LAB 9

DropTail vs RED:
https://ijarcce.com/wp-content/uploads/2012/03/IJARCCE4A-a-shubhangi-Comparison-Analysis-of-Different.pdf
https://www.diva-portal.org/smash/get/diva2:831714/FULLTEXT01.pdf

Q: What is TCP synchronization problem? In which case it is more prominent.

A: TCP global synchronization in computer networks can happen to TCP/IP flows during periods of congestion because each sender will reduce their transmission rate at the same time when packet loss occurs.

This causes problems when used on TCP/IP routers handling multiple TCP streams, especially when bursty traffic is present. While the network is stable, the queue is constantly full, and there are no problems except that the full queue results in high latency. However, the introduction of a sudden burst of traffic may cause large numbers of established, steady streams to lose packets simultaneously.

TCP has automatic recovery from dropped packets, which it interprets as congestion on the network (which is usually correct). The sender reduces its sending rate for a certain amount of time, and then tries to find out if the network is no longer congested by increasing the rate again subject to a ramp-up. This is known as the slow-start algorithm.

Q: Which queuing mechanism (Drop Tail or RED) is better for bursty traffic?

A: RED is better. DT introduces global synchronization in several connections, when the packets are dropped.

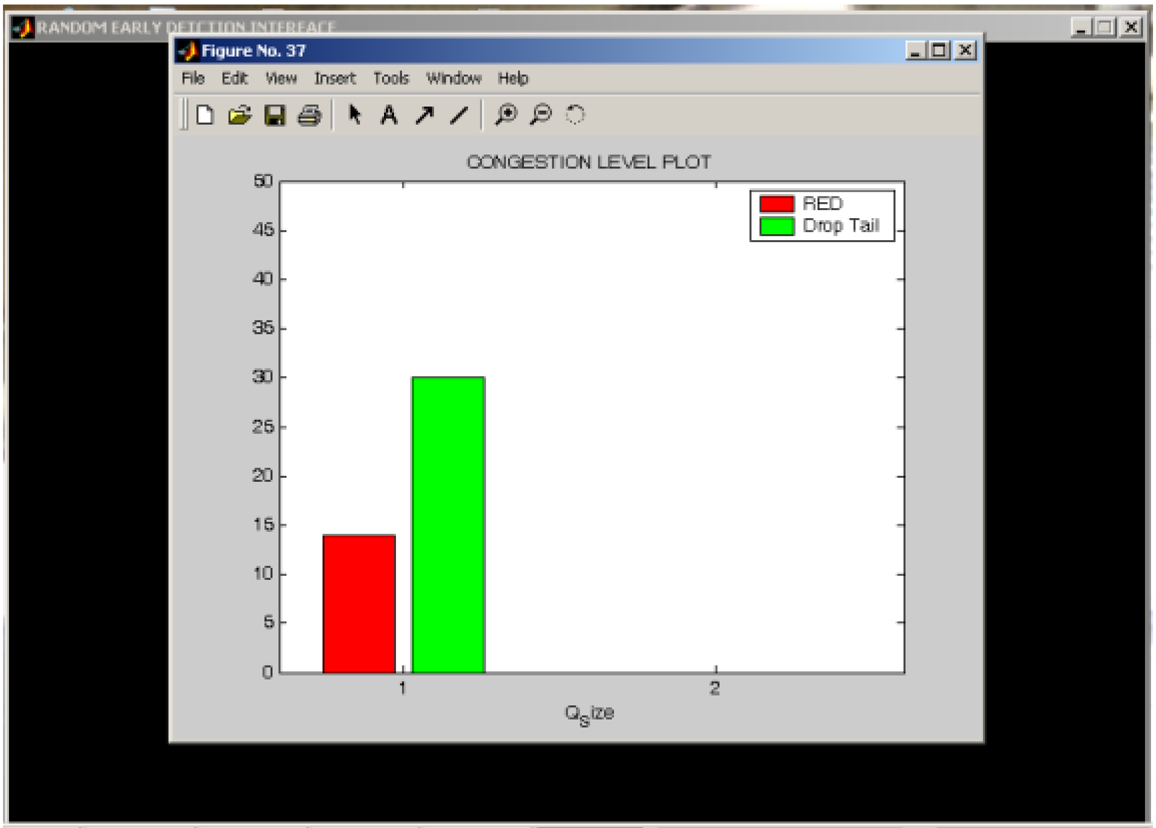
No Global Synchronization - By marking packets for early discard, the number of consecutive drops can be reduced. Many Internet designers were concerned that consecutive drops when queues became full could cause global instability in the network as many queues signal their source to reduce their window at the same time

Fairness – Reduces the bias against bursty traffic, as mentioned earlier. RED will avoid a situation in which bursty traffic faces extreme packet loss compared to smooth traffic. DT will favour smooth traffic and has bias against Bursty traffic. RED will also affect UDP packet streams,

Q: Observe the Queue size variation in both the cases. Compare both the methods w.r.t queuing delay.

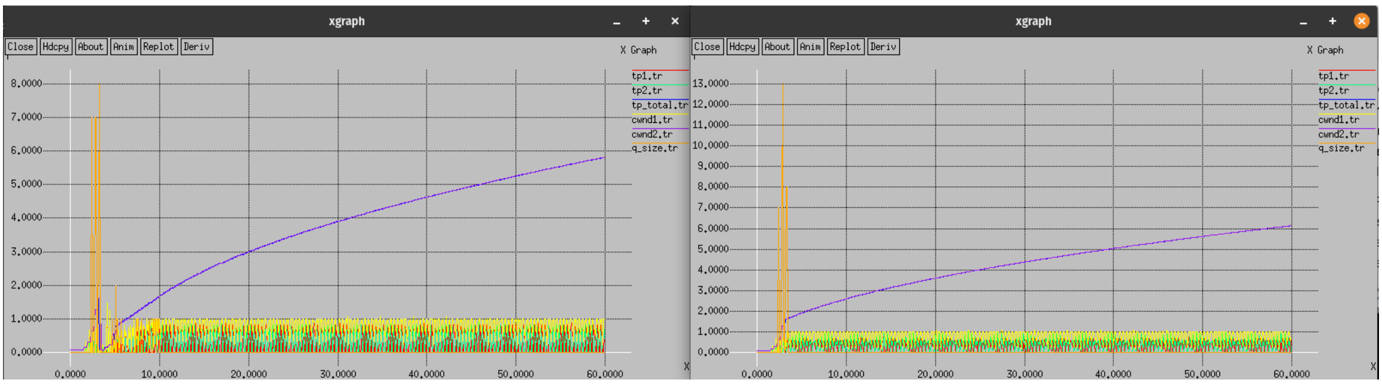
In RED, the queue size is usually lesser as packets can be dropped after their size reaches a min threshold.

Congestion level plot below shows that the red color about RED buffer level and green color about DT buffer level.



Display: RED Interface window congestion level plot

RED vs Droptail



Congestion Avoidance – RED allows for queue congestion to be managed before a critical overflow point is reached. Also, keeping the queue size lower **decreases delay for those packets that are not dropped.**

Q: How does changing the RED parameters affect the plots?

A:

Parameters of RED:

'Wq' is an exponentially weighted average factor, which determines the time constant for the averaging of the queue size. If 'wq' is too low, then the estimated average queue size is probably responding too slowly to transient congestion. If 'wq' is too high, then the estimated average queue size will track the

instantaneous queue size too closely, and may result in an unstable
system.
The optimal values for qmin and qmax depend on the desired average queue size. If the typical traffic patterns are fairly bursty then qmin must be correspondingly large to allow the link utilization to be maintained at an acceptably high level. This value should also be associated with the buffer size of the network. If qmin is set too small, it leads to low bandwidth utilization.
Conversely, if qmin is set too high, it may result in unfair competition for bandwidth among multiple links, thereby canceling out the benefits of the RED algorithm. The optimal value for qmax depends in part on the maximum average delay that can be allowed by the gateway. The RED functions most effectively when the difference between qmax and qmin is larger than the typical increase in the calculated average queue size, in one roundtrip time. A useful rule of thumb is to set qmax to at least twice the size of qmin. If the difference between qmax and qmin is set too small, then the computed average queue size can regularly oscillate up to qmax. This behavior is similar to the oscillations of the queue size up to the maximum queue size experienced with DT gateways.

Q: Can you guess the default advertInterval approximately using the trace files? (Hint: Run simulation without link-failure).

A: Approx 2 seconds

Q: Is it possible to simulate the Count-to-Infinity problem using DV protocol in the given topology?

A: The main issue with Distance Vector Routing (DVR) protocols is Routing Loops, since Bellman-Ford Algorithm cannot prevent loops. This routing loop in DVR network causes Count to Infinity Problem. Routing loops usually occur when any interface goes down or two-routers send updates at the same time.
https://www.geeksforgeeks.org/route-poisoning-and-count-to-infinity-problem-in-routing/