1. Introduction

1.1 Purpose

There are many documents available today describing the QoS functionality of the Cisco Systems Catalyst 6500. This document is not intended to replace any of these documents only to complement them. It is recommended that the reader review all available White Papers and application notes if they wish to gain complete understanding of the features available.

1.2 Acknowledgements

In the quest to validate the findings of this Application Note it was necessary to obtain the assistance of a few people in and outside Cisco. I would especially like to thank Chris Paggen of the Technical Marketing team, as he corrected some of my earlier mistakes and took the time to prove the theories in the Lab when I was unable.

2. Policing Function

This document describes the operation of Catalyst 6500 bandwidth policing. Detailed descriptions of the other QoS features are covered by the following URL's:

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/6000hw/hw_inst/01over.htm

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/sw_5_3/cofigide/qos.htm

http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/ios127xe/qos.htm

2.1 Components of the 6500 Policing

The Catalyst 6500 currently only polices data ***inbound*** into a switch port. This is a function of the Policy Feature Card on the Supervisor Engine. If there is a need to police traffic inbound and outbound please refer to Appendix A.

Traffic policing on the 6500 employs a 'token bucket' type system. The token bucket policing system consists of three elements:

i. Replenishment rate (R)
ii. Token bucket size (B)
iii. Fixed time interval (T)

The basic operation of the token bucket system is to impose a rate on an incoming data stream while, by use of the token bucket, allowing the configured rate to exceed the value for a determinable length of time.

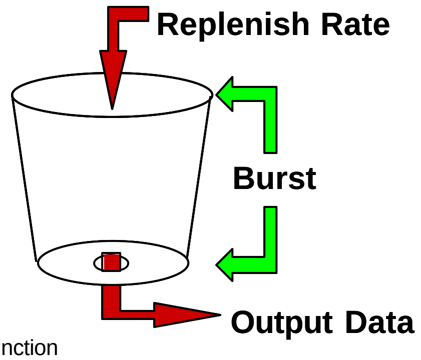

Figure 1. Policing function

The replenishment rate is the speed at which tokens are added to the token bucket, it correlates directly to the speed at which the port input rate is restricted, the value is expressed in bits per second. The token bucket size is a fixed value, it is the amount of data or tokens that can be stored during the measured time interval and it is expressed as a finite amount of data bits.

If a policed port input rate value exceeds the replenishment rate value, and there is no capacity in the token bucket, the excess data is classed out of profile and can be dealt with in one of two ways:

i. Reassign Class of Service.
ii. Drop packet.

Class of service reassignment is dealt with extensively in documents referenced earlier. From now on this document will assume that the default exceed action of the policer is to drop the packet.

2.2 Operational Description of the Policing Function

The policing function of the 6500 controls the data after it has completely entered the port, and is in the packet buffer. Thus all counters and statistics displayed by the CLI are measured before the policing function.

When the data arrives into the switch port, or via a VLAN on a switch trunk port it is compared against the QoS ACL entries configured. If the data stream matches an ACL entry the policer allocates one token for every data bit in the payload of the IP and IPX packets entering the switch. If the data is non IP or IPX then the whole Layer two frame is measured, including CRC.

When the data is passed to the policer it removes an amount of tokens from the total tokens in the bucket equal to the amount of tokens allocated for the incoming data. If there were enough tokens in the bucket the packet continues through the switching process, if not the packet is either dropped or reclassified (then forwarded on into the system). The token bucket is replenished at a rate equal to the configured rate on the CLI.

This process happens once every time interval of 0.00025 seconds, thus the tokens are added to the bucket at a rate 1/4000th of the configured value. The configured burst value (or bucket depth) is different; it is valid at each 0.00025-second time interval or 4000 times a second.

When QoS policing is configured and the process initially starts the token bucket contains the *full amount of burst bits (or tokens)* allocated to the policer ACL, tokens are then *removed* from the bucket every time interval. The amount of tokens removed from the bucket at each interval is either the maximum for the port media (as indicated below) or the amount of tokens equal to the next packet in the input buffer of the port that matches the policer ACL. (see figure 2.)

The maximum amount of tokens that can be removed for each media is *line rate / 4000*:

Ethernet –	2500 bits
Fast Ethernet –	25000 bits
Gigabit Ethernet –	250000 bits

The minimum number of tokens in the bucket must be at least equal to the bits in the incoming Ethernet frame in order for the packet to continue. Even if the incoming packet is one token larger the number available in the bucket, the packet is dropped and the tokens remain in the bucket until the next 0.00025-second cycle.

After all the tokens in the bucket have been consumed, tokens are only replenished at the configured rate. If any tokens remain in the bucket after one time interval they remain until the next. If this occurs it allows the input rate to exceed the configure rate for the duration of the remaining tokens.

So, if the actual data arrival rate equals or exceeds the configured replenish rate the token bucket remains empty and the port output rate will equal the configured replenish rate. If the input rate falls below that of the refresh rate the bucket will begin to fill.

The following is a flow chart depicting the whole policing process.

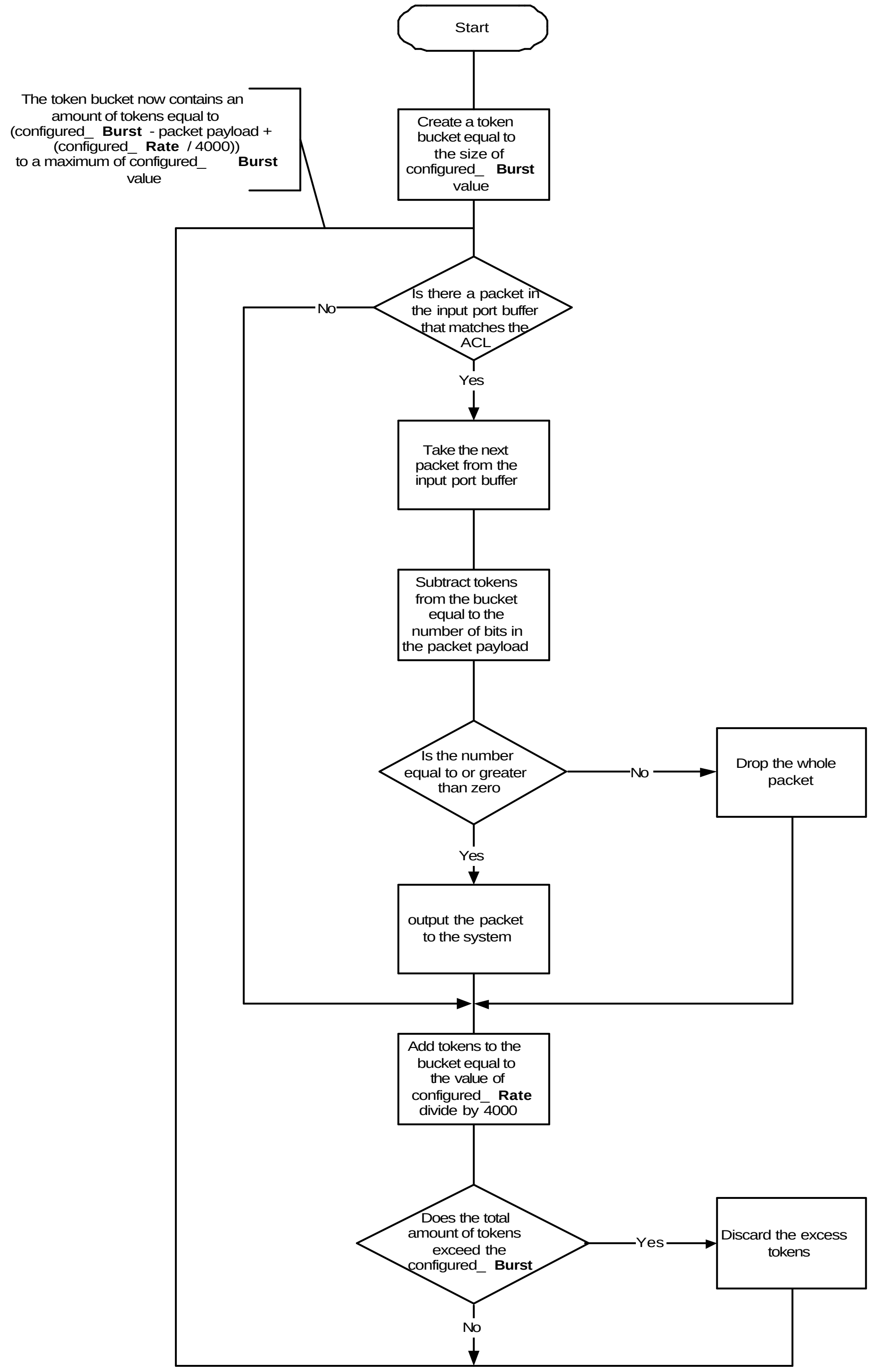
Start
The token bucket now contains an amount of tokens equal to (configured_ **Burst** - packet payload + (configured_ **Rate** / 4000)) to a maximum of configured_ **Burst** value
Create a token bucket equal to the size of configured_ **Burst** value
Is there a packet in the input port buffer that matches the ACL
No
Yes
Take the next packet from the input port buffer
Subtract tokens from the bucket equal to the number of bits in the packet payload
Is the number equal to or greater than zero
No
Drop the whole packet
Yes
output the packet to the system
Add tokens to the bucket equal to the value of configured_ **Rate** divide by 4000
Does the total amount of tokens exceed the configured_ **Burst**
Yes
Discard the excess tokens
No

Figure 2. Policing description Flow Chart

2.3 Policing Example

The following is an example configuration entry allowing only 8,000,000 bits per second to pass through a switch port. Although a burst above the Allowed rate is not required the minimum value of burst that is required by the CLI is 1.

set qos policer aggregate 8megonly rate 8000 burst 1 drop

The parameters of interest here are ***rate 8000*** and ***burst 1***, the rest of the QoS command line entries are explained in the documents indicated earlier in this section. The ***rate 8000*** parameter specifies that the allowed rate is 8,000,000 bits as the unit of rate is 1,000 and the ***burst 1*** parameter specifies that the burst value is 1,000 bits as again the unit of burst is 1,000.

Note: Although the above example is theoretically correct in allowing only 8,000,000 bits per second to pass through a switch port without a burst above the configured rate, in reality this will return unexpected results. To successfully implement policing using the 6500 in 'real world' environments additional thought is required when formulating the policing command.

Example:

If we assume the above policer ACL is applied to a 100Mbs port we then know that the maximum input rate of the port we can expect is 25,000 bits every time interval (100,000,000 / 4,000) if the port is running at full capacity.

So, at time interval T=0 there are 1,000 tokens available in the bucket and 25,000 tokens input to the port, this means that 24,000 tokens will be dropped. At time interval T=1 the Allowed rate tries to place 2,000 tokens in the bucket (8,000,000 / 4,000) but it can only accept 1,000 (size of the token bucket), thus the port only outputs another 1,000 bits and 24,000 are dropped again.

This process equates to an output rate from the port of 1,000 bits x 4,000 time intervals, or 4Mbs.

So from this simple example we can see that although we do not wish to have a burst (thus we assume we should use the minimum of 1), we will still need to increase the size of the token bucket to accommodate the amount of tokens needed to sustain the required output rate.

So the correct configuration to allow a rate of 8Mbs without a burst above the required rate will be :

set qos policer aggregate 8megonly rate 8000 burst 2 drop

When configuring a QoS policer entry calculate the minimum burst rate by dividing the required-policed rate as entered via the CLI by 4,000.

2.4 Practical Deployment of Traffic Policing

From previous discussions in this paper, it can be seen that the most important element that affects how the policer behaves is the payload size of the packet entering the port.

As mentioned earlier, an IP packet is only forwarded if tokens equal in size to the payload are available in the token bucket. Thus if all the packets received by the port are 1518 bytes, the amount of tokens that must be available to allow at least one packet to be forwarded is 12,000 (1,500 x 8).

The easiest way to portray this is with an example:

A Web Server is connected to a 6500 switch port by 100Mbs Ethernet. It is a popular site and when not policed the inbound data rate is at least 50Mbs. The Service Provider wishes to restrict the rate to 10Mbs as this is the bandwidth paid for by the owner of the Web Site.

From earlier discussion we know that the QoS policer configuration for this port should be:

```
set qos policer aggregate 10megonly rate 10000 burst 3 drop
```

However, once we add the QoS configuration to the port the data throughput suddenly drops to a value much lower than we require. The reason for this is that the majority of the incoming packets are between 750 and 1518 bytes so, when an incoming packet of size 1000 bytes (8000 bits) payload tries to remove that amount of tokens there are never enough.

If we wish to allow any packet size to be forwarded, and we wish to restrict the rate to 10Mbs, we need to increase the *Burst* value (or token bucket size) to at least 12,000 (the maximum payload size of a 1518 byte Ethernet frame) while leaving the replenish rate at 10Mbs. By using this rule we can improve the accuracy of the policer.

Another way this can be demonstrated is by calculating how many replenish cycles are required before a frame can be forwarded. I.e. if the token bucket starts full with 12,000 bits and all the packets into the policed port are 1518 bytes the cycle will operate:

@ T=0 the bucket contains 12,000 tokens, the port needs to remove 12,000, there are enough tokens so the packet is forwarded.

@ T=1 2,500 tokens are added to the bucket, the port needs to remove 12,000, there are not enough tokens so the packet is dropped.

@ T= 2 another 2,500 tokens are added so the bucket now contains 5,000, the port needs to remove 12,000, there are not enough tokens so the packet is dropped.

@ T=3 another 2,500 tokens are added so the bucket now contains 7,500, the port needs to remove 12,000, there are not enough tokens so the packet is dropped.

@ T=4 another 2,500 tokens are added so the bucket now contains 10,000, the port needs to remove 12,000, there are not enough tokens so the packet is dropped.

@ T=5 another 2,500 tokens are added so the bucket now contains 12,000 (the excess are discarded), the port needs to remove 12,000, there are enough tokens so the packet is forwarded.

It takes 5 cycles for the bucket to get fully replenished, so if the port continues to receive 1518 byte frames the policer will forward one frame every 5 or 800 per second (4000 / 5) or at 9.6Mbs. If the packet size were to decrease this would yield a rate even closer to the required.

So from here we can assume that when policing ports, we need to configure the Burst value to accommodate the largest expected frame.

It is recommended that the minimum Burst value configured for each media speed is:

10Mbs Ethernet –	Burst 13
100Mbs Ethernet –	Burst 13
1000Mbs Ethernet –	*Allowed rate / 4000* (See below)

Initial thoughts were that a 10Mbs interface couldn't pass anymore than 2,500 bits within the 0.00025-second time interval (10,000,000 / 4000). But, a switch port will not forward a packet for policing until the whole packet has entered the port, so the amount of tokens that must be available to accommodate a frame on a 10Mbs Ethernet port is 12,000.

We have demonstrated earlier the reasons for a Burst of 12 on a 100Mbs port.

When working with a Gigabit Ethernet port the burst parameter takes another turn. As the arrival rate of a GigE interface can be up to 1000,000,000 bits per second, or 250,000 bits per 0.00025-second interval, we can see that more than one maximum Ethernet frame can be passed to the policer within one time interval. So, to calculate a Burst value we can use a theory suggested earlier, *divide the required-policed rate as entered via the CLI by 4,000.*

Example: A group of high performance Web Servers share one Gigabit Ethernet link into a Service Providers network, if allowed the Web Servers could generate an average of 800Mbs of data. The Service Provider wishes to restrict the usage to 150Mbs. Using the earlier rule the configuration entry would be (Burst is calculated by Rate / 4000):

```
set qos policer aggregate 150megonly rate 150000 burst 38 drop
```

We can see the original input rate of 800Mbs will be reduced to 150Mbs, but these figures are not our main concern. The real figure that will allow the port to operate at the configured rate is the Burst. The original input rate was forwarding on average 16 full payloads of IP data every 0.00025-second interval and in order for the required rate to be sustainable the policer must pass through 3 full size IP payloads per time interval (36,000 x 4,000 = 14,400,000 bits).

So the effect of having maximum payload sizes coming into the port, is to reduce the instantaneous rate to 144.4Mbs. But practically there will be a mixture of frame sizes resulting in a policed rate closer to the required configured rate. The only time the exact configured rate would be achieved is if the incoming frames fit into the configured burst, i.e. in the above example if the incoming packets are 1171.9 bytes in size then four would fit into the burst and the output rate would equal 150Mbs (9375 bits x 4 packets in buffer x 4000 per second). So, practically the actual rate may never match the configured rate.

2.5 Allowing the Input Port to Burst above the Configured Rate

Up to this point we have only discussed limiting the input rate of a port by adjusting the burst value to allow the configured rate to be sustained. There may be a requirement to allow the input rate to exceed the configured rate for a prolonged period of time.

As an example, let's assume that we have a Web Server attached at 10Mbs and it has a configured rate of 4Mbs. If there is a 'flash crowd' situation where that Web Server suddenly becomes very popular and the owner of the server wants to reduce the amount of '*Unable To Establish Connection*' messages returned, the Service Provider can configure the Burst parameter to sustain an increased rate for a predetermined period of time.

When calculating the size of the Burst parameter this formula can be used:

$$\frac{B}{In} + \frac{BC}{In^2} = T$$

Where: **B** = Burst Parameter, **In** = Actual Data Input Rate, **C** = Configured Data Rate and **T** = Time the increased rate can be sustained for.

So, using the above example, if the Web Server were to increase it's actual data rate from 4Mbs to 8Mbs and the configured Burst parameter was 20,000,000 then the time the Server will be allowed the increase it's rate for is 3.75 seconds.

Note. There is a caveat associated with this process. If the attached server exceeds the configured rate and thus consumes the Burst tokens for the port, it will be unable to burst again until the actual input rate falls below the configured rate. Also, as the time allowed to burst is calculated using a constant figure for **In**, if that value were to deviate then the time calculated would vary.

Appendix A.

As indicated earlier in this document it is currently only possible to police data rates Inbound to a Catalyst 6500. If bi-directional policing is required it is possible to configure the connection to allow bi-directional policing either per Port or per VLAN.

This is possible due to the operation of the 6500 and it's MAC address handling. For the reader of this paper to comprehend the proposals of this document they must understand the Layer 2 address resolution function of the 6500. It is also useful to remember that IP subnets are not related to VLANs, i.e. it is possible for the same subnet to exist on different VLANs if they are connected.

For all the following examples it is assumed that the 6500 contains a MSFC routing engine. This is not a requirement for bi-directional policing, it is equally possible to police in both directions when connecting to an external router.

A.1 Per Port Bi-Directional Policing

For the Network Administrator, configuring bi-directional policing requires extra ports on the Catalyst 6500. Bi-directional policing requires that Input policing is configured on two ports but applied in 'different directions'. With a crossover cable connected between the two ports on the 6500 and two VLANs to be created, one port is associated with the input from the Network and one is associated with the input from the server.

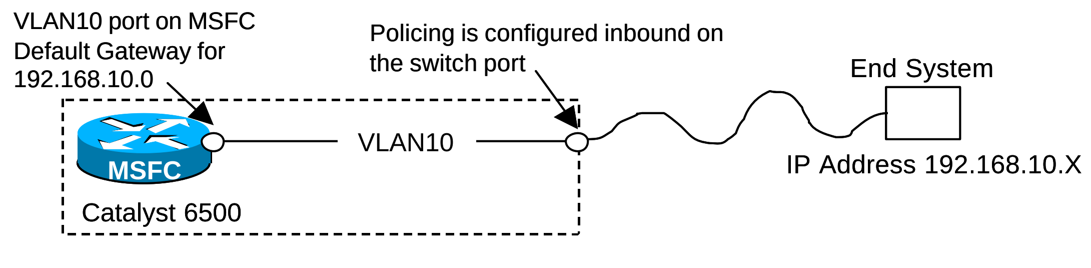

Figure 3. Regular Policing Configuration

The above figure shows how Policing is applied to a port on a Catalyst 6500, the Layer 3 port on the MSFC is in the same logical VLAN as the Server (End System). In order for the MSFC to establish an entry for the IP address of the Server it goes through the standard ARP process, this also populates the Layer 2 CAM table of the switch, associating the MAC address of the Server with the switch port and the MSFC MAC with the virtual port ID.

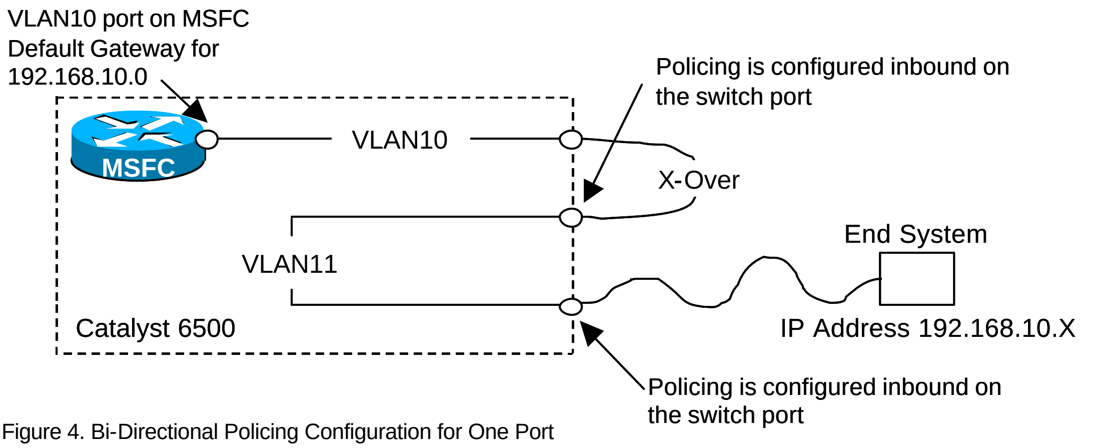

Figure 4. Bi-Directional Policing Configuration for One Port

Figure 4. shows how bi-directional Policing is achieved. As in regular policing the MSFC is configured with a VLAN interface, but the difference is that the Server is configured in a *different* VLAN. The two VLANs are then connected together (by cross-over cable) and input Policing is applied to the ports inbound to the switch *from* the Server and inbound to the switch *from* the MSFC. So, although the server is in a different VLAN to its default gateway it still has access to it.

In this configuration the ARP process and thus the data forwarding process works in a slightly different way to the original. For the Server to find the MAC address of the MSFC (or the other way round) the Layer 2 broadcast packet is sent to the switch and because the switch doesn't have a CAM entry for the MSFC it forwards the broadcast frame out of every port (other than the one it received it on, which is standard operation for the switch) thus it is sent out of the port that connects the two VLANs together. The switch then forwards the broadcast to the MSFC virtual port thus completing the Layer 2 address resolution process, with the CAM table showing the Server and MSFC being attached to two ports. This may at first appear strange, but because they are in two different VLANs it does not cause the switch any problems with packet forwarding.

Now that connectivity is made you can apply inbound policing to both ports, but in different directions and using different aggregates or micro flow lists. This also obviously allows different asymmetrical rates to be configured.

IMPORTANT Note. Whenever connecting a crossover cable between two ports on the same switch please be aware that unless the two ports are in separate VLANs it can cause a 'loop' in the network and severely effect network operation. It is advised that extensive testing is undertaken and a full understanding of what is to be achieved before applying this design to a live network.

A.2 Per VLAN Bi-Directional Policing

If there is more than one Server connected to the switch and bi-directional policing is required the configuration is very similar to that of single port policing.

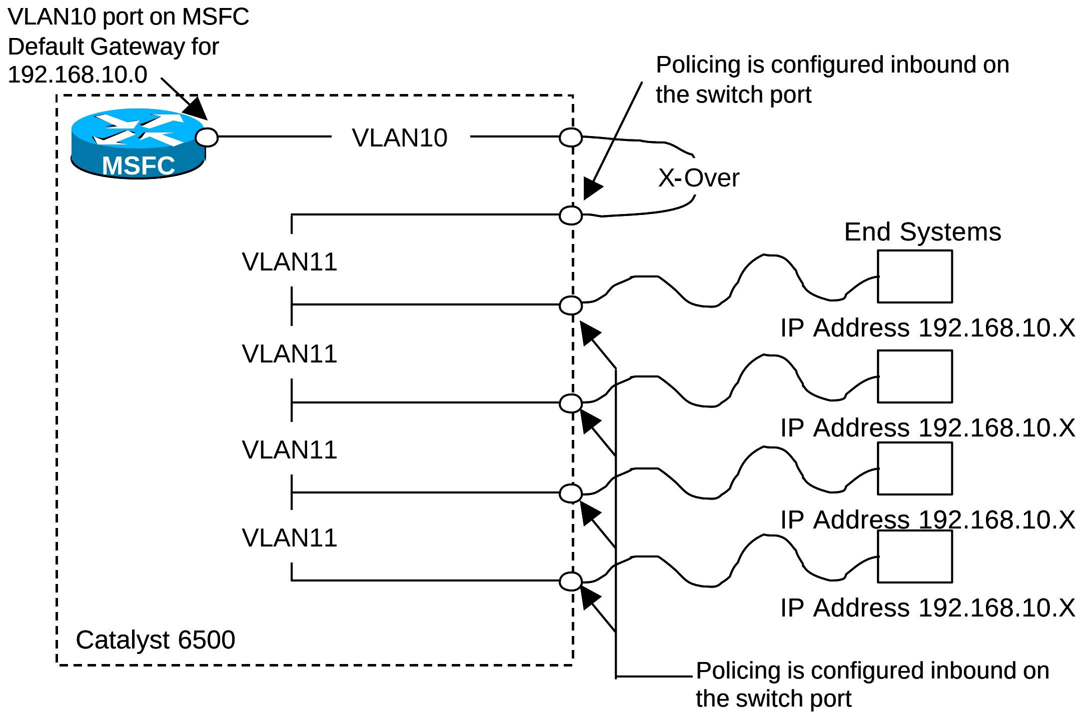

Figure 5. Bi-Directional Policing Configuration for Multiple Ports in the Same VLAN

As can be seen in figure 5. all the server ports are configured in the same VLAN, this means that each server inbound port can have a separate policing entry, but for the port that connects them to the MSFC, this must have an entry that incorporates all the policing rules for all the Servers that require limiting.

The Layer 2 and 3 address resolution process operates in the same way and this configuration will result in each server entry being against both its own connected port and the port that connects the two VLANs together.