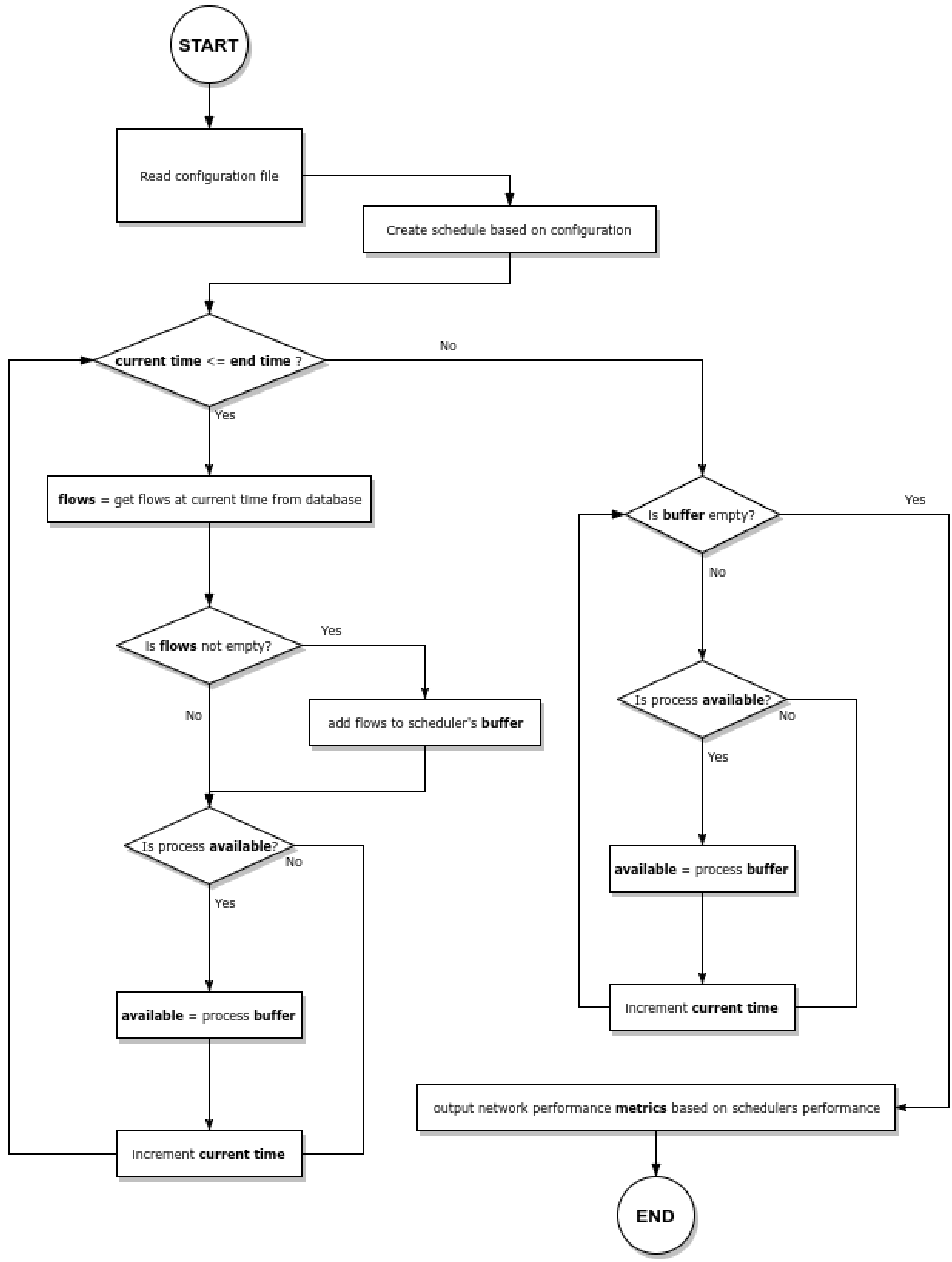

Fig. 1. **UNQS Main Program Logic**

TABLE I
TOP 9 OUT-FLOWS

DESTINATION (# of flows)	TOTAL BYTES	%
KDDI CORPORATION (1)	62409079786	60.6
MULTICAST (3)	12310903701	11.9
Google LLC(3)	5202456530	5.0
Facebook, Inc. (1)	2771346197	2.7
Apple Inc.(1)	1054547100	1.0

TABLE II
TOP 9 IN-FLOWS

SOURCE (# of flows)	TOTAL BYTES	%
KDDI CORPORATION(1)	79135454093	43.0
Google LLC(3)	27958817762	15.2
WorldStream B.V.(1)	3426880570	1.9
M247 Ltd(2)	4286472241	2.3
Converge ICT Solutions Inc.(2)	3967550023	2.2

TABLE III
TOP 10 DESTINATION PORTS

PORT NUMBER	PROTOCOL	TOTAL FLOWS	%
443	HTTPS	65088	11.7
53	DNS	60765	11.0
1900	SSDP	52579	9.5
5060	SIP	43101	7.8
445	Microsoft-DS	28746	5.2
80	HTTP	18485	3.3
5355	LLMNR	17090	3.1
0	Reserved	15732	2.8
7437	Faximum	13734	2.5
161	SNMP	9142	1.6
67	Bootstrap Protocol Server	8602	1.6

tocol for IPv4; Dynamic Host Configuration Protocol (DHCP) server receives requests upon booting the client's computer

B. Network Performance Metrics

Three network performance metrics were accounted for as a result of the simulation, namely, duration, throughput, and flow loss. They are described in the items listed below.

1) **Duration**
Duration d is a function of the arrival time t_0 and last switched time t_n seen as follows:

$$d(t_0, t_n) = t_n - t_0$$

2) **Throughput**
Throughput $tput$ was computed as

$$tput = \frac{count(t_s)}{d}$$

where t_s is the total size (in bytes) that was successfully switched, and d as the computed duration.

3) **Flow Loss**
The last network performance parametric looked at is the flow loss l, solved by

$$l = \frac{count(f_d)}{count(f_d + f_s)} \times 100$$

where all dropped flows f_l are counted and divided by the total number of flows f_d plus f_s (switched flows) multiplied by 100, to get the percentage.

C. Simulation Results

Simulation was run for the bandwidth settings 32.5 Mbps, 35.0 Mbps, 37.5 Mbps, 40 Mbps, and the timeout was set as a constant of 60 seconds.

Table IV contains a summary of the simulation using the network performance metrics as defined in the previous section. It is observed that as the bandwidth is increased, the flows being dropped reduced while the flows being switched increased and became flat. Aside from the information concerning the flows, the simulation results showed that the througput and duration values across the four bandwidth constraints have significantly small difference and thus have similar performance.

V. CONCLUSION AND RECOMMENDATION

Traffic data that was collected using `ntopng` was successfully processed using the FIFO scheduling algorithm as simulated by UNQS. The output performance metrics showed the effect of increasing the bandwidth in order to switch all the flows. It was observed that the actual traffic data had a throughput of 3.45 Mbps. This value was used as a starting point to find out the probable advised bandwidth setting. After several attempts in varying the bandwidth constraint with

TABLE IV
SIMULATION RESULTS PER BANDWIDTH

	BANDWIDTH (Mbps)			
METRICS	**32.5**	**35.0**	**37.5**	**40.0**
flows_dropped_size (Gb)	72.59	18.87	0	0
flows_dropped_cnt	36	9	0	0
flows_switched_size (Gb)	866.46	920.18	939.05	939.05
flows_switched_cnt	554,356	554,383	554,392	554,392
duration (days)	8.22	8.22	8.22	8.22
throughput (Mbps)	1.2	1.3	1.3	1.3

intervals of 500 Kbps, the shift in the network performance metrics was observed between 35.0 Mbps and 37.5 Mbps. While there were dropped flows under bandwidth constraint 35 Mbps, there was none under a bandwidth of 37.5 Mbps. The results for 32.5 Mbps and 40.0 Mbps were included to have an additional data for a better view of the emerging pattern. Therefore it is concluded that with the existing traffic data, a bandwidth of 37.5 Mbps is recommended as the most optimal.

Recommendations based on the analysis of the traffic data that was collected are listed below.

1) Block ports that are still not tagged by the firewall, like ports 0 and 7437, which have potential receive harmful traffic data.
2) Maintain and use an internal DNS server to further reduce external DNS traffic.
3) Turn off MS-DS traffic that consumes a large amount of the out-bound bandwidth.
4) Mirror Linux, Microsoft, and Apple traffic to monitor possible intrusions and unnecessary traffic within the network in order formulate and enforce policies to block and limit such traffic.

Future works may explore more complex scheduling algorithms aside from the native FIFO queueing algorithm like PQ and WFQ in order to determine more optimal scheduling techniques that will minimize the bandwidth constraint and save expenses. Additionally, data could be collected over a longer time period to discover a pattern in network traffic behavior depending on certain periods of time, such as peak season during enrollment period, etc.. Doing so could give a better picture of the maximum bandwidth usage and thus determine the optimal, cost-efficient, and effective bandwidth setting, which can be paired with a queueing technique for further cost reductions and improved performance.

REFERENCES

[1] C. M. Pazos, M. Gerla, and G. Rigolio, "Flow control and bandwidth management in next generation internets," *Telecommunication Systems*, vol. 11, no. 304, pp. 394–412, 1999. [Online]. Available: http://search.proquest.com/docview/212047202?accountid=47253

[2] C. News, "Bandwidth management pays off," *Communication News*, vol. 38, no. 11, p. 54âĂŞ56, Nov 2001.

[3] D. Awduche, A. Chiu, A. Elwalid, and X. Xiao, "Overview and principles of internet traffic engineering," *The Internet Society*, 2002.

[4] Teach-ICT, "Bandwidth." [Online]. Available: http://www.teach-ict.com/as_a2_ict_new/ocr/A2_G063/333_networks_coms/bandwidth/miniweb/pg2.htm

[5] S. Tong and O. Yang, "Bandwidth management for supporting differentiated- service aware traffic engineering," *Parallel and Distrubuted Systems*, vol. 18, no. 9, pp. 1320–1331, 2007.

[6] R. Kanu, S. Kuyoro, S. Ogunlere, and A. Adegbenjo, "Management and control of bandwidth in computer networks," *International Journal of Computer Networks and Wireless Communications*, vol. 2, no. 3, pp. 342–348, 2012.

[7] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM Symposium 1988*, 1988.

[8] D. Gross and C. M. Harris, *Fundamentals of queueing theory*. Wiley, 1974.

[9] E. Lazowska, J. Zahorjan, G. Graham, and K. Sevcik, *Quantitative system performance: Computer system analysis using queueing network models*. Prentice-Hall, Inc., 1984.

[10] A. Paul, "Qos in data networks: Protocols and standards," 1999.

[11] F. Duzbeck, "Bandwidth management is here to stay," *Network World*, vol. 23, no. 13, p. 37, 2006. [Online]. Available: http://search.proquest.com/docview/215977831?accountid=47253

[12] I. L. Stats, "Number of internet users," 2016. [Online]. Available: http://www.internetlivestats.com/internet-users/

[13] D. Comer, *Computer Networks and Internets*. Prentice-Hall International, Inc., 1999.

[14] J. Jerkins and J. Wang, "Ffrom network measurement collection to traffic peformance modeling: challenges and lessons learned," *Journal of the Brazilian Computer Society*, vol. 5, no. 3, 1999. [Online]. Available: http://dx.doi.org/10.1590/S0104-65001999000100003

[15] *WAN and Application Optimization Solution Guide [PDF document]*. Cisco Systems Inc., 2008. [Online]. Available: http://www.cisco.com/c/dam/en/us/td/docs/nsite/wan_optimization/WANoptSolutionGd.

[16] P. Schneider, "Tcp/ip traffic classification based on port numbers (diploma thesis) [pdf document]," 1996. [Online]. Available: http://www.schneider-grin.ch/media/pdf/diploma_thesis.pdf

[17] L. Karim, N. Nasser, T. Taleb, and A. Alqallaf, "An efficient priority packet scheduling algorithm for wireless sensor network," *IEEE ICC 2012 Ad-hoc and Sensor Networking Symposium*, pp. 335–338, 2012.

[18] R. Muhilan, S. Arulselvi, and T. Kiran Kumar, "Packet scheduling algorithms simulator wfq, scfq and wf2q," *International Journal of Scientific & Technology Research*, vol. 2, no. 10, pp. 249–252, Oct 2013.

Leensey M. Lawas She is a BS Computer Science undergraduate student. She not only writes code, but also songs, poems, and stories.