Project 2 – Comparison of RED vs. DropTail Queuing

Zhongyi Luo
903141808

Simulator: ns-3.24.1 Platform: Mac OSX 10.10.5, deepthought

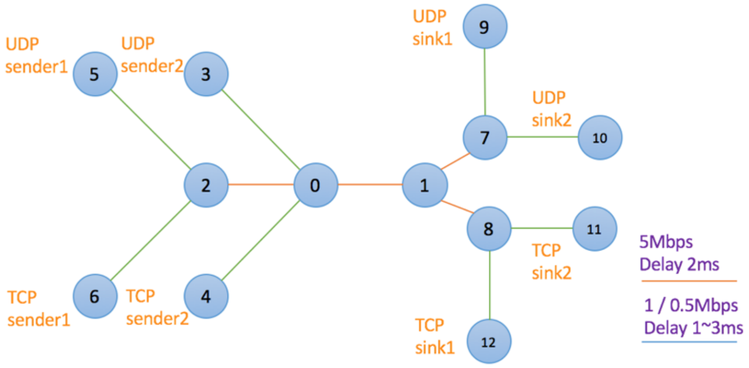
Global Variable:

QueueMode: {RED, DropTail}; WindowSize: {2000, 8000, 16000, 32000, 64000}; QueueSize: {2000, 8000, 16000, 32000, 64000}; RouterLink (BottleLink) Delay: {10ms, 20ms, 30ms}; RouterLink(BottleLink)Rate: {0.5Mps, 1Mps};

RED related parameter:

Threshold setting as ratio of queue limit. According to the demonstration of ns3 tutorial and paper *Tuning RED for Web Traffic*, Max-threshold should be at least two times of Min-threshold, hence I used following value in pairs:
Max-threshold: {0.04, 0.08, 0.16, 0.32, 0.48, 0.64, 0.8, 0.8, 0.96, 0.85} * QueueLimit;
Min-threshold: {0.02, 0.04, 0.08, 0.16, 0.24, 0.32, 0.32, 0.4, 0.48, 0.6} * QueueLimit.
Wait: {true, false} -------- whether to wait between dropping consecutive packets
Gentle: {true, false} -------- whether to increases dropping probability slowly when average queue exceeds Max-thresh
QueueLimit took the same set of value as QueueSize in General Parameter.

1. Topology.



UDP sender rate is 500kb/s, packet size is 1024.
On & Off time uses ns3 constant random variable.
TCP sender segment size is 512.

2. Comparison between DropTail and RED

Assumption:

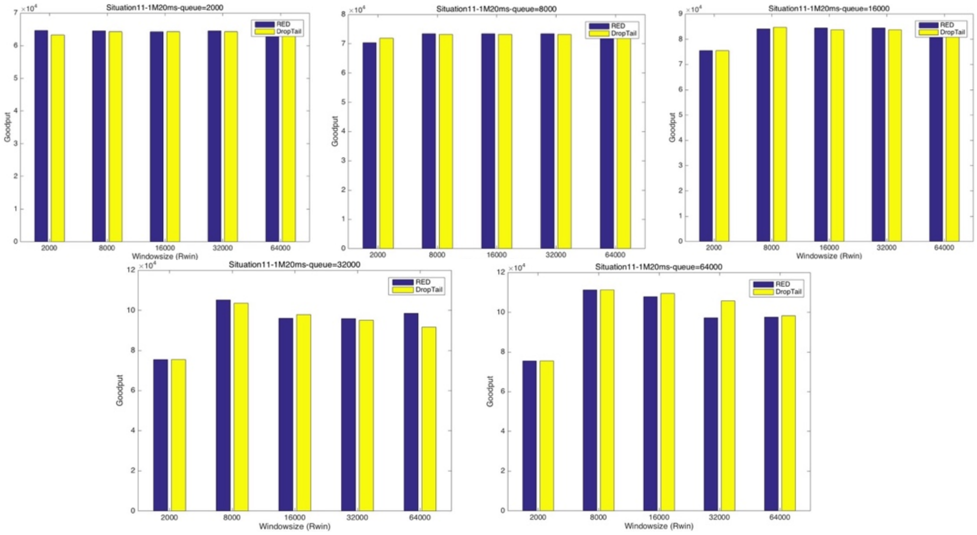
(1) There can be multiple choices in RED settings, giving it much more flexibility in performance. Paper *Tuning RED for Web Traffic* pointed out the possible difference between good and bad RED parameter choices. Therefore, to compare RED and DropTail, I consider RED better than DropTail if there exists one setting among all that make RED goodput higher than DropTail.

(2) Overall goodput refer to the sum of TCP and UDP, compare between TCP and UDP is not the focus of my experiment.

.

2.1 Comparison under different Window Size (Rwin)

Following is the result of varying window size from 2000 to 64000, under given Queuesize. (Wait and Gentle mode are set to true; Bottleneck delay 20ms, DataRate 1M):



Observation:

(1), RED (Optimal is considered out of ten set of threshold setting) and DropTail genrally have similar trend under variation of window size. Looking into origin data, there are 17 cases out of 25 that RED performs better than (or equal to) DropTail.

(2), In the case of WindowSize 64000 and QueueLimit 32000, the RED outperforms DropTail with largest amount; in the case of WindowSize 32000 and QueueSize 64000, the DropTail outperforms RED with largest amount.

Explanation:

(1), From this result, when QueueLimit is small, Receiver Window is not the major factor that brings difference between RED and DropTail Mode. Except for suppressing performance of both when it equals 2000. Low receiver WindowSize suppress both the goodput of RED and DropTail.

Interestingly, after QueueSize reaches 16000, RED and DropTail have same results in the cases of WindowSize=2000. This means when Receiver Window is small, the traffic load is low, most transient queuelength of RED didn't even reach the Min-Threshold of those "large Queue" (Limit >16000).

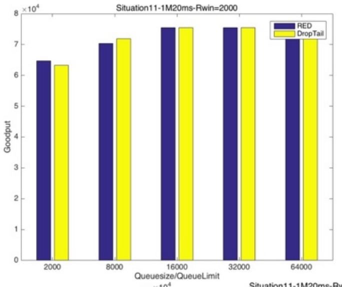
RED performs better in 17 cases out of 25, this roughly means RED has can perform better than DropTail in most circumstances, given proper parameter.

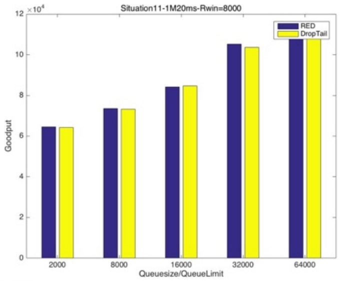
(2), For these two cases, I assume RED can perform better given heavier traffic and longer latency. It's natural since RED has set different probability to drop/mark packets before they reach the actual limit of queue. And this helps it to avoid congestion more efficiently than DropTail Mode.

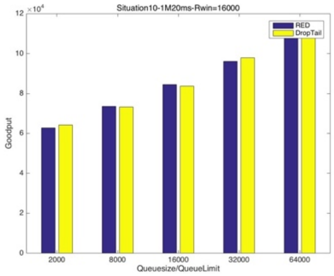
When DropTail performs better than RED, it means congestion can be avoided appropriately using DropTail, and RED would lead to unnecessary packet drop. RED QueueLimit is same as DropTail QueueSize, making it drop more packet than DropTail in my setting.

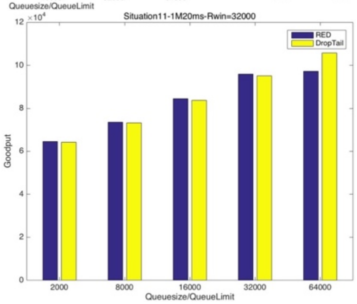
2.2 Comparison under different Queue Size

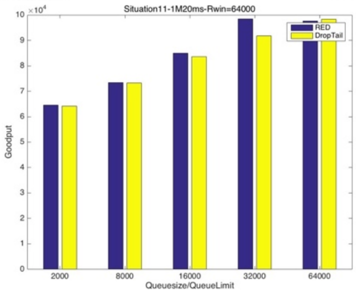
Following is the result of varying QueueSize from 2000 to 64000, under given WindowSize. (Wait and Gentle mode are set to true; Bottleneck delay 20ms, DataRate 1M):











Observation:

(1), Given Fixed WindowSize, as Queuesize increases, both RED and DropTail mode goodputs are in a non-decrease trend.

(2), For the 15 cases of QueueSize <= WindowSize, only 1 indicates DropTail outperforms RED.

Explanation:

(1), For both RED and DropTail situation, when Receiver window size is fixed, the potential maximum traffic load is also determined. As queue size/queue limit increases, it can hold more packets, which means lower possibility to drop packets and trigger retransmission. Hence, the increase of goodput along with queue size is natural. When the goodput stop increasing after Queue length reach certain value, it means time for packets to wait in a queue increased and become comparable with time for retransmission.

(2), For these 15 cases, I assume they describe the situation when Queue setting is not that enough for heavy traffic load and congestion are detected more often.

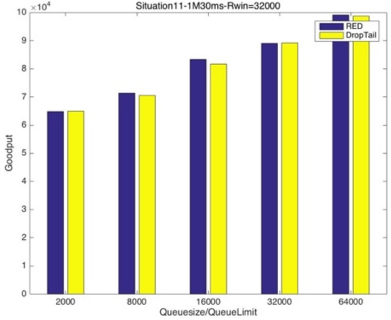
In my experiment, I choose same value for RED QueueLimit as QueueSize. Hence, the actual utilized size of RED queue is actually smaller than that of DropTail. On one hand, RED helps better to avoid severe congestion ("good side"); on the other hand, RED queue can drop more packets than DropTail even when there's no congestion ("bad side").

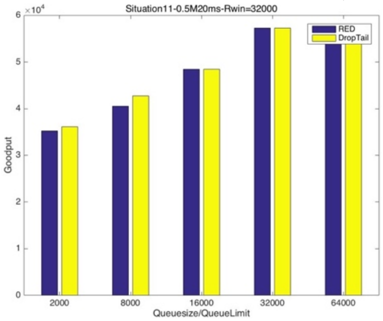
Hence, in these 15 cases, the "good side" of RED has more influence than its "bad side", providing more goodput than DropTail queue.

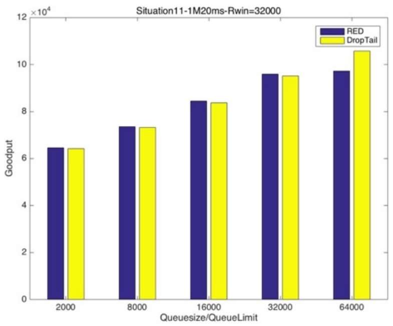
2.3 Comparison under different Delay (RTT) and Rate

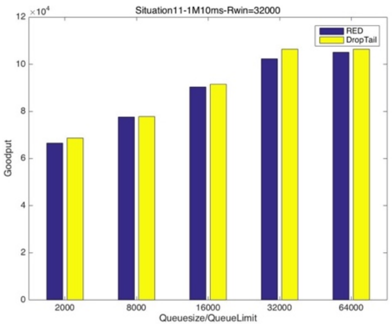
In my experiment, to find out influence of traffic load and latency, I tried 4 settings for bottleneck link: 1Mbps data rate & 30ms delay; 0.5Mbps data rate & 20ms delay; 1Mbps data rate & 20ms delay; 1Mbps data rate & 10ms delay.

Results are as follows: (WindowSize=32000, "Wait" and "Gentle" are true).









Observation:

(1) Cutting the link rate by half result in losing half of the goodput when window size is small; losing less than half goodput when window size is large.

(2) In the case of link delay = 30ms, RED goodput is better than (or equal to) DropTail Goodput at any queue limit; in the case of link delay 10 ms, DropTail Goodput is better than (or equal to) DropTail Goodput at any queue size.

Explanation:

(1) When data rate is cut by half (0.5M), the same amount of data requires almost twice of the time to be sent. Hence, when window size is low (congestion influence is not obvious), goodput also nearly cut by half. However, in large window size (large traffic load) cases, slower data rate also slower the procedure to fill up the queue. Thus makes its goodput larger than half of 1M settings.

(2) From the second finding, it can be seen that, with a same set of threshold setting, RED queue mode performs better than DropTail when RTT is longer (compared to 20ms).

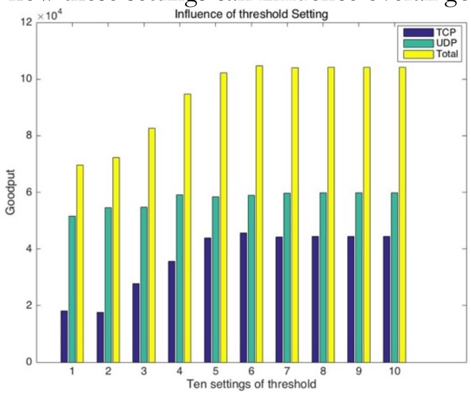
Following is the discussion over the choice of RED parameters.

3. Performance under different RED parameter

3.1 Performance with different Max & Min threshold

10 set of Max & Min threshold was used in this experiment.
Max-threshold: {0.04, 0.08, 0.16, 0.32, 0.48, 0.64, 0.8, 0.8, 0.96, 0.85};
Min-threshold: {0.02, 0.04, 0.08, 0.16, 0.24, 0.32, 0.32, 0.4, 0.48, 0.6}.
These are ratio values, which means:
Maxpacket = Max-threshold * QueueSize;
Minpacket = Min-threshold * QueueSize;
Config :: SetDefault ("ns3::RedQueue::MinTh", DoubleValue (Minpacket));
Config :: SetDefault ("ns3::RedQueue::MaxTh", DoubleValue (Maxpacket));

Following shows how these settings can influence overall goodput:



Observation:

(1), When UDP goodputs does not fluctuate much with the threshold setting, TCP has shown a non-decrease trend as threshold value increases.

(2), The maximum value was gained at Min-Thresh=0.32 & Max-Thresh=0.64.

Explanation:

(1), Unlike TCP, UDP does not handle flow control, acknowledgement as well as inherent order. It is completely unconcerned about packet losses. Hence, unless the traffic load or congestion state influence UDP indirectly, RED Queue settings won't have too much impact on UDP goodput. As for the trend of TCP goodput, it can be a result of increasing size of "usable" RED queue.

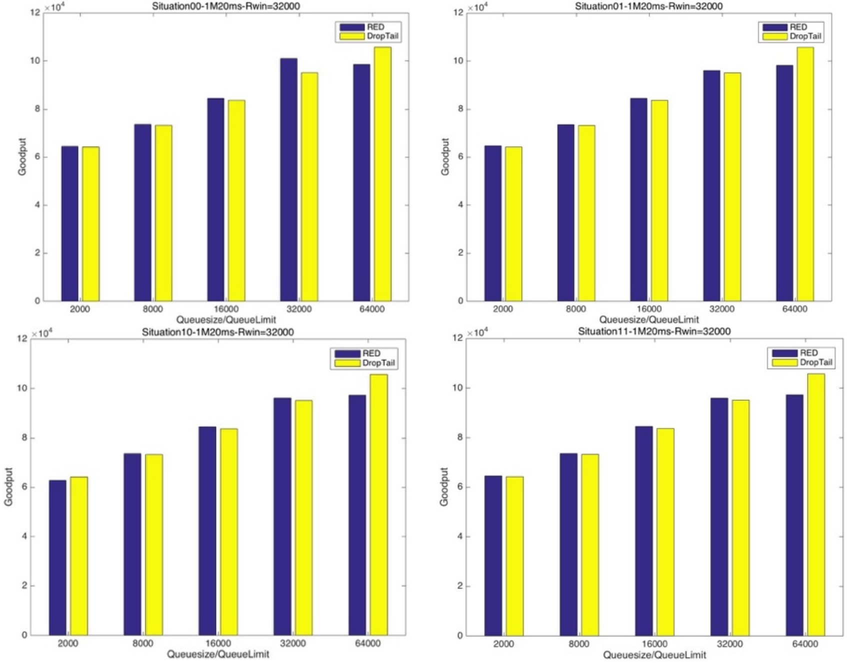
(2), In RED Queue, when packet arrives, if Average Queue Length is lower than Min-Threshold, packet will be pushed into queue; if Average Queue Length is higher than Min-Threshold, packet will have a varying possibility to be marked/dropped; if Average Queue Length is higher than Max-Threshold, there's a fixed possibility to be marked/dropped.

At the beginning, increasing threshold value means increasing the region that won't drop packets (for certain possibility). In my tested cases, the larger gap between Min-Thresh and Max-Thresh is, the more throughput the RED mode get. When it reaches a certain point (Min-Thresh =0.32 & Max-Thresh=0.64), the goodput stopped climbing, indicating a "balance" point of "enough under threshold queue size" and "efficiency in avoiding congestion".

3.2 Performance with 'wait' and 'gentle' setting

Apart from Max-Threshold and Min-Threshold value, there are also other parameters that could bring a difference to RED Queue performance.

I also changed the Boolean Value of 'Wait' and 'Gentle' in ns3 RED Queue Parameter. When 'Wait' is true, there will be an interval between dropping packets; when 'Gentle' is true, there will be gradual increase in dropping rate. Result are as follows:



The four figures above is just a view for the case WindowSize-32000, QueueSize-{2000,8000,16000,32000,64000}, and {Wait, Gentle} = {{False, False},{ False, True},{True, False}, { True, True} }.

Similar to the figures above, change of 'Wait' and 'Gentle' setting of RED doesn't bring much changes as Max & Min Threshold. Some minor improvements can be seen when both Boolean value are set to TRUE.

To have a general view of these two parameters, after look into generated goodput of all possible cases (4*4*25). The ratio of 'RED outperforms/equals DROPTAIL' cases to each 25 cases are as follow:

	Bottleneck Link Setting: DataRate (Mbps), Delay (ms)			
{Wait, Gentle}	1 Mbps, 20 ms	1Mps, 10ms	1Mps, 30ms	0.5Mps, 20ms
False, False	15:25	7:25	15:25	16:25
False, True	16:25	7:25	16:25	15:25
True, False	15:25	6:25	16:25	14:25
True, True	17:25	8:25	18:25	14:25

Hence, apart form the case {Bottleneck data rate 0.5Mps Delay 20ms}, RED gives best performance when 'Wait' and 'Gentle' are all set to True.

This is reasonable, because setting a waiting period between packet drop can better avoid sudden change in actual queuelength. Thus the measurement of average Queue Length can be more precise and reliable. Along with gradual change in drop packet possibility, both factors let RED scheme work in a more desirable manner.

4. Conclusion

- In comparison of RED and DropTail, given appropriate parameter setting, RED Queue performs better than DropTail in most cases (17 out of 25). However, the gap between these two goodputs is not that distinct. I think it is probably due to my setting RED QueueLimit = DropTail QueueSize. RED Queue Limit is often for temporary burst in Queue Length, only a fraction between Min-Threshold and Max-Threshold is used for queuing. Hence, RED goodput can be further increased and be better compared with DropTail result, if RED QueueLimit > DropTail QueueSize.
- Secondly, the performance of RED Queue Mode is sensitive to the Min & Max-Threshold Setting. Goodput increase when the gap between Threshold increases. In most cases, Optimal performance is obtained when Min-Threshold = 0.36, Max-Threshold =0.64. (Sometimes larger than this setting).
- At last, for things in common, WindowSize can suppress goodput when value is low; increasing QueueSize/QueueLimit usually improve both goodput for both RED and DropTail. Also, for both two queueing mode, TCP goodput is more influenced by the choice of Queuing parameters than UDP.

Reference

[1]. Christiansen, M., Jeffay, K., Ott, D., & Smith, F. D. (2000, August). Tuning RED for web traffic. In ACM SIGCOMM Computer Communication Review (Vol. 30, No. 4, pp. 139-150). ACM.