Router Buffer Sizing in Mixed Traffic

Sherrin Benjamin Jacob, Shilpa Verma

Abstract— In today's electronic world, it is observed that 90-95% of Internet traffic is transmitted over TCP, but there's an increasing demand for real time traffic with the boom of VoIP, online gaming and IPTV applications etc. When the interaction between TCP and UDP traffic happens, they balance well for a specific range of buffer size and later as the buffer size increases the loss of UDP packet also increases, with significantly no change in the TCP traffic's throughput. So the mixture of TCP and UDP traffic doesn't go well with large sized buffer. Therefore, it is necessary to decide an appropriate buffer size not only for Internet, but also for optical packet switched networks as they have very low buffering capacity.

Index Terms— Congestion Window Size, Packet Size, NS-2, TCP, UDP, Packet Loss, Router Buffers

———————————— ◆ ————————————

1 Introduction

Internet routers are packet switches, in which packets get buffered during the time of congestion. Hence sizing these buffers is a crucial issue. Conventionally, it was practiced that buffer size of a router B should be RTT*W, where RTT=average Round trip time and W=capacity of bottleneck link [1]. Arguably, router buffers are the single biggest contributor to uncertainty in the Internet. Buffers cause queuing delay and delay-variance; when they overflow they cause packet loss, and when they under- flow they can degrade throughput [3].

Buffer bloat is a very common phenomenon found in routers, switches, gateways and broadband gear. This occurs due to the presence of excessively large buffers in systems, particularly in communication systems. Buffers with bloated sizes have latency measured in seconds, rather than microseconds or milliseconds. According to the telephone standards, the maximum desirable latencies are in the 150–200 ms range, and human perception for some latency is as low as 10 ms. Although some buffering is required to smooth bursts in communications systems, but packet loss is the only way to signal the presence of congestion in the network, and congestion-avoiding protocols such as TCP rely on timely congestion notification to regulate their transmission speeds. When an oversized buffer is put into the system, there are traces of higher packet loss.

According to the networking fundamentals, datagram network with infinite storage, first-in-first-out queuing, and a finite packet lifetime will under overload or drop all packets. However some of the buffers in the Internet are effectively infinite in size. Not all packet loss is evil: some packet loss can be essential for correct operation. However once bloated buffers fill, there's no timely congestion notification by packet loss or explicit congestion notification. Only the buffers on either side of the bottleneck link fill, and if those buffers are not well managed, they can fill up completely, causing much higher packet loss. Other buffers in the path, remaining nearly empty, remain dark and undetectable. Any application that saturates a link with bloated buffers can induce buffer bloat pain, for instance uploading videos to YouTube, emailing messages with large images attached, backing up large files or file systems, downloading large files, such as ISO images, a Linux distribution image, or a movie via Bit torrent, watching Netflix, and even visiting certain kinds of web pages can all fill these buffers [2].

There has been vital discussions regarding the size of buffers required at core Internet routers from past few years. Recently researchers have made arguments with the help of theory and experimentation that proves the same.

2 Review Of Literature

Operating the router buffers in the "anomalous region" can result in increased UDP packet loss, with only a marginal improvement in end-to-end TCP throughput, which is undesirable from a network operator's point of view. Extensive simulations in [1] show the presence of anomalous loss performance for real time traffic. These simulations consider the real video traces, short range Poisson models and long range fractional Brownian motion models along with several other TCP flows. These two quantitative models are used to explain the anomaly of packet loss. The 1st model captures the buffer sharing dynamics between real time and TCP traffic and also shows the impact of greedy nature of TCP traffic on the effective buffers available due to the real time traffic. The 2nd model is Markov chain based approach which gives a numerical evaluation of packet loss. In [1], it is suggested that several other TCP parameters such as round trip times, number of flows, and relative mix of long lived and short lived TCP flows also impact on the severity of anomaly. The sizes of UDP packets have an effect on the anomaly. When the sizes of the UDP packets were larger, the losses were also large [1].

According to [4], impact of flow arrivals and departures in networks are the main motivation for the extensive research in Buffer sizing. Networks are characterized by a vast disparity in the operating speeds of access routers and core routers. When there are flow arrivals and departures, [4] show that the core routers are rarely congested even at high loads of 98%. Since there is no congestion on the core router, the flows are largely limited by their access speeds. The impact of small buffers on a single congested link where the access limitations are absent is very trivial. It is rather well known that TCP approximates processor sharing, when the file-sizes are large. Therefore, at any time, very few active flows are present in the network even at significantly high loads. For example, under processor sharing, even at 90% loading, the probability that more than 50 flows are active is about 0.005. Therefore, the assumption that a large number of users exist in the system does not hold. Thus, large reductions in buffer sizes due to statistical multiplexing effects the performance of the router. In fact, reducing buffer-sizes in these networks would result in dramatic degradation in the overall performance. If the core router is heavily loaded up to 98%, it can still operate with very small buffers if the core to access speed ratio is small. Here when a particular access to core router ratio is assigned, there exists a threshold operating load below which small buffers seem to be adequate and above this threshold, a buffer of size O(C*RTT) is required.

The internal architecture of commercial routers and the precise set of queues a packet goes through are not easily available hence it becomes difficult to analyze the effect of reducing buffer size. Further, there some routers that tend to have hidden buffers that are not accessible to the user, and hence buffer size cannot be controlled directly. The main result using the drop-tail scheme is that while link utilization is largely independent of router architecture, buffer size and offered load, other metrics such as loss and delay are much more sensitive.

Hence it is prudent to size router buffers at a value that balances the performance of both TCP and UDP trace appropriately. Operating the router buffers at a very small value can adversely impact the performance of both TCP and UDP trace. Furthermore, operating it in the “anomalous region" can result in increased UDP packet loss, with only a marginal improvement in end-to-end TCP throughput. Moreover, building an all-optical packet router and buffering of packets in the optical domain is a rather complex and expensive operation [5].

Understanding the nature of network traffic is critical In order to properly design and implement computer networks and network services like the WWW, recent examinations of local-area network traffic and wide-area network traffic have challenged the commonly assumed models for network traffic, for example, the poisson process. If the traffic were to follow a poisson or markovian arrival process, it would have a typical burst length which would tend to be smoothed by averaging over a long enough time scale. The measurements of real traffic indicate that significant traffic variance i.e. burstiness is present on a wide range of time scales. Traffic that is bursty on many or all time scales can be described statistically using the notion of self-similarity. Self-similarity is the property that is associated with one type of Fractal—an object whose appearance is unchanged regardless of the scale at which it is viewed. In the case of stochastic objects like time series, self-similarity is used in the distributional sense: when viewed at varying scales, the object’s correlation structure remains unchanged. As a result, such a time series exhibits bursts—extended periods above the mean—at a wide range of time scales. Since a self-similar process has observable bursts at a wide range of time scales, it can exhibit long-range dependence [6].

The Internet was designed as a best-effort network, which means delivering information without any delay guarantees. Many years later, it has grown up and is used to provide new services, which sometimes have real-time requirements. One of the first real-time services deployed was VoIP, which is nowadays widely used, and is replacing traditional telephony systems. The problem of using a best-effort network for deploying a real-time interactive service has been largely discussed, as the users of traditional telephony would not like to change to a new technology unless the offered quality was similar to the one they are used to. Thus, many studies were carried out, trying to identify the different network impairments, and to quantify their effect on the quality perceived by users. A number of hosts are connected via the same access router to a server. If gaming service is considered, this is also a scenario in an Internet café. Gaming has been reported as one of the main activities deployed by the users of these businesses, which are still a very popular way for connecting to the Internet.

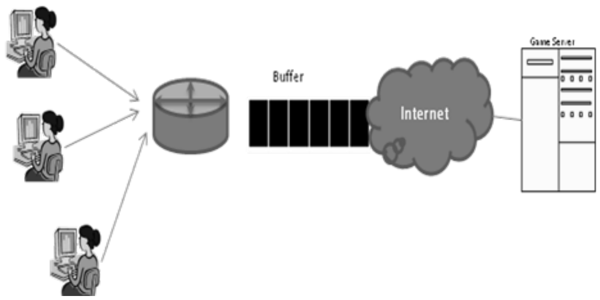


Fig. 1 Scenario of Internet Café

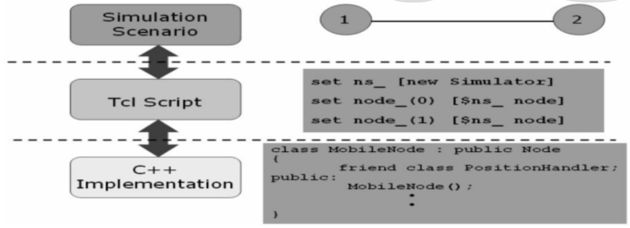
In Fig 1, the connection will be shared by many services, some of them real-time. The total traffic offered to the router may vary and in some moments it can even be over the link capacity. So the router will add delays and discard packets, and the buffer size and policy will have an influence on the distribution of these impairments. In particular, some policies are packet-size aware, as they penalize different sizes in different manners [7].

In the field of communication safe and reliable transmission of data is an important issue. Emergence of computer networks and internet brings tremendous changes in the field of communication. A computer network forms, when two or more computer systems interconnect over a same link for the sharing of resources. Wired computer network is vastly recommended because they are flexible and multi adaptive in nature. In wired network various queues has been used out for the safe and reliable transmission of data. Thus, the implementation and proper selection of the queues is one of the most crucial issues. The choice of the particular queue from the various queues is totally depended upon the need of transmission of data. The transmission of packet over a medium at any instance of time requires a packet processing routine. Thus, to maintain a proper processing of the packets over a source an interface must be deployed. This interface object must be able to accept the request from source objects to transmit a packet, even when the medium is busy in transmitting a previous packet. The various queues, which were implemented, can be discussed into the following categories: Drop Tail Queue, Random Early Detection (RED Queue), Stochastic Fair Queuing, Deficit Round Robin (DRR), Fair Queue

(FQ). The simulator is a tool for demonstrating the various protocols, algorithms and to serve as an aid in the better understanding of the protocols. The RED shows maximum number of packet loss in case of CBR traffic while SFQ shows maximum number of packet loss in case of FTP. The FQ shows minimum number of packet loss in CBR and FTP traffic [8].

High quality multimedia applications present in Internet Protocol networks have rigid requirements in terms of packet loss, delay, and delay jitter. In IP networks, loss characteristics affect the quality of the established video and/or voice connections, and influence the throughput of bulk data transfers. These characteristics cannot be captured by specifying only the loss probability but requires a detailed study of loss patterns. Gilbert model which are widely used for measuring TCP/UDP traffic flows was originally derived from UDP loss traces obtained from genuine Internet measurements. This is used to validate the loss process generated by simulations [9].

A large congestion window size in the wireless multi-hop networks worsens TCP performance by increasing the probability of packet contention and packet losses from excessive collisions. There have been some previous works on TCP bandwidth measurement methods. In TCP-Westwood, the sender estimates the available bandwidth dynamically by measuring and averaging the rate of returning ACKs. In TCP-Peach, the sender probes the available network bandwidth in only one RTT with the help of low-priority dummy packets. A similar method is used in TCP-Jersey, which was derived from the time-sliding window estimator. Hence in wireless multi-hop networks, it is necessary to limit the TCP congestion window size so as not to increase the probability of congestion loss, thereby ensuring a good TCP performance [10].



```
set ns_ [new Simulator]
set node_(0) [$ns_ node]
set node_(1) [$ns_ node]
```

```
class MobileNode : public Node
{
        friend class PositionHandler;
public:
         MobileNode();
              :
}
```

Fig. 2 NS2 from Developer's View

NS-2 (Network Simulation Version 2) is an object-oriented simulator developed as part of the VINT project at the University of California in Berkeley. This project was funded by DARPA in collaboration with XEROX Palo Alto Research Centre (PARC) and Lawrence Berkeley National Laboratory (LBNL). NS-2 is extensively used by the community of networking research. It also provides support for simulation of multicast, routing, TCP protocols over wired and wireless networks, etc. This simulator is event-driven and runs in a non-real time fashion. It consists of C++ core methods and uses TCL and Object TCL shell as an interface allowing the input file (simulation script) to describe the model to simulate. Users can define arbitrary network topologies composed of nodes, routers, links and shared media. A rich set of protocol objects can then be attached to nodes, which are usually called as agents. In NS-2 network physical activities are translated to events, events are queued and processed in the order of their scheduled occurrences. And the simulation time progresses with the events processed. And the simulation "time" may not be the real life time as "inputted". NS-2, can model essential network components, traffic models and applications. Typically, it can configure transport layer protocols, routing protocols, interface queues, and also link layer mechanisms.

From the NS-2 developer view, Figure 2 shows the layered architecture of NS. The event schedulers and most of the network components are implemented in C++ and available to TCL Script, thus the lowest level of NS-2 is implemented by C++, and the TCL script level is on the top of it to make simulations much easier to be conducted. Then, upon the TCL level, there's overview of the network, i.e. the simulation scenario. These elements are combined as so called NS-2 software. In order to successfully carry out one simulation, the data must be provided to NS-2:

1) Appearance of the network: The whole topology view of sensor network or mobile network, this includes the position of nodes with (x, y, z) coordinate, the node movement parameters, the movement starting time, the movement is to which direction, and the node movement speed with pausing time between two supposed movement.

2) Internal of the network: Since the simulation is on the network traffic, so it is important to inform the NS-2 about which are source nodes, about the connections to be made, what kind of connection to be used (simplex/duplex).

3) Configuration of the layered structure of each node in the network: This includes the detailed configuration of network components on sensor node, and a need to drive the simulation. It is necessary to direct the simulation results to a particular trace file, which is required to be mentioned in the TCL script, and also about how to organize a simulation process [11].

3 Our Contributions to the Study of Router's Buffer Sizing

Our work consisted of 6 phases.

Phase I: NS2 simulations are performed to study the UDP packets loss by varying the number of UDP flows. Here the number of nodes created was 52. Of which 25 pairs of Source – Destination nodes in which TCP traffic flowed. And 1 pair of Source- Destination node in which UDP traffic flowed. For every Source- Destination pair different start times were allotted to get the real world traffic results. Both TCP and UDP traffic was made go through 2 routers named r0 and r1. All the Source nodes were connected to r0 and all the Destination nodes were connected to r1. Buffer sizes between these two routers were varied from 5-50kB.

Phase II: In this phase different topologies like Ring, Bottle Neck, and Derived with similar network configurations were simulated.

Phase III: NS-2 simulations are performed to study the UDP packets loss by varying the Queue types like Class Based Queuing, Fair Queuing, and Stochastic Fair Queuing, Drop Tail.

Phase IV: In the above 2 phases, the total number of nodes used were 54. In order to understand the real world scenario the number of nodes was increased to 200. 100 nodes are Source – Destination pairs through which TCP traffic were passed. The number Source- Destination pairs were varied according to the percentage of UDP traffic. When 5% of UDP traffic was simulated, 95 source nodes of TCP and 5 source nodes of UDP and as the percentage of UDP traffic was increased, the number of TCP nodes were reduced. In these simulations, congestion window size and time to live field was also set for every traffic flow. These two parameters have an immense effect on the traffic flow. The queue type used was Drop Tail as it said to be one of efficient queues for Bottle Neck network topology. The Buffer Size was varied from 5kB to 50kB.

Phase V: Here AWK scripts were coded to measure TCP Throughput, UDP Throughput and UDP Loss Probability.

Phase VI: Here graphs are plotted on the values obtained from the trace files and AWK script. Graphs help in analysing the data better. So graphs were plotted for different topologies, different queue types, and varying number of TCP and UDP nodes. On the x axis, Buffer Size was plotted and on y1 axis TCP throughput and y2 axis UDP throughput were plotted.

4 Results And Discussions

The graphs shown below are obtained by running the simulations and measuring the parameters like TCP Throughput, UDP Throughput and UDP Loss Probability.

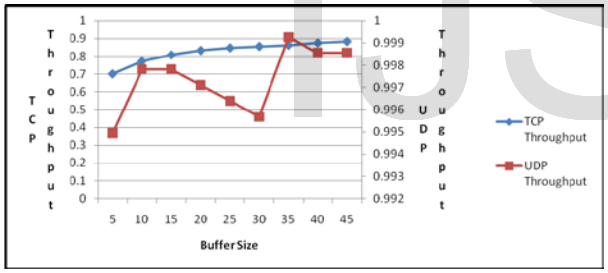


Fig. 3. TCP & UDP Throughput with varying buffer sizes with 25 traffic flows of TCP and 1 traffic flow of UDP

Here Figure 3, shows the Throughput values when 25 traffic flows of TCP are sent and 1 traffic flow of UDP is sent. It can be observed that throughput of TCP traffic keeps on increasing with the increasing Buffer Size. However, the throughput of UDP traffic keeps increasing till the buffer size is 20 and then throughput reduces and it reaches the peak, and then remains stable. Hence it seems that keeping a buffer size of 40kB is favourable for both UDP and TCP traffic.

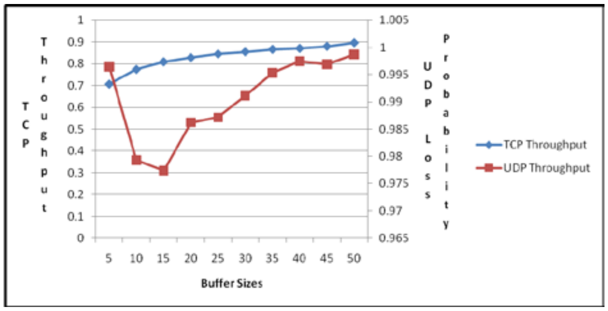


Fig. 4. TCP & UDP Throughput with varying buffer sizes with 25 traffic flows of TCP and 2 traffic flows of UDP

As it seen that the throughput of TCP traffic is linearly increasing whereas the throughput of UDP is at peak when the buffer size is fixed as 45-50kB.

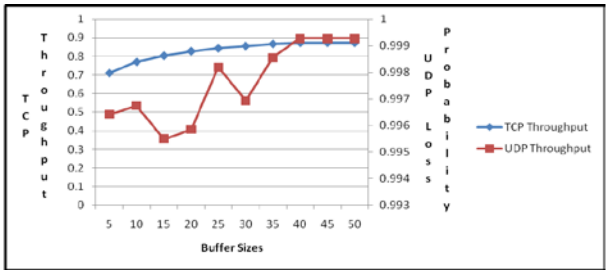


Fig. 5. TCP & UDP Throughput with varying buffer sizes with 25 traffic flows of TCP and 3 traffic flows of UDP

Here Figure 5 shows the throughput values of 25 traffic flows of TCP and 3 traffic flows of UDP. The throughput values of TCP traffic is steadily increasing, whereas the throughput of UDP traffic is the highest when the buffer size is 40-45kB. Hence a buffer size of 40-45 kB is beneficial when mixed traffic is flowing in the network.

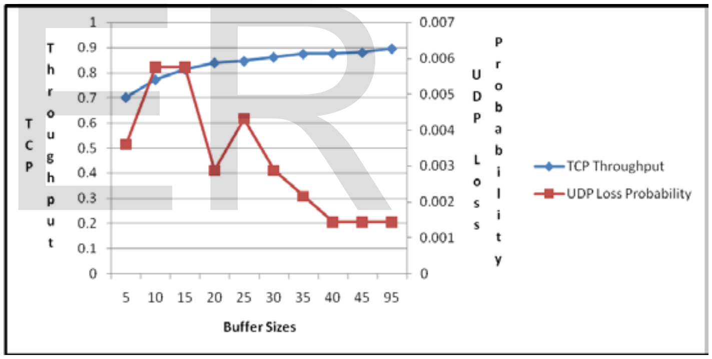


Fig. 6. TCP, UDP Throughput and UDP Loss Probability with varying buffer sizes with 25 traffic flows of TCP and 4 traffic flows of UDP

Here Figure 6 shows the Throughput values for 25 traffic flows of TCP and for 4 traffic flows of UDP. The throughput values of TCP traffic keep on rising. In case of UDP traffic, the throughput keeps on increasing till the buffer size is 20kB but when the buffer size increased to 25kb, there's a slight reduction in its throughput. Later throughput of UDP traffic steadily increases to a constant value. It can be stated that as the number of flows of UDP arise, the throughput values are balanced for both the traffic. Hence, a buffer size of 40-45kB would be very much favourable for both types of traffic.

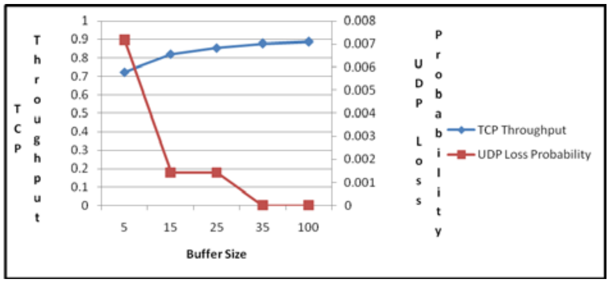


Fig.7.Varying Buffer Sizes on Derived Topology Drop Tail

For Figure 7, a network of 25 pairs of Source-Destination nodes was designed in Derived topology with the queue type as Drop Tail. Here there's only 1 flow of UDP traffic and 25 flows of TCP traffic. The loss probability of UDP traffic is extremely good because there's only 1 flow, whereas for TCP it was steadily increasing. Hence it can be said that with the buffer size of 35-40kB would suffice the needs of both the traffic.

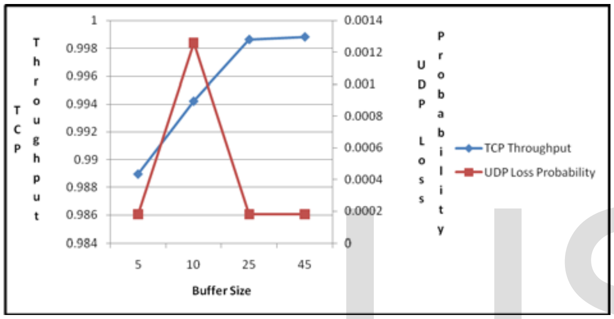


Fig 8: Varying Buffer Sizes on Bottle Neck Topology with FQ

For Figure 8, a network of 25 pairs of Source-Destination nodes was designed in Bottle-Neck topology with the queue type as Fair Queuing. Here there's only 1 flow of UDP traffic and 25 flows of TCP traffic. The loss probability of UDP traffic is constant because there's only 1 flow, whereas for TCP it is constant. Hence it can be said that with the buffer size of 35-40kB would be enough for the needs of both the traffic.

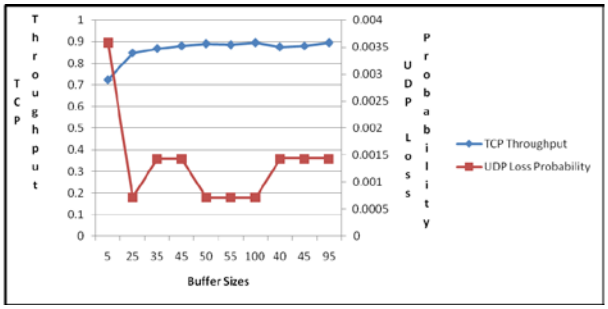


Fig 9: Varying Buffer Sizes on Bottle Neck Topology with RED

For Figure 9, a network of 25 pairs of Source-Destination nodes was designed in Bottle-Neck topology with the queue type as Random Early Detection. Here there's only 1 flow of UDP traffic and 25 flows of TCP traffic. The throughput of UDP traffic is increasing but there's a slight reduction when the buffer size is between 35-45kB. Then it steadily increases and reaches the peak. Whereas for TCP traffic, the throughput values steadily increases except when the buffer size is 55kB.

Hence, it can be stated that by keeping the buffer size to 45-50kB would be beneficial for both the traffic.

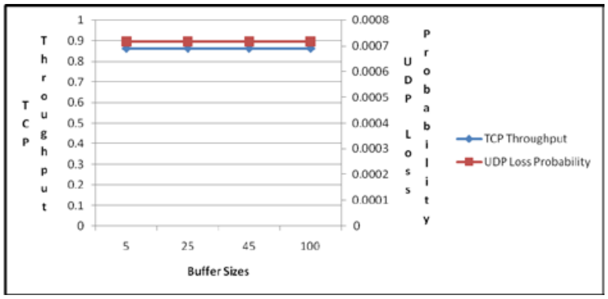


Fig 10: Varying Buffer Sizes on Bottle Neck Topology with SFQ

Here in Figure 10, a network of 25 Source- Destination pairs were simulated with 25 traffic flows of TCP traffic and 1 traffic flow of UDP. In this network, topology used was Bottle Neck and the queue type used was Stochastic Fair Queuing. Due to the type of topology used and the type of the queue used, the throughput values remain constant for both the traffic.

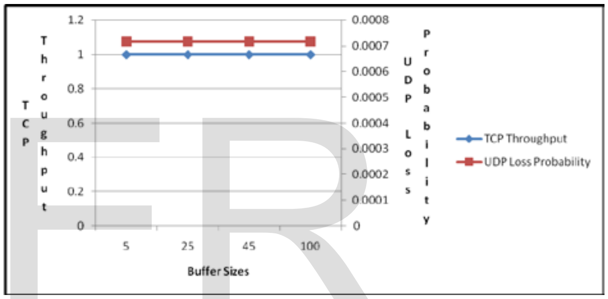


Fig. 11. Varying Buffer Sizes on Ring Topology with FQ

Here in Figure 11, a network of 25 Source- Destination pairs was simulated with 25 traffic flows of TCP and 1 Source-Destination pair with traffic flow of UDP. Here the topology used was Ring and queue type used was Fair Queuing. The throughput value of both TCP and UDP traffic remains constant.

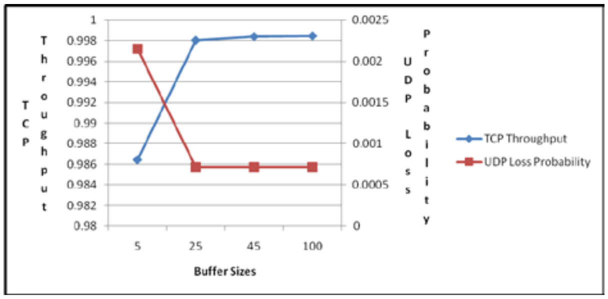


Fig 12: Varying Buffer Sizes on Ring Topology with RED

Here in Figure 12, a network of 25 Source- Destination pairs was simulated with 25 traffic flows of TCP and 1 Source- Destination pair with traffic flow of UDP. Here the topology used was Ring and queue type used was Random Early Detection. The throughput value of both TCP and UDP traffic remains constant.

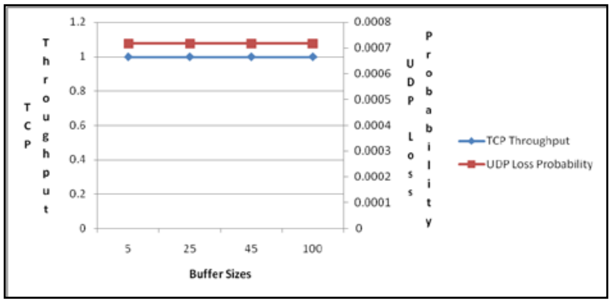


Fig. 13. Varying Buffer Sizes on Ring Topology with SFQ

Here in Figure 13, a network of 25 Source- Destination pairs was simulated with 25 traffic flows of TCP and 1 Source-Destination pair with traffic flow of UDP. Here the topology used was Ring and queue type used was Stochastic Fair Queuing. The throughput value of both TCP and UDP traffic remains constant.

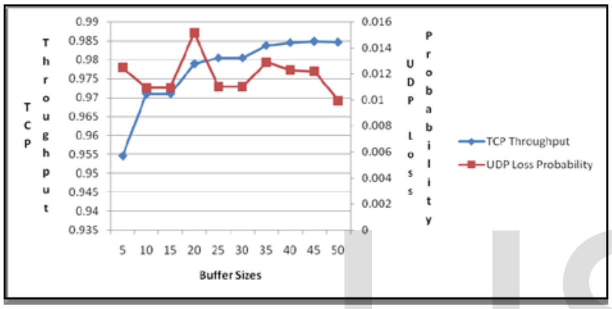


Fig. 14. Varying Buffer Sizes on Bottle Neck Topology with parameters like congestion window size and time to live with 5% of UDP Traffic

Here in Figure 14, a network of 95 Source- Destination pairs was simulated with 95 traffic flows of TCP and 5 Source-Destination pairs with traffic flow of UDP. Here the topology used was Bottle Neck and queue type used was Drop Tail. The throughput value of both TCP and UDP traffic kept on increasing as the Buffer Size was increased. But there's a drop in the UDP loss probability when the buffer size is kept as 45kB. Hence it seems favourable for both UDP and TCP traffic to maintain the buffer size between 35-40kB. Here parameters like congestion window size and time to live was also set. Congestion Window Size is set to be 15 frames and Time to live is set to be 10.

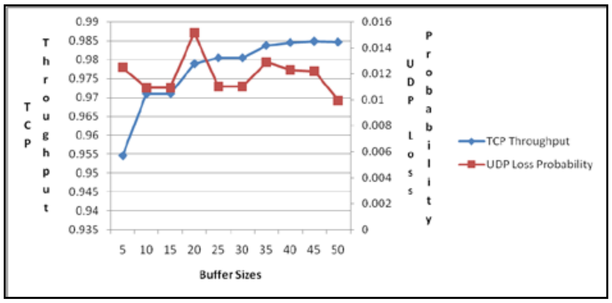


Fig 15: Varying Buffer Sizes on Bottle Neck Topology with parameters like congestion window size and time to live with 10% of UDP Traffic

Here in Figure 15, a network of 90 Source- Destination pairs was simulated with 90 traffic flows of TCP and 10 Source- Destination pairs with traffic flow of UDP. Here the topology used was Bottle Neck and queue type used was Drop Tail. The throughput value of both TCP and UDP traffic kept on increasing as the Buffer Size was increased. But there's a drop in the UDP loss probability when the buffer size is kept as 35kB. Hence it seems favourable for both UDP and TCP traffic to maintain the buffer size between 40-45kB.

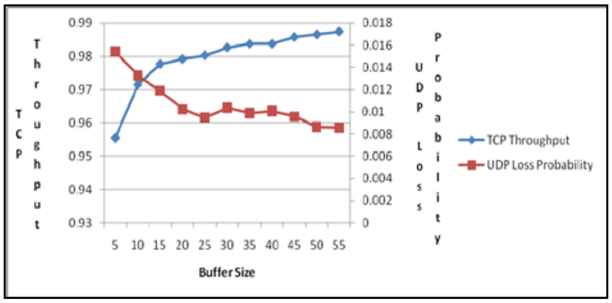


Fig. 16. Varying Buffer Sizes on Bottle Neck Topology with parameters like congestion window size and time to live with 15% of UDP Traffic

Here in Figure 16, a network of 85 Source- Destination pairs was simulated with 85 traffic flows of TCP and 15 Source- Destination pairs with traffic flow of UDP. Here the topology used was Bottle Neck and queue type used was Drop Tail. The throughput value of both TCP and UDP traffic kept on increasing as the Buffer Size was increased. But there's a drop in the UDP loss probability when the buffer size is kept as 30kB. Hence it seems favourable for both UDP and TCP traffic to maintain the buffer size between 35-45kB.

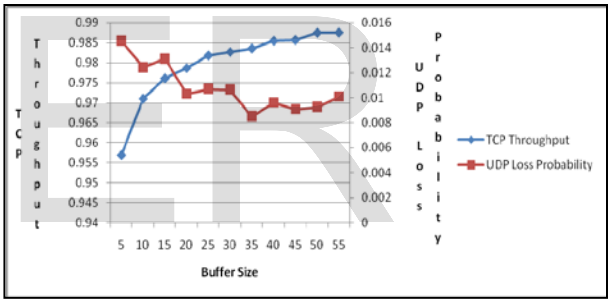


Fig 17: Varying Buffer Sizes on Bottle Neck Topology with parameters like congestion window size and time to live with 20% of UDP Traffic

Here in Figure 17, a network of 80 Source- Destination pairs was simulated with 80 traffic flows of TCP and 20 Source-Destination pairs with traffic flow of UDP. Here the topology used was Bottle Neck and queue type used was Drop Tail. The throughput value of both TCP and UDP traffic kept on increasing as the Buffer Size was increased. But there's a drop in the UDP throughput when the buffer size is kept as 15kB. Hence it seems favourable for both UDP and TCP traffic to maintain the buffer size between 35-45kB.

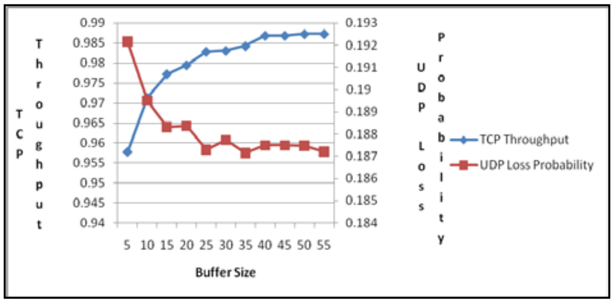


Fig 18: Varying Buffer Sizes on Bottle Neck Topology with parameters like congestion window size and time to live with 25% of UDP Traffic

For Figure 18, a network of 75 Source- Destination pairs was simulated with 75 traffic flows of TCP and 25 Source- Destination pairs with traffic flow of UDP. Here the topology used was Bottle Neck and queue type used was Drop Tail. The throughput value of both TCP and UDP traffic kept on increasing as the Buffer Size was increased. But there's a drop in the UDP throughput when the buffer size is kept as 15kB. Hence it seems favourable for both UDP and TCP traffic to maintain the buffer size between 35-45kB.

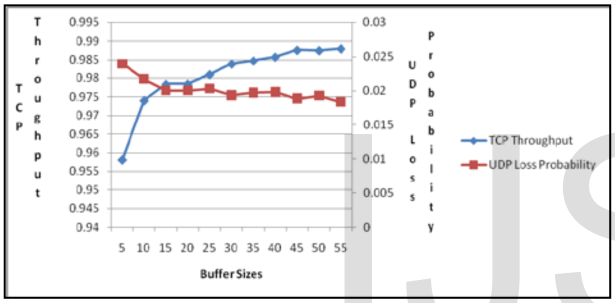


Fig 19: Varying Buffer Sizes on Bottle Neck Topology with parameters like congestion window size and time to live with 30% of UDP Traffic

For Figure 19, a network of 70 Source- Destination pairs was simulated with 70 traffic flows of TCP and 30 Source- Destination pairs with traffic flow of UDP. Here the topology used was Bottle Neck and queue type used was Drop Tail. The throughput value of both TCP and UDP traffic kept on increasing as the Buffer Size was increased. But there's a drop in the UDP throughput when the buffer size is kept as 25kB. Hence it seems favourable for both UDP and TCP traffic to maintain the buffer size between 30-45kB.

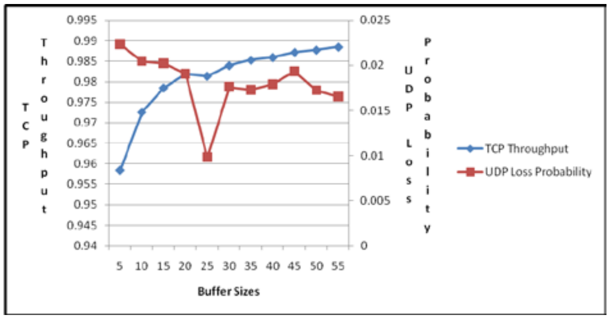


Fig 20: Varying Buffer Sizes Bottle Neck Topology with parameters like congestion window size and time to live with 35% of UDP Traffic

For Figure 20, a network of 65 Source- Destination pairs was simulated with 65 traffic flows of TCP and 35 Source- Destination pairs with traffic flow of UDP. Here the topology used was Bottle Neck and queue type used was Drop Tail. The throughput value of both TCP and UDP traffic kept on increasing as the Buffer Size was increased. But there's a drop in the UDP throughput when the buffer size is kept as 30kB. Hence it seems favourable for both UDP and TCP traffic to maintain the buffer size between 40-45kB.

5 Conclusion

The Router's Buffer Size is a very important parameter which has to be carefully set for lowering the congestion rates and thus lower the packet loss rates too. NS-2 is very useful tool for studying the packet losses by the trace files generated by the same.

Acknowledgment

I would like to thank the GOD Almighty for strengthening and guiding me each day.

I wish to express my profound sense of gratitude to Ms. Shilpa Verma, Asst. Professor, TSEC, for her guidance, constructive analysis and constant encouragement.

I would also like to thank my family and friends for all the love, encouragement and patience that you've showered on me.

References

[1] A. Vishwanath, V. Sivaraman, and G. N. Rouskas," Anomalous Loss Performance for Mixed Real Time and TCP Traffic in Routers with very Small Buffers",in IEEE /ACM Transactions on Networking, Vol. 19,No.4, August 2011,pp. 933-946Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[2] Jim Gettys, "Bufferbloat: Dark Buffers in the Internet", IEEE Internet Computing May/ June 2011 pp 94-95.

[3] Guido Appenzeller, Issac Keslassy, Nick Mc Keown,"Sizing Router Buffers" published in ACM SIGCOMM 04 pp 282-291.

[4] Ashvin Lakshmikantha, R Srikant, Carolyn Beck "Impact of File arrivals and departures on buffer sizing in core routers" published in Networking, IEEE Transactions, Vo 19,No 2, April 2011, pp 347-358.

[5] Arun Vishwanath, Vijay Sivaraman, Marina Thottan "Perspectives on Router Buffer Sizing: Recent Results and Open Problems" published in ACM SIGCOMM Computer Communication Review, Vol 39, Issue 2, April 2009, pp 34-39

[6] Mark E. Crovella, Azer Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes" published in IEEE/ACM Transactions on Networking Vol 5 No. 6 December 1997, pp 835-847.

[7] Jose Saldana, Julian Fernandez-Navajas, Jose Ruiz-Mas, Eduardo Virute Navarro, Luis Casadesus "The Effect of Router Buffer Size on Subjective Gaming Quality Estimators based on Delay and Jitter" published in 4th IEEE International Workshop on Digital Entertainment, Networked Virtual Environments and Creative Technology , 2012, pp 502-507.

[8] Mohit Agarwal, Navneet Tiwari, Lalla Atul Singh Chaurasia, Jatan Saraf "Queuing for Multisource Network- Type of Queue Decides Quality of Services of Network" presented in International Symposium on Computing, Communications and Control(ISCCC 2009), published in Proceedings of CSIT Vol 1 (2011), pp 293-297.

[9] Velibor Markovski, Fei Xue, and Ljiljana Trajkovi´ "Simulation and

analysis of packet loss in video transfers using User Datagram Protocol” published in the Journal of Supercomputing Vol 20, Issue 2, September 2001, pp 175-196.

[10] In Huh, Jae Yong Lee, and Byung Chul Kim “Decision of Maximum Congestion Window Size for TCP Performance Improvement by Bandwidth and RTT Measurement in Wireless Multi-Hop Networks” published in International Journal of Information Processing Systems, Vol.2, No.1, March 2006.

[11] Yinfei Pan “Design Routing Protocol Performance Comparison in NS2: AODV comparing to DSR as Example”