Performance Analysis of Network Traffic Models for DiffServ QoS Router.

A.O. Ameen[1]; D.D. Olatinwo[1]; and S.O. Olatinwo[2]

[1]Department of Computer Science, Faculty of Communication and Information Sciences, University of Ilorin, Ilorin, Nigeria.
[2]Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria

E-mail: ahmedameeny2k4@gmail.com
damibaola@gmail.com

ABSTRACT

This paper presents the design and evaluation of network traffic models to examine the strengths and weaknesses of the TSW2CM and TSW3CM algorithms on different data traffic using transmission control protocol (TCP) and file transfer protocol as the traffic protocol (agent) and traffic generator. The TSW2CM and TSW3CM algorithms were implemented on the designed traffic models to define the treatment each traffic scenario receives as it transverses through the routers in the network.

The traffic models were simulated, analyzed, and evaluated. The evaluation was done using packet loss and one-way latency (packet delay) as QoS parameters, and the simulation was carried out on network simulator 2 (NS-2). The results obtained in this study were further analyzed using a ranking system approach which revealed the strengths and weaknesses of various applications (TCP traffic flow) on TSW2CM and TSW3CM algorithms using the above mentioned QoS parameters. The ranking system revealed that TCP-traffic based on TSW2CM and TSW3CM algorithms had a packet loss rate of 0.23% and 0.90%, and one-way packet delay values of 0.058282 and 0.045672, respectively. Therefore, the evaluation results revealed that applications that require a low packet loss rate can be deployed on TCP protocol using TSW2CM algorithm, and applications that requires a low latency (packet delay) can be deployed on TCP protocol using TSW3CM algorithm.

(Keywords: TCP, multimedia application, time sliding window, TSW2CM, TSW3CM, IETF, differentiated service, DiffServ, QoS router, quality of service)

INTRODUCTION

In recent years, there has been a tremendous increase in the advent of modern multimedia applications on the Internet such as video streaming, Voice over Internet Protocol (VoIP), telemedicine, and interactive distance learning, to mention just a few, and this has led to the congestive nature of the modern routers. The applications are routed through the DiffServ routers in a network as data traffic to their destinations and they require some guarantee to finish within a stipulated period of time. This demand some certain service quality, otherwise referred to as Quality-of-Service (QoS) provisioning, considering their time sensitive nature and their varying performance requirements such as throughput, packet loss rate, fairness, and end-to-end delay (latency), as they transverse through the DiffServ routers to their destinations. QoS is an integral aspect of internet and computer networks, and it is used to offer a predictable service and control end-to-end congestion [1].

The current internet is confronted with QoS issues due to the limitations of the ancient Transmission Control Protocol/Internet Protocol (TCP/IP) traffic management models to support real-time applications, thereby, the real-time applications on the internet are often faced with network congestive messages such as “network busy” or “try again later” [2]. Moreover, the current internet was built on the best effort services architecture which treats all packets equally. The best effort services architecture is effective in handling traditional internet (non-real time) applications such as emailing, web surfing, file transfer and so on.

To improve on the best effort service architecture, the Internet Engineering Task Force (IETF) came up with two different architectures namely the integrated services architecture (IntServ) and differentiated services architecture (DiffServ). The IntServ architecture improved on the best effort service architecture by reserving resources such as bandwidth and buffer for each flow in order to meet the QoS requirements of each application (real-time and non-real time). However, this architecture is confronted with scalability problem because it has only one single router that combines the edge router and the core router functions; this also resulted in complexity problem.

The DiffServ architecture corrected the scalability problem in the IntServ architecture by splitting the single router used in the IntServ into two, namely the edge router and core router, and replaced the per flow service with per class service. It also moved the complex processing task from the core router to the edge router. Also, the DiffServ architecture introduced an admission control mechanism for checking whether to accept or reject traffic stream using traffic conditioners such as time sliding window (TSW) marker algorithms [3]. To handle admission control in a DiffServ network, the time sliding window two color marker (TSW2CM) and time sliding window three color marker (TSW3CM) algorithms have been proposed as variants of TSW. In order to examine the strengths and weaknesses of the TSW2CM and TSW3CM algorithms for different data traffic, this study proposed the design and evaluation of a TCP-based traffic model. Therefore, the efficient algorithm which is capable of providing a better QoS will be determined among the two algorithms.

QOS THEORY

<u>QoS in Internet</u>

In computer networks, QoS is the capability of a network to provide resource assurance and service differentiation to meet the demands of time-sensitive applications that requires some guarantees to finish within a bounded time period such as VoIP, VoD and so on [4]. According to the author in [2], QoS is defined as the collective effectiveness of service performance that determines the degree of satisfaction of an end user of a given service. The main essence of QoS in Internet is to provide priority such as dedicated bandwidth, controlled delay and jitter, as required by real-time traffic, and to improve packet loss rate. It is also important to make sure that providing priority to one or more flows does not make other flows fail. Moreover, QoS in Internet determine if the service offered by a network meets the users' quality demands.

<u>Internet QoS Performance Metrics</u>

The QoS performance metrics are used to express the service quality of a network using parameters such as delay, packet loss, to mention a few [1]. These parameters are discussed as follows:

Latency (Packet Delay): Latency can also be referred to as packet delay or an end-to-end delay. It can be described as the amount of time that a packet takes from the source node to the destination node. To determin the delay in sending a packet from the source to the destination node, one-way packet delay is required. One-way packet delay is calculated by subtracting the time a packet is packets arrival time at the ingress router (en-queued) a(n) from the time the packet is the time packet got to the destination (de-queued) d(n). Therefore, the latency (packet delay) (δ) is determined using Equation 1 [7-8]:

$$\delta = d(n) - a(n) \tag{1}$$

Packet Loss: The packet loss defines the packets that are dropped along the path in a network as a result of some routers failure to deliver some packets when their buffers are full when they arrived. The entire packet or some may be dropped depending on the state of the network or the applications on the network. Therefore, the consequences of packet loss can be less or more [7]. Packet loss rate can be calculated using Equation 2.

$$Lo = \frac{(Pkt\ enq - pkt\ rec)}{pkt\ enq} * 100 \tag{2}$$

where:

Lo is the packet loss rate, **pkt enq** is the packet en-queued at the ingress router and **pkt rec** is the total packet that got to the destination.

<u>DiffServ Architecture</u>

The DiffServ is a scalable quality-of-service architecture that consists of two important traffic components, namely the traffic classifiers and traffic conditioners. The identified traffic components are discussed as follows.

Traffic Classification

The traffic classifier operates at the ingress edge router and classifies traffic into real-time traffic and non-real time traffic using the differentiated service code point (DSCP) on each packet. There are two types of traffic classifiers namely; the Behavior Aggregate (BA) classifier and the Multi Field (MF) classifier. The knowledge of the packet classifications is important in order to apply the appropriate metering, marking, shaping, and dropping functions to each packet class according to the Service Level Agreement.

Traffic Conditioning

Traffic conditioning functions include metering, marking, shaping and dropping. The traffic conditioning functions are described as follows:

Meter: Metering is an imaginary bucket that measures traffic against traffic profile. Metering measures the traffic profile (In-profile or Out-profile) of each traffic to ensure they comply with the service level agreement (SLA) between the user and the network operator. Moreover, the traffic meter measures the temporal properties of the traffic and passes the result to an important component in the architecture known as marker.

Marker: Marking is used to mark traffic as real-time traffic and non-real time traffic. Marking is done using traffic marker algorithms. The traffic marker set the DS field of a particular code point based on the information from the classifier and meter. This implies that, a packet is assigned to a particular class of service (BA or MF) and may be marked as in-profile or out-profile. Hence, marking of the packets determines the treatment a packet receives as it transverse through the network domain [8].

Shaper: Shaper is used for shaping packets. The sharper ensures that traffic stream conforms to a packet profile. It delays some of the packets in a traffic stream to bring them into compliance with a traffic profile. A sharper usually has a buffer with large size (space) and packets may be discarded if there is no sufficient reduce the average rate of the traffic [8].

Dropper: The dropper works with the sharper to ensure the conformance of a traffic stream to a particular traffic profile through a smart process of dropping some packets in contrast to a sharper.

Variants of Time Sliding Window Marker Algorithms

In this study, the variants of time sliding window traffic marker algorithms evaluated are Time Sliding Window Three Color Marker (TSW3CM) algorithm and Time Sliding Window Two Color Marker (TSW2CM) algorithm.

Time Sliding Window Three Color Marker (TSW3CM)

The TSW3CM is designed as a component in a DiffServ traffic conditioner to meter (measure) traffic stream through a traffic estimator and mark packets to be yellow, green or red color. The marking of the packets is based on the measured throughput (target rate) of the traffic stream and compare with the Committed Target Rate (CTR) and the Peak Target Rate (PTR) [9]. Figure 2 shows the TSW2CM algorithm.

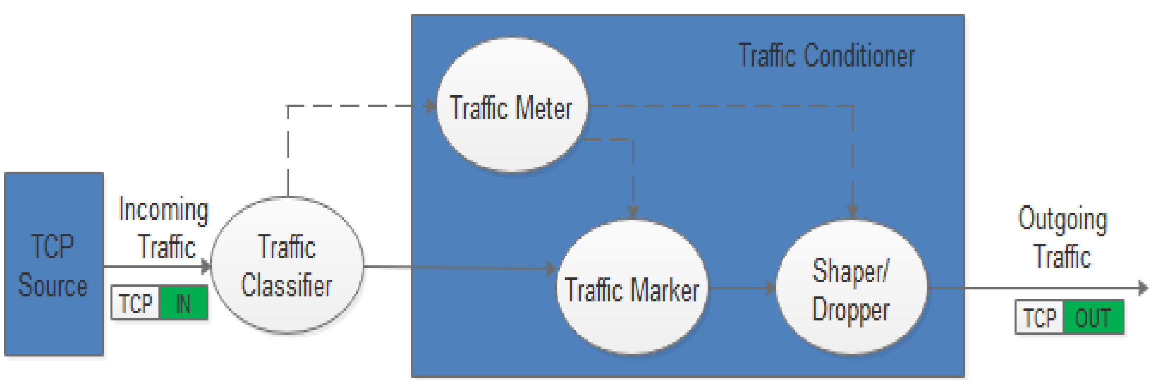

Figure 1: DiffServ Traffic Classifier and Conditioner.

The TSW3CM is designed to mark packets that contributes to the sending rate with any of the three colors based on the following estimated average rate: if the estimated average rate is less than or equal to the CTR, packets of the stream are marked green color, packets are marked with yellow color with a probability of P0 and green color with a probability of (1-P0) if the estimated average rate is greater than the CTR but less than or equal to the PTR. P0 defines the fraction of the packets that contributes to the measured rate beyond the CTR. Also, packets are marked with red with a probability of P1, yellow with a probability of P2 and green with a probability of (1-(P1+P2)) if the estimated average rate is greater than the PTR. P1 defines the fraction of packets that contributes to the measured rate beyond the PTR, P2 defines the fraction of packets that contributes to the measured rate between CTR and PTR [9].

avg_rate = Estimated Average Sending Rate of Traffic Stream

if (avg_rate <= CTR) the packet is marked as green;

else if (avg_rate <= PTR) AND (avg_rate > CTR)

$$\text{calculate P0} = \frac{(avg\_rate \; - \; CTR)}{avg\_rate}$$

with probability P0 the packet is marked as yellow;

with probability (1-P0) the packet is marked as green;

else

$$\text{calculate P1} = \frac{(avg\_rate \; - PTR)}{avg\_rate}$$

$$\text{calculate P2} = \frac{(PTR \; - \; CTR)}{avg\_rate}$$

with probability P1 the packet is marked as red;

with probability P2 the packet is marked as yellow;

with probability (1-(P1+P2)) the packet is marked as green;

Figure 2: General TSW3CM Marking Algorithm [9]

Time Sliding Window Two Color Marker (TSW2CM)

The Time Sliding Window Two Color Marker (TSW2CM) is a simplified version of the TSW3CM. The TSW2CM is only configured with one rate and packets are marked either green or red. It meters a traffic stream according to one traffic conditioning parameter, which is the Committed Target Rate (CTR). The TSW2CM is composed of a rate estimator and a packet marker, the rate estimator provides an estimate of the traffic streams arrival rate. Figure 3 shows the TSW2CM algorithm.

avg_rate = Estimated Average Sending Rate of Traffic Stream

if (avg_rate <= CTR) the packet is marked as green;

else if (avg_rate > CTR) the packet is marked as red;

$$\text{calculate P0} = \frac{(avg\_rate \; - \; CTR)}{avg\_rate}$$

with probability P0 the packet is marked as red;

with probability (1-P0) the packet is marked as green.

Figure 3: General TSW2CM Marking.

RELATED WORKS

Most of the existing studies on traffic marker algorithms in DiffServ network do not effectively address the problems of congestion due to the congestive nature of modern routers, unfairness, packet delays and packet losses. Moreover, most of the existing studies in literature do not consider QoS design factors such as packet delays, and packet loss. Therefore, they have failed to provide adequate direction necessary for the future Internet QoS design.

Kulhari and Pandey (2016) proposed and evaluated a congestion control algorithm at the DiffServ edge routers in order to overcome the congestion problem at queues in DiffServ network by allocating buffer to the incoming traffic according to the present available resource and the priority of the packets. Network simulator 2

(NS-2) tool was used to carry out the experiment, throughput and packet loss rate were considered as the QoS performance metrics. However, the aspect of packet delay rate has not been considered in the proposal, this is a serious drawback of this study, especially when there is need to deal with real-time applications. Laatu *et al*. (2015) evaluated the performance of TCP flows in DiffServ networks to examine the fairness issues among individual TCP flows.

In this study, network simulator 2 (NS-2) tool was used to carry out the simulation while throughput and fairness rate were used as the QoS performance metrics. However, this study did not consider packet delay rate and packet loss rate. This is a serious drawback of this study especially when there is need to deal with real time applications.

RESEARCH METHODOLOGY

Research Approach

This study adopted a simulation approach to carry out the design and evaluation of the proposed traffic model used to implement the traffic marker algorithms: TSW2CM and TSW3CM so as to create a platform for comparison between TCP-based traffic flow on TSW3CM algorithm and TCP-based traffic flow using TSW2CM algorithm. The TCP traffic agent employed a file transfer protocol (FTP) as the traffic generator, and a software simulator called network simulator-2 (NS-2) was used for the simulation. Data was generated based on simulation experiment and traced into output files, analyzed and evaluated based on performance metrics. The performance metrics analyzed was subjected to ranking system approach for the purpose of comparison.

Experimental Design

The network topology designed in Figure 4 represents the topologies used in this research to implement the traffic marker algorithms. Moreover, since multimedia applications are expected to be routed through the core router, TCP was used in this study as the network traffic agent which guarantees delivery through the use of acknowledgements and sequence delivery of data.

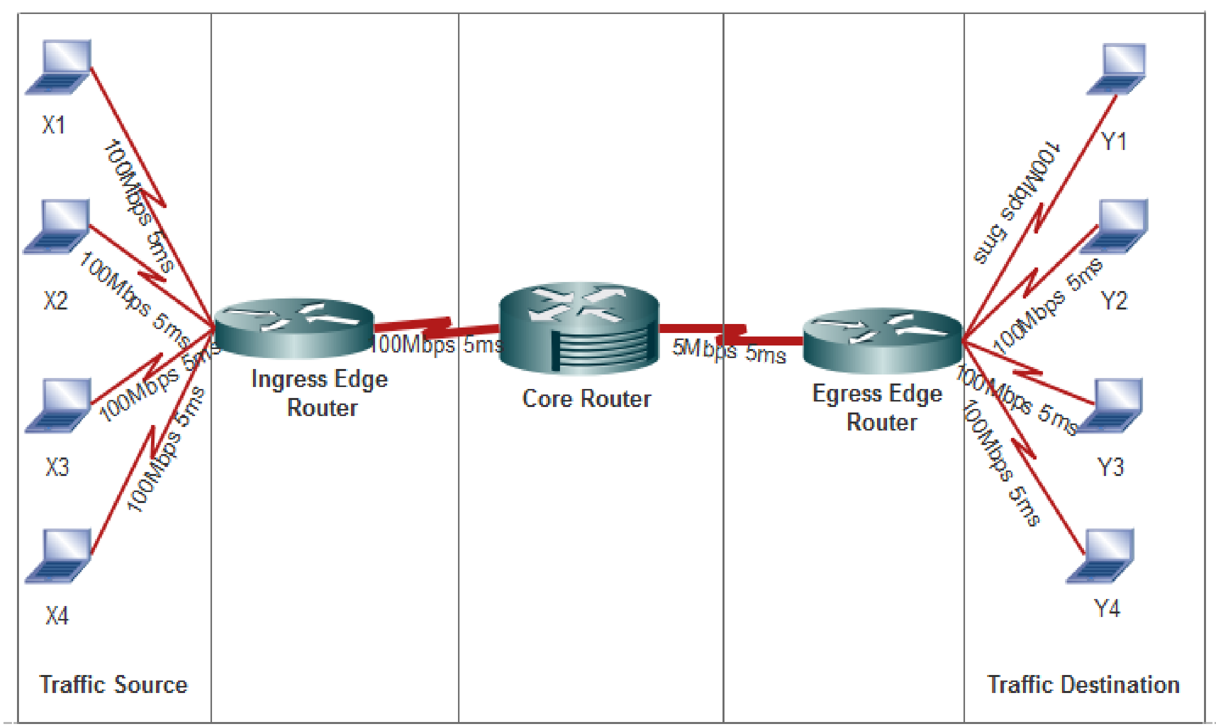

Figure 4: Network Traffic Model.

In the network topology, the same parameter settings were used to implement the two marker algorithms simulated (TSW2CM and TSW3CM) in order to create a platform for comparison among the traffic marker algorithms. The network topology designed consists of eleven nodes (four nodes are for the sources, two nodes are for the edge router, one node for the core router and the remaining four nodes for the destination). The node-to-node network links from the sources to destinations were configured as bandwidth of 100Mbps and link delays of 5ms expect from the core router to the egress edge router which was configured as 5Mbps of bandwidth and 5ms of link delay. The core router to the egress edge router configuration was set to 5Mbps of bandwidth and 5ms of link delay to study the effect of congestion at the core router using the two traffic marker algorithms (TSW3CM and TSW2CM).

The sources (X_1, X_2, X_3 and X_4) generates traffic streams with file transfer protocol (FTP) for TCP traffic agent and sends them to ingress edge router. Network simulator-2 (NS-2) was used for the simulation which provides tools for visualization. The simulation experiment was carried out for 80 seconds for each traffic marker algorithm implemented on the traffic model using sources with different queues and the data generated in the course of the simulation were traced into files.

RESULTS AND DISCUSSION

The evaluation results obtained from the simulation in this study were based on packet loss and one-way latency (delay) as performance metrics. The movement of the packets from core router to edge router in Figure 4 was traced into an output file for the purpose of analysis using the aforementioned metrics. The analyzed results for the TSW3CM and TSW2CM algorithms using TCP as the network traffic agents.

Analysis Based on One-Way Latency

This section describes the evaluation results obtained based on one-way packet delay rate for the two traffic marker algorithms. The lower the packet delay value using TCP as traffic agent the better the performance of the traffic marker. Using TCP as traffic agent, the delay of TSW3CM algorithm (with 0.045672) was better than the delay of TSW2CM algorithm (with 0.058282). In summary, applications that require low one-way packet delay on TCP as traffic agents could use TSW3CM algorithm. Table 1 and Figure 5 shows the one-way packet delay rate evaluation for both traffic marker algorithms (TSW2CM and TSW3CM) using TCP as traffic agent.

Analyses Based on Packet Loss Rate

This section describes the packet loss rate. The packet loss are the packets that arrived at the ingress edge router (en-queued) but did not reach the destination because of congestion or bridging of Service Level Agreement (SLA) between the services provider and the subscriber as shown in Figure 4.

The lower the packet loss rate value, the better the performance of the traffic markers. Table 2 and Figure 6 shows the loss rate evaluation for both traffic marker algorithms (TSW2CM and TSW3CM) using TCP as traffic agent. The lower the loss rate average value, the better the performance of the traffic marker algorithm.

Table 1: One-Way Latency Evaluation for Traffic Based on Traffic Marker Algorithms using TCP as Traffic Agents.

Traffic Markers	One-Way Latency Values (Packets) (ms)	%	Rank
TSW2CM-TCP	0.058282	5.828%	1st
TSW3CM-TCP	0.045672	4.567%	2nd

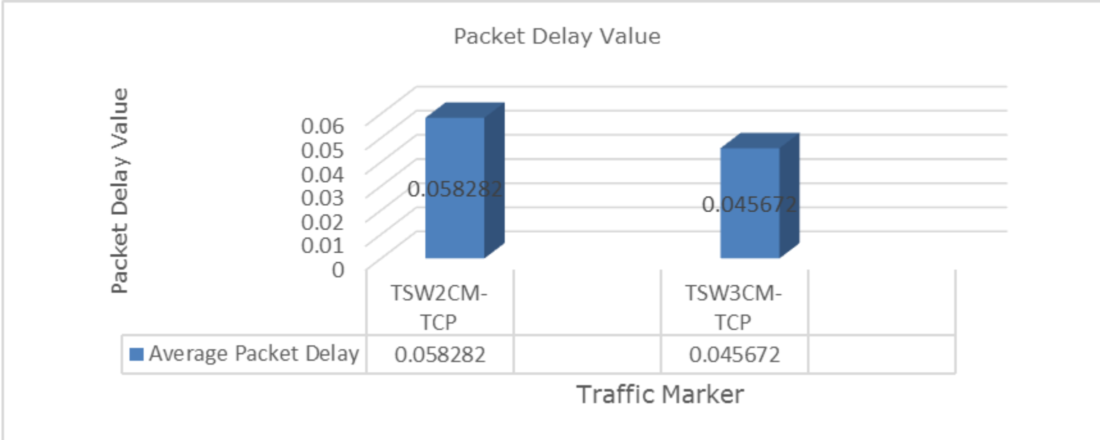

Figure 5: The One-Way Packet Delay for the Traffic Marker Algorithms using TCP.

Table 2: Loss Rate Evaluation for Traffic Based on Traffic Marker Algorithms using TCP.

Traffic Markers	Loss Rate Values (Packets)	%	Rank
TSW2CM-TCP	115	0.23%	1st
TSW3CM-TCP	457	0.90%	2nd

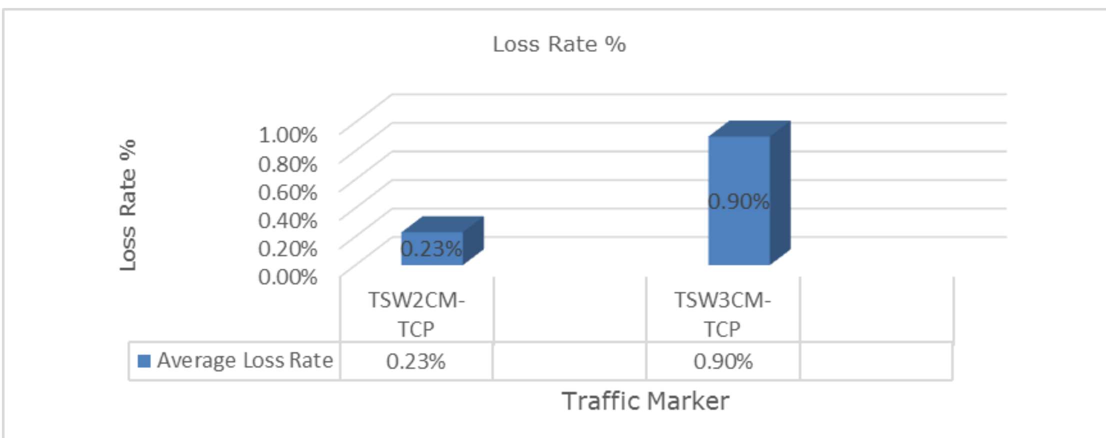

Figure 6: Loss Rate Evaluation for the Traffic Marker Algorithms using TCP.

Table 3: Ranking System Evaluations for Traffic Based on Time Sliding Window Marker Algorithms using TCP as Traffic Agents.

TRAFFIC MARKER	Performance Metrics Evaluation Using TCP			
	One-Way Latency	Rank	Loss Rate %	Rank
TSW2CM-TCP	0.058282	2nd	0.23%	1st
TSW3CM-TCP	0.045672	1st	0.90%	2nd

TSW2CM was ranked first (with 0.23%) and TSW3CM was ranked second (with 0.90%). Hence, applications that require low loss rate on TCP traffic agents could make use of TSW2CM marker algorithms.

CONCLUSION

In this study, the strengths and weaknesses of the existing traffic marker algorithms have been evaluated to determine their efficiencies for various applications (real-time and non-real time) using packet loss rate and one-way latency as performance metrics while TCP was used as the traffic agents. Conclusively, applications that requires low packet loss rate can be deployed on TCP protocol using TSW2CM marker algorithm, and applications that requires low one-way delay can be deployed on TCP protocol using TSW3CM marker algorithm.

REFERENCES

1. Chen, R. 2003. "A Performance Evaluation of Multiplexer Scheduling Algorithms". MSc Thesis, Department of Electrical and Computer Engineering, Concordia University: Montreal, Quebec, Canada.

2. Moreno, L.F. 2012. "Network Performance Evaluation with Real Time Application Ensuring Quality of Service with Ns2". *International Journal of Next-Generation Networks*. 4(3):1-10.

3. Su, H. and M. Atiquzzaman. 2003. "ItswTCM: A New Aggregate Marker to Improve Fairness in DiffServ". *Computer Communications-Elsevier*. 26(9):1018-1027.

4. Zoric, S. and M. Bolic. 2010. "Fairness of Scheduling Algorithms for Real-Time Traffic in DiffServ Based Networks". *IEEE Communication*. 30(5):1591-1596.

5. Firoiu, V., J.Y. Le Boudec, D. Towsley, and Z.L. Zhang. 2002. "Theories and Models for Internet Quality of Service. Proceedings of the IEEE". 90(9):1565 – 1591.

6. Strauss, M.D. 2005. "A Simulation Study of Traffic Conditioner Performance". MSc. Thesis, Faculty of Engineering, Built Environment and Information Technology, University of Pretoria: Pretoria, South Africa. 1-108.

7. Pandit, K. 2006. "Quality of Service Performance Analysis based on Network Calculus". Ph.D. Thesis, Elektrotechnik and Informationstechnik, der Technischen Universitat: Darmstadt, Germany.

8. Stankiewicz, R. 2007. "Analytical Models of Selected DiffServ Network Elements Supporting Assured Forwarding". Ph.D. Thesis, Department of Telecommunications, Faculty of Electrical Engineering, Automatics, Computer Science and Electronics, AGH University of Science and Technology: Kraków Poland.

9. Fang, W., N. Seddigh, and B. Nandy. 2000. "A Time Sliding Window Three Colour Marker (TSWTCM)". Internet Engineering Task Force Draft, RFC 2859.

10. Kulhari, A. and A.C. Pandey. 2016. "Traffic Shaping at Differentiated Services Enabled Edge Router Using Adaptive Packet Allocation to Router Input Queue". IEE 2nd International Conference on Computational Intelligence & Communication Technology. 242-247. doi:10.1109/CICT.2016.55.

11. Laatu, V., J. Harju, and P. Loula. 2015. "The Impacts of Aggregation on the Performance of TCP flows in DS Networks". Retrieved from: https://www.globus.org/sites/default/files/terena.pdf. [Accessed July, 2016].

SUGGESTED CITATION

Ameen, A.O., D.D. Olatinwo, and S.O. Olatinwo. 2017. "Performance Analysis of Network Traffic Models for DiffServ QoS Router". *Pacific Journal of Science and Technology.* 18(1):117-124.

[Pacific Journal of Science and Technology](http://www.akamaiuniversity.us/PJST.htm)