ECE 6110 – CAD for Communication Networks
Lab 2 – Comparison of Random Early Detection (RED) vs. Drop Tail Queuing
By Yash Shah

I. Introduction

A simple dumbbell style topology was constructed containing two bottlenecks using either Random Early Detection (RED) or Drop Tail queues. The topology contains three source node applications, one using UDP and the other two using TCP protocol. Figure 1 shows the network topology with the bandwidth and delay properties of each channel link. The sources all start sending data on a 15 Mbps with 5 ms delay. The link between the first two routers forms the 1st bottleneck with 10Mbps link speed, while the second router link has 1Mbps link speed. Finally, three sink nodes, each dedicated to a separate source application receives all the data on a channel with 10 Mbps bandwidth and 1ms delay. The simulation was run for 10 seconds.

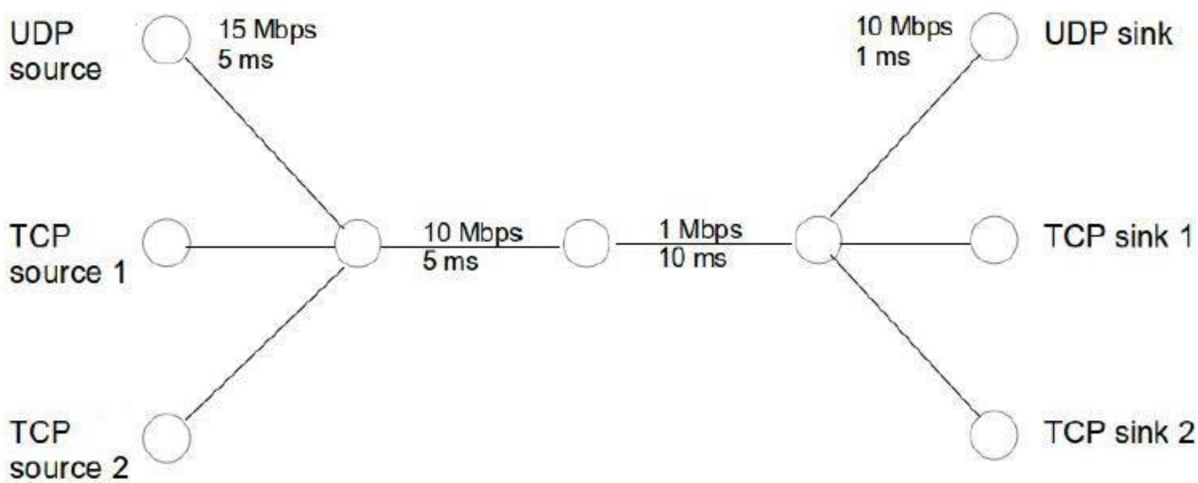


Figure 1: Network Topology

RED queuing involves some more sophisticated techniques that attempt to overcome some of the flaws of Drop Tail queuing. RED queues start dropping packets when the queue gets sufficiently filled before congestion occurs with the assumption that congestion is going to occur soon. This attempts to reduce excessive packet loss by dropping packets probabilistically before queues overflow.
This probabilistic dropping of packets helps prevent uneven flows from occurring. On the most basic level, RED queues start dropping packets probabilistically when the average queue size is greater than the minimum threshold (minTh) and less than the maximum threshold (maxTh). When the average queue size becomes greater than maxTh, forced packet drops begin occurring. This signifies persistent congestion.

Drop Tail queuing follows a First In First Out (FIFO) queueing system, where lockout can occur where a small number of flows can take over the queue during congestion. This happens because slower flows reach queues when they are full and packets get dropped as compared to faster flows. Latency is increased for all flows when the queue is constantly full.

In this experiment, four RED parameters were considered: minimum threshold (minTh) (kept at 50% of queue size), maximum threshold (maxTh)) (kept at 80% of queue size), maximum probability (maxP) and weight (Wq). For Drop Tail, the queue size was varied.

II. RED vs. Drop Tail Queuing Setup and Assumptions

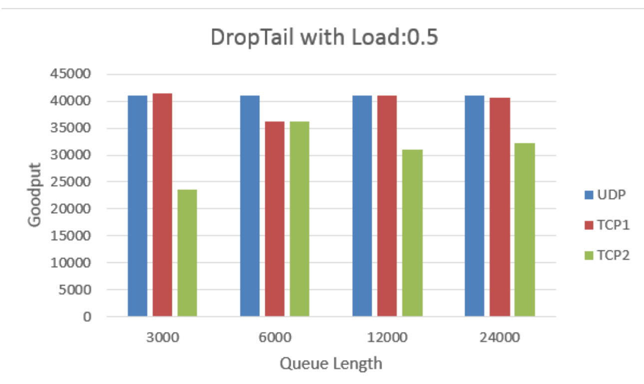
First, the network was simulated using Drop Tail queues at the two bottlenecks. During the simulation the throughput was measured as a function of the queue length of the Drop Tail queue. Three different network loads were simulated. The assumptions made for the simulations were:

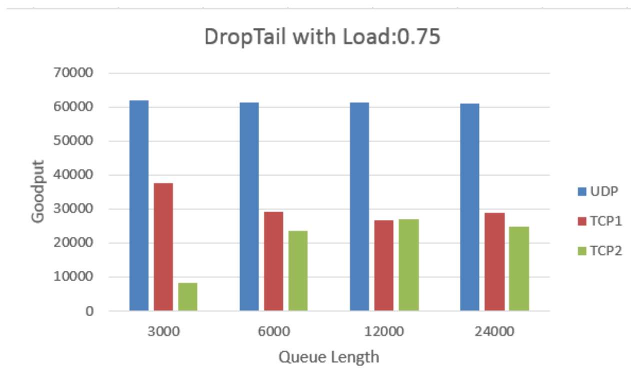
1. The receiver windows are always large enough.
2. The Send-applications are always on.
3. Data is transmitted out at a constant rate in bits/second.
4. The load stays constant on the network.
5. 10 seconds of simulation is enough to obtain general representative data.

Same Assumptions hold for the simulation of RED queueing protocol.

III. Results

From Figure 2 showing the Drop Tail queue results, we observe that UDP applications results in much higher throughput than the TCP application. This makes sense because the smaller load means less build-up of the queues and less dropped packets. The TCP throughputs consistently decreases as network load increase in packet drops. While the throughput of the UDP application increases proportional to the decrease in TCP throughput. This is probably because of UDP's connectionless nature and lack of acknowledgements.





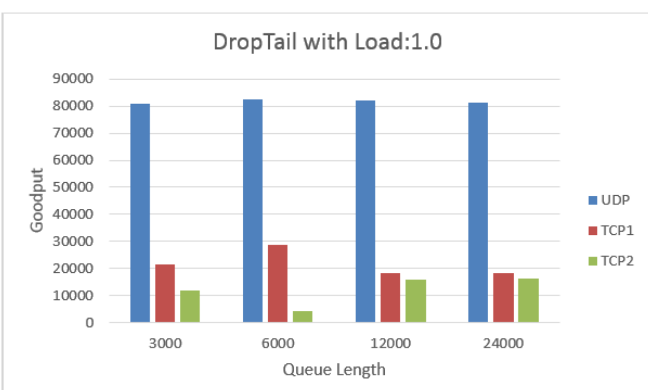


Figure 2(a,b,c): Droptail Goodput for varying Queue Sizes

Figure 3 shows the results of simulating the network with RED queues. The throughput of the UDP application stayed constant with varying queue sizes and constant load, while the TCP throughput varied under the same conditions due to one flow acquiring more network resources filling up the queue leading to other links increase in packet drops.

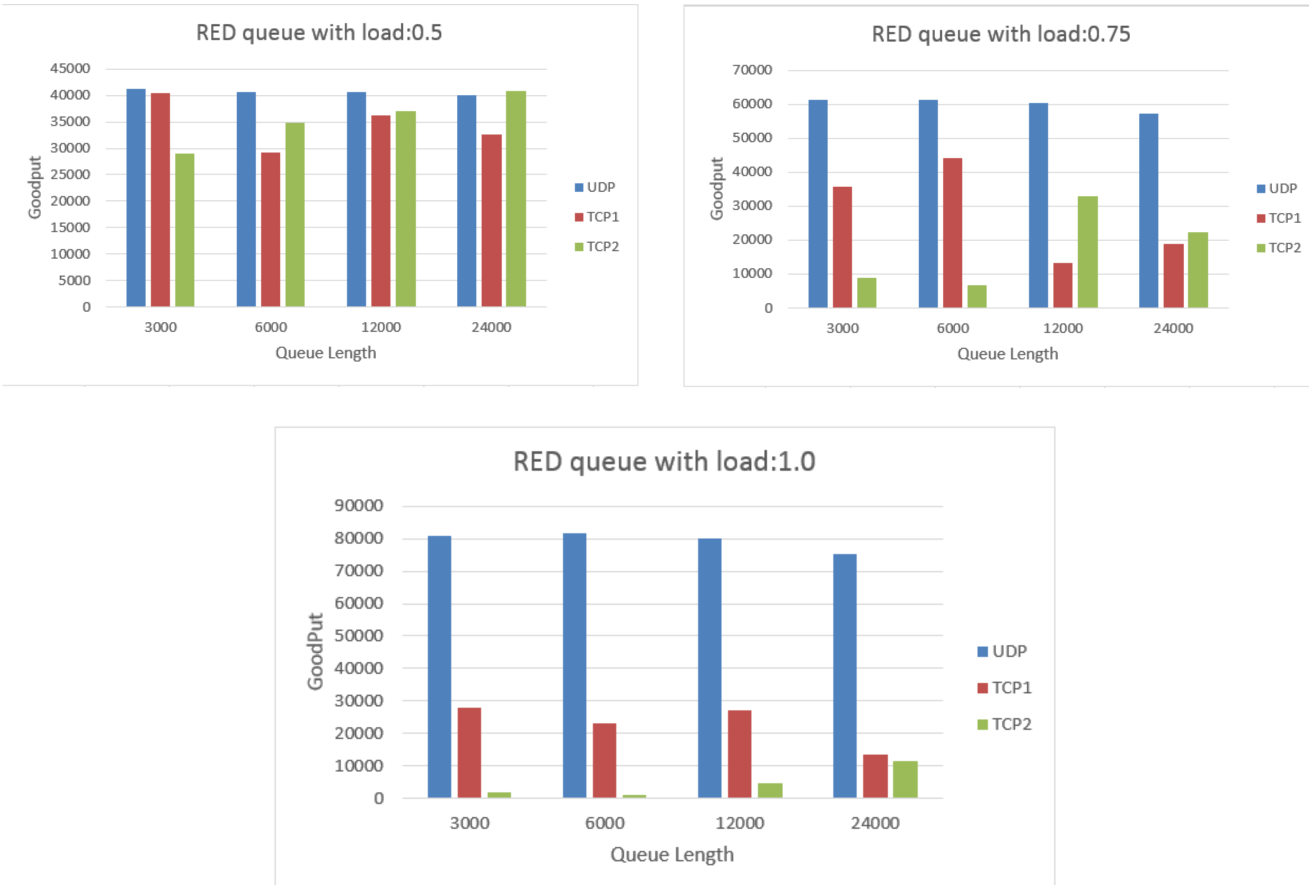


Figure 3(a,b,c): RED Goodput with varying Queue size and load

The trend for the TCP applications was a decrease with increased load. This signifies that smaller loads resulted in less filling of queues. With UDP application completely outperforming the TCP applications by a significant margin.
Comparing the results from RED queuing versus Drop Tail queuing the results for the UDP application the observed throughput were very similar. The throughput of the TCP applications for RED queuing at 50% load was slightly better than Droptail queue. While larger loads TCP links in Droptail queue performed better than RED. This was unexpected because RED queuing was designed to handle non-ideal situations better than Drop Tail.

IV. Conclusion

Overall, the performance of the simulations were unexpected. Primarily, the TCP applications in the RED simulations performed much worse than those in the Drop Tail simulations. This is likely due to errors in the simulation, as discrepancies between Drop Tail and RED queuing should never be this significant. The UDP application, at times, beat out the TCP applications. This is most likely due to the fact that the load was split between the three, but also because UDP protocol lacks acknowledgements and is connectionless, therefore can gain higher throughput than TCP due the lack of ACK overhead.