Allied Telesis™

AlliedWare Plus™ OS

How To   Configure QoS to prioritize SSH, Multicast, and VoIP Traffic

Introduction

This How To Note explains how to create a QoS policy that prioritizes SSH, multicast, and VoIP traffic in a congested network.

How you prioritize traffic within a QoS policy can differ based on the traffic type. This How To Note describes one approach to prioritizing three different traffic types within an example network. The methods used to classify and prioritize traffic is different for each traffic type. For the SSH and multicast traffic, hardware Access Control Lists (ACLs) are used to classify the traffic based on the protocol type and IP range. For the VoIP traffic, the DSCP value within the packet header, in conjunction with a hardware ACLs, is used to classify and prioritize the traffic.

This How To Note also explains how to order the class-maps within a QoS policy so that the correct traffic is prioritized, and how to monitor the network to confirm that the QoS policy is working as intended.

Contents

Which products and software versions does this How To Note apply to?

This configuration applies to x900 Series switches and Switchblade x908 switches running the AlliedWare Plus™ operation system, software version 5.2.2 and above.

Related How To Notes

You also may find the following How To Notes useful:

- *Overview of Quality of Service Features on x900-12, x900-24, and SwitchBlade x908 Switches*

- *How To Configure QoS on x900-24, x900-12, and SwitchBlade x908 Series Switches*

These Notes are available from www.alliedtelesis.com/resources/literature/howto.aspx.

Network scenario

In this example network, large volumes of traffic, up to 107Mbps from two data sources, is being sent downstream to two client PCs. However, a 10Mbps link is causing congestion between the data sources and the end users. A diagram of this example network is below.
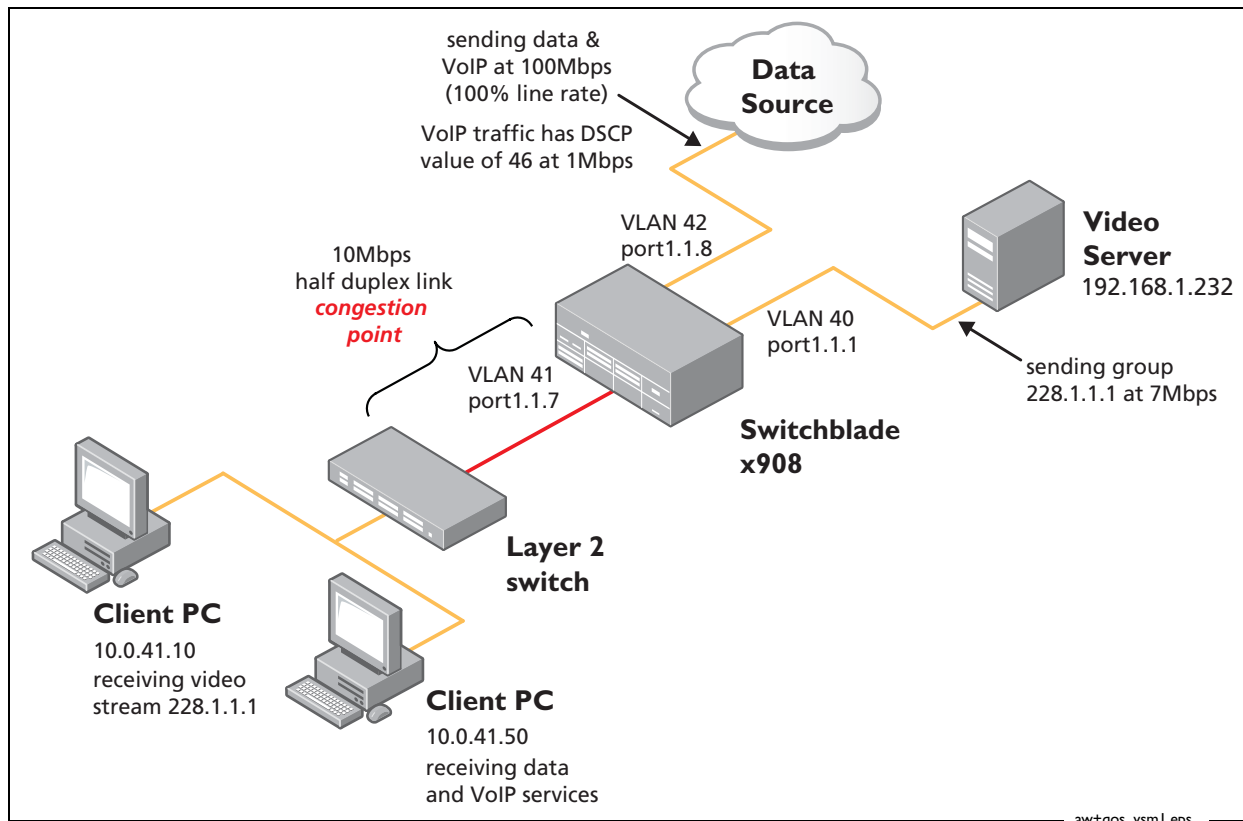
There are three VLANs in the network, each with their own IP subnet:

- VLAN 40 192.168.1.0/24

- VLAN 41 10.0.41.0/24

- VLAN 42 10.0.42.0/24

We want the network to prioritize SSH, multicast, and VoIP traffic, so that the downstream clients do not lose any of this traffic even if other traffic types are causing network congestion. An additional requirement is to also block unicast traffic between the two data source subnets (VLANs 40 and 42).

This network will use the QoS feature set on the Switchblade x908 to prioritize the different traffic types. We will create a QoS policy on the switch and apply it to the ports attached to the two data sources. The QoS policy will send:

- SSH traffic to the highest priority queue (queue 7)
  ACLs are used to classify this traffic based on its protocol.

- multicast traffic to the second highest priority queue (queue 6)
  ACLs are used to classify this traffic based on its IP range.

- VoIP traffic to the third highest priority queue (queue 5)
  ACLs are used to identify the traffic flow. To prioritizes the VoIP traffic within the traffic flow, we will use the DSCP value that is in the packet header of each VoIP packet.

- all other traffic to the default priority queue (queue 2)

The policy is applied only to the ports attached to the data sources. Traffic coming from the clients into the Switchblade x908 has already passed the congestion point and does not need to be prioritized.

We will also create ACLs on the Switchblade x908 to block all non-multicast traffic between VLANs 40 and 42. These will be applied to the ports that the VLANs are associated with.

Non-QoS configuration needed on the switch

The following non-QoS configuration is also required on the Switchblade x908:

VLAN configuration

To create VLANs 40, 41, and 42, use the commands:

```
vlan database
 vlan 40 name vlan40
 vlan 41 name vlan41
 vlan 42 name vlan42
 vlan 40-42 state enable
```

Add them to the required interfaces, for example for port1.1.1 use the commands:

```
interface port1.1.1
 switchport
 switchport mode access
 switchport access vlan 40
```

Then assign the IP subnet range to the VLAN, for example for VLAN 40 use the commands:

```
interface vlan40
 ip address 192.168.1.254/24
```

You will also need to assign the default VLAN (vlan1) an IP subnet:

```
interface vlan1
 ip address 192.168.35.254/24
```

multicast routing

You must enable multicast routing on the switch, so that it can route multicast traffic from the video server on VLAN 40 to the clients on VLAN 41:

```
ip multicast-routing
ip pim bsr-candidate vlan40
ip pim rp-candidate vlan40 priority 0
```

Configure PIM sparse mode on the interfaces that will route multicast traffic:

```
interface vlan40
 ip address 192.168.1.254/24
 ip igmp
 ip pim sparse-mode
!
interface vlan41
 ip address 10.0.41.254/24
 ip igmp
 ip pim sparse-mode
```

Using a DSCP value to prioritize traffic

The VoIP service on the network already uses DiffServ marking (DSCP) to identify that it is VoIP traffic. The DSCP value of the VoIP traffic is 46.

The example configuration in this How To Note shows how to prioritize VoIP traffic between two specific sources. It does this by matching the traffic flow and then assigning VoIP traffic in that flow to a specific egress queue. It is not prioritized by matching the DSCP value only, which you could do by using the **match dscp** command.

1. Map the DSCP value to an egress queue

Map any packets entering the switch with a DSCP value of 46 to queue 5 on the egress port, using the **mls qos map premark-dscp** command:

```
mls qos map premark-dscp 46 to new-queue 5
```

These packets must also match an ACL assigned to a class-map that has **trust dscp** configured.

You can also map the DSCP value with a CoS value. This is useful in scenarios where switches further within the network use the CoS value for their QoS policies. To add a new CoS value to the VoIP packets, use the command:

```
mls qos map premark-dscp 46 to new-cos 5
```

2. Create the ACL for the traffic flow

Next you need to create the ACL for the traffic flow from the VoIP service to the client PC. Use the command:

```
access-list 3003 permit ip 10.0.42.0/24 10.0.41.0/24
```

3. Create the class-map and add the ACL

Create a class-map for each traffic flow, and add an ACL to each:

```
class-map vp
  match access-group 3003
```

4. Configure the class-map to "trust" DSCP values

When you add the class-map to the policy-map, set the class-map to trust DSCP values:

```
policy-map policy1
class vp
  trust dscp
```

If a class-map does not have **trust dscp** command set, then the DSCP value in a packet is ignored when the class-map sets the egress queue. The change to the CoS value in the packets would also not occur. This means that any other VoIP packets that do not match this class-map will not have their egress queue or CoS value determined by the **mls qos map premark-dscp** command.

For all packets that match the class-map, but that have a different DSCP value (or no DSCP value), the policy will put the traffic into the default egress queue (queue 2).

Note that when traffic matches the ACLs in more than one class-map, the order that the class-maps are in the policy-map determines which ACL is used. See "Adding the class-maps to the policy" on page 8 for more information.

Configuring access control lists

In this example network, ACLs are used for:

- **Blocking unwanted traffic** between subnets–these ACLs are applied directly to an interface

- **Creating the classifiers for the QoS policy**–these ACLs are associated with a class-map then applied to an interface through the QoS policy-map

The AlliedWare Plus™ operating system provides two types of ACLs: hardware ACLs and software ACLs. This example uses hardware ACLs that match IP traffic (number range 3000 to 3699). Hardware ACLs will permit traffic access unless the traffic is explicitly denied by an ACL action.

Blocking unwanted traffic

One requirement in this network is for no unicast traffic to flow between the two data sources. To create the ACLs to stop IP unicast traffic between VLAN 40 and VLAN 42, use the commands:

```
access-list 3009 deny ip 192.168.1.0/24 10.0.42.0/24

access-list 3010 deny ip 10.0.42.0/24 192.168.1.0/24
```

The ACL number (3xxx) does not affect the order in which a packet is processed. ACLs are matched by the order that they are added to the interface. When a packet arrives on an interface, it is tested against the first ACL on the interface, and if a match is found then the packet is processed based on the action (permit or deny) configured for this ACL. If a match is not found against the first ACL, the switch then tests against the next ACL on the interface. See "ACL and QoS policy processing order" on page 8 for more information.

Creating the classifiers for the QoS policy

The QoS policy for this network requires classifiers to match SSH, multicast, and VoIP traffic. For SSH traffic, the classifiers match the protocol type. For multicast traffic, the classifiers match the multicast IP range. For VoIP traffic, the classifiers match the traffic flow only, and the DSCP value in each packet determines the egress queue.

1. Create the ACLs

Create ACL 3001 to classify SSH traffic (TCP traffic with a port number of 22). This classifies all SSH packets with any source and destination IP address:

```
access-list 3001 permit tcp any any eq 22
```

Create ACL 3002 to classify traffic from host 192.168.1.232 (the video server) to any multicast group addresses in 228.x.x.x range:

```
access-list 3002 permit ip 192.168.1.232/32 228.0.0.0/8
```

Create ACL 3003 to classify any traffic flowing from the VoIP server subnet to the client subnet:

```
access-list 3003 permit ip 10.0.42.0/24 10.0.41.0/24
```

2. Add the ACLs to the class-maps

The access-lists are then applied to a class-map.

```
class-map ssh
 match access-group 3001
 !
```

```
class-map mcast
 match access-group 3002
!
class-map vp
 match access-group 3003
```

ACL and QoS policy processing order

If interface ACLs and a QoS policy are both applied to a given interface, then the ACLs are always processed before the QoS policy. If a packet matches any ACL on the interface, it will not be processed by QoS. For example, if you had the following configuration:

```
interface portx.x.x
  service-policy input PolicyName  <== applies a QoS policy
  ip access-group 3021             <== ACL on interface
  ip access-group 3042             <== ACL on interface
```

Then the switch checks whether traffic matches ACL 3021 first, then ACL 3042, then the ACLs in the QoS policy. It is processed by the first matching ACL.

Adding the class-maps to the policy

Class-maps are added to an interface through a policy. The order that you add the class-maps to the policy-map is important as packets are processed by the first class-map with a matching ACL. That means that you must add the class-maps in the order that you want to check for matching packets.

Before creating a policy, decide which traffic prioritization is most important, so that you can determine the order that you will add the class-maps to the policy-map. Take into consideration the priority you want for certain traffic and whether the ACLs matching that traffic will collect other traffic as well. In this example network, the class-maps are added based on the egress queue priority for the traffic.

The priority of traffic is set using either pre-defined mappings (such as for the VoIP traffic - see "Using a DSCP value to prioritize traffic" on page 5) or within the policy-map itself. In this network, the priority of the SSH and multicast traffic is set within the policy-map.

1. Create the policy-map

Use the command:

```
policy-map policy1
```

2. Add the class-map for SSH traffic

Class ssh matches access-list 3001 for SSH traffic. The **set queue 7** command places this traffic in queue 7 on the egress port (the highest priority queue).

```
class ssh
   set queue 7
```

3. Add the class-map for multicast traffic

Class mcast will remark the multicast traffic (matching ACL 3002) to a CoS value of 6 and a DSCP value of 60. These values may be used to prioritize this traffic on another switch further in the network. This multicast traffic will be placed in queue 6 on the egress port.

```
class mcast
   set cos 6
   set dscp 60
   set queue 6
```

4. Add the class-map for VoIP traffic

Class vp matches ACL 3003 and instructs the switch to trust the incoming DSCP value of this traffic. Traffic matching this class is processed according to any configured **premark-dscp map** commands. In this example, we have set this to send VoIP traffic to queue 5 and also give it a CoS value of 5. The class-map will send all other matching traffic that is not VoIP traffic to queue 2 and will not alter the CoS value for that traffic.

```
class vp
   trust dscp
```

The default class-map

A class called **default** is created whenever a QoS policy is configured. It is added to the end of the policy. All traffic that has not matched a previous class-map is processed by the default class-map.
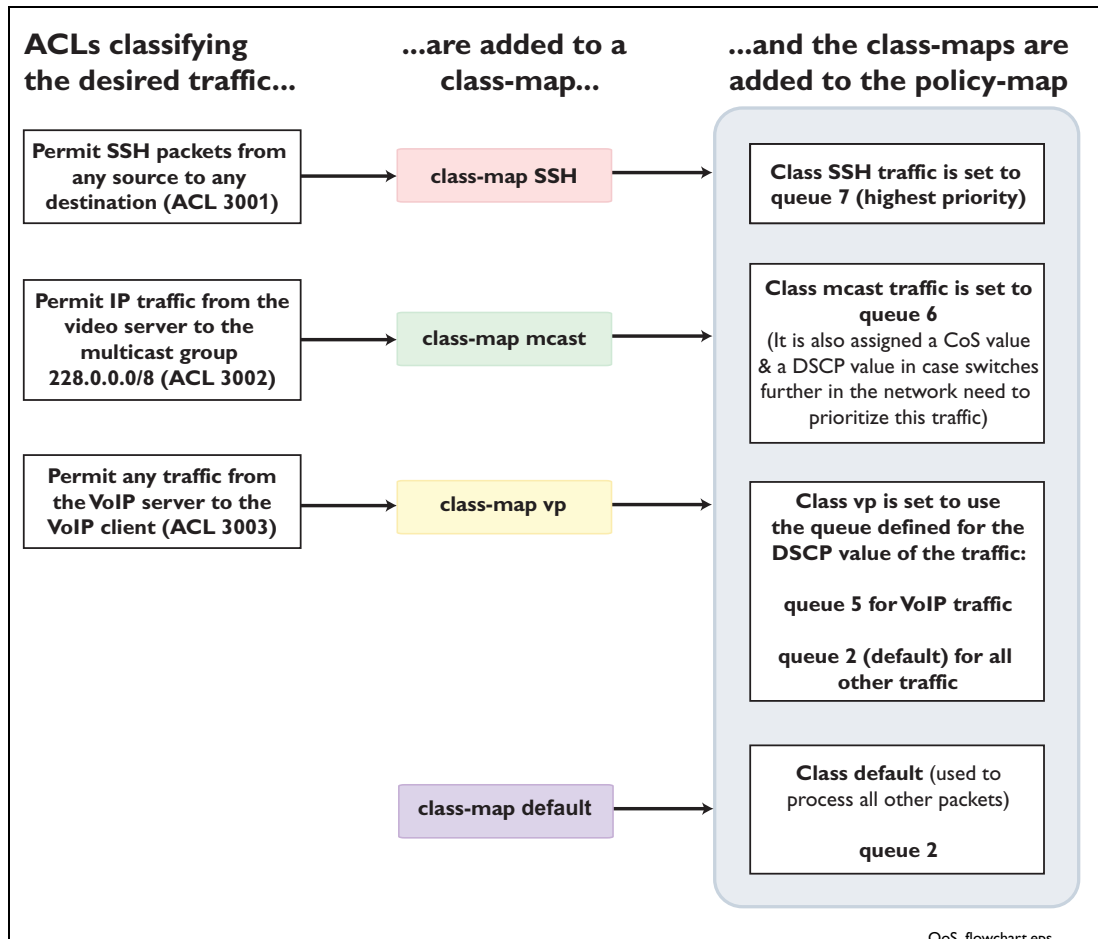
The default class-map is configurable within the policy-map. Use the command:

```
class default
```

Its default settings is to send all traffic it processes to queue 2.

Relationship between the ACLs and the policy

The following figure shows the relationship between the ACLs configured to create the classifiers for the class-maps, and the policy-map.

ACLs classifying the desired traffic...

Permit SSH packets from any source to any destination (ACL 3001)

Permit IP traffic from the video server to the multicast group 228.0.0.0/8 (ACL 3002)

Permit any traffic from the VoIP server to the VoIP client (ACL 3003)

...are added to a class-map...

class-map SSH

class-map mcast

class-map vp

class-map default

...and the class-maps are added to the policy-map

Class SSH traffic is set to queue 7 (highest priority)

Class mcast traffic is set to queue 6
(It is also assigned a CoS value & a DSCP value in case switches further in the network need to prioritize this traffic)

Class vp is set to use the queue defined for the DSCP value of the traffic:

queue 5 for VoIP traffic

queue 2 (default) for all other traffic

Class default (used to process all other packets)

queue 2

QoS_flowchart.eps

Applying the QoS policy to the ports

The QoS policy is applied to the ports where traffic is received that could match the classifiers (ACLs). When attached to a port, the policy checks ingress traffic on that port and determines the priority the traffic will have at the egress port. This prioritisation occurs regardless of whether the egress port has the QoS policy associated with it.

In this example we are applying policy-map policy1 to the ports attached to the two data sources (ports 1.1.1 and 1.1.8). This prioritizes the SSH, multicast, and VoIP traffic arriving from the data services. Use the **service-policy input** command to add the policy to the interfaces:

```
interface port1.1.1
 switchport
 switchport mode access
 switchport access vlan 40
 service-policy input policy1
 ip access-group 3009

interface port1.1.8
 switchport
 switchport mode access
 switchport access vlan 42
 service-policy input policy1
 ip access-group 3010
```

The ACLs 3009 and 3010 are also added to the interfaces. These deny unicast traffic between the two data sources (which are the only members within their VLANs).

Understanding how packets are matched

To configure a working QoS policy it is vital that you understand how the switch determines the correct ACL to use for the packet, as a packet could match more than one ACL.

The class-maps are configured in this order in the policy-map:

1. class ssh

2. class mcast

3. class vp

4. class default

In this network example, an SSH packet can match both ACL 3001 and 3003 if the SSH packet is coming travelling from 10.0.42.0/24 to 10.0.41.0/24. Which ACL the QoS policy uses depends on the order of the class-maps.

The first configured class in the policy-map is class ssh:

```
class ssh
   set queue 7
```

When the QoS policy processes a packet, it first checks class-map ssh to see whether the class-map's ACL matches the packet:

```
class-map ssh
 match access-group 3001
```

If the packet matches the ACL, as it does in this case, then the packet will use class ssh and be sent to queue 7 on the egress port:

```
class ssh
   set queue 7
```

However, if the first class had been class vp:

```
class vp
   trust dscp
```

Then the QoS policy would have looked at the ACL for class-map vp first:

```
class-map vp
 match access-group 3003
```

So the packet is tested against ACL 3003 for the VoIP traffic. If the packet matches, as it does in this case, then the packet will use class vp and the switch is instructed to trust the DSCP value:

```
class vp
   trust dscp
```

So, if the class vp had been applied to the policy-map higher than class ssh, all SSH traffic between the two devices would have been queued to the default traffic queue of 2.

In this network, the QoS policy is applied to ports 1.1.1 and 1.1.8. There are is also an ACL on each port to block traffic between the data sources. The following flow chart shows how a packet is processed when it arrives at port 1.1.1 or 1.1.8.

packet_flowchart.eps
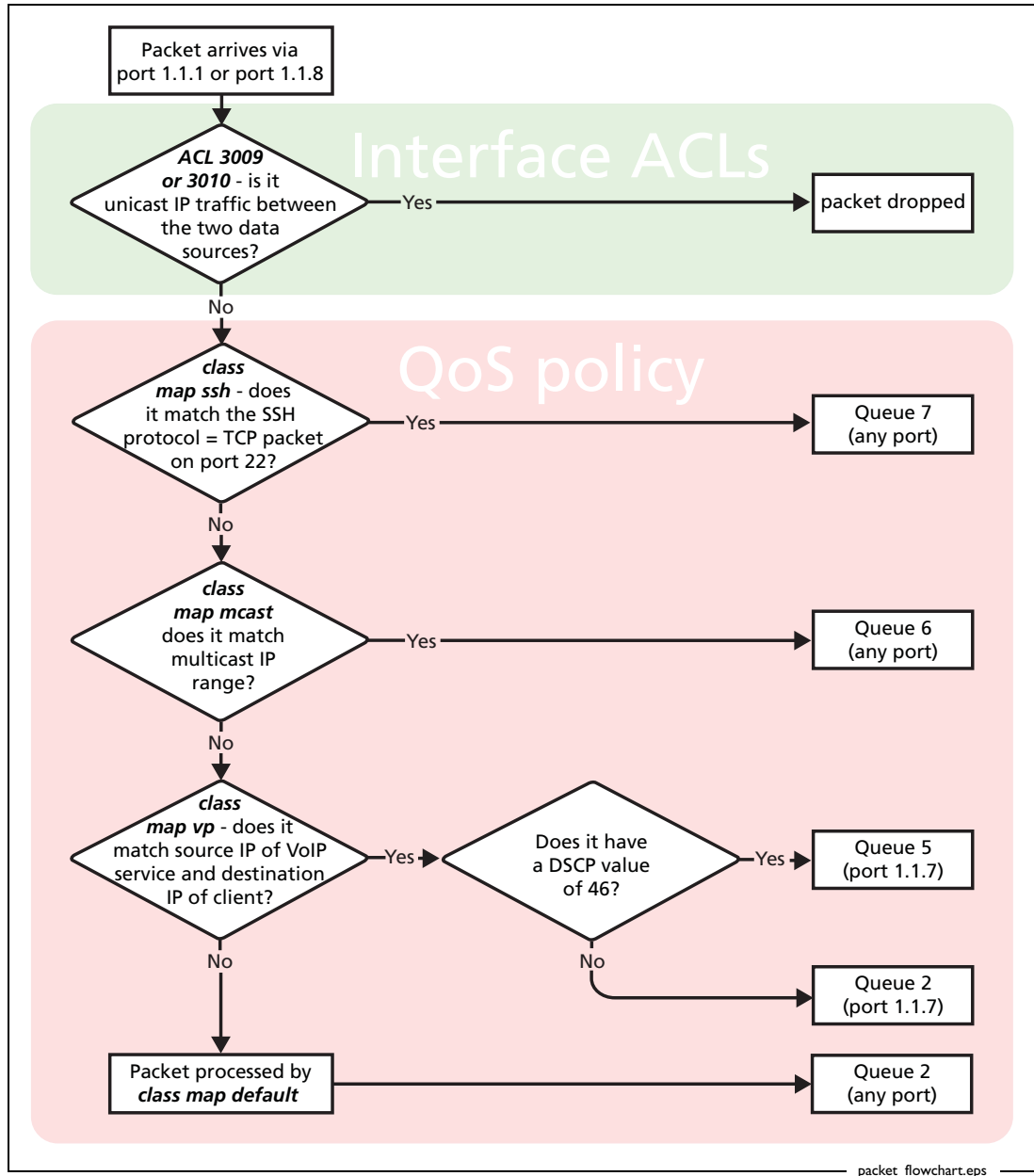
Monitoring the network

The switch provides a variety of commands to monitor its operation, for example to check that traffic is flowing correctly, IGMP reports are being received, and that unexpected congestion is not occurring.

▶ Checking traffic is being transmitted or received on a port

Use the **show platform table port count** command to see if traffic is being transmitted and received on a port. This command has the same output as the AlliedWare **show switch port count** command.

```
 Switch Port Counters
 --------------------------------------------------------------------------

Port 1.1.1    Ethernet MAC counters:
 Combined receive/transmit packets by size (octets) counters:
  64                          0 512 - 1023                      0
  65 - 127                    0 1024 - MaxPktSz                 0
  128 - 255                   0
  256 - 511                   0

 General Counters:
 Receive                        Transmit
  Octets                      0 Octets                          0
  Pkts                        0 Pkts                            0
  CRCErrors                   0
  MulticastPkts               0 MulticastPkts                   0
  BroadcastPkts               0 BroadcastPkts                   0
  FlowCtrlFrms                0 FlowCtrlFrms                    0
  OversizePkts                0
  Fragments                   0
  Jabbers                     0
  UpsupportOpcode             0
  UndersizePkts               0
                                Collisions                      0
                                LateCollisions                  0
                                ExcessivCollsns                 0

 Miscellaneous Counters:
  MAC TxErr                   0
  MAC RxErr                   0
  Drop Events                 0
```

▶ Checking for congestion on an interface's egress queues

Use the **show mls qos interface <egress port> queuecounters** command to look for congestion on an egress port. If any of the 8 queues on the egress port overflow then the Queue length counter will be non-zero. This is useful to see if congestion is causing a port queue to discard packets and which queue is doing this. In this QoS demonstration we would expect to see queue 2 (the default queue) overflow as higher priority packets (in higher priority queues) are serviced first.

```
AW+#show mls qos interface port1.1.8 queue-counters

Interface port1.1.8 Queue Counters:
  Port queue length         0 (maximum 896)
  Egress Queue length:
    Queue 0                 0 (maximum 112)
    Queue 1                 0 (maximum 112)
    Queue 2                 0 (maximum 112)
    Queue 3                 0 (maximum 112)
    Queue 4                 0 (maximum 112)
    Queue 5                 0 (maximum 112)
    Queue 6                 0 (maximum 112)
    Queue 7                 0 (maximum 112)
```

Note that the **show mls qos interface <egress port> queuecounters** command only works if you have configured the switch to store QoS counters. To enable this feature, use the **platform enhancedmode qoscounters** command. You will need to restart the switch before the feature will take effect.

▶ Checking the multicast traffic

Use the **show platform table l2mc** command to look at multicast groups on the switch. We can see that the entry for our multicast stream 228.1.1.1, which maps to a Layer 2 MAC of 01:00:5e:01:01:01 and has a Local Port of 1.1.7. This is the port connected to the Layer 2 switch.

```
[Instance 1.0]
L2 Multicast Group and VLAN Broadcast table:
-------------------------------------------------------------------------
        Total number of entries = 6
-------------------------------------------------------------------------
Index  MAC                 VID  MCGroup CPU_MEM NumPorts
       PORT_LIST
-------------------------------------------------------------------------
0      01:00:5e:01:01:01 40   4099    0       0
       Local Ports = None

1      01:00:5e:01:01:01 41   4102    0       1
       Local Ports = 1.1.7

2      01:00:5e:7f:ff:fa 40   4098    0       1
       Local Ports = 1.1.1

3      01:00:5e:00:01:23 40   4101    0       0
       Local Ports = None

4      01:00:5e:7f:ff:fd 40   4100    0       0
       Local Ports = None

5      01:00:5e:00:01:16 40   4097    0       1
       Local Ports = 1.1.1
```

▶ Checking for multicast groups

Use the **show ip igmp group** command to look at multicast group memberships on the switch. We can see that the multicast group 228.1.1.1 has received an IGMP join from port 1.1.7, which is connected to the layer 2 switch.

```
IGMP Connected Group Membership
Group Address    Interface            Uptime   Expires  Last Reporter
224.0.1.22       port1.1.1            00:02:36 00:02:13 192.168.1.232
228.1.1.1        port1.1.7            00:02:13 00:03:22 10.0.41.50
```

▶ Verifying traffic is going into the correct egress queue

One way to prove that traffic is being sent to an egress queue is to temporarily disable the queue and check that traffic sent to that queue is not reaching its destination. Use the commands:

```
Configure terminal
interface <egress port>
wrr-queue disable queues [0] [1] [2] [3] [4] [5] [6] [7]
```

For example, to check VoIP traffic is being sent to queue 5:

1. Establish a call between the client and the server.

2. Disable the queue that the traffic is sent to:
```
interface port1.1.7
wrr-queue disable queues 5
```

3. Check that the call has disconnected, and that you cannot reestablish it.

4. Enable the queue:
```
interface port1.1.7
no wrr-queue disable queues 5
```

5. Establish a new call to confirm the queue has been re-enabled.

To prove that multicast traffic is being sent to queue 6, follow these steps:

1. View the video stream on a client PC.

2. Disable the queue that the traffic is sent to:
```
interface port1.1.7
wrr-queue disable queues 6
```

3. Check that the video stream has been disrupted.

4. Enable the queue:
```
interface port1.1.7
no wrr-queue disable queues 6
```

5. Confirm that the client PC can receive the video stream again.

Complete Switchblade x908 configuration script

```
ip multicast-routing
!
no platform e2efc
!
mls qos enable
mls qos map premark-dscp 46 to new-cos 5
mls qos map premark-dscp 46 to new-queue 5

access-list 3001 permit tcp any any eq 22
access-list 3002 permit ip 192.168.1.232/32 228.0.0.0/8
access-list 3003 permit ip 10.0.42.0/24 10.0.41.0/24
access-list 3009 deny ip 192.168.1.0/24 10.0.42.0/24
access-list 3010 deny ip 10.0.42.0/24 192.168.1.0/24
!
class-map ssh
 match access-group 3001
!
class-map mcast
 match access-group 3002
!
class-map vp
 match access-group 3003
!
policy-map policy1
 class ssh
  set queue 7
 class mcast
  set cos 6
  set dscp 60
  set queue 6
 class vp
  trust dscp
 class default
!
vlan database
 vlan 40 name vlan40
 vlan 41 name vlan41
 vlan 42 name vlan42
 vlan 40-42 state enable
!
interface port1.1.1
 switchport
 switchport mode access
 switchport access vlan 40
 service-policy input policy1
 ip access-group 3009
!
interface port1.1.7
 switchport
 switchport mode access
 switchport access vlan 41
!
interface port1.1.8
 switchport
 switchport mode access
 switchport access vlan 42
 service-policy input policy1
 ip access-group 3010
!
```

```
interface vlan1
 ip address 192.168.35.254/24
!
interface vlan40
 ip address 192.168.1.254/24
 ip igmp
ip pim sparse-mode
!
interface vlan41
 ip address 10.0.41.254/24
 ip igmp
ip pim sparse-mode
!
interface vlan42
 ip address 10.0.42.254/24
!
end
```

C613-16141-00 REV A-e

Connecting The IP World

Allied Telesis