Chapter 4

RANDOMEARLY DETECTION

4.1 Introduction

This chapter presents Random Early Detection (RED) gateways for congestion avoidance in packet switched networks. The gateway detects incipient congestion by computing the average queue size. The gateway could notify connections of congestion either by dropping packets arriving at the gateway or by setting a bit in packet headers. When the average queue size exceeds a preset threshold, the gateway drops or marks each arriving packet with a certain probability, where the exact probability is a function of the average queue size.

The RED algorithm is a congestion avoidance technique highly used in modern communication networks for avoiding network congestion. Compared to existing algorithms, the RED system monitors network traffic loads in an effort to anticipate and avoid congestion at common network bottlenecks i.e. the system triggers before any congestion actually occurs [5].

One of the major problems for all the algorithms is the global synchronization at the gateways which is due to insufficient space in the buffer (queue) to accommodate the incipient congestion or transient congestion. Increasing the size of the queue will not solve the problem, because this will require a long delay for transmitting all packets in the queue [4].

4.2 Congestion Avoidance Mechanisms

In order to avoid data congestion problem at high speed gateways, there are several mechanisms which have been proposed. Some of them are used extensively in traffic control in high speed gateways.

These mechanisms drop packets at the gateways when the packet queue reaches certain threshold. Since there is a queue as a buffer in every mechanism, so, while the arriving packets are being enqueued, meanwhile some packets could be dequeued through egress port.
Therefore depending on the ratio of ingress and egress baud rate, size of the queued packets in the buffer may exceed some certain threshold value and consequently cause to drop some arriving packets. These threshold levels vary in different mechanisms, like queue overflow or a value less than overflow.

This section introduces and reviews briefly some existing congestion control mechanisms by describing their advantages. Finally, we present the algorithm that we choose to design and to implement. As we'll see all existing congestion control mechanisms drop some packets arriving at the gateway if the size of the stored packets in the buffer (queue) reaches a certain value. Some algorithms fill up the queue completely or make it overflow and then start dropping, and some verify on a threshold level. All of them choose an appropriate packet to drop but each of them do it in their own different way.

Generally there are two major factors concerning these traffic managers;

- When decide to drop the packets in respect to the instantaneoussize of the queue.
- Which packets are chosen to drop in respect to the stored packetsin the queue[4].

Below is a list of very commonly used congestion avoidance mechanisms in high speed gateways

1 – Drop Tail (DT).
2 – Random Drop (RD).
3 – ERD (Early Random Drop).
4 – PPD (Partial Packet Discard).
5 – EPD (Early Packet Discard).
6 – IP Source Quench.
7 – DEC bit.
8 – RED (Random Early Detection).
9- WRED (Weighted Random Early Detection).

4.2.1 Drop tail (DT)

This is one of the simplest mechanisms with fewer throughputs. It is a FIFO queuing mechanism that starts to drop the packets from the tail of the queue once the queue is full [19]. It means, after recognizing the condition to drop a packet is satisfied, the last arrived packet at the gateway will be dropped.

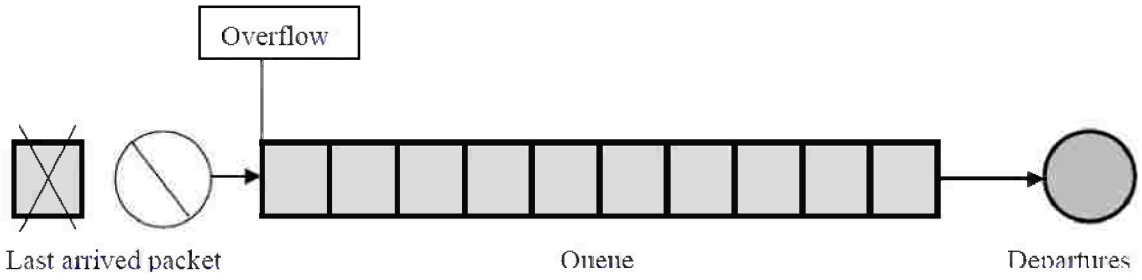


Figure 4.1 Drop Tail overflowing and drop order [4].

A major problem of Drop Tail is global synchronization. This is because dropping packets from several Virtual Channels (VCs) forces these sources to resend them again later, and then consequently resending them may cause congestion again at the gateway and that's why the global synchronization appears by such mechanisms. Figure 4.1 simply illustrates Tail drop congestion control gateways [4].

Drawbacks:

- FIFO queuing mechanism drops the packets from the tail when the queue overflows.
- DT introduces global synchronization in several connections, when the packets are dropped.

4.2.2 Random drop (RD)

Another congestion control called Random Drop, which gives feedback to the sources by dropping packets at the gateway, based on statistical situation of the gateway.

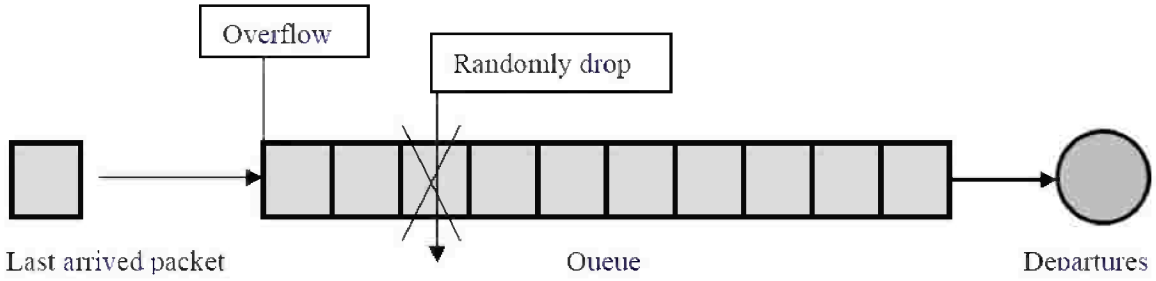


Figure 4.2 Random Drop, overflowing and drop order [4].

Unlike the Drop Tail mechanism, instead of dropping the last arrived packet, in Random Drop a packet is selected randomly from all incoming sources arrived at the gateway to be dropped. Consequently such packets belong to those particular users with a probability proportional to the average rate of data transmission [21]. Therefore, dropping packets occurs on such users whose traffic generation is much more than those generating less traffic. In other word, the users who generate less amount of traffic, experience smaller amount of packet loss. Random Drop which was originally proposed by Van Jacobson, did not improve the congestion recovery behavior of the gateways. And the performance was surprisingly worse than the other corresponding mechanisms in a single gateway bottleneck. Figure 4.2 simply illustrates the behavior of the Random Drop gateway [4].

Drawbacks:

- A key problem of this policy is that dropping a randomly selected packet results in users, generating much traffic having a greater number of packets dropped compared with those generating little traffic.
- This method is best suited for the RD congestion recovery or RD congestion avoidance[5].

4.2.3 Early random drop (ERD)

This mechanism of congestion avoidance has been first investigated briefly by Hashem. In this mechanism the packets are dropped at the gateway with a fixed drop probability once the size of the queue exceeds a certain threshold level. Many active researchers in this regard believe that both drop level and drop probability in ERD congestion avoidance should be adjusted dynamically according to the network traffic of the gateway [19].

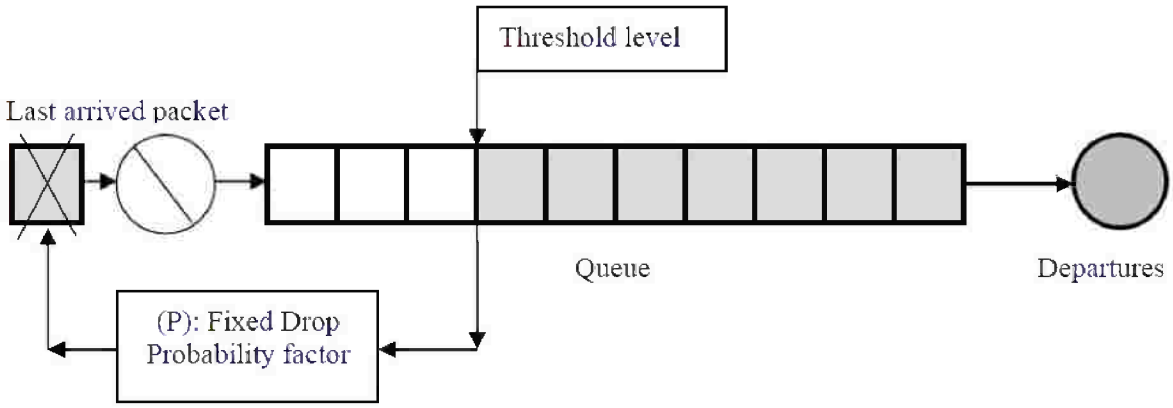


Figure 4.3 Early Random Drop[4].

Hashemfound out with Drop Tail gateways that since packets are dropped on queue overflow, it results decreasing the windows of these connections at the same time. And consequently results a loss of throughput at the gateway. Therefore, Early Random Drop gateways have better chance and are more recommended versus Drop Tail because of their broader view of traffic distribution. However it suffers from some disadvantages. For example, it has not been well successful in controlling the misbehaving of the users in the way they send data. Figure 4.3 simply illustrates the proceeding of Early Random Drop gateway.

Drawbacks:

- Reduces global synchronization.
- Does not control misbehaving users (UDP) [5].

4.2.4IP source quench

There are some different methods of congestion control mechanisms that are being used at the high speed gateways in the Internet that send some kind of a feedback to the senders reporting the congestion at the gateway. This is a congestion recovery policy. IP Source Quench is one of these methods that use such policy as described completely in [21] (RFC 1254). According to its definition, whenever a gateway responds to congestion by dropping an arrived packet, it sends a message to its source to notify the existing congestion at the gateway. This message in IP Source Quench is called ICMP (Internet Control Message Protocol). But basically the packets are not supposed to be dropped during the normal operation of the network gateway. Thereby it is very desirable to control the Sources before they overload the gateways.

A question is when to send an ICMP message. RFC 1254 says that according to the experiments based on a reasonable engineering decision, Source Quench should be applied when about half of the queue (buffer space) is filled up. However, it could be arguable to try to find another threshold, but they have not found it necessary yet.

Another question is what to do when an ICMP Source Quench is received. First, TCP or any other protocol will be informed of receiving such message. Then it demands the TCP implementations to reduce the amount of their data transmission rate toward the gateway.

Drawbacks:

- A significant drawback of this policy is that its details are discretionary, or alternatively, that the policy is really a family of varied policy[5].

4.2.5DECbitgateway

DECbit is another method that uses a recovery policy by sending a feedback to the sender. But instead of the message in IP Source Quench, DECbit sends just a 1-bit feedback as a congestion indication bit which has already been allocated in the header of the packet used to inform the sender of existing congestion at the gateway. The congestion indication bit will be enabled whenever the average queue length reaches normally 1 or greater than 1 after every arriving packet, and average queue length is calculated during last "busy + idle" period plus the current "busy" period. (The busy period means the gateway is transmitting the packets and the idle means no transmission is in process). In order to control the congestion in DECbit, if the indication feedback bit is enabled in at least 50% of the packets, then it means notifying the congestion. And if it decreases the sending window by 87.5% otherwise it is increased linearly by one packet.

As seen in this congestion control mechanism, the destination has to echo the congestion indication bit to the source, and it means that there must be the constraint of existing such special bit in the header of every arriving packet, and this could be a disadvantage versus those mechanisms not requiring such constraint[4].

4.3 Random Early Detection

RED is another congestion control mechanism proposed by Sally Floyd and Van Jacobson in early 1990s [19] that the major discussion in this work is focused on. Although it has been proposed many years ago, nevertheless because of its efficiency and considerable throughput in congestion avoidance at the gateways, it is still being used,however it has been modified several times till now [4].

RED is one of the active queue management control mechanism deployed at gateways. The RED detects initial congestion by computing the average queue size. RED gateways keep the average queue size low while allowing occasional bursts of packets in the queue. The router could notify connections of congestions either by setting a bit in packet headers or by dropping packets arriving at the router.

When the average queue size exceeds a preset threshold, the gateway drops or marks each arriving packet with a certain probability, where the exact probability is a function of the average queue size. RED gateways keep the average queue size low while allowing occasional burst of packets in the queue. Figure 4.4 shows a network that uses RED gateway with a number of source and destination host[5].

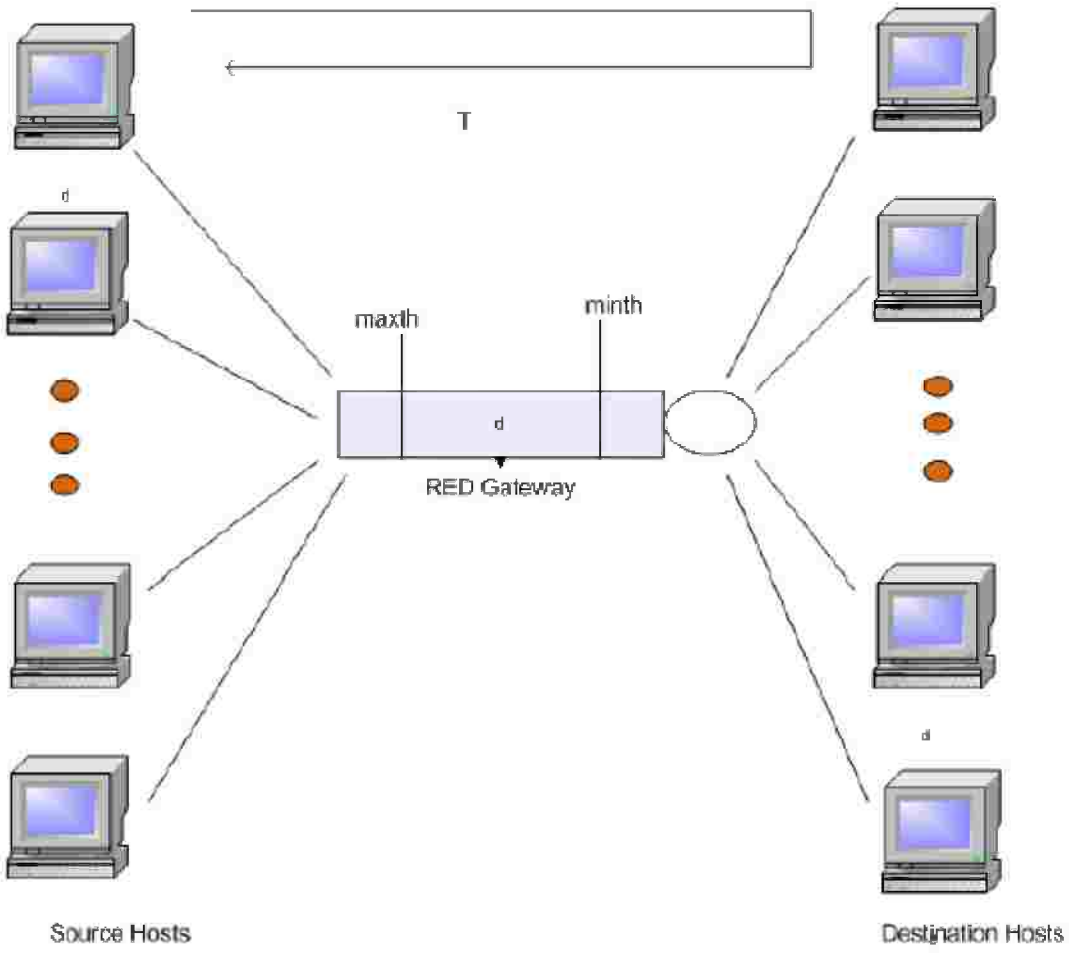


Figure 4.4 Network with RED gateway[5].

4.3.1 RED analysis

The RED Gateways control the congestion by computing the average queue size in the networks based on packet switching. It computes the average queue size after every arriving packet at the gateway and detects the congestion and notifies the sources by dropping arrived packets. The congestion detection in RED mechanism is based on two threshold levels on average queue size. These thresholds in RED are named as "Minimum Threshold Level" and "Maximum Threshold Level". After every arriving packet, the RED gateway computes the average queue size. Once the computed average queue size reaches the Minimum Threshold Level, then the arrived packet may be dropped based on a certain probability which depends on the average queue size. If the average queue size is reached or is greater than the Maximum Threshold Level, the arrived packet will be dropped. Therefore there are three areas in computed average queue size separated by these two thresholds.figure 4.5

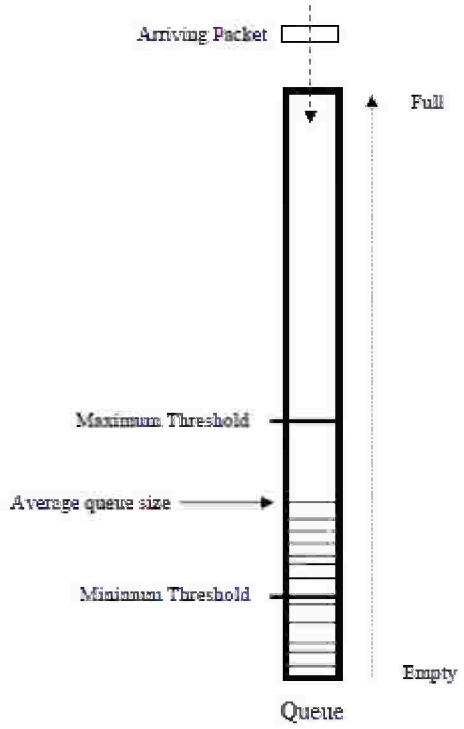


Figure 4.5 Illustrating the RED buffering mechanism [4].

In its most general form, Random Early Detection is the probabilistic discard or marking of packets as a function of queue fill before overflow conditions are reached. The probability of a packet being marked or dropped is determined by a monotonically increasing drop function d(k). The merits of RED have been greatly debated over the last ten years. The first paper detailing random early detection's merits was the Floyd and Jacobson paper of 1993. Some of the key reasons stated for RED adoption were as follows: [19].

- 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.
- Global TCP Synchronization Avoidance: 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 [19].

The RED dropping probability is a linear function that determines whether a packet will be admitted to the queue when *k* packets are currently waiting. The dropping probability d(k) is determined by a set of parameters that create the drop function. The important parameters include:[19]

- Avg : Average queue size.
- Min_th: Minimum threshold.
- Max_ th: Maximum threshold.

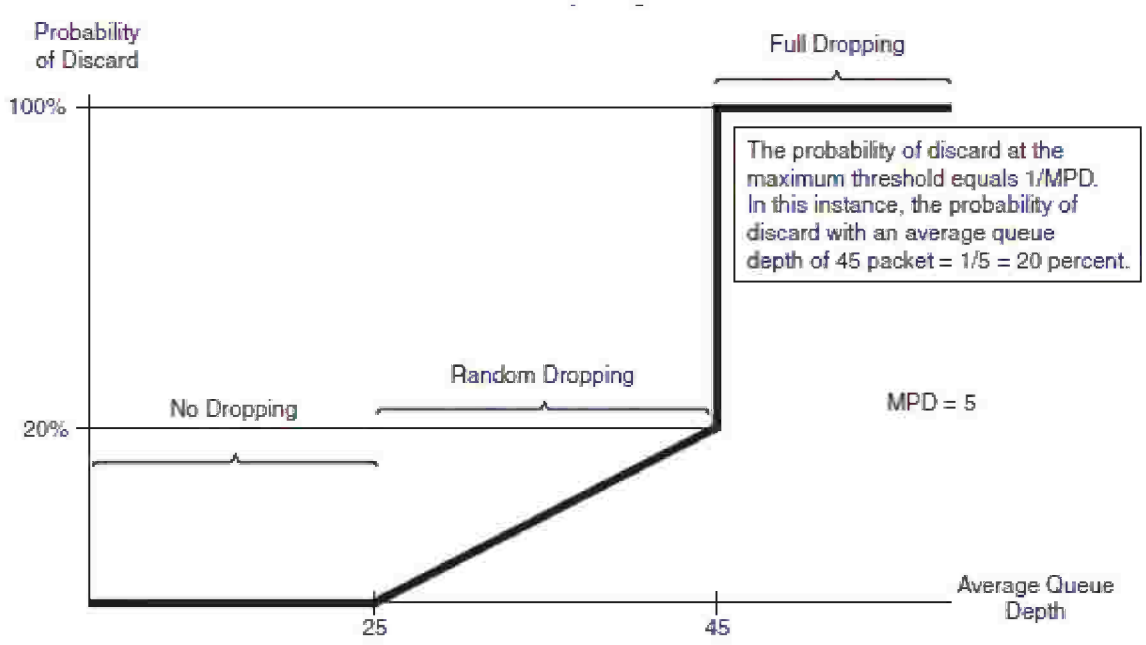


Figure 4.6 RED Dropping Function[13].

However, the probability of packet discard when the queue depth equals the maximum threshold is 1/(MPD). For example, if the mark probability denominator (MPD) were set to 10, when the queue depth reached the maximum threshold, the probability of discard would be 1/10 (that is, a 10 percent chance of discard).The minimum threshold, maximum threshold, and MPD comprise the RED profile. Figure 4.6 shows the three distinct ranges in a RED profile: no drop, random drop, and full drop.

4.3.2 RED Algorithm

In the RED algorithm, the average queue size has the main role in the calculations of the drop decision. The average queue size is calculated every arrived packet. Then the calculated average queue size is compared to two threshold levels (Minimum Threshold Level and Maximum Threshold Level), as illustrated in Figure 4.7 (Min_th, Max_th) [4].

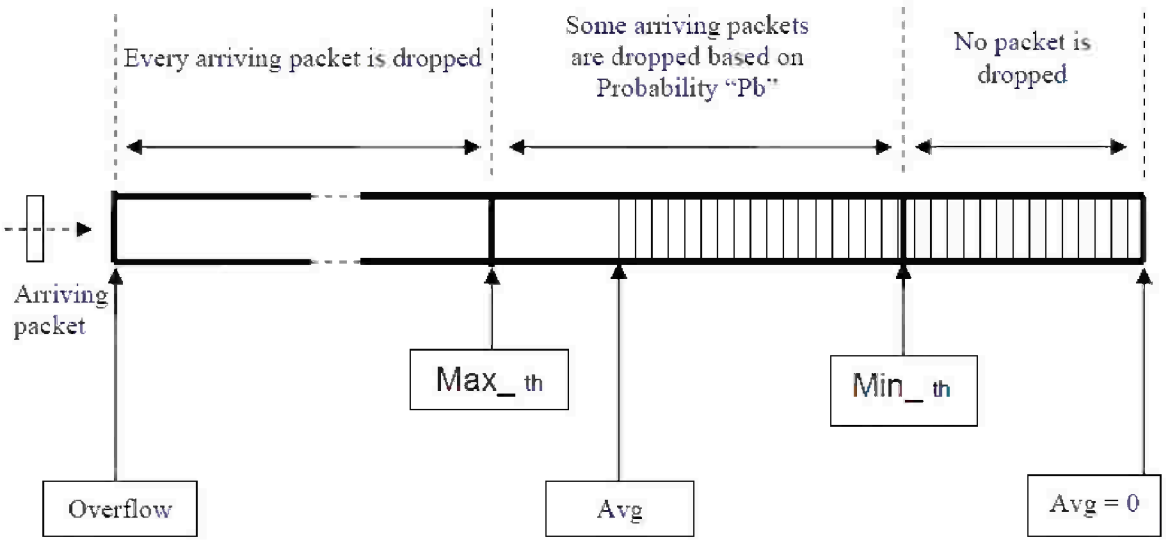


Figure 4.7 Illustrating detailed RED buffering mechanism [4].

If the average queue size (Avg) is below the minimum threshold level, no packet is dropped. If the average queue size is over the maximum threshold level, every arrived packet is dropped. And when the Avg is between the minimum threshold and maximum threshold levels which is the special case of the RED algorithm, the drop decision is based on the main calculations in the algorithm explained in section 4.3.5.

Let us note that goal of the RED algorithm is to keep the average queue size at a very low level in order to avoid global synchronization by dropping packets fairly to keep the average between minimum and maximum threshold levels. Now let's see how the average queue size is computed in RED [4].

4.3.3 Drop decision

The average queue size is compared to the minimum and maximum threshold levels for the drop decision with respect to three regions described earlier as follows:

- Avg< Min_ thKeep arrived packet.
- Min_ th≤ Avg< Max_ th = Drop by Probability.
- Avg≥ Max_ th = Drop arrived packet.

4.3.4 RED functionality

Since this probability is a function of the average queue size, every time a packet is marked with it, then the probability that a packet is marked from a particular connection is roughly proportional to that connection's share of the bandwidth at the router. Moreover this probability determines whether the packet is dropped or not as shown in Figure 4.8.The minimum threshold value should be set high enough to maximize the link utilization. If the minimum threshold is too low, packets may be dropped unnecessarily, and the transmission link will not be fully used.

To avoid global synchronization the difference between maximum and minimum threshold should be large. If the difference is too small, many packets may be dropped at once, resulting thus in global synchronization[5].

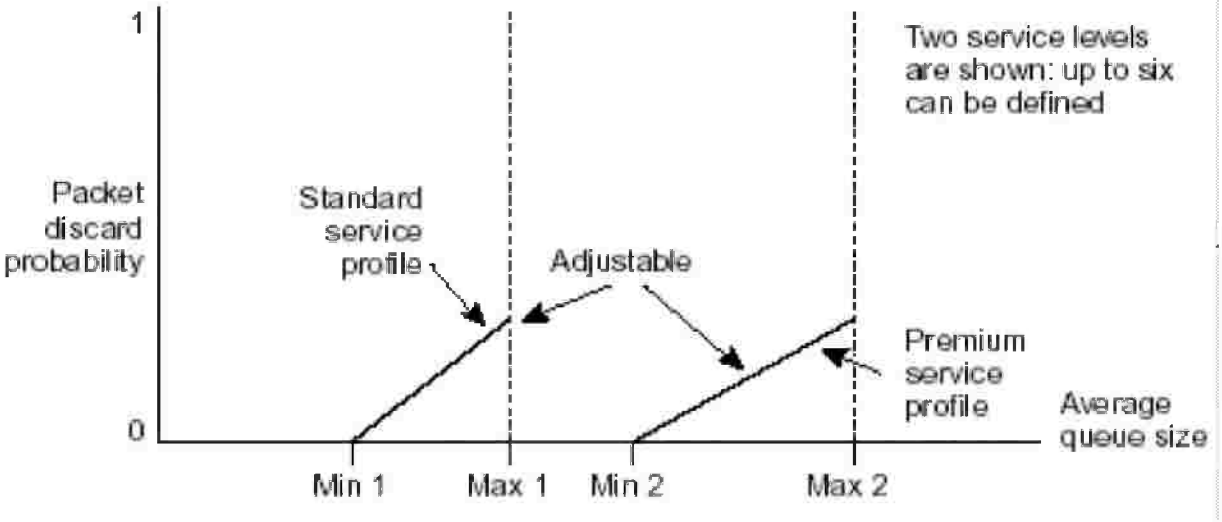


Figure 4.8 RED packet drop probability [5].

The Min_th and Max_th must be chosen such that:
1. Min_th: must be large enough to accommodate bursty traffic.
2. Max_th: must not lead to long average delay.
3. rule of thumb: set Max_th at least twice Min_th[5].

4.3.5 RED's impact on traffic flow

Usually, UDP based application can be considered as smooth traffic. At the end host TCP congestion control mechanism is implemented. The upper layer application take care of congestion and further retransmission whereas UDP hosts neglect packet loss and keep pumping data into network. It does notmean that RED algorithm has no impact on UDP application. It has impact on all links of internet traffic, including both TCP and UDP connections[5].

RED will drop packets using one of three methods:

- **No drop:** used when there is no congestion.
- **Random drop:** used to prevent a queue from becoming saturated, based on thresholds.
- **Tail drop:** used when a queue does become saturated.

4.3.6 Advantages of RED

RED prevents the queue from filling to capacity, by randomly dropping packets in the queue. RED algorithm helps alleviate two TCP issues caused by tail drop:

- **TCP Global Synchronization** occurs when a large number of TCP packets are dropped simultaneously. Hosts will reduce TCP traffic (referred to as slow start) in response, and then ramp up again… simultaneously. This results in cyclical periods of extreme congestion, followed by periods of under-utilization of the link.
- **TCP Starvation**occurs when TCP flows are stalled during times of congestion (as detailed above), allowing non-TCP traffic to saturate a queue (and thus starving out the TCP traffic)[22].

RED indiscriminately drops random packets. It has no mechanism to differentiate between traffic flows. Thus, RED is mostly deprecated[22].
Finally here we can recall some benefits and principals of RED mechanism:

- Provides both congestion recovery and congestion avoidance.
- Avoids global synchronization against bursty traffic.
- Works with TCP and non-TCP transport-layer protocol.
- Monitors the average queue size.
- Uses randomization method to distribute the congestion notification [4].

4.3.7 Disadvantages of RED

The parameters of RED are very sensitive and erroneous configuration will degrade RED performance to the level of DT.

Threshold:

The values of the minimum and maximum thresholds are assigned depending on the desirable actual average of the queue size. The maximum threshold represents the maximum average queue size that is allowed in the queue. Hence, higher maximum thresholds will increase delays. On the other hand, a lower maximum threshold will decrease throughput. In addition, higher minimum thresholds increase throughput and link utilization.

4.3.8Applications and limitations

In addition to the design goals outlined above for the congestion avoidance schemes we will describe how our goals have been met by RED gateways.

- **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.
- **Simplicity:**The RED gateway algorithm could be implemented with moderate overhead in current networks.
- **Appropriate time scales -** After notifying a connection of congestion by marking a packet, it takes at least a round trip time for the gateway to see a reduction in the arrival rate. In RED gateway the time scale for the detection of congestion roughly matches the time scale required for connections to respond the congestion. RED gateways do not notify connections to reduce their windows as a result of transient congestion at the gateway.
- **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[5].

Limitations:

- One of RED's main weaknesses is that the average queue size varies with the level of congestion and with the parameter settings. That is, when the link is lightly congested and/or maxp is high, the average queue size is near minth; when the link is more heavily congested and/or maxp is low, the average queue size is closer to, or even above, maxth. As a result, the average queuing delay from RED is sensitive to the traffic load and to parameters, and is therefore not predictable in advance. Delay being a major component of the quality of service delivered to their customers, network operators would naturally like to have a rough a priori estimate of the average delays in their congested routers; to achieve such predictable average delays with RED would require constant tuning of RED's parameters to adjust the current traffic conditions.
- A second, related weakness of RED is that the throughput is also sensitive to the traffic load and to RED parameters. In particular, RED often does not perform well when the average queue becomes larger than max_{th}, resulting in significantly decreased throughput and increased dropping rates. Avoiding this regime would again require constant tuning of the RED parameters.
- Third disadvantage/drawback is RED treats all the flows uniformly i.e. it fails to differentiate miss-behaved flows and well-behaved flows while dropping the packets. So, if a well-behaved packet comes when the queue is congested it also drops with the same probability as the ill-behaved flow [5].

4.4 Weighted Random Early Detection (WRED)

Whereas queuing provides congestion management, mechanisms such as WRED provide congestion avoidance. Specifically, WRED can prevent an output queue from ever filling to capacity, which would result in packet loss for all incoming packets. This section examines the need for and the configuration of WRED on Cisco routers.Cisco does not support RED, but fortunately it supportssomething better: Weighted Random Early Detection(WRED). Unlike RED, WRED has a profile for each prioritymarking. For example, a packet with an IP Precedence valueof 0 might have a minimum threshold of 20 packets, whereasa packet with an IP Precedence of 1 might have a minimumthreshold of 25 packets. In this example, packets with an IPPrecedence of 0 would start to be discarded before packetswith an IP Precedence of 1.

Although WRED can be configured from interface-configurationmode or from virtual-circuit-configuration mode, this chapter focuses on an MQC-based WRED configuration. To enable WRED and to specify the marking that WRED pays attention to (that is, IP Precedence or DSCP), issue the following policy-map-class configuration-mode command:

Router(config-pmap-c)#**random-detect** [**dscp-based** | **prec-based**]

After WRED is configured, the IOS assigns default minimum threshold and maximum threshold values. However, you can alter those default parameters with the following commands:

Router(config-pmap-c)#*random-detect dscpdscp_valueminimum-threshold maximum-threshold mark-probabilitydenominator*
(Used for dscp-based WRED.)

To reinforce this syntax, consider the following example, where the goal is to configure WRED for the **WREDTEST** class-map. After the class-map's queue depth reaches 25 packets, a DSCP value of AF13 might be discarded. Packets that are marked with a DSCP value of AF12 should not be discarded until the queue depth reaches 30 packets, and finally, packets that are marked with a DSCP value of AF11 should have no chance of discard until the queue depth reaches 35 packets. If the queue depth exceeds 100 packets, there should be a 100 percent chance of discard for these three DSCP values. However, when the queue depth is exactly 100 packets, the chance of discard for these various packet types should be 25 percent. Also, CB-WRED requires that CB-WFQ be configured for the interface. So, as an additional requirement, you make 25 percent of the interface's bandwidth available to the **WREDTEST** class of traffic.

```
Router(config-pmap)#class WREDTEST
Router(config-pmap-c)#bandwidth percent 25
Router(config-pmap-c)#random-detect dscp-based
Router(config-pmap-c)#random-detect dscp af13 25 100 4
Router(config-pmap-c)#random-detect dscp af12 30 100 4
Router(config-pmap-c)#random-detect dscp af11 35 100 4
```

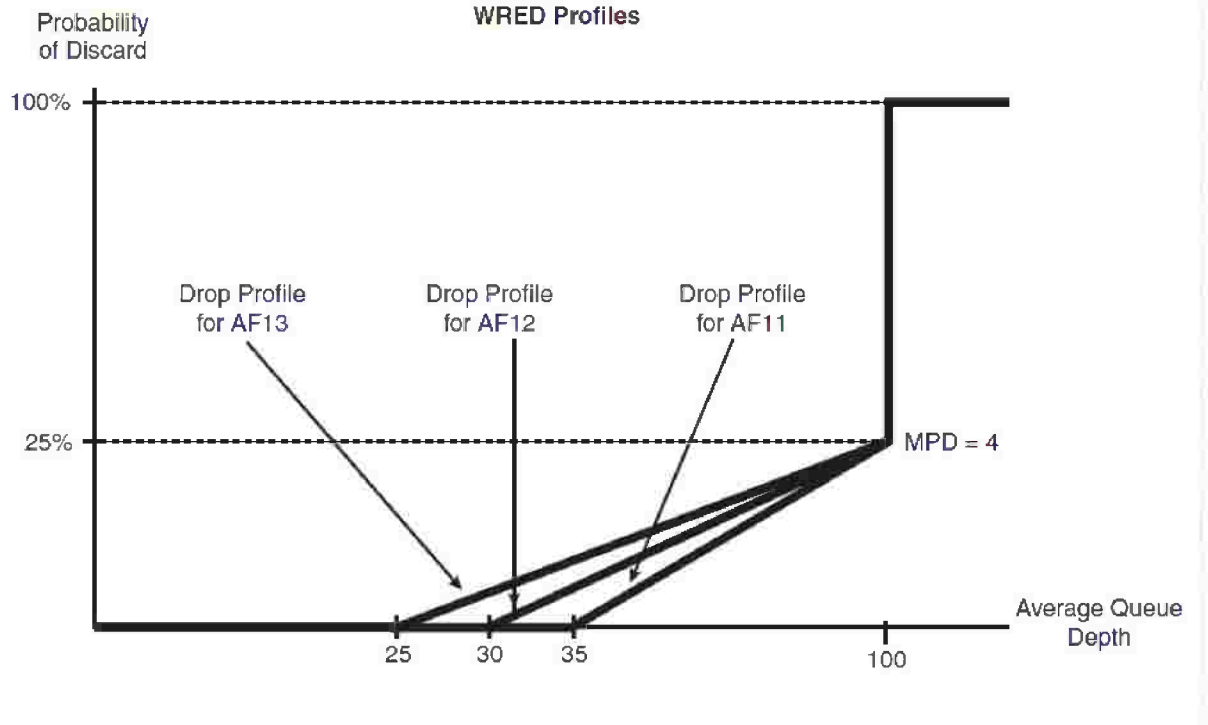


Figure 4.9WRED Profiles [13].

Examine the solution, and notice that the MPD is 4. This value was chosen to meet the requirement of a 25 percent chance of discard when the queue depth equals the maximum

threshold (that is, 1/4 = .25). Also, notice that a DSCP value of AF13 is dropped before a DSCP value of AF12, which is dropped before a DSCP value of AF11. This approach is consistent with the definition of the per-hop behaviors(PHBs), because the last digit in the AF DSCP name indicates its drop preference. For example, a value of AF13 would drop before a value of AF12.Figure 4.9.

To view the minimum threshold, maximum threshold, and MPD settings for the various IP Precedence or DSCP values, you can issue the **show policy-map interface** *interface-identifier* command[13].

4.4.1 Extension of Markov Chain analysis to WRED

The WRED algorithm that we want to use is different than the one implemented in the OPNET Tool. The OPNET Tool has different minimum average queue size for each packet with different ToS Byte, but the same maximum average queue size and dropping probability (Pmax). This case is presented in Figure 4.10[20].

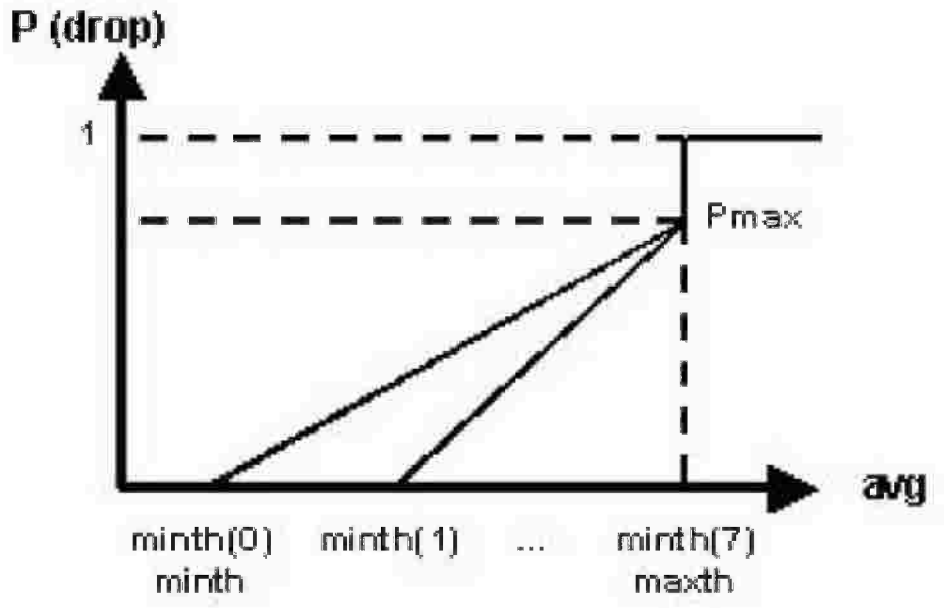


Figure 4.10The WRED mechanism in OPNET[20].

Using the same approximations, WRED can be analyzed in a very similar way to RED. WRED uses the same parameters as RED, but it has the ability to perform RED on traffic classes individually. For example, this figure shows a WRED system in which both classes have the same max_p but different min_{th}. The Class 2 traffic begins RED discard when the queue contains one packet, and Class 1 traffic discard begins at k=2.

WRED analysis is similar to RED analysis, but the transition rates for the CTMC will differ and the general complexity of the Markov chain will increase. This section will take the reader through a complete example of the calculation of WRED blocking probability for a system with two bursty input sources that are assigned different RED blocking probabilities.

4.4.2 WRED configuration

There are two methods to configuring WRED. Basic WRED configuration is accomplished by configuring minimum and maximum packet *thresholds* for each IP Precedence or DSCP value.

- The minimum threshold indicates the minimum number of packets that must be queued, before packets of a specific IP Precedence or DSCP value will be *randomly* dropped.
- The maximum threshold indicates the number of packets that must be queued, before *all* new packets of a specific IP Precedence or DSCP value are dropped. When the maximum threshold is reached, WRED essentially mimics the tail drop method of congestion control.
- The mark probability denominator (MPD) determines the number of packets that will be dropped, when the size of the queue is in between the minimum and maximum thresholds. This is measured as a fraction, specifically 1/MPD. For example, if the MPD is set to 5, one out of every 5 packets will be dropped. In other words, the chance of each packet being dropped is 20% [22].

Table 4.1 RED Parameters[22].

Precedence	Minimum Threshold	Maximum Threshold	MPD
0	10	25	5
1	12	25	5
2	14	25	5
3	16	25	5

If the WRED configuration matched the above, packets with a precedence of *0* would be randomly dropped once *10* packets were queued. Packets with a precedence of *2* would similarly be dropped once *14* packets were queued.table 4.1.

The maximum queue size is *25*, thus all new packets of any precedence would be dropped once *25* packets were queued[22].

WRED configuration can be based on either IP Precedence or a DSCP value. To configure WRED thresholds using IP Precedence:

```
Router(config)# interface fa0/1
Router(config-if)# random-detect
Router(config-if)# random-detect precedence 0 10 25 5
Router(config-if)# random-detect precedence 1 12 25 5
Router(config-if)# random-detect precedence 2 14 25 5
Router(config-if)# random-detect precedence 3 16 25 5
Router(config-if)# random-detect precedence 4 18 25 5
Router(config-if)# random-detect precedence 5 20 25 5
```

The first random-detect command enables WRED on the interface. Thesubsequent random-detect commands apply a minimum threshold,maximum threshold, and MPD value, for each specified IP Precedence level[22].

To configure WRED thresholds using DSCP values:
Router(config)# interface fa0/10
Router(config-if)# random-detect
Router(config-if)# random-detect dscp-based af11 14 25 5
Router(config-if)# random-detect dscp-based af12 12 25 5
Router(config-if)# random-detect dscp-based af13 10 25 5
Router(config-if)# random-detect dscp-based af21 20 25 5
Router(config-if)# random-detect dscp-based af22 18 25 5
Router(config-if)# random-detect dscp-based af23 16 25 5
To view the WRED status and configuration on all interfaces:
Router# show interface random-detect
Router# show queuing
WRED is not compatible with Custom Queuing (CQ), Priority Queuing (PQ) or Weighted Fair Queuing (WFQ), and thus *cannot be enabled* on interfaces using one of those queuing methods[22].

4.4.3Advanced WRED configuration

Advanced WRED configuration involves tuning WRED maximum and minimum thresholds on a per-queue basis, rather than to specific IP Precedence or DSCP values. In this instance, the min and max thresholds are based on percentages, instead of a specific number of packets. This is only supported on higher model Catalyst switches.WRED only affects standard queues. Traffic from strict priority queues is never dropped by WRED[22].

4.4.4 Configuring advanced WRED with WRR

Why two separate minimum and maximum thresholds per queue? Because packets of a specific CoS value can be mapped to a specific threshold of a specific queue. Observe:
Switch(config-if)# *wrr-queue cos-map 1 1 0 1*
Switch(config-if)# *wrr-queue cos-map 1 2 2 3*

The first command creates a *map*, associating queue *1*, threshold *1* with CoS values of *0* and *1*. The second command creates a *map*, associating queue *1*, threshold *2* with CoS values of *2* and *3*.

All traffic marked with CoS value 0 or 1 will have a minimum threshold of 5 percent, and a maximum threshold of 40 percent (per the earlier commands).

All traffic marked with CoS value 2 or 3 will have a minimum threshold of10 percent, and a maximum threshold of 100 percent.

The above *wrr-queue* commands are actually the *default* settings on higher end Catalyst switches.

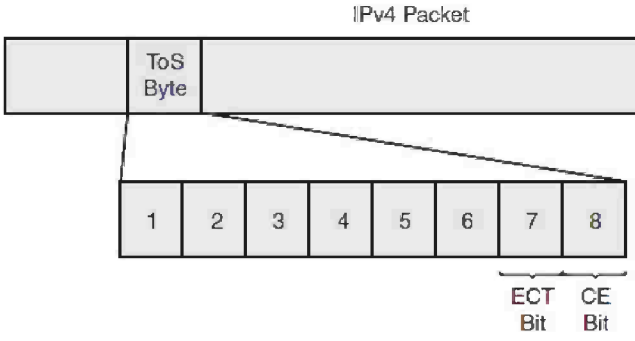
To view the QoS settings on a Catalyst interface:

Switch# *show mlsqos interface fa0/10*
To view the queuing information for a Catalyst interface:
Switch# *show mlsqos interface fa0/10 queuing*
To view QoS mapping configurations:
Switch# *show mlsqos maps*[22].

4.5 Explicit Congestion Notification (ECN) Configuration

WRED discards packets, and that is one way for the router to indicate congestion. However, routers can now indicate a congestion condition by signaling, using an approach called ECN.

ECN uses the 2 last bits in the ToS byte to indicate whether a device is ECN capable, and if so, whether congestion is being experienced.



Bit Combination	Meaning
00	Router is not ECN capable.
01	Router is ECN capable but is not experiencing congestion.
10	Router is ECN capable but is not experiencing congestion.
11	Router is ECN capable and is currently experiencing congestion.

Figure 4.11 Explicit Congestion Notification (ECN) [13].

Cisco routers can use ECN as an extension to WRED and mark packets that exceed a specified value, instead of dropping the packets. If the queue depth is at or below the WRED minimum threshold, the packets are sent normally, just as with WRED. Also, if the queue depth is above the WRED maximum threshold, all packets are dropped, just as with WRED. But if the queue depth is currently in the range from the minimum threshold through the maximum threshold, one of the following things can happen:

- If both endpoints are ECN capable, the ECT and CE bits are set to a 1 and sent to the destination, indicating that the transmission rate should be reduced.
- If neither endpoints supports ECN, the normal WRED behavior occurs.

- A packet with its ECN and CE bits marked can reach a destination router that already has a full queue. In such an instance, the notification is dropped.

Use the following command to enable ECN:

Router(config-pmap-c)#**random-detectecn**

Note that although WRED also can be configured in interface-configuration mode, ECN must be configured through MQC. Because ECN is configured by the three-step MQC process, the same verification and troubleshooting commands apply. Specifically, you can use the **show policy-map** and **show policy-map interface** *interface-identifier* commands to verify the ECN configuration[13].