

WRED

WRED

WRED(short for Weighted Random Early Detection) is a congestion avoidance mechanism which makes use of the congestion control mechanism of TCP (Transmission Control Protocol). When an output queue begins to experience congestion, WRED starts dropping packets *selectively*.By dropping some packets earlier than the point when the queue is full, WRED prevents the situation where a large number of packets get dropped at once. The principle specifies as follows:

- (1)The length of the queue(measured with kbps or pktps) is calculated.
- (2)If the queue length is less than the minimum threshold, the packet is placed in the queue.
- (3)If the queue length is more than the maximum threshold, the packet is dropped.
- (4)If the queue length is more than the minimum threshold but less than the maximum threshold, the packet is either dropped or queued, based on the packet drop probability. The greater the chance of the packet is dropped when the drop_probability value of its corresponding queue is greater.Usually,the drop_probability value of the lower priority queues with less importance are the greatest.

1 Create two wred_queues for interface te-1/1/35

```
root@PicOS-OVS$ovs-vsctl set interface te-1/1/35 wred_queues:0=@wred1 wred_queues:3=@wred2 -- --id=@wred1
create wred_queue enable=true min_thresh=100 max_thresh=200 drop_probability=50 -- --id=@wred2 create
wred_queue enable=true min_thresh=200 max_thresh=400 drop_probability=40
52f26264-3656-4840-8f0e-fb7be8e52ef8
9b36b9dc-903a-4bb6-8e25-f6e2b1fd2b4d
root@PicOS-OVS$ovs-ofctl add-flow br0 in_port=34,dl_src=22:11:11:11:11:11,actio
ns=set_queue:3,output=35
```

Notice: The drop_probability is according to the percentage to drop packets.

2 Check wred queues

```

Check all wred_queue:
root@PicOS-OVS$ovs-vsctl list wred_queue
_uuid          : 52f26264-3656-4840-8f0e-fb7be8e52ef8
drop_probability : 50
enable         : true
max_thresh     : 200
min_thresh     : 100
_uuid          : 9b36b9dc-903a-4bb6-8e25-f6e2b1fd2b4d
drop_probability : 40
enable         : true
max_thresh     : 400
min_thresh     : 200
root@PicOS-OVS$

Check one wred_queue:
root@PicOS-OVS$ovs-vsctl list wred_queue 9b36b9dc-903a-4bb6-8e25-f6e2b1fd2b4d
_uuid          : 9b36b9dc-903a-4bb6-8e25-f6e2b1fd2b4d
drop_probability : 40
enable         : true
max_thresh     : 400
min_thresh     : 200
root@PicOS-OVS$

Check interface wred_queue:
root@PicOS-OVS$ovs-vsctl list interface te-1/1/35
_uuid          : db795f1d-71e9-448c-9ac4-f2304c4bf74d
.....
.....
status         : {}
type           : "pica8"
wred_queues    : {0=f37f30cf-8470-4edd-925f-5c2abcd061eb, 3=b97fa496-57b2-40f1-a525-b978abc64403}
root@PicOS-OVS$

```

3 Reset one wred queue according to its uuid

```

root@PicOS-OVS$ovs-vsctl set wred_queue 9b36b9dc-903a-4bb6-8e25-f6e2b1fd2b4d max_thresh=300

root@PicOS-OVS$ovs-vsctl list wred_queue 9b36b9dc-903a-4bb6-8e25-f6e2b1fd2b4d
_uuid          : 9b36b9dc-903a-4bb6-8e25-f6e2b1fd2b4d
drop_probability : 40
enable         : true
max_thresh     : 300
min_thresh     : 200
root@PicOS-OVS$

```

4 Add one wred queue to wred_queues

```

root@PicOS-OVS$ovs-vsctl set interface te-1/1/35 wred_queue:1=@wrednew -- --id=@wrednew create wred_queue
enable=true min_thresh=100 max_thresh=500 drop_probability=40
d60801bd-827b-45b7-aa5d-1a2db9f1e513
root@PicOS-OVS$

```

5 Remove one or all wred queue

```

Remove one wred queue:
root@PicOS-OVS$ovs-vsctl remove interface te-1/1/35 wred_queues 3
root@PicOS-OVS$
root@PicOS-OVS$ovs-vsctl list wred_queue
_uuid                : 52f26264-3656-4840-8f0e-fb7be8e52ef8
drop_probability     : 50
enable               : true
max_thresh           : 200
min_thresh           : 100
_uuid                : d60801bd-827b-45b7-aa5d-1a2db9f1e513
drop_probability     : 40
enable               : true
max_thresh           : 500
min_thresh           : 100
root@PicOS-OVS$

Delete all wred queues:
root@PicOS-OVS$ovs-vsctl clear interface te-1/1/35 wred_queues
root@PicOS-OVS$
root@PicOS-OVS$ovs-vsctl list wred_queue
root@PicOS-OVS$

```

WRED-ECN

From PicOS2.10.0, pica8 switch support enable ecn.

ECN (short for Explicit Congestion Notification) is a property of queue. When congestion occurs, WRED drops packets based on the queue length exceeding certain threshold value. But if you enable ECN on a queue, the packets are marked instead of dropped. Downstream routers and hosts would see this marking as an indication of network congestion and slow down their packet transmission rates.

ECN is a value in the DS (Differentiated Services) field of the IPv4 protocol header. ECN uses the two least significant (right-most) bits of the 8-bit DF field to encode four different codepoints:

- (1)00 - Not ECN-Support Transport
- (2)01 - ECN-Support Transport(1)
- (3)10 - ECN-Support Transport(0)
- (4)11 - Congestion Experienced

When both end hosts support ECN they mark their packets with either 10 or 01. When WRED and ECN is enabled and the queue is congested, the PicOS changes the ECN field of all such packets to 11. When ECN is not enabled, the ECN bits are not changed.



1, WRED-ECN only support in unicast flow entry.

Command:

ECN take effect when enabling WRED on a port. The ecn value of each queue can be different, you can use command like this, disable ecn in queue2 and enable ecn in queue5.

```
ovs-vsctl set-port-wred-ecn ge-1/1/1 2=false 5=true
```

If user want to make the ecn values of all queues are equal, you can use command like this for simplification.

```
ovs-vsctl set-port-wred-ecn ge-1/1/1 true/false
```

Show wred-ecn

```
ovs-vsctl show-port-wred-ecn ge-1/1/1
```

Example:

1, Add bridge ports and configure wred in physical port te-1/1/4 queue0.

```
ovs-vsctl add-br br0
ovs-vsctl add-port br0 te-1/1/2 vlan_mode=trunk tag=1 -- set interface te-1/1/2 options:link_speed=1G
ovs-vsctl add-port br0 te-1/1/3 vlan_mode=trunk tag=1 -- set interface te-1/1/3 options:link_speed=1G
ovs-vsctl add-port br0 te-1/1/4 vlan_mode=trunk tag=1 -- set interface te-1/1/4 options:link_speed=1G

ovs-vsctl set interface te-1/1/4 wred_queues:0=@wred1 -- --id=@wred1 create wred_queue enable=true
min_thresh=100 max_thresh=200 drop_probability=50
```

2, enable ecn in port te-1/1/4.

```
ovs-vsctl set-port-wred-ecn te-1/1/4 true
ovs-vsctl show-port-wred-ecn te-1/1/4
```

3, add unicast flow entry.

```
ovs-ofctl add-flow br0 in_port=2,d1_dst=22:00:00:00:00:11,d1_vlan=1,ip,actions=push_mpls:0x8847,set_field:10-
\>mpls_label,set_queue:0,output:4
ovs-ofctl add-flow br0 in_port=3,d1_dst=22:00:00:00:00:22,d1_vlan=1,ip,actions=push_mpls:0x8847,set_field:20-
\>mpls_label,set_queue:0,output:4
ovs-appctl pica/dump-flows
```