INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING & COMPUTER SYSTEMS (IJSECS)
ISSN: 2289-8522 e-ISSN: 2180-0650
VOL. 6, ISSUE 2, 33 – 42
DOI: https://doi.org/10.15282/ijsecs. 6.2.2020.5.0073

ORIGINAL ARTICLE

Performance Analysis and Optimization of Buffer Sizes for Heterogeneous Network Traffics

M.A. Ahmad*, D. D. Wisdom[1], S. Isaac[2], and U.C. Arinze[3]

*Department of Computer Science, Kaduna State University, Kaduna, Nigeria
[1]Department of Mathematics, Computer Unit, Usmanu Danfodiyo University Sokoto (UDUS), Nigeria
[2]Department of Computer Science, Kaduna State University, Kaduna, Nigeria
[3]Department of Computer Science, Faculty of Physical Sciences, University of Nigeria, Nsukka (UNN), Enugu, Nigeria

ABSTRACT – The digital era have paved a way for a frequent demand for efficiency of wireless telecommunication network due to it increase economic and societal benefits. Transmission control protocol (TCP) and User datagram Protocol (UDP) are the basic protocol responsible for information transfer from end to end. However, the efficiency to which messages travels across the network, and the time taken to deliver is a key concern. Hence throughput and latency becomes a challenge. Since, selecting an appropriate buffer size is still a major challenge. And Smaller size of buffers results to lower response delay at the expense of higher probability of loss rate. Similarly, bigger size of buffers led to buffer bloat with an excessive delay incurred due to thier sizes. Thus, we have conducted a performance analysis using NS2 simulator for Heterogeneous Network Traffics in order to determine an appropriate buffer size with superior average throughput as well as response delay. In the proposed study buffer sizes of five (5) to Ten (10) where selected; and the results proved that buffer sizes of 5 to 10 KB achieves better throughput and delay at the congested router. In addition, the proposed results presented also publicized that buffer sizes have been optimized accordingly.

ARTICLE HISTORY
Revised: 22 June 2020
Accepted: 1 Sept 2020

KEYWORDS
Buffer Sizes
Delay
Heterogeneous-Traffics
Throughput

Introduction

Wireless telecommunication networks employ standard protocol suites known as, Transmission Control Protocol (TCP) as well as User Datagram Protocol (UDP) for transferring packets from end to end. However, TCP is the most widely known reliable protocol for the transmission of packets as loss-sensitive data packets as Electronic-commerce (E-commerce) as well as Emails. While UDP is mainly used for forwarding of datagrams transmission more rapid of non-delay-sensitive packets the protocol is also known as delay tolerance protocol. This implies that, when using applications as real-time voice and video applications, UDP protocol is faster in terms of delivery of data packets; but with less concern about response delay as well as packet loss.

TCP has built-in reliability structures that encompass sequence number as well as re-sending order of segments; that enables detecting and resend missing/out-of-order segment(s) [3]. More-so, TCP involves flow control algorithm known as sliding window technique that prevents the sender from crushing a TCP receiver with overloaded segments; hence segments are sent based on three-way handshake (3WHS) agreements [7][9][15]. However, known of the incorporated TCP features is good for real-time services such as audio as well as video on the network. The Real-time services do not suspend and wait for missing data packets; nor slow down/speed up traffics arrival on the network.

Essentially, when using protocol services as UDP. Thus, no need establishing or slit down end to end connection. Therefore, UDP is less guarantee of error-free as well as in-order delivery of packets from source to destination.

TCPs port identifier is usually used to recognize distinct request data packets. The TCP ports are built with the host network address in order to generate an opening [2]. Normally, a TCP internet link is specified by the couple of openings at the end, which is used to forward data packets as full-duplex [1]. The standard suites both begin in a closed state when there is no link and forward data packet (s) in sequence ordering when the link is established using 3WHS, open agreed segments size transmission link by the sliding window protocol [8] precisely shown in Figure 1 as follows:

*CORRESPONDING AUTHOR | M. A. Ahmad | danieldaudawisdom1@gmail.com

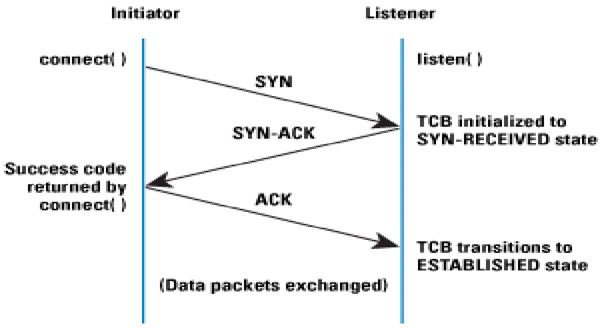


Figure 1: TCP 3WHS.

A TCP scheme for rapid web response over lossy as well as response delay network path is proposed [8]. The standard TCP is the most used protocol due to its reliability and in-order delivery of segments. The protocol has a built-in flow control algorithm that enables sequencing of data packets transmission with an efficient congestion controller in order to minimize the rate of data packets congestion on the network. Congestion occurs on the network, whenever the network load is greater than its capacity [1].

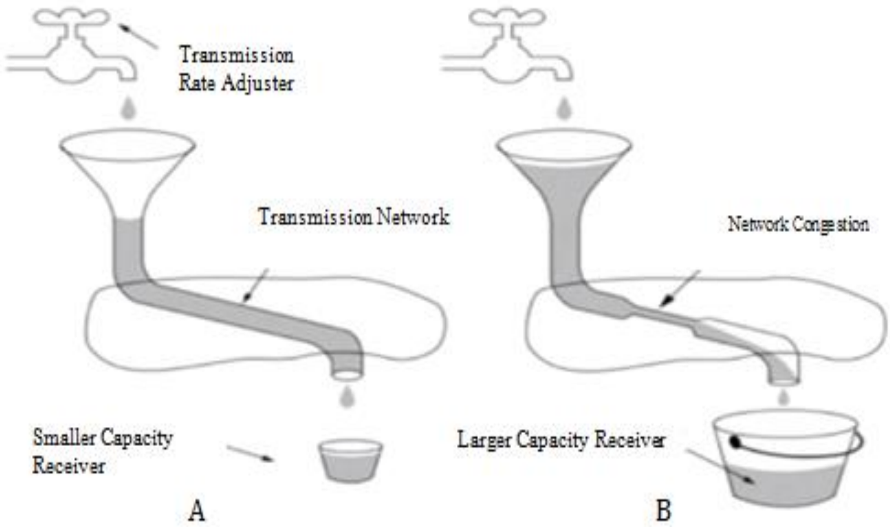


Figure 2: Illustrates network congestion. It can be observed that in Figure 2 (B) the likely case of congestion occurring and re-occurring due to the network load being greater than the capacity.

This paper proposes a suitable buffer size for heterogeneous network traffic, and the paper seeks to address the issues of buffer sizes for heterogeneous networks. In the study, a buffer size of 5 to 10KB is proposed. Since, Buffer sizes is an important concern in TCP flow control, in order to differentiate links that TCP flows experience packet loss rate (PLR) as well as where they do not. An under-provisioned internet network may experience much-queuing delay producing much packet loss rate. Thus, the choice of buffer size has great importance on the overall network performance. Therefore, this paper conducts a performance analysis with an NS2 simulations study of multi-flow traffics run in order to obtain an efficient buffer size that results in achieving better throughput with good response time without additional signalling issues. The rest of this paper is structured as follows: Section 2. related work, Section 3. Simulation-Based Method, Section 4. and Section 5 concludes the study.

Related Work

[14] Propose an initial window (IW) strategy such that, instead of defining a value for the IW, the scheme used automatic algorithm increasing the IW. This results in a safer deployment as well as tried to avoid the repeated reconsidering of the IW size over time. But with evident drawbacks whenever endpoints changed their network path or use bandwidth-on-demand strategy since the paths feature changes over time with much shorter intervals [11]. Propose TCP fast open scheme, and the scheme enables data packets forwarding in SYN-segment form and consume receiver sides during connection establishment of the 3WHS. Which enables saving of about one RTT as related to normal TCP. However, the scheme is restricted to services as less delay tolerance; more-so the proposed scheme has no initial sequence number protection, which makes initial data packets vulnerable. While [13], Propose a TCP congestion control scheme for rate-limited systems that transfer packets in bursts with no full utilization of their rates. The scheme proposes a new-(CWV), a technique that enables a TCP link to restart rapidly from an idle/application-limited intervals. Similarly, [2] proposed a technique to improve TCP performance over wireless links. The scheme suggested an increase on the IW

following simulations and experiments to investigate the impact of the larger IW on short-lived TCP connections and other TCP connections sharing the path. They conclude that increasing the TCP IW to 3 MSS may help to improve perceived performance, respectively. And likewise [5], proposed an increase of TCP IW from three segments to ten segments (about 15 KB) this research calls for starting data transmission with a larger initial congestion window (IW) of about ten segments. TCP slow start algorithm usually indicates an IW of one as well as the two segments. Initial trials demonstrate the profits in decreasing object transmission times at modest cost in terms of increased congestion as well as data loss.

[12], defined how to control scheme had been used to solve concerns of buffer sizes in core internet routers. The scheme tries to forecast when and the extent which synchronization occurs. The scheme discovers a range of time a network is stable for certain buffer size as well as unstable for others. Its tries to explain how to appropriately select buffer size to a minimum stable level, and examine the factors that affect stability namely: AQM parameters, RTT, traffic mixes, as well as TCP Congestion Avoidance Algorithms (CAA). While [4], Proposed that buffer of router interface, should be made smaller, less than the link bandwidth-delay products, with no loss operation, so long the TCP link contain many flows of the TCP. Though, the scheme outstretched some challenges with previous commendations as well as stated use of smaller buffer may result in a higher loss rate of about 5-15% in congested links that forwards much TCP flows even if the link is fully used. More-so, smaller buffers resulted in low throughput for many larger TCP flows. Finally, the proposed schemes examine the trade-off for loss rate as well as response delay.

[6], probed the well-known rule-of-thumb which specified that a bandwidth-delay-product of buffer at every router (s) was essential so that link utilization is not lost as it may be excessively larger. The scheme stated that buffers in the supporting side might be minimized much more, as less as few data packets, when desirous to sacrifice a smaller amount of link size. The proposed scheme contended that when the TCP sources are not seriously in burst size, then few than twenty packet buffers are enough for a higher throughput to be achieved.

[10], the proposed scheme employed two technique. Firstly, the scheme measured the issue of non-persistent TCP flows with heavy-tailed size supply, as well as focused on the impact of buffers size on TCP network performance. With the aim of examining the proposed scheme buffers size, that enhances the TCP throughput with the aid of test-bed experiment such as simulation as well as performance analysis. The proposed scheme discovered that output/input sizes ratios at an internet link mainly determine the essential buffer sizes. The proposed scheme probed that it was needful to further revisit the current arguments on smaller or larger buffer sizes from a better approach, respectively..

SIMULATION-BASED METHOD

Performance analysis and optimization of Buffer Sizes for Heterogeneous Network Traffics on Lossy and Low-Bandwidth Link scheme is proposed based on an extensive NS2 simulation study. The proposed study approach was due to the easy access of the NS2 simulator as well as cheap cost with efficient performance results depicted in Figures 5-10 as follows:

A Network Topology

In the proposed scheme, NS2 simulation was conducted using a single bottleneck dumbbell topology as shown in Figure 3 as follows:

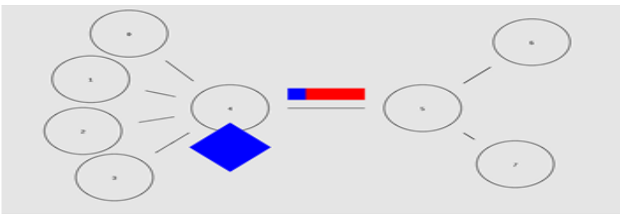


Figure 3: Single Bottleneck Dumbbell Simulation Topology

In this paper, the bandwidth at the point links was set higher to about (1Gbps), enabling the 1Mbps ISP access links between routers and bottleneck. With traffics flow from left (HTTP) server connecting the network routers. Such that traffic destination flows when connected to the access routers on the right hand side respectively.

B Transport Protocol

In this research, New Reno TCP was used for all receiving-hosts with Selective Acknowledgement (SACK) enabled. In the proposed scheme, Explicit Congestion Notification (ECN), as well as the Nagle algorithm, were deactivated. The latency was considered as average response delay of discrete web object requests. The NS2 code was changed to allow the TSL startup algorithm to select different IW values as well as ssthresh upon SYN-ACK loss during 3WHS connection establishment.

Table 1: TCP Simulation Parameters

Parameters	Value
TCP Version	New Reno
Maximum Segment Size (MSS)	1460 Bytes
Initial Congestion Window (IW)	3 MSS
Initial Retransmission Timeout (RTO)	1 Second
Maximum Receive Window	1000 Packets
Segments Per ACK	1
SACK	Turned ON
Nagle Algorithm	Turned OFF
ECN	Turned OFF
TSL Startup	Turned ON
IW after SYN Loss	1 / 3
SSThresh after SYN Loss	2 / 16 / 1000

Performance Metrics

The performance metrics for this research are namely: Bottleneck Bandwidth, Packet Loss Rate, Buffer size, Latency and throughput.

Bandwidths

Bandwidth is the rate of data forwarded from ends to end across the network. Bandwidth is also known as the link capacity of a network (speed of the network). At the same time, the bottleneck link is the internet link with enforced bandwidth from end to end network path. Bottleneck bandwidth is how fast a data packet can travel across the network known as the network speed. While available bandwidth is how faster the network connection is able to transfer data packets from end to end. This is important especially, in cases where a host A is to send messages to Host B, and host A keeps sending data packets faster to B unnecessarily, this may result in buffer bloat, subsequently resulting from packeting loss on the network with poor Quality of service (QoS).

Packets Loss Rate

When one or more data packets fail to arrive at the intended destination host, it is said that data packet loss (PL) has occurred on the network, which usually occurs due to network congestion. While data packet loss rate is parts of the data sent across a network that did not deliver to its host destination. However, in the case of TCP, which loss occurs, a TCP inbuilt algorithm is able to retransmit loss data segments due to the inbuilt reliability features enabled in the TCP protocol. While in the case of UDP no such inbuilt retransmission ability, ones a segment is a loss it may never be retransmitted. However, the increase in packet loss is a critical concern in networks.

Buffer sizes

The network router uses buffers space to control data packets at the transmission time. And as the data packets flow across the network, the need to minimize possible issues of network congestion is key. Hence, data packets are stored temporarily in order to mitigate traffic arrival in a bursty form compensating the well-known problems of data packets variation speed on the network.

Latency

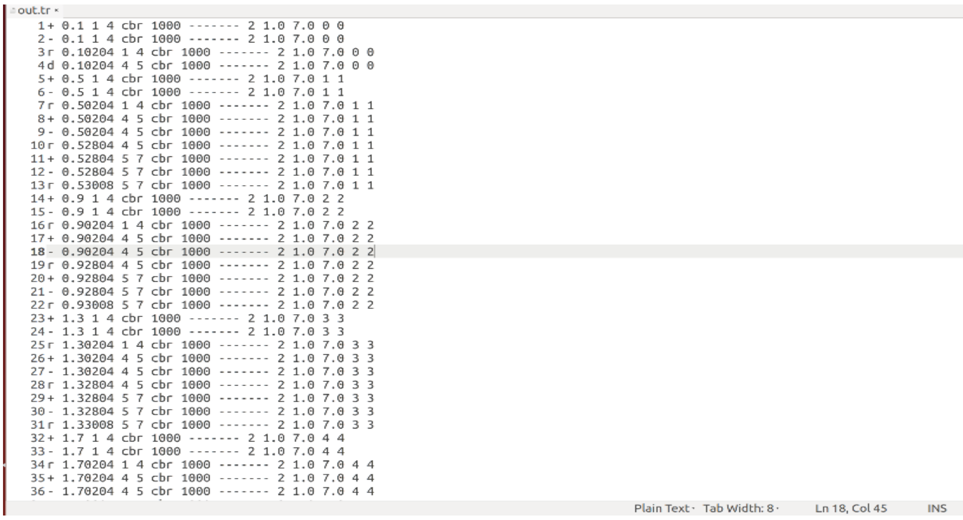
The time taken for data packets to arrive at its intended destination is known as latency. It is usually called the network response delay measured in milliseconds (ms).

Throughput

Is the rate of data packet transfer from end to end within an enforced time interval? Measured in bit per second (Bps), Megabits per second (Mbps) as well as in Gigabits per second (Gbps)

Simulation Results

In the proposed Performance Analysis and Optimization of Buffer Sizes for Heterogeneous Network Traffics on Lossy and Low-Bandwidth Links scheme, we conducted an extensive simulation study changing as well as setting parameters values, such as namely: packet loss rate (PLR=1%,) bandwidth-delay product (BW=0.05mb) for all the buffer-sizes respectively; in order to obtain the following average throughputs, response delay, as well as data packet loss. The experiment was done using drop-tail queue mechanisms to observe their individual differences given in Figures 4-10 and Tables 3-5 as follows:




```
1 + 0.1 1 4 cbr 1000 ------- 2 1.0 7.0 0 0
2 - 0.1 1 4 cbr 1000 ------- 2 1.0 7.0 0 0
3 r 0.10204 1 4 cbr 1000 ------- 2 1.0 7.0 0 0
4 d 0.10204 4 5 cbr 1000 ------- 2 1.0 7.0 0 0
5 + 0.5 1 4 cbr 1000 ------- 2 1.0 7.0 1 1
6 - 0.5 1 4 cbr 1000 ------- 2 1.0 7.0 1 1
7 r 0.50204 1 4 cbr 1000 ------- 2 1.0 7.0 1 1
8 + 0.50204 4 5 cbr 1000 ------- 2 1.0 7.0 1 1
9 - 0.50204 4 5 cbr 1000 ------- 2 1.0 7.0 1 1
10 r 0.52804 4 5 cbr 1000 ------- 2 1.0 7.0 1 1
11 + 0.52804 5 7 cbr 1000 ------- 2 1.0 7.0 1 1
12 - 0.52804 5 7 cbr 1000 ------- 2 1.0 7.0 1 1
13 r 0.53008 5 7 cbr 1000 ------- 2 1.0 7.0 1 1
14 + 0.9 1 4 cbr 1000 ------- 2 1.0 7.0 2 2
15 - 0.9 1 4 cbr 1000 ------- 2 1.0 7.0 2 2
16 r 0.90204 1 4 cbr 1000 ------- 2 1.0 7.0 2 2
17 + 0.90204 4 5 cbr 1000 ------- 2 1.0 7.0 2 2
18 - 0.90204 4 5 cbr 1000 ------- 2 1.0 7.0 2 2
19 r 0.92804 4 5 cbr 1000 ------- 2 1.0 7.0 2 2
20 + 0.92804 5 7 cbr 1000 ------- 2 1.0 7.0 2 2
21 - 0.92804 5 7 cbr 1000 ------- 2 1.0 7.0 2 2
22 r 0.93008 5 7 cbr 1000 ------- 2 1.0 7.0 2 2
23 + 1.3 1 4 cbr 1000 ------- 2 1.0 7.0 3 3
24 - 1.3 1 4 cbr 1000 ------- 2 1.0 7.0 3 3
25 r 1.30204 1 4 cbr 1000 ------- 2 1.0 7.0 3 3
26 + 1.30204 4 5 cbr 1000 ------- 2 1.0 7.0 3 3
27 - 1.30204 4 5 cbr 1000 ------- 2 1.0 7.0 3 3
28 r 1.32804 4 5 cbr 1000 ------- 2 1.0 7.0 3 3
29 + 1.32804 5 7 cbr 1000 ------- 2 1.0 7.0 3 3
30 - 1.32804 5 7 cbr 1000 ------- 2 1.0 7.0 3 3
31 r 1.33008 5 7 cbr 1000 ------- 2 1.0 7.0 3 3
32 + 1.7 1 4 cbr 1000 ------- 2 1.0 7.0 4 4
33 - 1.7 1 4 cbr 1000 ------- 2 1.0 7.0 4 4
34 r 1.70204 1 4 cbr 1000 ------- 2 1.0 7.0 4 4
35 + 1.70204 4 5 cbr 1000 ------- 2 1.0 7.0 4 4
36 - 1.70204 4 5 cbr 1000 ------- 2 1.0 7.0 4 4
```


Figure 4: Shows screenshot of trace file out.tr

The simulation topology (Figure 3) was used to obtain results. Nodes 0, 2 as well as 3 are the TCP agents; node 1 is the UDP agent; while nodes 4 and 5 are the bottleneck routers while nodes 6, as well as 7, are TCP sink and UDP null agent as well.

The process scripts' results of the proposed scheme simulation study run using many bottleneck bandwidths as well as packet loss rate at all buffer-sizes for the queue mechanisms used as performance metrics are depicted as follows:

Table 3: Drop-Tail queue at 2% Link Loss Rate for the Existing vs the proposed buffer-sizes

Existing BS				Proposed BS			
Buffer Size	TCP Delay (ms)	TCP Throughput (kbps)	Packet Lost	Buffer Size	TCP Delay (ms)	TCP Throughput (kbps)	Packet Lost
PLR= 0.01; BOTTLENECK LINK = 0.05MB							
2	178.3490	18.0147	24	5	374.3300	28.4992	21
5	381.2210	30.0792	28	6	450.8480	27.5560	7
10	775.3910	28.7432	11	7	564.6020	28.0558	6
15	1072.1700	29.0030	11	8	589.8180	27.4100	7
30	1354.5700	28.5293	5	9	661.7560	28.2247	14
50	1354.5700	28.5293	5	10	692.5040	27.1950	3
PLR= 0.01; BOTTLENECK LINK = 0.1MB							
3	103.6710	59.0899	13	5	158.1560	70.2890	4
5	160.1740	72.1404	3	6	192.7710	70.9049	1
10	320.6560	73.8926	11	7	227.6620	72.8275	4

15	394.9320	74.9065	3	8	258.6510	71.9656	3
30	490.6810	74.9956	4	9	286.7190	72.7190	12
50	430.8980	74.6579	1	10	288.2440	71.9867	7
PLR= 0.01; BOTTLENECK LINK = 0.5MB							
5	40.9624	351.4700	9	5	41.2760	299.0650	6
7	48.5070	383.2250	10	6	43.3020	332.8850	8
10	57.6076	403.1830	5	7	47.4652	344.1870	8
15	67.0747	406.4690	4	8	49.9440	349.6760	5
30	71.8355	412.6930	4	9	53.7582	364.1020	5
50	71.8355	412.6930	4	10	56.4941	372.2190	7

Table 4: Drop-Tail queue at 2% Link Loss Rate for the Existing vs proposed buffer-sizes

PLR= 0.02; BOTTLENECK LINK = 0.05MB

Existing BS				Proposed BS			
Buffer Size	TCP Delay (ms)	TCP Throughput (kbps)	Packet Lost	Buffer Size	TCP Delay (ms)	TCP Throughput (kbps)	Packet Lost
2	175.4780	19.5681	38	5	350.3140	26.7790	16
5	362.0350	28.5775	19	6	409.2030	26.7112	6
10	661.1730	28.2249	10	7	531.8340	26.8254	7
15	946.8790	28.6506	13	8	541.9130	25.4549	12
30	1043.9000	28.4120	8	9	595.8100	27.0619	12
50	1043.9000	28.4120	8	10	619.8830	26.3498	8
PLR= 0.02; BOTTLENECK LINK = 0.1MB							
3	102.7600	58.8391	22	5	151.0290	69.0657	9
5	153.1650	71.1429	11	6	177.0660	68.0179	6
10	263.1400	71.9833	7	7	209.9610	71.3704	7
15	327.2680	73.5945	10	8	237.2240	70.2140	11
30	327.2680	73.5945	10	9	236.8120	68.6734	7
50	327.2680	73.5945	10	10	249.8370	69.2628	7

PLR= 0.02; BOTTLENECK LINK = 0.5MB							
5	39.3818	310.3260	15	5	38.8984	266.6810	9
7	43.6110	322.9690	10	6	40.9940	288.2760	11
10	49.1507	328.8220	9	7	43.0229	300.7090	10
15	51.4326	328.3110	9	8	44.8172	294.4260	3
30	51.4326	328.3110	9	9	47.2889	309.9500	13
50	51.4326	328.3110	9	10	48.4071	313.0460	12

Table 5: Drop-Tail queue at 5% Link Loss Rate for the Existing vs proposed buffer-sizes

PLR= 0.05; BOTTLENECK LINK = 0.05MB

Existing BS				Proposed BS			
Buffer Size	TCP Delay (ms)	TCP Throughput (kbps)	Packet Lost	Buffer Size	TCP Delay (ms)	TCP Throughput (kbps)	Packet Lost
2	173.9200	18.7237	47	5	342.7320	26.2755	24
5	327.1210	27.3123	21	6	370.4870	24.6768	18
10	554.0080	29.4352	25	7	444.5360	26.2080	18
15	522.2170	29.0075	22	8	476.3130	27.4355	24
30	522.2170	29.0075	22	9	484.6800	27.1216	19
50	522.2170	29.0075	22	10	502.4790	27.2342	20
PLR= 0.05; BOTTLENECK LINK = 0.1MB							
3	99.7677	55.1611	27	5	138.5680	61.3584	15
5	139.0470	66.5842	16	6	158.6030	58.9879	15
10	180.2190	68.4256	13	7	173.3390	57.7602	13
15	180.2190	68.4256	13	8	175.3640	64.1372	15
30	180.2190	68.4256	13	9	181.0150	65.1035	17
50	180.2190	68.4256	13	10	181.0150	65.1035	17
PLR= 0.05; BOTTLENECK LINK = 0.5MB							
5	35.7222	202.5380	17	5	35.6100	183.1530	15
7	35.9693	203.8630	17	6	35.5351	182.7040	10
10	35.9693	202.5380	17	7	35.5351	182.7040	10
15	35.9693	202.5380	17	8	35.5351	182.7040	10
30	35.9693	202.5380	17	9	35.5351	182.7040	10
50	35.9693	202.5380	17	10	35.5351	182.7040	10

Performance Evaluation

Figure 5, 6 as well as 7 illustrates the average delay and throughput of simulations at different PLRs for drop-tail queue mechanism for the existing buffer sizes from 0-50. It indicates that as delay increases, the throughput decreases and vice versa. It shows that although there is a little increase in throughput at buffer-size 15, it comes with a compromise (i.e. higher delay) for 1% PLR irrespective of bottleneck bandwidth. PLR of 2% resulted in constant average delay and throughput at buffer-sizes 30-50 for bottleneck bandwidth of 0.05mb while at bottleneck bandwidths 0.1mb and 0.5mb the average delay and throughput are constant from buffer-sizes 15-50. PLR 5% resulted in a constant delay and throughput for buffer-sizes 15-50 at 0.05mb bottleneck bandwidth, and at 0.1mb and 0.5mb, average delay and throughput remained constant from buffer-sizes 10-50.

The proposed buffer size of 5 – 10 is observed to have a better average in delay and throughput compared to the Existing buffer sizes of 0–50, as shown in Figure. 6, 8 and 10.

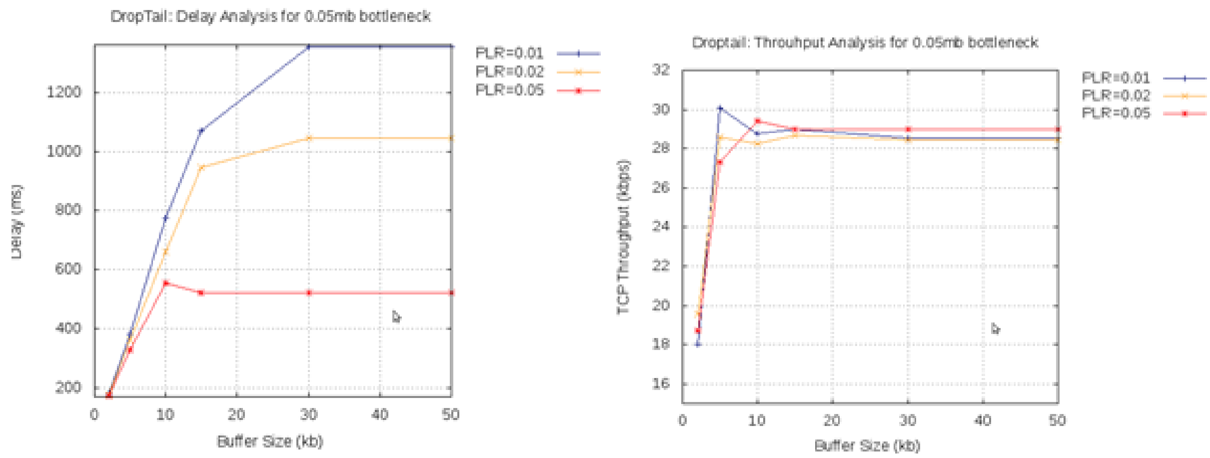


Figure 5: Delay vs Throughput for Drop-Tail Queue for 0.05MB bottleneck link

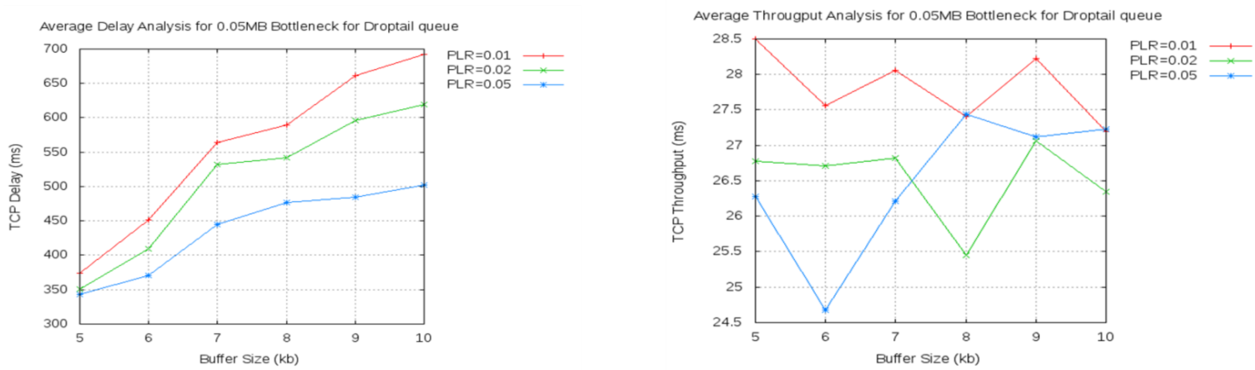


Figure 6: Delay vs Throughput for Drop-Tail Queue for 0.05MB bottleneck link (Proposed)

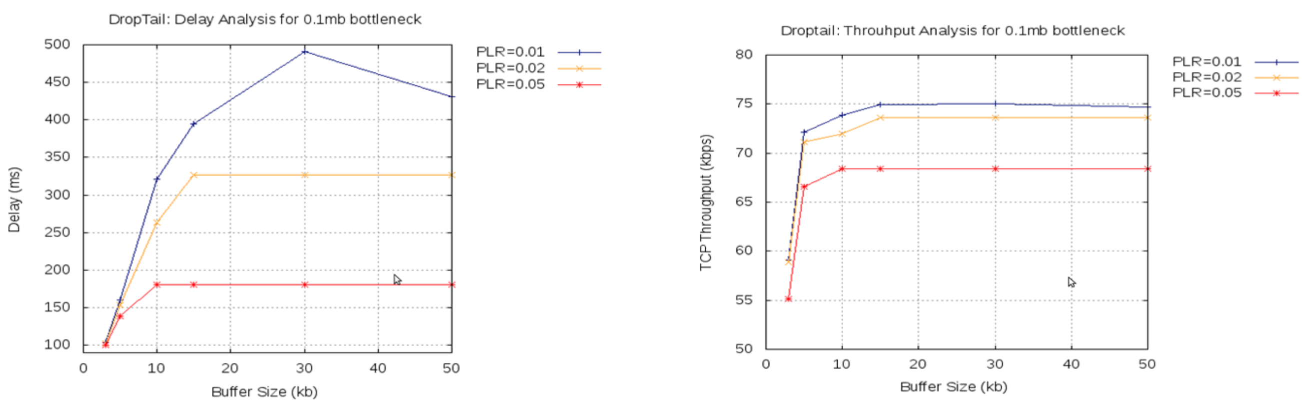


Figure 7: Delay vs Throughput for Drop-Tail Queue for 0.1MB bottleneck link

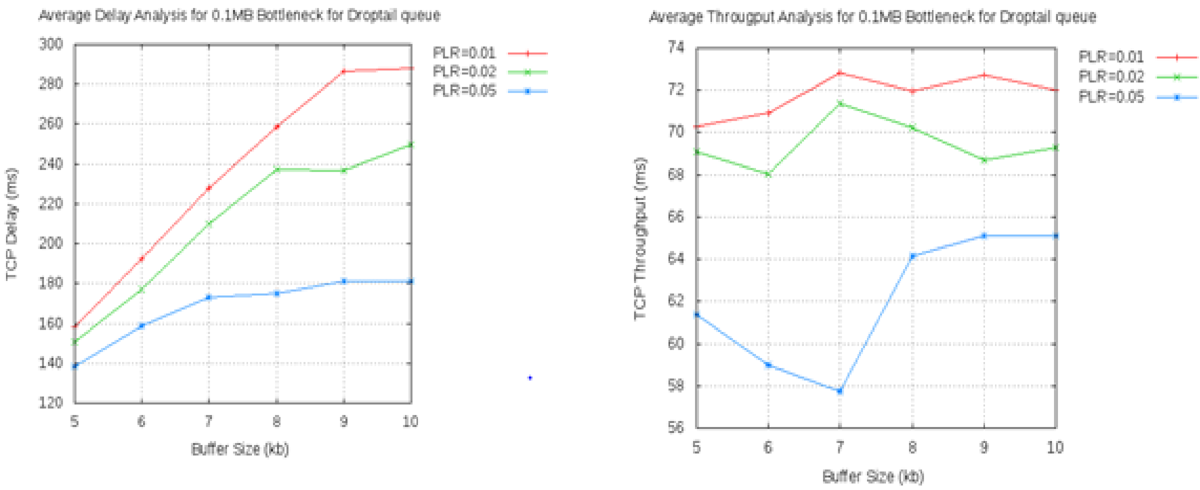


Figure 8: Delay vs Throughput for Drop-Tail Queue for 0.1MB bottleneck link (Proposed)

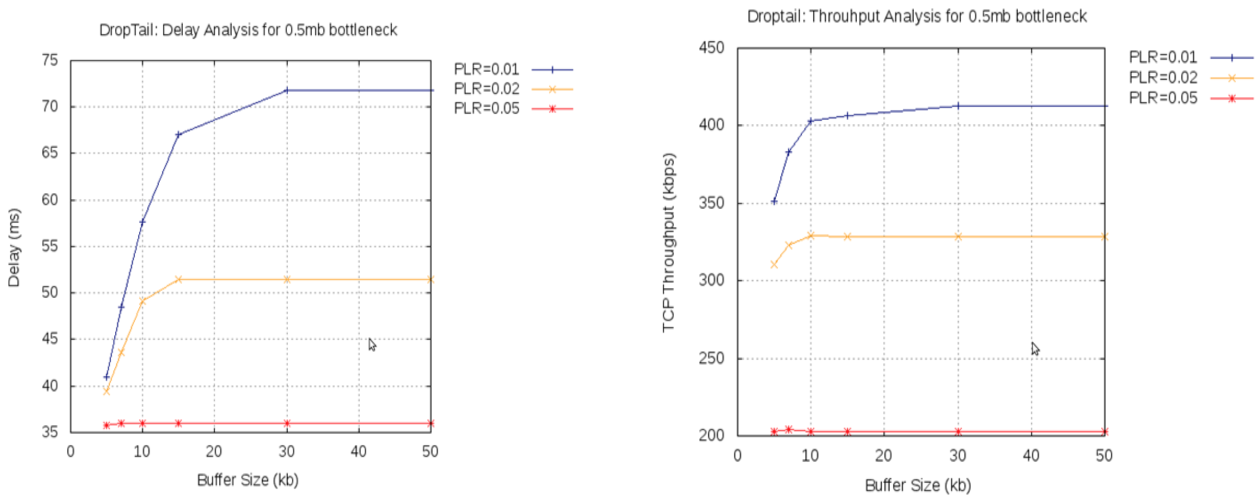
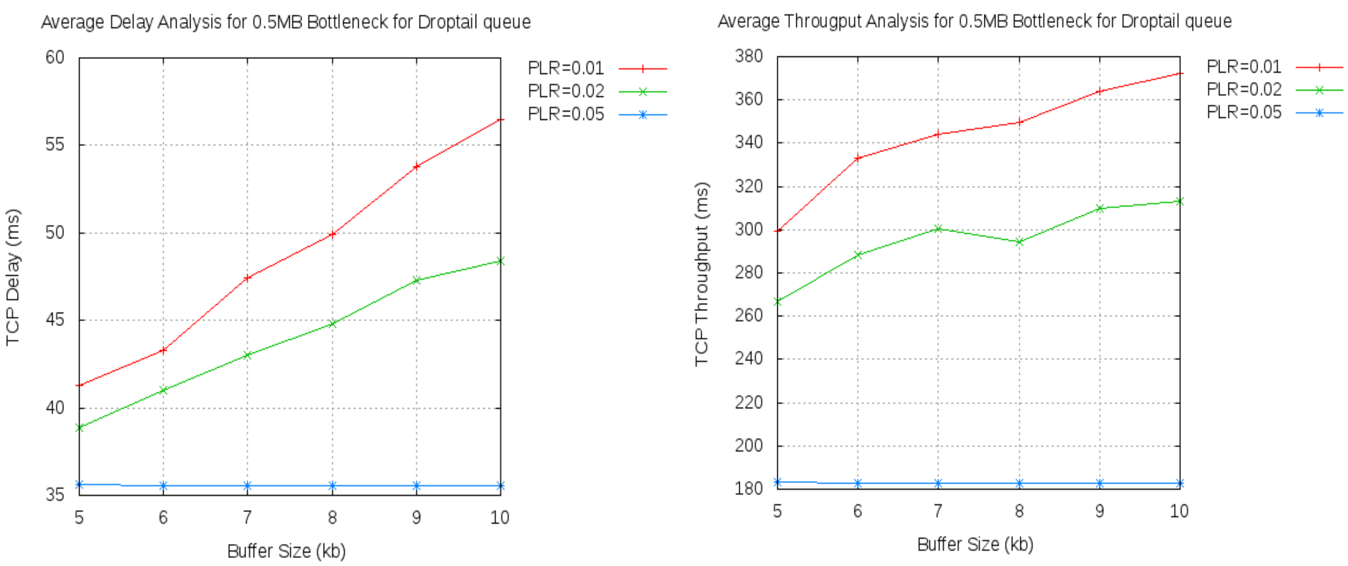


Figure 9: Delay vs Throughput for Drop-Tail Queue for 0.5MB bottleneck link



In our proposed scheme, the bottleneck link use was that of lossy as well as low bandwidth ranging from 0.05mb, 0.1mb and 0.5mb respectively. Thus, the analysis of the results achieved from the experiment are depicted as well as discussed as follows:

Experimental Results

At the simulation time, we have observed the influence of many packet loss rate (PLR) such as PLR at 1%, at 2%, as well as at 5% on the performance metric. And it was revealed that the more the packet loss (PL) experience, the lower the throughput as well as, the higher the response delay incurred in transmission of packets. Thus, PL has a great effect on throughput as well as delays resulting in poor QoS. Furthermore, we noted that as buffer-size increased from 10 ahead, severe network degradation was achieved on both throughputs as well as delay. More-so, the bottleneck link had effects on the performance of the metrics; such that, the larger the bottleneck links, the higher the throughput as well. While the lower the delay incurred. Similarly, the smaller the links, the smaller its throughput and the higher the delay as buffer sizes is being increased. In addition, after simulating buffer-sizes of ranges 15, almost all the performance metrics revealed same average outputs, Table 3 to 5 and Figure 5-10 as follows

Conclusion

This paper proposes a suitable buffer size for heterogeneous network traffic, and the paper addressed the challenges of buffer sizes for heterogeneous networks. In the study, a buffer size of 5 to 10KB is proposed in order to achieved efficient network performance. Since, Buffer sizes is an important concern in TCP flows, in order to differentiate links that TCP flows experience packet loss rate (PLR) as well as where they do not. An under-provisioned internet network may experience much-queuing delay producing much packet loss rate. Thus, the choice of buffer size has great importance on the overall network performance. Therefore, we conducted a performance analysis, as well as an extensive NS2 simulations study of multi-flow, traffics run in order to obtain an efficient buffer size that results in superior average throughput as well as response time without additional signalling as well as computation cost by the network routers. And our proposed buffer size of five (5) to ten (10) obtained from this study publicized that having a buffer size of 5 to 10 KB advocate the best throughput with a negligible delay. In addition, our proposed results also proved that buffer sizes had been optimized accordingly for heterogeneous network traffics.

Acknowledgement

The authors wish to thank the anonymous reviewers who took out their time and constructively critic the paper which have improved the research paper.

References

[1] M. A. Ahmad, D.D. Wisdom and S. Isaac, "Optimized Buffer Sizes to Improve Quality of Service (QoS) in Networks with Heterogeneous Traffic", Special Issue on *Advancies in Sciences, Technology and Engineering Systems Journal, (ASTESJ) China* 2020.

[2] N. Ariel, G. Kaempfer, O. Shimon, R. Sidi, & S. Barkai, "Optimization of the Transmission Control Protocol Particularly for wireless Connections", U.S. Patent 2013/0058212, March 7, 2013.

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

[4] A. Dhamdhere, J. Hao & C. Dovrolis, C., "Buffer Sizing for congested internet links", *Proceedings IEEE 24Th Annual Joint Conference of the IEEE Computer and Communications Societies.* DOI:10.1109/infcom.2005.1498335.

[5] N. Dukkipati, T. Refice, Y. Cheng, J. Chu, T. Herbert, A. Agarwal, A. Jain, & N. Sutin, "An Argument for Increasing TCP's Initial Congestion Window", *ACM Comput. Commun. Rev*., 40, 2010.

[6] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, & T. Roughgarden, "Part III: Routers with very small buffers", *ACM SIGCOMM Computer Communication Review,35*(3), 83. 2015.

[7] B. Kaur, M. Kaur, P. Mudgil, & H. Singh, "Importance of sliding window protocol", *International Journal of Research in Engineering and Technology*, *2*(10). 2013.

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

[9] I. Khan, and M. Hassan, "Transport Layer Protocols and Services", *International Journal of Research in Computer and Communication Technology, 5,* 2016, 2320-5156.

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

[11] S. Radhakrishnan, Y. Cheng, J. Chu, A. Jain, and B. Raghavan, " TCP Fast Open", *In Proc. of the ACM Int. Conf. on Emerging Networking Experiments and Technologies (CoNEXT),* Dec. 2011

[12] G. Raina, D. Towsley, and D. Wischik, Part II. *ACM SIGCOMM Computer Communication Review*, *35*(3), (July 2005)4. doi:10.1145/1070873.1070885

[13] A. Sathiaseelan, R. Secchi, G. Fairhurst, " Enhancing TCP to Support Rate-Limited Traffic School of Engineering University of Aberdeen", *CSWS'12, Nice, France*, December 10, 2012.

[14] J. Touch, "Automating the Initial Window in TCP", Internet Draft*, work-in-Progress, draft-touch -tcpm-automatic-IW-03.txt,* July 2012.

[15] G. Vu-Brugier, R. Stanojevic, D. Leith, & R. Shorten, " A critique of recently proposed buffer sizing strategies". *ACM SIGCOMM Computer Communication Review*, *37*(1), 43. 2007.