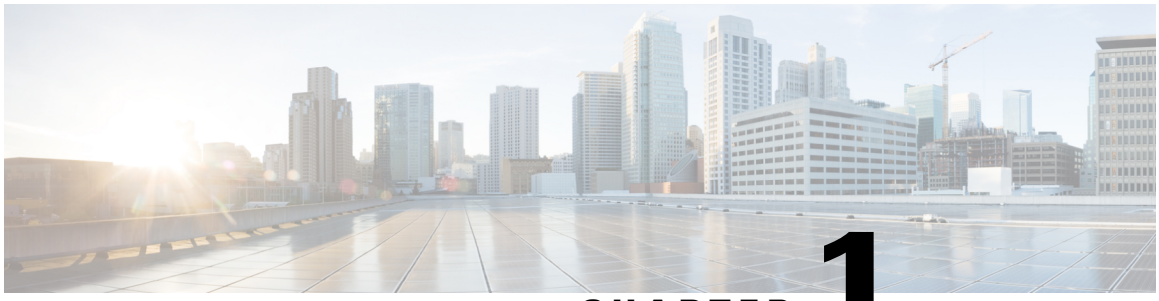
LAN Switching Configuration Guide, Cisco IOS Release 15S

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Configuring Routing Between VLANs

This module provides an overview of VLANs. It describes the encapsulation protocols used for routing between VLANs and provides some basic information about designing VLANs. This module contains tasks for configuring routing between VLANS.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Routing Between VLANs

Virtual Local Area Network Definition

A virtual local area network (VLAN) is a switched network that is logically segmented on an organizational basis, by functions, project teams, or applications rather than on a physical or geographical basis. For example, all workstations and servers used by a particular workgroup team can be connected to the same VLAN, regardless of their physical connections to the network or the fact that they might be intermingled with other

teams. Reconfiguration of the network can be done through software rather than by physically unplugging and moving devices or wires.

A VLAN can be thought of as a broadcast domain that exists within a defined set of switches. A VLAN consists of a number of end systems, either hosts or network equipment (such as bridges and routers), connected by a single bridging domain. The bridging domain is supported on various pieces of network equipment; for example, LAN switches that operate bridging protocols between them with a separate bridge group for each VLAN.

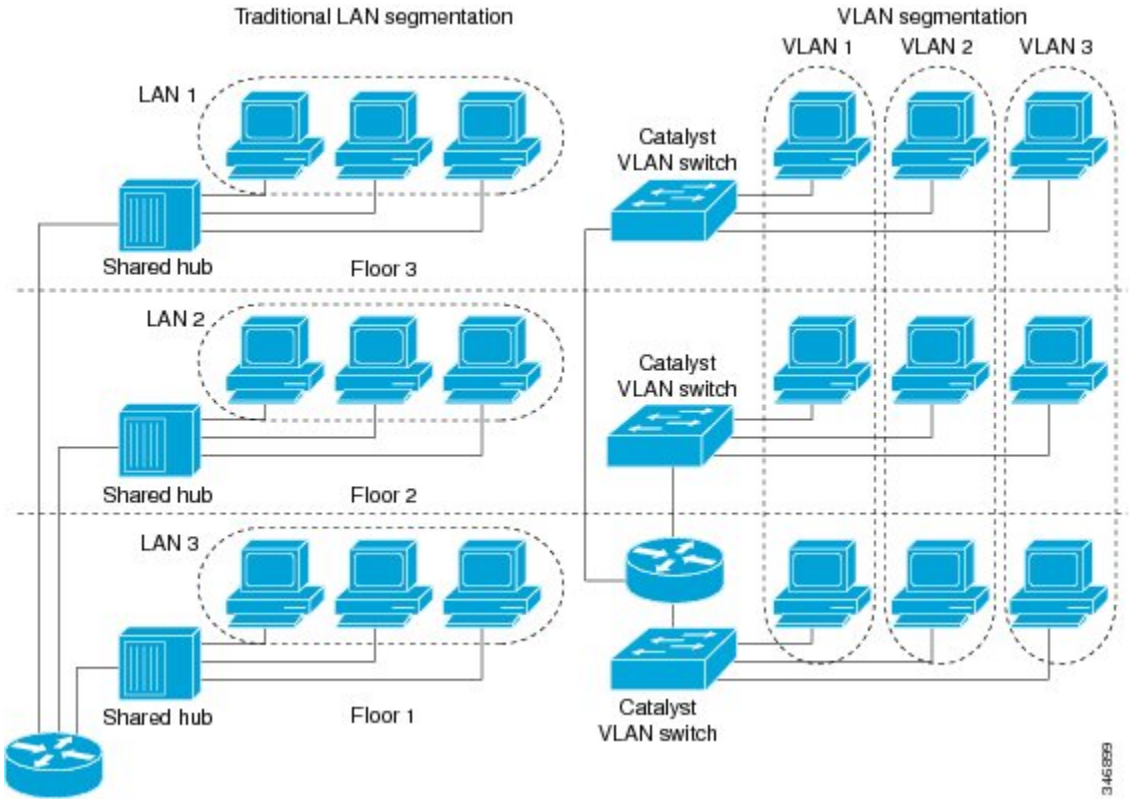
VLANs are created to provide the segmentation services traditionally provided by routers in LAN configurations. VLANs address scalability, security, and network management. Routers in VLAN topologies provide broadcast filtering, security, address summarization, and traffic flow management. None of the switches within the defined group will bridge any frames, not even broadcast frames, between two VLANs. Several key issues described in the following sections need to be considered when designing and building switched LAN internetworks:

LAN Segmentation

VLANs allow logical network topologies to overlay the physical switched infrastructure such that any arbitrary collection of LAN ports can be combined into an autonomous user group or community of interest. The technology logically segments the network into separate Layer 2 broadcast domains whereby packets are switched between ports designated to be within the same VLAN. By containing traffic originating on a particular LAN only to other LANs in the same VLAN, switched virtual networks avoid wasting bandwidth, a drawback inherent to traditional bridged and switched networks in which packets are often forwarded to LANs with no need for them. Implementation of VLANs also improves scalability, particularly in LAN environments that support broadcast- or multicast-intensive protocols and applications that flood packets throughout the network.

The figure below illustrates the difference between traditional physical LAN segmentation and logical VLAN segmentation.

Figure 1: LAN Segmentation and VLAN Segmentation



Security

VLANs improve security by isolating groups. High-security users can be grouped into a VLAN, possibly on the same physical segment, and no users outside that VLAN can communicate with them.

Broadcast Control

Just as switches isolate collision domains for attached hosts and only forward appropriate traffic out a particular port, VLANs provide complete isolation between VLANs. A VLAN is a bridging domain, and all broadcast and multicast traffic is contained within it.

VLAN Performance

The logical grouping of users allows an accounting group to make intensive use of a networked accounting system assigned to a VLAN that contains just that accounting group and its servers. That group's work will not affect other users. The VLAN configuration improves general network performance by not slowing down other users sharing the network.

Network Management

The logical grouping of users allows easier network management. It is not necessary to pull cables to move a user from one network to another. Adds, moves, and changes are achieved by configuring a port into the appropriate VLAN.

Network Monitoring Using SNMP

SNMP support has been added to provide mib-2 interfaces sparse table support for Fast Ethernet subinterfaces. Monitor your VLAN subinterface using the **show vlans** EXEC command. For more information on configuring SNMP on your Cisco network device or enabling an SNMP agent for remote access, see the "Configuring SNMP Support" module in the *Cisco IOS Network Management Configuration Guide* .

Communication Between VLANs

Communication between VLANs is accomplished through routing, and the traditional security and filtering functions of the router can be used. Cisco IOS software provides network services such as security filtering, quality of service (QoS), and accounting on a per-VLAN basis. As switched networks evolve to distributed VLANs, Cisco IOS software provides key inter-VLAN communications and allows the network to scale.

Before Cisco IOS Release 12.2, Cisco IOS support for interfaces that have 802.1Q encapsulation configured is IP, IP multicast, and IPX routing between respective VLANs represented as subinterfaces on a link. New functionality has been added in IEEE 802.1Q support for bridging on those interfaces and the capability to configure and use integrated routing and bridging (IRB).

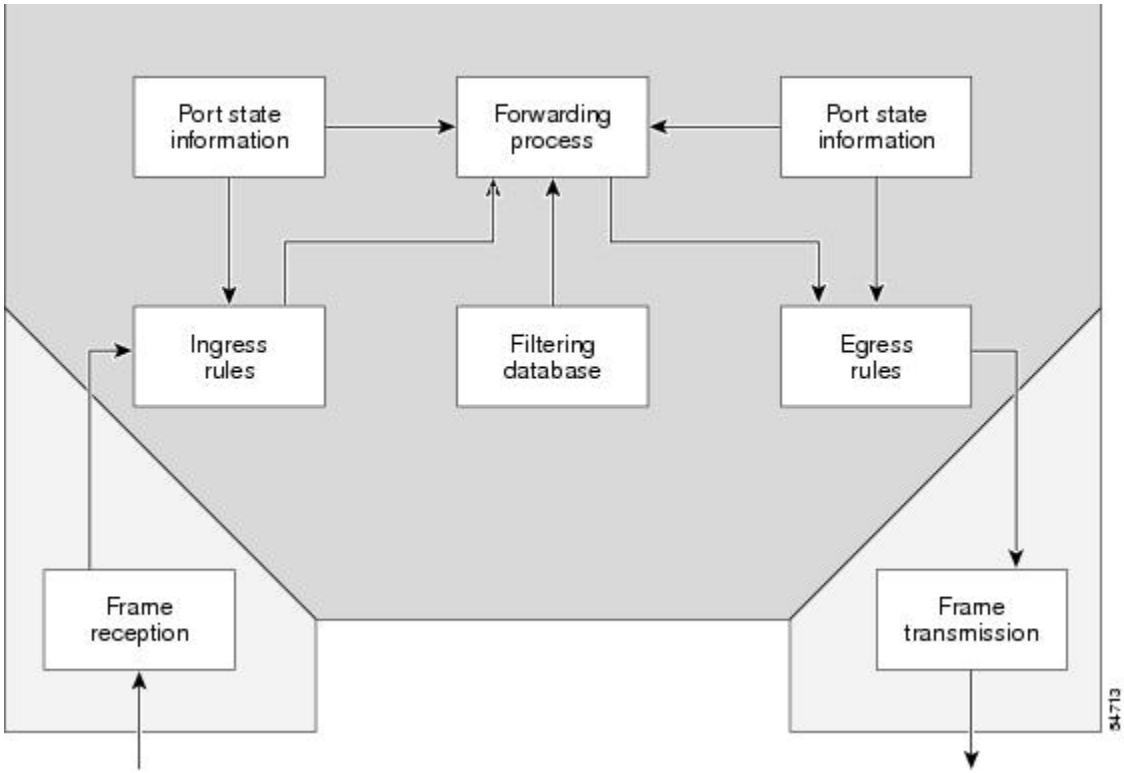
Relaying Function

The relaying function level, as displayed in the figure below, is the lowest level in the architectural model described in the IEEE 802.1Q standard and presents three types of rules:

- Ingress rules--Rules relevant to the classification of received frames belonging to a VLAN.
- Forwarding rules between ports--Rules decide whether to filter or forward the frame.

- Egress rules (output of frames from the switch)--Rules decide if the frame must be sent tagged or untagged.

Figure 2: Relaying Function

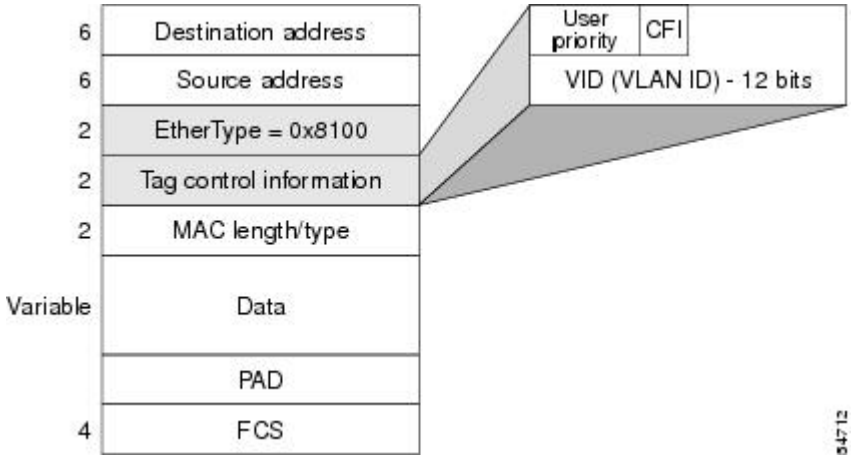


The Tagging Scheme

The figure below shows the tagging scheme proposed by the 802.3ac standard, that is, the addition of the four octets after the source MAC address. Their presence is indicated by a particular value of the EtherType field (called TPID), which has been fixed to be equal to 0x8100. When a frame has the EtherType equal to 0x8100, this frame carries the tag IEEE 802.1Q/802.1p. The tag is stored in the following two octets and it contains 3 bits of user priority, 1 bit of Canonical Format Identifier (CFI), and 12 bits of VLAN ID (VID). The 3 bits of user priority are used by the 802.1p standard; the CFI is used for compatibility reasons between Ethernet-type networks and Token Ring-type networks. The VID is the identification of the VLAN, which is basically used by the 802.1Q standard; being on 12 bits, it allows the identification of 4096 VLANs.

After the two octets of TPID and the two octets of the Tag Control Information field there are two octets that originally would have been located after the Source Address field where there is the TPID. They contain either the MAC length in the case of IEEE 802.3 or the EtherType in the case of Ethernet version 2.

Figure 3: Tagging Scheme

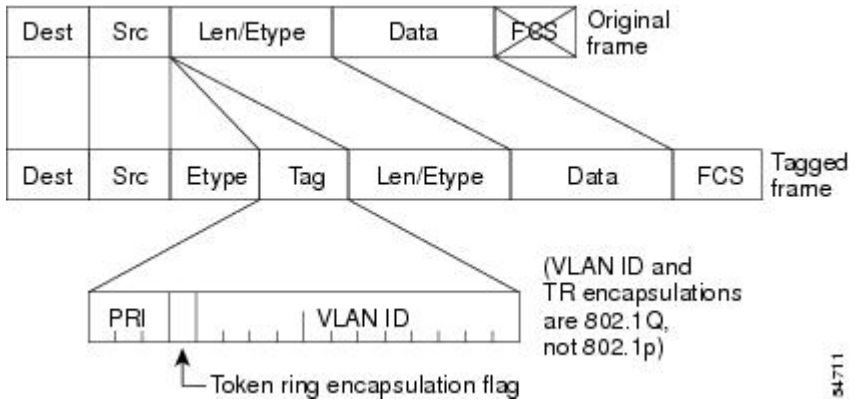


The EtherType and VLAN ID are inserted after the MAC source address, but before the original Ethertype/Length or Logical Link Control (LLC). The 1-bit CFI included a T-R Encapsulation bit so that Token Ring frames can be carried across Ethernet backbones without using 802.1H translation.

Frame Control Sequence Recomputation

The figure below shows how adding a tag in a frame recomputes the Frame Control Sequence. 802.1p and 802.1Q share the same tag.

Figure 4: Adding a Tag Recomputes the Frame Control Sequence

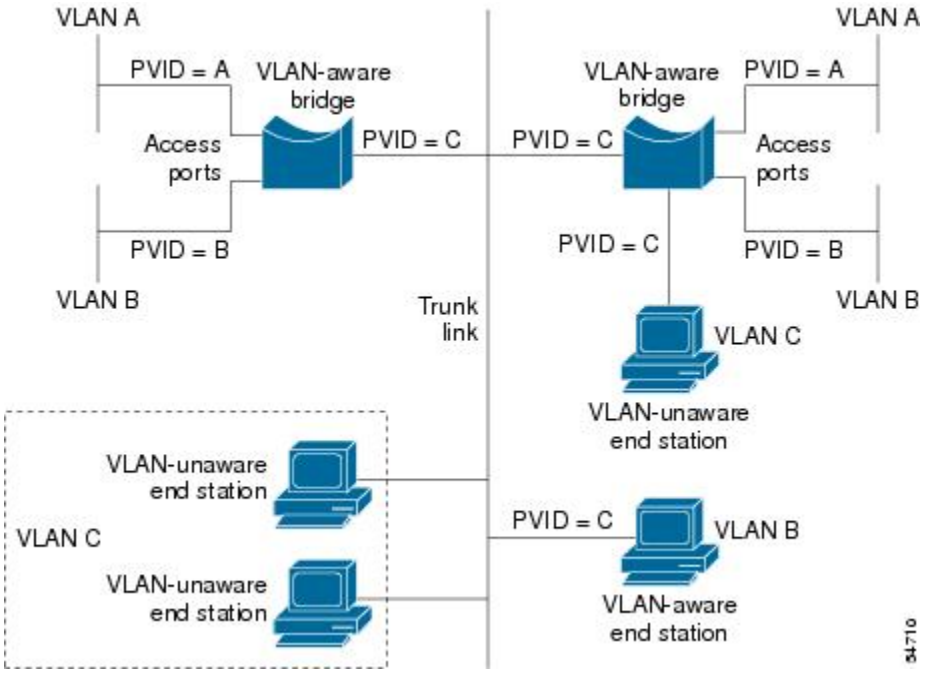


Native VLAN

Each physical port has a parameter called PVID. Every 802.1Q port is assigned a PVID value that is of its native VLAN ID (default is VLAN 1). All untagged frames are assigned to the LAN specified in the PVID

parameter. When a tagged frame is received by a port, the tag is respected. If the frame is untagged, the value contained in the PVID is considered as a tag. Because the frame is untagged and the PVID is tagged to allow the coexistence, as shown in the figure below, on the same pieces of cable of VLAN-aware bridge/stations and of VLAN-unaware bridges/stations. Consider, for example, the two stations connected to the central trunk link in the lower part of the figure below. They are VLAN-unaware and they will be associated to the VLAN C, because the PVIDs of the VLAN-aware bridges are equal to VLAN C. Because the VLAN-unaware stations will send only untagged frames, when the VLAN-aware bridge devices receive these untagged frames they will assign them to VLAN C.

Figure 5: Native VLAN



PVST+

PVST+ provides support for 802.1Q trunks and the mapping of multiple spanning trees to the single spanning tree of 802.1Q switches.

The PVST+ architecture distinguishes three types of regions:

- A PVST region
- A PVST+ region
- A MST region

Each region consists of a homogenous type of switch. A PVST region can be connected to a PVST+ region by connecting two ISL ports. Similarly, a PVST+ region can be connected to an MST region by connecting two 802.1Q ports.

At the boundary between a PVST region and a PVST+ region the mapping of spanning trees is one-to-one. At the boundary between a MST region and a PVST+ region, the ST in the MST region maps to one PVST

in the PVST+ region. The one it maps to is called the common spanning tree (CST). The default CST is the PVST of VLAN 1 (Native VLAN).

All PVSTs, except for the CST, are tunneled through the MST region. Tunneling means that bridge protocol data units (BPDUs) are flooded through the MST region along the single spanning tree present in the MST region.

Ingress and Egress Rules

The BPDU transmission on the 802.1Q port of a PVST+ router will be implemented in compliance with the following rules:

- The CST BPDU (of VLAN 1, by default) is sent to the IEEE address.
- All the other BPDUs are sent to Shared Spanning Tree Protocol (SSTP)-Address and encapsulated with Logical Link Control-Subnetwork Access Protocol (LLC-SNAP) header.
- The BPDU of the CST and BPDU of the VLAN equal to the PVID of the 802.1Q trunk are sent untagged.
- All other BPDUs are sent tagged with the VLAN ID.
- The CST BPDU is also sent to the SSTP address.
- Each SSTP-addressed BPDU is also tailed by a Tag-Length-Value for the PVID checking.

The BPDU reception on the 802.1Q port of a PVST+ router will follow these rules:

- All untagged IEEE addressed BPDUs must be received on the PVID of the 802.1Q port.
- The IEEE addressed BPDUs whose VLAN ID matches the Native VLAN are processed by CST.
- All the other IEEE addressed BPDUs whose VLAN ID does not match the Native VLAN and whose port type is not of 802.1Q are processed by the spanning tree of that particular VLAN ID.
- The SSTP addressed BPDU whose VLAN ID is not equal to the TLV are dropped and the ports are blocked for inconsistency.
- All the other SSTP addressed BPDUs whose VLAN ID is not equal to the Native VLAN are processed by the spanning tree of that particular VLAN ID.
- The SSTP addressed BPDUs whose VLAN ID is equal to the Native VLAN are dropped. It is used for consistency checking.

Integrated Routing and Bridging

IRB enables a user to route a given protocol between routed interfaces and bridge groups or route a given protocol between the bridge groups. Integrated routing and bridging is supported on the following protocols:

- IP
- IPX
- AppleTalk

VLAN Colors

VLAN switching is accomplished through *frame tagging* where traffic originating and contained within a particular virtual topology carries a unique VLAN ID as it traverses a common backbone or trunk link. The VLAN ID enables VLAN switching devices to make intelligent forwarding decisions based on the embedded VLAN ID. Each VLAN is differentiated by a *color* , or VLAN identifier. The unique VLAN ID determines the *frame coloring* for the VLAN. Packets originating and contained within a particular VLAN carry the identifier that uniquely defines that VLAN (by the VLAN ID).

The VLAN ID allows VLAN switches and routers to selectively forward packets to ports with the same VLAN ID. The switch that receives the frame from the source station inserts the VLAN ID and the packet is switched onto the shared backbone network. When the frame exits the switched LAN, a switch strips the header and forwards the frame to interfaces that match the VLAN color. If you are using a Cisco network management product such as VlanDirector, you can actually color code the VLANs and monitor VLAN graphically.

Implementing VLANS

Network managers can logically group networks that span all major topologies, including high-speed technologies such as, ATM, FDDI, and Fast Ethernet. By creating virtual LANs, system and network administrators can control traffic patterns and react quickly to relocations and keep up with constant changes in the network due to moving requirements and node relocation just by changing the VLAN member list in the router configuration. They can add, remove, or move devices or make other changes to network configuration using software to make the changes.

Issues regarding creating VLANs should have been addressed when you developed your network design. Issues to consider include the following:

- Scalability
- Performance improvements
- Security
- Network additions, moves, and changes

Communication Between VLANs

Cisco IOS software provides full-feature routing at Layer 3 and translation at Layer 2 between VLANs. Five different protocols are available for routing between VLANs:

All five of these technologies are based on OSI Layer 2 bridge multiplexing mechanisms.

Inter-Switch Link Protocol

The Inter-Switch Link (ISL) protocol is used to interconnect two VLAN-capable Ethernet, Fast Ethernet, or Gigabit Ethernet devices, such as the Catalyst 3000 or 5000 switches and Cisco 7500 routers. The ISL protocol is a packet-tagging protocol that contains a standard Ethernet frame and the VLAN information associated with that frame. The packets on the ISL link contain a standard Ethernet, FDDI, or Token Ring frame and the VLAN information associated with that frame. ISL is currently supported only over Fast Ethernet links, but a single ISL link, or trunk, can carry different protocols from multiple VLANs.

Procedures for configuring ISL and Token Ring ISL (TRISL) features are provided in the Configuring Routing Between VLANs with Inter-Switch Link Encapsulation section.

IEEE 802.10 Protocol

The IEEE 802.10 protocol provides connectivity between VLANs. Originally developed to address the growing need for security within shared LAN/MAN environments, it incorporates authentication and encryption techniques to ensure data confidentiality and integrity throughout the network. Additionally, by functioning at Layer 2, it is well suited to high-throughput, low-latency switching environments. The IEEE 802.10 protocol can run over any LAN or HDLC serial interface.

Procedures for configuring routing between VLANs with IEEE 802.10 encapsulation are provided in the Configuring Routing Between VLANs with IEEE 802.10 section.

IEEE 802.1Q Protocol

The IEEE 802.1Q protocol is used to interconnect multiple switches and routers, and for defining VLAN topologies. Cisco currently supports IEEE 802.1Q for Fast Ethernet and Gigabit Ethernet interfaces.

Note Cisco does not support IEEE 802.1Q encapsulation for Ethernet interfaces.

Procedures for configuring routing between VLANs with IEEE 802.1Q encapsulation are provided in the Configuring Routing Between VLANs with IEEE 802.1Q Encapsulation.

ATM LANE Protocol

The ATM LAN Emulation (LANE) protocol provides a way for legacy LAN users to take advantage of ATM benefits without requiring modifications to end-station hardware or software. LANE emulates a broadcast environment like IEEE 802.3 Ethernet on top of an ATM network that is a point-to-point environment.

LANE makes ATM function like a LAN. LANE allows standard LAN drivers like NDIS and ODI to be used. The virtual LAN is transparent to applications. Applications can use normal LAN functions without the underlying complexities of the ATM implementation. For example, a station can send broadcasts and multicasts, even though ATM is defined as a point-to-point technology and does not support any-to-any services.

To accomplish this, special low-level software is implemented on an ATM client workstation, called the LAN Emulation Client (LEC). The client software communicates with a central control point called a LAN Emulation Server (LES). A broadcast and unknown server (BUS) acts as a central point to distribute broadcasts and multicasts. The LAN Emulation Configuration Server (LECS) holds a database of LECs and the ELANs they belong to. The database is maintained by a network administrator.

These protocols are described in detail in the *Cisco Internetwork Design Guide* .

ATM LANE Fast Simple Server Replication Protocol

To improve the ATM LANE Simple Server Replication Protocol (SSRP), Cisco introduced the ATM LANE Fast Simple Server Replication Protocol (FSSRP). FSSRP differs from LANE SSRP in that all configured LANE servers of an ELAN are always active. FSSRP-enabled LANE clients have virtual circuits (VCs) established to a maximum of four LANE servers and BUSs at one time. If a single LANE server goes down,

the LANE client quickly switches over to the next LANE server and BUS, resulting in no data or LE ARP table entry loss and no extraneous signalling.

The FSSRP feature improves upon SSRP such that LANE server and BUS switchover for LANE clients is immediate. With SSRP, a LANE server would go down, and depending on the network load, it may have taken considerable time for the LANE client to come back up joined to the correct LANE server and BUS. In addition to going down with SSRP, the LANE client would do the following:

- Clear out its data direct VCs
- Clear out its LE ARP entries
- Cause substantial signalling activity and data loss

FSSRP was designed to alleviate these problems with the LANE client. With FSSRP, each LANE client is simultaneously joined to up to four LANE servers and BUSs. The concept of the master LANE server and BUS is maintained; the LANE client uses the master LANE server when it needs LANE server BUS services. However, the difference between SSRP and FSSRP is that if and when the master LANE server goes down, the LANE client is already connected to multiple backup LANE servers and BUSs. The LANE client simply uses the next backup LANE server and BUS as the master LANE server and BUS.

VLAN Interoperability

Cisco IOS features bring added benefits to the VLAN technology. Enhancements to ISL, IEEE 802.10, and ATM LANE implementations enable routing of all major protocols between VLANs. These enhancements allow users to create more robust networks incorporating VLAN configurations by providing communications capabilities between VLANs.

Inter-VLAN Communications

The Cisco IOS supports full routing of several protocols over ISL and ATM LANE VLANs. IP, Novell IPX, and AppleTalk routing are supported over IEEE 802.10 VLANs. Standard routing attributes such as network advertisements, secondaries, and help addresses are applicable, and VLAN routing is fast switched. The table below shows protocols supported for each VLAN encapsulation format and corresponding Cisco IOS software releases in which support was introduced.

Table 1: Inter-VLAN Routing Protocol Support

Protocol	ISL	ATM LANE	IEEE 802.10
IP	Release 11.1	Release 10.3	Release 11.1
Novell IPX (default encapsulation)	Release 11.1	Release 10.3	Release 11.1
Novell IPX (configurable encapsulation)	Release 11.3	Release 10.3	Release 11.3
AppleTalk Phase II	Release 11.3	Release 10.3	--
DECnet	Release 11.3	Release 11.0	--

Protocol	ISL	ATM LANE	IEEE 802.10
Banyan VINES	Release 11.3	Release 11.2	--
XNS	Release 11.3	Release 11.2	--
CLNS	Release 12.1	--	--
IS-IS	Release 12.1	--	--

VLAN Translation

VLAN translation refers to the ability of the Cisco IOS software to translate between different VLANs or between VLAN and non-VLAN encapsulating interfaces at Layer 2. Translation is typically used for selective inter-VLAN switching of nonroutable protocols and to extend a single VLAN topology across hybrid switching environments. It is also possible to bridge VLANs on the main interface; the VLAN encapsulating header is preserved. Topology changes in one VLAN domain do not affect a different VLAN.

Designing Switched VLANs

By the time you are ready to configure routing between VLANs, you will have already defined them through the switches in your network. Issues related to network design and VLAN definition should be addressed during your network design. See the *Cisco Internetwork Design Guide* and the appropriate switch documentation for information on these topics:

- Sharing resources between VLANs
- Load balancing
- Redundant links
- Addressing
- Segmenting networks with VLANs--Segmenting the network into broadcast groups improves network security. Use router access lists based on station addresses, application types, and protocol types.
- Routers and their role in switched networks--In switched networks, routers perform broadcast management, route processing, and distribution, and provide communication between VLANs. Routers provide VLAN access to shared resources and connect to other parts of the network that are either logically segmented with the more traditional subnet approach or require access to remote sites across wide-area links.

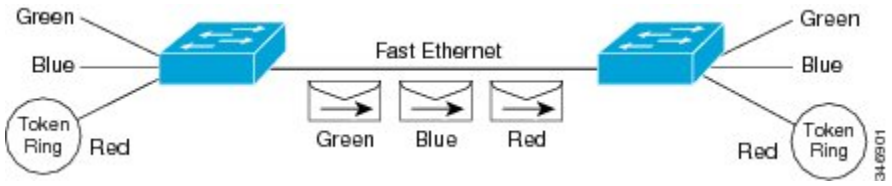
Frame Tagging in ISL

ISL is a Cisco protocol for interconnecting multiple switches and maintaining VLAN information as traffic goes between switches. ISL provides VLAN capabilities while maintaining full wire speed performance on Fast Ethernet links in full- or half-duplex mode. ISL operates in a point-to-point environment and will support up to 1000 VLANs. You can define virtually as many logical networks as are necessary for your environment.

With ISL, an Ethernet frame is encapsulated with a header that transports VLAN IDs between switches and routers. A 26-byte header that contains a 10-bit VLAN ID is propounded to the Ethernet frame.

A VLAN ID is added to the frame only when the frame is prepended for a nonlocal network. The figure below shows VLAN packets traversing the shared backbone. Each VLAN packet carries the VLAN ID within the packet header.

Figure 6: VLAN Packets Traversing the Shared Backbone



You can configure routing between any number of VLANs in your network. This section documents the configuration tasks for each protocol supported with ISL encapsulation. The basic process is the same, regardless of the protocol being routed. It involves the following tasks:

- Enabling the protocol on the router
- Enabling the protocol on the interface
- Defining the encapsulation format as ISL or TRISL
- Customizing the protocol according to the requirements for your environment

IEEE 802.1Q-in-Q VLAN Tag Termination on Subinterfaces

IEEE 802.1Q-in-Q VLAN Tag Termination simply adds another layer of IEEE 802.1Q tag (called “metro tag” or “PE-VLAN”) to the 802.1Q tagged packets that enter the network. The purpose is to expand the VLAN space by tagging the tagged packets, thus producing a “double-tagged” frame. The expanded VLAN space allows the service provider to provide certain services, such as Internet access on specific VLANs for specific customers, and yet still allows the service provider to provide other types of services for their other customers on other VLANs.

Generally the service provider’s customers require a range of VLANs to handle multiple applications. Service providers can allow their customers to use this feature to safely assign their own VLAN IDs on subinterfaces because these subinterface VLAN IDs are encapsulated within a service-provider designated VLAN ID for that customer. Therefore there is no overlap of VLAN IDs among customers, nor does traffic from different customers become mixed. The double-tagged frame is “terminated” or assigned on a subinterface with an expanded **encapsulation dot1q** command that specifies the two VLAN ID tags (outer VLAN ID and inner VLAN ID) terminated on the subinterface. See the figure below.

IEEE 802.1Q-in-Q VLAN Tag Termination is generally supported on whichever Cisco IOS features or protocols are supported on the subinterface; the exception is that Cisco 10000 series Internet router only supports PPPoE. For example if you can run PPPoE on the subinterface, you can configure a double-tagged frame for PPPoE. The only restriction is whether you assign ambiguous or unambiguous subinterfaces for the inner VLAN ID. See the figure below.

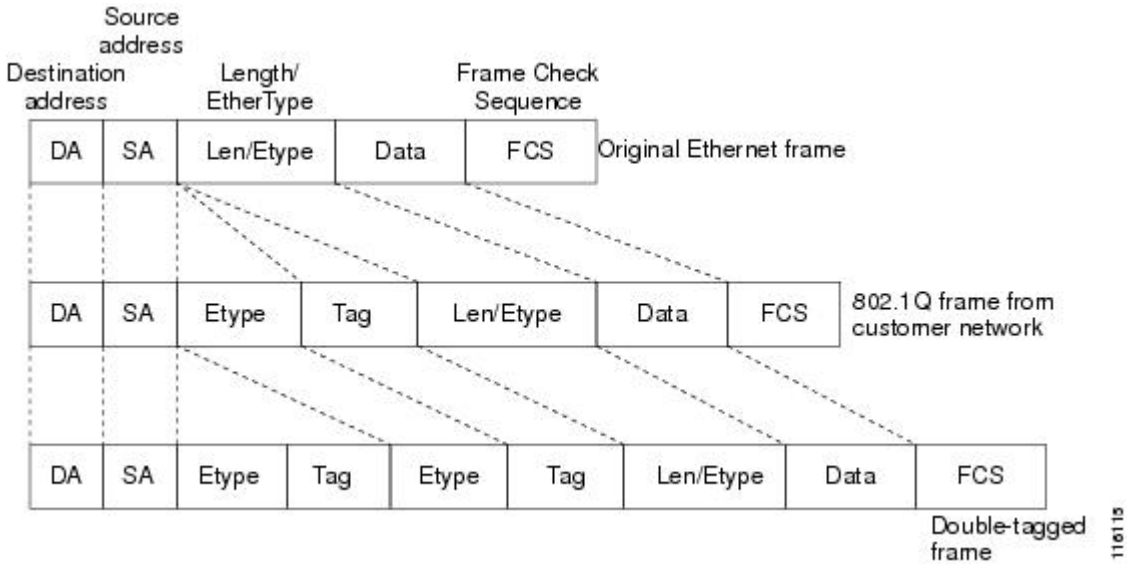
Note The Cisco 10000 series Internet router only supports Point-to-Point Protocol over Ethernet (PPPoE) and IP packets that are double-tagged for Q-in-Q VLAN tag termination. Specifically PPPoEoQ-in-Q and IPoQ-in-Q are supported.

The primary benefit for the service provider is reduced number of VLANs supported for the same number of customers. Other benefits of this feature include:

- PPPoE scalability. By expanding the available VLAN space from 4096 to approximately 16.8 million (4096 times 4096), the number of PPPoE sessions that can be terminated on a given interface is multiplied.
- When deploying Gigabyte Ethernet DSL Access Multiplexer (DSLAM) in wholesale model, you can assign the inner VLAN ID to represent the end-customer virtual circuit (VC) and assign the outer VLAN ID to represent the service provider ID.

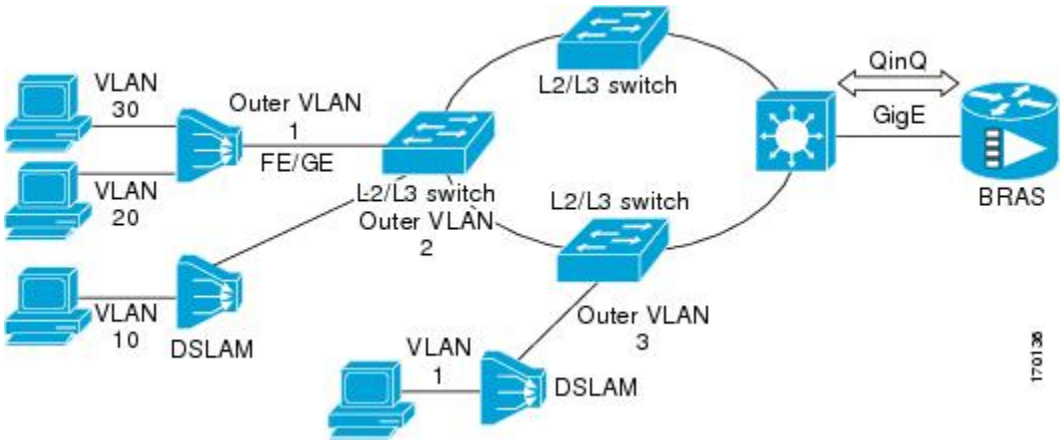
The Q-in-Q VLAN tag termination feature is simpler than the IEEE 802.1Q tunneling feature deployed for the Catalyst 6500 series switches or the Catalyst 3550 and Catalyst 3750 switches. Whereas switches require IEEE 802.1Q tunnels on interfaces to carry double-tagged traffic, routers need only encapsulate Q-in-Q VLAN tags within another level of 802.1Q tags in order for the packets to arrive at the correct destination as shown in figure below.

Figure 7: Untagged, 802.1Q-Tagged, and Double-Tagged Ethernet Frames



Cisco 10000 Series Internet Router Application

For the emerging broadband Ethernet-based DSLAM market, the Cisco 10000 series Internet router supports Q-in-Q encapsulation. With the Ethernet-based DSLAM model shown in the figure below, customers typically get their own VLAN and all these VLANs are aggregated on a DSLAM.



VLAN aggregation on a DSLAM will result in a lot of aggregate VLANs that at some point need to be terminated on the broadband remote access servers (BRAS). Although the model could connect the DSLAMs directly to the BRAS, a more common model uses the existing Ethernet-switched network where each DSLAM VLAN ID is tagged with a second tag (Q-in-Q) as it connects into the Ethernet-switched network.

The only model that is supported is PPPoE over Q-in-Q (PPPoEoQinQ). This can either be a PPP terminated session or as a L2TP LAC session.

The Cisco 10000 series Internet router already supports plain PPPoE and PPP over 802.1Q encapsulation. Supporting PPP over Q-in-Q encapsulation is new. PPP over Q-in-Q encapsulation processing is an extension to 802.1q encapsulation processing. A Q-in-Q frame looks like a VLAN 802.1Q frame, only it has two 802.1Q tags instead of one.

PPP over Q-in-Q encapsulation supports configurable outer tag Ethertype. The configurable Ethertype field values are 0x8100 (default), 0x9100, and 0x9200. See the figure below.

Security ACL Application on the Cisco 10000 Series Internet Router

The IEEE 802.1Q-in-Q VLAN Tag Termination feature provides limited security access control list (ACL) support for the Cisco 10000 series Internet router.

If you apply an ACL to PPPoE traffic on a Q-in-Q subinterface in a VLAN, apply the ACL directly on the PPPoE session, using virtual access interfaces (VAIs) or RADIUS attribute 11 or 242.

You can apply ACLs to virtual access interfaces by configuring them under virtual template interfaces. You can also configure ACLs by using RADIUS attribute 11 or 242. When you use attribute 242, a maximum of 30,000 sessions can have ACLs.

ACLs that are applied to the VLAN Q-in-Q subinterface have no effect and are silently ignored. In the following example, ACL 1 that is applied to the VLAN Q-in-Q subinterface level will be ignored:

```
Router(config)# interface FastEthernet3/0/0.100
Router(config-subif)# encapsulation dot1q 100 second-dot1q 200
Router(config-subif)# ip access-group 1
```

Unambiguous and Ambiguous Subinterfaces

The **encapsulation dot1q** command is used to configure Q-in-Q termination on a subinterface. The command accepts an Outer VLAN ID and one or more Inner VLAN IDs. The outer VLAN ID always has a specific value, while inner VLAN ID can either be a specific value or a range of values.

A subinterface that is configured with a single Inner VLAN ID is called an unambiguous Q-in-Q subinterface. In the following example, Q-in-Q traffic with an Outer VLAN ID of 101 and an Inner VLAN ID of 1001 is mapped to the Gigabit Ethernet 1/0.100 subinterface:

```
Router(config)# interface gigabitEehernet1/0.100
Router(config-subif)# encapsulation dot1q 101 second-dot1q 1001
```

A subinterface that is configured with multiple Inner VLAN IDs is called an ambiguous Q-in-Q subinterface. By allowing multiple Inner VLAN IDs to be grouped together, ambiguous Q-in-Q subinterfaces allow for a smaller configuration, improved memory usage and better scalability.

In the following example, Q-in-Q traffic with an Outer VLAN ID of 101 and Inner VLAN IDs anywhere in the 2001-2100 and 3001-3100 range is mapped to the Gigabit Ethernet 1/0.101 subinterface.:

```
Router(config)# interface gigabitethernet1/0.101
Router(config-subif)# encapsulation dot1q 101 second-dot1q 2001-2100,3001-3100
```

Ambiguous subinterfaces can also use the **any**keyword to specify the inner VLAN ID.

See the Monitoring and Maintaining VLAN Subinterfaces section for an example of how VLAN IDs are assigned to subinterfaces, and for a detailed example of how the **any** keyword is used on ambiguous subinterfaces.

Only PPPoE is supported on ambiguous subinterfaces. Standard IP routing is not supported on ambiguous subinterfaces.

Note On the Cisco 10000 series Internet router, Modular QoS services are only supported on unambiguous subinterfaces.

How to Configure Routing Between VLANS

Configuring a VLAN Range

Using the VLAN Range feature, you can group VLAN subinterfaces together so that any command entered in a group applies to every subinterface within the group. This capability simplifies configurations and reduces command parsing.

The VLAN Range feature provides the following benefits:

- Simultaneous Configurations: Identical commands can be entered once for a range of subinterfaces, rather than being entered separately for each subinterface.
- Overlapping Range Configurations: Overlapping ranges of subinterfaces can be configured.
- Customized Subinterfaces: Individual subinterfaces within a range can be customized or deleted.

Restrictions

- Each command you enter while you are in interface configuration mode with the **interface range** command is executed as it is entered. The commands are not batched together for execution after you exit interface configuration mode. If you exit interface configuration mode while the commands are being executed, some commands might not be executed on some interfaces in the range. Wait until the command prompt reappears before exiting interface configuration mode.
- The **no interface range** command is not supported. You must delete individual subinterfaces to delete a range.

Configuring a Range of VLAN Subinterfaces

Use the following commands to configure a range of VLAN subinterfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface range** {{**ethernet** | **fastethernet** | **gigabitethernet** | **atm**} *slot* / *interface* **.** *subinterface* **-**{{**ethernet** | **fastethernet** | **gigabitethernet** | **atm**}*slot* / *interface* **.** *subinterface*}
4. **encapsulation dot1Q** *vlan-id*
5. **no shutdown**
6. **exit**
7. **show running-config**
8. **show interfaces**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface range** {{**ethernet** | **fastethernet** | **gigabitethernet** | **atm**} *slot* / *interface* **.** *subinterface*	Selects the range of subinterfaces to be configured. **Note** The spaces around the dash are required. For example, the command **interface range fastethernet 1 - 5**is valid; the command **interface range fastethernet 1-5** is not valid.

	Command or Action	Purpose
	-{{ethernet | **fastethernet** | **gigabitethernet** | **atm**}*slot* / *interface* **.** *subinterface*} **Example:** `Router(config)# interface range fastethernet5/1.1 - fastethernet5/1.4`	
Step 4	**encapsulation dot1Q** *vlan-id* **Example:** `Router(config-if)# encapsulation dot1Q 301`	Applies a unique VLAN ID to each subinterface within the range. • *vlan-id* --Virtual LAN identifier. The allowed range is from 1 to 4095. • The VLAN ID specified by the *vlan-id* argument is applied to the first subinterface in the range. Each subsequent interface is assigned a VLAN ID, which is the specified *vlan-id* plus the subinterface number minus the first subinterface number (VLAN ID + subinterface number - first subinterface number).
Step 5	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Activates the interface. • This command is required only if you shut down the interface.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `Router# show running-config`	Verifies subinterface configuration.
Step 8	**show interfaces** **Example:** `Router# show interfaces`	Verifies that subinterfaces have been created.

Configuring Routing Between VLANs with Inter-Switch Link Encapsulation

This section describes the Inter-Switch Link (ISL) protocol and provides guidelines for configuring ISL and Token Ring ISL (TRISL) features. This section contains the following:

Configuring AppleTalk Routing over ISL

AppleTalk can be routed over VLAN subinterfaces using the ISL and IEEE 802.10 VLAN encapsulation protocols. The AppleTalk Routing over ISL and IEEE 802.10 Virtual LANs feature provides full-feature Cisco IOS software AppleTalk support on a per-VLAN basis, allowing standard AppleTalk capabilities to be configured on VLANs.

To route AppleTalk over ISL or IEEE 802.10 between VLANs, you need to customize the subinterface to create the environment in which it will be used. Perform the steps in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **appletalk routing** [**eigrp** *router-number*]
4. **interface** *type slot* / *port* **.** *subinterface-number*
5. **encapsulation isl** *vlan-identifier*
6. **appletalk cable-range** *cable-range* [*network* **.** *node*]
7. **appletalk zone** *zone-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**appletalk routing** [**eigrp** *router-number*] **Example:** `Router(config)# appletalk routing`	Enables AppleTalk routing globally on either ISL or 802.10 interfaces.
Step 4	**interface** *type slot* / *port* **.** *subinterface-number* **Example:** `Router(config)# interface Fddi 1/0.100`	Specifies the subinterface the VLAN will use.

	Command or Action	Purpose
Step 5	**encapsulation isl** *vlan-identifier* **Example:** **Example:** or **Example:** **encapsulation sde** *said* **Example:** `Router(config-if)# encapsulation sde 100`	Defines the encapsulation format as either ISL (**isl**) or IEEE 802.10 (**sde**), and specifies the VLAN identifier or security association identifier, respectively.
Step 6	**appletalk cable-range** *cable-range* [*network* **.** *node*] **Example:** `Router(config-if)# appletalk cable-range 100-100 100.2`	Assigns the AppleTalk cable range and zone for the subinterface.
Step 7	**appletalk zone** *zone-name* **Example:** `Router(config-if)# appletalk zone 100`	Assigns the AppleTalk zone for the subinterface.

Configuring Banyan VINES Routing over ISL

Banyan VINES can be routed over VLAN subinterfaces using the ISL encapsulation protocol. The Banyan VINES Routing over ISL Virtual LANs feature provides full-feature Cisco IOS software Banyan VINES support on a per-VLAN basis, allowing standard Banyan VINES capabilities to be configured on VLANs.

To route Banyan VINES over ISL between VLANs, you need to configure ISL encapsulation on the subinterface. Perform the steps in the following task in the order in which they appear:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vines routing** [*address*]
4. **interface** *type slot* / *port* **.** *subinterface-number*
5. **encapsulation isl** *vlan-identifier*
6. **vines metric** [*whole* [*fraction*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**vines routing** [*address*] **Example:** `Router(config)# vines routing`	Enables Banyan VINES routing globally.
Step 4	**interface** *type slot* / *port* **.** *subinterface-number* **Example:** `Router(config)# interface fastethernet 1/0.1`	Specifies the subinterface on which ISL will be used.
Step 5	**encapsulation isl** *vlan-identifier* **Example:** `Router(config-if)# encapsulation isl 200`	Defines the encapsulation format as ISL (**isl**), and specifies the VLAN identifier.
Step 6	**vines metric** [*whole* [*fraction*]] **Example:** `Router(config-if)#vines metric 2`	Enables VINES routing metric on an interface.

Configuring DECnet Routing over ISL

DECnet can be routed over VLAN subinterfaces using the ISL VLAN encapsulation protocols. The DECnet Routing over ISL Virtual LANs feature provides full-feature Cisco IOS software DECnet support on a per-VLAN basis, allowing standard DECnet capabilities to be configured on VLANs.

To route DECnet over ISL VLANs, you need to configure ISL encapsulation on the subinterface. Perform the steps described in the following task in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Router(config)# **decnet**[*network-number*] **routing**[*decnet-address*]
4. **interface** *type slot* / *port* **.** *subinterface-number*
5. **encapsulation isl** *vlan-identifier*
6. **decnet cost** [*cost-value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	Router(config)# **decnet**[*network-number*] **routing**[*decnet-address*] **Example:** `Router(config)# decnet routing 2.1`	Enables DECnet on the router.
Step 4	**interface** *type slot* / *port* **.** *subinterface-number* **Example:** `Router(config)# interface fastethernet 1/0.1`	Specifies the subinterface on which ISL will be used.

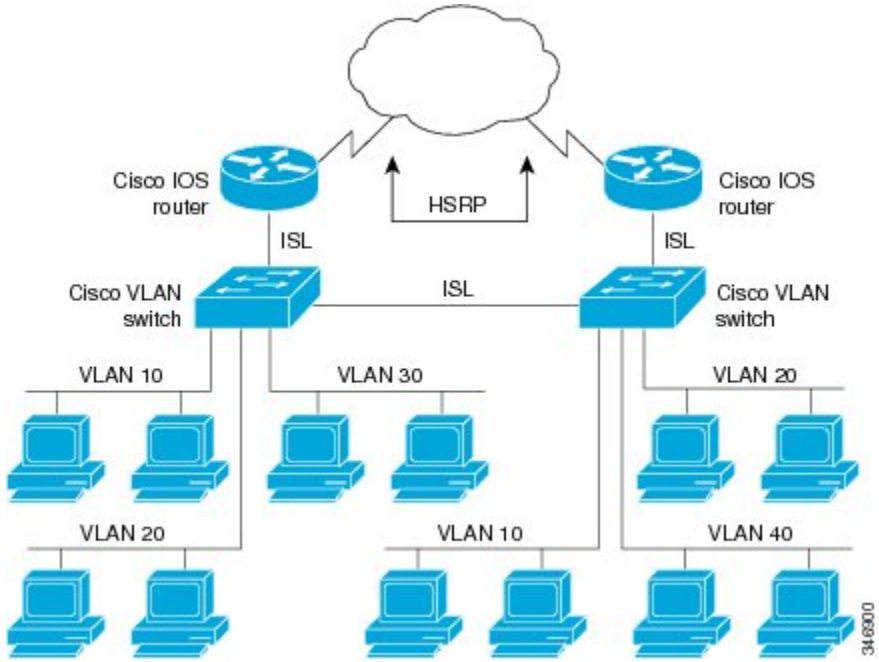
	Command or Action	Purpose
Step 5	**encapsulation isl** *vlan-identifier* **Example:** `Router(config-if)# encapsulation isl 200`	Defines the encapsulation format as ISL (**isl**), and specifies the VLAN identifier.
Step 6	**decnet cost** [*cost-value*] **Example:** `Router(config-if)# decnet cost 4`	Enables DECnet cost metric on an interface.

Configuring the Hot Standby Router Protocol over ISL

The Hot Standby Router Protocol (HSRP) provides fault tolerance and enhanced routing performance for IP networks. HSRP allows Cisco IOS routers to monitor each other's operational status and very quickly assume packet forwarding responsibility in the event the current forwarding device in the HSRP group fails or is taken down for maintenance. The standby mechanism remains transparent to the attached hosts and can be deployed on any LAN type. With multiple Hot Standby groups, routers can simultaneously provide redundant backup and perform loadsharing across different IP subnets.

The figure below illustrates HSRP in use with ISL providing routing between several VLANs.

Figure 8: Hot Standby Router Protocol in VLAN Configurations



A separate HSRP group is configured for each VLAN subnet so that Cisco IOS router A can be the primary and forwarding router for VLANs 10 and 20. At the same time, it acts as backup for VLANs 30 and 40. Conversely, Router B acts as the primary and forwarding router for ISL VLANs 30 and 40, as well as the secondary and backup router for distributed VLAN subnets 10 and 20.

Running HSRP over ISL allows users to configure redundancy between multiple routers that are configured as front ends for VLAN IP subnets. By configuring HSRP over ISLs, users can eliminate situations in which a single point of failure causes traffic interruptions. This feature inherently provides some improvement in overall networking resilience by providing load balancing and redundancy capabilities between subnets and VLANs.

To configure HSRP over ISLs between VLANs, you need to create the environment in which it will be used. Perform the tasks described in the following sections in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *port* **.** *subinterface-number*
4. **encapsulation isl** *vlan-identifier*
5. **ip address** *ip-address mask* [**secondary**]
6. Router(config-if)# **standby** [*group-number*] **ip**[*ip-address*[**secondary**]]
7. **standby** [*group-number*] **timers** *hellotime holdtime*
8. **standby** [*group-number*] **priority** *priority*
9. **standby** [*group-number*] **preempt**
10. **standby** [*group-number*] **track** *type-number*[*interface-priority*]
11. **standby** [*group-number*] **authentication** *string*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *port* **.** *subinterface-number* **Example:** `Router(config)# interface FastEthernet 1/1.110`	Specifies the subinterface on which ISL will be used and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**encapsulation isl** *vlan-identifier* **Example:** `Router(config-if)#  encapsulation isl 110`	Defines the encapsulation format, and specifies the VLAN identifier.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.1.1.2 255.255.255.0`	Specifies the IP address for the subnet on which ISL will be used.
Step 6	Router(config-if)# **standby** [*group-number*] **ip**[*ip-address*[**secondary**]] **Example:** `Router(config-if)# standby 1 ip 10.1.1.101`	Enables HSRP.
Step 7	**standby** [*group-number*] **timers** *hellotime holdtime* **Example:** `Router(config-if)# standby 1 timers 10 10`	Configures the time between hello packets and the hold time before other routers declare the active router to be down.
Step 8	**standby** [*group-number*] **priority** *priority* **Example:** `Router(config-if)# standby 1 priority 105`	Sets the Hot Standby priority used to choose the active router.
Step 9	**standby** [*group-number*] **preempt** **Example:** `Router(config-if)# standby 1 priority 105`	Specifies that if the local router has priority over the current active router, the local router should attempt to take its place as the active router.
Step 10	**standby** [*group-number*] **track** *type-number*[*interface-priority*] **Example:** `Router(config-if)# standby 1 track 4 5`	Configures the interface to track other interfaces, so that if one of the other interfaces goes down, the Hot Standby priority for the device is lowered.
Step 11	**standby** [*group-number*] **authentication** *string* **Example:** `Router(config-if)# standby 1 authentication hsrpword7`	Selects an authentication string to be carried in all HSRP messages.

What to Do Next

Note For more information on HSRP, see the "Configuring HSRP" module in the *Cisco IOS IP Application Services Configuration Guide* .

Configuring IP Routing over TRISL

The IP routing over TRISL VLANs feature extends IP routing capabilities to include support for routing IP frame types in VLAN configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip routing**
4. **interface** *type slot / port* **.** *subinterface-number*
5. **encapsulation tr-isl trbrf-vlan** *vlanid* **bridge-num** *bridge-number*
6. **ip address** *ip-address mask*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip routing** **Example:** `Router(config)# ip routing`	Enables IP routing on the router.

	Command or Action	Purpose
Step 4	**interface** *type slot* / *port* **.** *subinterface-number* **Example:** `Router(config)# interface FastEthernet4/0.1`	Specifies the subinterface on which TRISL will be used and enters interface configuration mode.
Step 5	**encapsulation tr-isl trbrf-vlan** *vlanid* **bridge-num** *bridge-number* **Example:** `Router(config-if# encapsulation tr-isl trbrf-vlan 999 bridge-num 14`	Defines the encapsulation for TRISL. • The DRiP database is automatically enabled when TRISL encapsulation is configured, and at least one TrBRF is defined, and the interface is configured for SRB or for routing with RIF.
Step 6	**ip address** *ip-address mask* **Example:** `Router(config-if# ip address 10.5.5.1 255.255.255.0`	Sets a primary IP address for an interface. • A mask identifies the bits that denote the network number in an IP address. When you use the mask to subnet a network, the mask is then referred to as a *subnet mask*. **Note** TRISL encapsulation must be specified for a subinterface before an IP address can be assigned to that subinterface.

Configuring IPX Routing on 802.10 VLANs over ISL

The IPX Encapsulation for 802.10 VLAN feature provides configurable IPX (Novell-FDDI, SAP, SNAP) encapsulation over 802.10 VLAN on router FDDI interfaces to connect the Catalyst 5000 VLAN switch. This feature extends Novell NetWare routing capabilities to include support for routing all standard IPX encapsulations for Ethernet frame types in VLAN configurations. Users with Novell NetWare environments can now configure any one of the three IPX Ethernet encapsulations to be routed using Secure Data Exchange (SDE) encapsulation across VLAN boundaries. IPX encapsulation options now supported for VLAN traffic include the following:

- Novell-FDDI (IPX FDDI RAW to 802.10 on FDDI)
- SAP (IEEE 802.2 SAP to 802.10 on FDDI)
- SNAP (IEEE 802.2 SNAP to 802.10 on FDDI)

NetWare users can now configure consolidated VLAN routing over a single VLAN trunking FDDI interface. Not all IPX encapsulations are currently supported for SDE VLAN. The IPX interior encapsulation support can be achieved by messaging the IPX header before encapsulating in the SDE format. Fast switching will also support all IPX interior encapsulations on non-MCI platforms (for example non-AGS+ and non-7000). With configurable Ethernet encapsulation protocols, users have the flexibility of using VLANs regardless of their NetWare Ethernet encapsulation. Configuring Novell IPX encapsulations on a per-VLAN basis facilitates migration between versions of Netware. NetWare traffic can now be routed across VLAN boundaries with standard encapsulation options (*arpa* , *sap* , and *snap*) previously unavailable. Encapsulation types and

corresponding framing types are described in the "Configuring Novell IPX " module of the *Cisco IOS Novell IPX Configuration Guide* .

Note Only one type of IPX encapsulation can be configured per VLAN (subinterface). The IPX encapsulation used must be the same within any particular subnet; a single encapsulation must be used by all NetWare systems that belong to the same VLAN.

To configure Cisco IOS software on a router with connected VLANs to exchange different IPX framing protocols, perform the steps described in the following task in the order in which they are appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipx routing** [*node*]
4. **interface** *fddi slot* / *port* **.** *subinterface-number*
5. **encapsulation sde** *vlan-identifier*
6. **ipx network** *network* **encapsulation** *encapsulation-type*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipx routing** [*node*] **Example:** `Router(config)# ipx routing`	Enables IPX routing globally.
Step 4	**interface** *fddi slot* / *port* **.** *subinterface-number* **Example:** `Router(config)# interface 2/0.1`	Specifies the subinterface on which SDE will be used and enters interface configuration mode.

	Command or Action	Purpose
Step 5	**encapsulation sde** *vlan-identifier* **Example:** `Router(config-if)# encapsulation isl 20`	Defines the encapsulation format and specifies the VLAN identifier.
Step 6	**ipx network** *network* **encapsulation** *encapsulation-type* **Example:** `Router(config-if)# ipx network 20 encapsulation sap`	Specifies the IPX encapsulation among Novell-FDDI, SAP, or SNAP.

Configuring IPX Routing over TRISL

The IPX Routing over ISL VLANs feature extends Novell NetWare routing capabilities to include support for routing all standard IPX encapsulations for Ethernet frame types in VLAN configurations. Users with Novell NetWare environments can configure either SAP or SNAP encapsulations to be routed using the TRISL encapsulation across VLAN boundaries. The SAP (Novell Ethernet_802.2) IPX encapsulation is supported for VLAN traffic.

NetWare users can now configure consolidated VLAN routing over a single VLAN trunking interface. With configurable Ethernet encapsulation protocols, users have the flexibility of using VLANs regardless of their NetWare Ethernet encapsulation. Configuring Novell IPX encapsulations on a per-VLAN basis facilitates migration between versions of Netware. NetWare traffic can now be routed across VLAN boundaries with standard encapsulation options (*sap* and *snap*) previously unavailable. Encapsulation types and corresponding framing types are described in the "Configuring Novell IPX " module of the *Cisco IOS Novell IPX Configuration Guide* .

Note Only one type of IPX encapsulation can be configured per VLAN (subinterface). The IPX encapsulation used must be the same within any particular subnet: A single encapsulation must be used by all NetWare systems that belong to the same LANs.

To configure Cisco IOS software to exchange different IPX framing protocols on a router with connected VLANs, perform the steps in the following task in the order in which they are appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipx routing** [*node*]
4. **interface** *type slot / port* **.** *subinterface-number*
5. **encapsulation tr-isl trbrf-vlan** *trbrf-vlan* **bridge-num** *bridge-num*
6. **ipx network** *network* **encapsulation** *encapsulation-type*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipx routing** [*node*] **Example:** `Router(config)# source-bridge ring-group 100`	Enables IPX routing globally.
Step 4	**interface** *type slot* / *port* **.** *subinterface-number* **Example:** `Router(config)# interface TokenRing 3/1`	Specifies the subinterface on which TRISL will be used and enters interface configuration mode.
Step 5	**encapsulation tr-isl trbrf-vlan** *trbrf-vlan* **bridge-num** *bridge-num* **Example:** `Router(config-if)#encapsulation tr-isl trbrf-vlan 999 bridge-num 14`	Defines the encapsulation for TRISL.
Step 6	**ipx network** *network* **encapsulation** *encapsulation-type* **Example:** `Router(config-if)# ipx network 100 encapsulation sap`	Specifies the IPX encapsulation on the subinterface by specifying the NetWare network number (if necessary) and the encapsulation type.

What to Do Next

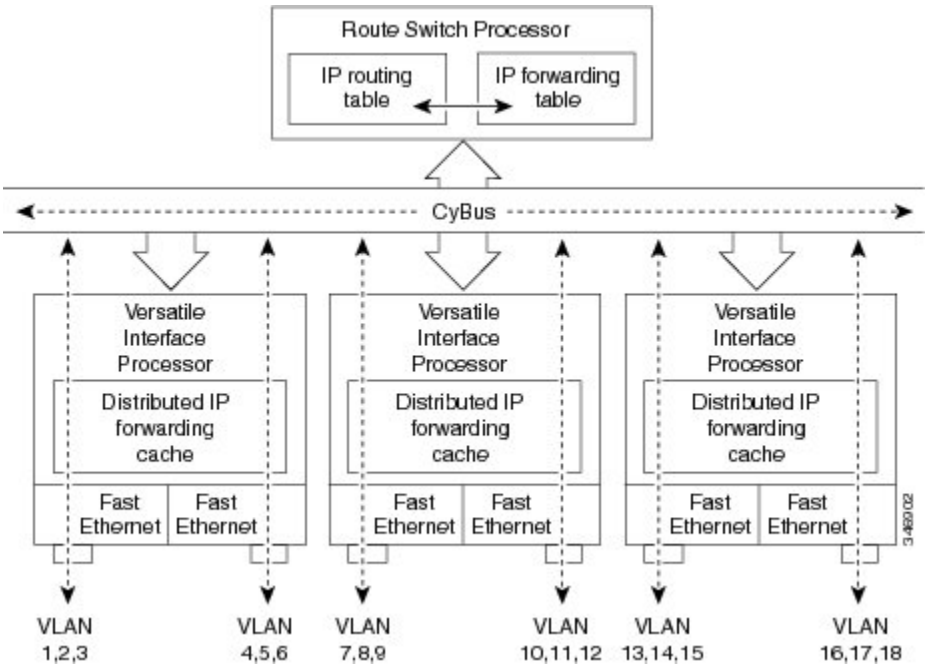
Note The default IPX encapsulation format for Cisco IOS routers is “novell-ether” (Novell Ethernet_802.3). If you are running Novell Netware 3.12 or 4.0, the new Novell default encapsulation format is Novell Ethernet_802.2 and you should configure the Cisco router with the IPX encapsulation format “sap.”

Configuring VIP Distributed Switching over ISL

With the introduction of the VIP distributed ISL feature, ISL encapsulated IP packets can be switched on Versatile Interface Processor (VIP) controllers installed on Cisco 7500 series routers.

The second generation VIP2 provides distributed switching of IP encapsulated in ISL in VLAN configurations. Where an aggregation route performs inter-VLAN routing for multiple VLANs, traffic can be switched autonomously on-card or between cards rather than through the central Route Switch Processor (RSP). The figure below shows the VIP distributed architecture of the Cisco 7500 series router.

Figure 9: Cisco 7500 Distributed Architecture



This distributed architecture allows incremental capacity increases by installation of additional VIP cards. Using VIP cards for switching the majority of IP VLAN traffic in multiprotocol environments substantially increases routing performance for the other protocols because the RSP offloads IP and can then be dedicated to switching the non-IP protocols.

VIP distributed switching offloads switching of ISL VLAN IP traffic to the VIP card, removing involvement from the main CPU. Offloading ISL traffic to the VIP card substantially improves networking performance. Because you can install multiple VIP cards in a router, VLAN routing capacity is increased linearly according to the number of VIP cards installed in the router.

To configure distributed switching on the VIP, you must first configure the router for IP routing. Perform the tasks described below in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip routing**
4. **interface** *type slot / port-adapter / port*
5. **ip route-cache distributed**
6. **encapsulation isl** *vlan-identifier*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip routing** **Example:** `Router(config)# ip routing`	Enables IP routing on the router. • For more information about configuring IP routing, see the appropriate Cisco IOS *IP Routing Configuration Guide* for the version of Cisco IOS you are using.
Step 4	**interface** *type slot / port-adapter / port* **Example:** `Router(config)# interface FastEthernet1/0/0`	Specifies the interface and enters interface configuration mode.
Step 5	**ip route-cache distributed** **Example:** `Router(config-if)# ip route-cache distributed`	Enables VIP distributed switching of IP packets on the interface.
Step 6	**encapsulation isl** *vlan-identifier* **Example:** `Router(config-if)# encapsulation isl 1`	Defines the encapsulation format as ISL, and specifies the VLAN identifier.

Configuring XNS Routing over ISL

XNS can be routed over VLAN subinterfaces using the ISL VLAN encapsulation protocol. The XNS Routing over ISL Virtual LANs feature provides full-feature Cisco IOS software XNS support on a per-VLAN basis, allowing standard XNS capabilities to be configured on VLANs.

To route XNS over ISL VLANs, you need to configure ISL encapsulation on the subinterface. Perform the steps described in the following task in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **xns routing** [*address*]
4. **interface** *type slot* / *port* **.** *subinterface-number*
5. **encapsulation isl** *vlan-identifier*
6. **xns network** [*number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**xns routing** [*address*] **Example:** `Router(config)# xns routing 0123.4567.adcb`	Enables XNS routing globally.
Step 4	**interface** *type slot* / *port* **.** *subinterface-number* **Example:** `Router(config)# interface fastethernet 1/0.1`	Specifies the subinterface on which ISL will be used and enters interface configuration mode.
Step 5	**encapsulation isl** *vlan-identifier* **Example:** `Router(config-if)# encapsulation isl 100`	Defines the encapsulation format as ISL (**isl**), and specifies the VLAN identifier.

	Command or Action	Purpose
Step 6	**xns network** [*number*] **Example:** `Router(config-if)# xns network 20`	Enables XNS routing on the subinterface.

Configuring CLNS Routing over ISL

CLNS can be routed over VLAN subinterfaces using the ISL VLAN encapsulation protocol. The CLNS Routing over ISL Virtual LANs feature provides full-feature Cisco IOS software CLNS support on a per-VLAN basis, allowing standard CLNS capabilities to be configured on VLANs.

To route CLNS over ISL VLANs, you need to configure ISL encapsulation on the subinterface. Perform the steps described in the following task in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **clns routing**
4. **interface** *type slot* / *port* **.** *subinterface-number*
5. **encapsulation isl** *vlan-identifier*
6. **clns enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**clns routing** **Example:** `Router(config)# clns routing`	Enables CLNS routing globally.

	Command or Action	Purpose
Step 4	**interface** *type slot* / *port* **.** *subinterface-number* **Example:** `Router(config-if)# interface fastethernet 1/0.1`	Specifies the subinterface on which ISL will be used and enters interface configuration mode.
Step 5	**encapsulation isl** *vlan-identifier* **Example:** `Router(config-if)# encapsulation isl 100`	Defines the encapsulation format as ISL (**isl**), and specifies the VLAN identifier.
Step 6	**clns enable** **Example:** `Router(config-if)# clns enable`	Enables CLNS routing on the subinterface.

Configuring IS-IS Routing over ISL

IS-IS routing can be enabled over VLAN subinterfaces using the ISL VLAN encapsulation protocol. The IS-IS Routing over ISL Virtual LANs feature provides full-feature Cisco IOS software IS-IS support on a per-VLAN basis, allowing standard IS-IS capabilities to be configured on VLANs.

To enable IS-IS over ISL VLANs, you need to configure ISL encapsulation on the subinterface. Perform the steps described in the following task in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** [*tag*]
4. **net** *network-entity-title*
5. **interface** *type slot* / *port* **.** *subinterface-number*
6. **encapsulation isl** *vlan-identifier*
7. **clns router isis network** [*tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**router isis** [*tag*] **Example:** `Router(config)# isis routing test-proc2`	Enables IS-IS routing, and enters router configuration mode.
Step 4	**net** *network-entity-title* **Example:** `Router(config)# net 49.0001.0002.aaaa.aaaa.aaaa.00`	Configures the NET for the routing process.
Step 5	**interface** *type slot* / *port* **.** *subinterface-number* **Example:** `Router(config)# interface fastethernet 2.`	Specifies the subinterface on which ISL will be used and enters interface configuration mode.
Step 6	**encapsulation isl** *vlan-identifier* **Example:** `Router(config-if)# encapsulation isl 101`	Defines the encapsulation format as ISL (**isl**), and specifies the VLAN identifier.
Step 7	**clns router isis network** [*tag*] **Example:** `Router(config-if)# clns router is-is network test-proc2`	Specifies the interfaces that should be actively routing IS-IS.

Configuring Routing Between VLANs with IEEE 802.10 Encapsulation

This section describes the required and optional tasks for configuring routing between VLANs with IEEE 802.10 encapsulation.

HDLC serial links can be used as VLAN trunks in IEEE 802.10 VLANs to extend a virtual topology beyond a LAN backbone.

AppleTalk can be routed over VLAN subinterfaces using the ISL or IEEE 802.10 VLANs feature that provides full-feature Cisco IOS software AppleTalk support on a per-VLAN basis, allowing standard AppleTalk capabilities to be configured on VLANs.

AppleTalk users can now configure consolidated VLAN routing over a single VLAN trunking interface. Prior to introduction of this feature, AppleTalk could be routed only on the main interface on a LAN port. If AppleTalk routing was disabled on the main interface or if the main interface was shut down, the entire physical interface would stop routing any AppleTalk packets. With this feature enabled, AppleTalk routing on subinterfaces will be unaffected by changes in the main interface with the main interface in the “no-shut” state.

To route AppleTalk over IEEE 802.10 between VLANs, create the environment in which it will be used by customizing the subinterface and perform the tasks described in the following steps in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **appletalk routing** [**eigrp** *router-number*]
4. **interface fastethernet** *slot* / *port* **.** subinterface-number
5. **appletalk cable-range** *cable-range* [*network* **.** *node*]
6. **appletalk zone** *>zone-name*
7. **encapsulation sde** *said*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**appletalk routing** [**eigrp** *router-number*] **Example:** `Router(config)# appletalk routing`	Enables AppleTalk routing globally.

	Command or Action	Purpose
Step 4	**interface fastethernet** *slot* / *port* **.** subinterface-number **Example:** `Router(config)# interface fastethernet 4/1.00`	Specifies the subinterface the VLAN will use and enters inerface configuration mode.
Step 5	**appletalk cable-range** *cable-range* [*network* **.** *node*] **Example:** `Router(config-if)# appletalk 100-100 100.1`	Assigns the AppleTalk cable range and zone for the subinterface.
Step 6	**appletalk zone** *>zone-name* **Example:** `Router(config-if)# appletalk zone eng`	Assigns the AppleTalk zone for the subinterface.
Step 7	**encapsulation sde** *said* **Example:** `Router(config-if)# encapsulation sde 100`	Defines the encapsulation format as IEEE 802.10 (sde) and specifies the VLAN identifier or security association identifier, respectively.

What to Do Next

Note For more information on configuring AppleTalk, see the "Configuring AppleTalk" module in the *Cisco IOS AppleTalk Configuration Guide* .

Configuring Routing Between VLANs with IEEE 802.1Q Encapsulation

This section describes the required and optional tasks for configuring routing between VLANs with IEEE 802.1Q encapsulation. The IEEE 802.1Q protocol is used to interconnect multiple switches and routers, and for defining VLAN topologies.

Prerequisites

Configuring routing between VLANs with IEEE 802.1Q encapsulation assumes the presence of a single spanning tree and of an explicit tagging scheme with one-level tagging.

You can configure routing between any number of VLANs in your network.

Restrictions

The IEEE 802.1Q standard is extremely restrictive to untagged frames. The standard provides only a per-port VLANs solution for untagged frames. For example, assigning untagged frames to VLANs takes into consideration only the port from which they have been received. Each port has a parameter called a *permanent virtual identification* (Native VLAN) that specifies the VLAN assigned to receive untagged frames.

The main characteristics of the IEEE 802.1Q are that it assigns frames to VLANs by filtering and that the standard assumes the presence of a single spanning tree and of an explicit tagging scheme with one-level tagging.

This section contains the configuration tasks for each protocol supported with IEEE 802.1Q encapsulation. The basic process is the same, regardless of the protocol being routed. It involves the following tasks:

- Enabling the protocol on the router
- Enabling the protocol on the interface
- Defining the encapsulation format as IEEE 802.1Q
- Customizing the protocol according to the requirements for your environment

To configure IEEE 802.1Q on your network, perform the following tasks. One of the following tasks is required depending on the protocol being used.

- Configuring AppleTalk Routing over IEEE 802.1Q, on page 39 (required)
- Configuring IP Routing over IEEE 802.1Q, on page 41 (required)
- Configuring IPX Routing over IEEE 802.1Q, on page 42 (required)

The following tasks are optional. Perform the following tasks to connect a network of hosts over a simple bridging-access device to a remote access concentrator bridge between IEEE 802.1Q VLANs. The following sections contain configuration tasks for the Integrated Routing and Bridging, Transparent Bridging, and PVST+ Between VLANs with IEEE 802.1Q Encapsulation:

- Configuring a VLAN for a Bridge Group with Default VLAN1, on page 44 (optional)
- Configuring a VLAN for a Bridge Group as a Native VLAN, on page 45 (optional)

Configuring AppleTalk Routing over IEEE 802.1Q

AppleTalk can be routed over virtual LAN (VLAN) subinterfaces using the IEEE 802.1Q VLAN encapsulation protocol. AppleTalk Routing provides full-feature Cisco IOS software AppleTalk support on a per-VLAN basis, allowing standard AppleTalk capabilities to be configured on VLANs.

To route AppleTalk over IEEE 802.1Q between VLANs, you need to customize the subinterface to create the environment in which it will be used. Perform the steps in the order in which they appear.

Use the following task to enable AppleTalk routing on IEEE 802.1Q interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **appletalk routing** [**eigrp** *router-number*]
4. **interface fastethernet** *slot* / *port* **.** subinterface-number
5. **encapsulation dot1q** *vlan-identifier*
6. **appletalk cable-range** *cable-range* [*network* **.** *node*]
7. **appletalk zone** *zone-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**appletalk routing** [**eigrp** *router-number*] **Example:** `Router(config)# appletalk routing`	Enables AppleTalk routing globally.
Step 4	**interface fastethernet** *slot* / *port* **.** subinterface-number **Example:** `Router(config)# interface fastethernet 4/1.00`	Specifies the subinterface the VLAN will use and enters interface configuration mode.
Step 5	**encapsulation dot1q** *vlan-identifier* **Example:** `Router(config-if)# encapsulation dot1q 100`	Defines the encapsulation format as IEEE 802.1Q (**dot1q**), and specifies the VLAN identifier.
Step 6	**appletalk cable-range** *cable-range* [*network* **.** *node*] **Example:** `Router(config-if)# appletalk cable-range 100-100 100.1`	Assigns the AppleTalk cable range and zone for the subinterface.

	Command or Action	Purpose
Step 7	**appletalk zone** *zone-name* **Example:** `Router(config-if)# appletalk zone eng`	Assigns the AppleTalk zone for the subinterface.

What to Do Next

Note For more information on configuring AppleTalk, see the "Configuring AppleTalk" module in the *Cisco IOS AppleTalk Configuration Guide* .

Configuring IP Routing over IEEE 802.1Q

IP routing over IEEE 802.1Q extends IP routing capabilities to include support for routing IP frame types in VLAN configurations using the IEEE 802.1Q encapsulation.

To route IP over IEEE 802.1Q between VLANs, you need to customize the subinterface to create the environment in which it will be used. Perform the tasks described in the following sections in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. ip routing
4. **interface fastethernet** *slot* / *port* **.** subinterface-number
5. **encapsulation dot1q** vlanid
6. **ip address** *ip-address mask*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	ip routing **Example:** `Router(config)# ip routing`	Enables IP routing on the router.
Step 4	**interface fastethernet** *slot* / *port* **.** subinterface-number **Example:** `Router(config)# interface fastethernet 4/1.101`	Specifies the subinterface on which IEEE 802.1Q will be used and enters interface configuration mode.
Step 5	**encapsulation dot1q** vlanid **Example:** `Router(config-if)# encapsulation dot1q 101`	Defines the encapsulation format at IEEE.802.1Q (dot1q) and specifies the VLAN identifier.
Step 6	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip addr 10.0.0.11 255.0.0.0`	Sets a primary IP address and mask for the interface.

What to Do Next

Once you have IP routing enabled on the router, you can customize the characteristics to suit your environment. See the appropriate *Cisco IOS IP Routing Configuration Guide* for the version of Cisco IOS you are using.

Configuring IPX Routing over IEEE 802.1Q

IPX routing over IEEE 802.1Q VLANs extends Novell NetWare routing capabilities to include support for routing Novell Ethernet_802.3 encapsulation frame types in VLAN configurations. Users with Novell NetWare environments can configure Novell Ethernet_802.3 encapsulation frames to be routed using IEEE 802.1Q encapsulation across VLAN boundaries.

To configure Cisco IOS software on a router with connected VLANs to exchange IPX Novell Ethernet_802.3 encapsulated frames, perform the steps described in the following task in the order in which they appear.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipx routing** [*node*]
4. **interface fastethernet** *slot* / *port* **.** subinterface-number
5. **encapsulation dot1q** vlanid
6. **ipx network** *network*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ipx routing** [*node*] **Example:** `Router(config)# ipx routing`	Enables IPX routing globally.
Step 4	**interface fastethernet** *slot* / *port* **.** subinterface-number **Example:** `Router(config)# interface fastethernet 4/1.102`	Specifies the subinterface on which IEEE 802.1Q will be used and enters interface configuration mode.
Step 5	**encapsulation dot1q** vlanid **Example:** `Router(config-if)# encapsulation dot1q 102`	Defines the encapsulation format at IEEE.802.1Q (**dot1q**) and specifies the VLAN identifier.
Step 6	**ipx network** *network* **Example:** `Router(config-if)# ipx network 100`	Specifies the IPX network number.

Configuring a VLAN for a Bridge Group with Default VLAN1

Use the following task to configure a VLAN associated with a bridge group with a default native VLAN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface fastethernet** *slot* / *port* **.** subinterface-number
4. **encapsulation dot1q** vlanid
5. **bridge-group** *bridge-group*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface fastethernet** *slot* / *port* **.** subinterface-number **Example:** `Router(config)# interface fastethernet 4/1.100`	Selects a particular interface to configure and enters interface configuration mode.
Step 4	**encapsulation dot1q** vlanid **Example:** `Router(config-subif)# encapsulation dot1q 1`	Defines the encapsulation format at IEEE.802.1Q (dot1q) and specifies the VLAN identifier. • The specified VLAN is by default the native VLAN. **Note** If there is no explicitly defined native VLAN, the default VLAN1 becomes the native VLAN.
Step 5	**bridge-group** *bridge-group* **Example:** `Router(config-subif)# bridge-group 1`	Assigns the bridge group to the interface.

Configuring a VLAN for a Bridge Group as a Native VLAN

Use the following task to configure a VLAN associated to a bridge group as a native VLAN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface fastethernet** *slot* / *port* **.** subinterface-number
4. **encapsulation dot1q** *vlanid* **native**
5. **bridge-group** *bridge-group*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface fastethernet** *slot* / *port* **.** subinterface-number **Example:** `Router(config)# interface fastethernet 4/1.100`	Selects a particular interface to configure and enters interface configuration mode.
Step 4	**encapsulation dot1q** *vlanid* **native** **Example:** `Router(config-subif)# encapsulation dot1q 20 native`	Defines the encapsulation format at IEEE.802.1Q (**dot1q**) and specifies the VLAN identifier. VLAN 20 is specified as the native VLAN. **Note** If there is no explicitly defined native VLAN, the default VLAN1 becomes the native VLAN.
Step 5	**bridge-group** *bridge-group* **Example:** `Router(config-subif)# bridge-group 1`	Assigns the bridge group to the interface.

What to Do Next

Note If there is an explicitly defined native VLAN, VLAN1 will only be used to process CST.

Configuring IEEE 802.1Q-in-Q VLAN Tag Termination

Encapsulating IEEE 802.1Q VLAN tags within 802.1Q enables service providers to use a single VLAN to support customers who have multiple VLANs. The IEEE 802.1Q-in-Q VLAN Tag Termination feature on the subinterface level preserves VLAN IDs and keeps traffic in different customer VLANs segregated.

You must have checked Feature Navigator to verify that your Cisco device and software image support this feature.

You must be connected to an Ethernet device that supports double VLAN tag imposition/disposition or switching.

The following restrictions apply to the Cisco 10000 series Internet router for configuring IEEE 802.1Q-in-Q VLAN tag termination:

- Supported on Ethernet, FastEthernet, or Gigabit Ethernet interfaces.
- Supports only Point-to-Point Protocol over Ethernet (PPPoE) packets that are double-tagged for Q-in-Q VLAN tag termination.
- IP and Multiprotocol Label Switching (MPLS) packets are not supported.
- Modular QoS can be applied to unambiguous subinterfaces only.
- Limited ACL support.

Perform these tasks to configure the main interface used for the Q-in-Q double tagging and to configure the subinterfaces.

Configuring EtherType Field for Outer VLAN Tag Termination

The following restrictions are applicable for the Cisco 10000 series Internet router:

- PPPoE is already configured.
- Virtual private dial-up network (VPDN) is enabled.

The first task is optional. A step in this task shows you how to configure the EtherType field to be 0x9100 for the outer VLAN tag, if that is required.

After the subinterface is defined, the 802.1Q encapsulation is configured to use the double tagging.

To configure the EtherType field for Outer VLAN Tag Termination, use the following steps. This task is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **dot1q tunneling ethertype** *ethertype*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/0/0`	Configures an interface and enters interface configuration mode.
Step 4	**dot1q tunneling ethertype** *ethertype* **Example:** `Router(config-if)# dot1q tunneling ethertype 0x9100`	(Optional) Defines the Ethertype field type used by peer devices when implementing Q-in-Q VLAN tagging. • Use this command if the Ethertype of peer devices is 0x9100 or 0x9200 (0x9200 is only supported on the Cisco 10000 series Internet router). • Cisco 10000 series Internet router supports both the 0x9100 and 0x9200 Ethertype field types.

Configuring the Q-in-Q Subinterface

Use the following steps to configure Q-in-Q subinterfaces. This task is required.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* **.** *subinterface-number*
4. **encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*| *vlan-id* **-** *vlan-id* [**,** *vlan-id* **-** *vlan-id*]}
5. **pppoe enable** [**group** *group-name*]
6. **exit**
7. Repeat Step 3 to configure another subinterface.
8. Repeat Step 4 and Step 5 to specify the VLAN tags to be terminated on the subinterface.
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **.** *subinterface-number* **Example:** `Router(config)# interface gigabitethernet 1/0/0.1`	Configures a subinterface and enters subinterface configuration mode.
Step 4	**encapsulation dot1q** *vlan-id* **second-dot1q** {**any** | *vlan-id*| *vlan-id* **-** *vlan-id* [**,** *vlan-id* **-** *vlan-id*]} **Example:** `Router(config-subif)# encapsulation dot1q 100 second-dot1q 200`	(Required) Enables the 802.1Q encapsulation of traffic on a specified subinterface in a VLAN. • Use the **second-dot1q** keyword and the *vlan-id*argument to specify the VLAN tags to be terminated on the subinterface. • In this example, an unambiguous Q-in-Q subinterface is configured because only one inner VLAN ID is specified. • Q-in-Q frames with an outer VLAN ID of 100 and an inner VLAN ID of 200 will be terminated.
Step 5	**pppoe enable** [**group** *group-name*]	Enables PPPoE sessions on a subinterface.

	Command or Action	Purpose
	Example: `Router(config-subif)# pppoe enable group vpn1`	• The example specifies that the PPPoE profile, vpn1, will be used by PPPoE sessions on the subinterface.
Step 6	**exit** **Example:** `Router(config-subif)# exit`	Exits subinterface configuration mode and returns to interface configuration mode. • Repeat this step one more time to exit interface configuration mode.
Step 7	Repeat Step 3 to configure another subinterface. **Example:** `Router(config-if)# interface gigabitethernet 1/0/0.2`	(Optional) Configures a subinterface and enters subinterface configuration mode.

Step 8	Repeat Step 4 and Step 5 to specify the VLAN tags to be terminated on the subinterface.
	Example:
	`Router(config-subif)# encapsulation dot1q 100 second-dot1q 100-199,201-600`
	Example:
	Example:
	`Router(config-subif)# pppoe enable group vpn1`
	Example:
		Step 4 enables the 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
		• Use the **second-dot1q** keyword and the *vlan-id*argument to specify the VLAN tags to be terminated on the subinterface.
		• In the example, an ambiguous Q-in-Q subinterface is configured because a range of inner VLAN IDs is specified.
		• Q-in-Q frames with an outer VLAN ID of 100 and an inner VLAN ID in the range of 100 to 199 or 201 to 600 will be terminated.
		Step 5 enables PPPoE sessions on the subinterface. The example specifies that the PPPoE profile, vpn1, will be used by PPPoE sessions on the subinterface.
		Note Step 5 is required for the Cisco 10000 series Internet router because it only supports PPPoEoQinQ traffic.

Step 9	**end** **Example:** `Router(config-subif)# end`	Exits subinterface configuration mode and returns to privileged EXEC mode.

Verifying the IEEE 802.1Q-in-Q VLAN Tag Termination

Perform this optional task to verify the configuration of the IEEE 802.1Q-in-Q VLAN Tag Termination feature.

SUMMARY STEPS

1. **enable**
2. **show running-config**
3. **show vlans dot1q** [**internal** | *interface-type interface-number* **.***subinterface-number*[**detail**] | *outer-id*[*interface-type interface-number* | **second-dot1q** [*inner-id*| **any**]] [**detail**]]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **show running-config**

Use this command to show the currently running configuration on the device. You can use delimiting characters to display only the relevant parts of the configuration.

The following shows the currently running configuration on a Cisco 7300 series router:

Example:

```
Router# show running-config
.
.
.
interface FastEthernet0/0.201
 encapsulation dot1Q 201
 ip address 10.7.7.5 255.255.255.252
!
interface FastEthernet0/0.401
 encapsulation dot1Q 401
 ip address 10.7.7.13 255.255.255.252
!
interface FastEthernet0/0.201999
 encapsulation dot1Q 201 second-dot1q any
 pppoe enable
!
interface FastEthernet0/0.2012001
 encapsulation dot1Q 201 second-dot1q 2001
 ip address 10.8.8.9 255.255.255.252
!
interface FastEthernet0/0.2012002
 encapsulation dot1Q 201 second-dot1q 2002
 ip address 10.8.8.13 255.255.255.252
!
interface FastEthernet0/0.4019999
 encapsulation dot1Q 401 second-dot1q 100-900,1001-2000
 pppoe enable
!
interface GigabitEthernet5/0.101
 encapsulation dot1Q 101
 ip address 10.7.7.1 255.255.255.252
!
interface GigabitEthernet5/0.301
 encapsulation dot1Q 301
 ip address 10.7.7.9 255.255.255.252
!
```

```
interface GigabitEthernet5/0.301999
 encapsulation dot1Q 301 second-dot1q any
 pppoe enable
!
interface GigabitEthernet5/0.1011001
 encapsulation dot1Q 101 second-dot1q 1001
 ip address 10.8.8.1 255.255.255.252
!
interface GigabitEthernet5/0.1011002
 encapsulation dot1Q 101 second-dot1q 1002
 ip address 10.8.8.5 255.255.255.252
!
interface GigabitEthernet5/0.1019999
 encapsulation dot1Q 101 second-dot1q 1-1000,1003-2000
 pppoe enable
.
.
.
```

The following shows the currently running configuration on a Cisco 10000 series Internet router:

Example:

```
Router# show running-config
.
.
.
interface FastEthernet1/0/0.201
 encapsulation dot1Q 201
 ip address 10.7.7.5 255.255.255.252
!
interface FastEthernet1/0/0.401
 encapsulation dot1Q 401
 ip address 10.7.7.13 255.255.255.252
!
interface FastEthernet1/0/0.201999
 encapsulation dot1Q 201 second-dot1q any
 pppoe enable
!
interface FastEthernet1/0/0.4019999
 encapsulation dot1Q 401 second-dot1q 100-900,1001-2000
 pppoe enable
!
interface GigabitEthernet5/0/0.101
 encapsulation dot1Q 101
 ip address 10.7.7.1 255.255.255.252
!
interface GigabitEthernet5/0/0.301
 encapsulation dot1Q 301
 ip address 10.7.7.9 255.255.255.252
!
interface GigabitEthernet5/0/0.301999
 encapsulation dot1Q 301 second-dot1q any
 pppoe enable
!
interface GigabitEthernet5/0/0.1019999
 encapsulation dot1Q 101 second-dot1q 1-1000,1003-2000
 pppoe enable
.
.
.
```

Step 3 **show vlans dot1q** [**internal** | *interface-type interface-number* **.***subinterface-number*[**detail**] | *outer-id*[*interface-type interface-number* | **second-dot1q** [*inner-id*| **any**]] [**detail**]]

Use this command to show the statistics for all the 802.1Q VLAN IDs. In this example, only the outer VLAN ID is displayed.

Note The **show vlans dot1q**command is not supported on the Cisco 10000 series Internet router.

Example:

```
Router# show vlans dot1q
Total statistics for 802.1Q VLAN 1:
   441 packets, 85825 bytes input
   1028 packets, 69082 bytes output
Total statistics for 802.1Q VLAN 101:
   5173 packets, 510384 bytes input
   3042 packets, 369567 bytes output
Total statistics for 802.1Q VLAN 201:
   1012 packets, 119254 bytes input
   1018 packets, 120393 bytes output
Total statistics for 802.1Q VLAN 301:
   3163 packets, 265272 bytes input
   1011 packets, 120750 bytes output
Total statistics for 802.1Q VLAN 401:
   1012 packets, 119254 bytes input
   1010 packets, 119108 bytes output
```

Monitoring and Maintaining VLAN Subinterfaces

Use the following task to determine whether a VLAN is a native VLAN.

SUMMARY STEPS

1. **enable**
2. **show vlans**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show vlans** **Example:** `Router# show vlans`	Displays VLAN subinterfaces.

Monitoring and Maintaining VLAN Subinterfaces Example

The following is sample output from the **show vlans**command indicating a native VLAN and a bridged group:

```
Router# show vlans
Virtual LAN ID:  1 (IEEE 802.1Q Encapsulation)
   vLAN Trunk Interface:   FastEthernet1/0/2
 This is configured as native Vlan for the following interface(s) :
FastEthernet1/0/2
   Protocols Configured:   Address: Received:        Transmitted:
Virtual LAN ID:  100 (IEEE 802.1Q Encapsulation)
   vLAN Trunk Interface:   FastEthernet1/0/2.1
   Protocols Configured:   Address: Received:        Transmitted:
        Bridging        Bridge Group 1 0                  0
```

The following is sample output from the **show vlans**command that shows the traffic count on Fast Ethernet subinterfaces:

```
Router# show vlans
Virtual LAN ID:  2 (IEEE 802.1Q Encapsulation)
    vLAN Trunk Interface:   FastEthernet5/0.1

    Protocols Configured:   Address:              Received:        Transmitted:
           IP               172.16.0.3                  16               92129

Virtual LAN ID:  3 (IEEE 802.1Q Encapsulation)

    vLAN Trunk Interface:   Ethernet6/0/1.1

    Protocols Configured:   Address:              Received:        Transmitted:
           IP               172.20.0.3                1558               1521

Virtual LAN ID:  4 (Inter Switch Link Encapsulation)

    vLAN Trunk Interface:   FastEthernet5/0.2

    Protocols Configured:   Address:              Received:        Transmitted:
           IP               172.30.0.3                   0                  7
```

ConfigurationExamplesforConfiguringRoutingBetweenVLANs

Single Range Configuration Example

The following example configures the Fast Ethernet subinterfaces within the range 5/1.1 and 5/1.4 and applies the following VLAN IDs to those subinterfaces:

Fast Ethernet5/1.1 = VLAN ID 301 (vlan-id)

Fast Ethernet5/1.2 = VLAN ID 302 (vlan-id = 301 + 2 - 1 = 302)

Fast Ethernet5/1.3 = VLAN ID 303 (vlan-id = 301 + 3 - 1 = 303)

Fast Ethernet5/1.4 = VLAN ID 304 (vlan-id = 301 + 4 - 1 = 304)

```
Router(config)# interface range fastethernet5/1.1 - fastethernet5/1.4

Router(config-if)# encapsulation dot1Q 301
Router(config-if)# no shutdown

Router(config-if)#
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/1.1, changed state to up
```

```
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/1.2, changed state to up
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/1.3, changed state to up
*Oct  6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/1.4, changed state to up
*Oct  6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/1.1, changed
 state to up
*Oct  6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/1.2, changed
 state to up
*Oct  6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/1.3, changed
 state to up
*Oct  6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/1.4, changed
 state to up
```

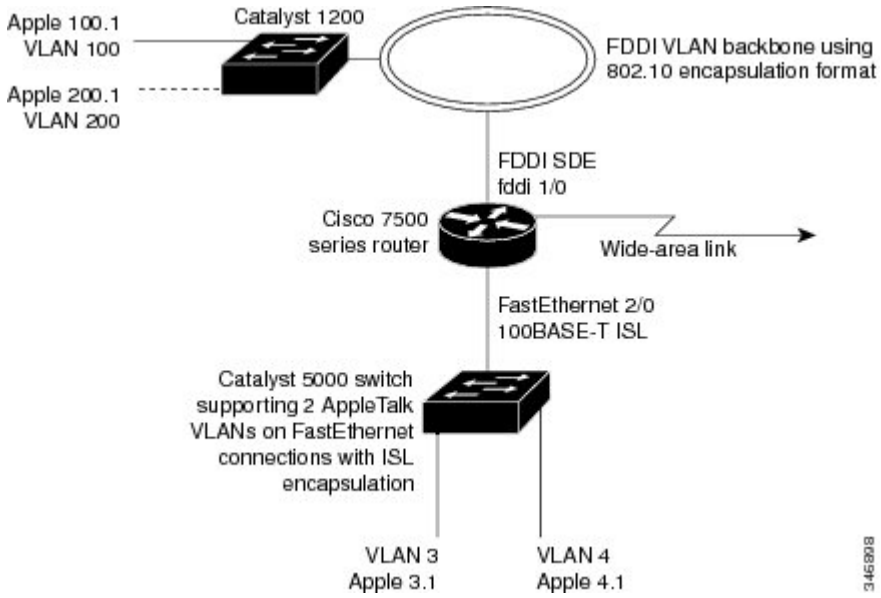
ISL Encapsulation Configuration Examples

This section provides the following configuration examples for each of the protocols described in this module:

AppleTalk Routing over ISL Configuration Example

The configuration example illustrated in the figure below shows AppleTalk being routed between different ISL and IEEE 802.10 VLAN encapsulating subinterfaces.

Figure 10: Routing AppleTalk over VLAN Encapsulations



As shown in the figure above, AppleTalk traffic is routed to and from switched VLAN domains 3, 4, 100, and 200 to any other AppleTalk routing interface. This example shows a sample configuration file for the Cisco 7500 series router with the commands entered to configure the network shown in the figure above.

Cisco 7500 Router Configuration

```
!
appletalk routing
interface Fddi 1/0.100
 encapsulation sde 100
 appletalk cable-range 100-100 100.2
 appletalk zone 100
```

```
!
interface Fddi 1/0.200
 encapsulation sde 200
 appletalk cable-range 200-200 200.2
 appletalk zone 200
!
interface FastEthernet 2/0.3
 encapsulation isl 3
 appletalk cable-range 3-3 3.2
 appletalk zone 3
!
interface FastEthernet 2/0.4
 encapsulation isl 4
 appletalk cable-range 4-4 4.2
 appletalk zone 4
!
```

Banyan VINES Routing over ISL Configuration Example

To configure routing of the Banyan VINES protocol over ISL trunks, you need to define ISL as the encapsulation type. This example shows Banyan VINES configured to be routed over an ISL trunk:

```
vines routing
interface fastethernet 0.1
 encapsulation isl 100
 vines metric 2
```

DECnet Routing over ISL Configuration Example

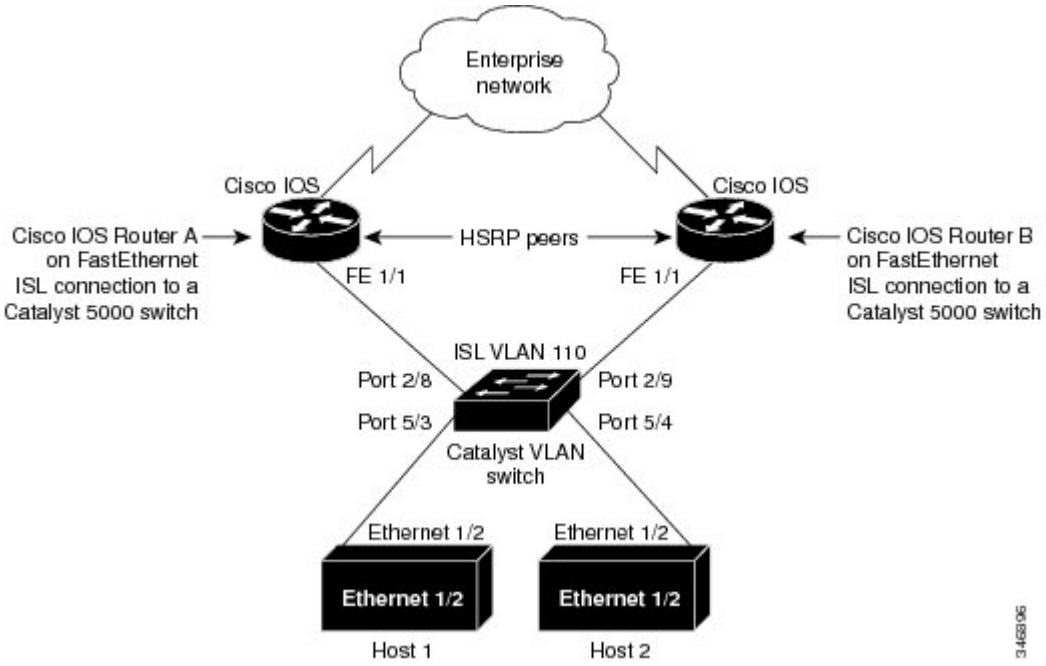
To configure routing the DECnet protocol over ISL trunks, you need to define ISL as the encapsulation type. This example shows DECnet configured to be routed over an ISL trunk:

```
decnet routing 2.1
interface fastethernet 1/0.1
 encapsulation isl 200
 decnet cost 4
```

HSRP over ISL Configuration Example

The configuration example shown in the figure below shows HSRP being used on two VLAN routers sending traffic to and from ISL VLANs through a Catalyst 5000 switch. Each router forwards its own traffic and acts as a standby for the other.

Figure 11: Hot Standby Router Protocol Sample Configuration



The topology shown in the figure above shows a Catalyst VLAN switch supporting Fast Ethernet connections to two routers running HSRP. Both routers are configured to route HSRP over ISLs.

The standby conditions are determined by the standby commands used in the configuration. Traffic from Host 1 is forwarded through Router A. Because the priority for the group is higher, Router A is the active router for Host 1. Because the priority for the group serviced by Host 2 is higher in Router B, traffic from Host 2 is forwarded through Router B, making Router B its active router.

In the configuration shown in the figure above, if the active router becomes unavailable, the standby router assumes active status for the additional traffic and automatically routes the traffic normally handled by the router that has become unavailable.

Host 1 Configuration

```
interface Ethernet 1/2
 ip address 10.1.1.25 255.255.255.0
 ip route 0.0.0.0 0.0.0.0 10.1.1.101
```

Host 2 Configuration

```
interface Ethernet 1/2
 ip address 10.1.1.27 255.255.255.0
```

```
 ip route 0.0.0.0 0.0.0.0 10.1.1.102
!
```

Router A Configuration

```
interface FastEthernet 1/1.110
 encapsulation isl 110
 ip address 10.1.1.2 255.255.255.0
 standby 1 ip 10.1.1.101
 standby 1 preempt
 standby 1 priority 105
 standby 2 ip 10.1.1.102
 standby 2 preempt
!
end
!
```

Router B Configuration

```
interface FastEthernet 1/1.110
 encapsulation isl 110
 ip address 10.1.1.3 255.255.255.0
 standby 1 ip 10.1.1.101
 standby 1 preempt
 standby 2 ip 10.1.1.102
 standby 2 preempt
 standby 2 priority 105
router igrp 1
!
network 10.1.0.0
network 10.2.0.0
!
```

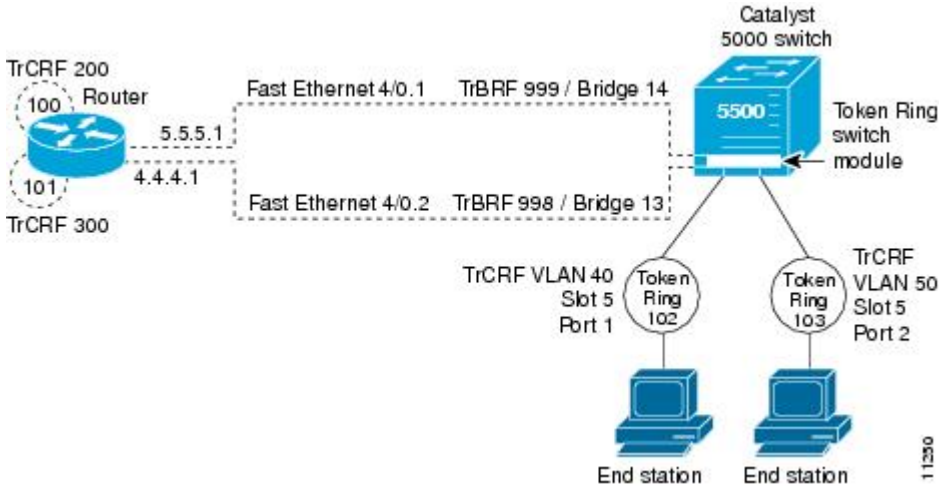
VLAN Switch Configuration

```
set vlan 110 5/4
set vlan 110 5/3
set trunk 2/8 110
set trunk 2/9 110
```

IP Routing with RIF Between TrBRF VLANs Example

The figure below shows IP routing with RIF between two TrBRF VLANs.

Figure 12: IP Routing with RIF Between TrBRF VLANs



The following is the configuration for the router:

```
interface FastEthernet4/0.1
 ip address 10.5.5.1 255.255.255.0
 encapsulation tr-isl trbrf-vlan 999 bridge-num 14
 multiring trcrf-vlan 200 ring 100
 multiring all
!
 interface FastEthernet4/0.2
 ip address 10.4.4.1 255.255.255.0
 encapsulation tr-isl trbrf-vlan 998 bridge-num 13
 multiring trcrf-vlan 300 ring 101
 multiring all
```

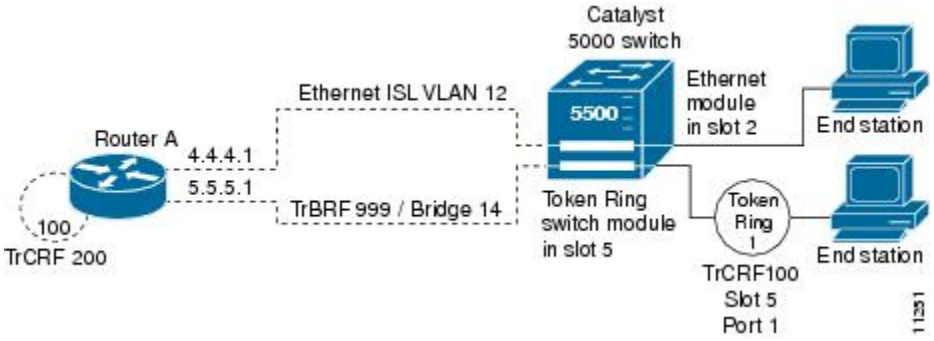
The following is the configuration for the Catalyst 5000 switch with the Token Ring switch module in slot 5. In this configuration, the Token Ring port 102 is assigned with TrCRF VLAN 40 and the Token Ring port 103 is assigned with TrCRF VLAN 50:

```
#vtp
set vtp domain trisl
set vtp mode server
set vtp v2 enable
#drip
set set tokenring reduction enable
set tokenring distrib-crf disable
#vlans
set vlan 999 name trbrf type trbrf bridge 0xe stp ieee
set vlan 200 name trcrf200 type trcrf parent 999 ring 0x64 mode srb
set vlan 40 name trcrf40 type trcrf parent 999 ring 0x66 mode srb
set vlan 998 name trbrf type trbrf bridge 0xd stp ieee
set vlan 300 name trcrf300 type trcrf parent 998 ring 0x65 mode srb
set vlan 50 name trcrf50 type trcrf parent 998 ring 0x67 mode srb
#add token port to trcrf 40
set vlan 40   5/1
#add token port to trcrf 50
set vlan 50   5/2
set trunk 1/2 on
```

IP Routing Between a TRISL VLAN and an Ethernet ISL VLAN Example

The figure below shows IP routing between a TRISL VLAN and an Ethernet ISL VLAN.

Figure 13: IP Routing Between a TRISL VLAN and an Ethernet ISL VLAN



The following is the configuration for the router:

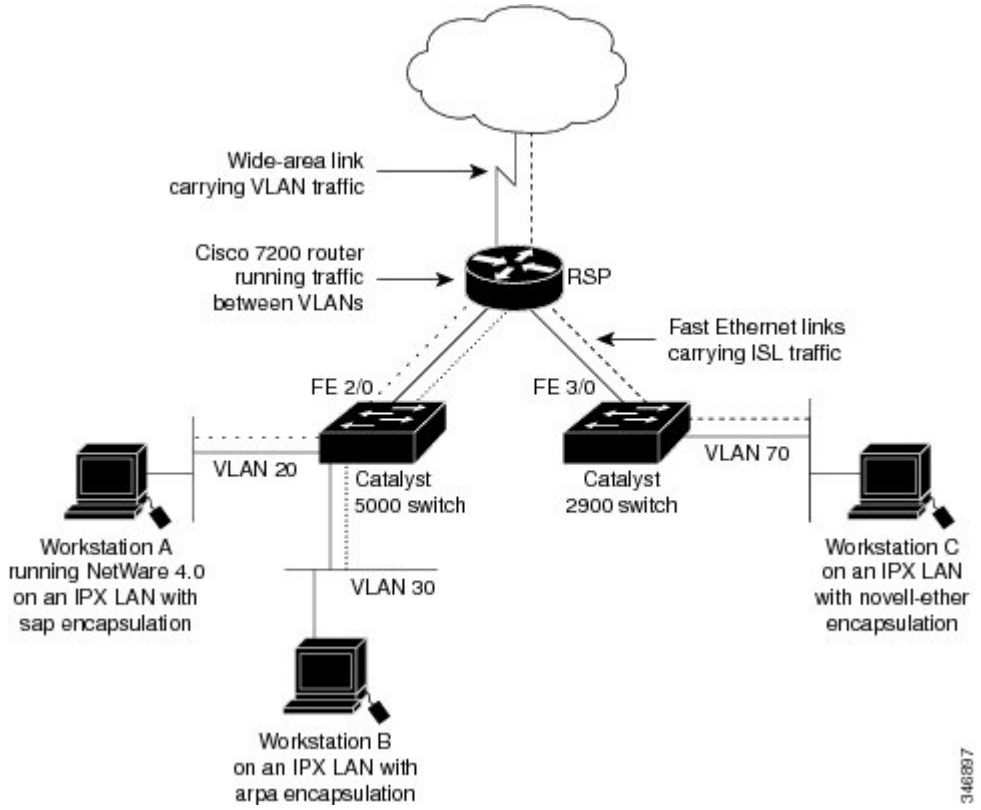
```
interface FastEthernet4/0.1
 ip address 10.5.5.1 255.255.255.0
 encapsulation tr-isl trbrf-vlan 999 bridge-num 14
 multiring trcrf-vlan 20 ring 100
 multiring all
!
interface FastEthernet4/0.2
 ip address 10.4.4.1 255.255.255.0
 encapsulation isl 12
```

IPX Routing over ISL Configuration Example

The figure below shows IPX interior encapsulations configured over ISL encapsulation in VLAN configurations. Note that three different IPX encapsulation formats are used. VLAN 20 uses SAP encapsulation, VLAN 30

uses ARPA, and VLAN 70 uses novell-ether encapsulation. Prior to the introduction of this feature, only the default encapsulation format, “novell-ether,” was available for routing IPX over ISL links in VLANs.

Figure 14: Configurable IPX Encapsulations Routed over ISL in VLAN Configurations



VLAN 20 Configuration

```
ipx routing
interface FastEthernet 2/0
 no shutdown
interface FastEthernet 2/0.20
 encapsulation isl 20
 ipx network 20 encapsulation sap
```

VLAN 30 Configuration

```
ipx routing
interface FastEthernet 2/0
 no shutdown
interface FastEthernet 2/0.30
 encapsulation isl 30
 ipx network 30 encapsulation arpa
```

VLAN 70 Configuration

```
ipx routing
interface FastEthernet 3/0
 no shutdown
```

```
interface Fast3/0.70
 encapsulation isl 70
 ipx network 70 encapsulation novell-ether
```

IPX Routing on FDDI Interfaces with SDE Example

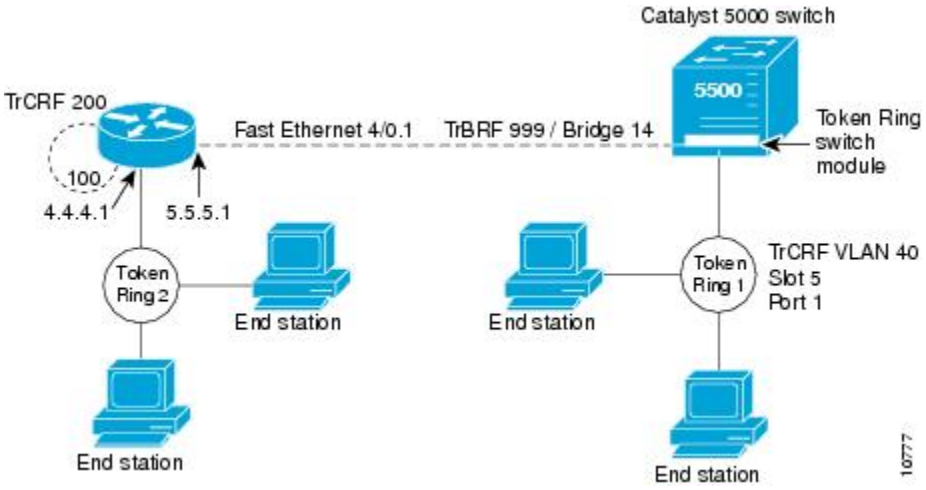
The following example enables IPX routing on FDDI interfaces 0.2 and 0.3 with SDE. On FDDI interface 0.2, the encapsulation type is SNAP. On FDDI interface 0.3, the encapsulation type is Novell's FDDI_RAW.

```
ipx routing
interface fddi 0.2 enc sde 2
 ipx network f02 encapsulation snap
interface fddi 0.3 enc sde 3
 ipx network f03 encapsulation novell-fddi
```

Routing with RIF Between a TRISL VLAN and a Token Ring Interface Example

The figure below shows routing with RIF between a TRISL VLAN and a Token Ring interface.

Figure 15: Routing with RIF Between a TRISL VLAN and a Token Ring Interface



The following is the configuration for the router:

```
source-bridge ring-group 100
!
interface TokenRing 3/1
 ip address 10.4.4.1 255.255.255.0
!
interface FastEthernet4/0.1
 ip address 10.5.5.1 255.255.255.0
 encapsulation tr-isl trbrf 999 bridge-num 14
 multiring trcrf-vlan 200 ring-group 100
 multiring all
```

The following is the configuration for the Catalyst 5000 switch with the Token Ring switch module in slot 5. In this configuration, the Token Ring port 1 is assigned to the TrCRF VLAN 40:

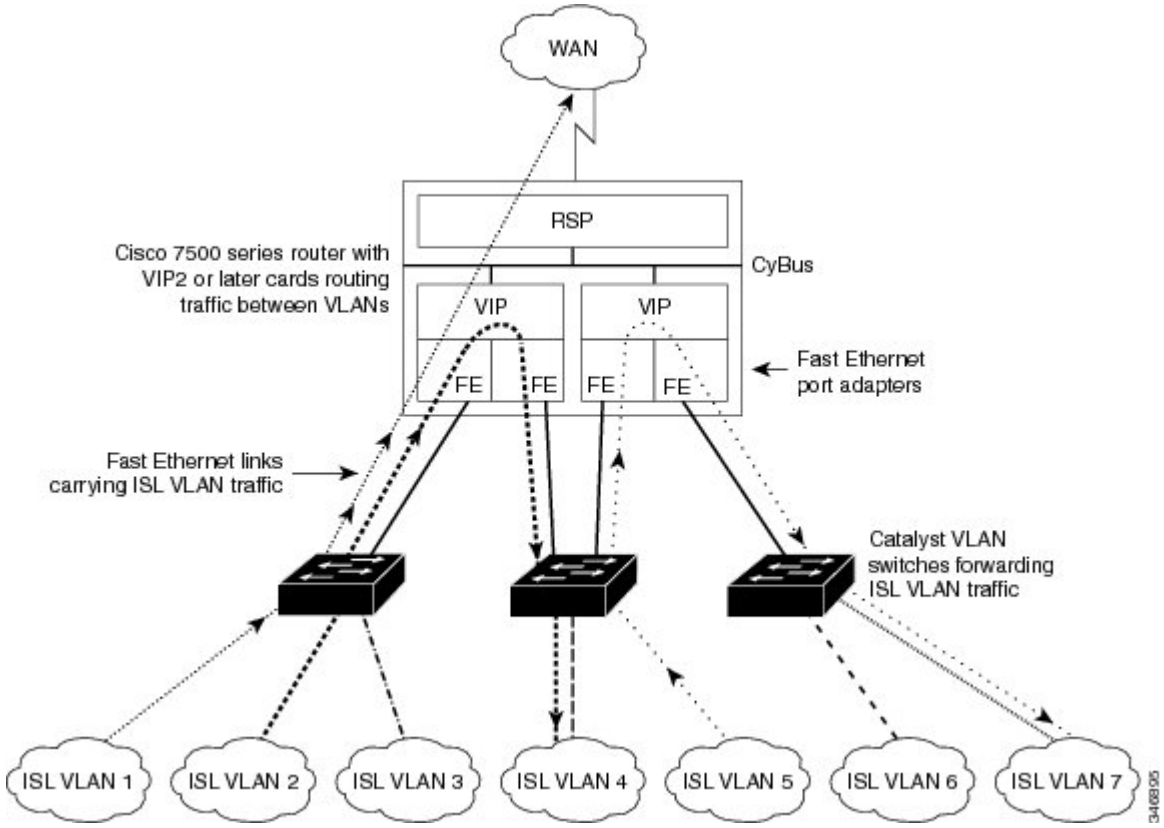
```
#vtp
set vtp domain trisl
set vtp mode server
set vtp v2 enable
```

```
#drip
set set tokenring reduction enable
set tokenring distrib-crf disable
#vlans
set vlan 999 name trbrf type trbrf bridge 0xe stp ieee
set vlan 200 name trcrf200 type trcrf parent 999 ring 0x64 mode srt
set vlan 40 name trcrf40 type trcrf parent 999 ring 0x1 mode srt
#add token port to trcrf 40
set vlan 40   5/1
set trunk 1/2 on
```

VIP Distributed Switching over ISL Configuration Example

The figure below shows a topology in which Catalyst VLAN switches are connected to routers forwarding traffic from a number of ISL VLANs. With the VIP distributed ISL capability in the Cisco 7500 series router, each VIP card can route ISL-encapsulated VLAN IP traffic. The inter-VLAN routing capacity is increased linearly by the packet-forwarding capability of each VIP card.

Figure 16: VIP Distributed ISL VLAN Traffic



In the figure above, the VIP cards forward the traffic between ISL VLANs or any other routing interface. Traffic from any VLAN can be routed to any of the other VLANs, regardless of which VIP card receives the traffic.

These commands show the configuration for each of the VLANs shown in the figure above:

```
interface FastEthernet1/0/0
 ip address 10.1.1.1 255.255.255.0
```

```
 ip route-cache distributed
 full-duplex
interface FastEthernet1/0/0.1
 ip address 10.1.1.1 255.255.255.0
 encapsulation isl 1
interface FastEthernet1/0/0.2
 ip address 10.1.2.1 255.255.255.0
 encapsulation isl 2
interface FastEthernet1/0/0.3
 ip address 10.1.3.1 255.255.255.0
 encapsulation isl 3
interface FastEthernet1/1/0
 ip route-cache distributed
 full-duplex
interface FastEthernet1/1/0.1
 ip address 172.16.1.1 255.255.255.0
 encapsulation isl 4
interface Fast Ethernet 2/0/0
 ip address 10.1.1.1 255.255.255.0
 ip route-cache distributed
 full-duplex
interface FastEthernet2/0/0.5
 ip address 10.2.1.1 255.255.255.0
 encapsulation isl 5
interface FastEthernet2/1/0
 ip address 10.3.1.1 255.255.255.0
 ip route-cache distributed
 full-duplex
interface FastEthernet2/1/0.6
 ip address 10.4.6.1 255.255.255.0
 encapsulation isl 6
interface FastEthernet2/1/0.7
 ip address 10.4.7.1 255.255.255.0
 encapsulation isl 7
```

XNS Routing over ISL Configuration Example

To configure routing of the XNS protocol over ISL trunks, you need to define ISL as the encapsulation type. This example shows XNS configured to be routed over an ISL trunk:

```
xns routing 0123.4567.adcb
interface fastethernet 1/0.1
 encapsulation isl 100
 xns network 20
```

CLNS Routing over ISL Configuration Example

To configure routing of the CLNS protocol over ISL trunks, you need to define ISL as the encapsulation type. This example shows CLNS configured to be routed over an ISL trunk:

```
clns routing
interface fastethernet 1/0.1
 encapsulation isl 100
 clns enable
```

IS-IS Routing over ISL Configuration Example

To configure IS-IS routing over ISL trunks, you need to define ISL as the encapsulation type. This example shows IS-IS configured over an ISL trunk:

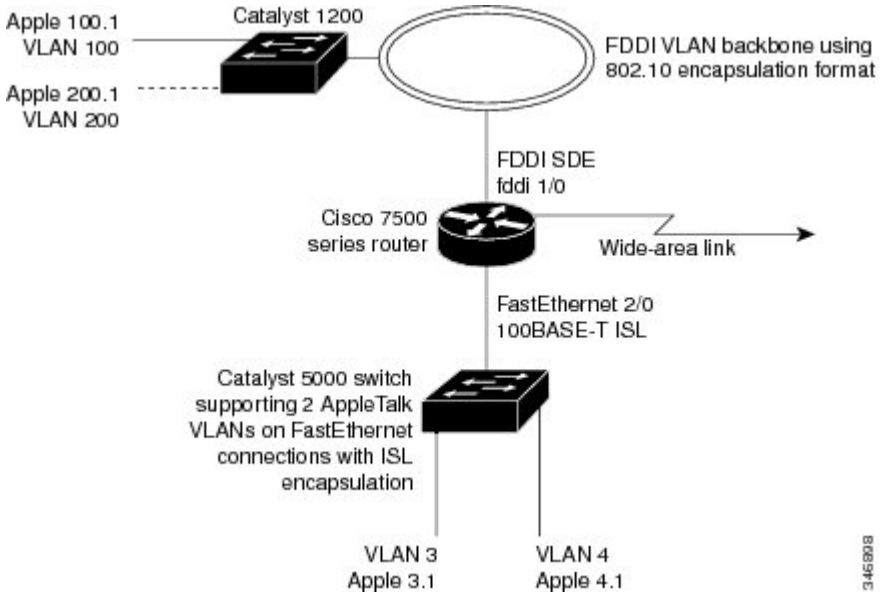
```
isis routing test-proc2
net 49.0001.0002.aaaa.aaaa.aaaa.00
```

```
interface fastethernet 2.0
 encapsulation isl 101
 clns router is-is test-proc2
```

Routing IEEE 802.10 Configuration Example

The figure below shows AppleTalk being routed between different ISL and IEEE 802.10 VLAN encapsulating subinterfaces.

Figure 17: Routing AppleTalk over VLAN encapsulations



As shown in the figure above, AppleTalk traffic is routed to and from switched VLAN domains 3, 4, 100, and 200 to any other AppleTalk routing interface. This example shows a sample configuration file for the Cisco 7500 series router with the commands entered to configure the network shown in the figure above.

Cisco 7500 Router Configuration

```
!
interface Fddi 1/0.100
 encapsulation sde 100
 appletalk cable-range 100-100 100.2
 appletalk zone 100
!
interface Fddi 1/0.200
 encapsulation sde 200
 appletalk cable-range 200-200 200.2
 appletalk zone 200
!
interface FastEthernet 2/0.3
 encapsulation isl 3
 appletalk cable-range 3-3 3.2
 appletalk zone 3
!
interface FastEthernet 2/0.4
 encapsulation isl 4
 appletalk cable-range 4-4 4.2
```

```
 appletalk zone 4
!
```

IEEE 802.1Q Encapsulation Configuration Examples

Configuration examples for each protocols are provided in the following sections:

Configuring AppleTalk over IEEE 802.1Q Example

This configuration example shows AppleTalk being routed on VLAN 100:

```
!
appletalk routing
!
interface fastethernet 4/1.100
  encapsulation dot1q 100
  appletalk cable-range 100-100 100.1
  appletalk zone eng
!
```

Configuring IP Routing over IEEE 802.1Q Example

This configuration example shows IP being routed on VLAN 101:

```
!
ip routing
!
interface fastethernet 4/1.101
  encapsulation dot1q 101
  ip addr 10.0.0.11 255.0.0.0
!
```

Configuring IPX Routing over IEEE 802.1Q Example

This configuration example shows IPX being routed on VLAN 102:

```
!
ipx routing
!
interface fastethernet 4/1.102
  encapsulation dot1q 102
  ipx network 100
!
```

VLAN 100 for Bridge Group 1 with Default VLAN1 Example

The following example configures VLAN 100 for bridge group 1 with a default VLAN1:

```
interface FastEthernet 4/1.100
encapsulation dot1q 1
bridge-group 1
```

VLAN 20 for Bridge Group 1 with Native VLAN Example

The following example configures VLAN 20 for bridge group 1 as a native VLAN:

```
interface FastEthernet 4/1.100
encapsulation dot1q 20 native
bridge-group 1
```

VLAN ISL or IEEE 802.1Q Routing Example

The following example configures VLAN ISL or IEEE 802.10 routing:

```
ipx routing
appletalk routing
!
interface Ethernet 1
ip address 10.1.1.1 255.255.255.0
appletalk cable-range 1-1 1.1
appletalk zone 1
ipx network 10 encapsulation snap
!
router igrp 1
network 10.1.0.0
!
end
!
#Catalyst5000
!
set VLAN 110 2/1
set VLAN 120 2/2
!
set trunk 1/1 110,120
# if 802.1Q, set trunk 1/1 nonegotiate 110, 120
!
end
!
ipx routing
appletalk routing
!
interface FastEthernet 1/1.110
encapsulation isl 110
!if 802.1Q, encapsulation dot1Q 110
ip address 10.1.1.2 255.255.255.0
appletalk cable-range 1.1 1.2
appletalk zone 1
ipx network 110 encapsulation snap
!
interface FastEthernet 1/1.120
encapsulation isl 120
!if 802.1Q, encapsulation dot1Q 120
ip address 10.2.1.2 255.255.255.0
appletalk cable-range 2-2 2.2
appletalk zone 2
ipx network 120 encapsulation snap
!
router igrp 1
network 10.1.0.0
network 10.2.1.0.0
!
end
!
ipx routing
appletalk routing
!
interface Ethernet 1
ip address 10.2.1.3 255.255.255.0
appletalk cable-range 2-2 2.3
```

```
appletalk zone 2
ipx network 120 encapsulation snap
!
router igrp 1
network 10.2.0.0
!
end
```

VLAN IEEE 802.1Q Bridging Example

The following examples configures IEEE 802.1Q bridging:

```
interface FastEthernet4/0
 no ip address
 no ip route-cache
 half-duplex
!
interface FastEthernet4/0.100
 encapsulation dot1Q 100
 no ip route-cache
 bridge-group 1
!
interface FastEthernet4/0.200
 encapsulation dot1Q 200 native
 no ip route-cache
 bridge-group 2
!
interface FastEthernet4/0.300
 encapsulation dot1Q 1
 no ip route-cache
 bridge-group 3
!
interface FastEthernet10/0
 no ip address
 no ip route-cache
 half-duplex
!
interface FastEthernet10/0.100
 encapsulation dot1Q 100
 no ip route-cache
 bridge-group 1
!
interface Ethernet11/3
 no ip address
 no ip route-cache
 bridge-group 2
!
interface Ethernet11/4
 no ip address
 no ip route-cache
 bridge-group 3
!
bridge 1 protocol ieee
bridge 2 protocol ieee
bridge 3 protocol ieee
```

VLAN IEEE 802.1Q IRB Example

The following examples configures IEEE 802.1Q integrated routing and bridging:

```
ip cef
appletalk routing
ipx routing 0060.2f27.5980
!
bridge irb
!
interface TokenRing3/1
```

```
 no ip address
 ring-speed 16
 bridge-group 2
!
interface FastEthernet4/0
 no ip address
 half-duplex
!
interface FastEthernet4/0.100
 encapsulation dot1Q 100
 bridge-group 1
!
interface FastEthernet4/0.200
 encapsulation dot1Q 200
 bridge-group 2
!
interface FastEthernet10/0
ip address 10.3.1.10 255.255.255.0
 half-duplex
 appletalk cable-range 200-200 200.10
 appletalk zone irb
 ipx network 200
!
interface Ethernet11/3
 no ip address
 bridge-group 1
!
interface BVI 1
 ip address 10.1.1.11 255.255.255.0
 appletalk cable-range 100-100 100.11
 appletalk zone bridging
 ipx network 100
!
router rip
 network 10.0.0.0
 network 10.3.0.0
!
bridge 1 protocol ieee
 bridge 1 route appletalk
 bridge 1 route ip
 bridge 1 route ipx
bridge 2 protocol ieee
!
```

Configuring IEEE 802.1Q-in-Q VLAN Tag Termination Example

Some ambiguous subinterfaces can use the **any** keyword for the inner VLAN ID specification. The **any** keyword represents any inner VLAN ID that is not explicitly configured on any other interface. In the following example, seven subinterfaces are configured with various outer and inner VLAN IDs.

Note The **any** keyword can be configured on only one subinterface of a specified physical interface and outer VLAN ID.

```
interface GigabitEthernet1/0/0.1
 encapsulation dot1q 100 second-dot1q 100
interface GigabitEthernet1/0/0.2
 encapsulation dot1q 100 second-dot1q 200
interface GigabitEthernet1/0/0.3
 encapsulation dot1q 100 second-dot1q 300-400,500-600
interface GigabitEthernet1/0/0.4
 encapsulation dot1q 100 second-dot1q any
interface GigabitEthernet1/0/0.5
 encapsulation dot1q 200 second-dot1q 50
interface GigabitEthernet1/0/0.6
```

```
 encapsulation dot1q 200 second-dot1q 1000-2000,3000-4000
interface GigabitEthernet1/0/0.7
 encapsulation dot1q 200 second-dot1q any
```

The table below shows which subinterfaces are mapped to different values of the outer and inner VLAN ID on Q-in-Q frames that come in on Gigabit Ethernet interface 1/0/0.

Table 2: Subinterfaces Mapped to Outer and Inner VLAN IDs for GE Interface 1/0/0

Outer VLAN ID	Inner VLAN ID	Subinterface mapped to
100	1 through 99	GigabitEthernet1/0/0.4
100	100	GigabitEthernet1/0/0.1
100	101 through 199	GigabitEthernet1/0/0.4
100	200	GigabitEthernet1/0/0.2
100	201 through 299	GigabitEthernet1/0/0.4
100	300 through 400	GigabitEthernet1/0/0.3
100	401 through 499	GigabitEthernet1/0/0.4
100	500 through 600	GigabitEthernet1/0/0.3
100	601 through 4095	GigabitEthernet1/0/0.4
200	1 through 49	GigabitEthernet1/0/0.7
200	50	GigabitEthernet1/0/0.5
200	51 through 999	GigabitEthernet1/0/0.7
200	1000 through 2000	GigabitEthernet1/0/0.6
200	2001 through 2999	GigabitEthernet1/0/0.7
200	3000 through 4000	GigabitEthernet1/0/0.6
200	4001 through 4095	GigabitEthernet1/0/0.7

A new subinterface is now configured:

```
interface GigabitEthernet1/0/0.8
 encapsulation dot1q 200 second-dot1q 200-600,900-999
```

The table below shows the changes made to the table for the outer VLAN ID of 200. Notice that subinterface 1/0/0.7 configured with the **any** keyword now has new inner VLAN ID mappings.

Table 3: Subinterfaces Mapped to Outer and Inner VLAN IDs for GE Interface 1/0/0--Changes Resulting from Configuring GE Subinterface 1/0/0.8

Outer VLAN ID	Inner VLAN ID	Subinterface mapped to
200	1 through 49	GigabitEthernet1/0/0.7
200	50	GigabitEthernet1/0/0.5
200	51 through 199	GigabitEthernet1/0/0.7
200	200 through 600	GigabitEthernet1/0/0.8
200	601 through 899	GigabitEthernet1/0/0.7
200	900 through 999	GigabitEthernet1/0/0.8
200	1000 through 2000	GigabitEthernet1/0/0.6
200	2001 through 2999	GigabitEthernet1/0/0.7
200	3000 through 4000	GigabitEthernet1/0/0.6
200	4001 through 4095	GigabitEthernet1/0/0.7

Additional References

The following sections provide references related to configuring a VLAN range.

Related Documents

Related Topic	Document Title
IP LAN switching commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS LAN Switching Command Reference
SNMP	Configuring SNMP Support module in the *Cisco IOS Network Management Configuration Guide*
HSRP	Configuring HSRP" module in the *Cisco IOS IP Application Services Configuration Guide*
Encapsulation types and corresponding framing types	Configuring Novell IPX module in the *Cisco IOS Novell IPX Configuration Guide*
AppleTalk	Configuring AppleTalk module in the *Cisco IOS AppleTalk Configuration Guide*

Standards

Standard	Title
IEEE 802.10 standard	802.10 Virtual LANs

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing standards has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Routing Between VLANs

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

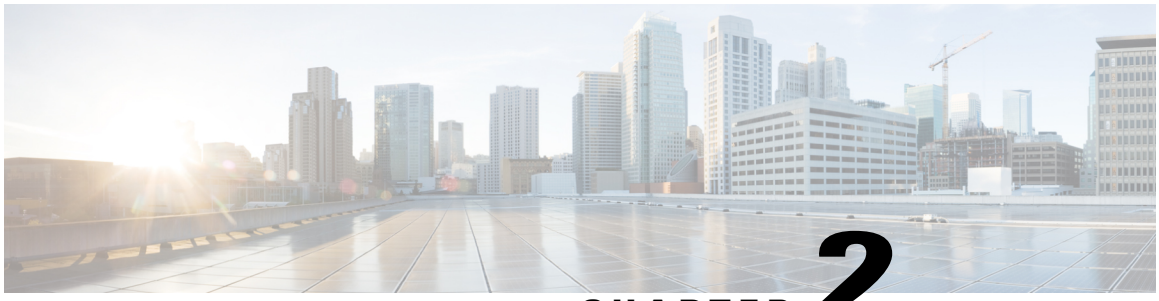
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 4: Feature Information for Routing Between VLANs

Feature Name	Releases	Feature Information
IEEE 802.1Q-in-Q VLAN Tag Termination	12.0(28)S, 12.3(7)(X17) 12.0(32)S1, 12.2(31)SB 12.3(7)T 12.3((7)XI1	Encapsulating IEEE 802.1Q VLAN tags within 802.1Q enables service providers to use a single VLAN to support customers who have multiple VLANs. The IEEE 802.1Q-in-Q VLAN Tag Termination feature on the subinterface level preserves VLAN IDs and keeps traffic in different customer VLANs segregated.
Configuring Routing Between VLANs with IEEE 802.1Q Encapsulation	12.0(7)XE 12.1(5)T 12.2(2)DD 12.2(4)B 12.2(8)T 12.2(13)T Cisco IOS XE 3.8(S) Cisco IOS XE 3.9(S)	The IEEE 802.1Q protocol is used to interconnect multiple switches and routers, and for defining VLAN topologies. The IEEE 802.1Q standard is extremely restrictive to untagged frames. The standard provides only a per-port VLANs solution for untagged frames. For example, assigning untagged frames to VLANs takes into consideration only the port from which they have been received. Each port has a parameter called a *permanent virtual identification* (Native VLAN) that specifies the VLAN assigned to receive untagged frames. In Cisco IOS XE Release 3.8(S), support was added for the Cisco ISR 4400 Series Routers. In Cisco IOS XE Release 3.9(S), support was added for the Cisco CSR 1000V Series Routers.

Feature Name	Releases	Feature Information
Configuring Routing Between VLANs with Inter-Switch Link Encapsulation	12.0(7)XE 12.1(5)T 12.2(2)DD 12.2(4)B 12.2(8)T 12.2(13)T	ISL is a Cisco protocol for interconnecting multiple switches and maintaining VLAN information as traffic goes between switches. ISL provides VLAN capabilities while maintaining full wire speed performance on Fast Ethernet links in full- or half-duplex mode. ISL operates in a point-to-point environment and will support up to 1000 VLANs. You can define virtually as many logical networks as are necessary for your environment.
Configuring Routing Between VLANs with IEEE 802.10 Encapsulation	12.0(7)XE 12.1(5)T 12.2(2)DD 12.2(4)B 12.2(8)T 12.2(13)T	AppleTalk can be routed over VLAN subinterfaces using the ISL or IEEE 802.10 VLANs feature that provides full-feature Cisco IOS software AppleTalk support on a per-VLAN basis, allowing standard AppleTalk capabilities to be configured on VLANs.

Feature Name	Releases	Feature Information
VLAN Range	12.0(7)XE 12.1(5)T 12.2(2)DD 12.2(4)B 12.2(8)T 12.2(13)T	Using the VLAN Range feature, you can group VLAN subinterfaces together so that any command entered in a group applies to every subinterface within the group. This capability simplifies configurations and reduces command parsing. In Cisco IOS Release 12.0(7)XE, the **interface range** command was introduced. The **interface range** command was integrated into Cisco IOS Release 12.1(5)T. In Cisco IOS Release 12.2(2)DD, the **interface range** command was expanded to enable configuration of subinterfaces. The **interface range** command was integrated into Cisco IOS Release 12.2(4)B. The VLAN Range feature was integrated into Cisco IOS Release 12.2(8)T. This VLAN Range feature was integrated into Cisco IOS Release 12.2(13)T.
256+ VLANS	12.1(2)E, 12.2(8)T Cisco IOS XE 3.8(S) Cisco IOS XE 3.9(S)	The 256+ VLAN feature enables a device to route more than 256 VLAN interfaces. This feature requires the MSFC2. The routed VLAN interfaces can be chosen from any of the VLANs supported on the device. Catalyst switches can support up to 4096 VLANs. If MSFC is used, up to 256 VLANs can be routed, but this can be selected from any VLANs supported on the device. In Cisco IOS XE Release 3.8(S), support was added for the Cisco ISR 4400 Series Routers. In Cisco IOS XE Release 3.9(S), support was added for the Cisco CSR 1000V Series Routers.



CHAPTER 2

Resilient Ethernet Protocol (REP)

The Resilient Ethernet Protocol (REP) is a Cisco proprietary protocol that provides an alternative to the Spanning Tree Protocol (STP). REP provides a way to control network loops, handle link failures, and improve convergence time. It controls a group of ports connected in a segment, ensures that the segment does not create any bridging loops, and responds to link failures within the segment. REP provides a basis for constructing complex networks and supports VLAN load balancing.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Resilient Ethernet Protocol

- The router supports REP only when the router is running the metro IP access or the metro access image.
- You must configure each segment port; an incorrect configuration can cause forwarding loops in networks.

- REP can manage only a single failed port within the segment; multiple port failures within the REP segment causes high loss of network connectivity.
- You should configure REP only in networks with redundancy. Configuring REP in a network without redundancy causes loss of network connectivity.

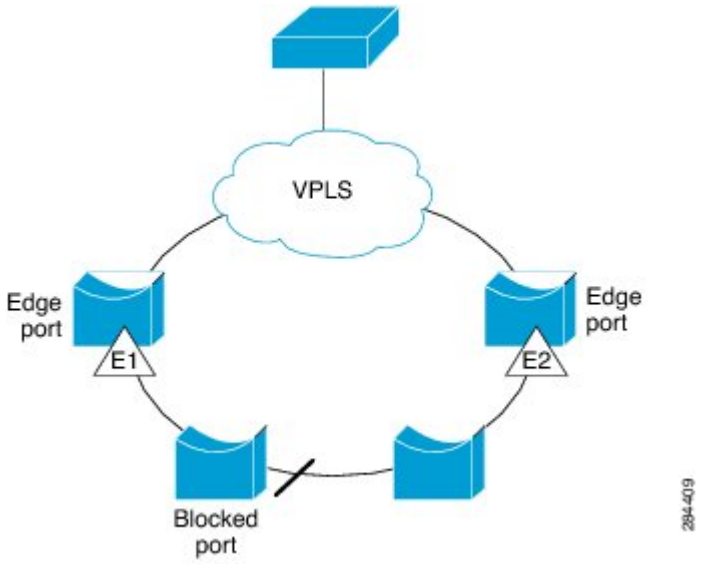
Information About REP

REP Segments

A REP segment is a chain of ports connected to each other and configured with a segment ID. Each segment consists of standard (nonedge) segment ports and two user-configured edge ports. A router can have no more than two ports that belong to the same segment, and each segment port can have only one external neighbor. A segment can go through a shared medium, but on any link, only two ports can belong to the same segment. REP is supported only on Trunk Ethernet Flow Point (EFP) interfaces.

The figure below shows an example of a segment consisting of six ports spread across four switches. Ports E1 and E2 are configured as edge ports. When all ports are operational (as in the segment on the left), a single port is blocked, shown by the diagonal line. When there is a failure in the network, the blocked port returns to the forwarding state to minimize network disruption.

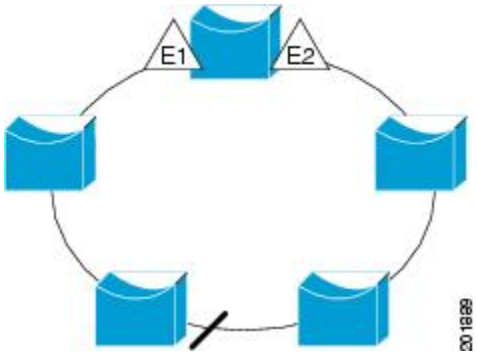
Figure 18: REP Open Segments



The segment shown in the figure above is an open segment; there is no connectivity between the two edge ports. The REP segment cannot cause a bridging loop, and you can safely connect the segment edges to any network. All hosts connected to routers inside the segment have two possible connections to the rest of the network through the edge ports, but only one connection is accessible at any time. If a failure occurs on any segment or on any port on a REP segment, REP unblocks all ports to ensure that connectivity is available through the other gateway.

The segment shown in the figure below is a ring segment, and it has both edge ports located on the same router. With this configuration, you can create a redundant connection between any two routers in the segment.

Figure 19: REP Ring Segment



REP segments have the following characteristics:

- If all ports in a segment are operational, one port (referred to as the *alternate* port) is in the blocked state for each VLAN. If VLAN load balancing is configured, two ports in the segment control the blocked state of VLANs.
- If one or more ports in a segment is not operational, and cause a link failure, all ports forward traffic on all VLANs to ensure connectivity.
- In case of a link failure, alternate ports are unblocked as quickly as possible. When the failed link is up, a logically blocked port per VLAN is selected with minimal disruption to the network.

You can construct almost any type of network based on REP segments. REP also supports VLAN load balancing, which is controlled by the primary edge port but can occurring at any port in the segment.

Link Integrity

REP does not use an end-to-end polling mechanism between edge ports to verify link integrity. It implements local link failure detection. When enabled on an interface, the REP Link Status Layer (LSL) detects its REP-aware neighbor and establishes connectivity within the segment. All VLANs are blocked on an interface until the REP LSL detects the neighbor. After the neighbor is identified, REP determines which neighbor port should become the alternate port and which ports should forward traffic.

Each port in a segment has a unique port ID. The port ID format is similar to that used by the spanning tree algorithm: a port number (unique on the bridge), associated to a MAC address (unique in the network). When a segment port is up, LSL sends packets that include the segment ID and the port ID. The port is declared as operational after it performs a three-way handshake with a neighbor in the same segment. A segment port does not become operational under the following conditions:

- No neighbor has the same segment ID.
- More than one neighbor has the same segment ID.
- The neighbor does not acknowledge the local port as a peer.

Each port creates an adjacency with its immediate neighbor. Once the neighbor adjacencies are created, the ports negotiate to determine one blocked port for the segment, which is the alternate port. All other ports become unblocked. By default, REP packets are sent to a PortFast Bridge Protocol Data Unit (BPDU) class MAC address. The packets can also be sent to the Cisco multicast address, which at present is used only to send blocked port advertisement (BPA) messages when there is a failure in the segment. The packets are dropped by devices not running REP.

Fast Convergence

Because REP runs on a physical-link basis and not on a per-VLAN basis, only one hello message is required for all VLANs, thus reducing the load on the protocol. We recommend that you create VLANs consistently on all switches in a given segment and configure VLANs on REP trunk ports. To avoid the delay introduced by relaying messages in software, REP also allows some packets to be flooded to a regular multicast address. These messages operate at the hardware flood layer (HFL) and are flooded to the whole network, not just the REP segment. Switches that do not belong to the segment treat the messages as data traffic. You can control flooding of these messages by configuring a dedicated administrative VLAN for the whole domain.

The estimated convergence recovery time is less than 200 milliseconds (ms) for the local segment.

VLAN Load Balancing

One edge port in a REP segment acts as the primary edge port and the other as the secondary edge port. It is the primary edge port that always participates in VLAN load balancing in the segment. REP VLAN load balancing is achieved by blocking some VLANs at a configured alternate port and all other VLANs at the primary edge port. When you configure VLAN load balancing, you can specify the alternate port using any one of the following ways:

- By entering the port ID of the interface. To identify the port ID of a port in the segment, enter the **show interface rep detail** command for the port.
- By entering the neighbor offset number of a port in the segment, which identifies the downstream neighbor port of an edge port. The neighbor offset number range is -256 to +256; a value of 0 is invalid. The primary edge port has an offset number of 1; positive numbers above 1 identify downstream neighbors of the primary edge port. Negative numbers indicate the secondary edge port (offset number -1) and its downstream neighbors.

  Note You configure offset numbers on the primary edge port by identifying a port's downstream position from the primary (or secondary) edge port. You cannot enter an offset value of 1 because 1 is the offset number of the primary edge port .

- By entering the **preferred** keyword to select the port that you previously configured as the preferred alternate port in the **rep segment preferred** command.

When the REP segment is complete, all VLANs are blocked. VLAN load balancing can be triggered in one of the following two ways:

- You can manually trigger VLAN load balancing at any time by entering the **rep preempt segment** *segment-id* command on the router that has the primary edge port.

- You can configure a preempt delay time by entering the **rep preempt delay** *seconds* command. After a link failure and recovery, VLAN load balancing begins after the configured preemption time period elapses. The delay timer restarts if another port fails before the time has elapsed.

Note A VLAN load balancing does not start working until triggered by either a manual intervention or a link failure and recovery.

When VLAN load balancing is triggered, the primary edge port sends out a message to alert all interfaces in the segment about the preemption. When the message is received by the secondary edge port, a message is generated in the network to notify the alternate port to block the set of VLANs specified in the message and to notify the primary edge port to block the remaining VLANs.

You can also configure a particular port in the segment to block all VLANs. VLAN load balancing is initiated only by the primary edge port and is not possible if the segment is not terminated by an edge port on each end. The primary edge port determines the local VLAN load balancing configuration.

To reconfigure VLAN load balancing, you must reconfigure the primary edge port. When you change the VLAN-load balancing configuration, the primary edge port again waits for the **rep preempt segment** command or for the configured preempt delay period after a port failure and recovery before executing the new VLAN load balancing configuration. If you change an edge port to a regular segment port, the existing VLAN load balancing status does not change. Configuring a new edge port might cause a new topology configuration.

Spanning Tree Protocol Interaction

REP does not interact with STP or with Flex Links but can coexist with both of them. A port that belongs to a segment is removed from spanning tree control, and STP BPDUs are not accepted or sent from segment ports. Therefore, STP cannot run on a segment.

To migrate from an STP ring configuration to a REP segment configuration, begin by configuring a single port in the ring as part of the segment and continue by configuring contiguous ports to minimize the number of segments. Each segment always contains a blocked port, so multiple segments mean multiple blocked ports and a potential loss of connectivity. You can configure the edge ports when the segment has been configured in both directions up to the location of the edge ports.

REP Ports

Ports in REP segments take one of following three roles or states: Failed, Open, or Alternate.

- A port configured as a regular segment port starts as a failed port.
- After neighbor adjacencies are determined, the port transitions to the alternate port state, blocking all VLANs on the interface. Blocked port negotiations occur, and when the segment settles, one blocked port remains in the alternate role, and all other ports become open ports.
- When a failure occurs in a link, all ports move to the failed state. When the alternate port receives the failure notification, the port changes to the open state forwarding all VLANs.

A regular segment port converted to an edge port, or an edge port converted to a regular segment port, does not always result in a topology change. If you convert an edge port into a regular segment port, VLAN load

balancing is not implemented unless it has been configured. For VLAN load balancing, you must configure two edge ports in the segment.

A segment port that is reconfigured as a spanning tree port restarts according to the spanning tree configuration. By default, this port is a designated blocking port. If the PortFast BPDU Guard Enhancement feature is configured or if STP is disabled, the port goes into the forwarding state.

REP Integrated with VPLS

Normally, in a Virtual Private LAN Service (VPLS) network core, all nodes are connected in a full-mesh topology and each node has connectivity to all other nodes. In the full-mesh topology, there is no need for a node to retransmit data to another node. In Figure 3, the common ring provides a path where the packet can be forwarded to another network provider edge (N-PE) router, breaking split horizon model.

REP emulates a common link connection the REP ring supports the VPLS full-mesh model, but maintains the split horizon properties so the super-loop does not exist. The emulated common link uses the Clustering over the WAN (CWAN) line card, which is also used for the VPLS uplink. This emulated common link forwards data from the ring to either the VPLS uplink or to the other side of the ring; blocks data coming from the VPLS core network; and handles access to pseudowire for Hierarchical-VPLS (H-VPLS) topologies.

Default REP Configuration

REP is disabled on all interfaces. When enabled, the interface is a regular segment port unless it is configured as an edge port.

When REP is enabled, the sending of segment topology change notices (STCNs) is disabled, all VLANs are blocked, and the administrative VLAN is VLAN 1.

When VLAN load balancing is enabled, the default is manual preemption with the delay timer disabled. If VLAN load balancing is not configured, the default after manual preemption is to block all VLANs at the primary edge port.

REP Segments and REP Administrative VLANs

A segment is a collection of ports connected in a chain and configured with a segment ID. To configure REP segments, you should configure the REP administrative VLAN (or use the default VLAN 1) and then add ports to the segment in interface configuration mode. You should configure two edge ports in the segment, with one as the primary edge port and the other, by default, as the secondary edge port. A segment has only one primary edge port. If you configure two ports in a segment as primary edge ports, for example, ports on different switches, REP selects one of them to serve as the primary edge port. You can also optionally configure where to send segment STCNs and VLAN load balancing. For more information about configuring REP Administrative VLANs, see the *Configuring the REP Administrative VLAN* section.

REP Configuration Guidelines

Follow these guidelines when configuring REP:

- We recommend that you begin by configuring one port and then configure contiguous ports to minimize the number of segments and the number of blocked ports.

- If more than two ports in a segment fail when no external neighbors are configured, one port goes into a forwarding state for the data path to help maintain connectivity during configuration. In the **show rep interface** command output, the Port Role for this port shows as “Fail Logical Open”; the Port Role for the other failed port shows as “Fail No Ext Neighbor”. When the external neighbors for the failed ports are configured, the ports go through the alternate port state transitions and eventually go to an open state or remain as the alternate port, based on the alternate port selection mechanism.
- REP ports must be Layer 2 IEEE 802.1Q or Trunk EFP ports.
- We recommend that you configure all trunk ports in the segment with the same set of allowed VLANs.
- Be careful when configuring REP through a Telnet connection. Because REP blocks all VLANs until another REP interface sends a message to unblock it. You might lose connectivity to the router if you enable REP in a Telnet session that accesses the router through the same interface.
- You cannot run REP and STP on the same segment or interface.
- If you connect an STP network to a REP segment, be sure that the connection is at the segment edge. An STP connection that is not at the edge could cause a bridging loop because STP does not run on REP segments. All STP BPDUs are dropped at REP interfaces.
- If REP is enabled on two ports on a router, both ports must be either regular segment ports or edge ports. REP ports follow these rules:
  - If only one port on a router is configured in a segment, the port should be an edge port.
  - If two ports on a router belong to the same segment, both ports must be edge ports or must be regular segment ports.
  - If two ports on a router belong to the same segment and one is configured as an edge port and the other as a regular segment port (a misconfiguration), the edge port is treated as a regular segment port.
- REP interfaces come up in a blocked state and remain in a blocked state until they are safe to be unblocked. You need to be aware of this status to avoid sudden connection losses.
- REP ports cannot be configured as one of the following port types:
  - Switched Port Analyzer (SPAN) destination port
  - Tunnel port
  - Access port
- There can be a maximum of 22 REP segments per router.

REP Support on a Trunk EFP

Resilient Ethernet Protocol (REP) can be configured on Trunk EFP ports at the interface level on Cisco ASR 903 Series Routers. Trunk EFP ports can have several bridged VLAN services running on them. VLANs can be set to blocking and forwarding state on a Trunk EFP port. A user must enable REP on a port. By default, REP is disabled on all ports.

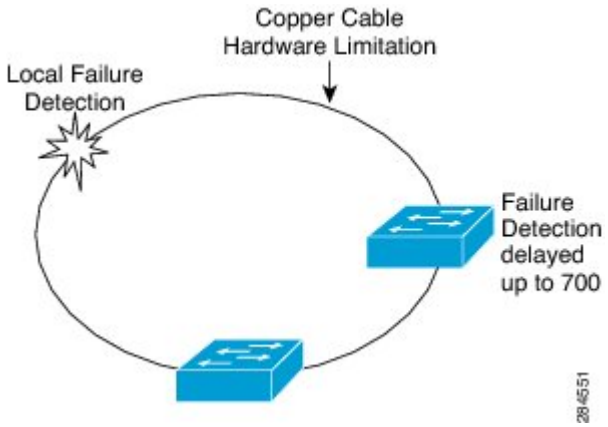
REP Configurable Timers

In a ring network topology, the Fast Last Link Status (LSL) process detects a neighboring port and maintains a connection with it. The timer on a port can be configured within 200-10000 ms to receive LSL frames. If no LSL frames are received from 200 to10000 ms from the neighboring port, the link between routers is considered as down. The tear-down operation and action is taken to bring up the link and restore traffic.

In the ring network topology, REP might fail to converge the traffic within 50 ms. For example, if the topology is made of copper cable, REP might fail to converge the traffic due to hardware limitations of the copper interface. In such a scenario, a remote end can take up to 700 ms to detect shutdown failure of a local port. The REP LSL is enhanced to achieve higher timer granularity and faster failure detection on the remote side.

The figure below shows the delay in failure detection due to hardware limitation of a Copper interface.

Figure 20: Delay in Failure Detection



SSO Support for REP Fast Hello

When a router crashes, it takes between 3 to 5 seconds for the router to get into active mode and start sending REP Fast Hello packets. If the value of the age out timer configured by the **lsl age out timer** command is less than 3 seconds, the remote end detects a port failure and reconverges. After reconverging, the router sends out a BPDU with a special type, length, and, value (TLV) to the connected port. The router learns the port's local and remote sequence number so that the subsequent REP three-way link integrity check does not fail. The Stateful Switchover (SSO) support for REP ensures that a Fast Hello packet can be sent from the router before the LSL interval expires.

REP Edge No-Neighbor Support

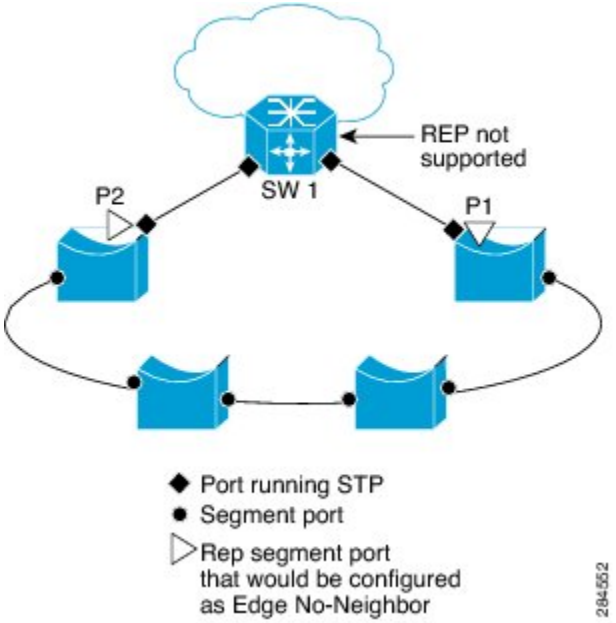
In a ring network topology, aggregation nodes do not support REP. A REP segment can be created with no-neighbor ports to achieve convergence of switches. The figure below shows P1 and P2 as Edge No-Neighbor ports in a ring topology. In this configuration P1 and P2 can block traffic. If there is a failure on any of the links, all the switches with REP configuration converge. Since P1 and P2 are not edges, they do not support the following tasks:

- Perform VLAN load balancing.

- Detect topology changes to other segments and the Spanning Tree Protocol (STP).
- Choose the port that can preempt.
- Display the complete segment topology.

The Edge No-Neighbor support enables defining a new type of edge that has an internal neighbor. In the figure below, P1 and P2 are configured as Edge No-Neighbor ports rather than intermediate segment ports. These ports inherit properties of edge ports and overcome the limitations listed above. Thus, the Edge No-Neighbor port (P1 or P2) can send the Multiple Spanning Tree (MST) protocol, a Topology Change Notification (TCN), and a REP TCN for another segment towards the aggregation switch.

Figure 21: Ring Topology with Edge No-Neighbor Ports



How to Configure REP

Configuring the REP Administrative VLAN

To avoid the delay introduced by relaying messages that are related to link-failures or VLAN-blocking notifications during VLAN load balancing, REP floods packets at the hardware flood layer (HFL) to a regular multicast address. These messages are flooded to the whole network and not just the REP segment. You can control flooding of these messages by configuring an administrative VLAN for the whole domain.

Follow these guidelines when configuring the REP administrative VLAN:

- There can be only one administrative VLAN on a router and on a segment. However, this is not enforced by the software.
- If you do not configure an administrative VLAN, the default is VLAN 1.

- If you want to configure REP on an interface, ensure that the REP administrative VLAN is part of the Trunk EFP encapsulation list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **rep admin vlan** *vlan-id*
4. **end**
5. **show interface** [*interface-id*] **rep** [**detail**]
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**rep admin vlan** *vlan-id* **Example:** `Router(config)# rep admin vlan 2`	Configures a REP administrative VLAN. • Specify the administrative VLAN. The range is from 2 to 4094. The default is VLAN 1.
Step 4	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 5	**show interface** [*interface-id*] **rep** [**detail**] **Example:** `Router# show interface gigabitethernet0/1 rep detail`	Displays the REP configuration and status for a specified interface. • Enter the physical interface or port channel ID.
Step 6	**copy running-config startup-config** **Example:** `Router# copy running-config startup-config`	(Optional) Save your entries in the router startup configuration file.

Configuring Trunk EFP on an Interface

Before You Begin

For the REP operation, you must configure Trunk EFP on an interface. This task is required and must be done before configuring REP support on a Trunk EFP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service instance trunk** *service-instance-id* **ethernet**
5. **encapsulation dot1q vlan** *range*
6. **rewrite ingress tag pop 1 symmetric**
7. **bridge-domain from-encapsulation**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/1`	Specifies the interface, and enters interface configuration mode. • Enter the interface ID.
Step 4	**service instance trunk** *service-instance-id* **ethernet** **Example:** `Router(config-if)# service instance trunk 1 ethernet`	Configures a service instance on an interface and enters service instance configuration mode.

	Command or Action	Purpose
Step 5	**encapsulation dot1q vlan** *range* **Example:** `Router(config-if-srv)# encapsulation dot1q vlan 10`	Defines the match criteria to be used to map dot1q frames ingress on an interface to the appropriate service instance. • The range of VLAN-IDs is from 1 to 20.
Step 6	**rewrite ingress tag pop 1 symmetric** **Example:** `Router(config-if-srv)# rewrite ingress tag pop 1 symmetric`	Specifies the encapsulation adjustment to be performed on the frames ingress to the service instance.
Step 7	**bridge-domain from-encapsulation** **Example:** `Router(config-if-srv)# bridge-domain from-encapsulation`	Derives bridge domains from encapsulation.
Step 8	**end** **Example:** `Router (config-if-srv)end`	Returns to privileged EXEC mode.

Configuring REP Support on a Trunk EFP

Before You Begin

For the REP operation, you must enable REP on each segment interface and identify the segment ID. This task is required and must be done before other REP configurations. You must also configure a primary and secondary edge port on each segment. All other steps are optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface type number*
4. **rep segment** *segment-id* [**edge** [**primary**]] [**preferred**]
5. **rep stcn** {**interface** *type number* | **segment** *id-list* | **stp**}
6. **rep block port** {**id** *port-id* | *neighbor-offset* | **preferred**} **vlan** {*vlan-list* | **all**}
7. **rep preempt delay** *seconds*
8. **end**
9. **show interface** *type number* **rep** [**detail**]
10. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/1`	Specifies the interface and enters interface configuration mode. • Enter the interface type and number.
Step 4	**rep segment** *segment-id* [**edge** [**primary**]] [**preferred**] **Example:** `Router(config-if)# rep segment 3 edge preferred`	Enables REP on the interface and identifies a segment number. • The segment ID range is from 1 to 1024. **Note** You must configure two edge ports, including one primary edge port for each segment. • (Optional) **edge—**Configures the port as an edge port. Each segment has only two edge ports. Entering the **edge** without the **primary** keyword configures the port as the secondary edge port. • (Optional) **primary—**Configures the port as the primary edge port, the port on which you can configure VLAN load balancing. **Note** Although each segment can have only one primary edge port, if you configure edge ports on two different switches and enter the **primary** keyword on both switches, the configuration is valid. However, REP selects only one of these ports as the segment primary edge port. You can identify the primary edge port for a segment by entering the **show rep topology** privileged EXEC command. • (Optional) **preferred—**Indicates that the port is the preferred alternate port or the preferred port for VLAN load balancing. **Note** Configuring a port as preferred does not guarantee that it becomes the alternate port; it merely gives the port a slight edge over equal contenders. The alternate port is usually a previously failed port.
Step 5	**rep stcn** {**interface** *type number* | **segment** *id-list* | **stp**} **Example:** `Router(config-if)# rep stcn segment 2-5`	(Optional) Configures the edge port to send STCNs. • Use the **interface** *type number* keyword-argument pair to designate a physical interface or port channel to receive STCNs.

	Command or Action	Purpose
		• Use the **segment** *id-list* keyword-argument pair to identify one or more segments to receive STCNs. The range is from 1 to 1024. • Enter the**stp** to send STCNs to STP networks.
Step 6	**rep block port** {**id** *port-id* | *neighbor-offset* | **preferred**} **vlan** {*vlan-list* | **all**} **Example:** `Router(config-if)# rep block port 0009001818D68700 vlan all`	(Optional) Configures VLAN load balancing on the primary edge port, identifies the REP alternate port in one of three ways, and configures the VLANs to be blocked on the alternate port. • Enter the **id** *port-id*keyword-pair to identify the alternate port by port ID. The port ID is automatically generated for each port in the segment. You can view interface port IDs by entering the **show interface** *type number* **rep** [**detail**] command. • Enter a *neighbor-offset* number to identify the alternate port as a downstream neighbor from an edge port. The range is from -256 to 256, with negative numbers indicating the downstream neighbor from the secondary edge port. A value of **0** is invalid. Enter **-1** to identify the secondary edge port as the alternate port. **Note** Because you enter this command at the primary edge port (offset number 1), you cannot enter an offset value of 1 to identify an alternate port. • Enter the**preferred** keyword to select the regular segment port previously identified as the preferred alternate port for VLAN load balancing. • Enter the**vlan** *vlan-list* keyword-argument pair to block one VLAN or a range of VLANs. • Enter the**vlan all** keyword to block all VLANs. **Note** Enter this command only on the REP primary edge port.
Step 7	**rep preempt delay** *seconds* **Example:** `Router(config-if)# rep preempt delay 60`	(Optional) Configures a preempt time delay. • Use this command if you want VLAN load balancing to automatically trigger after a link failure and recovery. • The time delay range is between15 to 300 seconds. The default is manual preemption with no time delay. **Note** Use this command only on the REP primary edge port.
Step 8	**end** **Example:** `Router(config-if-srv)# end`	Returns to privileged EXEC mode.
Step 9	**show interface** *type number* **rep** [**detail**]	(Optional) Verifies the REP interface configuration.

	Command or Action	Purpose
	Example: `Router# show interface Gigabitethernet0/0/1 rep detail`	• Enter the interface type and number and the optional **detail** keyword, if desired.
Step 10	**copy running-config startup-config** **Example:** `Router# copy running-config startup-config`	(Optional) Saves your entries in the router startup configuration file.

Setting the Preemption for VLAN Load Balancing

To set the preemption for VLAN load balancing, complete these steps on the router that has the segment with the primary edge port.

Restrictions

If you do not enter the **rep preempt delay** *seconds* command on the primary edge port to configure a preemption time delay, the default is to manually trigger VLAN load balancing on the segment. Use the **show rep topology** command to see which port in the segment is the primary edge port.

Before You Begin

Be sure that all other segment configurations have been completed before setting the preemption for VLAN load balancing. When you enter the **rep preempt segment** *segment-id* command, a confirmation message appears before the command is executed because preemption for VLAN load balancing can disrupt the network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **rep preempt segment** *segment-id*
4. **end**
5. **show rep topology**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**rep preempt segment** *segment-id* **Example:** `Router(config)# rep preempt segment 1`	Manually triggers VLAN load balancing on the segment. • Enter the segment ID. **Note** You will be asked to confirm the action before the command is executed.
Step 4	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 5	**show rep topology** **Example:** `Router# show rep topology`	Displays the REP topology information.

Configuring SNMP Traps for REP

You can configure the router to send REP-specific traps to notify the Simple Network Management Protocol (SNMP) server of link operational status changes and any port role changes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp mib rep trap-rate** *value*
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp mib rep trap-rate** *value* **Example:** `Router(config)# snmp mib rep trap-rate 500`	Enables the router to send REP traps, and sets the number of traps sent per second. • Enter the number of traps sent per second. The range is from 0 to 1000. The default is 0 (no limit imposed; a trap is sent at every occurrence). **Note** To remove the traps, enter the **no snmp mib rep trap-rate** command.
Step 4	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 5	**show running-config** **Example:** `Router# show running-config`	(Optional) Displays the running configuration, which van be used to verify the REP trap configuration.
Step 6	**copy running-config startup-config** **Example:** `Router# copy running-config startup-config`	(Optional) Saves your entries in the router startup configuration file.

Monitoring the REP Configuration

SUMMARY STEPS

1. **enable**
2. **show interface** [*interface-id*] **rep** [**detail**]
3. **show rep topology** [**segment** *segment-id*] [**archive**] [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show interface** [*interface-id*] **rep** [**detail**] **Example:** `Router# show interface gigabitethernet0/1 rep detail`	(Optional) Displays the REP configuration and status for a specified interface. • Enter the physical interface or port channel ID, and the optional **detail** keyword, if desired.
Step 3	**show rep topology** [**segment** *segment-id*] [**archive**] [**detail**] **Example:** `Router# show rep topology`	(Optional) Displays REP topology information for a segment or for all segments, including the primary and secondary edge ports in the segment. • Enter the optional keywords and arguments, as desired.

Configuring REP Configurable Timers

Before You Begin

For the REP operation, you must enable REP on each segment interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **rep segment** *segment-id* [**edge** [**no-neighbor**] [**primary**]] [**preferred**]
5. **rep stcn** {**interface** *type number* | **segment** *id-list* | **stp**}
6. **rep block port** {**id** *port-id* | *neighbor-offset* | **preferred**} **vlan** {*vlan-list* | **all**}
7. **rep lsl-retries** *number-of-tries*
8. **rep lsl-age-timer** *timer-value*
9. **rep preempt delay** *seconds*
10. **end**
11. **show interface** *type number* **rep** [**detail**]
12. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface Gigabitethernet 0/0/1`	Specifies the interface and enters interface configuration mode. • Enter the interface type and number.
Step 4	**rep segment** *segment-id* [**edge** [**no-neighbor**] [**primary**]] [**preferred**] **Example:** `Router(config-if)# rep segment 1 edge preferred`	Enables REP on the interface and identifies a segment number. • The segment ID range is from 1 to 1024. **Note** You must configure two edge ports, including one primary edge port for each segment. • (Optional) **edge—**Configures the port as an edge port. Each segment has only two edge ports. Entering the **edge** keyword without the **primary** keyword configures the port as the secondary edge port. • (Optional)**no-neighbor—**Configures the segment edge as one with no external REP neighbor on a port.

	Command or Action	Purpose
		• (Optional) **primary—**Configures the port as the primary edge port, the port on which you can configure VLAN load balancing. **Note** Although each segment can have only one primary edge port, if you configure edge ports on two different switches and enter the **primary** keyword on both switches, the configuration is valid. However, REP selects only one of these ports as the segment primary edge port. You can identify the primary edge port for a segment by entering the **show rep topology** privileged EXEC command. • (Optional) **preferred—**Indicates that the port is the preferred alternate port or the preferred port for VLAN load balancing. **Note** Configuring a port as preferred does not guarantee that it becomes the alternate port; it merely gives the port a slight edge over equal contenders. The alternate port is usually a previously failed port.
Step 5	**rep stcn** {**interface** *type number* | **segment** *id-list* | **stp**} **Example:** `Router(config-if)# rep stcn segment 2-5`	(Optional) Configures the edge port to send STCNs. • Use the **interface** *type number* keyword and arguments pair to designate a physical interface or port channel to receive STCNs. • Use the **segment** *id-list* keyword and arguments pair to identify one or more segments to receive STCNs. The range is from 1 to 1024. • Enter the **stp** keyword to send STCNs to STP networks.
Step 6	**rep block port** {**id** *port-id* | *neighbor-offset* | **preferred**} **vlan** {*vlan-list* | **all**} **Example:** `Router(config-if)# rep block port 0009001818D68700 vlan all`	(Optional) Configures VLAN load balancing on the primary edge port, identifies the REP alternate port in one of three ways, and configures VLANs to be blocked on the alternate port. • Enter the **id** *port-id* keyword and arguments pair to identify the alternate port by port ID. The port ID is automatically generated for each port in the segment. You can view interface port IDs by entering the **show interface** *type number* **rep** [**detail**] command. • Enter a *neighbor-offset* number to identify the alternate port as a downstream neighbor from an edge port. The range is from -256 to 256, with negative numbers indicating the downstream neighbor from the secondary edge port. A value of **0** is invalid. Enter **-1** to identify the secondary edge port as the alternate port. **Note** Because you enter this command at the primary edge port (offset number 1), you cannot enter an offset value of 1 to identify an alternate port. • Enter the **preferred** keyword to select the regular segment port previously identified as the preferred alternate port for VLAN load balancing. • Enter the **vlan** *vlan-list* keyword and arguments pair to block one VLAN or a range of VLANs. • Enter the **vlan all** keyword to block all VLANs.

	Command or Action	Purpose
		Note Enter this command only on the REP primary edge port.
Step 7	**rep lsl-retries** *number-of-tries* **Example:** `Router(config-if)# rep lsl-retries 3`	Configures the number of retries permitted by LSL.
Step 8	**rep lsl-age-timer** *timer-value* **Example:** `Router(config-if)# rep lsl-age-timer 200`	Configures the failure detection time. • The valid range is from 120 to 10000. We recommend that you configure the minimum range as 200 for better performance.
Step 9	**rep preempt delay** *seconds* **Example:** `Router(config-if)# rep preempt delay 60`	• (Optional) Configures a preempt time delay. • Use this command if you want VLAN load balancing to automatically trigger after a link failure and recovery. • The time delay range is from 15 to 300 seconds. The default is manual preemption with no time delay. **Note** Use this command only on the REP primary edge port.
Step 10	**end** **Example:** `Router(config-if-srv)# end`	Returns to privileged EXEC mode.
Step 11	**show interface** *type number* **rep** [**detail**] **Example:** `Router# show interface Gigabitethernet0/0/1 rep detail`	(Optional) Displays the REP interface configuration. • Enter the interface type and number and the optional **detail** keyword, if desired.
Step 12	**copy running-config startup-config** **Example:** `Router# copy running-config startup-config`	(Optional) Saves your entries in the router startup configuration file.

Configuring REP as an Edge No-Neighbor Port

Before You Begin

For the REP operation, you must enable REP on each segment interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **rep segment** *segment-id* [**edge** [**no-neighbor**] [**primary**]] [**preferred**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface GigabitEthernet 0/0/1`	Specifies the interface and enters interface configuration mode. • Enter the interface type and number.
Step 4	**rep segment** *segment-id* [**edge** [**no-neighbor**] [**primary**]] [**preferred**] **Example:** `Router(config-if)# rep segment 1 edge no-neighbor preferred`	Enables REP on the interface and identifies a segment number. • The segment ID range is from 1 to 1024. **Note** You must configure two edge ports, including one primary edge port for each segment. • (Optional) **edge**-Configures the port as an edge port. Each segment has only two edge ports. Entering **edge** without the **primary** keyword configures the port as the secondary edge port. • (Optional)**no-neighbor**-Indicates the segment edge as one with no external REP neighbor on a port. • (Optional) **primary**-Configures the port as the primary edge port, the port on which you can configure VLAN load balancing.

	Command or Action	Purpose
		Note Although each segment can have only one primary edge port, if you configure edge ports on two different switches and enter the **primary** keyword on both switches, the configuration is valid. However, REP selects only one of these ports as the segment primary edge port. You can identify the primary edge port for a segment by entering the **show rep topology** privileged EXEC command. • (Optional) **preferred**-Indicates that the port is the preferred alternate port or the preferred port for VLAN load balancing. **Note** Configuring a port as preferred does not guarantee that it becomes the alternate port; it merely gives the port a slight edge over equal contenders. The alternate port is usually a previously failed port.

Configuration Examples for REP

Configuring the REP Administrative VLAN

This example shows how to configure the administrative VLAN as VLAN 100.

```
Router# configure terminal
Router(config)# rep admin vlan 100
Router(config-if)# end
```

Configuring REP Support on a Trunk EFP

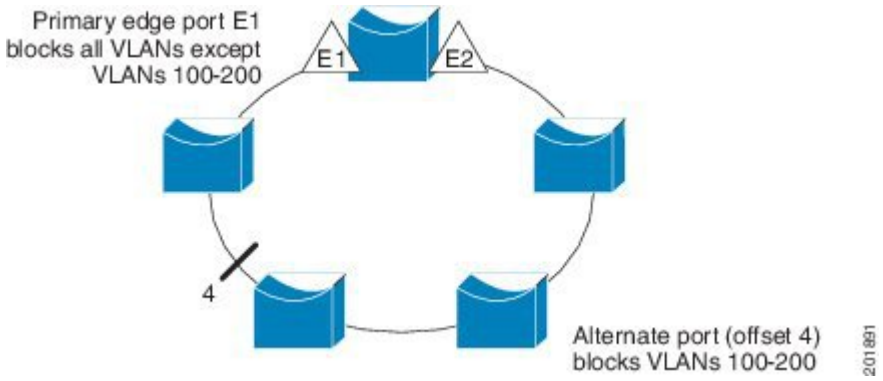
This example shows how to configure REP support on a Trunk EFP. An interface is configured as the primary edge port for segment 1 to send STCNs to segments 2 through 5; the alternate port is configured as the port with port ID 0009001818D68700 to block all VLANs after a preemption delay of 60 seconds after a segment port failure and recovery.

```
Router# configure terminal
Router(config)# interface gigabitethernet0/0/1
Router(config-if)# rep segment 1 edge primary
Router(config-if)# rep stcn segment 2-5
Router(config-if)# rep block port id 0009001818D68700 vlan all
Router(config-if)# rep preempt delay 60
Router(config-if)# service instance trunk 1 ethernet
Router(config-if-srv)# encapsulation dot1q
Router(config-if-srv)# rewrite ingress tag pop 1 symmetric
Router(config-if-srv)# bridge-domain from-encapsulation
Router(config-if-srv)# end
```

This example shows how to configure the VLAN blocking configuration as shown in the figure below. The alternate port is the neighbor with neighbor offset number 4. After manual preemption, VLANs 100 to 200

are blocked at this port and all other VLANs are blocked at the primary edge port E1 (Gigabit Ethernet port 0/0/1).

Figure 22: Example of VLAN Blocking




```
Router# configure terminal
Router(config)# interface gigabitethernet0/0/1
Router(config-if)# rep segment 1 edge primary
Router(config-if)# rep block port 4 vlan 100-200
Router(config-if)# end
```

Setting the Preemption for VLAN Load Balancing

```
Router>end
Router# configure terminal
Router(config)rep preempt segment 1
Router(config)# end
```

Configuring SNMP Traps for REP

This example shows how to configure the router to send REP traps at a rate of 10 traps per second:

```
Router> enable
Router# configure terminal
Router(config)# snmp mib rep trap-rate 10
Router(config)# end
```

Monitoring the REP Configuration

The following is sample output of the **show interface rep detail** command. Use the **show interface rep detail** command on one of the REP interfaces to monitor and verify the REP configuration.

```
Router# show interface GigabitEthernet 0/0/1 rep detail

GigabitEthernet0/1 REP enabled
Segment-id: 2 (Edge)
PortID: 00010019E7144680
Preferred flag: No
Operational Link Status: TWO_WAY
Current Key: 0002001121A2D5800E4D
Port Role: Open
Blocked Vlan: <empty>
```

```
Admin-vlan: 100
Preempt Delay Timer: disabled
Load-balancing block port: none
Load-balancing block vlan: none
STCN Propagate to: none
LSL PDU rx: 3322, tx: 1722
HFL PDU rx: 32, tx: 5
BPA TLV rx: 16849, tx: 508
BPA (STCN, LSL) TLV rx: 0, tx: 0
BPA (STCN, HFL) TLV rx: 0, tx: 0
EPA-ELECTION TLV rx: 118, tx: 118
EPA-COMMAND TLV rx: 0, tx: 0
EPA-INFO TLV rx: 4214, tx: 4190
```

Configuring REP Configurable Timers

```
Router# configure terminal
Router(config)# interface GigabitEthernet 0/0/4
Router(config-if)# rep segment 4 edge preferred
Router(config-if)# rep stcn segment 2-5
Router(config-if)# rep block port 0009001818D68700 vlan all
Router(config-if)# rep lsl-retries 3
Router(config-if)# rep lsl-age-timer 200
Router(config-if)# rep preempt delay 300
Router(config-if)# exit
Router# show interface GigabitEthernet 0/0/1 rep detail
Router# copy running-config startup-config
```

Configuring REP Edge No-Neighbor Support

```
Router> enable
Router# configure terminal
Router(config)# interface GigabitEthernet0/2
Router(config-if)# rep segment t1 edge no-neighbor primary
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
LAN Switching commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples.	Cisco IOS LAN Switching Command Reference
Introduction to spanning tree protocols	Spanning Tree Protocol (STP)/802.1D
Spanning Tree PortFast BPDU Guard Enhancement feature	Spanning Tree PortFast BPDU Guard Enhancement

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Resilient Ethernet Protocol

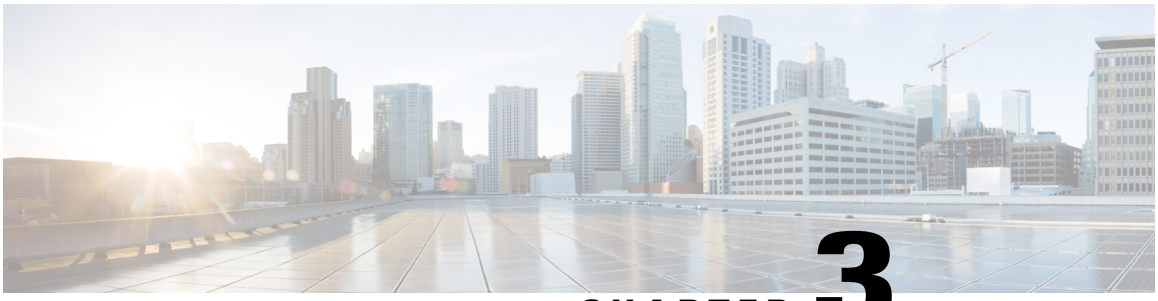
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5: Feature Information for Resilient Ethernet Protocol

Feature Name	Releases	Feature Information
REP Configurable Timers	Cisco IOS XE Release 3.5.1S	REP Configurable Timers on REP to detect link failures in a link between routers in a ring topology. In Cisco IOS XE Release 3.5.1S, support was added for the Cisco ASR 903 Router. In Cisco IOS XE Release 3.11S, support was added for the Cisco ASR 901 Routers. The following sections provide information about this feature: REP Configurable Timers Configuring REP Configurable Timers Configuring REP Configurable Timers

Feature Name	Releases	Feature Information
REP Edge No-Neighbor Support	Cisco IOS XE Release 3.5.1S	The Edge No-Neighbor Support on REP enables defining a new type of edge that has an internal neighbor. In Cisco IOS XE Release 3.5.1S, support was added for the Cisco ASR 903 Router. REP Edge No-Neighbor Support Configuring REP Edge No-Neighbor Support
REP Support on Trunk EVC	Cisco IOS XE Release 3.5S	REP can be configured on Trunk Ethernet Flow Point (EFP) ports at an interface level on ASR 903 Series Routers. The following command was introduced by this feature: **service instance trunk**.
SSO Support for REP Fast Hello	Cisco IOS XE Release 3.5.1S	SSO Support for REP Fast Hello is provided to ensure that a Fast Hello packet is sent from an active router before the LSL timeout interval expires. In Cisco IOS XE Release 3.5.1S, support was added for the Cisco ASR 903 Router. In Cisco IOS XE Release 3.11S, support was added for the Cisco ASR 901 Router. The following sections provide information about this feature: SSO Support for REP Fast Hello



CHAPTER 3

cGVRP

The Compact Generic Attribute Registration Protocol (GARP) VLAN Registration Protocol (GVRP) (cGVRP) feature reduces CPU time for the transmission of 4094 VLAN states on a port.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for cGVRP

- A non-Cisco device can only interoperate with a Cisco device through .1Q trunks.
- VLAN Mapping is not supported with GVRP.
- cGVRP and Connectivity Fault Management (CFM) can coexist but if the line card (LC) or supervisor does not have enough mac-match registers to support both protocols, the cGVRP ports on those LCs

are put in error disabled state. To use Layer 2 functionality, disable cGVRP on those ports and configure shut/no shut.

- cGVRP functionality applies only to interfaces configured for Layer 2 (switchport) functionality.
- Native VLAN Tagging causes frames sent to the native VLAN of the .1Q trunk ports to be encapsulated with .1Q tags. Problems may arise with other GVRP participants on the LAN because they may not be able to admit tagged GVRP PDUs. Caution must be exercised if both features are enabled at the same time.
- 802.1X authentication and authorization takes place after the port becomes link-up and before the Dynamic Trunking Protocol (DTP) negotiations start prior to GVRP running on the port.
- Port Security works independently from GVRP and it may be limited to the number of other GVRP participants on a LAN that a GVRP enabled port on a device can communicate with.
- GVRPs cannot be configured and used on a sub-interface.
- GVRP and UniDirectional Link Routing (UDLR) should not be enabled on the same interface because UDLR limits frames in one direction on the port and GVRP is a two way communication protocol.
- Additional memory is required to store GARP/GVRP configurations and states per GVRP enabled port, but it can be dynamically allocated on demand.
- GARP Multicast Registration Protocol (GMRP) is not supported.

Information About cGVRP

GARP GVRP Definition

GVRP enables automatic configuration of switches in a VLAN network allowing network devices to dynamically exchange VLAN configuration information with other devices. GVRP is based on GARP which defines procedures for registering and deregistering attributes with each other. It eliminates unnecessary network traffic by preventing attempts to transmit information to unregistered users.

GVRP is defined in IEEE 802.1Q.

cGVRP Overview

GVRP is a protocol that requires extensive CPU time in order to transmit all 4094 VLAN states on a port. In Compact mode only one PDU is sent and it includes the states of all the 4094 VLANs on a port.

VLAN pruning can be accomplished faster by running in a special mode, Fast Compact Mode, and on point-to-point links.

In Compact GVRP a GVRP PDU may be sent out the port if the port is in forwarding state in a spanning tree instance. GVRP PDUs must be transmitted in the native VLAN of .1Q trunks.

GVRP Interoperability with VTP and VTP Pruning

VTP Pruning is an extension of VTP. It has its own Join message that can be exchanged with VTP PDUs. VTP PDUs can be transmitted on both .1Q trunks and ISL trunks. A VTP capable device is in either one of the three VTP modes: Server, Client, or Transparent.

When VTP Pruning and GVRP are both enabled globally, VTP Pruning is run on ISL trunks, and GVRP is run on .1Q trunks.

Compact GVRP has two modes: Slow Compact Mode, and Fast Compact Mode. A port can be in Fast Compact Mode if it has one GVRP enabled peer on the same LAN segment, and the peer is capable of operating in Compact Mode. A port is in Slow Compact Mode if there are multiple GVRP participants on the same LAN segment operating in Compact Mode.

GVRP Interoperability with Other Software Features and Protocols

This section briefly describes GVRP interoperability with the following software features and protocols.

STP

Spanning Tree Protocol (STP) may run in one of the three STP modes: Multiple Spanning Tree(MST), Per VLAN Spanning Tree (PVST), or Rapid PVST. An STP mode range causes the forwarding ports to leave the forwarding state as STP has to reconverge. This may cause GVRP to have its own topology change as Join messages my be received on some new ports and Leave timers may expire on some others.

DTP

DTP (DDSN Transfer Protocol) negotiates the port mode (trunk versus non-trunk) and the trunk encapsulation type between two DTP enabled ports. After negotiation DTP may set the port to either ISL trunk, or .1Q trunk, or non-trunk. DTP negotiation occurs after ports become link-up and before they become forwarding in spanning trees. If GVRP is administratively enabled on a port and the device, it should be initialized after the port is negotiated to be a .1Q trunk.

VTP

VTP (Virtual Terminal Protocol) version 3 expands the range of VLANs that can be created and removed via VTP. VTP Pruning is available for VLAN 1 through 1005 only.

EtherChannel

When multiple .1Q trunk ports are grouped by either Port Aggregation Protocol (PAgP) or Link Aggregation Control Protocol (LACP) to become an EtherChannel, the EtherChannel can be configured as a GVRP participant. The physical ports in the EtherChannel cannot be GVRP participants by themselves. Since an EtherChannel is treated like one virtual port by STP, the GVRP application can learn the STP state change of the EtherChannel just like any physical port. The EtherChannel, not the physical ports in the channel, constitutes the GARP Information Propagation (GIP) context.

High Availability

High Availability (HA) is a redundancy feature in IOS. On platforms that support HA and State SwitchOver (SSO), many features and protocols my resume working in a couple of seconds after the system encounters a failure such as a crash of the active supervisor in a Catalyst 7600 switch. GVRP needs to be configured to enable user configurations, and protocol states should be synched to a standby system. If there is a failure of the active system, the GVRP in the standby system which now becomes active, has all the up-to-date VLAN registration information.

How to Configure cGVRP

Configuring Compact GVRP

To configure compact GVRP, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **grvp global**
4. **gvrp timer join** *timer* **-** *value*
5. **gvrp registration normal**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**grvp global** **Example:** `Router(config)# gvrp global`	Configures global GVRP and enables GVRP on all .1Q trunks.
Step 4	**gvrp timer join** *timer* **-** *value*	Sets the period timers that are used in GARP on an interface,

	Command or Action	Purpose
	Example: `Router(config)# gvrp timer join 1000`	• Enter the timer-value. The timer-value range is between 200 and 2147483647.
Step 5	**gvrp registration normal** **Example:** `Router(config)# gvrp registration normal`	Sets the registrar for normal response to incoming GVRP messages.
Step 6	**end** **Example:** `Router(config)# end`	Exits interface configuration mode.

Disabling mac-learning on VLANs

To disable mac-learning on VLANs, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **gvrp mac-learning auto**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**gvrp mac-learning auto** **Example:** `Router(config)# gvrp mac-learning auto`	Disables learning of mac-entries.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode.

Enabling a Dynamic VLAN

To enable a dynamic VLAN, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **gvrp vlan create**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**gvrp vlan create** **Example:** `Router(config)# gvrp vlan create`	Enables a dynamic VLAN when cGRVP is configured.

	Command or Action	Purpose
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode.

Troubleshooting the cGVRP Configuration

To troubleshoot the cGVRP configuration, use one or more of the commands listed below.

Use the **show gvrp summary**command and the **show gvrp interface**command to display configuration information and interface state information. Use the **debug gvrp** command to enable all or a limited set of output messages related to an interface.

SUMMARY STEPS

1. **enable**
2. **show gvrp summary**
3. **show gvrp interface**
4. **debug gvrp**
5. **clear gvrp statistics**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show gvrp summary** **Example:** `Router# show gvrp summary`	Displays the GVRP configuration.
Step 3	**show gvrp interface** **Example:** `Router# show gvrp interface`	Displays the GVRP interface states.

	Command or Action	Purpose
Step 4	**debug gvrp** **Example:** `Router# debug gvrp`	Displays GVRP debugging information.
Step 5	**clear gvrp statistics** **Example:** `Router# clear gvrp statistics`	Clears GVRP statistics on all interfaces.
Step 6	**end** **Example:** `Router# end`	Exits privileged EXEC mode.

Configuration Examples for cGVRP

Configuring cGVRP Example

The following example shows how to configure compact GVRP.

```
Router> enable
Router# configure terminal
Router(config)# gvrp global
Router(config)# gvrp timer join 1000
Router(config)# gvrp registration normal
Router(config)# end
```

Disabling mac-learning on VLANs Example

The following example shows how to disable mac-learning on VLANs configured with cGVRP.

```
Router> enable
Router# configure terminal
Router(config)# gvrp mac-learning auto
Router(config)# end
```

Enabling a Dynamic VLAN Example

The following example shows how to configure a dynamic VLAN.

```
Router> enable
Router# configure terminal
Router(config)# gvrp vlan create
Router(config)# end
```

Verifying CE Port Configurations Examples

This section contains examples that can be used to verify the CE port configurations. It contains the following examples:

The examples provide sample output of the **show running-config** command, the **show grvp summary** command, and the **show grvp interface** command. The output of these commands is based on the following topology:

- CE (customer edge) 1 port on a gigabitethernet 3/15 interface
- Router 1 with a gigabitethernet 3/1 interface
- A .1Q trunk across a gigabitethernet 3/1 interface
- Router 2 with a gigabitethernet 2/15 interface
- CE 2 port

Verifying CE Ports Configured as Access Ports Example

The following is sample output of the **show running-config interface** command, the **show grvp summary**,and the **show grvp interface**command. In this configuration the CE ports are configured as access ports.

```
Router1# show running-config interface gigabitethernet 3/15
Building configuration...
Current configuration : 129 bytes
```

```
!
interface GigabitEthernet3/15
 switchport
 switchport access vlan 2
 switchport mode access
 spanning-tree portfast trunk
end
Router1# show running-config interface gigabitethernet 3/1
Building configuration...
Current configuration : 109 bytes
!
interface GigabitEthernet3/1
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
end
Router2# show running-config interface gigabitethernet 12/15
Building configuration...
Current configuration : 168 bytes
!
interface GigabitEthernet12/15
 switchport
 switchport access vlan 2
 switchport trunk encapsulation dot1q
 switchport mode access
 spanning-tree portfast trunk
end
Router2# show running-config interface gigabitethernet 3/1
Building configuration...
Current configuration : 144 bytes
!
interface GigabitEthernet3/1
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport backup interface Gi4/1
end
Router1# show gvrp summary
GVRP global state           : enabled
GVRP VLAN creation          : disabled
VLANs created via GVRP      : none
MAC learning auto provision : disabled
Learning disabled on VLANs  : none
Router1# show gvrp interface
Port      Status    Mode            Registrar State
Gi3/1     on        fastcompact        normal
Port      Transmit Timeout   Leave Timeout   Leaveall Timeout
Gi3/1     200                600             10000
Port      Vlans Declared
Gi3/1     2
Port      Vlans Registered
Gi3/1     2
Port      Vlans Registered and in Spanning Tree Forwarding State
Gi3/1     2
Router2# show gvrp summary
GVRP global state           : enabled
GVRP VLAN creation          : disabled
VLANs created via GVRP      : none
MAC learning auto provision : disabled
Learning disabled on VLANs  : none
Router2# show gvrp interface
Port      Status    Mode            Registrar State
Gi3/1     on        fastcompact        normal
Port      Transmit Timeout   Leave Timeout   Leaveall Timeout
Gi3/1     200                600             10000
Port      Vlans Declared
Gi3/1     2
Port      Vlans Registered
Gi3/1     2
Port      Vlans Registered and in Spanning Tree Forwarding State
Gi3/1     2
```

Verifying CE Ports Configured as ISL Ports Example

The following is sample output of the **show running-config interface** command, the **show grvp summary**,the **show grvp interface**command, and the **show vlan summary** command. In this configuration the CE ports are configured as ISL ports.

```
Router1# show running-config interface gigabitethernet 3/15
Building configuration...
Current configuration : 138 bytes
!
interface GigabitEthernet3/15
 switchport
 switchport trunk encapsulation isl
 switchport mode trunk
 spanning-tree portfast trunk
end
Router1# show running-config interface gigabitethernet 3/1
Building configuration...
Current configuration : 109 bytes
!
interface GigabitEthernet3/1
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
end
Router2# show running-config interface gigabitethernet 12/15
Building configuration...
Current configuration : 139 bytes
!
interface GigabitEthernet12/15
 switchport
 switchport trunk encapsulation isl
 switchport mode trunk
 spanning-tree portfast trunk
end
Router2# show running-config interface gigabitethernet 3/1
Building configuration...
Current configuration : 144 bytes
!
interface GigabitEthernet3/1
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport backup interface Gi4/1
end
Router1# show gvrp summary

GVRP global state           : enabled
GVRP VLAN creation          : disabled
VLANs created via GVRP      : none
MAC learning auto provision : disabled
Learning disabled on VLANs  : none
Router1# show gvrp interface
Port      Status    Mode            Registrar State
Gi3/1     on        fastcompact        normal
Port      Transmit Timeout   Leave Timeout   Leaveall Timeout
Gi3/1     200                600             10000
Port      Vlans Declared
Gi3/1     1-10
Port      Vlans Registered
Gi3/1     1-2
Port      Vlans Registered and in Spanning Tree Forwarding State
Gi3/1     1-2
Router1# show vlan summary
Number of existing VLANs           : 14
 Number of existing VTP VLANs      : 14
 Number of existing extended VLANs : 0
Router2# show gvrp summary
GVRP global state           : enabled
```

```
GVRP VLAN creation          : disabled
VLANs created via GVRP      : none
MAC learning auto provision : disabled
Learning disabled on VLANs  : none
Router2# show gvrp interface
Port      Status    Mode            Registrar State
Gi3/1     on        fastcompact        normal
Port      Transmit Timeout   Leave Timeout   Leaveall Timeout
Gi3/1     200                600             10000
Port      Vlans Declared
Gi3/1     1-2
Port      Vlans Registered
Gi3/1     1-10
Port      Vlans Registered and in Spanning Tree Forwarding State
Gi3/1     1-2
Router2# show vlan summary
Number of existing VLANs           : 6
 Number of existing VTP VLANs      : 6
 Number of existing extended VLANs : 0
```

Verifying CE Ports Configured in Fixed Registration Mode Example

The following is sample output of the **show running-config interface** command and the **show grvp interface**command. In this configuration the CE ports are configured in fixed registration mode.

```
Router1# show running-config interface gigabitethernet 3/15
Building configuration...
Current configuration : 165 bytes
!
interface GigabitEthernet3/15
 gvrp registration fixed
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 spanning-tree portfast trunk
end
Router1# show gvrp interface gigabitethernet 3/15
Port      Status    Mode            Registrar State
Gi3/15    on        fastcompact        fixed
Port      Transmit Timeout   Leave Timeout   Leaveall Timeout
Gi3/15    200                600             10000
Port      Vlans Declared
Gi3/15    1-2
Port      Vlans Registered
Gi3/15    1-4094
Port      Vlans Registered and in Spanning Tree Forwarding State
Gi3/15    1-10
```

Verifying CE Ports Configured in Forbidden Registration Mode Example

The following is sample output of the **show running-config interface** command and the **show grvp interface**command. In this configuration the CE ports are configured in forbidden registration mode.

```
Router1# show running-config interface gigabitethernet 3/15
Building configuration...
Current configuration : 169 bytes
!
interface GigabitEthernet3/15
 gvrp registration forbidden
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 spanning-tree portfast trunk
end
Router1# show
 gvrp
```

```
 interface gigabitethernet 3/15
Port      Status    Mode            Registrar State
Gi3/15    on        fastcompact     forbidden
Port      Transmit Timeout   Leave Timeout   Leaveall Timeout
Gi3/15    200                600             10000
Port      Vlans Declared
Gi3/15    1-2
Port      Vlans Registered
Gi3/15    none
Port      Vlans Registered and in Spanning Tree Forwarding State
Gi3/15    none
```

Verifying CE Ports Configured with a .1Q Trunk Example

The following is sample output of the **show running-config interface** command, the **show grvp summary**,andthe **show grvp interface**command. In this configuration the CE ports are configured with a .1Q trunk.

```
Router1# show running-config interface gigabitethernet 3/15
Building configuration...
Current configuration : 165 bytes
!
interface GigabitEthernet3/15
 gvrp registration fixed
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 spanning-tree portfast trunk
end
Router2# show running-config interface gigabitethernet 12/15
Building configuration...
Current configuration : 166 bytes
!
interface GigabitEthernet12/15
 gvrp registration fixed
 switchport
 switchport trunk encapsulation dot1q
 switchport mode trunk
 spanning-tree portfast trunk
end
Router1# show gvrp summary

GVRP global state           : enabled
GVRP VLAN creation          : disabled
VLANs created via GVRP      : none
MAC learning auto provision : disabled
Learning disabled on VLANs  : none
Router1# show gvrp interface
Port      Status    Mode            Registrar State
Gi3/1     on        fastcompact        normal
Gi3/15    on        fastcompact        fixed
Port      Transmit Timeout   Leave Timeout   Leaveall Timeout
Gi3/1     200                600             10000
Gi3/15    200                600             10000
Port      Vlans Declared
Gi3/1     1-10
Gi3/15    1-2
Port      Vlans Registered
Gi3/1     1-2
Gi3/15    1-4094
Port      Vlans Registered and in Spanning Tree Forwarding State
Gi3/1     1-2
Gi12/15   1-10
Router2# show gvrp summary
GVRP global state           : enabled
GVRP VLAN creation          : disabled
VLANs created via GVRP      : none
MAC learning auto provision : disabled
Learning disabled on VLANs  : none
```

```
Router2# show gvrp interface
Port      Status    Mode            Registrar State
Gi3/1     on        fastcompact         normal
Gi12/15   on        fastcompact         fixed
Port      Transmit Timeout   Leave Timeout   Leaveall Timeout
Gi3/1     200                600             10000
Gi12/15   200                600             10000
Port      Vlans Declared
Gi3/1     1-2
Gi12/15   1-2
Port      Vlans Registered
Gi3/1     1-10
Gi12/15   1-4094
Port      Vlans Registered and in Spanning Tree Forwarding State
Gi3/1     1-2
Gi12/15   1-2
```

Verifying cGVRP Example

The following is sample output from the **show grvp summary** command. Use the **show grvp summary**command to verify the compact GVRP configuration.

```
Router# show
 gvrp
 summary
GVRP global state           : enabled
GVRP VLAN creation          : disabled
VLANs created via GVRP      : none
MAC learning auto provision : disabled
Learning disabled on VLANS  : none
```

Verifying Disabled mac-learning on VLANs Example

The following is sample output from the **show gvrp summary**command and the **show gvrp interface**command. Use these two commands to verify that mac-learning has been disabled.

```
Router# show
 gvrp
 summary
GVRP global state           : enabled
GVRP VLAN creation          : enabled
VLANs created via GVRP      : 2-200
MAC learning auto provision : enabled
Learning disabled on VLANs  : 1-200
Router# show gvrp interface
Port      Status    Mode            Registrar State
Gi3/15    on        fastcompact        normal
Gi4/1     on        fastcompact        normal
Port      Transmit Timeout   Leave Timeout   Leaveall Timeout
Gi3/15    200                600             10000
Gi4/1     200                600             10000
Port      Vlans Declared
Gi3/15    1-200
Gi4/1     none
Port      Vlans Registered
Gi3/15    none
Gi4/1     1-200
Port      Vlans Registered and in Spanning Tree Forwarding State
Gi3/15    none
Gi4/1     1-200
Router# show mac- dy
Legend: * - primary entry
        age - seconds since last seen
        n/a - not available
```

```
  vlan   mac address     type    learn     age              ports
------+---------------+--------+-----+----------+--------------------------
No entries present.
```

Verifying Dynamic VLAN Example

The following is sample output from the **show gvrp summary**command and the **show gvrp interface**command. Use these two commands to verify the dynamic VLAN configuration.

```
Router# show
 gvrp
 summary
GVRP global state           : enabled
GVRP VLAN creation          : enabled
VLANs created via GVRP      : 2-200
MAC learning auto provision : disabled
Learning disabled on VLANs  : none
Router# show gvrp interface
Port      Status    Mode            Registrar State
Gi3/15    on        fastcompact         normal
Gi4/1     on        fastcompact         normal
Port      Transmit Timeout   Leave Timeout   Leaveall Timeout
Gi3/15    200                600             10000
Gi4/1     200                600             10000
Port      Vlans Declared
Gi3/15    1-200
Gi4/1     none
Port      Vlans Registered
Gi3/15    none
Gi4/1     1-200
Port      Vlans Registered and in Spanning Tree Forwarding State
Gi3/15    none
Gi4/1     1-200
```

Additional References

Related Documents

Related Topic	Document Title
IP LAN switching commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS LAN Switching Services Command Reference

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing standards has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for cGVRP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 6: Feature Information for cGVRP

Feature Name	Releases	Feature Information
cGVRP	12.2(33)SRB	The Compact (c) Generic Attribute Registration Protocol (GARP) VLAN Registration Protocol (GVRP) feature reduces CPU time for transmittal of 4094 VLAN states on a port. GVRP enables automatic configuration of switches in a VLAN network allowing network devices to dynamically exchange VLAN configuration information with other devices. GVRP is based on GARP which defines procedures for registering and deregistering attributes with each other. It eliminates unnecessary network traffic by preventing attempts to transmit information to unregistered users. GVRP is defined in IEEE 802.1Q. The following commands were introduced or modified: **clear gvrp statistics, debug gvrp, gvrp global, gvrp mac-learning, gvrp registration, gvrp timer, gvrp vlan create, show gvrp interface, show gvrp summary.**