When Will You Be An IPexpert?...

Cisco® Catalyst™ 3550 Series Switch Tutorial

1. The Origin of the Cisco® Catalyst™ 3550 Ethernet Switch
    a. What they replaced. (3500XL)

2. Features of the Catalyst™ 3550 Ethernet Switch
    a. Layer 3 / IP Unicast Routing
        i. RIP
        ii. IGRP / EIGRP
        iii. OSPF
        iv. BGP
        v. HSRP
        vi. Distance Vector Multicast Routing Protocol (DVMRP) tunneling
        vii. Multicast Routing
            1. PIM
            2. IGMP
            3. MVR Multicast VLAN Registration Protocol

    b. Security
        i. Gaining Access
            1. Multilevel security on console access (prevents unauth users from altering switch config)
            2. Securing Telnet Access to the Switch
        ii. SSH
        iii. 802.1x Authentication
        iv. RADIUS / TACACS+
        v. Router ACL's
        vi. VLAN-Maps
        vii. When to Use Access-Lists and VLAN-Maps
        viii. Port-Based Security
            1. Port-Based ACLs (PACLs)
            2. Port-Based Traffic Control
                a. Storm Control
                b. Proteced Ports (Similar to Private VLAN)
                c. Port Blocking
                d. Port Security

    c. Quality of Service (QoS)
        i. Advanced QoS
        ii. Automatic QoS (Auto QoS)
        iii. Rate-limiting
        iv. Using Class-Maps
        v. Policy Maps
        vi. Classify, Police, and Mark Using Policy Maps.
        vii. Classify, Police, and Mark Traffic Using Aggregate Policers

    d. Layer 2 VLANS / Spanning Tree Protocol (STP): IEEE 802.1D

i. VTP
ii. Voice VLAN
iii. Standard-Range VLAN Configuration
iv. Configuring Extended VLANs
v. Spanning-tree root guard (STRG)
vi. Loopguard
vii. Uplinkfast
viii. Backbone Fast Configuration
ix. Portfast
x. CrossStack UplinkFast (CSUF)
xi. Per VLAN Spanning Tree Plus (PVST+)
xii. ISL
xiii. 802.1q
xiv. RSTP - IEEE 802.1w Rapid Spanning Tree Protocol (RSTP)
xv. MST – IEEE 802.1s Multiple Spanning Tree
xvi. Fallback Bridging
xvii. UniDirectional Link Detection (UDLD) and Aggressive UDLD

e. EtherChannel

f. Misc
i. Switch Optimization
ii. WCCP
iii. SNMP
iv. SPAN and RSPAN
v. Multi-VRF CE (Virtual Routing Forwarding Customer Edge, also called VRF-lite)
vi. DHCP Option 82 Subscriber Identification
vii. Service Provider-Oriented Functions
1. Layer 2 Protocol Tunneling
viii. Clustering

g. Links

3. Appendix A – Command Reference

The Origin of the Cisco® Catalyst™ 3550 Ethernet Switch

What They Replaced

The Cisco® Catalyst™ 3550 switch has been introduced to replace the aging Catalyst™ 3500XL Layer 2 switch that previously was part of Cisco's answer to the access layer switch market.

As you may already know the 3500XL switch is part of the Catalyst "XL" family which includes the 2900XL, 2900XL LRE, and the 3500XL. Something new to Cisco's lower end switches was that they ran a complete version of IOS. An interesting note is that Cisco in an effort to standardize the IOS over their entire product line, created a switch that ran router software. The XL series switches had quite a number of commands that were "left-over" router commands. For example, you could type "ip address 192.1681.1 255.255.255.0" under interface FastEthernet 0/1 and the switch would take the command as well as display this under the running-configuration. The XL series switches are strictly Layer 2 devices, meaning they had no layer 3 capability outside of the management interface (Telnet, SNMP, etc). This means that your recently entered IP address is useless; however the switch did take the command without error. This was one of the many frustrating "features" of the XL series switches. The IOS was not completely custom-fit for the devices, therefore leaving behind a myriad of unusable commands.

Engineers that were new to the Cisco world appreciated the fact that these devices ran the IOS that was like the software that ran on the routers. Older Engineers that were extremely familiar with Cisco's other LAN switching products such as the Catalyst™ 5000 platform were unimpressed with the devices operating system, and command structure. The 3500XL switch is IOS based; where as the Catalyst™ 5000 is "set-based" which means that the bulk of the commands entered into the 5000 begin with a "set" (e.g set vlan). One difference picked up by those that configured the higher port-density XL switches is that you have to configure every port individually, unlike the 5000 in which you can specify a range of ports in your configuration. These minor setbacks have since been fixed in the newer Catalyst 3550 platform, although Cisco has chosen to stick with IOS as opposed to the set-based OS running on the older Catalyst switches. Cisco has also been migrating the Catalyst 6000/6500 series to Native IOS. Soon every switch from the 8540 down to the 2950 will run Cisco IOS out of the box.

As previously mentioned the Catalyst 3500XL series switch is a Layer 2 device which means that it has no routing capability. If a decision requires the switch to look at anything more than the MAC address then the 3500XL falls short. The Cisco Catalyst 3550 Series switches are a line of enterprise-class, stackable, multilayer switches that provide high availability, security and quality of service (QoS) to enhance the operation of the network. With a range of Fast Ethernet and Gigabit Ethernet configurations, the Catalyst 3550 Series can serve as both a powerful access layer switch for medium enterprise wiring closets and as a backbone switch for mid-sized networks. For the first time, customers can deploy network-wide intelligent services, such as advanced QoS, rate-limiting, Cisco security access control lists (ACLs), multicast management, and high-performance IP routing—while maintaining the simplicity of traditional LAN switching. Embedded in the Catalyst 3550 Series is the Cisco Cluster Management Suite (CMS) Software, which allows users to simultaneously configure and troubleshoot multiple Catalyst desktop switches using a standard Web browser. Cisco CMS Software provides new configuration wizards that greatly simplify the implementation of converged applications and network-wide services.

Basic 3524XL Stats:

10Gbps Switching Fabric, 5Gbps Forwarding Rate, 6.5 million packets-per-second
4mb Shared Memory for Layer 2 switching, Storage of 8,192 MAC addresses

Basic 3550-24-EMI Stats:

8.8Gbps Switching Fabric, 4.4Gbps Forwarding Rate, 6.6 million packets-per-second
2mb Shared Memory shared by all ports, 64mb RAM / 16mb Flash, Storage of 8,000 MACs
16,000 Unicast Routes, 2,000 Multicast Routes, Max MTU 1546 for MPLS bridging.

What is the difference between the SMI and EMI 3550 Switches you ask? The EMI enables a richer set of enterprise-class features including, advanced hardware-based IP unicast and multicast routing, and the Web Cache Communication Protocol (WCCP). Additional details about the differences between the SMI and EMI are provided on CCO (www.cisco.com/go/Catalyst3550). You may ask yourself, can I configure the SMI image to perform Layer 3 Routing? Yes, there is support for basic IP unicast routing via Static and RIPv1/v2 using the SMI. The EMI provides advanced IP unicast and multicast routing. These advanced routing protocols are Open Shortest Path First (OSPF), Interior Gateway Routing Protocol (IGRP), Enhanced Interior Gateway Routing Protocol (EIGRP), Border Gateway Protocol version 4 (BGPv4), and Protocol Independent Multicast (PIM).

These are just a few differences between the new Catalyst 3550 and the legacy Catalyst 3500XL switch. If you would like to learn more about the 3500XL switch, feel free to do so on Cisco's Website www.cisco.com

The duration of this document we will talk about the 3550 and it's amazing capabilities.

Features of the Catalyst™ 3550 Ethernet Switch

Layer 3 Routing

The Catalyst 3550 Switch can be configured very similarly to an IOS based Router. You have a few options with the 3550 in your configurations. You can configure each individual port as a routed port (Layer 3 interface), or you can configure VLAN interfaces to act as SVI's – Switched Virtual Interfaces.

When you configure a port to act as a routed port, it is no different than configuring a Fast Ethernet port on any router. You can assign an IP address to this interface, as well as apply access-lists, QoS related configuration, etc. However you do have to tell the switch that the port is no longer acting as a layer 2 interface by issuing the command **no switchport** followed by entering the desired IP address.

For example:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet0/10
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.1.2.3 255.255.0.0
Switch(config-if)# no shutdown
```

We will cover SVI configuration in the Fallback Bridging section of this paper.

The 3550 supports the following IP Unicast routing protocols; RIP v1/v2, IGRP/EIGRP, OSPF, and BGP. Configuration of each of these protocols is beyond the scope of this documentation. Keep in mind however, that the configuration of the above protocols is possible, and is no different from the same configuration on a Router.

Here is an example of what RIP configuration would look like:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# router rip
Switch(config-router)# network 10.0.0.0
Switch(config-router)# end
```

As you will notice, there is nothing unique about this routing protocol configuration, the same goes for all of the other supported protocols. The one thing you will see is that we entered the "ip routing" command in global configuration mode. This command is required if you are going to transform the switch from a layer 2 device to a device capable of routing IP packets.

HSRP

HSRP is Cisco's standard method of providing high network availability by providing first-hop redundancy for IP hosts on an IEEE 802 LAN configured with a default gateway IP address. HSRP routes IP traffic without relying on the availability of any single router. It enables a set of router interfaces to work together to present the appearance of a single virtual router or default gateway to the hosts on a LAN. When HSRP is configured on a network or segment, it provides a virtual Media Access Control (MAC) address and an IP address that is shared among a group of configured routers. HSRP allows two or more HSRP-configured routers to use the MAC address and IP network address of a virtual router. The virtual router does not exist; it represents the common target for routers that are configured to provide backup to each other. One of the routers is selected to be the active router and another to be the standby router, which assumes control of the group MAC address and IP address should the designated active router fail.

Note: Routers in an HSRP group can be any router interface that supports HSRP, including Catalyst 3550 routed ports and switch virtual interfaces (SVIs).

Multicasting

The Cisco IOS software supports these protocols to implement IP multicast routing:
• Internet Group Management Protocol (IGMP) is used among hosts on a LAN and the routers (and multilayer switches) on that LAN to track the multicast groups of which hosts are members.
• Protocol-Independent Multicast (PIM) protocol is used among routers and multilayer switches to track which multicast packets to forward to each other and to their directly connected LANs.
• Distance Vector Multicast Routing Protocol (DVMRP) is used on the multicast backbone of the Internet (MBONE). The Cisco IOS software supports PIM-to-DVMRP interaction.
• Cisco Group Management Protocol (CGMP) is used on Cisco routers and multilayer switches connected to Layer 2 Catalyst switches to perform tasks similar to those performed by IGMP.

Cisco multicast routers and multilayer switches using PIM can interoperate with non-Cisco multicast routers that use the DVMRP. PIM devices dynamically discover DVMRP multicast routers on attached networks by listening to DVMR probe messages. When a DVMRP neighbor has been discovered, the PIM device periodically sends DVMRP report messages advertising the unicast sources reachable in the PIM domain. By default,

Directly connected subnets and networks are advertised. The device forwards multicast packets that have been forwarded by DVMRP routers and, in turn, forwards multicast packets to DVMRP routers. DVMRP interoperability is automatically activated when a Cisco PIM device receives a DVMRP probe message on a multicast-enabled interface. No specific IOS command is configured to enable DVMRP interoperability; however, you must enable multicast routing.

MVR

Multicast VLAN Registration (MVR) is designed for applications using wide-scale deployment of multicast traffic across an Ethernet ring-based service provider network (for example, the broadcast of multiple television channels over a service-provider network). MVR allows a subscriber on a port to subscribe and unsubscribe to a multicast stream on the network-wide multicast VLAN. It allows the single multicast VLAN to be shared in the network while subscribers remain in separate VLANs. MVR provides the ability to continuously send multicast streams in the multicast VLAN, but to isolate the streams from the subscriber VLANs for bandwidth and security reasons.

MVR assumes that subscriber ports subscribe and unsubscribe (join and leave) these multicast streams by sending out IGMP join and leave messages. These messages can originate from an IGMP version-2-compatible host with an Ethernet connection. Although MVR operates on the underlying mechanism of IGMP snooping, the two features operate independently of each other. One can be enabled or disabled without affecting the behavior of the other feature. However, if IGMP snooping and MVR are both enabled, MVR reacts only to join and leave messages from multicast groups configured under MVR. Join and leave messages from all other multicast groups are managed by IGMP snooping.

The switch CPU identifies the MVR IP multicast streams and their associated MAC addresses in the switch forwarding table, intercepts the IGMP messages, and modifies the forwarding table to include or remove the subscriber as a receiver of the multicast stream, even though the receivers might be in a different VLAN from the source. This forwarding behavior selectively allows traffic to cross between different VLANs.

This example shows how to enable MVR, configure the MVR group address, set the query time to 1 second (10 tenths), specify the MVR multicast VLAN as VLAN 22, set the MVR mode as dynamic, and verify the results:

```
Switch(config)# mvr
Switch(config)# mvr group 228.1.23.4
Switch(config)# mvr querytime 10
Switch(config)# mvr vlan 22
Switch(config)# mvr mode dynamic
Switch(config)# end
Switch# show mvr

MVR Running: TRUE
MVR multicast vlan: 22
MVR Max Multicast Groups: 256
MVR Current multicast groups: 1
MVR Global query response time: 10 (tenths of sec)
MVR Mode: dynamic
```

Security

To prevent unauthorized access to your switch you should configure one or more of the following security features. Passwords on the console and vty lines, username/password pairs stored locally on the switch for individual access, username/password pairs stored on a centrally located server (i.e. TACACS+, or RADIUS). You can also configure privilege levels for passwords. When a user enters the password you have given them they can be granted access at a pre-defined privilege level.

If you have placed the switch in a location that you cannot completely secure, you might want to consider disabling password recovery. With the "**no service password-recovery**," you can disable the option for someone who has physical access to perform password recovery and gain full access to your switch. If you have password-recovery disabled and a user interrupts the boot process they are asked if they would like to proceed and erase the configuration, if they do not want to erase the configuration, the normal configuration is loaded and the user still can't gain access.

SSH

SSH is a protocol that provides a secure, remote connection to a Layer 2 or a Layer 3 device. There are two versions of SSH: SSH version 1 and SSH version 2. This software release only supports SSH version 1. SSH provides more security for remote connections than Telnet by providing strong encryption when a device is authenticated. The SSH feature has an SSH server and an SSH integrated client. The client supports these user authentication methods: TACACS+, RADIUS and Local Username authentication.

To configure SSH on your switch perform the following commands after you have verified you have the crypto image:

hostname
ip domain-name
crypto key generate rsa

```
Switch(config)# hostname Switch
Switch(config)# ip domain-name ipexpert.net
Switch(config)# crypto key generate rsa  (at this point ssh will be
enabled)
```

For local authentication add the following configuration
username/password

```
Switch(config)# username bob password dsb

Switch(config)# line vty 0 4
Switch(config)# login local <-- Required if you want to do local
authentication
Switch(config)# transport input ssh ←-If you want to only allow SSH
```

802.1x

The IEEE 802.1X standard defines a client-server-based access control and authentication protocol that restricts unauthorized clients from connecting to a LAN through publicly accessible ports. The authentication server authenticates each client connected to a switch port before making available any services offered by the switch or the LAN.

Until the client is authenticated, 802.1X access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.

Device roles

Client—the device (workstation) that requests access to the LAN and switch services and responds to requests from the switch. The workstation must be running 802.1X-compliant client software such as that offered in the Microsoft Windows XP operating system. (The client is the *supplicant* in the IEEE 802.1X specification.)

Authentication server—performs the actual authentication of the client. The authentication server validates the identity of the client and notifies the switch whether or not the client is authorized to access the LAN and switch services. Because the switch acts as the proxy, the authentication service is transparent to the client. In this release, the Remote Authentication Dial-In User Service (RADIUS) security system with Extensible Authentication Protocol (EAP) extensions is the only supported authentication server; it is available in Cisco Secure Access Control Server version 3.0. RADIUS operates in a client/server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.

Switch (edge switch or wireless access point)—controls the physical access to the network based on the authentication status of the client. The switch acts as an intermediary (proxy) between the client and the authentication server, requesting identity information from the client, verifying that information with the authentication server, and relaying a response to the client. The switch includes the RADIUS client, which is responsible for encapsulating and decapsulating the Extensible Authentication Protocol (EAP) frames and interacting with the authentication server.

When the switch receives EAPOL frames and relays them to the authentication server, the Ethernet header is stripped and the remaining EAP frame is re-encapsulated in the RADIUS format. The EAP frames are not modified or examined during encapsulation, and the authentication server must support EAP within the native frame format. When the switch receives frames from the authentication server, the server's frame header is removed, leaving the EAP frame, which is then encapsulated for Ethernet and sent to the client.

There are three states a port can be in when using dot1x: force-authorized, force-unauthorized, and auto. If a port is in force-authorized status, then the switch will not prompt for authentication, and will allow all communication through this port. In the force-unauthorized status, the client doesn't even get a chance to authenticate; it is the equivalent of shutting the port down. Even if the user is dot1x capable, the switch just ignores any attempt to communicate through the switch.

Dot1x is supported on Layer 2 static-access ports and Layer 3 routed ports, but is not supported on the following port types: Trunk Port, Dynamic port, Dynamic-access port, EtherChannel Port, Secure Port, or SPAN Ports.

Sample config

```
Switch# configure terminal
Switch(config)# aaa new-model
Switch(config)# radius-server host 172.120.39.46 key rad123
Switch(config)# aaa authentication dot1x default group radius
Switch(config)# interface fastethernet0/1
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
*Italicized values indicate sample values.
```

RADIUS / TACACS+

RADIUS is a distributed client/server system that secures networks against unauthorized access. RADIUS clients run on supported Cisco routers and switches, including Catalyst 3550 multilayer switches and Catalyst 2950 series switches. Clients send authentication requests to a central RADIUS server, which contains all user authentication and network service access information. The RADIUS host is normally a multiuser system running RADIUS server software from Cisco (Cisco Secure Access Control Server version 3.0), Livingston, Merit, Microsoft, or another software provider.

This configuration will prompt for a username/password when you telnet/ssh to the switch:

```
Switch(config)# aaa new-model
Switch(config)# radius-server host 172.120.39.46 key rad123
Switch(config)# aaa authentication login default group radius
```

TACACS+ is a security application that provides centralized validation of users attempting to gain access to your switch. TACACS+ services are maintained in a database on a TACACS+ daemon typically running on a UNIX or Windows NT workstation. You should have access to and should configure a TACACS+ server before the configuring TACACS+ features on your switch.

TACACS+ provides for separate and modular authentication, authorization, and accounting facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service—authentication, authorization, and accounting—independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The goal of TACACS+ is to provide a method for managing multiple network access points from a single management service. Your switch can be a network access server along with other Cisco routers and access servers. A network access server provides connections to a single user, to a network or subnetwork, and to interconnected networks.

Sample configuration using TACACS+ instead of RADIUS

```
Switch(config)# aaa new-model
Switch(config)# tacacs-server host 172.120.39.46 key tac123
Switch(config)# aaa authentication login default group tacacs+
```

Router ACLs

You can apply router ACLs on switch virtual interfaces (SVIs), which are Layer 3 interfaces to VLANs; on physical Layer 3 interfaces; and on Layer 3 EtherChannel interfaces. Router ACLs are applied on interfaces for specific directions (inbound or outbound). You can apply one IP access list in each direction. Router ACL's are identical to the ACL's you configured on a Router. You have the option of standard and extended IP ACL's. On a side note you can not configure Dynamic or Reflexive ACL's on the 3550.
Examples of Standard and Extended

```
Switch(config)# access-list 2 permit 36.48.0.3
Switch(config)# access-list 2 deny 36.48.0.0 0.0.255.255
Switch(config)# access-list 2 permit 36.0.0.0 0.255.255.255
Switch(config)# interface gigabitethernet0/1
```

```
Switch(config-if)# ip access-group 2 in

Switch(config)# access-list 102 permit tcp any host 128.88.1.2 eq 25
Switch(config)# access-list 102 permit icmp any any
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# ip access-group 102 in
```

You can also create Named ACL's as well as Time-based ACL's

VLAN Maps

VLAN maps can access-control *all* traffic. You can apply VLAN maps on the switch to all packets that are routed into or out of a VLAN or are bridged within a VLAN. VLAN maps are used strictly for security packet filtering. Unlike router ACLs, VLAN maps are not defined by direction (input or output). You can configure VLAN maps to match Layer 3 addresses for IP traffic. All non-IP protocols are access-controlled through MAC addresses and Ethertype using MAC VLAN maps. (IP traffic *is not* access controlled by MAC VLAN maps.) You can enforce VLAN maps only on packets going through the switch; you cannot enforce VLAN maps on traffic between hosts on a hub or on another switch connected to this switch. With VLAN maps, forwarding of packets is permitted or denied, based on the action specified in the map.

Things to keep in mind when configuring a VLAN Map

If there is no router ACL configured to deny traffic on a routed VLAN interface (input or output), and *no* VLAN map configured, all traffic is permitted.

Each VLAN map consists of a series of entries. The order of entries in an VLAN map is important. A packet that comes into the switch is tested against the first entry in the VLAN map. If it matches, the action specified for that part of the VLAN map is taken. If there is no match, the packet is tested against the next entry in the map.

If the VLAN map has at least one match clause for the type of packet (IP or MAC) and the packet does not match any of these match clauses, the default is to drop the packet. If there is no match clause for that type of packet in the VLAN map, the default is to forward the packet.

The system might take longer to boot if you have configured a very large number of ACLs.

When a switch has an IP access list or MAC access list applied to a Layer 2 interface, you can create VLAN maps, but you cannot apply a VLAN map to any of the switch VLANs. An error message is generated if you attempt to do so.

VLAN Maps are similar to Route map configuration. You first have to create an ACL, and then in your VLAN map apply this ACL, while still in VLAN-map configuration mode you have to decide on an action to perform on the matched traffic (drop, forward). Traffic is compared sequentially to the VLAN Map, once a match is made no further comparisons are performed.

This example will match all TCP traffic in VLANs 20-22 and drop it:

```
Switch(config)# ip access-list extended ip1
Switch(config-ext-nacl)# permit tcp any any
Switch(config-ext-nacl)# exit
```

```
Switch(config)# vlan access-map VLANmap1 10
Switch(config-access-map)# match ip address ip1
Switch(config-access-map)# action drop

Switch(config)# vlan filter VLANmap1 vlan-list 20-22
```

A cool feature of the VLAN map is that you don't have to necessarily have the switch acting as a Layer 3 device. Let's say hypothetically that you have 3 Catalyst 3550 Series switches in your network. You only want one of these switches acting as your Layer 3 "router," and you want your other 2 switches simply acting as intelligent layer 2 devices. You can configure VLAN maps on your Layer 2 switches to forward or drop certain traffic and filter this at the ingress point. For example, let's say you have a PC hanging off of each layer 2 switch (PC 1 and PC 2), each of these switches (Switch X and Y) are connected to the switch acting as the router (Switch Z). Let's continue in our imaginary network and say that PC 1 is connected to Switch X, and PC 2 is connected to Switch Y, let's also say that we do not want PC 1 to access HTTP information on PC 2. How can we accomplish this with layer 2 devices you might ask? Simple my friend, VLAN Maps!! You can configure something similar to this:

```
Switch(config)# ip access-list extended http
Switch(config-ext-nacl)# permit tcp host PC1 host PC2 eq www
Switch(config-ext-nacl)# exit

Switch(config)# ip access-list extended match_all
Switch(config-ext-nacl)# permit ip any any
Switch(config-ext-nacl)# exit


Switch(config)# vlan access-map map2 10
Switch(config-access-map)# match ip address http
Switch(config-access-map)# action drop
Switch(config-access-map)# exit

Switch(config)# vlan access-map map2 20
Switch(config-access-map)# match ip address match_all
Switch(config-access-map)# action forward

Switch(config)# vlan filter map2 vlan 1
```

This will kill HTTP traffic from PC1 to PC2 at Switch X, therefore reducing bandwidth, and unnecessary processor utilization at Siwtch Z (router). This traffic will just not get bridged to the forwarding engine.

If you wanted to accomplish this with a Router ACL you would have to enable IP routing on your Switches X and Y.

Port Based Security

Port-based ACL (PACL) you can also apply ACLs to Layer 2 interfaces on a switch. Port ACLs are supported on physical interfaces only and not on EtherChannel interfaces. Port ACLs are applied on interfaces for inbound traffic only. These access lists are supported on Layer 2 interfaces:

Standard IP access lists using source addresses

Extended IP access lists using source and destination addresses and optional protocol type information

MAC extended access lists using source and destination MAC addresses and optional protocol type information

As with router ACLs, the switch examines ACLs associated with features configured on a given interface and permits or denies packet forwarding based on how the packet matches the entries in the ACL. However, ACLs can only be applied to Layer 2 interfaces in the inbound direction.

Storm Control

Storm control prevents switchports on a LAN from being disrupted by a broadcast, multicast, or unicast storm on one of the physical interfaces. A LAN storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. Errors in the protocol-stack implementation or in the network configuration can cause a storm. Storm control (or traffic suppression) monitors incoming traffic statistics over a time period and compares the measurement with a predefined suppression level threshold. The threshold represents the percentage of the total available bandwidth of the port. The switch supports separate storm control thresholds for broadcast, multicast, and unicast traffic. If the threshold of a traffic type is reached, further traffic of that type is suppressed until the incoming traffic falls below the threshold level.
By default there is no storm control enabled for any traffic type (broadcast, multicast unicast). Here is an example of configuring a multicast threshold at 53%

```
Switch# configure terminal
Switch(config)# interface fastethernet0/17
Switch(config-if)# storm-control multicast level 53
```

These values are approximations and will begin limiting traffic when the traffic rate has surpassed the configured rate.

Protected Ports (Similar to Private VLANs)

Some applications require that no traffic be forwarded between ports on the same switch so that one neighbor does not see the traffic generated by another neighbor. In such an environment, the use of protected ports ensures that there is no exchange of unicast, broadcast, or multicast traffic between these ports on the switch.
Protected ports have these features:

A protected port does not forward any traffic (unicast, multicast, or broadcast) to any other port that is also a protected port. Traffic cannot be forwarded between protected ports at Layer 2; all traffic passing between protected ports must be forwarded through a Layer 3 device.

Forwarding behavior between a protected port and a nonprotected port proceeds as usual.

```
Switch# configure terminal
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# switchport protected
Switch(config-if)# end
```

You can also disable unknown multicasts and unicasts from being flooded to a protected port with the “switchport block unicast,” and “switchport block multicast” commands.

Port Blocking

As mentioned earlier, you can block the flooding of unknown multicast requests and unicast requests with the commands:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# switchport block multicast
Switch(config-if)# switchport block unicast
```

Port Security

You can use the port security feature to restrict input to an interface by limiting and identifying MAC addresses of the stations allowed to access the port. When you assign secure MAC addresses to a secure port, the port does not forward packets with source addresses outside the group of defined addresses. If you limit the number of secure MAC addresses to one and assign a single secure MAC address, the workstation attached to that port is assured the full bandwidth of the port. If a port is configured as a secure port and the maximum number of secure MAC addresses is reached, when the MAC address of a station attempting to access the port is different from any of the identified secure MAC addresses, a security violation occurs. Also, if a station with a secure MAC address configured or learned on one secure port attempts to access another secure port, a violation is flagged.

There are three types of secured MAC address:

Static secure MAC addresses—these are manually configured by using the **switchport port-security mac-address** *mac-address* interface configuration command, stored in the address table, and added to the switch running configuration.
Dynamic secure MAC addresses—these are dynamically configured, stored only in the address table, and removed when the switch restarts.

Sticky secure MAC addresses—these are dynamically configured, stored in the address table, and added to the running configuration. If these addresses are saved in the configuration file, when the switch restarts, the interface does not need to dynamically reconfigure them.

There are also 3 actions that can be performed in case of a violation;

Protect—when the number of secure MAC addresses reaches the maximum limit allowed on the port, packets with unknown source addresses are dropped until you remove a sufficient number of secure MAC addresses to drop below the maximum value.

Restrict—a port security violation restricts data and causes the SecurityViolation counter to increment.

Shutdown—a port security violation causes the interface to immediately shut down and an SNMP trap notification is sent. When a secure port is in the error-disabled state, you can bring it out of this state by entering the **errdisable recovery cause** *psecure-violation* global configuration command, or you can manually re-enable it by entering the **shutdown** and **no shutdown** interface configuration commands. This is the default mode.

There are other limitations of a secure port that you definitely need to keep in mind. These can be found within 3550 Documentation (www.cisco.com/go/documentation).

Here is an example of configuring Port Security, allowing up to 50 MAC addresses to be learned, and making them sticky. By default it is in "shutdown" mode

```
Switch(config)# interface fastethernet0/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 50
Switch(config-if)# switchport port-security mac-address sticky
Switch(config-if)# end
```

You can also configure an aging time. By default these Secure MAC's will not be aged out and will stay in the MAC table until the switch is powered off, in the case of normal port security. If you are using the sticky option these MAC's will be stored until you clear them manually. You can configure the switch to age the MAC in two different manners; inactivity, or absolute. You can say, "After 120 minutes of inactivity I want MAC addresses on this port to age out of the MAC table." Or you could say "I want MAC addresses to age out after 30 minutes, no matter what."

This example sets the aging time at 2 minutes, and is based on inactivity

```
Switch(config-if)# switchport port-security aging time 2
Switch(config-if)# switchport port-security aging type inactivity
```

Quality of Service

Advanced QoS

Typically, networks operate on a best-effort delivery basis, which means that all traffic has equal priority and an equal chance of being delivered in a timely manner. When congestion occurs, all traffic has an equal chance of being dropped.

When you configure QoS, you can select specific network traffic, prioritize it according to its relative importance, and use congestion-management and congestion-avoidance techniques to provide preferential treatment. Implementing QoS in your network makes network performance more predictable and bandwidth utilization more effective.

The QoS implementation is based on the DiffServ architecture, an emerging standard from the Internet Engineering Task Force (IETF). This architecture specifies that each packet is classified upon entry into the network. The classification is carried in the IP packet header, using 6 bits from the deprecated IP type of service (TOS) field to carry the classification (*class*) information. Classification can also be carried in the Layer 2 frame.

Prioritization bits in Layer 2 frames:

- Layer 2 Inter-Switch Link (ISL) frame headers have a 1-byte User field that carries an IEEE 802.1p class of service (CoS) value in the three least-significant bits. On interfaces configured as Layer 2 ISL trunks, all traffic is in ISL frames.
- Layer 2 802.1Q frame headers have a 2-byte Tag Control Information field that carries the CoS value in the three most-significant bits, which are called the User Priority bits. On interfaces configured as Layer 2 802.1Q trunks, all traffic is in 802.1Q frames except for traffic in the native VLAN.
- Other frame types cannot carry Layer 2 CoS values.
- Layer 2 CoS values range from 0 for low priority to 7 for high priority.

Prioritization bits in Layer 3 packets:

- Layer 3 IP packets can carry either an IP precedence value or a Differentiated Services Code Point (DSCP) value. QoS supports the use of either value because DSCP values are backward-compatible with IP precedence values.
- IP precedence values range from 0 to 7.
- DSCP values range from 0 to 63.

To provide the same forwarding treatment to packets with the same class information and different treatment to packets with different class information, all switches and routers that access the Internet rely on class information. Class information in the packet can be assigned by end hosts or by switches or routers along the way, based on a configured policy, detailed examination of the packet, or both. Detailed examination of the packet is expected to happen closer to the network edge so that core switches and routers are not overloaded.

Switches and routers along the path can use class information to limit the amount of resources allocated per traffic class. The behavior of an individual device when handling traffic in the DiffServ architecture is called per-hop behavior. If all devices along a path provide a consistent per-hop behavior, you can construct an end-to-end QoS solution.

Implementing QoS in your network can be a simple or complex task and depends on the QoS features offered by your internetworking devices, the traffic types and patterns in your network, and the granularity of control that you need over incoming and outgoing traffic.

By default the 3550 will have these values for the CoS to DSCP mapping

CoS value	0	1	2	3	4	5	6	7
DSCP value	0	8	16	24	32	40	48	56

You can modify these values with the command: mls qos map cos-dscp

```
Switch# configure terminal
Switch(config)# mls qos map cos-dscp 10 15 20 25 30 35 40 45
Switch(config)# end
```

That command modifies dscp1…dscp8, with a range up to 63.

Auto QoS

You can configure the 3550 switch to take the QoS values of a packet that are set by the originator of the flow, and "trust" them. To trust the DSCP value configured by another device you would use the following command under the ingress interface:

```
Switch(config-if)# mls qos trust dscp
```

If you wanted to trust the CoS value, you would use this command

```
Switch(config-if)# mls qos trust cos
```

Rate Limiting

You can configure the 3550 switch to limit traffic just like you can on a router with the "police," command entered in policy-map configuration mode. For example if you wanted to limit traffic to an average rate of 5mb with burst capability to 2mb, and drop exceeding traffic, you would use this command

```
police 5000000 2000000 exceed-action drop
```

Class Maps and Policy Maps

QoS configuration is modular in fashion, meaning you configure different modules of your policy and then pull it all together under the interface. Class maps are used to define the traffic that will be policed, or manipulated. Under class map configuration you can specify an access-list to match, IP precedence, CoS or DSCP values. This example classifies traffic that came from the IP address 10.1.1.1

access-list 10 permit 10.1.1.1

class-map bobclass
 match access-group 10

This is the first module of our QoS configuration, now we can create a policy map to specify what we want to do to our classified traffic.

Policy Maps

Policy maps are the 2nd module of this whole puzzle. They are used to police and mark the classified traffic. For example

policy-map bobpolicy
 class bobclass
 set ip dscp 56
 police 2500000 200000 exceed-action drop

These modules are all pulled together under the preferred interface with the command: service policy [input | output] bobpolicy

Between the above two examples you learned how to classify, police, and mark using policy maps. Now we will show how to classify, police, and mark using Aggregate policers. Aggregate policers allow the switch to use the same policer for multiple flows, and are recommended for a smaller number of combined flows.

This example shows how to create an aggregate policer and attach it to multiple classes within a policy map. In the configuration, the IP ACLs permit traffic from network 10.1.0.0 and from host 11.3.1.1. For traffic coming from network 10.1.0.0, the DSCP in the incoming packets is trusted. For traffic coming from host 11.3.1.1, the DSCP in the packet is changed to 56. The traffic rate from the 10.1.0.0 network and from host 11.3.1.1 is policed. If the traffic exceeds an average rate of 48000 bps and a normal burst size of 8000 bytes, its DSCP is marked down (based on the policed-DSCP map) and sent. The policy map is attached to an ingress interface.

```
Switch(config)# access-list 1 permit 10.1.0.0 0.0.255.255
Switch(config)# access-list 2 permit 11.3.1.1
Switch(config)# mls qos aggregate-police transmit1 48000 8000 exceed-
action
policed-dscp-transmit

Switch(config)# class-map ipclass1
Switch(config-cmap)# match access-group 1
Switch(config-cmap)# exit

Switch(config)# class-map ipclass2
Switch(config-cmap)# match access-group 2
Switch(config-cmap)# exit

Switch(config)# policy-map aggflow1
```

```
Switch(config-pmap)# class ipclass1
Switch(config-pmap-c)# trust dscp
Switch(config-pmap-c)# police aggregate transmit1
Switch(config-pmap-c)# exit

Switch(config-pmap)# class ipclass2
Switch(config-pmap-c)# set ip dscp 56
Switch(config-pmap-c)# police aggregate transmit1
Switch(config-pmap-c)# exit

Switch(config-pmap)# exit

Switch(config)# interface gigabitethernet0/1
Switch(config-if)# service-policy input aggflow1
Switch(config-if)# exit
```

The policed-DSCP map is a configuration that maps DSCP levels to lower levels for when you've chosen to "mark them down"

```
Switch# configure terminal
Switch(config)# mls qos map policed-dscp 50 51 52 53 54 55 56 57 to 0
Switch(config)# end
```

This example says, that when you want to mark down the DSCP levels 50-57, it will mark them to 0.

Layer 2 VLANs / Spanning Tree Protocol 802.1d

VTP

VLAN Trunking Protocol was developed by Cisco Systems as a means of deploying VLAN configuration network wide. VTP allows central management of VLAN configuration, modification, and deletion. You can specify all the parameters of a VLAN on one switch and have these changes propagated to each switch within the VTP Domain. There are 3 modes that a switch can be in when it's participating in VTP, they are: Server, Transparent, and Client.

When you configure the switch to act as a Server, this is the device that you create, modify and delete you VLANs on. This device then turns around and sends out a VTP update across its trunk links. If you configure a switch to act as a Client, it will take the update received from the Server and update his own VLAN database with this new information. It will also forward these updates out its other trunk ports. Each VTP update is sent with a configuration revision number to indicate how priority of the update. The higher the number, the more recent and accurate the configuration is considered to be. Each time you modify the VLAN database the VTP Server will increase the configuration revision number by one and send this update out. When the clients receive these updates, they know that the database has changed and they need to overwrite their database with this new configuration. Lastly if you configure a switch to act as a Transparent VTP device, it will not update its own table with these update, but it will forward them on to other VTP devices. To briefly overview the modes of operation; Server – can create, modify, and delete VLANs, VLANs created on the server are stored in NVRAM. This is the default mode of operation for a switch. If you have no other switches in your network, then leave the switch in this mode so that you can create VLANs.

Client – cannot create, modify or delete VLANs, VLANs learned from a Server are not stored in NVRAM and erased when the switch is rebooted.

Transparent – can create, modify, and delete VLANs, but they are not propagated to any other switch. This device still forwards VTP updates received on its trunk ports.

You can configure VTP in VLAN configuration mode, as well as global configuration. These examples show both modes of configuration. They also show configuring a VTP domain name and password, both of which need to be identical on every other switch in the domain.

```
Switch# config terminal
Switch(config)# vtp mode server
Switch(config)# vtp domain eng_group
Switch(config)# vtp password mypassword
Switch(config)# end

Switch# vlan database
Switch(vlan)# vtp server
Switch(vlan)# vtp domain eng_group
Switch(vlan)# vtp password mypassword
Switch(vlan)# exit
APPLY completed.
Exiting....
```

Configuration of a client or transparent switch is identical to the config above, except you specify client or transparent in place of server.

Voice VLAN

Voice VLANs are VLANs that are for the sole purpose of carrying Voice over IP traffic. They are their own broadcast domain, and the switch can perform QoS on both Voice traffic and the other IP traffic coming into the port. In typical Voice VLAN configuration you plug your PC into the Cisco 7960 IP phone, and then plug your phone into a switchport. The IP Phone configures a trunk to the switch and sends the Voice traffic tagged with both an 802.1q header and a priority of 5, or 802.1q header with a priority of 5. The IP traffic that is forwarded from the PC goes to the switch untagged or on the "native VLAN"
Example config with 802.1q

```
Switch(config)# mls qos
Switch(config)# interface interface-id
Switch(config-if)# mls qos trust cos
Switch(config-if)# switchport voice vlan vlan-id
```

If you don't want to trunk, you can do the same thing with 802.1p

```
Switch(config)# mls qos
Switch(config)# interface interface-id
Switch(config-if)# mls qos trust cos
Switch(config-if)# switchport voice vlan dot1p
```

Standard VLAN Configuration

You can configure VLAN's within the range of 1-1005. You can do this in either VLAN configuration mode or in global configuration mode

Example

```
Switch# configure terminal
Switch(config)# vlan 20
```

```
Switch(config-vlan)# name test20
Switch(config-vlan)# end

Switch# vlan database
Switch(vlan)# vlan 20 name test20
Switch(vlan)# exit
APPLY completed.
Exiting....
```

Extended Range VLANs

When the switch is in VTP transparent mode (VTP disabled), you can create extended-range VLANs (in the range 1006 to 4094). Extended-range VLANs enable service providers to extend their infrastructure to a greater number of customers. The extended-range VLAN IDs are allowed for any switchport commands that allow VLAN IDs. You always use config-vlan mode (accessed by entering the **vlan** *vlan-id* global configuration command) to configure extended-range VLANs. The extended range is not supported in VLAN configuration mode (accessed by entering the **vlan database** privileged EXEC command).

Example

```
Switch(config)# vtp mode transparent
Switch(config)# vlan 2000
Switch(config-vlan)# end
```

Spanning Tree Root Guard

The Layer 2 network of a service provider (SP) can include many connections to switches that are not owned by the SP. In such a topology, the spanning tree can reconfigure itself and select a *customer switch* as the root switch. You can avoid this situation by configuring root guard on SP switch interfaces that connect to switches in your customer's network. If spanning-tree calculations cause an interface in the customer network to be selected as the root port, root guard then places the interface in the root-inconsistent (blocked) state to prevent the customer's switch from becoming the root switch or being in the path to the root.
This is enabled with the command **spanning-tree guard root**

Loopguard

You can use loop guard to prevent alternate or root ports from becoming designated ports because of a failure that leads to a unidirectional link. This feature is most effective when it is configured on the entire switched network.
If your switch is running PVST or MSTP, you can enable this feature by using the **spanning-tree loopguard default** global configuration command.

UplinkFast

If a switch looses connectivity, it begins using the alternate paths as soon as the spanning tree selects a new root port. By enabling UplinkFast with the **spanning-tree uplinkfast** global configuration command, you can accelerate the choice of a new root port when a link or switch fails or when the spanning tree reconfigures itself. The root port transitions to the forwarding state immediately without going through the listening and learning states, as it would with the normal spanning-tree procedures. The UplinkFast feature is supported only when the switch is running PVST.
UplinkFast provides fast convergence after a direct link failure and achieves load balancing between redundant Layer 2 links using uplink groups. An uplink group is a set of Layer 2 interfaces (per VLAN), only one of which is forwarding at any given time.

Specifically, an uplink group consists of the root port (which is forwarding) and a set of blocked ports, except for self-looping ports. The uplink group provides an alternate path in case the currently forwarding link fails.

BackboneFast

BackboneFast detects indirect failures in the core of the backbone. BackboneFast is a complementary technology to the UplinkFast feature, which responds to failures on links directly connected to access switches. BackboneFast optimizes the maximum-age timer, which determines the amount of time the switch stores protocol information received on an interface. When a switch receives an inferior BPDU from the designated port of another switch, the BPDU is a signal that the other switch might have lost its path to the root, and BackboneFast tries to find an alternate path to the root. BackboneFast, which is enabled by using the **spanning-tree backbonefast** global configuration command, starts when a root port or blocked port on a switch receives inferior BPDUs from its designated bridge. An inferior BPDU identifies one switch as both the root bridge and the designated bridge. When a switch receives an inferior BPDU, it means that a link to which the switch is not directly connected (an indirect link) has failed (that is, the designated bridge has lost its connection to the root switch). Under spanning-tree rules, the switch ignores inferior BPDUs for the configured maximum aging time specified by the **spanning-tree max-age** global configuration command. The BackboneFast feature is supported only when the switch is running PVST.

PortFast

PortFast immediately brings an interface configured as an access or trunk port to the forwarding state from a blocking state, bypassing the listening and learning states. You can use Port Fast on ports connected to a single workstation or server, to allow those devices to immediately connect to the network, rather than waiting for the spanning tree to converge. Ports connected to a single workstation or server should not receive bridge protocol data units (BPDUs). A port with Port Fast enabled goes through the normal cycle of spanning-tree status changes when the switch is restarted.

Note: Because the purpose of Port Fast is to minimize the time ports must wait for spanning-tree to converge, it is effective only when used on ports connected to end stations. If you enable Port Fast on a port connecting to another switch, you risk creating a spanning-tree loop. If your switch is running PVST or MSTP, you can enable this feature by using the **spanning-tree portfast** interface configuration or the **spanning-tree portfast default** global configuration command

Cross Stack UplinkFast (CSUF)

Cross-stack UplinkFast (CSUF) provides a fast spanning-tree transition (fast convergence in less than 1 second under normal network conditions) across a stack of switches that use the GigaStack GBICs connected in a shared cascaded configuration (multidrop backbone). During the fast transition, an alternate redundant link on the stack of switches is placed in the forwarding state without causing temporary spanning-tree loops or loss of connectivity to the backbone. With this feature, you can have a redundant and resilient network in some configurations. You enable CSUF by using the **spanning-tree stack-port** interface configuration command. The CSUF feature is supported only when the switch is running PVST

PVST+

The IEEE 802.1Q standard for VLAN trunks imposes some limitations on the spanning-tree strategy for a network. The standard requires only one spanning-tree instance for *all* VLANs allowed on the trunks. However, in a network of Cisco switches connected through 802.1Q trunks, the switches maintain one spanning-tree instance for *each* VLAN allowed on the trunks.

When you connect a Cisco switch to a non-Cisco device through an 802.1Q trunk, the Cisco switch uses per-VLAN spanning tree+ (PVST+) to provide spanning-tree interoperability. It combines the spanning-tree instance of the 802.1Q VLAN of the trunk with the spanning-tree instance of the non-Cisco 802.1Q switch.
However, all PVST+ information is maintained by Cisco switches separated by a cloud of non-Cisco 802.1Q switches. The non-Cisco 802.1Q cloud separating the Cisco switches is treated as a single trunk link between the switches.
PVST+ is automatically enabled on 802.1Q trunks, and no user configuration is required. The external spanning-tree behavior on access ports and Inter-Switch Link (ISL) trunk ports is not affected by PVST+.

ISL

Inter-switch link was developed by Cisco Systems as means of supporting multiple VLANs over a single link. ISL is an encapsulation protocol that allows Ethernet frames to get encapsulated with the proper VLAN information. ISL adds a 26 byte header and a new 4 byte CRC frame at the end of the packet. With an ISL trunk port, all received packets are expected to be encapsulated with an ISL header, and all transmitted packets are sent with an ISL header. Native (non-tagged) frames received from an ISL trunk port are dropped. Catalyst 3550 Series switches, also participate in DTP (Dynamic Trunking Protocol) that enables two trunk capable switches to negotiate a trunk. The modes available are desirable and auto, which is default. When in desirable mode the switch will negotiate a trunk with another capable device that is in either auto or desirable mode. If two switches are in auto mode, they will not negotiate a trunk

```
Switch(config)# interface fastethernet0/4
Switch(config-if)# switchport mode dynamic desirable
Switch(config-if)# switchport trunk encapsulation isl
Switch(config-if)# end
```

802.1q

Dot1q is an IEEE standard for relaying multiple VLANs across the same link. 802.1q only adds 4 new bytes of information to the Ethernet frame, and replaces the old CRC with a new value.

```
Switch(config)# interface fastethernet0/4
Switch(config-if)# switchport mode dynamic desirable
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# end
```

RSTP

The RSTP takes advantage of point-to-point wiring and provides rapid convergence of the spanning tree. Reconfiguration of the spanning tree can occur in less than 1 second (in contrast to 50 seconds with the default settings in the 802.1D spanning tree), which is critical for networks carrying delay-sensitive traffic such as voice and video.

Root port—provides the best path (lowest cost) when the switch forwards packets to the root switch.
Designated port—connects to the designated switch, which incurs the lowest path cost when forwarding packets from that LAN to the root switch. The port through which the designated switch is attached to the LAN is called the designated port.
Alternate port—offers an alternate path toward the root switch to that provided by the current root port.
Backup port—acts as a backup for the path provided by a designated port toward the leaves of the spanning tree. A backup port can exist only when two ports are connected

together in a loopback by a point-to-point link or when a switch has two or more connections to a shared LAN segment.
Disabled port—has no role within the operation of the spanning tree.

MST

MSTP, which uses RSTP for rapid convergence, enables VLANs to be grouped into a spanning-tree instance, with each instance having a spanning-tree topology independent of other spanning-tree instances. This architecture provides multiple forwarding paths for data traffic, enables load balancing, and reduces the number of spanning-tree instances required to support a large number of VLANs.

The UplinkFast, BackboneFast, and cross-stack UplinkFast features are not supported with the RSTP and MSTP.

Per-VLAN RSTP is not supported. When you enable MST by using the **spanning-tree mode mst** global configuration command, RSTP is enabled.

PVST, PVST+ and MSTP are supported, but only one version can be active at any time; all VLANs run PVST, or all VLANs run MSTP.

VTP propagation of the MST configuration is not supported. However, you can manually configure the MST configuration (region name, revision number, and VLAN-to-instance mapping) on each switch within the MST region by using the command-line interface (CLI) or through the SNMP support.

For load balancing across redundant paths in the network to work, all VLAN-to-instance mapping assignments must match; otherwise, all traffic flows on a single link.

All MST boundary ports must be forwarding for load balancing between a PVST+ and an MST cloud. For this to happen, the IST master of the MST cloud should also be the root of the CST. If the MST cloud consists of multiple MST regions, one of the MST regions must contain the CST root, and all of the other MST regions must have a better path to the root contained with the MST cloud than a path through the PVST+ cloud. You might have to manually configure the switches in the clouds.

Partitioning the network into a large number of regions is not recommended. However, if this situation is unavoidable, we recommend that you partition the switched LAN into smaller LANs interconnected by routers or non-Layer 2 devices.
Example

```
Switch(config)# spanning-tree mst configuration
Switch(config-mst)# instance 1 vlan 10-20
Switch(config-mst)# name region1
Switch(config-mst)# revision 1
Switch(config-mst)# show pending
Pending MST configuration
Name      [region1]
Revision  1
Instance  Vlans Mapped
--------  --------------------
0         1-9,21-4094
1         10-20
-------------------------------
Switch(config-mst)# exit
Switch(config)#
```

Fallback Bridging

With fallback bridging, the switch bridges together two or more VLANs or routed ports, essentially connecting multiple VLANs within one bridge domain. Fallback bridging forwards traffic that the switch does not route and forwards traffic belonging to a nonroutable protocol such as DECnet. Fallback bridging does not allow the spanning trees from the VLANs being bridged to collapse; each VLAN has its own spanning-tree instance and a separate spanning tree, called the VLAN-bridge spanning tree, which runs on top of the bridge group to prevent loops. A VLAN bridge domain is represented with switch virtual interface (SVI). A set of SVIs and routed ports (which do not have any VLANs associated with them) can be configured (grouped together) to form a bridge group. Recall that an SVI represents a VLAN of switch ports as one interface to the routing or bridging function in the system. You associate only one SVI with a VLAN, and you configure an SVI for a VLAN only when you want to route between VLANs, to fallback-bridge nonroutable protocols between VLANs, or to provide IP host connectivity to the switch. A routed port is a physical port that acts like a port on a router, but it is not connected to a router. A routed port is not associated with a particular VLAN, does not support VLAN subinterfaces, but behaves like a normal routed interface.
A bridge group is an internal organization of network interfaces on a switch. Bridge groups cannot be used to identify traffic switched within the bridge group outside the switch on which they are defined. Bridge groups on the same switch function as distinct bridges; that is, bridged traffic and bridge protocol data units (BPDUs) are not exchanged between different bridge groups on a switch. An interface can be a member of only one bridge group. Use a bridge group for each separately bridged (topologically distinct) network connected to the switch. These are the reasons for placing network interfaces into a bridge group:

To bridge all nonrouted traffic among the network interfaces making up the bridge group. If the packet destination address is in the bridge table, the packet is forwarded on a single interface in the bridge group. If the packet destination address is not in the bridge table, the packet is flooded on all forwarding interfaces in the bridge group. The switch places source addresses in the bridge table as it learns them during the bridging process.

To participate in the spanning-tree algorithm by receiving, and in some cases sending, BPDUs on the LANs to which they are attached. A separate spanning-tree process runs for each configured bridge group. Each bridge group participates in a separate spanning-tree instance. A bridge group establishes a spanning-tree instance based on the BPDUs it receives on only its member interfaces.
This example shows configuring a Layer 3 port and an SVI, and combining them into the bridging process. Non-IP traffic will be bridged, and IP traffic will be routed.

```
Switch(config)# bridge 10 protocol vlan-bridge
Switch(config)# vlan 2
Switch(config-vlan)# exit

Switch(config)# interface vlan2
Switch(config-if)# ip address 172.20.128.1 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface gigabitethernet0/2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 2
Switch(config-if)# no switchport
Switch(config-if)# ip address 172.20.130.1 255.255.255.0
Switch(config-if)# no shutdown
Switch(config-if)# bridge-group 10
```

UniDirectional Link Detection (UDLD)

UDLD is a Layer 2 protocol that enables devices connected through fiber-optic or twisted-pair Ethernet cables to monitor the physical configuration of the cables and detect when a unidirectional link exists. All connected devices must support UDLD for the protocol to successfully identify and disable unidirectional links. When UDLD detects a unidirectional link, it administratively shuts down the affected port and alerts you. Unidirectional links can cause a variety of problems, including spanning-tree topology loops. UDLD works with the Layer 1 mechanisms to determine the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected interfaces. When you enable both autonegotiation and UDLD, Layer 1 and Layer 2 detections work together to prevent physical and logical unidirectional connections and the malfunctioning of other protocols. A unidirectional link occurs whenever traffic sent by the local device is received by the neighbor but traffic from the neighbor is not received by the local device. If one of the fiber strands in a pair is disconnected, as long as autonegotiation is active, the link does not stay up. In this case, the logical link is undetermined, and UDLD does not take any action. If both fibers are working normally from a Layer 1 perspective, UDLD at Layer 2 determines whether those fibers are connected correctly and whether traffic is flowing bidirectionally between the correct neighbors. This check cannot be performed by autonegotiation because autonegotiation operates at Layer 1.
You can enable UDLD globally for all Fiber-optic interfaces, or per interface for any media type.

```
Switch(config)# udld enable
Switch(config-if)# udld enable
```

EtherChannel

An EtherChannel consists of individual Fast Ethernet or Gigabit Ethernet links bundled into a single logical link. The EtherChannel provides full-duplex bandwidth up to 800 Mbps (Fast EtherChannel) or 8 Gbps (Gigabit EtherChannel) between your switch and another switch or host.

You create an EtherChannel for Layer 2 interfaces differently from Layer 3 interfaces. Both configurations involve logical interfaces.

With Layer 3 interfaces, you manually create the logical interface by using the **interface port-channel** global configuration command.

With Layer 2 interfaces, the logical interface is dynamically created.

With both Layer 3 and 2 interfaces, you manually assign an interface to the EtherChannel by using the **channel-group** interface configuration command. This command binds the physical and logical ports together.

The Port Aggregation Protocol (PAgP) facilitates the automatic creation of EtherChannels by exchanging packets between Ethernet interfaces. By using PAgP, the switch learns the identity of partners capable of supporting PAgP and learns the capabilities of each interface. It then dynamically groups similarly configured interfaces into a single logical link (channel or aggregate port); these interfaces are grouped based on hardware, administrative, and port parameter constraints. For example, PAgP groups the interfaces with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. After grouping the links into an EtherChannel, PAgP adds the group to the spanning tree as a single switch port.

This example shows configuration of a Layer 2 FEC.

```
Switch# configure terminal
Switch(config)# interface range gigabitethernet0/4 -5
Switch(config-if-range)# switchport mode access
Switch(config-if-range)# switchport access vlan 10
Switch(config-if-range)# channel-group 5 mode desirable
Switch(config-if-range)# end
```

This is an example of a Layer 3 FEC

```
Switch# configure terminal
Switch(config)# interface port-channel 5
Switch(config-if)# no switchport
Switch(config-if)# ip address 172.10.20.10 255.255.255.0

Switch(config)# interface range gigabitethernet0/4 -5
Switch(config-if-range)# no ip address
Switch(config-if-range)# channel-group 5 mode desirable
Switch(config-if-range)# end
```

Misc

Switch Optimization

By using Switch Database Management (SDM) templates, you can configure memory resources in the switch to optimize support for specific features, depending on how the switch is used in your network. You can select one of four templates to specify how system resources are allocated. You can then approximate the maximum number of unicast MAC addresses, Internet Group Management Protocol (IGMP) groups, quality of service (QoS) access control entries (ACEs), security ACEs, unicast routes, multicast routes, subnet VLANs (routed interfaces), and Layer 2 VLANs that can be configured on the switch. The four templates prioritize system memory to optimize support for these types of features:

QoS and security ACEs—The access template might typically be used in an access switch at the network edge where the route table sizes might not be substantial. Filtering and QoS might be more important because an access switch is the entry to the whole network.

Routing—The routing template maximizes system resources for unicast routing, typically required for a router or aggregator in the center of a network.

VLANs—The VLAN template disables routing and supports the maximum number of unicast MAC addresses. It would typically be selected for a Catalyst 3550 used as a Layer 2 switch.

Default—The default template gives balance to all functionalities (QoS, ACLs, unicast routing, multicast routing, VLANs and MAC addresses).

You can also enable the switch to support 144-bit Layer 3 TCAM, allowing extra fields in the stored routing tables, by reformatting the routing table memory allocation. Using the **extended-match** keyword with the default, access, or routing templates reformats the allocated TCAM by reducing the number of allowed unicast routes, and storing extra routing information in the lower 72 bits of the Layer 3 TCAM. The 144-bit Layer 3 TCAM is required when running the Web Cache Communication Protocol (WCCP) or multiple

VPN routing/forwarding (multi-VRF) instances in customer edge (CE) devices (multi-VRF CE) on the switch.

Example of configuring the routing template

```
Switch(config)# sdm prefer routing
Switch(config)# end
Switch# reload
Proceed with reload? [confirm]
```

Web Cache Communications Protocol (WCCP)

The WCCP and Cisco cache engines (or other caches running WCCP) localize web-traffic patterns in the network, enabling content requests to be fulfilled locally. WCCP enables supported Cisco routers and switches to transparently redirect content requests. With transparent redirection, users do not have to configure their browsers to use a web proxy. Instead, they can use the target URL to request content, and their requests are automatically redirected to a cache engine. The word *transparent* means that the end user does not know that a requested file (such as a web page) came from the cache engine instead of from the originally specified server.

When a cache engine receives a request, it attempts to service it from its own local cache. If the requested information is not present, the cache engine sends a separate request to the end server to retrieve the requested information. After receiving the requested information, the cache engine forwards it to the requesting client and also caches it to fulfill future requests.

This software release supports only WCCP version 2 (WCCPv2). Only a subset of WCCPv2 features are supported.

With WCCPv2, multiple routers or switches can service the cache-engine cluster (a series of cache engines); however, in this release, only one Catalyst 3550 switch can service the cluster. Content is not duplicated on the cache engines.

This example shows how to configure routed interfaces and to enable the web cache service. Fast Ethernet interface 0/1 is connected to the cache engine, is configured as a routed port with an IP address of 172.20.10.30, and is re-enabled. Gigabit Ethernet interface 0/1 is connected through the Internet to the web server, is configured as a routed port with an IP address of 175.20.20.10, and is re-enabled. Fast Ethernet interfaces 0/2 to 0/5 are connected to the clients and are configured as routed ports with IP addresses 175.20.30.20, 175.20.40.30, 175.20.50.40, and 175.20.60.50. The switch redirects HTTP packets received from the client interfaces to the cache engine.

```
Switch# configure terminal
Switch(config)# ip wccp web-cache
Switch(config)# interface fastethernet0/1
Switch(config-if)# no switchport
Switch(config-if)# ip address 172.20.10.30 255.255.255.0
Switch(config-if)# no shutdown
Switch(config-if)# exit

Switch(config)# interface gigabitethernet0/1
Switch(config-if)# no switchport
Switch(config-if)# ip address 175.20.20.10 255.255.255.0
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

```
Switch(config)# interface fastethernet0/2
Switch(config-if)# no switchport
Switch(config-if)# ip address 175.20.30.20 255.255.255.0
Switch(config-if)# no shutdown
Switch(config-if)# ip wccp web-cache redirect in
Switch(config-if)# exit

Switch(config)# interface fastethernet0/3
Switch(config-if)# no switchport
Switch(config-if)# ip address 175.20.40.30 255.255.255.0
Switch(config-if)# no shutdown
Switch(config-if)# ip wccp web-cache redirect in
Switch(config-if)# exit

Switch(config)# interface fastethernet0/4
Switch(config-if)# no switchport
Switch(config-if)# ip address 175.20.50.40 255.255.255.0
Switch(config-if)# no shutdown
Switch(config-if)# ip wccp web-cache redirect in
Switch(config-if)# exit

Switch(config)# interface fastethernet0/5
Switch(config-if)# no switchport
Switch(config-if)# ip address 175.20.60.50 255.255.255.0
Switch(config-if)# no shutdown
Switch(config-if)# ip wccp web-cache redirect in
Switch(config-if)# exit
```

SNMP

SNMP is an application-layer protocol that provides a message format for communication between managers and agents. The SNMP system consists of an SNMP manager, an SNMP agent, and a management information base (MIB). The SNMP manager can be part of a network management system (NMS) such as CiscoWorks. The agent and MIB reside on the switch. To configure SNMP on the switch, you define the relationship between the manager and the agent. The SNMP agent contains MIB variables whose values the SNMP manager can request or change. A manager can get a value from an agent or store a value into the agent. The agent gathers data from the MIB, the repository for information about device parameters and network data. The agent can also respond to a manager's requests to get or set data. An agent can send unsolicited traps to the manager. Traps are messages alerting the SNMP manager to a condition on the network. Traps can mean improper user authentication, restarts, link status (up or down), MAC address tracking, closing of a TCP connection, loss of connection to a neighbor, or other significant events.

This example shows how to permit any SNMP manager to access all objects with read-only permission using the community string *public*. The switch also sends VTP traps to the hosts 192.180.1.111 and 192.180.1.33 using SNMPv1 and to the host 192.180.1.27 using SNMPv2C. The community string *public* is sent with the traps.

```
Switch(config)# snmp-server community public
Switch(config)# snmp-server enable traps vtp
Switch(config)# snmp-server host 192.180.1.27 version 2c public
Switch(config)# snmp-server host 192.180.1.111 version 1 public
Switch(config)# snmp-server host 192.180.1.33 public
```

SPAN / RSPAN

You can analyze network traffic passing through ports or VLANs by using SPAN to send a copy of the traffic to another port on the switch that has been connected to a SwitchProbe device or other Remote Monitoring (RMON) probe. SPAN mirrors received or sent (or both) traffic on a source port and received traffic on one or more source ports or source VLANs, to a destination port for analysis.

Only traffic that enters or leaves source ports or traffic that enters source VLANs can be monitored by using SPAN; traffic that gets routed to ingress source ports or source VLANs cannot be monitored. For example, if incoming traffic is being monitored, traffic that gets routed from another VLAN to the source VLAN is not monitored; however, traffic that is received on the source VLAN and routed to another VLAN is monitored.

RSPAN extends SPAN by enabling remote monitoring of multiple switches across your network. The traffic for each RSPAN session is carried over a user-specified RSPAN VLAN that is dedicated for that RSPAN session in all participating switches. The SPAN traffic from the sources is copied onto the RSPAN VLAN through a reflector port and then forwarded over trunk ports that are carrying the RSPAN VLAN to any RSPAN destination sessions monitoring the RSPAN VLAN

This example shows how to set up a SPAN session, session 1, for monitoring source port traffic to a destination port. First, any existing SPAN configuration for session 1 is cleared, and then bidirectional traffic is mirrored from source port 1 to destination port 10.

```
Switch(config)# no monitor session 1
Switch(config)# monitor session 1 source interface fastEthernet0/1
Switch(config)# monitor session 1 destination interface
FastEthernet0/10 encapsulation dot1q
Switch(config)# end
```

This example shows how to clear any existing configuration on SPAN session 2, configure SPAN session 2 to monitor received traffic on all ports belonging to VLANs 1 through 3, and send it to destination port 7. The configuration is then modified to also monitor received traffic on all ports belonging to VLAN 10.

```
Switch(config)# no monitor session 2
Switch(config)# monitor session 2 source vlan 1 - 3 rx
Switch(config)# monitor session 2 destination interface
gigabitethernet0/7
Switch(config)# monitor session 2 source vlan 10 rx
Switch(config)# end
```

This example shows how to clear any existing configuration on SPAN session 2, configure SPAN session 2 to monitor traffic received on trunk port 4, and send traffic for only VLANs 1 through 5 and 9 to destination port 8.

```
Switch(config)# no monitor session 2
Switch(config)# monitor session 2 source interface gigabitethernet0/4
rx
Switch(config)# monitor session 2 filter vlan 1 - 5 , 9
Switch(config)# monitor session 2 destination interface
gigabitethernet0/8
Switch(config)# end
```

This example shows how to clear any existing RSPAN configuration for session 1, configure RSPAN session 1 to monitor multiple source interfaces, and configure the destination RSPAN VLAN and the reflector-port.

```
Switch(config)# no monitor session 1
```

```
Switch(config)# monitor session 1 source interface fastEthernet0/10 tx
Switch(config)# monitor session 1 source interface fastEthernet0/2 rx
Switch(config)# monitor session 1 source interface fastEthernet0/3 rx
Switch(config)# monitor session 1 source interface port-channel 102 rx
Switch(config)# monitor session 1 destination remote vlan 901
reflector-port fastEthernet0/1
Switch(config)# end
```

Multi-VRF CE (aka VRF-lite)

Multi-VRF CE is a feature that allows a service provider to support two or more VPNs, where IP addresses can be overlapped among the VPNs. Multi-VRF CE uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF. Interfaces in a VRF can be either physical, such as Ethernet ports, or logical, such as VLAN SVIs, but an interface cannot belong to more than one VRF at any time.

Note: Multi-VRF CE interfaces must be Layer 3 interfaces

Multi-VRF CE includes these devices:
Customer edge (CE) devices provide customers access to the service provider network over a data link to one or more provider edge routers. The CE device advertises the site's local routes to the router and learns the remote VPN routes from it. A Catalyst 3550 switch can be a CE.

Provider edge (PE) routers exchange routing information with CE devices by using static routing or a routing protocol such as BGP, RIPv2, OSPF, or EIGRP. The PE is only required to maintain VPN routes for those VPNs to which it is directly attached, eliminating the need for the PE to maintain all of the service provider VPN routes. Each PE router maintains a VRF for each of its directly connected sites. Multiple interfaces on a PE router can be associated with a single VRF if all of these sites participate in the same VPN. Each VPN is mapped to a specified VRF. After learning local VPN routes from CEs, a PE router exchanges VPN routing information with other PE routers by using internal BGP (IBPG).

Provider routers or core routers are any routers in the service provider network that do not attach to CE devices.

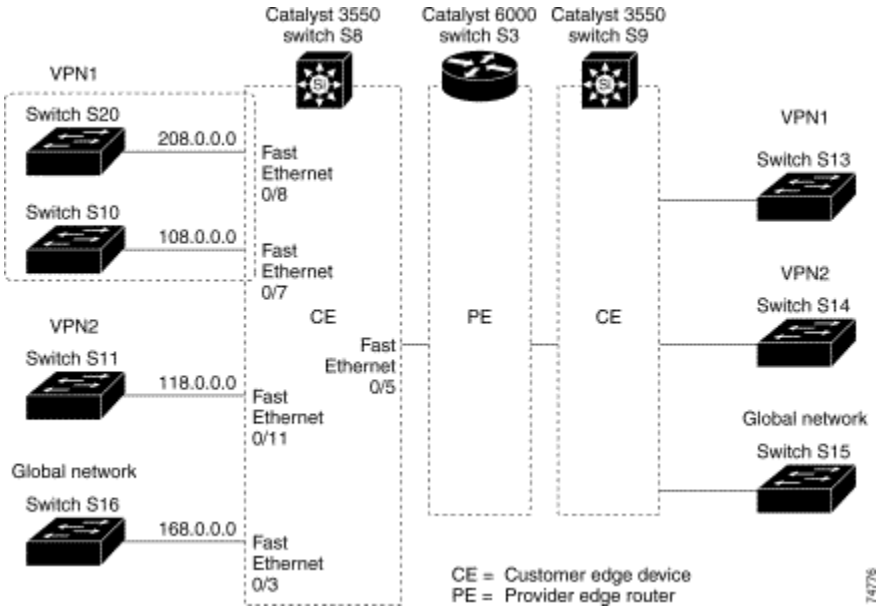
With multi-VRF CE, multiple customers can share one CE, and only one physical link is used between the CE and the PE. The shared CE maintains separate VRF tables for each customer and switches or routes packets for each customer based on its own routing table. Multi-VRF CE extends limited PE functionality to a CE device, giving it the ability to maintain separate VRF tables to extend the privacy and security of a VPN to the branch office.

To support multi-VRF CE, multiple routing tables are entered into the Layer 3 TCAM table. Because an extra field is needed in the routing table to identify the table to which a route belongs, you must modify the SDM template to enable the switch to support 144-bit Layer 3 TCAM. Use the **sdm prefer extended-match**, **sdm prefer access extended-match**, or **sdm prefer routing extended-match** global configuration command to reformat the TCAM space allocated to unicast routing in the default, access, or routing template, respectively. Reformatting the unicast routing TCAM halves the number of supported unicast routes in the template.

OSPF is the protocol used in VPN1, VPN2, and the global network. BGP is used in the CE to PE connections. The example commands show how to configure the CE Switch S8 and include the VRF configuration for Switches S20 and S11 and the PE router commands

related to traffic with Switch S8. Commands for configuring the other switches are not included but would be similar.

Multi-VRF CE Configuration Example



Configuring Switch S8

On Switch S8, enable routing and configure VRF.

```
Switch# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# ip vrf v11
Switch(config-vrf)# rd 800:1
Switch(config-vrf)# route-target export 800:1
Switch(config-vrf)# route-target import 800:1
Switch(config-vrf)# exit

Switch(config)# ip vrf v12
Switch(config-vrf)# rd 800:2
Switch(config-vrf)# route-target export 800:2
Switch(config-vrf)# route-target import 800:2
Switch(config-vrf)# exit
```

Configure the loopback and physical interfaces on Switch S8. Fast Ethernet interface 0/5 is a trunk connection to the PE. Interfaces 0/7 and 0/11 connect to VPNs:

```
Switch(config)# interface loopback1
Switch(config-if)# ip vrf forwarding v11
Switch(config-if)# ip address 8.8.1.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface loopback2
Switch(config-if)# ip vrf forwarding v12
Switch(config-if)# ip address 8.8.2.8 255.255.255.0
```

```
Switch(config-if)# exit


Switch(config)# interface FastEthernet0/5
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface FastEthernet0/8
Switch(config-if)# switchport access vlan 208
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface FastEthernet0/11
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit
```

Configure the VLANs used on Switch S8. VLAN 10 is used by VRF 11 between the CE and the PE. VLAN 20 is used by VRF 12 between the CE and the PE. VLANs 118 and 208 are used for VRF for the VPNs that include Switch S11 and Switch S20, respectively:

```
Switch(config)# interface Vlan10
Switch(config-if)# ip vrf forwarding v11
Switch(config-if)# ip address 38.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface Vlan20
Switch(config-if)# ip vrf forwarding v12
Switch(config-if)# ip address 83.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface Vlan118
Switch(config-if)# ip vrf forwarding v12
Switch(config-if)# ip address 118.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface Vlan208
Switch(config-if)# ip vrf forwarding v11
Switch(config-if)# ip address 208.0.0.8 255.255.255.0
Switch(config-if)# exit
```

Configure OSPF routing in VPN1 and VPN2.

```
Switch(config)# router ospf 1 vrf vl1
Switch(config-router)# redistribute bgp 800 subnets
Switch(config-router)# network 208.0.0.0 0.0.0.255 area 0
Switch(config-router)# exit

Switch(config)# router ospf 2 vrf vl2
Switch(config-router)# redistribute bgp 800 subnets
Switch(config-router)# network 118.0.0.0 0.0.0.255 area 0
Switch(config-router)# exit
```

Configure BGP for CE to PE routing.

```
Switch(config)# router bgp 800
```

```
Switch(config-router)# address-family ipv4 vrf vl2
Switch(config-router-af)# redistribute ospf 2 match internal
Switch(config-router-af)# neighbor 83.0.0.3 remote-as 100
Switch(config-router-af)# neighbor 83.0.0.3 activate
Switch(config-router-af)# network 8.8.2.0 mask 255.255.255.0
Switch(config-router-af)# exit

Switch(config-router)# address-family ipv4 vrf vl1
Switch(config-router-af)# redistribute ospf 1 match internal
Switch(config-router-af)# neighbor 38.0.0.3 remote-as 100
Switch(config-router-af)# neighbor 38.0.0.3 activate
Switch(config-router-af)# network 8.8.1.0 mask 255.255.255.0
Switch(config-router-af)# end
```

Configuring Switch S20

Switch S20 belongs to VPN 1.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing

Switch(config)# interface Fast Ethernet 0/7
Switch(config-if)# no switchport
Switch(config-if)# ip address 208.0.0.20 255.255.255.0
Switch(config-if)# exit

Switch(config)# router ospf 101
Switch(config-router)# network 208.0.0.0 0.0.0.255 area 0
Switch(config-router)# end
```

Configuring Switch S11

Switch S11 belongs to VPN 2.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing

Switch(config)# interface Gigabit Ethernet 0/3
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface Vlan118
Switch(config-if)# ip address 118.0.0.11 255.255.255.0
Switch(config-if)# exit

Switch(config)# router ospf 101
Switch(config-router)# network 118.0.0.0 0.0.0.255 area 0
Switch(config-router)# end
```

Configuring the PE Switch S3

On Switch S3 (the router), these commands only configure the connections to the CE device, Switch S8.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip vrf v1
Router(config-vrf)# rd 100:1
Router(config-vrf)# route-target export 100:1
Router(config-vrf)# route-target import 100:1
Router(config-vrf)# exit

Router(config)# ip vrf v2
Router(config-vrf)# rd 100:2
Router(config-vrf)# route-target export 100:2
Router(config-vrf)# route-target import 100:2
Router(config-vrf)# exit

Router(config)# ip cef

Router(config)# interface Loopback1
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 3.3.1.3 255.255.255.0
Router(config-if)# exit


Router(config)# interface Loopback2
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 3.3.2.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Fast Ethernet3/0.10
Router(config-if)# encapsulation dot1q 10
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 38.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Fast Ethernet3/0.20
Router(config-if)# encapsulation dot1q 20
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 83.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf v2
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.2.0 mask 255.255.255.0
Router(config-router-af)# exit

Router(config-router)# address-family ipv4 vrf vl
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.1.0 mask 255.255.255.0
Router(config-router-af)# end
```

DHCP Option 82 Subscriber Identification

The DHCP is widely used in LAN environments to dynamically assign host IP addresses from a centralized server, which significantly reduces the overhead of administrating IP addresses. The DHCP also helps conserve the limited IP address space because IP addresses no longer need to be permanently assigned to hosts; only those hosts that are connected to the network require IP addresses.

In the residential, metropolitan Ethernet-access environment, the DHCP can centrally manage the IP address assignment for a large number of subscribers. By enabling the DCHP option-82 feature on the switch, a subscriber is identified by the switch port through which it connects to the network (rather than by its MAC address). Multiple hosts on the subscriber LAN can be connected to the same port on the access switch and are uniquely identified.

With the DHCP option-82 feature enabled on the switch, port-to-port DHCP broadcast isolation is achieved when the client ports are within a single VLAN. During client-to-server exchanges, broadcast requests from clients connected to VLAN access ports are intercepted by the relay agent and are not flooded to other clients on the same VLAN. The relay agent forwards the request to the DHCP server. During server-to-client exchanges, the DHCP server sends a broadcast reply that contains the option-82 field. The relay agent uses this information to identify which port connects to the requesting client and avoids forwarding the reply to the entire VLAN.

When you enable the DHCP relay agent option 82 on the switch, these events occur:

The host (DHCP client) generates a DHCP request and broadcasts it on the network.

The switch (DHCP relay agent) intercepts the broadcast DHCP request packet and inserts the relay agent information option (option 82) in the packet. The relay information option contains the switch's MAC address (the remote ID suboption) and the port SNMP ifindex from which the packet is received (circuit ID suboption).

The switch forwards the DHCP request that includes the option-82 field to the DHCP server.

The DHCP server receives the packet. If the server is option-82 capable, it might use the remote ID, the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. Then the DHCP server echoes the option-82 field in the DHCP reply.

If the server does not support option 82, it ignores the option and does not echo it in the reply.

The DHCP server unicasts the reply to the relay agent. The relay agent makes sure that the packet is destined for it by checking the IP destination address in the packet, which is the same as the Layer 3 interface where the **ip helper-address** interface configuration command is configured. The relay agent removes the option-82 field and forwards the packet to the switch port that connects to the DHCP client, which sent the DHCP request.

This example shows how to enable the DHCP server, the relay agent, and the insertion and removal of the DHCP relay information (option 82). It creates a switch virtual interface with VLAN ID 10, assigns it an IP address, and specifies the DHCP packet forwarding address of 30.0.0.2 (DHCP server address). Two interfaces (Gigabit Ethernet 0/1 and 0/2) that connect to the DHCP clients are configured as static access ports in VLAN 10

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# service dhcp
Switch(config)# ip dhcp relay information option
Switch(config)# interface vlan 10
Switch(config-if)# ip address 10.0.0.1 255.0.0.0
```

```
Switch(config-if)# ip helper-address 30.0.0.2
Switch(config-if)# exit

Switch(config)# interface range gigabitethernet0/1 - 2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 10
Switch(config-if)# exit
```

Service Provider Oriented Functions

Business customers of service providers often have specific requirements for VLAN IDs and the number of VLANs to be supported. The VLAN ranges required by different customers in the same service-provider network might overlap, and traffic of customers through the infrastructure might be mixed. Assigning a unique range of VLAN IDs to each customer would restrict customer configurations and could easily exceed the VLAN limit of 4096 of the 802.1Q specification.

Using the 802.1Q tunneling feature, service providers can use a single VLAN to support customers who have multiple VLANs. Customer VLAN IDs are preserved and traffic from different customers is segregated within the service-provider infrastructure even when they appear to be on the same VLAN. The 802.1Q tunneling expands VLAN space by using a VLAN-in-VLAN hierarchy and tagging the tagged packets. A port configured to support 802.1Q tunneling is called a tunnel port. When you configure tunneling, you assign a tunnel port to a VLAN that is dedicated to tunneling. Each customer requires a separate VLAN, but that VLAN supports all of the customer's VLANs.

Customer traffic tagged in the normal way with appropriate VLAN IDs come from an 802.1Q trunk port on the customer device and into a tunnel port on the service-provider edge switch. The link between the customer device and the edge switch is an asymmetric link because one end is configured as an 802.1Q trunk port and the other end is configured as a tunnel port. You assign the tunnel port interface to an access VLAN ID unique to each customer.

This example shows how to configure an interface as a tunnel port, enable tagging of native VLAN packets, and verify the configuration. In this configuration, the VLAN ID for the customer connected to Gigabit Ethernet interface 7 is VLAN 22.

```
Switch(config)# interface gigabitethernet0/7
Switch(config-if)# switchport access vlan 22
% Access VLAN does not exist. Creating vlan 22
Switch(config-if)# switchport mode dot1q-tunnel
Switch(config-if)# exit
Switch(config)# vlan dot1q tag native
Switch(config)# end
```

Layer 2 Protocol Tunneling

Customers at different sites connected across a service-provider network need to run various Layer 2 protocols to scale their topology to include all remote sites, as well as the local sites. STP must run properly, and every VLAN should build a proper spanning tree that includes the local site and all remote sites across the service-provider infrastructure. Cisco Discovery Protocol (CDP) must discover neighboring Cisco devices from local and remote sites. VLAN Trunking Protocol (VTP) must provide consistent VLAN configuration throughout all sites in the customer network.

When protocol tunneling is enabled, edge switches on the inbound side of the service-provider infrastructure encapsulate Layer 2 protocol packets with a special MAC address and send them across the service-provider network. Core switches in the network do not

process these packets, but forward them as normal packets. Layer 2 protocol data units (PDUs) for CDP, STP, or VTP cross the service-provider infrastructure and are delivered to customer switches on the outbound side of the service-provider network. Identical packets are received by all customer ports on the same VLANs with the following results:

Users on each of a customer's sites are able to properly run STP and every VLAN can build a correct spanning tree based on parameters from all sites and not just from the local site.

CDP discovers and shows information about the other Cisco devices connected through the service-provider network.

VTP provides consistent VLAN configuration throughout the customer network, propagating through the service provider to all switches.

Layer 2 protocol tunneling can be used independently or to enhance 802.1Q tunneling. If protocol tunneling is not enabled on 802.1Q tunneling ports, remote switches at the receiving end of the service-provider network do not receive the PDUs and cannot properly run STP, CDP, and VTP. When protocol tunneling *is* enabled, Layer 2 protocols within each customer's network are totally separate from those running within the service-provider network. Customer switches on different sites that send traffic through the service-provider network with 802.1Q tunneling achieve complete knowledge of the customer's VLAN. If 802.1Q tunneling is not used, you can still enable Layer 2 protocol tunneling by connecting to the customer switch through access ports and enabling tunneling on the service-provider access port.

This example shows how to configure Layer 2 protocol tunneling for STP and CDP and verify the configuration.

```
Switch(config)# interface gigabitethernet0/7
Switch(config-if)# l2protocol-tunnel stp
Switch(config-if)# l2protocol-tunnel cdp
Switch(config-if)# l2protocol-tunnel shutdown-threshold 400
Switch(config-if)# exit
Switch(config)# l2protocol-tunnel cos 6
Switch(config)# end
```

Clustering

A switch cluster is a group of connected Catalyst switches that are managed as a single entity. In a switch cluster, 1 switch must be the *command switch* and up to 15 switches can be *member switches*. The total number of switches in a cluster cannot exceed 16 switches. The command switch is the single point of access used to configure, manage, and monitor the member switches. Cluster members can belong to only one cluster at a time.

The benefits of clustering switches include:

Management of Catalyst switches regardless of their interconnection media and their physical locations. The switches can be in the same location, or they can be distributed across a Layer 2 or Layer 3 (if your cluster is using a Catalyst 3550 multilayer switch as a Layer 3 router between the Layer 2 switches in the cluster) network.

Cluster members are connected to the command switch. This section includes management VLAN considerations for the Catalyst 1900, Catalyst 2820, Catalyst 2900 XL, Catalyst 2950, and Catalyst 3500 XL switches. For complete information about these

switches in a switch-cluster environment, refer to the software configuration guide for that specific switch.

Command-switch redundancy if a command switch fails. One or more switches can be designated as *standby command switches* to avoid loss of contact with cluster members. A *cluster standby group* is a group of standby command switches.

Management of a variety of Catalyst switches through a single IP address. This conserves on IP addresses, especially if you have a limited number of them. All communication with the switch cluster is through the command switch IP address.

Clustering is easiest configured via CMS, although it can be completed via CLI.

Links

Complete 3550 documentation
http://www.cisco.com/go/catalyst3550

Catalyst 3550 IOS Documentation (12.1(11)EA1)
http://www.cisco.com/univercd/cc/td/doc/product/lan/c3550/12111ea1/index.htm

Catalyst 3550-12 Powerpoint
http://tools.cisco.com/cmn/jsp/index.jsp?id=16188&redir=YES

Appendix A – Command Reference:

IN-1

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

N

P

Q

R

S

T

U

V

W