Modified Strategy to Improve QoS in Networks with Varied Traffics

[1]Eneh Agozie Hyacinth, [2]Emmanuel, Ibrahim Tukur, [3]Arinze Uchechukwu Christian, [4]Joshua Bature Hassan, [5]Olorunyomi Segun Omotayo
[1&3]Department of Computer Science, University of Nigeria, Nsukka (UNN), Nsukka, Enugu State, Nigeria
[2]Department of Mathematics/ComputerScience, Adamu Augie College of Education, Argungu Kebbi State
[4]Department of Computer Science, Federal University Oye Ekiti, Ekiti State, Nigeria
[5]Department of Computer Science, Data Centre, Ado-Ekiti, Ekiti State, Nigeria
E-mails: [1]agozieeneh@gmail.com; [2]udabah@gmail.com, [3]arinzechukwu@gmail.com;
[4]joshua.hassan1@fuoye.ng; [5]segun_olorunyomi@yahoo.com

ABSTRACT

The efficiency of buffers in network routers plays an important role in effectively accommodating packets that arrive in bursts at the routers interfaces. The choice of a suitable router buffer size is still a significant problem. Since, the use of small buffer guarantees low packet delay but higher chances of packet lost. Similarly, larger buffer leads to buffer bloat which causes higher delay in a network resulting to poor Quality of Service (QoS). Buffer bloat is a significant problem due to the high changing link characteristics of modern heterogeneous network traffics. The access links can have connections with speedy links (Gbps) with small amount of packet losses and have connection with a susceptible high packet loss with low-bandwidth links like wireless and last mile connections. Hence, what may be thought to be a rational buffer size might be flawed when link rates and delay fall below the minimum value. Thus, this paper analyses buffer size performance and optimization in networks with heterogeneous traffics for Random Early Detection (RED) Queues using NS2 simulations to obtain a range of better suited buffer sizes that improve the QoS without extra signaling and computation by routers. The results obtained established that having a buffer size between five to ten kilobytes yielded best average throughput with low average delay for RED Queue at the congested router Interface. In addition, the optimized buffer size scheme improved the QoS accordingly.

Keywords: Buffer Size, Network Traffic, Quality of Service, Simulation

Proceedings Reference Format
Eneh, A. Hyacinth, Emmanuel, I. Tukur, Arinze Uchechukwu, Joshua, B. Hassan Olorunyomi, Segun Omotayo (2021): Modified Strategy to Improve QoS in Networks with Varied Traffics. Proceedings of the 27th iSTEAMS Multidisciplinary Innovations & Technology Transfer (MINTT) Conference. Academic City University College, Accra, Ghana. June, 2021. Pp 239-254 www.isteams.net/ghana2021. DOI - https://doi.org/10.22624/AIMS/iSTEAMS-2021/V27P20

1. INTRODUCTION

Quality of service (QoS) is fundamental concern to users on the network, nowadays internet technology is taking a lead in a digital era with much economic and societal benefit achieved. Hence, the Internet has provided an unprecedented medium of communication, playing a key role in advancing global technology and economic development. The two essential communication protocols are namely Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), for transporting data between communicating network applications [1][14].

UDP is a connection-less protocol that is used for rapid transfer of data that can tolerate small amount of loss, that is, there is no need to establish, maintain or tear down a connection between source and destination nodes. Additionally, UDP does not guarantee error-free and in-sequence delivery of data between source and destination, which make the protocol unreliable.

TCP is a connection-oriented protocol that is used for reliable transfer of data, that is, it requires establishing, maintaining and tearing down connection between source and destination nodes. TCP has built-in reliability that includes sequence numbering with re-sending, which is used to detect and resend missing or out-of-sequence segments. The protocol also includes a complete flow control mechanism called "sliding window mechanism" in order to prevent any sender from overwhelming a receiver [2][8][12]. The features make TCP a transport protocol of choice for most Internet applications.

Flow control ensures orderly transmission of data while congestion control prevents congestion by controlling amount of data entering the network. Congestion occurs when one or more routers on an Internet path become overloaded. This can occur when core Internet routers receive packets faster than they can forward. Figure 1 (B) illustrates an analogy of the condition that leads to network congestion when incoming rate of packets to a router is greater than the maximum outgoing rate.

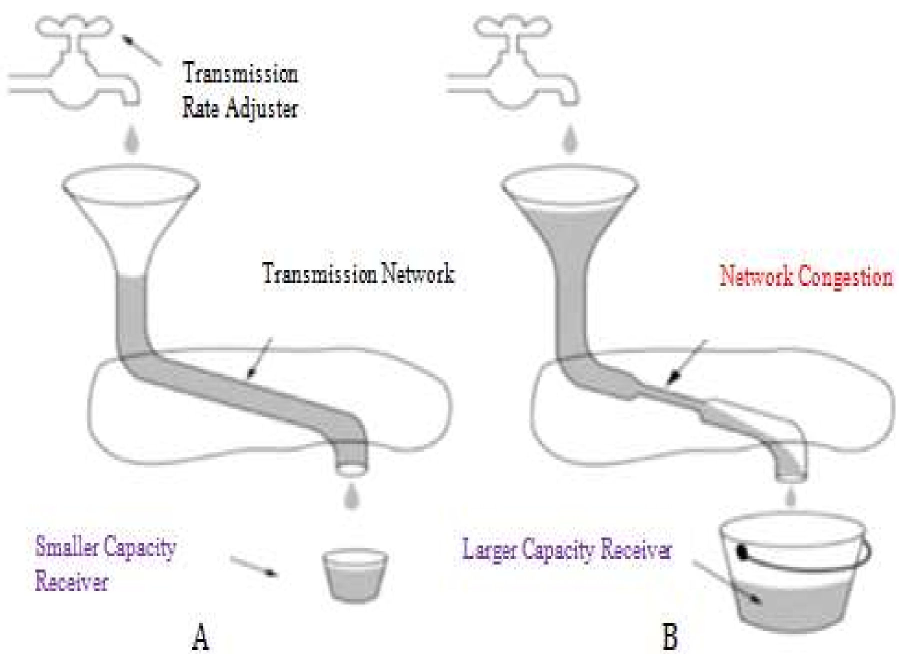


Figure 1: Analogy of Network Congestion

However, neither of these built-in TCP features is good for real-time applications such as audio and video on the Internet. Real-time applications cannot "pause" and wait for missing segments, nor would they slow down or speed up as traffic loads vary on the Internet.

The use of a Bandwidth Delay Product (BDP) sized buffer was the rule-of-thumb for a long period of time. It gives a measure of required buffer size for a particular network based on its average round trip time and link capacity. The rule-of-thumb comes from a desire to keep the bottleneck link as busy as possible, so that the throughput of the network is maximized by providing a buffer size equal to the bandwidth delay product. This buffer size prevents the link from going idle and thereby losing throughput [14] [15].

The rule-of-thumb complicates router buffer design due to the buffer sizes that can be required when the network capacity is large. For example, an 8Gbits/s router line card needs approximately 250ms x 8Gbits/s = 2Gbits of buffer and this will grow linearly with line rate [6][15]. It has been argued in [16][19] that the rule-of-thumb is now outdated and incorrect for backbone routers because of the large number of flows multiplexed together on a single backbone link. New proposals have thus been made to support the use of small and large buffers.

This paper investigates the effects of buffer sizes on network delay and throughput in networks with heterogeneous traffics using Random early detection (RED) Queue management. The objective is to determine a suitable and optimal buffer-size for router that minimize delay and maximize throughput. The rest of this research paper is organized as follows: Section 2 discusses the outcomes of related literatures. Section 3 discusses the method used for buffer sizing. Section 4 explains the experimentation procedure and parameters used for evaluation. Section 5 discusses the results of the experiments conducted and Section 6 concludes the research paper.

2. RELATED WORK

Raina *et al.* in [3] described the application of control theory in addressing the effective way of sizing buffers in core Internet routers. They establish that a network is generally stable or unstable depending on the buffer sizes. And proposed how to choose buffer sizes that will enable network stability with factors that may have effect on the stability, such as AQM parameters, round trip times (RTT), traffic mixes and the TCP congestion avoidance algorithm. Finally, the scheme described how certain changes to TCP's rules for increasing and decreasing window size make the entire network less prone to synchronization respectively.

The proposed Dhamdhere *et al.* in [4], used smaller buffers in router interfaces, less than the links bandwidth-delay product, without causing utilization loss, as long as the link carries many TCP flows. The scheme noted some key issues about the previous schemes and publicized that the use of such small buffers can lead high loss rates in congested access links that carry many flows. Even if the link is fully utilized, small buffers lead to lower throughput for most large TCP flows, and significant variability in the per-flow throughput and transfer latency. The existing schemes proposed that trade-off between loss rate and queuing delay, in terms of application-layer performance, was an important issue in the buffer sizing problem.

However, Enachescu *et al.* [5] examined the widely known rule-of-thumb which stated that a bandwidth-delay product of buffering at each router was required so as not to lose link utilization as this could be too large. They explored how buffers in the backbone can be significantly reduced even more, to as little as a few dozen packets, if willing to sacrifice a small amount of link capacity. The scheme also probed that if the TCP sources are not overly burst, then fewer than twenty packet buffers are sufficient for high throughput. Specifically, they showed using simulations that O(logW) buffers are sufficient, where W is the window size of each flow.

The change needed to be made to TCP was minimal as each sender just needed to pace packet injections from its window. Their main conclusion was that the results obtained suggested that packet buffers can be made as small as 10-20 respectively. While Vu-Brugier *et al.* in [7] used a research outcome that showed when the number of flows is sufficiently large, the buffer size can be decreased to the bandwidth-delay product divided by the square-root of the number of flows (sqrtN discipline), without introducing under-utilization of the link bandwidth. They compared the performances of normal and sqrtN for sizing the router buffer by focusing on the performance of both long-lived and short-lived TCP connections traversing the router under various network environments. The findings demonstrated that the sqrtN discipline would degrade the TCP performance in terms of the packet loss ratio and file transmission delay, and it may be useful only when transferring a file of size 50-100Kbytes. However, Hasegawa *et al.* [9] believed that there had not been a thorough verification of buffer sizing recommendation for short-lived flows, which make up the majority of Internet flows. Furthermore, they observed that the effects of network parameters, such as the link bandwidth and propagation delay, had not yet been investigated.

Hence, they compared the performance of the above two disciplines using simulations and focused on the performance of both long-lived and short-lived TCP connections that traverse the router under various network environments. Their results showed that sqrtN discipline would degrade the TCP performance in terms of the packet loss ratio and file transmission delay, and it may be useful only when the size of the file being transferred is approximately 50–100 Kbytes or when the propagation delay between the sender and the receiver hosts is significantly small.

Similarly, Nicholas and Jacobson in [10] turned towards AQM (Active Queue Management) to find a solution for persistently full buffers. which aimed in providing part of the buffer bloat solution by proposing an innovative approach to AQM that is suitable for today's Internet called Controlled Delay (CoDel). This is a “no-knobs” AQM that adapts to changing link rates and is suitable for deployment and experimentation in Linux-based routers. Unfortunately, AQM is still not widely deployed because of implementation difficulties and general misunderstanding about Internet packet loss and queue dynamics.

Lakshmikantha *et al.* in [11] re-examined the buffer-size requirements of core routers when flows arrive and depart and concluded that (1) if the core-to-access-speed ratio is large, then O(1) buffers are sufficient at the core routers; (2) otherwise, larger buffer sizes do improve the flow-level performance of the users. And their analysis offered two new insights; (1) It may not be appropriate to derive buffer-sizing rules by studying a network with a fixed number of users. In fact, depending upon the core-to-access-speed ratio, the buffer size itself may affect the number of flows in the system, so these two parameters (buffer size and number of flows in the system) should not be treated as independent quantities. (2) If the core-to-access-speed ratio is large then the O(1) buffer sizes are sufficient for good performance and that no loss of utilization results, as believed formerly.

Thus, it is clear there is no single agreement on what the buffer size should be. Hence, the use of small buffer was considered to reduce the network round trip time, which leads to a higher network throughput for each flow with TCP, as the flows will complete faster [17]. Also, Enachescu *et al.* in [18] argued that if the TCP sources are not overly burst, then fewer than twenty packets buffers are sufficient for high throughput. This network of tiny buffers can have buffer size of: $B = O(\log W)$ where W is the window size of each flow.

More so, issues have also been raised on small buffers in the network because they lead to packet loss rate of up to 15% in congested access links that carry many flows, which is an important reason for buffer size [4]. Hence, the need to set suited buffer sizes in order to achieve a robust and optimized QoS.

3. THE PROPOSED BUFFER SIZING SCHEME

This paper analyses buffer size performance and optimization in a network with heterogeneous traffics for Random Early Detection (RED) Queues. The goal is to propose a suitable and better buffer size for heterogeneous network traffics which is an extended version of the existing Optimal Buffer Size for Heterogeneous Traffics on Lossy and Low-Bandwidth Links formally published in International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS) [1], as against the existing scheme of buffer sizes of 5-50kb, which exhibit excessive response delay as a result of left-over buffering of packets with insignificant throughput resulting to Poor Quality of Service (QoS).

The choice of buffers in network routers plays an important role in accommodating packets that arrive in bursts at the routers interface efficiently. The use of small buffer guarantees low packet delay but higher chances of packet lost. Small buffers have the merit of reducing queuing delays, while maintaining full link utilization at the target link; a design that comes with the cost of high loss rate of packets. The use of large buffer results in a buffer bloat which causes high delay in a network. Buffer bloat in networks is the cause for higher latency in packet switching network which is usually caused due to excess buffering of packets. Buffer bloat also results in packet response delay variation (Jitter); as well as minimize the overall throughput of the network. Figure 1(B) is a typical example of buffer bloat with an incoming load greater than it capacity due to the high changing link characteristics of modern heterogeneous network traffic. Therefore, it is desirable to provide a more appropriate Buffer size that will be able to admit traffic aptly.

The existing buffer sizing scheme considers the use of 5kb to 50 kb, Buffer Sizes which increase response delay with insignificant throughput as shown in Tables and Figures. More so, as the buffer size increases from 15-50 an insignificant throughput is achieved with an intolerable response delay of packets incurred at the process, resulting to congestion as well as packet loss (Table 3- 5 and Figure 3-6), which subsequently results to performance degradation of the entire network know as poor QoS. Therefore, we proposed a scheme that uses a buffer size between 5kb to 10kb for the co-router. This is to determine buffer sizes that reduce excessive delay, admitting traffics more appropriately within their life time as well as improved the Quality of Service. The proposed scheme has the potential to significantly improve Quality of Service than the existing scheme and therefore providing a timely user perceived higher network response.

3.1 Experimentation

The capability of the proposed buffer size was evaluated using NS2 simulation experiments. The experiments were conducted using a set of NS2 experiments and realistic models that represent Internet connections at different congestion levels. The simulation-based method is cheaper and faster compared to performing live Internet experiments. The simulation models consist of the novel NS2 models built to closely resemble a real Internet scenario. The network topology, application traffic, and transport model used in the experiments are discussed in sections 4.1 through 4.5 as follows.

Network Topology

A home or an institution network normally has enough bandwidth to carry its own traffic. Similarly, the Internet backbone is generally highly provisioned, though sometimes it can get congested. On the other hand, the access link from the home or institution network to the Internet gateway is usually shared among multiple networks, and can reasonably be assumed to be the main bottleneck for wide-area Internet connections. Therefore, simulations in this work will be performed using the single bottleneck, dumbbell topology illustrated in Figure 2.

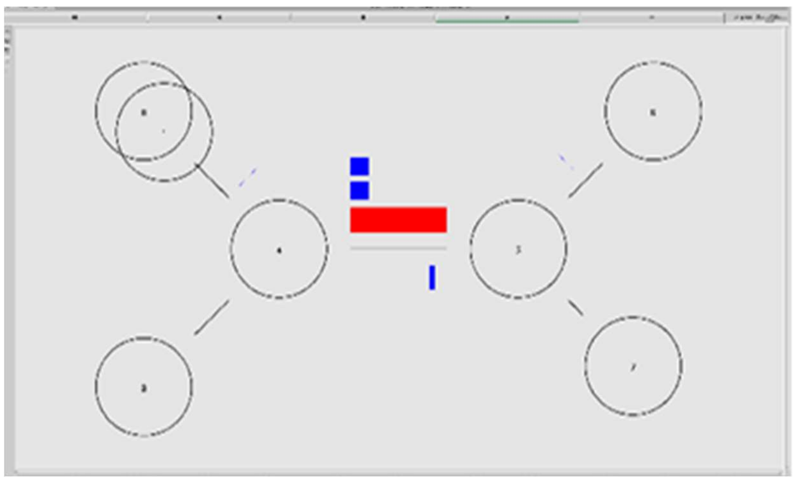


Figure 2: Single Bottleneck Dumbbell Simulation Topology

The topology in Figure 2 was created and used to generate our results. Nodes 0, 2 and 3 are TCP agents; node 1 is a UDP agent. Nodes 4 and 5 are bottleneck routers while nodes 6 and 7 are TCP sink and UDP null agents respectively. Below are processed scripts' results of the simulations run using different bottleneck bandwidths and packet loss rate at all buffer-sizes for the RED queue mechanisms used as our performance Metrics.

The available bandwidth at the end links is set high (i.e. 1Gbps), which causes the 10Mbps Internet service provider (ISP) access link between the routers to be the bottleneck. The main direction of traffic flow is from the left side where HTTP servers are connected to the Internet router, while the traffic destinations are connected to the access router on the right side respectively.

Transport Protocol

TCP New Reno is the transport protocol used for all End-hosts with Selective Acknowledgement (SACK) enabled. However, both Explicit Congestion Notification (ECN) and the Nagle algorithm were disabled. Other TCP parameter settings are listed in Table 1 as follows:

Table 1: TCP Simulation Parameters

#	Parameters	Values
1	TCP Version	New Reno
2	Maximum Segment Size (MSS)	1460 Bytes
3	Initial Congestion Window (IW)	3 MSS
4	Initial Retransmission Timeout (RTO)	1 Second
5	Maximum Receive Window	1000 Packets
6	Segments Per ACK	1
7	SACK	Turned ON
8	Nagle Algorithm	Turned OFF
9	ECN	Turned OFF
10	TSL Startup	Turned ON
11	IW after SYN Loss	1 / 3
12	SSThresh after SYN Loss	2 / 16 / 1000

The TCP latency is calculated as the average response time of individual web object requests. The NS2 code was changed to enable TSL startup algorithm i.e. choosing different values of IW and ssthresh upon SYN-ACK loss during the three ways Handshake (3WHS).

3.2 Performance Metric

The performance metrics play a critical role in determining the outcome of simulations. The metrics used for the evaluation are bandwidth bottleneck, packet loss rate, buffer size, latency and throughput respectively.

Bandwidth Bottleneck.

Bandwidth is the rate at which data can be forwarded between a sender and receiver over the network (capacity of a link). Sending data involves forwarding the data along an end-to-end chain of networking elements, the slowest element in the entire chain sets the bottleneck bandwidth, i.e., the maximum rate at which data can be sent along the chain. The bottleneck link is a network link with a limited bandwidth. Our analysis is restricted to an assessment of the bottleneck bandwidth as an end-to-end path property, rather than as the property of a particular element in the path.However, it is crucial to distinguish between bottleneck bandwidth and available bandwidth. The former gives an upper bound on how fast a connection can possibly transmit data, while the latter denotes how fast the connection can transmit data, or in some cases how fast it should transmit data to preserve network stability, even though it could transmit faster.For connection performance, bottleneck bandwidth is a fundamental quantity, because it indicates a limit on what the connection can hope to achieve. If the sender tries to transmit any faster, not only is it guaranteed to fail, but the additional traffic it generates in doing so will either lead to queuing delays somewhere in the network, or packet drops, if the overloaded element lacks sufficient buffer capacity.

Packet Loss Rate (PLR).

Packet loss is said to have occurred when one or more packets of data drops or fail to reach their destination in a network. Packet loss rate is the fraction of the total packets transmitted that did not reach the destination. Packet loss is usually caused by congestion, interference or errors in data transmission across the network (e.g. wired, wireless, mobile).

It is measured as a percentage of packets lost with respect to the packets sent. The effects of packet loss vary depending on the protocol/application concerned. Generally, TCP detects packet loss and performs retransmissions to ensure reliable, in order data transmission. Packet loss in a TCP connection is also used to avoid congestion and thus produces an intentionally reduced throughput for the connection. While UDP, does not have any inbuilt re-transmission capability and may not handle packet loss as well. However, irrespective of the protocol/application, too much loss of packets is definitely a problem.

Buffer Size.
Router uses buffer as memory space to handle data during the network routing processes. As data flows through a network, different rates of transmission occur between routers and network transport, which can create network congestion. Thus, the need to store temporarily packets to address bursts during data transmissions to compensate for variations in speed. The buffer is a crucial factor and plays an important role in the transmission of data over the network.

Latency.
This is the time interval between the request and response of data transfer (how long it takes data to travel between a source and destination). It is also known as delay and it is measured in milliseconds (ms).

Throughput.
This is the rate of data that was transferred over the network successfully from a source to its destination in a specific period. It is measured in either bits per second (bps), Megabits per second (Mbps) or Gigabits per second (Gbps).

3.3 Evaluation Tools
The simulation experiments were conducted using Tool Command Language (TCL), Network Animator (NAM), AWK and Gnuplot.

- TCL: A toolkit for building a graphical user interface which is dependent on the Network Simulator. It generates the trace files "out.nam", "out.tr", "winfile" as created in the TCL script. Examples as shown in figures below.
- NAM: A TCL based animation tool for viewing network simulation traces and real-world packet trace.
- AWK: named after Aho, Weinberger and Kernighan is a programming language that is used to filter or analyze trace files (i.e. text processing application for data analysis). This script consists of American Standard Code for Information Interchange (ASCII) codes with 12 fields organized in the following order:
- Gnuplot: is a command driven interactive function plotting program which can generate 2D or 3D plots of data (i.e. visualize trace files) and one of the most widely used on Linux platforms.

3.4 Evaluation Procedure
A number of simulation experiments were conducted using different parameter(s) to determine averages in throughput, delay and packet lost. The experiments were conducted with buffer sizes of 5 to 50 and 5 to 10 kilobytes for the existing and proposed schemes respectively. For each buffer size, packet loss rates of 1%, 2% and 5% and bottleneck links of 0.05MB, 0.1MB and 0.5MB were used as shown in Table 2.

Table 2: Simulation Experiments

Experiment	Packet Loss Rate	Bottleneck Link
1	1%	0.05MB
2	1%	0.10MB
3	1%	0.50MB
4	2%	0.05MB
5	2%	0.10MB
6	2%	0.50MB
7	5%	0.05MB
8	5%	0.10MB
9	1%	0.50MB

4. RESULTS

This section discusses the results of the evaluation conducted in order to determine the optimal buffer-size of router in heterogeneous networks. Tables 3 to 5 show the performances of the existing and proposed buffer sizes using the different packet loss rates and bottleneck Links specified in Section 3.5. The bottleneck link used is that of a lossy and low bandwidth of 0.1MB, 0.5MB and 0.05MB. Generally, the results in Tables 3 to 5 showed that the existing scheme has higher response delay with insignificant throughput than the proposed scheme. The results also showed that bottleneck links is proportional to throughput and inversely proportional to the TCP delay, that is, increasing the bottleneck link enhances the throughput and reduces the delay. Also, increase in the packet loss rate decreases the delay and throughput, particularly with the bottleneck links of 0.1MB and 0.5MB. Thus, increasing the bottleneck link in a network will improve its performance, and packet loss has a great effect on both throughput and delay, i.e. it reduces the quality of the overall performance).

Table 3: RED Queue at 1% Packet Loss Rate for Existing and Proposed buffer sizes

Existing Buffer Size				**Proposed Buffer Size**			
Buffer Size (kb)	TCP Delay (ms)	TCP Throughput (kbps)	Packet Lost	Buffer Size (kb)	TCP Delay (ms)	TCP Throughput (kbps)	Packet Lost
Bottleneck Link = 0.05MB							
5	379.3780	27.7542	3	5	354.2800	25.7539	2
10	712.8960	28.7432	17	6	417.7160	26.9682	5
15	748.3230	29.0030	13	7	496.1490	27.7549	7
30	721.8910	28.5293	12	8	578.8850	27.9907	5
50	721.8910	28.5293	12	9	603.4710	27.5866	9
				10	675.6530	27.9576	14

Bottleneck Link = 0.1MB							
5	160.0370	72.4086	1	5	160.4990	70.5213	3
10	301.1310	73.8220	10	6	193.9530	72.4657	4
15	325.8030	75.2564	11	7	227.6620	72.8275	4
30	336.0010	74.8774	8	8	265.7220	73.0628	4
50	336.0010	74.8774	8	9	265.0440	72.7522	5
				10	283.6580	71.3055	3
Bottleneck Link = 0.5MB							
5	43.8061	371.4270	7	5	42.9402	329.3970	7
10	58.2831	407.9680	9	6	47.1522	334.7500	11
15	63.3066	402.7640	5	7	47.4652	344.1870	8
30	62.6055	402.3050	0	8	49.9440	349.6760	5
50	62.6055	402.3050	0	9	52.0467	356,8990	5
				10	55.6323	369.1510	7

Table 4: RED Queue at 2% Packet Loss Rate for Small and Proposed Buffer Sizes

Existing Buffer Size				**Proposed Buffer Size**			
Buffer Size (kb)	TCP Delay (ms)	TCP Throughput (kbps)	Packet Lost	Buffer Size (kb)	TCP Delay (ms)	TCP Throughput (kbps)	Packet Lost
Bottleneck Link = 0.05MB							
5	374.4890	28.5175	12	5	346.8730	26.0388	10
10	630.0570	29.1356	19	6	408.2150	26.1400	8
15	644.5670	28.5363	10	7	468.2200	27.5074	7
30	644.5670	28.5363	10	8	507.8580	26.1490	5
50	644.5670	28.5363	10	9	560.1580	27.0998	15
				10	580.2610	26.1874	10

Bottleneck Link = 0.1MB							
5	154.5070	70.0270	6	5	155.4900	70.2230	8
10	255.7780	71.2884	9	6	176.4140	67.9717	4
15	274.7870	72.4131	10	7	209.9610	71.3704	7
30	274.7870	72.4131	10	8	237.9730	71.9849	10
50	274.7870	72.4131	10	9	237.9730	71.9849	10
				10	250.8000	66.7754	11
Bottleneck Link = 0.5MB							
5	41.0522	310.9810	11	5	41.0990	275.9120	12
10	48.9664	323.5170	7	6	42.9516	292.2550	5
15	50.0414	325.8210	10	7	43.0229	300.7090	10
30	50.0414	325.8210	10	8	44.8172	294.4260	3
50	50.0414	325.8210	10	9	46.2171	304.4290	10
				10	47.9351	306.3880	10

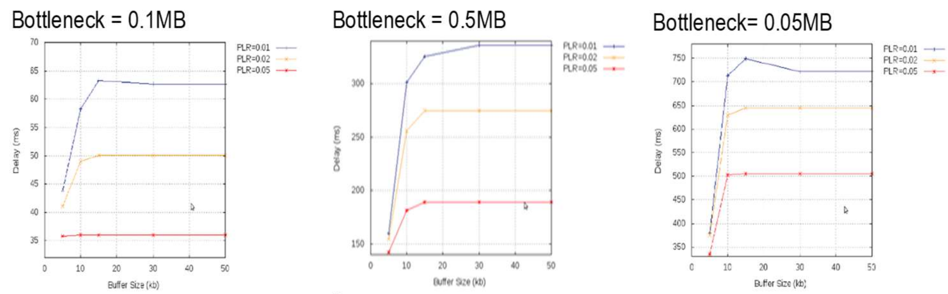
Table 5: RED Queue at 5% Packet Loss Rate for Existing and proposed buffer sizes

Existing Buffer Size				**Proposed Buffer Size**			
Buffer Size (kb)	TCP Delay (ms)	TCP Throughput (kbps)	Packet Lost	Buffer Size (kb)	TCP Delay (ms)	TCP Throughput (kbps)	Packet Lost
Bottleneck Link = 0.05MB							
5	335.4730	26.7971	17	5	327.6750	24.8665	14
10	502.6970	29.9532	17	6	368.1560	25.6647	18
15	505.7370	28.5635	15	7	405.6310	26.4920	23
30	505.7370	28.5635	15	8	441.1800	26.3780	18
50	505.7370	28.5635	15	9	471.8980	26.5292	22
				10	462.2820	27.2210	12

Bottleneck Link = 0.1MB							
5	142.4140	67.6430	17	5	139.1670	61.4775	15
10	181.5740	66.4090	9	6	160.4750	63.4623	17
15	188.9090	66.7687	14	7	173.3390	57.7602	13
30	188.9090	66.7687	14	8	175.3640	64.1372	15
50	188.9090	66.7687	14	9	175.3640	65.1372	15
				10	184.8670	63.2992	14
Bottleneck Link = 0.5MB							
5	35.7556	203.3020	17	5	35.5351	182.7040	10
10	35.9693	203.3020	17	6	35.5351	182.7040	10
15	35.9693	203.3020	17	7	35.5351	182.7040	10
30	35.9693	203.3020	17	8	35.5351	182.7040	10
50	35.9693	203.3020	17	9	35.5351	182.7040	10
				10	35.5351	182.7040	10

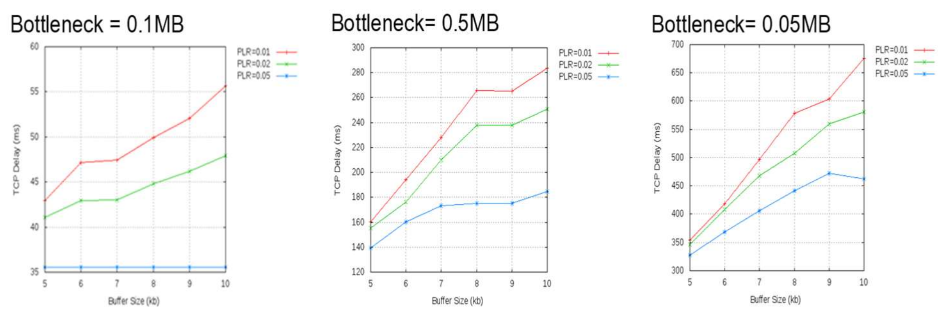
4.1 Buffer Size vs TCP Delay

The effects of varying buffer size on TCP delay for the existing and proposed schemes are shown in Figure 3 to Figure 6 respectively. For the existing scheme, the results in Figure 3 showed that the TCP delay mainly increased with buffer sizes of up to 15KB except when a bottleneck of 0.5MB (Figure 3b) was used, which increased the delay with buffer sizes of up to 30KB. The bottleneck links has substantial effect in reducing the TCP delay, that is, the larger the bottleneck link, the lower the delay observed. The results of the proposed scheme in Figure 4 also exhibit similar behavior with the existing scheme. However, the amount of TCP delay observed with the proposed scheme of using buffer sizes of 5KB to 10KB comparatively reduced depending on the bottleneck link used and packet loss rate. This shows that buffer size greater than 15kb does not reduced the TCP delay and thus using the 5kb to 10kb is better. Thus, the results showed that using a buffer size not greater than 10kb along with large bandwidth bottleneck link reduces the TCP delay.



a B c

Figure 3: **The Results of TCP Delay against Buffer-Size for the Existing Scheme**

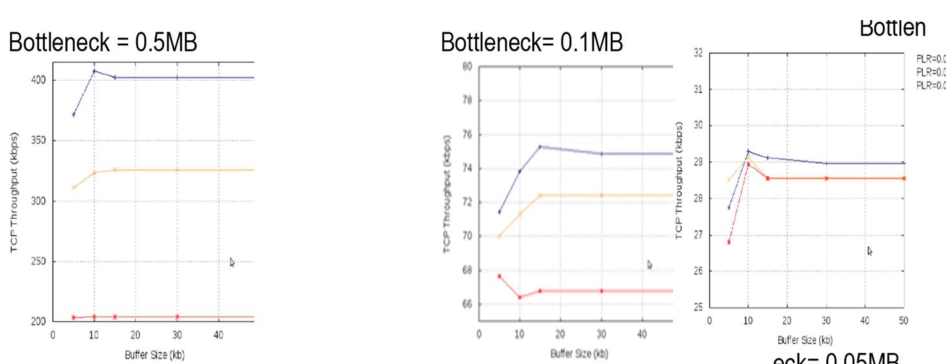


a b c

Figure 4: The Results of Delay against Buffer-Size for the Proposed Scheme

4.2 Buffer Size vs Throughput

The effects of varying buffer size on throughput for the existing and proposed schemes are shown in Figure 5 and Figure 6 respectively. For the existing scheme, the results in Figure 5 showed that the throughput mainly increased with buffer sizes of up to 10kb (Figure 5a and 5c) or 15kb (Figure 5b) depending on the bandwidth bottleneck and packet loss rate. Also, the throughput increased significantly with larger bandwidth bottleneck.



a b c

Figure 5: The Results of Throughput against Buffer-Size for the Existing Scheme

For the proposed scheme in Figure 6, the throughput increased or decreased with varying buffer sizes. Similar to the existing scheme, the throughput increased significantly with larger bandwidth bottleneck. Generally, the throughput achieved with the proposed scheme is almost the same with existing scheme. This shows that increasing the buffer size greater than 10kb do not yield additional throughput. Thus, the results showed that using a buffer size not greater than 10kb along with large bandwidth bottleneck link increases the throughput.

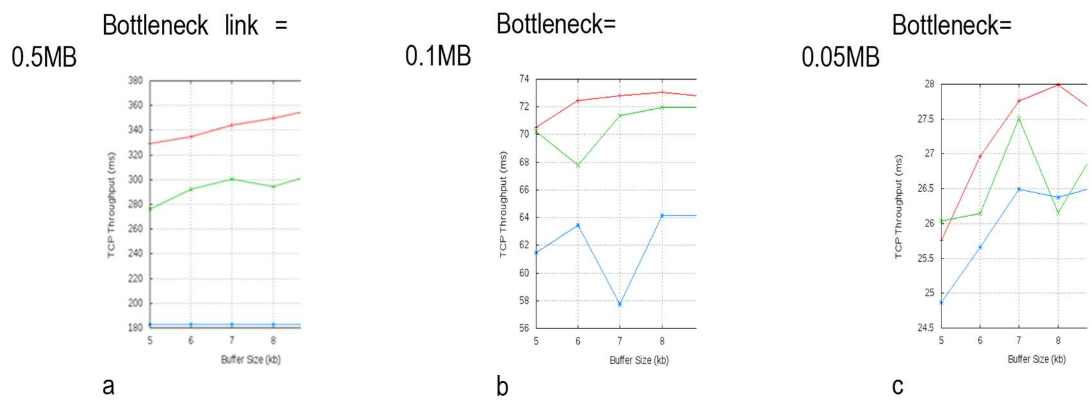


Figure 6: The Results of Throughput against Buffer-Size for the Proposed Scheme

5. CONCLUSION

The choice of a suitable buffer size for routers has been an open area of research that has gained considerable interest. The widely believed rule-of-thumb was challenged by this research study. The findings of the study indicated that under certain circumstances, as few as 5-10 Buffering Size is sufficient to get better QoS. This study is primarily TCP centric, since over 95% of today's Internet traffic is carried by TCP while UDP traffic accounts for about 5% and is used by multimedia applications such as audio/video whose adoption is growing in the Internet. Traffic patterns change significantly (i.e. with respect to buffer sizing requirements) over time. Buffer size is essential to distinguish between links at which TCP flows experience packet loss and those where they do not for RED Queues.

An under-provisioned links may experience essentially more queuing and generates essentially more packet loss. As such, the choice of buffer size has huge impact on overall Quality of Service. Hence, this paper proposes a Performance Analysis and Optimization of Buffer Sizes for Heterogeneous Network Traffics for RED Queues. The performance analysis and the NS2 Simulation results of the proposed scheme proved that the optimal value for setting a suitable buffer-size conveniently for Internet router just in time is between 5-10 Buffer Size. Shown in Table 3-5 and Figure 3-6 respectively. Similarly, the results in Tables 3 to 5 also showed that the existing scheme has higher delay with insignificant throughput than the proposed scheme. Hence, our proposed scheme has better performance with an Optimized QoS.

REFERENCES

[1] H. A Khalid, D.D Wisdom, M. A Ahmad, S Isaac, H Idris, and S.M. Karma (2020). Optimal Buffer Size for Heterogeneous Traffics on Lossy and Low-Bandwidth Links, *International Conference in Mathmatics, Computer Engineering and Computer Science (ICMCECS)* Anchor University Lagos, Nigeria (21, March 2020).

[2] B Kaur, M Kaur, P Mudgil and H Singh (2013). Importance of Sliding Window Protocol, *International Journal of Research in Engineering and Technology (IJRET*), Volume 2, Issue 10, (2013), PP. 2321-7308.

[3] G Raina, D Towsley and D Wischik (2012). Part II. *ACM SIGCOMM Computer Communication Review*, Volume 35, Issue3, (2012), PP. 79. doi: 10.1145/1070873.1070885.

[4] A Dhamdhere, J Hao and C Dovrolis (2005). Buffer Sizing for congested internet links, *Proceedings of the IEEE 24Th Annual Joint Conference of the IEEE Computer and Communications Societies*. doi:10.1109/infcom.(2005), PP.1498335.

[5] M Enachescu, Y Ganjali, A Goel, N Mc-Keown and T Rough-garden (2015). Part III: Routers with Very small buffers, *ACM SIGCOMM Computer Communication Review*, Volume 35, Issue 3, (2015) ,PP. 83.

[6] R Prasad, C Dovrolis and M Thottan (2009). Router Buffer Sizing for TCP Traffic and the Role of the Output/Input Capacity Ratio. *IEEE/ACM Transactions on Networking*, Volume 17, Issue 5, (2009), PP. 1645-1658. doi: 10.1109/tnet.2009.2014686

[7] G Vu-Brugier, R Stanojevic, D Leith, and R Shorten (2007). A critique of recently proposed buffer-sizing Strategies, *ACM SIGCOMM Computer Communication Review,* Volume 37, Issue 1, *(*2007), PP. 43.

[8] N Beheshti, Y Ganjali, M Ghobadi, N McKeown and G Salmon (2008). Experimental study of router buffer Sizing. *Proceedings of the 8Th ACM SIGCOMM Conference on Internet Measurement Conference – IMC* (2008).

[9] G Hasegawa, T Tomioka, K Tada and M Murata (2008).Simulation studies on router buffer sizing for short-Lived and pacing TCP flows, *Computer Communications,* Volume 31, Issue 16, (2008), PP. 3789-3798.

[10] K Nichols and V Jacobson (2012). Controlling Queue Delay, *ACM Digital Library*, Volume 10, Issue 5, (May 6, 2012) PP. 20.

[11] A Lakshmikantha C Beck and R Srikant (2011). Impact of File Arrivals and Departures on Buffer Sizing in Core Routers, *IEEE/ACM Transactions on Networking*, Volume 19, Issue 2, (March 3, 2011), PP. 347-358. doi:10.1109/tnet.2011.2114365.

[12] S Ali, S Nasir and S Qazi (2013). Impact of router buffer size on TCP/UDP performance, *3rd IEEE, 2013. International Conference on Computer, Control and Communication* (IC4)(2013).

[13] B M Bwalya and S Tembo (2017).Performance evaluation of buffer size for access networks in first generation Optical networks, *International Journal of Internet of Things*, Volume 6, Issue 3,(3 June, 2017) doi:10[4].5923/j.ijit.20170603.02

[14] I Khan and M Hassan (2016). Transport Layer Protocols and Services, *International Journal of Research in Computer and Communication Technology*, Volume 5, Issue 9 (October, 2016), pp. 2320-5156.

[15] G Patil, S McClean and G Raina (2011). Drop Tail and RED queue management with small buffers: Stability and hope bifurcation, ICTACT Journal on Communication Technology: *Special Issue on Next Generation Wireless Networks and Applications*, Volume I2, Issue 2, (June 2011).

[16] G Appenzeller, I Keslassy and N Mc-keown (2004). Sizing router buffers, *In SIGCOMM 2004:* Proceedings of the 2004 conference on Applications, technology and architecture and protocols for *Computer communications, pp.* 281-292, New York, NY, (2004), ACM Press.

[17] E. Roberts (2003). Internet2Architecture [Online]. Available: https://cs.stanford.edu/people/eroberts/courses/soco/projects/2003-04/internet-2/architecture.html.

[18] M Enachescu, Y Ganjali, A Goel, N Mc-Keown and T Roughgarden (2006).Part III; Routers with very small buffers, *ACM SIGCOMM Computer Communication Review,* Volume 35, (2006), PP. 83-90.

[19] K Avrachenkov, U Ayesta, E Altman, P Nain and C Barakat (2002). The effect of router buffer size on TCP Performance*, in the proceedings of LINIIS workshop,* St. Petersburg, (Jan 29-Feb 1 2002).