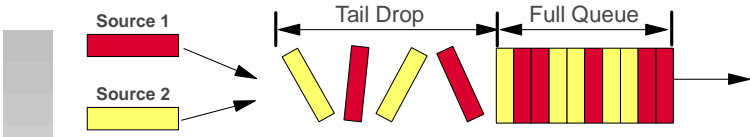


# **Random Early Detection (RED)**

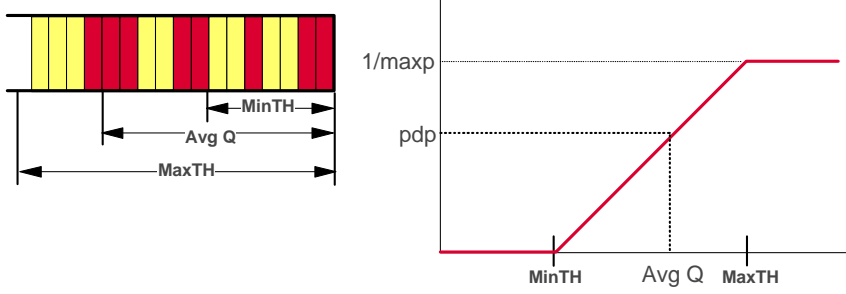
Document Owner : Lynda Linney  
Document Last Edited : 14/09/99, 09:55

## RED - Queue Management

- Why is queue management important?



- Queue Management Using RED



RED is a congestion avoidance technique which monitors network load in an effort to anticipate and avoid congestion. Congestion is typically avoided by packets being dropped. The most crude method for dropping packets is known as tail drop. When queues fill in times of congestion, the approach employed by tail drop is to drop packets until the congestion is eliminated and the queues are no longer full.

When packets from multiple TCP connections are dropped, this can result in global synchronisation. This is when periods of congestion are followed by periods of low link utilisation because all TCP hosts have reduced their transmission rates. Also the hosts will then increase their transmission rates once again when all the congestion is gone.

RED aims to control the average queue size by indicating to the end hosts when they should temporarily slow down transmission of packets. RED takes advantage of TCP's congestion control mechanism. By randomly dropping packets prior to periods of high congestion, RED tells the packet source to decrease its transmission rate. RED distributes losses over a time period, and aims to maintain low queue depth, but still allowing bursts.

The packet drop probability (pdp) is the likelihood of a packet being discarded. It is based on a minimum threshold (MinTH), a maximum threshold (MaxTH) and pdpDepth. pdpDepth will be explained later.

When the average queue depth is above the minimum threshold, RED starts to drop packets. The rate at which packets are dropped increases linearly as the average queue size increases until the average queue size reaches the maximum threshold. When the average queue size is above the maximum threshold, all packets are dropped. The minimum threshold value should be set high enough to maximise the link utilisation. If the value is too low, packets may be dropped unnecessarily and the link will not be fully utilised.

The difference between the maximum and minimum threshold should be large enough to avoid global synchronisation - i.e. if too many packets are dropped at once, the same problems occurs as with the tail drop approach.



## RED - Two Algorithms

- First controls average queue size to determine the degree of burstiness that is allowed

$$\text{AvgQueue} = \text{oldAvgQueue} \times (1 - Wq) + (qL \times Wq)$$

- Wq lies between 1 & 8
- low value : doesn't react to sudden large queue
- high value : responds quickly to congestion

- Second calculates packet-drop probability to determine how frequently a packet is dropped

$$\text{pdp} = \frac{1}{\text{maxp}} \times \frac{\text{avgQ} - \text{MinTH}}{\text{maxTH} - \text{minTH}}$$

new pdpdepth = old pdpdepth + pdp  
if pdpdepth ≥ maxp, packet dropped

There are two algorithms which RED uses to calculate when to drop packets. The first is referred to as a low pass filter. This is used to calculate the average queue length. The weight value controls how the router reacts to a temporary burst of traffic. If a small weight value is used, the router doesn't react suddenly to a burst. A large value will respond quickly to an increase in the queue depth. These parameters need to be tuned to a specific traffic characteristics.

The second equation calculates the packet drop probability for a given instance. The pdpDepth is a depth counter. The easiest way to explain this is with an example ...

Let's assume that the pdp calculate is the same at each instance.

1st packet processed:	pdp = 1/256	drop 1 in 256 packets processed
2nd packet processed:	pdp = 1/256	drop 1 in 256 packets processed
3rd packet processed:	pdp = 1/256	drop 1 in 256 packets processed
...		
256th packet processed:	pdp = 1/256	drop 1 in 256 packets processed

pdpDepth is the sum of these probabilities - i.e. pdpDepth = 1/256 + 1/256 ... 1/256. After the 256th packet, the pdpDepth is one and a packet should be dropped.



## Configuring RED

### 1. Enable RED

```
Config>FEATURE RED
Random Early Detection Config
RED Config>ENABLE RED
RED enabled
RED Config>
```

### 2. Enable RED on an interface

```
RED Config>SET INTERFACE
Enter RED Interface number [0]?
Exponential Maximum Packet Drop Probability (9 for 1/2e9) (5 - 10) [9]?
Advanced Setting (y/n)? [Yes]:
Maximum Device Queue = 16
Weight Factor (1 - 8) [4]?
Minimum Threshold Value (% of the max device queue) (0 - 100) [70]?
Maximum Threshold Value (% of the max device queue) (0 - 100) [100]?
Initial Average Queue Size (% of the max device queue) (0 - 100) [60]?
Accept input (y/n)? [Yes]:
RED Config>
```

RED is enabled on an interface. It can only be enabled on frame relay, PPP and multilink PPP interfaces. If the traffic being carried over this link does not react well to loss, consider very carefully if you wish to enable this function.

RED must be enabled globally before the interfaces can be configured. The **set interface** command is used to enable RED on a particular interface. After you have entered the interface number, you will be prompted for the “exponential maximum packet drop probability”. This effectively asks “how many x packets do wish to drop a packet. If the value is 9, this means that one of every 2<sup>9</sup> (512) packets will be dropped. You can then answer no for configured advance configuration options which will gives a configuration in which RED does not react to bursty traffic.

If you configure the advanced options you will be prompted for

1. **Weight factor** - this value determines how much influence a current queue has on calculating the average queue length. The minimum value of this parameter (1) means that the average queue length at a specific point in time remains closer to the previous average queue length. Therefore bursty traffic, which increases the queue length, has little effect on the new average queue length. This is referred to as conservative setting of RED. The maximum value of this parameter (8) is an aggressive setting. With this value, the average queue length is equal to the current queue length so bursty traffic has a significant effect on the calculation of new average queue length.
2. **Minimum threshold value** : This value designates the minimum queue requirement to calculate a packet’s drop probability and mark it accordingly. It is expressed as a percentage of the maximum device queue value, which is not configurable. For example, if you specify a value of 40 percent and the maximum device queue value is 16, then the minimum threshold value is set to 6 (0.4 x 16).
3. **Maximum threshold value** : this value designates the maximum queue requirement to calculate a packet’s drop probability and mark it accordingly It is expressed as a percentage of the maximum device queue value. For example, if you specify maximum threshold value is set to 16 (1.0 x16).
4. **Initial Average Queue Size** : This value designates the initial setting used for calculating packet drop probability. It is expressed as a percentage of the maximum device queue value. It prevents bursty traffic from increasing the weight on the average queue length calculation before an average queue value is established by the traffic itself. When the device is initialised, the queue length is zero and no indication of previous average queue length exists. You should specify a relatively low value.



## Monitoring RED

```
*TALK 5
+FEATURE RED
Random Early Detection console
RED Console>LIST
Enter RED Interface number [0]?

-----
Status  If   maxQ  avgQ  minT  maxT  qW  maxP  pktCnt  pdpDepth  passCnt  drpCnt
      (dvQ) (dvQ)  (dvQ) (dvQ)  (pkt) til drp  count      pkt      pkt
-----
Enable  0    16    14    6     16   4   1/512  1:284   1527    13931   47

Abbreviation:

maxQ = Maximum Queue Length, avgQ = Average Queue Size
minT = Minimum Threshold, maxT = Maximum Threshold
dvQ = Device Queue, qW = Queue Weight
maxP = Maximum Drop Probability: 1 drop in 512 pkts
pktCnt til drp = Packet Count before a drop occurs
pdpDepth = Probability Drop Depth: 1 drop in 2048 depth count
rtr-A RED Console>
```

Only one command exists to monitor RED which is shown above.

**BIBLIOGRAPHY:** RED was proposed by Sally Floyd and Van Jacobsen. It is written up in their paper entitled "Random Early Detection Gateways for Congestion Avoidance". This paper, and many other references, is available at <http://www.aciri.org/floyd/red.html>.