Performance Measurement of TCP and UDP Using Different Queuing Algorithm in High Speed Local Area Network

Manas Pratim Sarma

***Abstract*—In the current high speed networks, it is increasingly important to have mechanisms that keep throughput high but average queue sizes low. The queue management algorithm, which is applied to a router, plays an important role in providing Quality of Service (QoS). This paper present a Simulation-based performance Evaluation of two well known transport layer protocol TCP and UDP using two popular queue management methods; Random Early Detection (RED) and Drop Tail, in terms of throughput, queuing delay, packet drop rate and bandwidth utilization. In this paper we study the effect of buffer behavior on each one of these QoS measurements. TCP, UDP and shared topology scenarios are simulated by considering varying number of client case topologies and also the effect of performance in TCP and UDP with increasing the number of client. This simulation report is useful for implementing the queue management algorithm in router based on the traffic type and bandwidth.**

***Index Terms*—TCP, UDP, drop tail, RED queue, bandwidth delay product, throughput and end-to-end delay.**

I. Introduction

A. Difference between Transport Layer Protocol TCP and UDP as Follows

In this project TCP and UDP protocols are simulated and their performance is compared [1]. This comparison is mainly based on their congestion control and queue management mechanisms. TCP is a transport layer protocol used by applications that require guaranteed delivery. It is a connection oriented byte stream protocol. UDP is the connectionless transport layer protocol. The User Datagram Protocol offers only a minimal transport service nonguaranteed datagram. An application program running over UDP must deal directly with end-to-end communication problems that a connection-oriented protocol would handle. TCP is more reliable since it manages message acknowledgment and orders retransmissions in case of lost packets. UDP is a lightweight transport layer designed at top of IP. UDP uses a simple transmission model without implicit hand-shaking dialogues. TCP reads data as a byte stream and message is transmitted to segment boundaries. UDP messages are packets which are sent individually and on arrival are checked for their integrity.

TCP is used to control segment size, rate of data exchange, flow control and network congestion. Web browsing, email and file transfer are common applications that make use of TCP. TCP is preferred where error correction facilities are required at network interface level. UDP is largely used by time sensitive applications as well as by servers that answer small queries from huge number of clients. UDP is compatible with packet broadcast sending to all on a network and multicasting sending to all subscribers. UDP is commonly used in Domain Name System, Voice over IP, Trivial File Transfer Protocol and online games etc.

B. Difference between Application Layer Protocol FTP and CBR as Follows

File Transfer Protocol (FTP) is a standard network protocol operating on the application layer of the TCP Model. It is used to transfer files from one host to another host over a TCP [2]. FTP is built on Client-Server architecture and utilizes separate control and data connections between the client and server. Constant Bit Rate (CBR) service category is used for connections that transport traffic at a constant bit rate, where there is an inherent reliance on time synchronization between the traffic source and destination.

When too many packets are present in a part of a subnet its performance degrades leading to congestion. If all of a sudden, streams of packets begin arriving on a number of input lines and all need the same output line, the routers are no longer able to cope up. Hence a queue wills buildup and they begin losing packets. Congestion control has to do with making sure the subnet is able to carry the offered traffic. Load shedding is a congestion control mechanism in which when the routers are being inundated by packets that they cannot handle, they just throw them away. The queue management algorithm is one such way of deciding which packets to drop in a router to improve QoS measurements. Queuing management algorithm is responsible for packet admission control. Ideally the queue occupation level should be as low as possible, to ensure low delay; but it must also ensure maximum utilization of the outgoing link so that the queue is never empty.

Queue management can be classified into two categories; Passive Queue Management (PQM) and Active Queue Management (AQM) [3]. Drop tail is a representative of PQM algorithm which only sets a maximum length for each queue at the router. When the queue length is smaller than

Manuscript received February 14, 2013; revised April 10, 2013.

The author is with Computer Networks and Internet engineering Division, Centre for Development of Advanced Computing, Bangalore, India (e-mail: manassarma@cdac.in).

DOI: 10.7763/IJFCC.2013.V2.252

the maximum length, all packets are accepted, and if the queue reaches its maximum length all subsequent incoming packets are dropped until queue length decreases to be less than the maximum length. Drop tail routers can cause global synchronization problem, a phenomenon in which all senders sharing the same bottleneck router/link shut down their transmission windows at almost the same time. No indication of congestion until it's too late. May drop several packets at once-leads to global synchronization of flows. To solve this problem, AQM was proposed. One well known AQM algorithm is random early detection (RED). The Early Random Drop gateway drops arriving packets with a fixed drop probability if the queue length exceeds a certain drop level. A RED gateway detects incipient congestion based on the computation of the average queue size, and randomly drops or marks arriving packets before the gateway buffer gets full. It keeps the average queue size low, while allowing fluctuations in the actual queue size in order to accommodate bursty traffic and transient congestion. To avoid a bias against bursty traffic and the global Synchronization that exists in drop-tail gateways, the RED gateway uses randomization to choose which arriving packets to drop. The probability of dropping a packet from a particular connection is roughly proportional to that connection's share of the bandwidth through the gateway.

In RED two preset thresholds are used to detect incipient Congestion and control the average queue size. According to the estimated average queue length, a gateway operates in one of three different working states as shown in the Fig. 1. When the average queue length is less than the minimum threshold, the gateway is in the green state. All incoming packets are processed and forwarded properly, and no packet is dropped. When average queue length is between the minimum and maximum thresholds, the gateway is in the yellow state. Arriving packets are randomly dropped with a probability that is a function of the average queue length. When the average queue length is greater than the maximum threshold, the gateway is in red state in which every arriving packet is discarded. The behaviors of RED in green and red states are the same as those of drop-tail. Yellow is the key state in RED where the congestion-avoidance mechanism is implemented. The estimation of the average queue size and the calculation of drop probability are two key components of the RED algorithm. The success of RED depends on how to estimate the average queue size and set the drop probability. The filter used to compute the average queue size is an exponentially-weighted moving average [4].

$$avg \leftarrow (1-w_q)avg + w_q q \qquad (1)$$

where, w_q is a constant parameter preset by RED that determines the sensitivity of RED to the fluctuation of actual queue size, q is the actual queue size. The final packet-drop probability increases slowly as the number of received packets increases since the last marked/dropped packet.

II. Implementation Details

The scope of this project is to simulate the performance of TCP and UDP and also TCP-UDP shared topology using both drop tail and red queuing algorithms. This is done by considering varying number of client case topologies in different scenario [5].

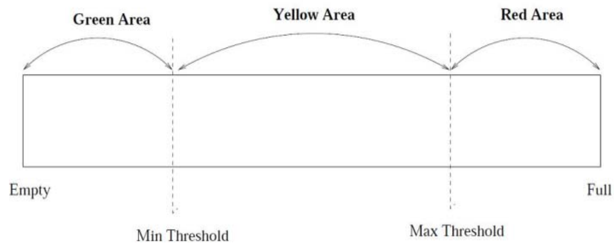

Fig. 1. Red queue three states.

```
If packet arrives {
    While buffer is not full{
                    Store the packet}
If queue length is max.queue{
                    Drop the packet}
                    }

Drop tail algorithm
---------------------------------------------------------------
for each packet arrival{
      compute the average queue size avg
      if minth <= avg < maxth{
            calculate probability Pa
            drop packet with probability Pa}
      else if maxth <= avg{
            drop the arriving packet}
}
RED algorithm
```

Table I specifies the how different types of application used using different protocol in different scenario. Different performance evaluation Metrics are considered to compare the performance of tcp and udp such as:

- Throughput
- Packet Loss Rate
- End to End Delay
- Bandwidth Utilization

TABLE I: Protocol and Application for Different Scenario

No. of client in the network	Protocol	Application	Shared Network
3 (non congested)	TCP , UDP	FTP, CBR	Yes
5 (congested)	TCP , UDP	FTP, CBR	Yes
8 (congested)	TCP ,UDP	FTP, CBR	Yes
10 (congested)	TCP ,UDP	FTP, CBR	Yes
25 (highly congested)	TCP	FTP	No
35 (highly congested)	TCP	FTP	No
50 (highly congested)	TCP	FTP	No

The productivity of the network is based on its throughput. Network throughput is the average rate of successful message delivery over a communication channel and the amount of traffic that a network can handle. The throughput is measured in bits per second (bps). The efficiency of an algorithm is measured in terms of the packets lost. Packet loss rate is measured as the percentage of the total number of dropped packets by the total packets transmitted over the link. The responsiveness of the network is limited by its delay. End-to-end delay refers to the time taken for a packet to be transmitted across a network from source to destination. It is the product of the number of links in the network multiplied to the sum of total transmission propagation, processing and queuing delays. It is measured in milliseconds (ms). Bandwidth is the maximum data transfer rate with in a link. Bandwidth Utilization refers to how effectively the entire bandwidth offered by the link. The Bandwidth Delay Product (BDP) determines the amount of data that can be in transit in the network. It refers to the Product of a data link's capacity (bits per second) and its end-and-end delay (in seconds). This can be used to determine the queue size for the queue management algorithms. A network with a large bandwidth-delay product is commonly known as a long fat network (LFN). A network is considered an LFN if its bandwidth-delay product is larger than 10^5 bits.

III. Simulation Setup

The simulations in this project are implemented on the Network Simulator tool (NS-2.34) in fedora 10 platform [6]. The simulated network consists of clients that are connected to a server through a single router. The clients are connected to the router via a10Mbps duplex link incorporating a delay of 3ms. The routed and server are linked with a 50Mbps duplex link having a 10ms delay. The packet size is fixed at 1500 bytes. The size of queue and the queue management algorithm is configured for each link. Either Drop Tail or RED queue management is implemented. The architecture of the topology is shown in Fig. 2.

IV. Configuring Topology

Then the traffic flow over the topology is established .This is done by defining the routing protocols and the applications that employ them. In this simulation TCP carries FTP traffic over its links.UDP supports CBR traffic for its clients. Three environments of clients were simulated [7].

Case i) All the hosts are considered to be TCP clients. These TCP clients send FTP packets.

Case ii) Here all clients are communicating using UDP connections which handle CBR traffic.

Case iii) Both TCP and UDP client share bandwidth. Here some clients communicate using TCP and some via UDP.

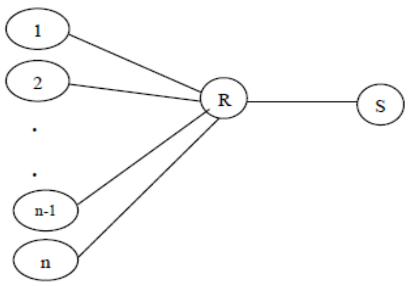

Fig. 2. Architecture of the network

where n is the highest number of client in the network.

A. Simulation Result

After simulating the scenario as in Fig. 2 with different cases the three parameter end to end delay, throughput and packet loss rate is computed and performance graph is plotted. In each case I am considering heavily congested scenario for TCP using Drop tail and Red queue, node (25, 35and 50).

B. End to end Delay

The following table and graph shows the end to end delay for both tcp and udp using drop tail and red queue.

In Fig. 3 up to node 6 no congestion in the network but when node increases to 10 then network becomes congested. So delay increase with the increase in node. Again for a heavily congested network i.e. when node number becomes up to 50 then using drop tail queue tcp suffers from high delay than using red queue. So red queue is better than drop tail queue for heavily congested network.

TABLE II: Delay Comparison of Udp and Tcp Using Drop Tail and Red Queue Management in Mille Second for Non Congested, Congested and Highly Congested Network.

	DROP TAIL QUEUE			RED QUEUE	
NODE	TCP	UDP	TCP-UDP	TCP	UDP
3	13.246	13.3	13.214	13.286	13.30
5	13.246	13.52	13.496	13.387	13.32
8	13.968	13.86	13.732	13.494	13.58
10	14.027	13.92	13.913	13.497	13.60
25	13.92			13.54	
35	14.46			13.61	
50	14.51			13.64	

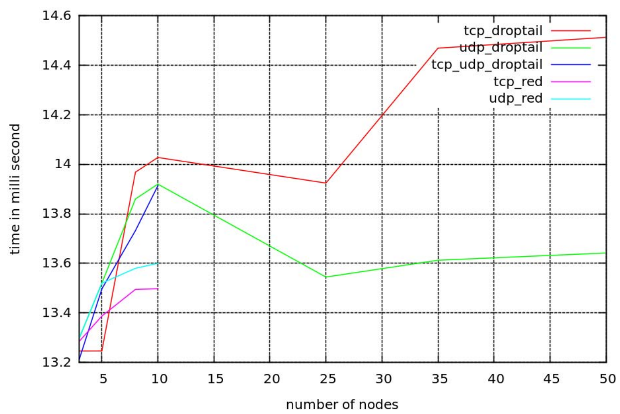

Fig. 3. Delay performance of tcp and udp using drop tail and red queue in non congested, congested and highly congested scenario.

C. Throughput

Case 1: Throughput in case of when all clients are either tcp or udp, The following table and graph shows the throughput for both tcp and udp using drop tail and red queue using number of client node from 3 to 50.

In Fig. 4 when number of node increases then throughput gradually decreases. UDP achieved better results in throughput, although its mean delay was higher compared to TCP. The reason UDP is faster than TCP is because there is no form of flow control or error correction and acknowledgement which also explains the fact that delay over UDP is higher compared to TCP. For heavily congested network when tcp uses drop tail queue throughput is higher than using red queue.

D. Packet Loss Rate

In the graph percentage of packet loss is plotted against the number of node in the network for tcp and udp using both drop tail and red queue.

In Fig. 5 initially when the number of node is less packet loss rate is almost zero for udp but when number of node increases i.e. for a congested network udp suffers from very high packet loss rate. For heavily congested scenario tcp suffers from high packet loss rate using red queue than drop tail queue.

TABLE III: THROUGHPUT COMPARISON OF UDP AND TCP USING DROP TAIL AND RED QUEUE MANAGEMENT IN KBPS FOR NON CONGESTED, CONGESTED AND HIGHLY CONGESTED NETWORK.

	DROP TAIL QUEUE		RED QUEUE	
NODE	TCP	UDP	TCP	UDP
3	9687.62	9998.04	8827.97	9998.04
5	9687.26	9997.94	8840.07	9997.94
8	6558.24	6248.71	5176.14	6248.71
10	4564.16	4998.97	4174.76	4998.97
25	1805.83		1623.59	
35	1281.28		1235	
50	898.677		881.935	

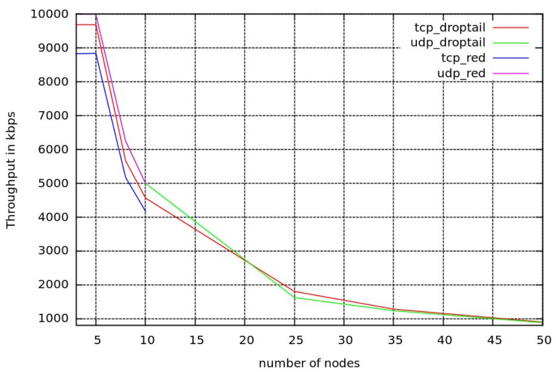

Fig. 4. Throughput performance of tcp and udp using drop tail and red queue in non-congested and congested and highly congested scenario.

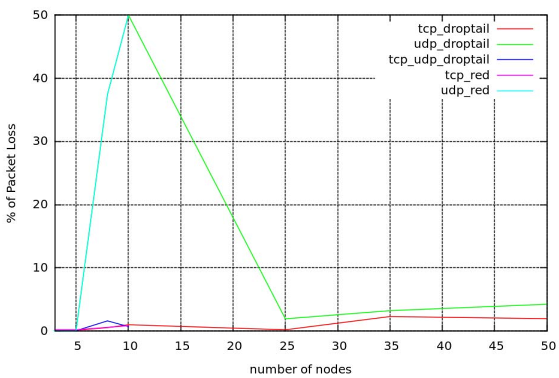

Fig. 5. Packet loss percentage measurement of tcp and udp using drop tail and red queue in non congested, congested and highly congested scenario.

E. Utilization

Network utilization is the percentage of the total bandwidth that is being used at a particular point of time. It is defined as

$$\%utilization = ((\text{total size of data bytes}) / \text{Bandwidth} \times \text{time interval}) \times 100 \quad (2)$$

F. Bandwidth Delay Product Calculation

In our simulated network BDP for router to server link is 4×10^6 bits. So this type of Ultra-high speed LANs falls into LFA network. Using BDP we can define the buffer size of the queue.

$$BDP = \text{link capacity (BW)} \times RTT \quad (3)$$

$$\text{And Throughput} <= \text{Buffer size} / RTT \quad (4)$$

$$\text{So Buffer size} >= \text{Throughput} \times RTT \quad (5)$$

In our simulated network required buffer size should be greater than or equal to 512 Kbyte to reach 50 Mbps throughput.

In the Fig. 6 and Fig. 7 the link utilization for tcp and udp initially increases up to a certain point but increase in number of node the link utilization becomes decreases and constant after some time. For both cases udp link utilization is higher than tcp.

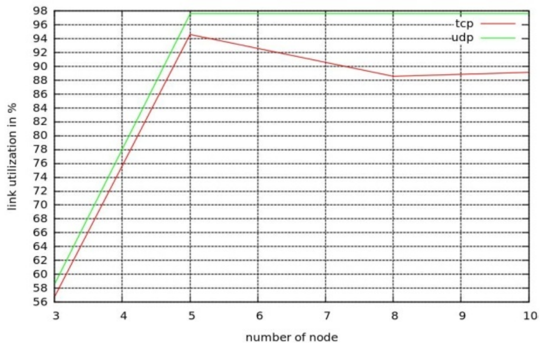

Fig. 6. Link utilization of tcp and udp Using drop tail queue.

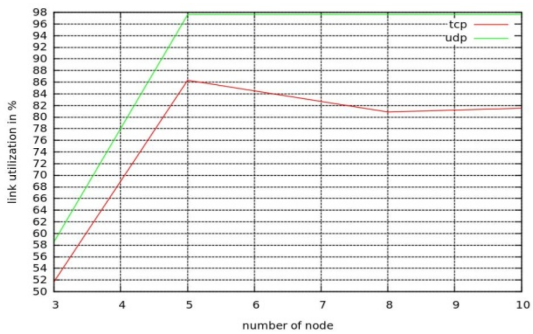

Fig. 7. Link utilization of tcp and udp using red active queue management.

TABLE IV: PACKET LOSS RATE FOR UDP AND TCP USING DROP TAIL AND RED QUEUE MANAGEMENT IN TERMS OF PERCENTAGE FOR NON CONGESTED, CONGESTED AND HIGHLY CONGESTED NETWORK.

	DROP TAIL QUEUE			RED QUEUE	
NODE	TCP	UDP	TCP-UDP	TCP	UDP
3	0.1055	0	0.05	0.1794	0
5	0.1090	0	0.05	0.1769	0
8	0.5367	37.44	1.611	0.6828	37.44
10	0.9930	49.99	0.7016	0.8323	49.99
25	0.19			1.93	
35	2.28			3.22	
50	1.95			4.23	

V. CONCLUSION

In this project we simulate almost all possible combination of TCP and UDP using Drop Tail queue management algorithm and RED queue management algorithm in different aspects with the number of client varies. The simulation results show RED outperforms Drop Tail in terms of queuing delay, and packet drop rate. However the efficient performance of drop tail in critical network applications with respect to some metrics cannot be ignored when we consider throughput. The performance of the queue management algorithms also depends upon the protocols upon which they are applied, i.e. TCP or UDP .the type of topology of the network; whether it is a shared topology of UDP and TCP; or purely one kind of topology of the clients also bears influence on the performance of buffer management. For some real time application (VoIP) UDP using Red queue will give better performance and for reliable delivery of packet TCP using Red queue is better other protocol and queue in High speed LAN.

VI. Future Work

In this paper we simulate the existing flavor of TCP, UDP and two well known queue management algorithm Drop tail and Red queue with number of source node varies. Next stage I will simulate different queuing algorithm with TCP congestion control mechanism and I am implementing a new queue mechanism which may give better performance in TCP congestion control than the existing queue mechanism [8].

References

[1] J. Wechta, M. Frickers, and F. Halsall, “Hop-by-Hop low as method to improve QOS in 802.3 LANs,” *Seventh International Workshop on Quality of Service*, *IWQoS '99*, pp. 239-247, 1999.

[2] M. Casonia, “TCP performance in hybrid EPON/OBS network,” in *Proc. International Conference on Optical Network Design and Modeling*, pp. 1-4, IEEE Xplore 18-20 Feb. 2009.

[3] A. Dana and A. Malekloo, “Performance comparison between Active and passive queue management,” *IJCSI International Journal of Computer Science Issues*, pp. 13-17, vol. 7, no. 3, no 5, May 2010.

[4] H. N. Wang and K. G. Shin, Refined Design of Random Early Detection Gateways, in *Proc. Global Telecommunications Conference - Globecom'99*, pp. 769-775, 1999.

[5] K. Fall and K. Varadhan, “The ns manual (formerly ns Notes and Documentation),” The VINT Project-A Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox Parc.

[6] M. P. Sarma. Computer Networks and NS2 blog, sample-tcl-code-for-lan-simulation. [Online]. Available: http://www.ns2bymanas.blogspot.in

[7] S. Patel, P. Gupta, and G. Singh “Performance measure of drop tail and RED algorithm” in *Proc. 2010 2nd International Conference on Electronic Computer Technology (ICECT 2010)*, pp. 35-38, 2010.

[8] K. Fall and K. Varadhan. The VINT Project, UC Berkley, LBL, USC/ISI. [Online]. Available: http://www.isi.edu/nsnam/ns/doc/index.html

Manas Pratim Sarma is currently working at Center for Development of Advanced Computing (CDAC), Bangalore, INDIA in Research and Development. His areas of interest are Network performance measurement, Network architecture design, Grid Computing, NMS software development, Cyber Security. He Completed his Bachelor Degree in Information Technology from North-Eastern Hill University, Shillong, INDIA in 2011. He publishes one Journal in Green Computing in IT and he is EC-COUNCIL recognize Certified Ethical Hacker.