Ethernet Virtual Connections (EVCs)

Note

- For complete syntax and usage information for the commands used in this chapter, see these publications:

  http://www.cisco.com/en/US/products/ps11846/prod_command_reference_list.html
- Cisco IOS Release 15.3SY supports only Ethernet interfaces. Cisco IOS Release 15.3SY does not support any WAN features or commands.

Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum

Prerequisites for EVCs

None.

Restrictions for EVCs

- LACP EtherChannels and the 802.1ad provider-bridge mode are mutually exclusive. LACP EtherChannels cannot transmit traffic when the 802.1ad provider-bridge mode is enabled.
- Maximum EFPs per switch: 10K.
- Maximum EFPs per bridge domain: 124.
- Maximum EFPs per interface: 4K.
- Maximum bridge domains per switch: 4K.
- Bridge domain configuration is supported only as part of the EVC service instance configuration.
- EVC support requires the following:
  - The spanning tree mode must be MST.
  - The **dot1ad** global configuration mode command must be configured.
- Service instances can be configured only on ports configured to trunk unconditionally with the **switchport nonegotiate** command.
- You can configure PFC QoS to support EVC ports.
- These are the supported EVC features:
  - Service instances—You create, delete, and modify EFP service instances on Ethernet interfaces.
  - Ethernet service protection on EVCs:

    —Ethernet Operations, Administration, and Maintenance (EOAM)

    —Connectivity fault management (CFM)

    —Ethernet Local Management Interface (E-LMI)
  - IPv6 access control lists (ACLs).
  - Encapsulation—You can map traffic to EFPs based on 802.1Q VLANs (a single VLAN or a list or range of VLANs).
  - You can configure EFPs as members of a bridge domain.
  - Bridge domains support push symmetric only: the supported rewrite configuration implies egress pushing (adding a tag)
  - Bridge domains support ingress rewrite
  - EVC forwarding
  - MAC address learning and aging
  - EVCs on EtherChannels
  - EVC MAC address security
  - Bridging between switchports and EFPs
  - MSTP (MST on EVC bridge domain)
  - EFP statistics (packets and bytes)
  - QoS aware EVC/EFP per service instance
- These Layer 2 port-based features can run with EVC configured on a port:
  - PAGP
  - LACP

  - UDLD
  - LLDP
  - CDP
  - MSTP
- These features are not supported on EVCs:
  - Layer 2 multicast frame flooding
  - Layer 2 protocol tunneling
  - QinQ tagging
  - VLAN Translation
  - EoMPLS
  - Bridge domain routing
  - Split horizon
  - Service instance groups; also called Ethernet flow point (EFP) groups
  - IPv6 access control lists (ACLs)

Information About EVCs

EVC Overview

Ethernet virtual circuits (EVCs) define a Layer 2 bridging architecture that supports Ethernet services. An EVC is defined by the Metro-Ethernet Forum (MEF) as an association between two or more user network interfaces that identifies a point-to-point or multipoint-to-multipoint path within the service provider network. An EVC is a conceptual service pipe within the service provider network. A bridge domain is a local broadcast domain that exists separately from VLANs.

Ethernet Flow Points

An Ethernet flow point (EFP) service instance is a logical interface that connects a bridge domain to a physical port or to an EtherChannel. Configuring a service instance on a Layer 2 port creates a pseudoport or EFP on which you configure EVC features. Each service instance has a unique number per interface, but you can use the same number on different interfaces because service instances on different ports are not related.

An EFP classifies frames from the same physical port to one of the multiple service instances associated with that port, based on user-defined criteria. Each EFP can be associated with different forwarding actions and behavior.

The three major characteristics (or parameters) of an EFP are

- Encapsulation
- Rewrite Information
- Forwarding instance or method (bridge-domain or xconnect)

An EVC broadcast domain is determined by a bridge domain and the EFPs that are attached to it. An incoming frame is matched against EFP matching criteria on the interface, learned on the matching EFP, and forwarded to one or more EFPs in the bridge domain. If there are no matching EFPs, the frame is dropped.

You can use EFPs to configure VLAN translation. For example, if there are two EFPs egressing the same interface, each EFP can have a different VLAN rewrite operation, which is more flexible than the traditional switch port VLAN translation model.

When an EFP is created, the initial state is UP. The state changes to DOWN under the following circumstances:

- The EFP is explicitly shut down by a user.
- The main interface to which the EFP is associated is down or removed.
- If the EFP belongs to a bridge domain, the bridge domain is down.
- The EFP is forced down as an error-prevention measure of certain features.

Service Instances and EFPs

Configuring a service instance on a Layer 2 port or EtherChannel creates a pseudoport or Ethernet flow point (EFP) on which you configure EVC features. Each service instance has a unique number per interface, but you can use the same number on different interfaces because service instances on different ports are not related.

If you have defined an EVC by entering the **ethernet evc** *evc-id* global configuration command, you can associate the EVC with the service instance (optional). There is no default behavior for a service instance. You can configure a service instance only on trunk ports with no allowed VLANs. Any other configuration is not allowed. After you have configured a service instance on an interface, switchport commands are not allowed on the interface. You can also configure a service instance on an EtherChannel group.

Use the **service instance** *number* **ethernet** [*name*] interface configuration command to create an EFP on a Layer 2 interface or EtherChannel and to enter service instance configuration mode. You use service instance configuration mode to configure all management and control date plane attributes and parameters that apply to the service instance on a per-interface basis.

- The **service instance** *number* is the EFP identifier, an integer from 1 to 4000.
- The optional **ethernet** *name* is the name of a previously configured EVC. You do not need to enter an EVC *name*, but you must enter **ethernet**. Different EFPs can share the same name when they correspond to the same EVC. EFPs are tied to a global EVC through the common name.

When you enter service instance configuration mode, you can configure these options:

- **default**—Sets a command to its defaults
- **description**—Adds a service instance specific description
- **encapsulation**—Configures Ethernet frame match criteria
- **errdisable**—Configures error disable
- **ethernet**—Configures Ethernet-lmi parameters
- **exit**—Exits from service instance configuration mode
- **l2protocol**—Configures Layer 2 control protocol processing
- **mac**—Commands for MAC address-based features
- **no**—Negates a command or sets its defaults
- **service-policy**—Attaches a policy-map to an EFP
- **shutdown**—Takes the service instance out of service

Enter the [**no**] **shutdown** service-instance configuration mode to shut down or bring up a service instance.

On a Layer 2 port with no service instance configured, multiple **switchport** commands are available (**access**, **backup**, **block**, **host**, **mode**, and **trunk**). When one or more service instances are configured on a Layer 2 port, no **switchport** commands are accepted on that interface.

Encapsulation (Flexible Service Mapping)

Encapsulation defines the matching criteria that map any of these in any combination to a service instance:

- A VLAN
- A range of VLANs
- The class of service (CoS) bits
- The Ethertype

VLAN tags and CoS can be a single value, a range, or a list. Ethertype can be a single type or a list of types. These are the encapsulation types:

- default
- dot1q
- priority-tagged
- untagged

Priority-tagged frames are always single-tagged. All Ethernet traffic is supported. The encapsulation classification options are:

- inner tag CoS
- inner tag VLAN

When you configure an encapsulation method, you enable flexible service mapping, which allows you to map an incoming packet to an EFP based on the configured encapsulation.

The default behavior for flexible service mapping based on the outer 802.1q VLAN tag value is nonexact, meaning that when the EFP encapsulation configuration does not explicitly specify an inner (second) VLAN tag matching criterion, the software maps both single-tagged and double-tagged frames to the EFP as long as the frames fulfill the criteria of outer VLAN tag values. The command-line interface (CLI) does allow you to specify exact mapping with the **exact** keyword. If this keyword is specified, the EFP is designated as single-tagged-frame-only and double-tagged frames are not classified to that EFP.

Using the CLI **encapsulation** command in service-instance configuration mode, you can set encapsulation criteria. You must configure one encapsulation command per EFP (service instance). After you have configured an encapsulation method, these commands are available in service instance configuration mode:

- **bridge-domain**—Configures a bridge domain.
- **rewrite**—Configures Ethernet rewrite criteria.

Table 41-1 Supported Encapsulation Types

Command	Description
encapsulation dot1q {**any** | *vlan-id* [**,***vlan-id* [**-***vlan-id*]]}	Defines the matching criteria to be used to map 802.1q frames ingressing on an interface to the appropriate EFP. The options are a single VLAN, a range of VLANs, or lists of VLANs or VLAN ranges. VLAN IDs are 1 to 4094. • Enter the **any** keyword to match or all VLANS (1-4094) • Enter a single VLAN ID for an exact match of the outermost tag. • Enter a VLAN range for a ranged outermost match.
encapsulation dot1q *vlan-id* **cos** *cos-value*	CoS value encapsulation defines match criteria after including the CoS for the C-Tag. The CoS value is a single digit between 1 and 7. You cannot configure CoS encapsulation with the **encapsulation untagged** command, but you can configure it with the **encapsulation priority-tagged** command. The result is an exact outermost VLAN and CoS match. You can also use VLAN ranges.
encapsulation untagged	Matching criteria to be used to map untagged Ethernet frames entering an interface to the appropriate EFP. Only one EFP per port can have untagged encapsulation. However, a port that hosts EFP matching untagged traffic can also host other EFPs that match tagged frames. **Note** Not supported with the **encapsulation priority-tagged** command.

Table 41-1 Supported Encapsulation Types (continued)

Command	Description
encapsulation priority-tagged	Specifies priority-tagged frames. A priority-tagged packet has VLAN ID 0 and a CoS value of 0 to 7. **Note** Not supported with the **encapsulation untagged** command.
encapsulation default	Configures the default EFP on a port, which matches all otherwise unmatched packets. If the default EFP is the only one configured on a port, it matches all ingress frames on that port. If you configure the default EFP on a port, you cannot configure any other EFP on the same port with the same bridge domain.

If a packet entering a port does not match any of the encapsulations on that port, the packet is dropped, resulting in filtering of the packet. The encapsulation must match the packet on the wire to determine filtering criteria. On the wire refers to packets ingressing the switch before any rewrites and to packets egressing the switch after all rewrites.

EFPs and MSTP

EFP bridge domains are supported by the Multiple Spanning Tree Protocol (MSTP). These restrictions apply when running STP with bridge domains.

- All incoming VLANs (outer-most or single) mapped to a bridge domain must belong to the same MST instance or loops could occur.
- For all EFPs that are mapped to the same MST instance, you must configure backup EFPs on every redundant path to prevent loss of connectivity due to STP blocking a port.
- When STP mode is PVST+ or PVRST, EFP information is not passed to the protocol. EVC only supports only MSTP.
- Changing STP mode from MST to PVST+ or PVRST for a multicast port is not allowed.

Bridge Domains

Bridge Domain Overview

A bridge domain defines a broadcast domain internal to a platform and allows the decoupling of a broadcast domain from a VLAN. This decoupling enables per-port VLAN significance, thus removing the scalability limitations associated with a single per-device VLAN ID space. Frames received from one of the EFPs participating in a bridge domain matches are bridged.

A service instance must be attached to a bridge domain. Flooding and communication behavior of a bridge domain is similar to that of a VLAN domain. Bridge-domain membership is determined by which service instances have joined it (based on encapsulation criteria), while VLAN domain membership is determined by the VLAN tag in the packet.

Note You must configure encapsulation before you can configure the bridge domain.

IGMP snooping is enabled by default on the switch and on all VLANs but is automatically disabled on a VLAN when you configure a bridge domain under 4094. The switches support up to 124 bridge domains.

Ethernet MAC Address Learning

MAC address learning is always enabled and cannot be disabled.

Flooding of Layer 2 Frames for Unknown MAC and Broadcast Addresses

A Layer 2 frame with an unknown unicast or broadcast destination MAC address is flooded to all the EFPs in the bridge domain except to the originating EFP.

Replication of frames involves recirculating the frame several times. Recirculation negatively affect forwarding performance and reduce the packet forwarding rate for all features.

Layer 2 Destination MAC Address-Based Forwarding

When bridging is configured, a unicast frame received from an EFP is forwarded based on the destination Layer 2 MAC address. If the destination address is known, the frame is forwarded only to the EFP/NNI associated with the destination address.

Because the bridge and EFP configurations are interrelated, bridging is supported only on EFPs. To support multiple bridge domains, MAC address entries are associated with the bridge domain of the EFP. Only unicast MAC addresses need to be dynamically learned.

The EVC infrastructure does not modify frame contents.

MAC Address Aging

The dynamically learned MAC address entries in the MAC table are periodically aged out and entries that are inactive for longer than the configured time period are removed from the table. The supported range of aging-time values, in seconds, is 5 through 1000000, with a granularity of 1. The default is 8 minutes. The **aging-time** parameter can be configured per bridge domain and is a relative value. The value is the aging time relative to the time a frame was received with that MAC address.

MAC Address Table

The MAC address table is used to forward frames based on Layer 2 destination MAC addresses. The table consists of static MAC addresses downloaded from the route processor (RP) and the MAC addresses dynamically learned by the data path.

While the MAC Learning feature is enabled, an entry is added to the MAC table when a new unique MAC address is learned on the data path and an entry is deleted from the table when it is aged out.

Rewrite Operations

The **rewrite** command pushes the 802.1ad tag onto ingress packets to forward the packet on the 802.1ad cloud.

Enter the **rewrite ingress tag push dot1ad** *vlan-id* **symmetric** service-instance configuration mode command to specify the encapsulation of additional dot1ad tag on the frame ingress to the EFP.

Note The **symmetric** keyword is required to complete rewrite to configuration.

When you enter the **symmetric** keyword, the egress counterpart performs the inverse action and pushes (adds) the encapsulation VLAN.

Layer 3 and Layer 4 ACL Support

Configuring an ACL on an EFP is the same as configuring an ACL on other types of interfaces.

Note ACLs are not supported for packets prefixed with a Multiprotocol Label Switching (MPLS) header, including when an MPLS packet contains either Layer 3 or Layer 4 headers of supported protocols.

Advanced Frame Manipulation

The Advanced Frame Manipulation feature supports a PUSH operation that adds one VLAN tag to both the incoming and outgoing frames of an EFP.

When a VLAN tag exists and a new one is added, the CoS field of the new tag is set to the same value as the CoS field of the existing VLAN tag; otherwise, the CoS field is set to a default of 0. Using QoS marking configuration commands, you can change the CoS marking.

Egress Frame Filtering

Egress frame filtering is performed to ensure that frames exiting an EFP contain a Layer 2 header that matches the encapsulation characteristics associated with the EFP. This filtering is done primarily to prevent unintended frame leaks and is always enabled on EFPs.

Default Settings for EVCs

None.

How to Configure EVCs

Configuring a service instance on a Layer 2 port creates an EFP on which you can configure EVC features. Perform this task to configure an EFP.

	Command or Action	Purpose
Step 1	`Router> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# dot1ad`	Enables 802.1ad provider-bridge mode. **Note** LACP EtherChannels and the 802.1ad provider-bridge mode are mutually exclusive. LACP EtherChannels cannot transmit traffic when the 802.1ad provider-bridge mode is enabled.
Step 4	`Router(config)# interface` *type number*	Enters interface configuration mode.
Step 5	`Router(config-if)# switchport`	Configures the port for Layer 2 switching. **Note** You must enter the **switchport** command once without any keywords to configure the LAN port as a Layer 2 port before you can enter additional **switchport** commands with keywords.
Step 6	`Router(config-if)# switchport mode trunk`	Configures the port to trunk unconditionally.
Step 7	`Router(config-if)# switchport nonegotiate`	Configures the trunk not to use DTP.
Step 8	`Router(config-if)# switchport trunk encapsulation dot1q`	Configures the trunk encapsulation as 802.1Q.
Step 9	`Router(config-if)# switchport trunk allowed vlan` *vlan* [,*vlan*[,*vlan*[,...]]	Configures the list of VLANs allowed on the trunk. **Note** If VLAN locking is enabled, enter VLAN names instead of VLAN numbers. For more information, see the "VLAN Locking" section on page 26-4.
Step 10	`Router(config-if)# dot1ad uni`	Configures the port as an 802.1ad provider-bridge user-to-network interface (UNI) port. **Note** The **dot1ad uni** interface mode command imposes some SPAN restrictions (see "Feature Incompatibilities" section on page 55-2).
Step 11	`Router(config-if)# no cdp enable`	Disables CPD on the port.
Step 12	`Router(config-if)# no lldp transmit`	(Required on PE ports) Disables LLDP.
Step 13	`Router(config-if)# spanning-tree bpdufilter enable`	Enables BPDU filtering on the port.
Step 14	`Router(config-if)# service instance` *number* `ethernet` [*name*]	Configures an Ethernet service instance (EFP) and enters service instance configuration mode. • The *number* is the EFP identifier, an integer from 1 through 4000. • (Optional) **ethernet** *name* is the name of a previously configured EVC. You do not need to use an EVC name in a service instance.

	Command or Action	Purpose
Step 15	`Router(config-if)# ip access-group` *access-list-number* | *access-list-name*} {**in** | **out**}	(Optional) Applies an IP access list or object group access control list (OGACL) to an interface.
Step 16	`Router(config-if-srv)# encapsulation` *encapsulation-type vlan-id* [**cos** *cos_value*]	Configures the encapsulation type for the service instance. • **default**—Configures matching for all otherwise unmatched packets. • **dot1q**—Configures 802.1Q encapsulation. See Table 41-1 for more information. • **priority-tagged**—Specify priority-tagged frames, VLAN-ID 0 and CoS value of 0 to 7. • **untagged**—Map to untagged VLANs. Only one EFP per port can have untagged encapsulation. • The CoS value defines match criterion, an integer from 1 through 7.
Step 17	`Router(config-if-srv)# rewrite ingress tag push dot1ad` *vlan-id* [**symmetric**]	(Optional) Specifies the encapsulation adjustment to be performed on a frame ingressing a service instance.
Step 18	`Router(config-if-srv)# bridge-domain` *bridge-id*	Configures the bridge domain.
Step 19	`Router(config-if-srv)# end`	Returns to privileged EXEC mode.

Configuring Multiple Service Instances

```
Router(config)# interface gigabitethernet1/1
Router(config-if)# switchport mode trunk
Router(config-if)# switchport nonegotiate
Router(config-if)# service instance 1 ethernet evc1
Router(config-if-srv)# encapsulation dot1q 201 cos 1
Router(config-if-srv)# rewrite ingress tag push dot1ad 300 symmetric
Router(config-if-srv)# bridge-domain 300
Router(config-if-srv)# end
Router(config-if)# service instance 2 ethernet evc2
Router(config-if-srv)# encapsulation default
Router(config-if-srv)# rewrite ingress tag push dot1ad 301 symmetric
Router(config-if-srv)# bridge-domain 301
Router(config-if-srv)# end
Router(config-if)# service instance 3 ethernet evc3
Router(config-if-srv)# encapsulation priority-tagged cos 1
Router(config-if-srv)# rewrite ingress tag push dot1ad 302 symmetric
Router(config-if-srv)# bridge-domain 302
```

Configuring a Service Instance

```
Router(config)# interface gigabitethernet1/1
Router(config-if)# switchport mode trunk
Router(config-if)# switchport nonegotiate
Router(config-if)# switchport trunk allowed vlan none
Router(config-if)# service instance 22 ethernet evc1
Router(config-if-srv)# encapsulation dot1q 100
Router(config-if-srv)# rewrite ingress tag push dot1ad 10 symmetric
Router(config-if-srv)# bridge-domain 10
```

Encapsulation Using a VLAN Range

```
Router(config)# interface gigabitethernet1/1
Router(config-if)# service instance 1 ethernet
Router(config-if-srv)# encapsulation dot1q 22-44 cos 1
```

```
Router(config-if-srv)# rewrite ingress tag push dot1ad 10 symmetric
Router(config-if-srv)# bridge-domain 10
```

Two Service Instances Joining the Same Bridge Domain

In this example, service instance 1 on interfaces Gigabit Ethernet 1/1 and 1/2 can bridge between each other.

```
Router(config)# interface gigabitethernet1/1
Router(config-if)# service instance 1 Ethernet
Router(config-if-srv)# encapsulation dot1q 10
Router(config-if-srv)# rewrite ingress tag push dot1ad 10 symmetric
Router(config-if-srv)# bridge-domain 10

Router(config)# interface gigabitethernet1/2
Router(config-if)# service instance 1 Ethernet
Router(config-if-srv)# encapsulation dot1q 10
Router(config-if-srv)# rewrite ingress tag push dot1ad 10 symmetric
Router(config-if-srv)# bridge-domain 10
```

Bridge Domains and VLAN Encapsulation

Use the VLAN ID configured with the **rewrite ingress tag push dot1ad** command as the bridge-domain number, rather than the VLAN ID configured with the **encapsulation dot1q** command, which can be the same or a different value.

```
Router(config)# interface gigabitethernet1/1
Router(config-if)# service instance 1 ethernet evc1
Router(config-if-srv)# encapsulation dot1q 10
Router(config-if-srv)# rewrite ingress tag push dot1ad 4000 symmetric
Router(config-if-srv)# bridge-domain 4000

Router(config)# interface gigabitethernet1/2
Router(config-if)# service instance 1 ethernet evc1
Router(config-if-srv)# encapsulation dot1q 10
Router(config-if-srv)# rewrite ingress tag push dot1ad 4000 symmetric
Router(config-if-srv)# bridge-domain 4000
```

Traffic cannot be forwarded if the the VLAN ID configured with the **encapsulation dot1q** commands do not match in a bridge domain. In this example, the service instances on Gigabit Ethernet 1/1 and 1/2 cannot forward between each other, because the encapsulation VLAN IDs do not match (filtering criteria). You can use the **rewrite** command to allow communication between these two.

```
Router(config)# interface gigabitethernet1/1
Router(config-if)# service instance 1 ethernet evc1
Router(config-if-srv)# encapsulation dot1q 10
Router(config-if-srv)# rewrite ingress tag push dot1ad 4000 symmetric
Router(config-if-srv)# bridge-domain 4000

Router(config)# interface gigabitethernet1/2
Router(config-if)# service instance 1 ethernet evc1
Router(config-if-srv)# encapsulation dot1q 99
Router(config-if-srv)# rewrite ingress tag push dot1ad 4000 symmetric
Router(config-if-srv)# bridge-domain 4000
```

Rewrite

In this example, the VLAN ID configured in the **rewrite ingress tag push dot1ad** command (4000 in the example) is pushed onto packets that match the VLAN ID configured in the **encapsulation dot1q** command (10 in the example). The symmetric keyword enables the inverse action on packets in the reverse direction: packets that egress from this service instance with VLAN ID 4000 are deencapsulated, resulting in VLAN ID 10 with cos 1.

```
Router(config)# interface gigabitethernet1/1
Router(config-if)# service instance 1 ethernet
Router(config-if-srv)# encapsulation dot1q 10 cos 1
Router(config-if-srv)# rewrite ingress tag push dot1ad 4000 symmetric
Router(config-if-srv)# bridge-domain 4000
```

Monitoring EVCs

Table 41-2 Supported show Commands

Command	Description
show ethernet service evc [**id** *evc-id* | **interface** *interface-id*] [**detail**]	Displays information about all EVCs, or a specific EVC when you enter an EVC ID, or all EVCs on an interface when you enter an interface ID. The **detail** option provides additional information about the EVC.
show ethernet service instance [**id** *instance-id* **interface** *interface-id* | **interface** *interface-id*] {[**detail**] | [**stats**]}	Displays information about one or more service instance (EFPs). If you specify an EFP ID and interface, only data pertaining to that particular EFP is displayed. If you specify only an interface ID, data is displayed for all EFPs on the interface.
show bridge-domain [*n*]	When you enter *n*, this command displays all the members of the specified bridge-domain, if a bridge-domain with the specified number exists. If you do not enter *n,* the command displays all the members of all bridge-domains in the system.
show ethernet service instance detail	This command displays detailed service instance information, including Layer 2 protocol information. This is an example of the output: **Router# show ethernet service instance detail** Service Instance ID: 2 Associated Interface: GigabitEthernet7/2 Associated EVC: evc2 L2protocol drop CE-Vlans: Encapsulation: dot1q 2 vlan protocol type 0x8100 Rewrite: ingress tag push dot1ad 2 vlan-type 0x88A8 symmetric Interface Dot1q Tunnel Ethertype: 0x8100 State: Up EFP Statistics: Pkts In Bytes In Pkts Out Bytes Out 0 0 0 0
show mac address-table	This command displays dynamically learned or statically configured MAC security addresses.

Tip For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html

Participate in the Technical Documentation Ideas forum