April, 2004

Configuring QoS on the OmniStack 6300

ALCATEL

Table of Contents

Configuring QoS on OmniStack 6300

The purpose of that document is to detail the QoS features explained in the "OmniStack 6300-24 User Guide" Please refer to the user guide for more information.

The document will go through the following features:
1. "Port based" rate limiting
2. Queue mode
3. Class of Service and priority queues
4. ACL marker: 802.1p, ToS, DSCP stamping on egress packets
5. ACL
6. "Flow based" QoS using Service Policy, Class-Map and Policy-Map

1. Port based rate limiting

Bandwidth shaping is configurable on an interface basis.
You can shape incoming traffic for an interface to 10Mbps with the "***rate-limit input 10*** " interface command.
You can shape outgoing traffic for an interface to 20Mbps with the "***rate-limit output 20*** " interface command.
The granularity of the bandwidth is 1Mbps.

Example

To set ingress shaping to 10Mbps on interface 1/1 and set egress shaping to 20Mbps on interface 1/2

```
Console(config)#interface ethernet 1/1
Console(config-if)# rate-limit input 10
Console(config-if)# exit
Console(config)#interface ethernet 1/2
Console(config-if)# rate-limit input 20
Console(config-if)# exit
```

Limitation

By definition, shaping is done on an interface basis; all traffic for that interface will be shaped.
In order to make a flow based rate limiting rule, the user has to set a "Service Policy".

Interoperability with OmniSwitch 6624/6648 7700/7800 8800

The OmniSwitch 6624/6648 7700/7800 8800 switches do not have a port based rate limiting concept.
They use the "policy rules" to make the bandwidth shaping for traffic matching the "policy conditions".
They do a "flow based" rate limiting.
OmniSwitch 6300 can do both "port based" rate limiting and "flow based" rate limiting.

2. Queue mode

The switch has 8 priority queues per egress port.
You can configure the queue mode to a strict or a Weighted Round-Robin (WRR):

- WRR (default): shares bandwidth at the egress ports by using scheduling weights 1 2 4 6 8 10 12 14 for queues 0 through 7 respectively; the higher the weight, the higher the bandwidth for that queue
- Strict: services the egress queues in sequential order, transmitting traffic in the higher priority queues before servicing any lower priority queues

The queue mode is set for the entire switch. All interfaces are either in a strict mode or a WRR mode.

However, on WRR mode, the weights are configurable on an interface basis.
Since the queues are linked to the egress port, you will always modify the weights for the egress interface.

Example

To set queue mode to WRR and WRR bandwidth for queues serving interface 1/3

```
Console(config)# queue mode wrr
Console(config)#interface ethernet 1/3
Console(config-if)# queue bandwidth 1 1 1 2 2 2 2 15
Console(config-if)# exit
```

Interoperability with OmniSwitch 6624/6648 7700/7800 8800

Usually, there is no such queue mode in these switches. The queues are always in a strict mode.
The only modules that can be configured in a WRR mode are the High-Density modules GNI-U12/GNI-C12 on OmniSwitch 7700/7800 and GNI-U24/GNI-C24 on OmniSwitch 8800.
The only location to configure a WRR in your network is on OmniStack 6300-34 or High-Density modules.

3. Priority CoS (Class of service)

The switch has 8 priority queues per egress port.
The switch uses CoS to determine the priority of a packet.
Each CoS value is mapped to a priority queue using the CoS/Queue table.
The default table is:

Priority Queue	2	0	1	3	4	5	6	7
CoS value	0	1	2	3	4	5	6	7

CoS means a lot of things.
CoS is an input value between 0 and 7 the switch uses to determine the priority queue.
The key is to know where that CoS came from.
The switch has several ways to interpret a CoS value:

- Using the 802.1p from tagged packet, default mode
- Using the port default priority for untagged packets
- Using the IP precedence (TOS) from IP packets
- Using the IP DSCP (diffServ) from IP packets
- Using an ACL to map specific packets to a CoS

Interoperability with OmniSwitch 6624/6648 7700/7800 8800

OmniSwitch 7700/7800 and 8800 only support 4 queues per egress port.
OmniSwitch 6624/6648 supports 8 queues per egress port.

3.1 802.1p Priority (tagged packets)

By default, the switch uses the 802.1p value seen in the incoming packets to determine the CoS.
In that mode, CoS and 802.1p are the same.
The priority queues are chosen using the table:

Priority Queue	2	0	1	3	4	5	6	7
802.1p (CoS)	0	1	2	3	4	5	6	7

All incoming-tagged packets are prioritized based on the 802.1p value they carry on their VLAN header.
The outgoing packets still carry the original 802.1p value. Packets are never modified.

Interoperability with OmniSwitch 6624/6648 7700/7800 8800

These switches do the same but they use a different table.
For instance OmniSwitch 7700/7800 and OmniSwitch 8800 map the 802.1p values to only 4 priority queues.

Priority Queue	0	0	1	1	2	2	3	3
802.1p	0	1	2	3	4	5	6	7

Packets with 802.1p 0 get the lowest priority whereas they get priority 2 on OmniStack 6300-24

3.2 Default priority / Default 802.1p (untagged packets)

By default, the switch uses the 802.1p value seen in the incoming packets to determine the CoS.
For incoming untagged packets, the switch has to define a “default 802.1p”
The switch uses the interface “***switchport priority default”*** to determine the default 802.1p for incoming untagged packets on that interface.

Also, the switch will automatically set the 802.1p value with the "default priority" if the packet has to go out on a tagged interface. (The egress interface has to insert a VLAN header in the packet)

All incoming untagged packets are prioritized based on the interface default priority.
These packets will have their 802.1p value set with the default priority value when exiting the switch

By default, the default 802.1p is 0

- Incoming untagged packets will be queued to priority queue 2 (see CoS/queue table)
- When exiting the switch, these packets will have their 802.1p set to 0 on the VLAN header

When setting "***Console(config-if)# switchport priority default 7***" on an interface

- Incoming untagged packets will be queued to priority queue 7 (see CoS/queue table)
- When exiting the switch, these packets will have their 802.1p set to 7 on the VLAN header

Note

When "map ip precedence" or "map ip dscp" or ACL-CoS are configured, the switch does not use the interface default priority to queue the incoming untagged packets but "ip precedence", "ip dscp" or ACL that matches the packets to determine to priority queue.
(See next chapters)

Interoperability with OmniSwitch 6624/6648 7700/7800 8800

These switches also support a port default priority for untagged packets.
However, there will be an incompatibility for the default priority 0 1 and 2 since they are not mapped to the same priority queues.
Default priority 0 is the lowest priority whereas on OmniStack 6300-24, default priority is priority 2.

3.3 Mapping CoS value to egress priority queue

As explained before, the switch always maps a CoS to an egress priority queue.
The switch never directly maps an 802.1p value (or ToS/Dscp), or an interface default priority to a queue.
It first links the 802.1p (or ToS/Dscp) or the default interface priority to a CoS value and then maps that CoS value to the egress priority queue.
You can change the CoS/Queue table to force some CoS values to be mapped to some specific queues.
That will not modify the packets, it just modifies the queuing of the packets.

The table is set for the entire switch.

Ex: To map CoS 1 to priority queue 5
Console(config-if)# queue cos-map 5 1

Priority Queue	2	**5**	1	3	4	5	6	7
CoS value	0	**1**	2	3	4	5	6	7

Also, you can map all CoS values to the same queue.
That will give the same priority for all packets.
To set priority 0 for all CoS you will use the command:
Console(config-if)# queue cos-map 0 0 1 2 3 4 5 6 7

Priority Queue	0	0	0	0	0	0	0	0
CoS value	0	1	2	3	4	5	6	7

Limitation

CLI requires going to the interface command mode to change the table, even if the table is not interface-related.

Interoperability with OmniSwitch 6624/6648 7700/7800 8800

There is no CoS/queue concept. However, they do support the same idea with the "mapping" policies.
You can configure some policies to map an 802.1p/TOS/DSCP to another 802.1p/TOS/DSCP, using the map groups.
That will change both priority queue and the 802.1p/TOS/DSCP value inside the packet.

Also, the policies are configured on a packet flow basis whereas on OmniStack 6300-24 the CoS/queue table is set for the entire chassis.
OmniStack 6300-24 does not support the trusted/untrusted feature, but setting priority 0 for all CoS values is a similar way of making the switch untrusted (all packets have the same priority)

3.4 IP precedence Priority (IP packets)

By default, the switch uses the 802.1p value seen in the incoming-tagged packets or the interface default priority for incoming untagged packets to determine the CoS.
That is L2 priority.
What if you want to do a L3 priority and use the ToS from the IP header?
IP defines 2 priorities on the ToS byte

- IP Precedence or ToS: 3 first bits of ToS byte. Value between 0 and 7.
- IP DSCP or DiffServ: 6 first bits of ToS byte. Value between 0 and 63.

The switch can be configured to take CoS from IP Precedence.
The switch also uses an IP Precedence/CoS table to map each IP Precedence value to a CoS value.
That table can be modified.

All settings are set for the entire switch.

All incoming IP packets are prioritized based on the ToS value they carry on their IP header.
The outgoing packets still carry the original ToS byte. Packets are never modified.

That mode only applies for IP packets.
Of course, non-IP packets will still be prioritized using their 802.1p or the interface default priority.

To determine the priority queue, the switch takes the CoS mapped to the IP Precedence.
Then, the switch reads the CoS/Queue table to map the CoS value to a priority queue.
The default tables are:

Priority Queue	2	0	1	3	4	5	6	7
CoS	0	1	2	3	4	5	6	7

CoS	0	1	2	3	4	5	6	7
IP Precedence	0	1	2	3	4	5	6	7

To configure the switch to use “IP precedence”

```
Console(config)# map ip precedence
```

To map a “IP precedence” value to a “CoS” value

```
Console(config-if)# map ip precedence 3 cos 0
Console(config-if)# map ip precedence 0 cos 7
```

Priority Queue	2	0	1	3	4	5	6	7
CoS	0	1	2	3	4	5	6	7

CoS	**7**	1	2	**0**	4	5	6	7
IP Precedence	**0**	1	2	**3**	4	5	6	7

⇨ Packets with IP ToS 0 will be queued to priority queue 7 (CoS 7)
⇨ Packets with IP ToS 3 will be queued to priority queue 2 (CoS 0)

Limitation

CLI requires going to the interface command mode to change the table, even if the table is not interface-related.

Interoperability with OmniSwitch 6624/6648 7700/7800 8800

These switches do the same but they use a different table.
For instance OmniSwitch 7700/7800 and OmniSwitch 8800 map the ToS values to only 4 priority queues.

Priority Queue	0	0	1	1	2	2	3	3
ToS	0	1	2	3	4	5	6	7

Packets with ToS 0 get the lowest priority whereas they get priority 2 on OmniStack 6300-24.
There is no ToS/CoS concept. However, you can modify the ToS/queue mapping by using the mapping policies.
The mapping policies will modify both the priority and the egress packet (i.e. the egress 802.1p/ToS/Dscp are modified based on the ingress 802.1p/ToS/Dscp value, using the "policy map group")
On OmniStack 6300-24, selecting IP Precedence Priority never alters the egress packets.

3.5 IP DSCP Priority (IP packets)

The switch can be configured to take CoS from IP DSCP.
The switch also uses an IP DSCP/CoS table to map each IP DSCP value to a CoS value.
That table can be modified.

All settings are set for the entire switch.

All incoming IP packets are prioritized based on the DSCP value they carry on their IP header.
The outgoing packets still carry the original ToS byte. Packets are never modified.

That mode only applies for IP packets.
Of course, non-IP packets will still be prioritized using their 802.1p or the interface default priority.

To determine the priority queue, the switch takes the CoS mapped to the IP DSCP.
Then, the switch reads the CoS/Queue table to map the CoS value to a priority queue.
The default tables are:

Priority Queue	2	0	1	3	4	5	6	7
CoS	0	1	2	3	4	5	6	7

CoS	0	0	0	0	0	0	0	0	1	0	2	0	2	0	2	0
DSCP	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
CoS	2	0	3	0	3	0	3	0	3	0	4	0	4	0	4	0
DSCP	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
CoS	4	0	4	0	4	0	5	0	5	0	5	0	0	0	7	0
DSCP	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**
CoS	6	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0
DSCP	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**

To configure the switch to use "IP dscp" (previously set to "IP precedence")

```
Console(config)# no map ip precedence
Console(config)# map ip dscp
```

To map a "IP dscp" value to a "CoS" value

```
Console(config-if)# map ip dscp 52 cos 7
Console(config-if)# map ip dscp 21 cos 2
```

Priority Queue	2	0	1	3	4	5	6	7
CoS	0	1	2	3	4	5	6	7

CoS	0	0	0	0	0	0	0	0	1	0	2	0	2	0	2	0
DSCP	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
CoS	2	0	3	0	3	**2**	3	0	3	0	4	0	4	0	4	0
DSCP	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
CoS	4	0	4	0	4	0	5	0	5	0	5	0	0	0	7	0
DSCP	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**
CoS	6	0	0	0	**7**	0	0	0	7	0	0	0	0	0	0	0
DSCP	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**

⇨ Packets with IP DSCP 21 will be queued to priority queue 1 (CoS 2)
⇨ Packets with IP DSCP 52 will be queued to priority queue 7 (CoS 7)

Limitation

CLI requires going to the interface command mode to change the table, even if the table is not interface-related.

Interoperability with OmniSwitch 6624/6648 7700/7800 8800

These switches do the same but they use a different table.
For instance OmniSwitch 7700/7800 and OmniSwitch 8800 map the DSCP values to only 4 priority queues.

Priority Queue	0	1	2	3
DSCP	0-15	16-31	32-47	48-63

Packets with DSCP 63 get the highest priority whereas they get priority 2 on OmniStack 6300-24.
There is no DSCP/CoS concept. However, you can modify the DSCP/queue mapping by using the mapping policies.
The mapping policies will modify both the priority and the egress packet (i.e. the egress 802.1p/ToS/Dscp are modified based on the ingress 802.1p/ToS/Dscp value, using the "policy map group")
On OmniStack 6300-24, selecting IP DSCP Priority never alters the egress packets.

3.6 ACL Priority (CoS ACL)

Instead of reading the CoS from the L2 packet (802.1p) or L3 packets (IP precedence or IP dscp), the switch can use ACLs to map a specific type of packet to a CoS.
With that feature you can give priority for a specific type of traffic.
You can map both IP ACLs and MAC ACLs to CoS.
After mapping an ACL to CoS, the switch still reads the CoS/Queue table to map the CoS value to a priority queue.

All incoming packets matching an ACL will be prioritized based on the CoS value that is mapped to the ACL.
The outgoing packets are never modified.

Packets that do not match the ACL will still be prioritized with 802.1p, interface default priority, IP Precedence or IP dscp.

Example

To give priority 7 to traffic from mac 00-00-00-00-00-03 (CoS 7 gives queue 7)
To give priority 0 to traffic from ip 2.0.0.3 (CoS 1 gives queue 0)
(See ACL chapter for ACL configuration)

```
Console(config)# access-list ip standard FROM_IP_3
Console(config-std-acl)# permit host 2.0.0.3
Console(config-std-acl)# exit

Console(config)# access-list mac FROM_MAC_3
Console(config-mac-acl)# permit host 00-00-00-00-00-03 any
Console(config-mac-acl)#exit

Console(config)# access-list ip mask-precedence in
Console(config-ip-mask-acl)#mask host
Console(config-ip-mask-acl)#exit

Console(config)#access-list mac mask-precedence in
Console(config-mac-mask-acl)#mask host any
Console(config-mac-mask-acl)#exit

Console(config)#interface ethernet 1/1
Console(config-if)#map access-list ip FROM_IP_3 cos 1
Console(config-if)#map access-list mac FROM_MAC_3 cos 7
```

Limitation

- MAC ACL is always executed first. If the same packet matches both MAC and IP ACLs, then packet will be mapped to CoS defined in the MAC ACL. In the example, packet with mac 00-00-00-00-00-03 **and** ip 2.0.0.3 will be queued to priority queue 7.

- ACL-CoS is configured on an interface basis. Only one IP ACL and/or one MAC ACL per port.

Interoperability with OmniSwitch 6624/6648 7700/7800 8800

ACL-CoSs are fully compliant with the ***"priority policies"*** configured on the OmniSwitches.
They both give priority on a packet flow basis by using an ACL or a "policy condition" to specify the packet flow.
However, an ACL-CoS is always attached to an interface whereas a ***"priority policy"*** is chassis wide.

OmniSwitch 6300 does not need a "qos apply" (that flushes the mac and arp tables) after creating a new ACL.
ACLs are enforced to a dedicated filtering hardware that does not interact with "source learning".
ACLs always work at "wire speed", there is no software processing at all.

4. ACL Marker: 802.1p, ToS, Dscp stamping

The previous chapters explained how a packet is getting prioritized using the CoS.
With CoS priority, egress packets are never modified; only the queuing differs with the different configurations.
To modify the 802.1p, IP Precedence or IP DSCP value on the egress packets, the switch uses a feature called "ACL Marker".
The feature allows the stamping of a specific 802.1p, IP precedence or IP dscp value to a packet matching an ACL.

With MAC ACL you can only set the priority (802.1p value)
With IP ACL, you can set:

- Priority (802.1p value)
- Precedence (TOS value 0-7)
- Dscp (value 0-63)
- Priority and precedence
- Priority and dscp

The ACL marker will always stamp the required value in the egress packet.
However, the stamping will also modify the priority of the packet.
As explained before, the priority is always mapped to a CoS value.
There are 3 CoS modes on incoming packets:

- 802.1p value from packet of interface default priority (default mode)
- IP precedence if switch is configured with "map ip precedence"
- IP dscp if switch is configured with "map ip dscp"

With ACL marker, if the stamped value type (802.1p, IP precedence or IP dscp) corresponds to the CoS mode, the stamped value will change the CoS value and therefore the priority queue

Example

The switch uses the default CoS mode. (switch takes CoS from 802.1p seen on incoming packets)
Tagged packets are sent to interface 1/1.
We configure an IP ACL and a MAC ACL to stamp 802.1p, IP Precedence or IP dscp.
You have the choice between 2 IP ACL Markers

- (1): to stamp "802.1p and IP precedence"
- (2): to stamp "802.1p and ip dscp".

Then we sniff the egress packets to verify the stamping was correctly made.

```
Console(config)# access-list ip standard FROM_IP_3
Console(config-std-acl)# permit host 2.0.0.3
Console(config-std-acl)# exit

Console(config)# access-list mac FROM_MAC_3
Console(config-mac-acl)# permit host 00-00-00-00-00-03 any
Console(config-mac-acl)#exit

Console(config)# access-list ip mask-precedence in
Console(config-ip-mask-acl)#mask host
Console(config-ip-mask-acl)#exit
```

Console(config)#access-list mac mask-precedence in
Console(config-mac-mask-acl)#mask host any
Console(config-mac-mask-acl)#exit

Console(config)#interface ethernet 1/1
Console(config-if)#match access-list mac FROM_MAC_3 set priority 7
Console(config-if)#match access-list ip FROM_IP_3 set priority 1 precedence 3 (1)
Or
Console(config-if)#match access-list ip FROM_IP_3 set priority 1 dscp 62 (2)

Packets from ip 2.0.0.3 will match ACL FROM_IP_3.
With ACL Marker (1)

- ⇨ 802.1p will be set to 1 and IP Precedence to 3 in egress packets
- ⇨ CoS will be 1 since we are in the default CoS mode (CoS comes from 802.1p)
- ⇨ priority queue will be 0 (CoS 1 gives queue 0)

With ACL Marker (2)

- ⇨ 802.1p will be set to 1 and IP DSCP to 62 in egress packets
- ⇨ CoS will be 1 since we are in the default CoS mode (CoS comes from 802.1p)
- ⇨ priority queue will be 0 (CoS 1 gives queue 0)

Packets from mac 00-00-00-00-00-03 will match ACL FROM_MAC_3

- ⇨ 802.1p will be set to 7 in egress packet
- ⇨ CoS will be 7 since we are in the default CoS mode (CoS comes from 802.1p)
- ⇨ priority queue will be 7 (CoS 7 gives queue 7)

Limitation

- ♦ MAC ACL always executed first. If the same packet matches both MAC and IP ACLs, then the packet will be stamped with 802.1p coming from MAC ACL. In the example, packet with mac 00-00-00-00-00-03 **and** ip 2.0.0.3 will be stamped with 802.1p 7. Packet will also be queued to priority queue 7.
- ♦ MAC ACL can only set the priority (802.1p value)
- ♦ ACL Marker is configured on an interface basis. Only one IP ACL and/or one MAC ACL per port.

Note

When a packet matches both MAC and IP ACLs, the 802.1p will always be stamped from the MAC ACL. However, the IP ACL is still executed for the “set precedence” or “set dscp”. In the example, packet with mac 00-00-00-00-00-03 **and** ip 2.0.0.3 will be stamped with 802.1p 7 and precedence 3 (1) or dscp 62 (2)

Issue

“set dscp” is currently broken (dscp value is not changed on egress packets).

Interoperability with OmniSwitch 6624/6648 7700/7800 8800

ACL Markers are fully compliant with the ***“stamp policies”*** configured on the OmniSwitches.
They both give priority and modify egress packets on a packet flow basis.
However, an ACL Marker is always attached to an interface whereas a ***“stamp policy”*** is chassis wide.

OmniSwitch 6300 does not need a “qos apply” (that flushes the mac and arp tables) after creating a new ACL.
ACLs are enforced to a dedicated filtering hardware that does not interact with “source learning”.
ACLs always work at “wire speed”, there is no software processing at all.

5. ACL

The switch supports both ingress ACL and egress ACL to filter incoming and outgoing traffic on an interface.
The switch has 3 kinds of ACLs:

- ♦ Standard IP: to filter source ip addresses
- ♦ Extended IP: to filter L3/L4 header packets
- ♦ MAC: to filter L2 header packets

ACLs are active on an interface basis.
Each interface can have multiple ACLs, but only one of each type:

- 1 Ingress IP ACL
- 1 Ingress MAC ACL
- 1 Egress IP ACL
- 1 Egress MAC ACL

Precedence

An ACL can have multiple rules to match multiple flows.
The order the rules are interpreted is **NOT** defined by the order you created the rules.
When a packet could match multiple rules, the switch will use the ACL masks to know what rule has to be analyzed first.
The precedence of the rules is based on the order you created the ACL masks

ACL masks

Within an ACL, a rule defines what bits are expected to be seen in the packet.
That does not tell the switch what are the bits to read inside the packets.
You need to tell the switch what are the bits you want to check when analyzing the packets.
This is the meaning of an ACL mask.
Then, once the switch knows what to analyze, it uses the ACL rules to compare to read bits with the expected values.
You will have to define 4 ACL masks:

- IP ACL mask in : to know how to analyze incoming packet using ip ACLs
- IP ACL mask out : to know how to analyze outgoing packet using ip ACLs
- MAC ACL mask in : to know how to analyze incoming packet using mac ACLs
- MAC ACL mask out : to know how to analyze outgoing packet using mac ACLs

The mask simply defines **«the type of the rule »**
If a rule defines a source ip host 2.0.0.1, the type is « ***source ip host*** » and you need to configure a source ip host mask.
If a second rule defines a second ip host 2.0.0.2, the type is also « ***source ip host*** »
There is no need to create a second mask; both rules have the same type.
They both use the same mask.

You need to create a mask for every «type of rule » you created.

The order you created the mask define the order the switch has to analyze the rules within an ACL.

Example:
Create an ACL to deny incoming traffic from subnet 2.0.0.0/8 and going to host 3.0.0.1 with ip dscp 56 source tcp port 3000 and destination tcp port 80

⇨ To create the ACL

```
Console(config)# access-list ip extended ACL_IN
Console(config-ext-acl)# deny tcp 2.0.0.0 255.0.0.0 host 3.0.0.1 dscp 56 source-port 3000
destination-port 80
Console(config-ext-acl)# exit
```

⇨ To create the mask (since we wanted to filter incoming traffic we need to set a ip mask in)

```
Console(config)# access-list ip mask-precedence in
Console(config-ip-mask-acl)# mask protocol 255.0.0.0 host dscp source-port 65535 destination-port
65535
Console(config-ip-mask-acl)#exit
```

The different options means:

- Protocol: need to check to protocol header in L3 header
- 255.0.0.0: need to check only the first byte of the source ip address in the L3 header, since we define a class A mask in the rule
- host: need to check the full destination ip address in the L3 header
- dscp: need to check the IP dscp in the L3 header
- source-port 65535: need to check the 2 bytes of the source port in the L4 header. 65535 means all bits have to be checked.
- destination-port 65535: need to check the 2 bytes of the destination port in the L4 header

Note

- The “***show access-list***” command always displays the rule in the precedence order, not the order they were created.
- Extended IP ACLs **do support dscp** even if user manual still says “ACL DSPC not supported”.

Limitations

- Up to 32 ACLs
- Up to 32 rules per ACL, however it is recommended not to exceed 20 rules
- Up to 7 entries per ACL mask
- Masks are shared among all interfaces
- Masks have to be defined before binding an ACL to an interface
- The order in which the mask are entered defines the precedence of the rules
- For egress ACL, all rules have to be “deny”
- Egress ACLs do not support the explicit “deny any any”
- MAC ACL always checked first. If a packet match both MAC ACL and IP ACL, only the MAC ACL will be executed

Interoperability with OmniSwitch 6624/6648 7700/7800 8800

ACLs are fully compliant with the ***“policy condition”*** configured on the OmniSwitches.
They both define the conditions a packet has to match.
However, an ACL is always attached to an interface whereas a ***“policy”*** is chassis wide.
Also, ACLs have more options such as frame type, tcp control flag or masks.
OmniSwitch 6624/6648 7700/7800 8800 can support up to 2048 policies.

OmniSwitch 6300 does not need a “qos apply” (that flushes the mac and arp tables) after creating a new ACL.
ACLs are enforced to a dedicated filtering hardware that does not interact with “source learning”.
ACLs always work at “wire speed”, there is no software processing at all.

5.1 Ip Standard ACL

```
Console(config)# access-list ip standard IN
Console(config-std-acl)# permit 2.0.0.0 255.255.255.0
Console(config-std-acl)# deny host 2.0.0.2
Console(config-std-acl)# exit

Console(config)# access-list ip standard OUT
Console(config-std-acl)# deny host 2.0.0.3
Console(config-std-acl)# exit

Console(config)# access-list ip mask-precedence in
Console(config-ip-mask-acl)#mask host
Console(config-ip-mask-acl)#mask 255.255.255.0
Console(config-ip-mask-acl)#exit

Console(config)# access-list ip mask-precedence out
Console(config-ip-mask-acl)#mask host
Console(config-ip-mask-acl)#exit

Console(config)# interface ethernet 1/3
Console(config-if)#ip access-group IN in
Console(config-if)#ip access-group OUT out
Console(config-if)#exit
```

You can verify the precedence order with the show command.
Because of the mask order, ACL IN first analyses the full ip address, then analyses the subnet.
Packet with source ip 2.0.0.2 will be denied because it first matches “deny host 2.0.0.2”.

```
Console#show access-list
IP standard access-list IN:
  deny host 2.0.0.2
  permit 2.0.0.0 255.255.255.0
```

```
IP standard access-list OUT:
  deny host 2.0.0.3
IP ingress mask ACL:
  mask host any
  mask 255.255.255.0 any
IP egress mask ACL:
  mask host any
Console#
```

5.2 IP extended ACL

```
Console(config)# access-list ip extended ACL_IN
Console(config-ext-acl)# deny tcp any any
Console(config-ext-acl)# permit tcp 2.0.0.0 255.255.0.0 2.0.0.0 255.255.0.0 precedence 1 source-port 1984
1984 destination-port 80
Console(config-ext-acl)# exit

Console(config)# access-list ip mask-precedence in
Console(config-ip-mask-acl)# mask protocol 255.255.0.0 255.255.0.0 precedence source-port 65535
destination-port 65535
Console(config-ip-mask-acl)# mask protocol any any
Console(config-ip-mask-acl)#exit

Console(config)# access-list ip extended ACL_OUT
Console(config-ext-acl)# deny tcp 2.0.0.0 255.255.0.0 any control-flag 2 2
Console(config-ext-acl)# exit

Console(config)# access-list ip mask-precedence out
Console(config-ip-mask-acl)#mask protocol 255.255.0.0 any control-flag 2
Console(config-ip-mask-acl)#exit

Console(config)# interface ethernet 1/3
Console(config-if)# ip access-group ACL_IN in
Console(config-if)# ip access-group ACL_OUT out
Console(config-if)# exit
```

You can verify the precedence order with the show command.
ACL_IN has 2 rules

- Deny all TCP packets
- Accept all TCP packets from subnet 2.0.0.0/16 with IP precedence 1, source TCP port between 1984-2047 and destination port 80. Any numbers between 1984-2047 give 1984 when using mask 1984

ACL_OUT has one rule

- Deny all SYN packets coming from subnet 2.0.0.0/16. On the TCP header, SYN flag set bit number 6 in the Control Flag byte. Therefore, using 2 with mask 2 will match any Control Flag byte with SYN set.

Because of the mask order, ACL_IN first analyses the permit rule, then the deny rule.
Incoming TCP packet 2.0.0.2:2000 precedence 1 -> 2.0.0.8:80 will be accepted even it match the first rule “deny tcp any any”

```
Console#show access-list
IP extended access-list ACL_IN:
  permit tcp 2.0.0.0 255.255.0.0 2.0.0.0 255.255.0.0 precedence 1 source-port 1984 1984 destination-port
80
  deny tcp any any
...
```

5.3 Mac ACL

```
Console(config)# access-list ip standard IPIN
Console(config-std-acl)# deny host 2.0.0.1
Console(config-std-acl)# exit
Console(config)# access-list ip mask-precedence in
Console(config-ip-mask-acl)# mask host
Console(config-ip-mask-acl)# exit

Console(config)# access-list mac MACIN
Console(config-mac-acl)# permit any host 00-00-00-00-00-03 vid 2 ethertype 800
Console(config-mac-acl)# exit
access-list mac mask-precedence in
Console(config-mac-mask-acl)# mask any host vid 4095 ethertype ffff
Console(config-mac-mask-acl)# exit

Console(config)# access-list mac MACOUT
Console(config-mac-acl)# deny tagged-802.3 host 00-00-00-00-00-04 any vid 2
Console(config-mac-acl)# exit
access-list mac mask-precedence out
Console(config-mac-mask-acl)# mask pktformat host any vid 4095
Console(config-mac-mask-acl)# exit

Console(config)# interface ethernet 1/3
Console(config-if)# ip access-group IPIN in
Console(config-if)# mac access-group MACIN in
Console(config-if)# mac access-group MACOUT out
Console(config-if)# exit
```

- IP packets from 2.0.0.1 are denied.
- EthernetII packets going to 00-00-00-00-03 on vlan 2 are permitted.
- Tagged 802.3 SNAP packets from 00-00-00-00-00-04 on vlan 2 are denied.
- IP packets with destination mac 00-00-00-00-00-03 and source ip 2.0.0.1 on vlan 2 are permitted (they will not be dropped by IPIN since MAC ACL is always executed first)

6 Service Policy, Class-Map and Policy-Map

OmniSwitch 6300 supports a full flow based QoS.
On every interface, the switch can classify ingress and egress packets and give them a specify action.
To classify packets or flows, the user configures a “Class-Map”
To set the actions for a “Class-Map”, the user configures a “Policy-Map”
To bind a Policy to an interface, the user configures a “Service-Policy” for that interface (ingress or egress).

The switch can classify packets on:

- ACLs
- IP Precedence
- IP Dscp
- Vlan

A policy can have the following actions:

- set CoS
- set Ip Precedence
- set Ip Dscp
- set a rate limiter (rate + burst size)
- set an “ exceed-rate” action to know what to do with packets that exceed the rate limiter (2 options: drop or “set ip dscp”)

Class-Map:

To classify traffic, the user creates a “Class-Map”. Each “Class-Map” defines a specific type of traffic that is fully identified by the “Class-Map” name.
Within a “Class-Map”, the user has to define the rules needed to classify packets.
There are 4 types of rule:

- ACLs. To classify packets with a mac, ip standard, or ip extended ACL. Useful to do a L2/L3/L4 classification
- IP Precedence. To classify packets with the “precedence” value seen in packet ToS byte
- IP Dscp. To classify packets with the “dscp” value seen in packet ToS byte
- Vlan. To classify packets with their vlan id

A “Class-Map” is always configured to “match-any”. If a Class has multiple rules, a packet will match the Class as soon as it matches one of the rules.
Therefore, you will mostly create only one rule per “Class-Map”.

Policy-Map:

Once traffic is classified, you can configure multiple actions for every type of classified traffic or “Class-Map”.
When configuring a “Policy-Map”, you specify a policy rule to define the actions for a specific “Class-Map”.
A “Policy-Map” can have multiple policy rules so that different “Class-Map” will have different actions.
For each “Class-Map” the following actions can be configured:

- Set CoS, IP precedence or IP dscp. A Policy can change the 802.1p value, the IP precedence or the IP Dscp value on packets matching the “Class-Map”
  At least one “set” action has to be configured.
- Rate limiting. A policy can meter a flow using a maximum rate in kbps and a burst size in bytes.
- Exceed . When flow exceeds the rate limiting, a policy can either drop the traffic or set the “ip dscp” to a specific value.

The granularity of the “maximum rate” is 1Mbps.
Giving a rate between 0 and 999 (kbps) will set the meter to 0 bps
Giving a rate between 1000 and 1999 (kbps) will set the meter to 1 Mbps
Giving a rate between 2000 and 2999 (kbps) will set the meter to 2 Mbps

Internal priority for traffic

Giving a “set cos”, “set ip precedence” or “set ip dscp” will change the internal priority of the traffic.
Ingress traffic is normally queued to one of the 8 priority queues of the egress ports.
This is done using the “queue cos-table” to give the priority queue from the CoS.
CoS could come from 802.1p from packets, switchport priority, Ip precedence from packets when “map ip precedence” is set or Ip dscp from packets when “map ip dscp” is set.
With “Policy-Map”, the prioritization works as follow:

- Action “set cos”: Internal priority will always come from the “set cos” value (queue cos-map table)
- Action “set ip precedence”: Internal priority will always come from the “set ip precedence” value.
  The “[no] map ip precedence” command will not change the internal priority
  Changing the “ip precedence cos table” will not change the internal priority
- Action “set ip dscp” : Internal priority will always come from the “set ip dscp” value.
  The “[no] map ip dscp” command will not change the internal priority
  Changing the “ip dscp cos table” will not change the internal priority

Service-Policy:

The last step is to attach a “Policy-Map” to an interface.
On an interface, a policy can be “input” or “ouput” depending on which side you want to classify traffic.
An input policy is used for ingress traffic on that interface.
An output policy is used for egress traffic on that interface.
An interface can only have one ingress policy and/or one egress policy.

The common mistake when attaching a policy to an interface is to forget to configure the ACL masks.
Indeed, like an ACL, you have to configure the proper masks so that the switch can classify your traffic.

ALC masks

Every “Class-Map” rule needs an ACL mask.
If the “Service-Policy” using that “Class-Map” is ingress, you have to set the proper ***Ingress ACLs masks***.
If the “Service-Policy” using that “Class-Map” is egress, you have to set the proper ***Egress ACLs masks***.
For “Class-Map” using an ACL rules, it is quite obvious to create the proper ACL masks.
For “Class-Map” using an “ip precedence”, “ip dscp” or “vlan” rule, it is not so obvious to create an ACL mask since you have not created any ACLs.

It is mandatory to create the following ACL masks when using a vlan, ip precedence or ip dscp rule

- IP precedence rule on input/ingress service policy
  Console(config)# access-list ip mask-precedence in
  Console(config-ip-mask-acl)# mask any any precedence
  Console(config-ip-mask-acl)# exit

- IP precedence rule on output/egress service policy
  ```
  Console(config)# access-list ip mask-precedence out
  Console(config-ip-mask-acl)# mask any any precedence
  Console(config-ip-mask-acl)# exit
  ```
- IP dscp rule on input/ingress service policy
  ```
  Console(config)# access-list ip mask-precedence in
  Console(config-ip-mask-acl)# mask any any dscp
  Console(config-ip-mask-acl)# exit
  ```
- IP dscp rule on output/egress service policy
  ```
  Console(config)# access-list ip mask-precedence out
  Console(config-ip-mask-acl)# mask any any dscp
  Console(config-ip-mask-acl)# exit
  ```
- vlan rule on input/ingress service policy
  ```
  Console(config)# access-list mac mask-precedence in
  Console(config-mac-mask-acl)# mask any any vid
  Console(config-mac-mask-acl)# exit
  ```
- vlan rule on output/egress service policy
  ```
  Console(config)# access-list mac mask-precedence out
  Console(config-mac-mask-acl)# mask any any vid
  Console(config-mac-mask-acl)# exit
  ```

Limitations

- Only 1 input Policy can be configured per interface
- Only 1 output Policy can be configured per interface
- A Policy can have up to 8 actions on FastEthernet interfaces
- A Policy can have up to 128 actions on GigabitEthernet interfaces
- A Class is always “match-any”, a packet matches a Class if it matches one of the rules defined in the Class
- Always create a “vlan mask” when using a “vlan” rule in a Class-Map
- Always create a “ip precedence mask” when using an “ip precedence” rule in a Class-Map
- Always create a “ip dscp mask” when using an “ip dscp” rule in a Class-Map
- A policy has to have at least one “set” action. A policy with only one rate limiting action is not allowed
- “output” service policy do not support rate limiting. If a rate limiting policy is attached to an interface as “output”, the rate limiting action will be ignored.

Issues

“Set ip dscp” action does not work, DSCP value is always set to 0.
Also, the Internal priority is always the same.

Interoperability with OmniSwitch 6624/6648 7700/7800 8800

This feature is the equivalent of the “policy condition”, “policy action”, “policy rule” feature on OmniSwitch 6624/6648 7700/7800 8800.
Also, this feature allows the user to create some “mapping” policies.
For instance, you can create a “Class-Map” ToS 5 with action “set ToS 7” in order to map ToS 5 to ToS 7.
OmniSwitch 6300 does not need a “qos apply” (that flushes the mac and arp tables) after creating a new policy.
Policies are enforced to a dedicated filtering hardware that does not interact with “source learning”.
Polices always work at “wire speed”, there is no software processing at all.

Examples

With the following setup, we will explain several configurations.

```
IXIA1 → 1/1 OmniSwitch 6300 1/3 → 1/3 OmniSwitch 7700 1/1 → IXIA3
IXIA2 → 1/2                                           1/2 → IXIA4
```

All ports are 100Mbps Full and tagged for vlan 2
2 flows are running at 95Mbps each.

- Flow1: IXIA1->IXIA3 000000:000003/2.0.0.3 -> 000000:000003/2.0.0.1
  with 802.1p 0 and ToS byte 0xC8 (ip precedence or tos 3 and dscp 26)
- Flow2: IXIA2->IXIA4 000000:000004/2.0.0.4 -> 000000:000003/2.0.0.2
  with 802.1p 3 and ToS byte 0x30 (ip precedence or tos 1 and dscp 12)

Note: The IP header has one byte called ToS byte. This byte is interpreted as follow:

- bits 0-1-2 give the ip precedence (0-7) also called tos
- bits 0-1-2-3-4-5 give the ip dscp (0-63) also called dscp
- bits 6-7 are not used and are usually 0

For instance, ToS byte 0x30 (00110000) gives precedence 1 (001) and dspc 12 (001100)

6.1 Ingress "Service Policy": 1 "Policy Map" + 1 vlan "Class Map" with action set cos

We want to set 802.1p to 7 for traffic in vlan 2.

```
Console(config)# access-list mac mask-precedence in
Console(config-mac-mask-acl)# mask any any vid
Console(config-mac-mask-acl)# exit

Console(config)# class-map VLAN2 match-any
Console(Config-cmap)# match vlan 2
Console(config-cmap)# exit

Console(config)# policy-map P1
Console(Config-pmap)# class VLAN2
Console(Config-pmap-c)# set cos 7
Console(config-pmap-c)# exit
Console(Config-pmap)# exit

Console(config)# interface ethernet 1/1
Console(Config-if)# service-policy input P1
Console(Config-if)# exit
Console(config)# interface ethernet 1/2
Console(Config-if)# service-policy input P1
Console(Config-if)# exit
```

Flow1 and Flow2 have the same throughput (same priority): both flows are set to CoS 7
Packets exiting OmniSwitch6300 have 802.1p 7 and ToS byte unchanged for Flow1
Packets exiting OmniSwitch6300 have 802.1p 7 and ToS byte unchanged for Flow2

6.2 Ingress "Service Policy": 1 "Policy Map" + 2 acl "Class Map" with action set ip precedence

We want to:

- set ip precedence 7 for flow1 (matching ACL F1)
- set ip precedence 5 for flow2 (matching ACL F2)

```
Console(config)# access-list ip standard F1
Console(config-std-acl)# permit host 2.0.0.3
Console(config-std-acl)# exit

Console(config)# access-list ip standard F2
Console(config-std-acl)# permit host 2.0.0.4
Console(config-std-acl)# exit

Console(config)# access-list ip mask-precedence in
Console(config-ip-mask-acl)#mask host
Console(config-ip-mask-acl)#exit

Console(config)# class-map FLOW1 match-any
Console(Config-cmap)# match access-list F1
Console(config-cmap)# exit

Console(config)# class-map FLOW2 match-any
Console(Config-cmap)# match access-list F2
Console(config-cmap)# exit

Console(config)# policy-map P1
Console(Config-pmap)# class FLOW1
Console(Config-pmap-c)# set ip precedence 7
Console(config-pmap-c)# exit
Console(Config-pmap)# class FLOW2
Console(Config-pmap-c)# set ip precedence 5
Console(config-pmap-c)# exit
Console(Config-pmap)# exit

Console(config)# interface ethernet 1/1
Console(Config-if)# service-policy input P1
Console(Config-if)# exit
Console(config)# interface ethernet 1/2
Console(Config-if)# service-policy input P1
Console(Config-if)# exit
```

Flow1 is getting prioritized: “set ip precedence 7” gives higher priority than “set ip precedence 5”
Packets exiting OmniSwitch6300 have 802.1p unchanged and ToS byte precedence 7 for Flow1
Packets exiting OmniSwitch6300 have 802.1p unchanged and ToS byte precedence 5 for Flow2

6.3 Ingress “Service Policy”: 1 “Policy Map” + 2 acl “Class Map” with action set ip dscp

We want to:

- set ip dscp 63 for flow1 (matching ACL F1)
- set ip dscp 62 for flow2 (matching ACL F2)

```
Console(config)# access-list ip standard F1
Console(config-std-acl)# permit host 2.0.0.3
Console(config-std-acl)# exit

Console(config)# access-list ip standard F2
Console(config-std-acl)# permit host 2.0.0.4
Console(config-std-acl)# exit

Console(config)# access-list ip mask-precedence in
Console(config-ip-mask-acl)#mask host
Console(config-ip-mask-acl)#exit

Console(config)# class-map FLOW1 match-any
Console(Config-cmap)# match access-list F1
Console(config-cmap)# exit

Console(config)# class-map FLOW2 match-any
Console(Config-cmap)# match access-list F2
Console(config-cmap)# exit

Console(config)# policy-map P1
Console(Config-pmap)# class FLOW1
Console(Config-pmap-c)# set ip dscp 63
Console(config-pmap-c)# exit
Console(Config-pmap)# class FLOW2
Console(Config-pmap-c)# set ip dscp 62
Console(config-pmap-c)# exit
Console(Config-pmap)# exit

Console(config)# interface ethernet 1/1
Console(Config-if)# service-policy input P1
Console(Config-if)# exit
Console(config)# interface ethernet 1/2
Console(Config-if)# service-policy input P1
Console(Config-if)# exit
```

Flow1 and Flow2 have the same throughput

Packets exiting OmniSwitch6300 have 802.1p unchanged and ToS byte reset to 0 for both flows.
That is not correct, we expected to have dscp 63 for Flow1 and dscp 62 for Flow2

6.4 Ingress “Service Policy”: 1 “Policy Map” + 1 “ip precedence” and 1 “ip dscp” “Class Map” with action set cos and set ip precedence

We want to:

- set cos 7 for flow1 (packets with ToS 3)
- set ip precedence 5 for flow2 (packets with DSCP 12)

```
Console(config)# access-list ip mask-precedence in
Console(config-ip-mask-acl)# mask any any precedence
Console(config-ip-mask-acl)# mask any any dscp
Console(config-ip-mask-acl)#exit

Console(config)# class-map FLOW1 match-any
Console(Config-cmap)# match ip precedence 3
Console(config-cmap)# exit

Console(config)# class-map FLOW2 match-any
Console(Config-cmap)# match ip dscp 12
Console(config-cmap)# exit

Console(config)# policy-map P1
Console(Config-pmap)# class FLOW1
Console(Config-pmap-c)# set cos 7
Console(config-pmap-c)# exit
Console(Config-pmap)# class FLOW2
Console(Config-pmap-c)# set ip precedence 5
Console(config-pmap-c)# exit
Console(Config-pmap)# exit

Console(config)# interface ethernet 1/1
Console(Config-if)# service-policy input P1
Console(Config-if)# exit
Console(config)# interface ethernet 1/2
Console(Config-if)# service-policy input P1
Console(Config-if)# exit
```

Flow1 is getting prioritized : “set cos 7” gives higher priority than “set ip precedence 5”
Packets exiting OmniSwitch6300 have 802.1p 7 and ToS byte unchanged for Flow1
Packets exiting OmniSwitch6300 have 802.1p unchanged and ToS byte precedence 5 for Flow2

6.5 Ingress “Service Policy”: 1 “Policy Map” + 1 “ip precedence” and 1 “ip dscp” “Class Map” with action set cos,set ip precedence and rate limiting (exceed drop)

We want to:

- set cos 7 for flow1 (packets with ToS 3) and shape it to 1Mbps
- set ip precedence 5 for flow2 (packets with DSCP 12) and shape it to 2Mbps

```
Console(config)# access-list ip mask-precedence in
Console(config-ip-mask-acl)# mask any any precedence
Console(config-ip-mask-acl)# mask any any dscp
Console(config-ip-mask-acl)#exit

Console(config)# class-map FLOW1 match-any
Console(Config-cmap)# match ip precedence 3
Console(config-cmap)# exit

Console(config)# class-map FLOW2 match-any
Console(Config-cmap)# match ip dscp 12
Console(config-cmap)# exit
Console(config)# policy-map P1
Console(Config-pmap)# class FLOW1
Console(Config-pmap-c)# set cos 7
Console(Config-pmap-c)# police 1000 1000 exceed-action drop
Console(config-pmap-c)# exit
Console(Config-pmap)# class FLOW2
Console(Config-pmap-c)# set ip precedence 5
Console(Config-pmap-c)# police 2000 1000 exceed-action drop
Console(config-pmap-c)# exit
Console(Config-pmap)# exit

Console(config)# interface ethernet 1/1
Console(Config-if)# service-policy input P1
Console(Config-if)# exit
Console(config)# interface ethernet 1/2
Console(Config-if)# service-policy input P1
Console(Config-if)# exit
```

Flow1 is shaped to 1Mbps
Flow2 is shaped to 2Mbps
Packets exiting OmniSwitch6300 have 802.1p 7 and ToS byte unchanged for Flow1
Packets exiting OmniSwitch6300 have 802.1p unchanged and ToS byte precedence 5 for Flow2

6.6 Ingress “Service Policy”: 1 “Policy Map” + 1 “ip precedence” and 1 “ip dscp” “Class Map” with action set cos, set ip precedence and rate limiting (exceed set ip dscp)

We want to:

- set cos 7 for flow1 (packets with ToS 3) and “set ip dscp to 63” if flow1 exceeds 1Mbps
- set ip precedence 5 for flow2 (packets with DSCP 12) and “set ip dscp to 63” if flow1 exceeds 2Mbps

```
Console(config)# access-list ip mask-precedence in
Console(config-ip-mask-acl)# mask any any precedence
Console(config-ip-mask-acl)# mask any any dscp
Console(config-ip-mask-acl)#exit


Console(config)# class-map FLOW1 match-any
Console(Config-cmap)# match ip precedence 3
Console(config-cmap)# exit

Console(config)# class-map FLOW2 match-any
Console(Config-cmap)# match ip dscp 12
Console(config-cmap)# exit

Console(config)# policy-map P1
Console(Config-pmap)# class FLOW1
Console(Config-pmap-c)# set cos 7
Console(Config-pmap-c)# police 1000 1000 exceed-action set ip dscp 63
Console(config-pmap-c)# exit
Console(Config-pmap)# class FLOW2
Console(Config-pmap-c)# set ip precedence 5
Console(Config-pmap-c)# police 2000 1000 exceed-action set ip dscp 62
Console(config-pmap-c)# exit
Console(Config-pmap)# exit

Console(config)# interface ethernet 1/1
Console(Config-if)# service-policy input P1
Console(Config-if)# exit
Console(config)# interface ethernet 1/2
Console(Config-if)# service-policy input P1
Console(Config-if)# exit
```

➔ When sending Flow1 lower than 1Mbps:
Packets exiting OmniSwitch6300 have 802.1p 7 and ToS byte unchanged

➔ When sending Flow1 higher than 1Mbps:
Packets exiting OmniSwitch6300 have 802.1p 7 and ToS byte unchanged for traffic not exceeding 1Mbps
Packets exiting OmniSwitch6300 have 802.1p 7 and ToS byte dscp 63 for traffic exceeding 1Mbps

➔ When sending Flow2 lower than 2Mbps:
Packets exiting OmniSwitch6300 have 802.1p unchanged and ToS byte precedence 5

➔When sending Flow2 higher than 2Mbps:
Packets exiting OmniSwitch6300 have 802.1p unchanged and ToS byte precedence 5 for traffic not exceeding 2Mbps
Packets exiting OmniSwitch6300 have 802.1p unchanged and ToS byte dscp 62 for traffic exceeding 2Mbps

6.7 Egress “Service Policy”: 1 “Policy Map” + 1 “ip precedence” and 1 “ip dscp” “Class Map” with action set cos, set ip precedence

We want to: (on egress interface 1/3)

- set cos 7 for flow1 (packets with ToS 3)
- set ip precedence 5 for flow2 (packets with DSCP 12)

```
Console(config)# access-list ip mask-precedence out
Console(config-ip-mask-acl)# mask any any precedence
Console(config-ip-mask-acl)# mask any any dscp
Console(config-ip-mask-acl)#exit

Console(config)# class-map FLOW1 match-any
Console(Config-cmap)# match ip precedence 3
Console(config-cmap)# exit

Console(config)# class-map FLOW2 match-any
Console(Config-cmap)# match ip dscp 12
Console(config-cmap)# exit

Console(config)# policy-map P1
Console(Config-pmap)# class FLOW1
Console(Config-pmap-c)# set cos 7
Console(config-pmap-c)# exit
Console(Config-pmap)# class FLOW2
Console(Config-pmap-c)# set ip precedence 5
Console(config-pmap-c)# exit
Console(Config-pmap)# exit

Console(config)# interface ethernet 1/3
Console(Config-if)# service-policy output P1
Console(Config-if)# exit
```

Packets exiting OmniSwitch6300 have 802.1p 7 and ToS byte unchanged for Flow1
Packets exiting OmniSwitch6300 have 802.1p unchanged and ToS byte precedence 5 for Flow2