BROADCOM WRED FOR FAP-BASED PLATFORM OPERATION GUIDE

Broadcom Corporation
5300 California Avenue
Irvine, California, USA 92677
Phone: 949-926-5000
Fax: 949-926-5203

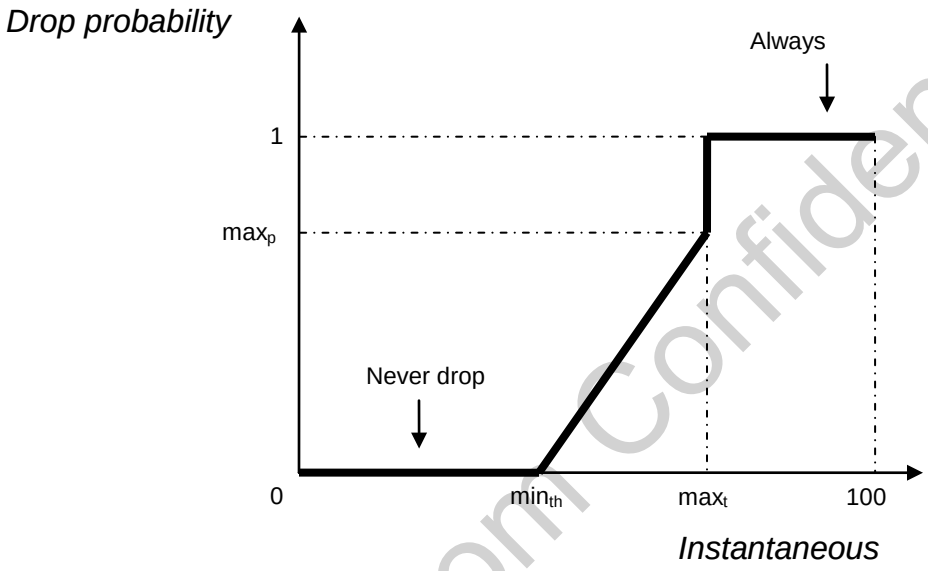
Revision History

Revision	Date	Change Description
1	11/03/2014	Initial release

Table of Contents

Overview

RED, Random Early Drop/Detection, is an active queue management algorithm that takes place before placing the packet into the queue. It enhances congestion control, preventing queue overflow. When RED is used, instead of dropping all packets above a certain queue threshold, packets are discarded with computed probability based on queue occupancy and the configured maximum drop probability. As queue occupancy grows the probability of dropping a packet in the queue grows as well.



Probability grows linearly from min_{th} to max_{th}:

- Q_{inst} <= min_{th}: p = 0
- min_{th} < Q_{inst} <= max_{th}: $p = max_p * (Q_{inst} - min_{th}) / (max_{th} - min_{th})$
- Q_{inst} > max_{th}: p = 1

WRED, Weighted Random Early Detection, is an extension to RED where a single queue may have several different queue thresholds / profiles. Each queue profile is associated to a particular traffic class / priority. In FAP WRED, we set the limitation to two queue profiles per queue for WRED.

The Broadcom WRED for FAP-based platform supports the following features:

- Total of 8 queue profiles are supported with 1 queue profile reserved for default queue profile. Even though this number is way smaller than the multiplication of number of ports and queues supported. It assumes queue profiles can be shared among different queues. (The number of queue profile supported can easily be modified.)

- Support both Ethernet and XTM interfaces
- Each queue profile will define a drop probability (in value of 0 to 100), minimum threshold, and maximum threshold (in value of packet count).
- Each queue now can support three packet drop algorithm of AQM (active queue management). They are Drop-Tail, RED, and WRED. The default is Drop-Tail.
- When queue is configured with WRED drop algorithm, it can support two queue profiles, one for low priority packet and the other one for high priority packet.

Command Line Interface

We have two user-space configuration tools to control FAP. They are “fapctl” and “tmctl.” Modifications are done in both user-space tools to support this newly added feature.

fapctl

Three new command options are introduced. TM status option is updated to display the queue profile and AQM applied to a queue.

New command options:

- `fapctl tm --queueprofile --queueprofileid <id> --dropprob <dropprob> --minthreshold <threshold> --maxthreshold <threshold>`
- `fapctl tm --queuedropalg --if <ifName> --queue <queue> (--dt | --red | --wred) (--queueprofileid <id> (--queueprofileidhi <id> --prioritymask0 <0xhex> --prioritymask1 <0xhex>))`
- `fapctl tm --xtmchanneldropalg --channel <channel> (--dt | --red | --wred) (--queueprofileid <id> (--queueprofileidhi <id> --prioritymask0 <0xhex> --prioritymask1 <0xhex>))`
    - The way that prioritymask0 and prioritymask1 work is by checking the Traffic Class value that gets passed along in the packet marking mechanism by NetFilter. The TC value will be Bit[10:5] in the packet mark. That’s a 6-bit value, which represent 0 to 63. User can use iptables or ebtables command to set the TC value on Bit[10:5] of packet mark. The value of prioritymask0 is used to determine whether TC value from 0 to 31 will be enqueued with queue profile with ID of queueprofileidhi. Prioritymask1 is from TC value from 32 to 63. An example that “--prioritymask0 0x8000042 --prioritymask1 0x1234000” means that packet with TC value of 1, 6, 31, 50, 52, 53, 57, or 60 will be handled with queue profile with ID queueprofileidhi.

tmctl

Four new command options are introduced. They are:

- `tmctl getqprof {--qprofid} <qprofid>`
- `tmctl setqprof {--qprofid} <qprofid> {--redminthr} <redminthr> {--redmaxthr} <redmaxthr> {--redpct} <redpercentage>`
- `tmctl getqdropalg {--devtype} <devtype> {--if} <interface> {--qid} <qid>`

- `tmctl setqdropalg {--devtype} <devtype> {--if} <interface> {--qid} <qid> {--dropalg} <dropalgorithm> {--qprofid} <qprofid> {--qprofidhi} <qprofidhi> {--priomask0} <hex mask> {--priomask1} <hex mask>`
  - For explanation of priomask0 and priomask1, please look at the explanation above in fapctl modification.

Example

In a system that we would like to make all the TCP packets being handled with random early drop, the following commands can be used:

```
#> iptables -t filter -A FORWARD -p tcp -j MARK --set-mark 0x3e0/0x7e0 ; we have
bit[10:5] in skb/fkb->mark used for traffic class ID.  Right here we set it to 0x31

#> fapctl tm --queueprofile --queueprofileid 1 --dropprob 100 --minthreshold 20 --
maxthreshold 50 ; create a queue profile

#> fapctl tm --queuedropalg --if eth1 --queue 0 --wred --queueprofileid 0 --
queueprofileidhi 1 --prioritymask0 0x80000000 --prioritymask1 0x0 ; setting up
prioritymask to capture TC = 0x31 and apply queue profile with ID=1 for the traffic.
This works on eth1 and queue#0 (assuming eth1 is the WAN port)

#> fapctl tm --queuedropalg --if eth3 --queue 0 --wred --queueprofileid 0 --
queueprofileidhi 1 --prioritymask0 0x80000000 --prioritymask1 0x0 ; setting up
prioritymask to capture TC = 0x31 and apply queue profile with ID=1 for the traffic.
This works on eth3 and queue#0 (assuming eth3 is the LAN port we use)

#> fapctl tm --apply --if eth3 ; to apply the queue profile for eth3

#> fapctl tm --apply --if eth1 ; to apply the queue profile for eth1

#> fapctl tm --xtmchanneldropalg --channel 0 --wred --queueprofileid 0 --
queueprofileidhi 1 --prioritymask0 0x80000000 --prioritymask1 0x0 ; setting up
prioritymask to capture TC = 0x31 and apply queue profile with ID=1 for the traffic.
This works on XTM channel#0
```