Simulation of Token Bucket Algorithm for Network Traffic Performance

Nor Afifah Mat Nawi[1], Robiah Yusof[2]
[1,2] Centre for Advanced Computing Technology,
Faculty of Information and Communication Technology
Universiti Teknikal Malaysia Melaka, Malaysia.
Email[2]: robiah@utem.edu.my

Abstract - The internet is the most important medium for users in the cyber world. Along with the development of internet technologies allows users to do multi-tasking, nonetheless, it initiates collision. The traffic collision is one of the problems that occurred in the wired network. In order to provide a high-performance network, the quality-of-service (QoS) is required. Traffic shaping is one of the tools that can provide QoS in the network traffic. However, implementing the basic mechanism of traffic shaping cannot solve this issue. In this research, the optimal size of bucket in Token Bucket Algorithm (TBA) is used as one of the techniques in traffic shaping. The variables of this algorithm are modified in order to offer the effective proposed token bucket size that can be used in wired network and tested using OPNET modeler. This algorithm has successfully reduced the delay in the network traffic and improve the response time when users request several applications concurrently.

***Index Terms*—** Token Bucket algorithm, Traffic shaping, Traffic collision, OPNET modeler, QoS

I. INTRODUCTION

Network traffic becomes busy due to users send or request various kind of data from all over the places. The development of technologies together with the internet has become the most important medium for users to complete the tasks. When all the tasks is done concurrently, it can affect the productivity of the business network's users, causing the unstable connections and slow speeds due to traffic collision. Collision can occur when there are many devices attempt to send data at the same time with same channel [1]. This phenomenon will affect the user's performance in completing their job and indirectly affect the company's business [2].

Unmanaged network traffic can cause collision [3]. Hence, users that connected to the unmanaged network shall experience low network performance due to packet delay especially during the peak hour. Thus, an administrator needs to manage their network to provide best network performance even in peak hour. Users who connected to network required mechanism in order to detect or avoid collision. The network needs to be managed using traffic shaping [4] and it can be shaped using Token Bucket Algorithm. In this research, Token Bucket Algorithm approach is implemented as this algorithm is often used in network to perform traffic shaping or policing.

In this algorithm, there will be a bucket that contain a number of tokens. This mean, if the size of bucket is large, the number of tokens is more compare to the bucket with smaller size. Once there are sufficiently many tokens, the packet that currently wait in the packet buffer enters the network, and removes its token from the bucket. Otherwise, if the token bucket does not have enough tokens, the head packet may not leave the buffer, and arriving packets may drop in the event that the buffer fills up [5].

Hence, when so many applications run at the same time, it takes longer time to response due to the limit or fixed size of bucket in Token Bucket algorithm, which lead to data collision in wired network. The time of bucket to refill with tokens also effect the packet delay during data transmission. Small size of bucket can refill token faster compare the bigger size of bucket. But, the algorithm with small bucket will face the delay more frequent compare to big bucket. Therefore, this research proposed enhance algorithm with effective proposed token bucket size to overcome the network delay.

The rest of the paper is structured as follows. Section 2 discusses the related work on traffic, protocol, performance requirement, QoS, and technique for shaping. Section 3 presents the research implementation of TBA. Section 4 discusses the result and findings of the research. Section 5 elaborate discussion on the findings and finally, Section 6 concludes and summarizes future directions of this work.

II. RELATED WORK

A. What is Traffic?

Traffic is defined as a sequence of packets that transmit from a source to another destination. RFC2722 defines traffic flow as a connection or call but limited by start and stop time [6]. Meanwhile, RFC3697 define traffic as a flow that contains sequence of packets that sent from specific source to the specific destination either in unicast, anycast, or multicast group [7].

B. Protocol

The data can be transmitted using a set of communication protocols known as internet protocol suite. This protocol is used to communicate with the devices. There are three protocols that are commonly used in networking which are TCP, ICMP and UDP protocol.

Transmission Control Protocol (TCP) provides a high reliable, connection oriented and error-checked transmission of a packet between the applications or hosts in IP network [8]. IP is the communication protocol in Internet protocol suite for relaying datagrams across the network. IP together with TCP is commonly referred to TCP/IP. TCP enhances a good functionality to the IP services. IP function is to transmit data from source to destination based on IP address in the packet header [9]. The first version of IP is IPv4 and current version is IPv6 [10].

Internet Control Messaging Protocol (ICMP) is one of the network protocols that are used in Internet Protocol (IP) network management and administration. This protocol required the element of IP implementations. ICMP is a control protocol, which mean it is designed not to carry the data, unlike TCP/IP and UDP. This protocol carries the information about the status of the network itself. ICMP is used by network devices likes routers to send error messages likes a requested service is not available or the host could be reached. This protocol also used to relay query messages [3]. ICMP is one of important part of the Internet Protocol Suite, as defined in [8]. There are five types of errors handled by ICMP; destination unreachable, source quench, time exceeded, parameter problems, and redirection.

User Datagram Protocol (UDP) is a simple protocol which is connectionless, no reliability, flow-control, or error-recovery functions. Even that, UDP gives a fast transmission which is reliable for real-time application [17]. UDP has been used by several applications-layer protocols as follow:

- Network File System (NFS) - use UDP ports 1021/1022.
- Simple Network Management protocol (SNMP) - UDP port 161/162
- Domain Name System (DNS) - UDP port 53
- Trivial File Transfer Protocol (TFTP) - UDP port 69

This research focused on TCP/IP protocol since this protocol provide high reliability and connection-oriented transmission compared to ICMP and UDP protocol.

C. *Performance Requirement*

Performance requirements involve user, application, host and network. This research focus on application performance requirements since applications support the basic connectivity and data transfer between the hosts [11]. Under application performance requirements, there are three variables which is delay, reliability and capacity. This research focus on delay.

Delay variables consist of timeliness and interactivity of user service requirements. Delay are also commonly called as latency and [11] has define it as the time difference of x packet to be send and received when it is transferred and processed the information. Delay is also specified as time taken for a bit of data travel across the network from one node to another.

Sources of delay including propagation (time for signal to reach its destination), transmission (time it takes to push the packet's bits onto the link), queuing (time the packet spends in routing queues), and processing (time the routers take to process the packet header). Meanwhile, there are three type of threshold for delay; interaction delay, human response time, and network propagation delay. These thresholds will help to distinguish low-performance delay and high-performance delay.

In delay, the service metric that are used including end-to-end, round-trip and system delay, latency, delay variation (jitter), and also timeliness. The commonly used to measure delay is end-to-end delay and delay variation. End-to-end delay come from various delay sources such as propagation, queuing, transmission and processing [11]. In IP environment, end-to-end delay is used to measure and monitor all sources of delay. Delay variation is combined with high-performance or specified delay to provide a total delay of application performance requirements that are sensitive to inter-arrival time of information. Example of application that produce delay variation are video, audio, and telemetry information. In this research, delay will be used as the service variables to measure the performance requirements. In Fig. 1, show the taxonomy of delay for application performance requirements which include sources of delay, threshold, and service metric of the delay variables.

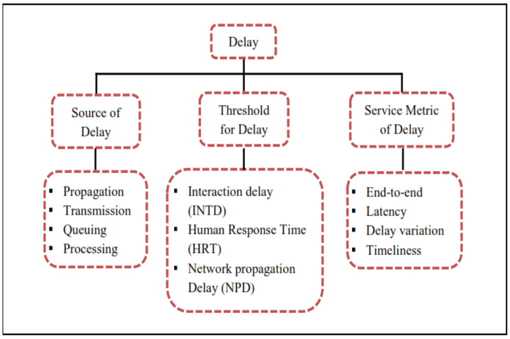


Fig. 1: Taxonomy of Delay for Application Performance Requirements

The sensitivity of network traffic shall depend on the type of application, either real-time application or non-real time application [12]. For real-time traffic, the sensitivity generally focuses on delay and jitter while non-real time application, the sensitivity is packet loss. In real-time traffic, delay is important because the packets are relevant only if it is received in the time period otherwise the packet is not considered important anymore. Similar to jitter, because it interrupts the consistency of the delay consistency. Hence, to minimize the delay, jitter, and packet loss, QoS have to be implemented in the network traffic as QoS allow the traffic to get the access privileged to network resources in terms of less delay and less jitter.

D. *Quality of Services (QoS)*

Services providers in IT fields has the most challenging tasks to ensure excellent QoS in their services and applications. QoS is used to classify the ability of network traffic to send different level of guaranteed services for different type of traffics [13]. In addition, QoS has two mechanism which is admission control and traffic control. Admission control is mechanism that specified which applications and users on network resources can used network segment. While traffic control regulates the data flows by using QoS tools.

- ***QoS Model***

According to [13] and [14], QoS are categorized in three models which is best-effort, integrated services (IntServ), and differentiated services (DiffServ). This research shall focus on best-effort and IntServ models since both real-time and non-real-time application are used.

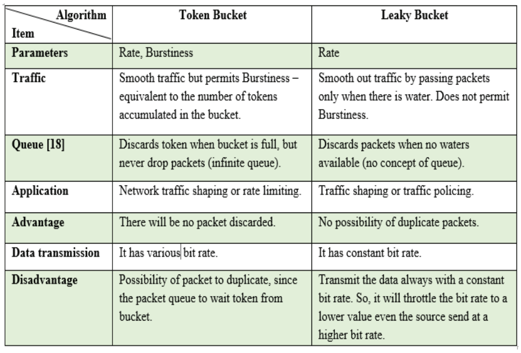
- ***QoS Tools***

QoS has several tools which has the capability to Classifying & Marking, Policing & Shaping, and Queuing & Scheduling. In general, a QoS tool is simply a building block. It receives input and process it, and then produce an output. The relationship between input and output depending on the mechanism of the tools. These tools shall do the classifying, scheduling, and marking packet based on the priority and then shaping traffic by limiting the rate of flow. This research implements tools which has the capability to do policing & shaping. The policing is responsible to ensure that traffic conforms to a defined rate called the bandwidth limit. Moreover, it also determines whether packets are conforming to administratively-defined traffic rates and take action accordingly. The output of the policer tool is the traffic that was present at input, but has a limited based on bandwidth-limit parameter, with excess traffic being discarded.

Then, shaping is a tool that causes a traffic flow to conform to a bandwidth value referred to as shaping rate. Excess traffic beyond the shaping rate is stored inside the shaper and transmitted only when doing so does not violate the defined shaping rate. The differences between the policer and shaper is in terms of dealing with the excess traffic. The policer discards the excess traffic, while the shaper stores the excess traffic to be transmitted later.

E. Technique for Shaping

Traffic shaping is one of the QoS tools used to manage and shape network traffic. Traffic shaper controls access to the available bandwidth, to ensure that traffic conform to the policies established for it, and also to regulate the flow of traffic to avoid congestion occurred if the transmitted traffic has exceeded the access speed of its remote. Furthermore, the traffic shaper is also used to optimize or guarantee the network performance, to improve the latency, and able to increase the usable bandwidth for certain packets by delaying others packets. There are two techniques available in traffic shaping which are Token Bucket Algorithm and Leaky Bucket Algorithm. Fig. 2 show the differences between both algorithm in perspectives of parameters, traffic, queue, application, advantage, data transmission and disadvantage.



Algorithm Item	Token Bucket	Leaky Bucket
Parameters	Rate, Burstiness	Rate
Traffic	Smooth traffic but permits Burstiness – equivalent to the number of tokens accumulated in the bucket.	Smooth out traffic by passing packets only when there is water. Does not permit Burstiness.
Queue [18]	Discards token when bucket is full, but never drop packets (infinite queue).	Discards packets when no waters available (no concept of queue).
Application	Network traffic shaping or rate limiting.	Traffic shaping or traffic policing.
Advantage	There will be no packet discarded.	No possibility of duplicate packets.
Data transmission	It has various bit rate.	It has constant bit rate.
Disadvantage	Possibility of packet to duplicate, since the packet queue to wait token from bucket.	Transmit the data always with a constant bit rate. So, it will throttle the bit rate to a lower value even the source send at a higher bit rate.

Fig. 2: Comparison of Token Bucket vs Leaky Bucket Algorithm

Based on Fig. 2, Token Bucket algorithm provide various bit rate compare to Leaky Bucket which has constant bit rate. Hence, TBA is suitable for this research as this research required various bandwidth rate for various data type.

Token Bucket Algorithm mechanism with the weighted fair queue is used as the shaper to manage network traffic. Token bucket is often used in network routers and gateways to perform traffic shaping or policing. The basic concept of TBA is illustrated in Fig.3. The bucket acted as a bandwidth threshold of the bandwidth capacity. It allows the packet to past through the traffic shaper if the packet has the token. The arrived packet will not going anywhere without the token and the token will have the same amount as given rate [5].

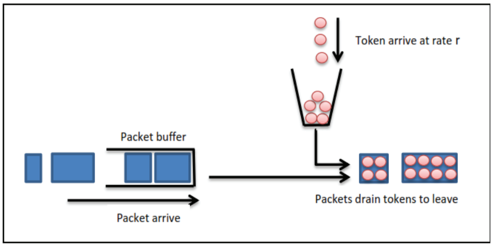
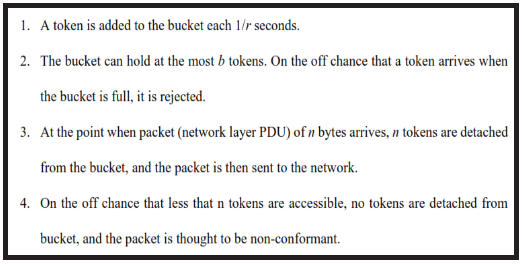


Fig. 3: Basic Concepts of Token Bucket Algorithm

This algorithm is limited by the number of tokens in the bucket. Every bytes of the data need to pass through the network traffic and it has to take token from the bucket. If there are no tokens left, the bytes of data cannot pass through the network and it has to wait. A bucket is refilled with the tokens at a specific time, once the bucket is refilled again, the bytes can retry to pass through. The token bucket algorithm is depicted in Fig.4.



1. A token is added to the bucket each 1/*r* seconds.
2. The bucket can hold at the most *b* tokens. On the off chance that a token arrives when the bucket is full, it is rejected.
3. At the point when packet (network layer PDU) of *n* bytes arrives, *n* tokens are detached from the bucket, and the packet is then sent to the network.
4. On the off chance that less that n tokens are accessible, no tokens are detached from bucket, and the packet is thought to be non-conformant.

Fig. 4: Token Bucket Algorithm

The Token Bucket Algorithm can be modified or change by three variables, the size of the bucket, the amount of token refilled, and the refill time [4]. The size of bucket in the algorithm has to be at least the same as the number of refilled tokens. This is because, according to [4], if the size of bucket is configured higher than the number of tokens refilled, than the limitation of the bandwidth will burst. According to [15], the bucket in the algorithm has a specified capacity. If the bucket fills to capacity, new tokens arrived will be discarded. Thus, at any time, the largest burst for a source to be send into traffic is roughly proportional to the size of the bucket. A token bucket permits burstiness, but bounds it. In other words, the capacity of the bucket effects the transmission time of sending a packet from source to destination.

In this research, the manipulated variable is the size of bucket in the Token Bucket algorithm. It is modified to ensure that the algorithm is more effective in delay application. The bucket size is configured in different size to test the effectiveness of this algorithm when non-real time and real time application is used in network traffic. This enhance algorithm is then tested in the simulation tools.

III. IMPLEMENTATION OF TOKEN BUCKET

This research is implemented using (Optimized Network Engineering Tools) OPNET Modeler 14.5. The scenarios are created to evaluate the effectiveness of the proposed token bucket size algorithm. The OPNET simulator [16] enables us to simulate a whole heterogeneous network with various protocols. Fig. 5 illustrates the workflow of the simulation setup for the scenario.

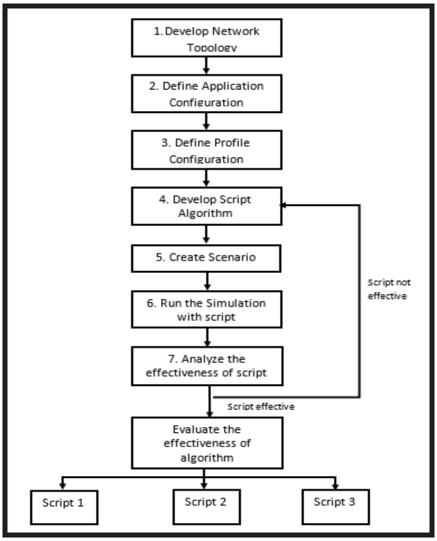

1. Develop Network Topology
2. Define Application Configuration
3. Define Profile Configuration
4. Develop Script Algorithm
5. Create Scenario
6. Run the Simulation with script
7. Analyze the effectiveness of script
Script not effective
Script effective
Evaluate the effectiveness of algorithm
Script 1
Script 2
Script 3

Fig. 5: Research Simulation Workflow

Step 1: Develop Network Topology

LAN architecture is used and two users are connected to a switch. The switch will then connect to a router, and this router is connected to other routers from other LAN. There will be communication between the client and server from different LAN.

Step 2: Define Application Configuration

The real-time and non-real-time application are used such as FTP, Email, Video Conferencing, and VoIP. All the configuration is done in Application Config in OPNET.

Step 3: Define Profile Configuration

In this step the profile used in the application is specified. There are two users created namely User1 and User2. Both users will used four profiles and they are configured in Profile Config. The profiles are FTP_Profile, Email_Profile, Video_Profile, and VoIP_Profile.

Step 4: Develop Script Algorithm

The script of enhance Token Bucket algorithm is develop using Visual Studio 10 with C++ language. The scripts are categorized as 1st size bucket of TBA, 2nd size bucket of TBA, and 3rd size bucket of TBA. The different size of bucket is measure based on the depth of the bucket. Then, this script, will be compiled into OPNET modeler. The process flow of the coding design is depicted in Fig. 6.

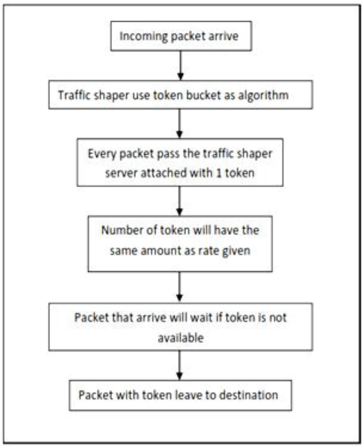

Incoming packet arrive
Traffic shaper use token bucket as algorithm
Every packet pass the traffic shaper server attached with 1 token
Number of token will have the same amount as rate given
Packet that arrive will wait if token is not available
Packet with token leave to destination

Fig. 6: Process Flow of TBA coding design

Step 5: Create Scenarios

These three scenarios are created in the network simulation models.

- Scenario 1: 1st size bucket of TBA - 4 Mbps
- Scenario 2: 2nd size bucket of TBA - 12 Mbps
- Scenario 3: 3rd size bucket of TBA – 22 Mbps

Step 6: Run the Simulation with Script

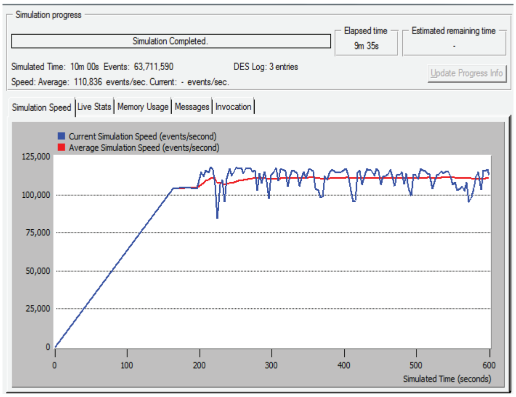
In this research, interest statistic of Ethernet (delay) is chosen in order to measure the delay in Ethernet traffic. Delay is a service metric used to measure the network performance based on application requirements. Combination of non-real time and real-time application will give a different delay since each application required different QoS in network performance.

Step 7: Analyzed the Effectiveness of Script

The result are analyzed to identify the most effective enhanced algorithm when dealing with non-real application and real-time applications.

IV. RESULTS AND FINDINGS

Referring to Fig. 7 until Fig. 14, the average of all application's packet delay are illustrated in dimensional graph for each scenario. In each graph the X-axis represents the simulation time in seconds, while Y-axis represent the average simulation speed (event/second). Fig. 7 shows the graph generated without integrating the enhanced algorithm script. Hence, no traffic shaping implemented in this network to overcome the network delay. The average speed is 110,836 events/sec within 9 minutes and 35 seconds with 63,711,590 events running in the simulation. The graph is fluctuated as it increase and decrease frequently due to unmanaged traffic and QoS in the applications.


Simulation progress

Simulation Completed.

Elapsed time: 9m 35s

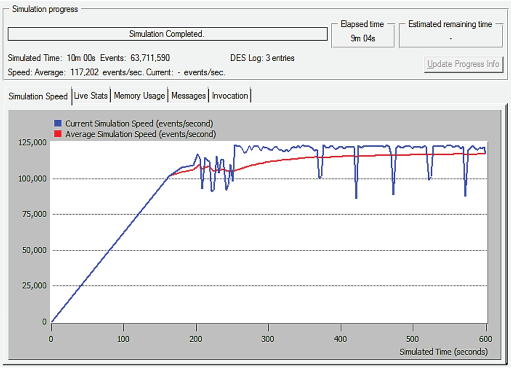
Estimated remaining time: -

Simulated Time: 10m 00s Events: 63,711,590 DES Log: 3 entries

Speed: Average: 110,836 events/sec. Current: - events/sec.

Fig. 7: Network Traffic without TBA script

In Fig.8, the graph is generated using script that contain token bucket size of 4 Mbps (Scenario 1). The average simulation speed in the network traffic is 117,202 events/sec within 9 m 04 sec time elapsed, and 63,711,590 events running in the simulation. The graph varies at 150th seconds whenever VoIP application simulation is started. From the beginning of simulation start time, the average of speed is increasing rapidly even in 100th sec and 125th sec during FTP and Email application simulation start time respectively


Simulation progress

Simulation Completed.

Elapsed time: 9m 04s

Estimated remaining time: -

Simulated Time: 10m 00s Events: 63,711,590 DES Log: 3 entries

Speed: Average: 117,202 events/sec. Current: - events/sec.

Fig. 8: Network Traffic with token bucket size of 4 Mbps (Scenario 1)

The Ethernet delay graph in Fig.9 for Scenario 1 shows that the delay is increasing rapidly up to 0.010 sec and maintain until end of simulation.

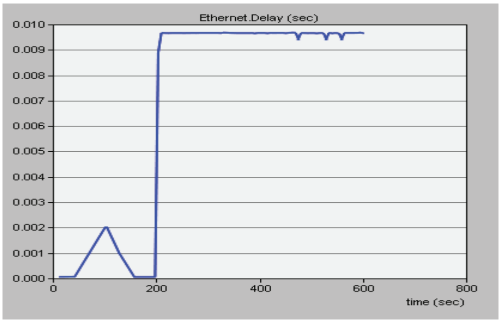
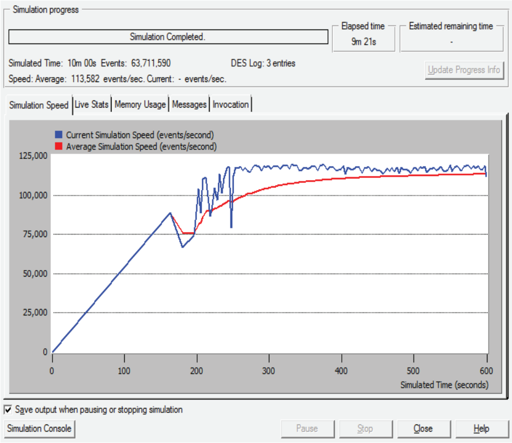


Fig. 9: Ethernet delay for Scenario 1

In Fig. 10, the graph is generated using script contain token bucket size of 12 Mbps (Scenario 2). The average simulation speed in the network traffic is 113,582 events/sec within 9m 21 sec time elapsed, and 63,711,590 events is running in the simulation. Hence, with the 12 Mbps size of token bucket, the delay is lower than Scenario 1 in Fig.8. The delay is decreasing using bigger size of bucket in Token Bucket Algorithm.


Simulation progress

Simulation Completed.

Elapsed time: 9m 21s

Estimated remaining time: -

Simulated Time: 10m 00s Events: 63,711,590 DES Log: 3 entries

Speed: Average: 113,582 events/sec. Current: - events/sec.

☑ Save output when pausing or stopping simulation

Fig. 10: Network Traffic with token bucket size of 12 Mbps (Scenario 2)

The Ethernet delay graph in Fig.11 for Scenario 2 shows that the delay increase speedily until reach 0.0056 sec and the delay maintained until end of simulation.

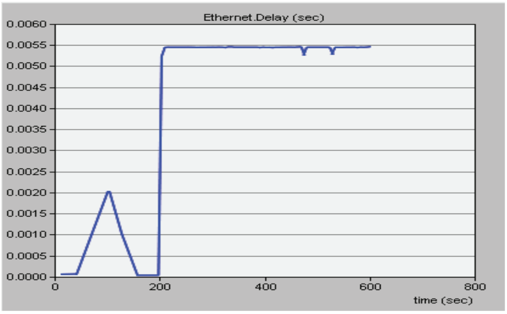
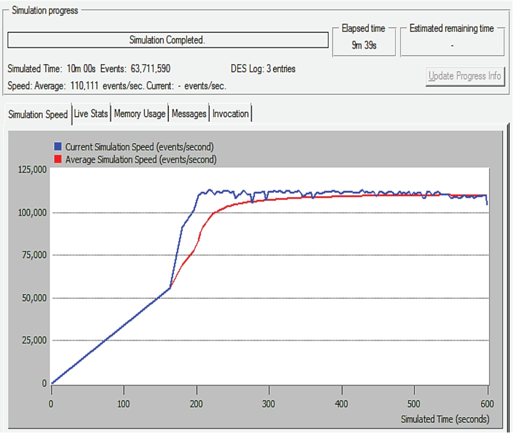


Fig. 11: Ethernet delay for Scenario 2

Refer to Fig. 12, the graph is generated using script contain token bucket size of 22 Mbps (Scenario 3). The average simulation speed in the network traffic is 110,836 events/sec within 9m 35 sec time elapsed, and 63,711,590 events is running in the simulation. The graph is more constant compare to Scenario 2. From the beginning of simulation start time, the average of speed is increasing constantly even in 100th seconds and 125th seconds for FTP and Email application respectively. Furthermore, there are no drop of simulation speed at 150th seconds during starting of VoIP application.


Simulation progress

Simulation Completed.

Elapsed time: 9m 39s

Estimated remaining time: -

Simulated Time: 10m 00s Events: 63,711,590 DES Log: 3 entries

Speed: Average: 110,111 events/sec. Current: - events/sec.

Fig. 12: Network Traffic with token bucket size of 22 Mbps (Scenario 3)

The Ethernet delay graph in Fig.13 for Scenario 3 shows that the delay is only increasing up to 0.0025 sec and its maintain until the end of simulation.

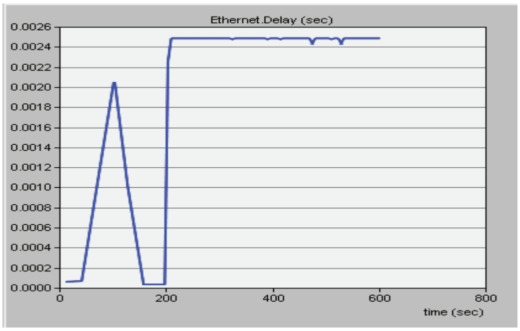


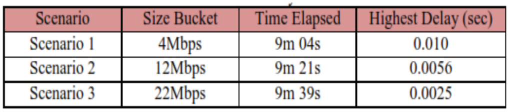
Fig. 13: Ethernet delay for Scenario 3

Based on these three scenarios, the result shows that as the size of the token bucket getting bigger which is form 4 Mbps to 22 Mbps, the the network traffic delay will become lesser which is from 0.010 sec to 0.0025 sec.

V. Discussion

Refering to the Network Traffic graph generated in Fig.8, Fig. 10 and Fig. 12 from Scenario 1, 2 and 3 resepectively, Scenarios 3 contain less delay compare to Scenario 1 and 2 as it has almost constant speed as the size of token bucket is increasing. Furthermore, the results of the Ethernet delay graph from Fig.9, Fig. 11 and Fig. 13 from Scenario 1, 2 and 3 resepectively are summarized in TABLE 1. The evaluation of modeled scenario with the best performance is based on the lowest delay in network traffic.

TABLE I
SUMMARIZATION OF ETHERNET DELAY GRAPH OUTPUT



Scenario	Size Bucket	Time Elapsed	Highest Delay (sec)
Scenario 1	4Mbps	9m 04s	0.010
Scenario 2	12Mbps	9m 21s	0.0056
Scenario 3	22Mbps	9m 39s	0.0025

In TABLE I, it shows that as the size of bucket is increasing which is from 4 Mbps to 22 Mbbps, the network traffic delay will become lesser which is from 0.010 sec to 0.0025 sec.

The size of the bucket effects the transmission time of sending a packet from source to destination due to bucket has its specified capacity. Every bytes of the data need to pass through the network traffic and it has to take the token from the bucket. If there are no tokens left, the bytes of data cannot pass through the network and it has to wait and this caused delay. By increasing the size of the token bucket, every bytes of data have the chance to obtain the token from the bucket and the data can pass through the network traffic without delay. This research has found that as the size capacity of the token bucket increase, the network delay will become lesser.

Thus, this research has successfully identified the relationship between the token bucket size and network traffic delay using Token Bucket Algorithm. TBA is suitable for this research as this research required various bandwidth rate for various data type which cannot be implemented in Leaky Bucket Algorithm due to it constant bit rate.

VI. CONCLUSION

In conclusion, this research has successfully identified the optimal size of token to be implemented in TBA. It shows that as the size of token bucket increase, the network traffic delay will become lesser. This result offers significant contributions towards researchers in the field of traffic shaper in QoS in term of providing an assessable study of TBA and network traffic model in various type of application in network. At the same time, it provides network administrator with a proposal of general analysis method of using optimal Token Bucket size algorithm based on application requirements. This research only focusing in bucket size in TBA, in future, the amount of token refilled and refill time in TBA can be further explored.

ACKNOWLEDGMENT

The authors would like to thank the Research Group of Information Security, Forensics and Computer Networking, Center For Advanced Computing Technology (C-ACT), Fakulti Teknologi Maklumat dan Komunikasi (FTMK), Univeristi Teknikal Malaysia Melaka (UTeM)

REFERENCES

[1] Rouse, M.. Collision. Retrieved May 20, 2020 from http://searchnetworking.techtarget.com/definition/collision

[2] Evolven. Downtime, Outages and Failures - Understanding Their True Costs. Retrieved April 14, 2020 from http://www.evolven.com/blog/downtime-outages-andfailures-understanding-their-true-costs.html

[3] Forouzan, B. A. (2012). Data Communications and Networking - Global Edition. McGraw-Hill Education – Europe. ISBN: 9814577510

[4] Faheem, Fahd Humayun, Mohammad Inayatullah Khan Babar, Mohammad Haseeb Zafar, and Megat Farez Zuhairi. Performance analysis of a Token Bucket Shaper for MPEG4 video and Real Audio signal. In Proceeding of IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), 2013

[5] Yahaya, M. K. (2006). Traffic Shaper Integrated with Open Source Traffic Grapher for Network Administration. In Bachelor Final Year Project, 2006

[6] Brownlee, N., Mills, C., & Ruth, G. Traffic Flow Measurement: Architecture. Retrieved Jan 10, 2020 from https://tools.ietf.org/html/rfc2722

[7] Rajahalme, J., Conta, A., Deering, S., & Carpenter, B. E. IPv6 Flow Label Specification. Retrieved Jan 15, 2020 from https://tools.ietf.org/html/rfc3697

[8] Postel, J. Transmission Control protocol. Retrieved Jan 15, 2020 from https://www.rfc-editor.org/rfc/rfc793.txt

[9] Kastenholz, F., & Partridge, C. Technical Criteria for Choosing IP The Next Generation (IPng). Retrieved Feb 10, 2020 from https://tools.ietf.org/html/rfc1726

[10] Stephen E. Deering, R. M. H. Internet Protocol, Version 6 (IPv6) Specification. Retrieved Jan 15, 2020 from https://tools.ietf.org/html/rfc2460

[11] McCabe, J. D. (1998). Practical Computer Network Analysis and Design. San Francisco: Morgan Kaufmann Publisher. ISBN: 978-1-55860-498-8

[12] Barreiros, M., & Lundqvist, P. (2016). QoS Enabled Networks: Tools & Foundations. John Wiley & Sons Ltd. ISBN: 978-1-119-10910-5

[13] Duran-Limon, H. A., & Blair, G. S. (2004). QoS Mana gement Specification Support for Multimedia Middleware. Journal of Systems and Software, 72(1), 1–23. ISSN: 01641212

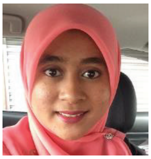
[14] Wallace, K. (2011). Implementing Cisco Unified Communications Voice over IP and QoS. USA: Cisco Press. ISBN-10: 0-13-210343-5

[15] Cisco. (2019). Comparing Traffic Policing and Traffic Shaping for Bandwidth Limiting. Retrieved May 20, 2020, from https://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-policing/19645-policevsshape.html

[16] Prokkola, J. (2006). OPNET - Network Simulator. Simulation and Tools for Telecommunications.

[17] Kurose, J. F., & Ross, K. W. (2013). Computer Networking A Top-Down Approach. Pearson. ISBN: 978-0-13-285620-1

[18] G. F. A. Ahammed and R. Banu, (2010). Analyzing the Performance of Active Queue Management Algorithms. Int. J. Comput. Networks Commun., vol. 2, no. 2, pp. 1–19, 2010.



Nor Afifah Mat Nawi holds Master of Internetworking Technology from Universiti Teknikal Malaysia Melaka, Malaysia. Her Research Interest includes wired and wireless networking and network security.

Robiah Yusof is currently a senior lecturer at the Universiti Teknikal Malaysia Melaka, Malaysia. She received her PhD in Network Security from Universiti Teknikal Malaysia Melaka. Her Research Interest include intrusion detection, malware, network security and network forensic.