Configuring ISO CLNS

First Published: March 01, 2004
Last Updated: February 26, 2010

The International Organization for Standardization (ISO) Connectionless Network Service (CLNS) protocol is a standard for the network layer of the Open System Interconnection (OSI) model. Before you can configure this protocol, you must understand addresses and routing processes. This module describes addresses, routing processes, and the steps you follow to configure ISO CLNS.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 for Configuring ISO CLNS" section on page 69.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp. An account on Cisco.com is not required.

Contents

- "Prerequisites for Configuring ISO CLNS" section on page 2
- "Restrictions for Configuring ISO CLNS" section on page 2
- "Information About Configuring ISO CLNS" section on page 2
- "How to Configure ISO CLNS" section on page 9
- "ISO CLNS Configuration Examples" section on page 50
- "Additional References" section on page 67
- "Feature Information for Configuring ISO CLNS" section on page 69

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Configuring ISO CLNS

To configure ISO CLNS, network service access points (NSAPs) to address the end systems and titles to identify network devices are required.

Restrictions for Configuring ISO CLNS

ISO CLNS does not perform connection setup or termination because paths are determined independently for each packet that is transmitted through a network.

Information About Configuring ISO CLNS

To configure ISO CLNS you should understand the following concepts:

- "Overview" section on page 2
- "Understanding Addresses" section on page 3

Overview

Cisco IOS software supports packet forwarding and routing for ISO CLNS on networks using a variety of data link layers: Ethernet, Token Ring, FDDI, and serial.

You can use CLNS routing on serial interfaces with HDLC, PPP, Link Access Procedure, Balanced (LAPB), X.25, SMDS, or Frame Relay encapsulation. To use HDLC encapsulation, you must have a router at both ends of the link. If you use X.25 encapsulation, you must manually enter the network service access point (NSAP)-to-X.121 mapping. The LAPB, X.25, Frame Relay, and SMDS encapsulations interoperate with other vendors.

The Cisco CLNS implementation also is compliant with the Government OSI Profile (GOSIP) Version 2.

As part of its CLNS support, Cisco routers fully support the following ISO and American National Standards Institute (ANSI) standard:

- ISO 9542—Documents the ES-IS routing exchange protocol.
- ISO 8473—Documents the ISO Connectionless Network Protocol (CLNP).
- ISO 8348/Ad2—Documents NSAP addresses.
- ISO 10589—Documents IS-IS Intradomain Routing Exchange Protocol.

Both the ISO-developed IS-IS routing protocol and the Cisco ISO Interior Gateway Routing Protocol (IGRP) are supported for dynamic routing of ISO CLNS. In addition, static routing for ISO CLNS is supported.

Note Cisco access servers currently support ES-IS routing protocol and not IS-IS routing protocol.

Understanding Addresses

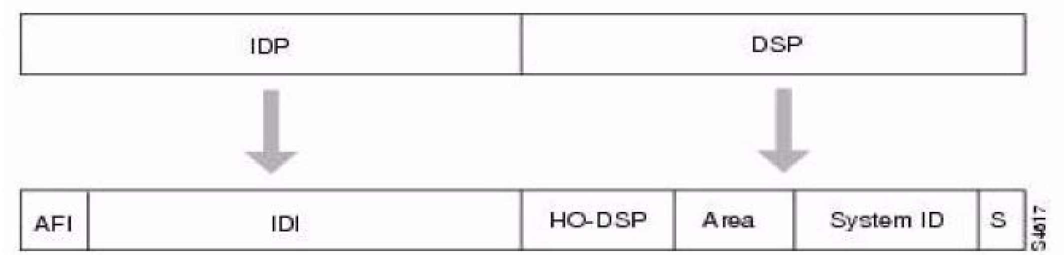
Addresses in the ISO network architecture are referred to as network service access point (NSAP) addresses and network entity titles (NETs). Each node in an OSI network has one or more NETs. In addition, each node has many NSAP addresses. Each NSAP address differs from one of the NETs for that node in only the last byte. This byte is called the *N-selector*. Its function is similar to the port number in other protocol suites.

Our implementation supports all NSAP address formats that are defined by ISO 8348/Ad2; however, Cisco provides ISO Interior Gateway Routing Protocol (IGRP) or Intermediate System-to-Intermediate System (IS-IS) dynamic routing only for NSAP addresses that conform to the address constraints defined in the ISO standard for IS-IS (ISO 10589).

An NSAP address consists of the following two major fields, as shown in Figure 1:

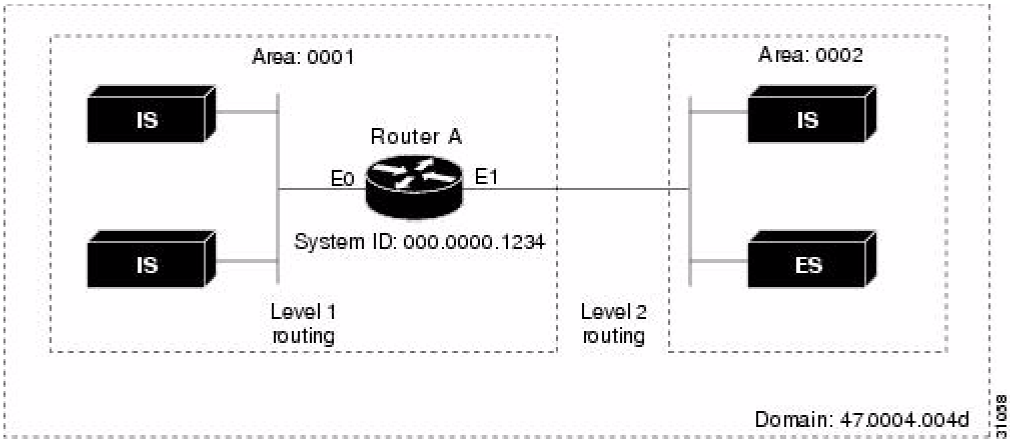
- The initial domain part (IDP) is made up of 1-byte authority and format identifier (AFI) and a variable-length initial domain identifier (IDI). The length of the IDI and the encoding format for the domain specific part (DSP) are based on the value of the AFI.
- The DSP is made up of a High Order DSP (HO-DSP), an area identifier, a system identifier, and a 1-byte N-selector (labeled S).

Figure 1 NSAP Address Fields



Assign addresses or NETs for your domains and areas. The domain address uniquely identifies the routing domain. All routers within a given domain are given the same domain address. Within each routing domain, you can set up one or more areas, as shown in Figure 2. Determine which routers are to be assigned to which areas. The area address uniquely identifies the routing area and the system ID identifies each node.

Figure 2 Sample Domain and Area Addresses



The key difference between the ISO IGRP and IS-IS NSAP addressing schemes is in the definition of area addresses. Both use the system ID for Level 1 routing (routing within an area). However, they differ in the way addresses are specified for area routing. An ISO IGRP NSAP address includes three separate fields for routing: the *domain*, *area*, and *system ID*. An IS-IS address includes two fields: a single continuous *area* field (comprising the domain and area fields) and the *system ID*.

The following topics on address are also covered in this section:

- "ISO IGRP NSAP Address" section on page 4
- "IS-IS NSAP Address" section on page 5
- "Addressing Rules" section on page 5
- "Addressing Examples" section on page 6
- "Sample Routing Table" section on page 6

ISO IGRP NSAP Address

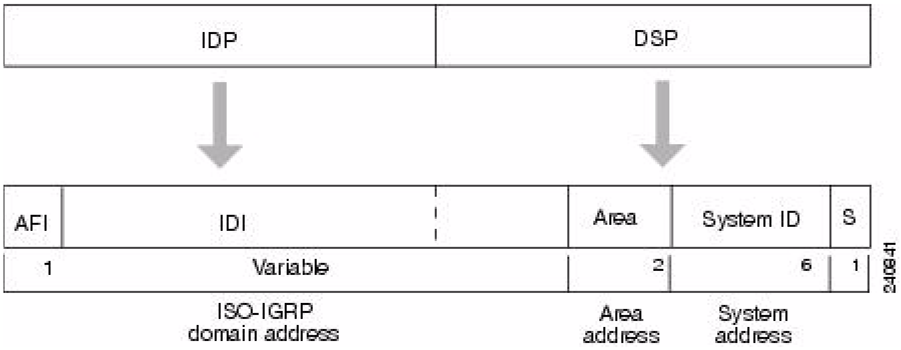
The ISO IGRP NSAP address is divided into three parts: a domain part, an area address, and a system ID. Domain routing is performed on the domain part of the address. Area routing for a given domain uses the area address. System routing for a given area uses the system ID part. The NSAP address is laid out as follows:

- The domain part is of variable length and comes before the area address.
- The area address is the 2 bytes before the system ID.
- The system ID is the 6 bytes before the N-selector.
- The N-selector (S) is the last byte of the NSAP address.

The Cisco ISO IGRP routing implementation interprets the bytes from the AFI up to (but not including) the area field in the DSP as a *domain identifier*. The area field specifies the *area*, and the system ID specifies the *system*.

Figure 3 illustrates the ISO IGRP NSAP addressing structure. The maximum address size is 20 bytes.

Figure 3 *ISO IGRP NSAP Addressing Structure*



IS-IS NSAP Address

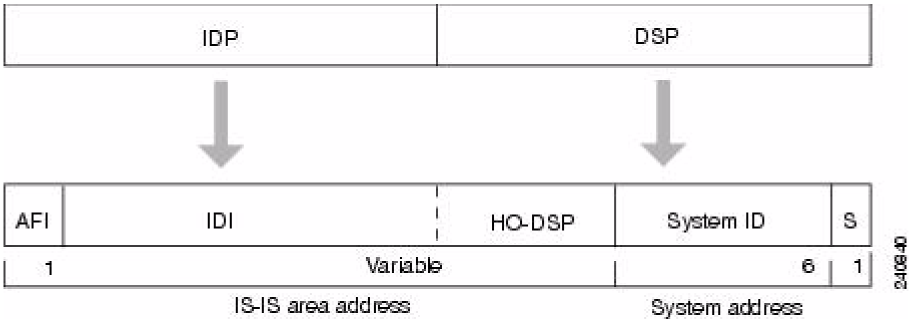
An IS-IS NSAP address is divided into two parts: an area address and a system ID. Level 2 routing (routing between areas) uses the area address. Level 1 routing (routing within an area) uses the system ID address. The NSAP address is defined as follows:

- The area address is the NSAP address, not including the system ID and N-selector.
- The system ID is found between the area address and the N-selector byte.
- The N-selector (S) is the last byte of the NSAP address.

The IS-IS routing protocol interprets the bytes from the AFI up to (but not including) the system ID field in the DSP as an *area identifier*. The system ID specifies the *system*.

Figure 4 illustrates the IS-IS NSAP addressing structure. The maximum address size is 20 bytes.

Figure 4 *IS-IS NSAP Addressing Structure*



Addressing Rules

All NSAP addresses must obey the following constraints:

- The NET for a system is normally written as an NSAP address with the N-selector byte set to zero.
- No two nodes can have addresses with the same NET; that is, addresses that match all but the N-selector (S) field in the DSP.
- No two nodes residing within the same area can have addresses in which the system ID fields are the same.

- ISO IGRP requires at least 10 bytes of length: 1 byte for domain, 2 bytes for area, 6 bytes for system ID, and 1 byte for N-selector.
- ISO IGRP and IS-IS should not be configured for the same area. Do *not* specify an NSAP address where all bytes up to (but not including) the system ID are the same when enabling both ISO IGRP and IS-IS routing.
- A router can have one or more area addresses. The concept of multiple area addresses is described in the "Assigning Multiple Area Addresses to IS-IS Areas" section in this guide.
- The Cisco implementation of IS-IS requires at least 8 bytes: one byte for area, 6 bytes for system ID, and 1 byte for N-selector.

Addressing Examples

The following examples show how to configure OSI network and Government OSI Profile (GOSIP) NSAP addresses using the ISO IGRP implementation.

The following example shows an OSI network NSAP address format:

```
|     Domain|Area|     System ID| S|
47.0004.004D.0003.0000.0C00.62E6.00
```

The following example shows an GOSIP NSAP address structure. This structure is mandatory for addresses allocated from the International Code Designator (ICD) 0005 addressing domain. Refer to the GOSIP document *U.S. Government Open Systems Interconnection Profile (GOSIP)*, draft version 2.0, April 1989, for more information.

```
|                    Domain|Area|     System ID| S|
47.0005.80.ffff00.0000.ffff.0004.0000.0c00.62e6.00
 |   |    |    |    |    |
AFI IDI  DFI  AAI  Resv  RD
```

Sample Routing Table

You enter static routes by specifying NSAP prefix and next hop NET pairs (by using the **clns route** command). The NSAP prefix can be any portion of the NSAP address. NETs are similar in function to NSAP addresses.

If an incoming packet has a destination NSAP address that does not match any existing NSAP addresses in the routing table, Cisco IOS software will try to match the NSAP address with an NSAP prefix to route the packet. In the routing table, the best match means the longest NSAP prefix entry that matches the beginning of the destination NSAP address.

Table 1 shows a sample static routing table in which the next hop NETs are listed for completeness, but are not necessary to understand the routing algorithm. Table 2 offers examples of how the longest matching NSAP prefix can be matched with routing table entries in Table 1.

Table 1 Sample Routing Table Entries

Entry	NSAP Address Prefix	Next Hop NET
1	47.0005.000c.0001	47.0005.000c.0001.0000.1234.00
2	47.0004	47.0005.000c.0002.0000.0231.00
3	47.0005.0003	47.0005.000c.0001.0000.1234.00
4	47.0005.000c	47.0005.000c.0004.0000.0011.00
5	47.0005	47.0005.000c.0002.0000.0231.00

Table 2 Hierarchical Routing Examples

Datagram Destination NSAP Address	Table Entry Number Used
47.0005.000c.0001.0000.3456.01	1
47.0005.000c.0001.6789.2345.01	1
47.0004.1234.1234.1234.1234.01	2
47.0005.0003.4321.4321.4321.01	3
47.0005.000c.0004.5678.5678.01	4
47.0005.0001.0005.3456.3456.01	5

Octet boundaries must be used for the internal boundaries of NSAP addresses and NETs.

Understanding ISO CLNS Routing Processes

The basic function of a router is to forward packets: receive a packet in one interface and send it out another (or the same) interface to the proper destination. All routers forward packets by looking up the destination address in a table. The tables can be built either dynamically or statically. If you are configuring all the entries in the table yourself, you are using *static* routing. If you use a routing process to build the tables, you are using *dynamic* routing. It is possible, and sometimes necessary, to use both static and dynamic routing simultaneously.

When you configure only ISO CLNS and not routing protocols, Cisco IOS software makes only forwarding decisions. It does not perform other routing-related functions. In such a configuration, the software compiles a table of adjacency data, but does not advertise this information. The only information that is inserted into the routing table is the NSAP and NET addresses of this router, static routes, and adjacency information.

You can route ISO CLNS on some interfaces and transparently bridge it on other interfaces simultaneously. To enable this type of routing, you must enable concurrent routing and bridging by using the **bridge crb** command. For more information on bridging, refer to the “Configuring Transparent Bridging” chapter in the *Cisco IOS Bridging and IBM Networking Configuration Guide*.

To understand ISO CLNS routing process, you should understand the following:

- “Dynamic Routing” section on page 7
- “Intermediate Systems and End Systems” section on page 8
- “Static Routing” section on page 8
- “Routing Decisions” section on page 8
- “GRE/CLNS Tunnel Support for IPv4 and IPv6 Packets” section on page 9

Dynamic Routing

Cisco supports the following two dynamic routing protocols for ISO CLNS networks:

- ISO IGRP
- IS-IS

When dynamically routing, you can choose either ISO IGRP or IS-IS, or you can enable both routing protocols at the same time. Both routing protocols support the concept of *areas*. Within an area, all routers know how to reach all the system IDs. Between areas, routers know how to reach the proper area.

ISO IGRP supports three levels of routing: *system routing*, *area routing*, and *interdomain routing*. Routing across domains (interdomain routing) can be done either statically or dynamically with ISO IGRP. IS-IS supports two levels of routing: *station routing* (within an area) and *area routing* (between areas).

Intermediate Systems and End Systems

Some intermediate systems (ISs) keep track of how to communicate with all the end systems (ESs) in their areas and thereby function as Level 1 routers (also referred to as *local routers*). Other ISs keep track of how to communicate with other areas in the domain, functioning as Level 2 routers (sometimes referred to as *area routers*). Cisco routers are always Level 1 and Level 2 routers when routing ISO IGRP; they can be configured to be Level 1 only, Level 2 only, or both Level 1 and Level 2 routers when routing IS-IS.

ESs communicate with ISs using the ES-IS protocol. Level 1 and Level 2 ISs communicate with each other using either ISO IS-IS or the Cisco ISO IGRP protocol.

Static Routing

Static routing is used when it is not possible or desirable to use dynamic routing. The following are some instances of when you would use static routing:

- If your network includes WAN links that involve paying for connect time or for per-packet charges, you would use static routing, rather than pay to run a routing protocol and all its routing update packets over that link.
- If you want routers to advertise connectivity to external networks, but you are not running an interdomain routing protocol, you *must* use static routes.
- If you must interoperate with equipment from another vendor that does not support any of the dynamic routing protocols that Cisco supports, you must use static routing.
- For operation over X.25, Frame Relay, or SMDS networks, static routing is generally preferable.

Note An interface that is configured for static routing cannot reroute *around* failed links.

Routing Decisions

A Connectionless Network Protocol (CLNP) packet sent to any of the defined NSAP addresses or NETs will be received by the router. Cisco IOS software uses the following algorithm to select which NET to use when it sends a packet:

- If no dynamic routing protocol is running, use the NET defined for the outgoing interface, if it exists; otherwise, use the NET defined for the router.
- If ISO IGRP is running, use the NET of the ISO IGRP routing process that is running on the interface.
- If IS-IS is running, use the NET of the IS-IS routing process that is running on the interface.

GRE/CLNS Tunnel Support for IPv4 and IPv6 Packets

GRE tunneling of IPv4 and IPv6 packets through CLNS networks enables Cisco CLNS tunnels (CTunnels) to interoperate with networking equipment from other vendors. This feature provides compliance with RFC 3147.

The optional GRE services defined in header fields, such as checksums, keys, and sequencing, are not supported. Any packet that is received and requests such services will be dropped.

How to Configure ISO CLNS

To configure ISO CLNS, you must configure the routing processes, associate addresses with the routing processes, and customize the routing processes for your particular network.

To configure the ISO CLNS protocol, you must use some combination of the tasks in the following sections:

- Configuring ISO IGRP Dynamic Routing (Optional)
- Configuring IS-IS Dynamic Routing (Optional)
- Configuring CLNS Static Routing (Optional)
- Configuring Miscellaneous Features (Optional)
- Configuring CLNS over WANs (Optional)
- Enhancing ISO CLNS Performance (Optional)
- Monitoring and Maintaining the ISO CLNS Network (Optional)
- Configuring TARP on ISO CLNS (Optional)

See the "ISO CLNS Configuration Examples" section at the end of this chapter for configuration examples.

Configuring ISO IGRP Dynamic Routing

The ISO IGRP is a dynamic distance-vector routing protocol designed by Cisco for routing an autonomous system that contains large, arbitrarily complex networks with diverse bandwidth and delay characteristics.

To configure ISO IGRP, perform the tasks in the following sections. The tasks in the "Configuring ISO IGRP Parameters" section are optional, although you might be required to to perform them depending upon your specific application.

- Enabling ISO IGRP (Required)
- Configuring ISO IGRP Parameters (Optional)

In addition, you can configure the following miscellaneous features described later in this chapter:

- Filter routing information—See the "Creating Packet-Forwarding Filters and Establishing Adjacencies" section.
- Redistribute routing information from one routing process to another—See the "Redistributing Routing Information" section.
- Configure administrative distances—See the "Specifying Preferred Routes" section.

Enabling ISO IGRP

To configure ISO IGRP dynamic routing, you must enable the ISO IGRP routing process, identify the address for the router, and specify the interfaces that are to route ISO IGRP. Optionally, you can set a level for your routing updates when you configure the interfaces. CLNS routing is enabled by default on routers when you configure ISO IGRP. You can specify up to ten ISO IGRP routing processes.

To configure ISO IGRP dynamic routing on the router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router iso-igrp [tag]`	Enables the ISO IGRP routing process and enters router configuration mode.
Step 2	`Router(config-router)# net network-entity-title`	Configures the NET or address for the routing process.

Although IS-IS allows you to configure multiple NETs, ISO IGRP allows only one NET per routing process.

You can assign a meaningful name for the routing process by using the *tag* option. You can also specify a name for a NET in addition to an address. For information on how to assign a name, see the "Specifying Shortcut NSAP Addresses" section later in this chapter.

You can configure an interface to advertise Level 2 information only. This option reduces the amount of router-to-router traffic by telling Cisco IOS software to send out only Level 2 routing updates on certain interfaces. Level 1 information is not passed on the interfaces for which the Level 2 option is set.

To configure ISO IGRP dynamic routing on the interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# clns router iso-igrp tag [level 2]`	Enables ISO IGRP on specified interfaces; also sets the level type for routing updates.

See the sections "Dynamic Routing in Overlapping Areas Example," "Dynamic Interdomain Routing Example," and "ISO CLNS over X.25 Example" at the end of this chapter for examples of configuring dynamic routing.

Configuring ISO IGRP Parameters

The Cisco ISO IGRP implementation allows you to customize certain ISO IGRP parameters. You can perform the optional tasks discussed in the following sections:

- Adjusting ISO IGRP Metrics (Optional)
- Adjusting ISO IGRP Timers (Optional)
- Enabling or Disabling Split Horizon (Optional)

Adjusting ISO IGRP Metrics

You have the option of altering the default behavior of ISO IGRP routing and metric computations. Altering the default behavior enables, for example, the tuning of system behavior to allow for transmissions via satellite. Although ISO IGRP metric defaults were carefully selected to provide excellent operation in most networks, you can adjust the metric.

Note Adjusting the ISO IGRP metric can dramatically affect network performance, so ensure that all metric adjustments are made carefully. Because of the complexity of this task, it is not recommended unless it is done with guidance from an experienced system designer.

You can use different metrics for the ISO IGRP routing protocol on CLNS. To configure the metric constants used in the ISO IGRP composite metric calculation of reliability and load, use the following command in router configuration mode

Command	Purpose
`Router(config-router)#` **`metric weights`** *`qos k1 k2 k3 k4 k5`*	Adjusts the ISO IGRP metric.

Two additional ISO IGRP metrics can be configured: the bandwidth and delay associated with an interface. Refer to the *Cisco IOS Interface Command Reference* publication for details about the **bandwidth** (interface) and **delay** interface configuration commands used to set these metrics.

Note Using the **bandwidth** (interface) and **delay** commands to change the values of the ISO IGRP metrics also changes the values of IP IGRP metrics.

Adjusting ISO IGRP Timers

The basic timing parameters for ISO IGRP are adjustable. Because the ISO IGRP routing protocol executes a distributed, asynchronous routing algorithm, it is important that these timers be the same for all routers in the network.

To adjust ISO IGRP timing parameters, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)#` **`timers basic`** *`update-interval holddown-interval invalid-interval`*	Adjusts the ISO IGRP timers (in seconds).

Enabling or Disabling Split Horizon

Split horizon blocks information about routes from being advertised out the interface from which that information originated. This feature usually optimizes communication among multiple routers, particularly when links are broken.

To either enable or disable split horizon for ISO IGRP updates, use the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# clns split-horizon`	Enables split horizon for ISO IGRP updates.
`Router(config-if)# no clns split-horizon`	Disables split horizon for ISO IGRP updates.

The default for all LAN interfaces is for split horizon to be enabled; the default for WAN interfaces on X.25, Frame Relay, or Switched Multimegabit Data Service (SMDS) networks is for split horizon to be disabled.

Configuring IS-IS Dynamic Routing

IS-IS is an ISO dynamic routing specification. IS-IS is described in ISO 10589. The Cisco implementation of IS-IS allows you to configure IS-IS as an ISO CLNS routing protocol.

IS-IS Configuration Task List

To configure IS-IS, perform the tasks in the following sections. Enabling IS-IS is required; the remainder of the tasks are optional, although you might be required to perform them depending upon your specific application.

- Enabling IS-IS (Required)
- Enabling Routing for an Area on an Interface (Optional)
- Assigning Multiple Area Addresses to IS-IS Areas (Optional)
- Configuring IS-IS Interface Parameters (Optional)
- Configuring Miscellaneous IS-IS Parameters (Optional)

In addition, you can configure the following miscellaneous features described later in this chapter:

- Filter routing information—See the "Creating Packet-Forwarding Filters and Establishing Adjacencies" section.
- Redistribute routing information from one routing process to another—See the "Redistributing Routing Information" section.
- Configure administrative distances—See the "Specifying Preferred Routes" section.

Enabling IS-IS

Unlike other routing protocols, enabling IS-IS requires that you create an IS-IS routing process and assign it to a specific interface, rather than to a network. You can specify more than one IS-IS routing process per Cisco unit, using the multiarea IS-IS configuration syntax. You then configure the parameters for each instance of the IS-IS routing process.

Small IS-IS networks are built as a single area that includes all the routers in the network. As the network grows larger, it is usually reorganized into a backbone area made up of the connected set of all Level 2 routers from all areas, which is in turn connected to local areas. Within a local area, routers know how to reach all system IDs. Between areas, routers know how to reach the backbone, and the backbone routers know how to reach other areas.

Routers establish Level 1 adjacencies to perform routing within a local area (intra-area routing). Routers establish Level 2 adjacencies to perform routing between Level 1 areas (interarea routing).

Some networks use legacy equipment that supports only Level 1 routing. These devices are typically organized into many small areas that cannot be aggregated due to performance limitations. Cisco routers are used to interconnect each area to the Level 2 backbone.

A single Cisco router can participate in routing in up to 29 areas and can perform Level 2 routing in the backbone. In general, each routing process corresponds to an area. By default, the first instance of the routing process configured performs both Level 1and Level 2 routing. You can configure additional router instances, which are automatically treated as Level 1 areas. You must configure the parameters for each instance of the IS-IS routing process individually.

Note The CPU memory required to run 29 ISIS processes will probably not be present in low-end platforms unless the routing information and area topology are limited.

For IS-IS multiarea routing, you can configure only one process to perform Level 2 routing, although you can define up to 29 Level 1 areas for each Cisco unit. If Level 2 routing is configured on any process, all additional processes are automatically configured as Level 1. You can configure this process to perform Level 1 routing at the same time. If Level 2 routing is not desired for a router instance, remove the Level 2 capability using the **is-type** command. Use the **is-type** command also to configure a different router instance as a Level 2 router.

To enable IS-IS, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# **router isis** [*area-tag*]`	Enables IS-IS routing for the specified routing process and places you in router configuration mode. Use the *area-tag* argument to identify the area to which this IS-IS router instance is assigned. A value for *tag* is required if you are configuring multiple IS-IS areas. The first IS-IS instance configured is Level 1-2 by default. Later instances are automatically Level 1. You can change the level of routing to be performed by a particular routing process using the **is-type** command.
Step 2	`Router(config-router)# **net** *network-entity-title*`	Configures NETs for the routing process. Specify a NET for each routing process if you are configuring multiarea IS-IS. You can specify a name for a NET and for an address.

You can assign a meaningful name for the routing process by using the *tag* option. You can also specify a name for a NET in addition to an address. For information on how to assign a name, see the "Specifying Shortcut NSAP Addresses" section later in this chapter.

See the "IS-IS Routing Configuration Examples" section at the end of this chapter for examples of configuring IS-IS routing.

Enabling Routing for an Area on an Interface

To enable CLNS routing and specify the area for each instance of the IS-IS routing process, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Enters interface configuration mode.
Step 2	`Router(config-if)# clns router isis [area-tag]`	Specifies that the interface is actively routing IS-IS when the network protocol is ISO-CLNS, and identifies the area associated with this routing process on this interface.
Step 3	`Router(config-if)# ip address ip-address mask`	Defines the IP address for the interface. An IP address is required on an interface if you want to use the Integrated IS-IS routing protocol over that interface. The Integrated ISIS routing protocol can be used as the routing protocol for IP based networks as well as CLNS based networks.

See the "IS-IS Routing Configuration Examples" section at the end of this chapter for examples of configuring IS-IS routing.

Assigning Multiple Area Addresses to IS-IS Areas

IS-IS routing supports the assignment of multiple area addresses on the same router. This concept is referred to as *multihoming*. Multihoming provides a mechanism for smoothly migrating network addresses, as follows:

- Splitting up an area—Nodes within a given area can accumulate to a point that they are difficult to manage, cause excessive traffic, or threaten to exceed the usable address space for an area. Multiple area addresses can be assigned so that you can smoothly partition a network into separate areas without disrupting service.
- Merging areas—Use transitional area addresses to merge as many as three separate areas into a single area that shares a common area address.
- Change to a different address—You may need to change an area address for a particular group of nodes. Use multiple area addresses to allow incoming traffic intended for an old area address to continue being routed to associated nodes.

You must statically assign multiple area addresses on a router. Cisco currently supports assignment of up to three area addresses on a router. All the addresses must have the same system ID. For example, you can assign one address (*area1* plus system ID), and two additional addresses in different areas (*area2* plus system ID and *area3* plus system ID) where the system ID is the same. The number of areas allowed in a domain is unlimited.

A router can dynamically learn about any adjacent router. As part of this process, the routers inform each other of their area addresses. If two routers share at least one area address, the set of area addresses of the two routers are merged. A merged set cannot contain more than three addresses. If there are more than three, the three addresses with the lowest numerical values are kept, and all others are dropped.

To configure multiple area addresses in IS-IS areas, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# router isis [area-tag]`	Enables IS-IS routing for the specified routing process and places you in router configuration mode. Use the *area-tag* argument to identify the area to which this IS-IS router instance is assigned. A value for *area-tag* is required if you are configuring multiarea IS-IS. A value for *area-tag* is optional if you are configuring conventional IS-IS. The first IS-IS instance configured is Level 1-2 by default. Later instances are automatically Level 1. You can change the level of routing to be performed by a particular routing process using the **is-type** command.
Step 2	`Router(config-router)# net network-entity-title`	Configures NETs for the routing process. Specify a NET for each routing process if you are configuring multiarea IS-IS. You can specify a name for a NET and for an address.

See the “NETs Configuration Examples” section at the end of this chapter for examples of configuring NETs and multiple area addresses.

Configuring IS-IS Interface Parameters

The Cisco IS-IS implementation allows you to customize certain interface-specific IS-IS parameters. You can perform the optional tasks discussed in the following sections:

- Adjusting IS-IS Link-State Metrics (Optional)
- Setting the Advertised Hello Interval and Hello Multiplier (Optional)
- Setting the Advertised Complete Sequence Number PDU Interval (Optional)
- Setting the Retransmission Interval (Optional)
- Setting the Retransmission Throttle Interval (Optional)
- Specifying Designated Router Election (Optional)
- Specifying the Interface Circuit Type (Optional)
- Configuring IS-IS Authentication Passwords (Optional)
- Limiting LSP Flooding (Optional)

You are not required to alter any of these parameters, but some interface parameters must be consistent across all routers in the network. Therefore, be sure that if you do configure any of these parameters, the configurations for all routers on the network have compatible values.

Adjusting IS-IS Link-State Metrics

You can configure a cost for a specified interface. The default metric is used as a value for the IS-IS metric and is assigned when there is no quality of service (QoS) routing performed. The only metric that is supported by Cisco IOS software and that you can configure is the *default-metric*, which you can configure for Level 1 or Level 2 routing or both. The range for the *default-metric* is from 0 to 63. The default value is 10.

To configure the link-state metric, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isis metric default-metric [level-1 | level-2]`	Configures the metric (or cost) for the specified interface.

Setting the Advertised Hello Interval and Hello Multiplier

You can specify the length of time (in seconds) between hello packets that Cisco IOS software sends on the interface. You can also change the default hello packet multiplier used on the interface to determine the hold time sent in IS-IS hello packets (the default is 3).

The hold time determines how long a neighbor waits for another hello packet before declaring the neighbor down. This time determines how quickly a failed link or neighbor is detected so that routes can be recalculated.

To set the advertised hello interval and multiplier, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# isis hello-interval seconds [level-1 | level-2]`	Specifies the length of time between hello packets that Cisco IOS software sends.
Step 2	`Router(config-if)# isis hello-multiplier multiplier [level-1 | level-2]`	Specifies the number of IS-IS hello packets a neighbor must miss before the router should declare the adjacency as down.

The hello interval can be configured independently for Level 1 and Level 2, except on serial point-to-point interfaces. (Because there is only a single type of hello packet sent on serial links, the hello packet is independent of Level 1 or Level 2.) Specify an optional level for X.25, SMDS, and Frame Relay multiaccess networks.

Use the **isis hello-multiplier** command in circumstances where hello packets are lost frequently and IS-IS adjacencies are failing unnecessarily. You can raise the hello multiplier and lower the hello interval (the **isis hello-interval** command) correspondingly to make the hello protocol more reliable without increasing the time required to detect a link failure.

Setting the Advertised Complete Sequence Number PDU Interval

Complete sequence number PDUs (CSNPs) are sent by the designated router to maintain database synchronization.

To configure the IS-IS CSNP interval for the interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isis csnp-interval seconds [level-1 | level-2]`	Configures the IS-IS CSNP interval for the specified interface.

The **isis csnp-interval** command does not apply to serial point-to-point interfaces. It does apply to WAN connections if the WAN is viewed as a multiaccess meshed network.

Setting the Retransmission Interval

You can configure the number of seconds between retransmission of Link-State PDUs (LSPs) for point-to-point links.

To set the retransmission level, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isis retransmit-interval seconds`	Configures the number of seconds between retransmission of IS-IS LSPs for point-to-point links.

The value you specify should be an integer greater than the expected round-trip delay between any two routers on the network. The setting of this parameter should be conservative, or needless retransmission will result. The value you determine should be larger for serial lines and virtual links.

Setting the Retransmission Throttle Interval

You can configure the maximum rate (number of milliseconds between packets) at which IS-IS LSPs will be re-sent on point-to-point links This interval is different from the retransmission interval, the time between successive retransmissions of the *same* LSP.

To set the retransmission throttle interval, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isis retransmit-throttle-interval milliseconds`	Configures the IS-IS LSP retransmission throttle interval.

This command is usually unnecessary, except when very large networks contain high point-to-point neighbor counts.

Specifying Designated Router Election

You can configure the priority to use for designated router election. Priorities can be configured for Level 1 and Level 2 individually. The designated router enables a reduction in the number of adjacencies required on a multiaccess network, which in turn reduces the amount of routing protocol traffic and the size of the topology database.

To configure the priority to use for designated router election, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isis priority value [level-1 | level-2]`	Configures the priority to use for designated router election.

Specifying the Interface Circuit Type

It is normally not necessary to configure this feature because the IS-IS protocol automatically determines area boundaries and keeps Level 1 and Level 2 routing separate. However, you can specify the adjacency levels on a specified interface.

To configure the adjacency for neighbors on the specified interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isis circuit-type [level-1 | level-1-2 | level-2-only]`	Configures the type of adjacency desired for neighbors on the specified interface (specifies the interface circuit type).

If you specify Level 1, a Level 1 adjacency is established if there is at least one area address common to both this node and its neighbors.

If you specify both Level 1 and Level 2 (the default value), a Level 1 and 2 adjacency is established if the neighbor is also configured as both Level 1 and Level 2 and there is at least one area in common. If there is no area in common, a Level 2 adjacency is established.

If you specify Level 2 only, a Level 2 adjacency is established. If the neighbor router is a Level 1 router, no adjacency is established.

Configuring IS-IS Authentication Passwords

You can assign different authentication passwords for different routing levels. By default, authentication is disabled. Specifying Level 1 or Level 2 enables the password only for Level 1 or Level 2 routing, respectively. If you do not specify a level, the default is Level 1.

To configure an authentication password for an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isis password password [level-1 | level-2]`	Configures the authentication password for an interface.

You can assign authentication passwords to areas and domains. An area password is inserted in Level 1 (station router) LSPs, CSNPs, and partial sequence number PDUs (PSNPs). A routing domain authentication password is inserted in Level 2 (area router) LSPs, CSNPs, and PSNPs.

To configure area or domain passwords, use the following commands in router configuration mode:

Command	Purpose
`Router(config-router)# area-password password`	Configures the area authentication password.
`Router(config-router)# domain-password password`	Configures the routing domain authentication password.

Limiting LSP Flooding

Limiting LSP flooding is important to IS-IS networks in general, and is not limited to configuring multiarea IS-IS networks. In a network with a high degree of redundancy, such as a fully meshed set of point-to-point links over a nonbroadcast multiaccess (NBMA) transport, flooding of LSPs can limit network scalability. You can reduce LSP flooding in two ways:

- Blocking Flooding on Specific Interfaces

  The advantage of full blocking over mesh groups is that it is easier to configure and understand, and fewer LSPs are flooded. Blocking flooding on all links permits the best scaling performance, but results in a less robust network structure. Permitting flooding on all links results in poor scaling performance.

- Configuring Mesh Groups

  The advantage of mesh groups over full blocking is that mesh groups allow LSPs to be flooded over one hop to all routers on the mesh, while full blocking allows some routers to receive LSPs over multiple hops. This relatively small delay in flooding can have an impact on convergence times, but the delay is negligible compared to overall convergence times.

Blocking Flooding on Specific Interfaces

You can completely block flooding (full blocking) on specific interfaces, so that new LSPs will not be flooded out over those interfaces. However, if flooding is blocked on a large number of links, and all remaining links go down, routers cannot synchronize their link-state databases even though there is connectivity to the rest of the network. When the link-state database is no longer updated, routing loops usually result.

To use CSNPs on selected point-to-point links to synchronize the link-state database, configure a CSNP interval using the **isis csnp-interval** command on selected point-to-point links over which normal flooding is blocked. You should use CSNPs for this purpose only as a last resort.

Configuring Mesh Groups

Configuring mesh groups (a set of interfaces on a router) can help to limit redundant flooding. All routers reachable over the interfaces in a particular mesh group are assumed to be densely connected (each router has many links to other routers), where many links can fail without isolating one or more routers from the network.

Normally, when a new LSP is received on an interface, it is flooded out over all other interfaces on the router. When the new LSP is received over an interface that is part of a mesh group, the new LSP will not be flooded out over the other interfaces that are part of that same mesh group.

Mesh groups rely on a full mesh of links between a group of routers. If one or more links in the full mesh goes down, the full mesh is broken, and some routers might miss new LSPs, even though there is connectivity to the rest of the network. When you configure mesh groups to optimize or limit LSP flooding, be sure to select alternative paths over which to flood in case interfaces in the mesh group go down.

To minimize the possibility of incomplete flooding, you should allow unrestricted flooding over at least a minimal set of links in the mesh. Selecting the smallest set of logical links that covers all physical paths results in very low flooding, but less robustness. Ideally you should select only enough links to ensure that LSP flooding is not detrimental to scaling performance, but enough links to ensure that under most failure scenarios no router will be logically disconnected from the rest of the network.

Configuring Miscellaneous IS-IS Parameters

The Cisco IS-IS implementation allows you to customize certain IS-IS parameters. You can perform the optional tasks discussed in the following sections:

- Specifying Router-Level Support (Optional)
- Ignoring IS-IS LSP Errors (Optional)
- Logging Adjacency State Changes (Optional)
- Changing IS-IS LSP Maximum Transmission Unit Size (Optional)
- Enabling Partitioning Avoidance (Optional)
- Changing the Routing Level for an Area (Optional)
- Modifying the Output of show Commands (Optional)

Specifying Router-Level Support

It is seldom necessary to configure the IS type because the IS-IS protocol will automatically establish the IS type. However, you can configure the router to act as a Level 1 (intra-area) router, as both a Level 1 router and a Level 2 (interarea) router, or as an interarea router only.

To configure the IS-IS level, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# is-type [level-1 | level-1-2 | level-2-only]`	Configures the IS-IS level at which the router is to operate.

Ignoring IS-IS LSP Errors

You can configure the router to ignore IS-IS LSPs that are received with internal checksum errors, rather than purging the LSPs. LSPs are used by the receiving routers to maintain their routing tables.

The IS-IS protocol definition requires that a received LSP with an incorrect data-link checksum be purged by the receiver, which causes the initiator of the LSP to regenerate it. However, if a network has a link that causes data corruption while still delivering LSPs with correct data-link checksums, a continuous cycle of purging and regenerating large numbers of LSPs can occur, rendering the network nonfunctional.

To allow the router to ignore LSPs with an internal checksum error, use the following command in router configuration mode:

Command	Purpose
`Router(config)# router isis`	Specifies the IS-IS routing protocol, and specifies an IS-IS process.
`Router(config-router)# ignore-lsp-errors`	Ignores LSPs with internal checksum errors rather than purging the LSPs.

Note By default, the **ignore-lsp-errors** command is enabled; that is, corrupted LSPs are dropped instead of purged for network stability. If you want to explicitly purge the corrupted LSPs, issue the **no ignore-lsp-errors** command.

Logging Adjacency State Changes

You can configure IS-IS to generate a log message when an IS-IS adjacency changes state (up or down). Generating a log message may be useful when monitoring large networks. Messages are logged using the system error message facility. Messages are of the following form:

```
%CLNS-5-ADJCHANGE: ISIS: Adjacency to 0000.0000.0034 (Serial0) Up, new adjacency
%CLNS-5-ADJCHANGE: ISIS: Adjacency to 0000.0000.0034 (Serial0) Down, hold time expired
```

To generate log messages when an IS-IS adjacency changes state, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# log-adjacency-changes`	Logs IS-IS adjacency state changes.

Changing IS-IS LSP Maximum Transmission Unit Size

Under normal conditions, the default maximum transmission unit (MTU) size should be sufficient. However, if the MTU of a link is lowered to less than 1500 bytes, the LSP MTU must be lowered accordingly on each router in the network. If LSP MTU is not lowered, routing will become unpredictable.

The MTU size must be less than or equal to the smallest MTU of any link in the network. The default size is 1497 bytes.

Caution The CLNS MTU of a link (which is the applicable value for IS-IS, even if it is being used to route IP) may differ from the IP MTU. To be certain about a link MTU as it pertains to IS-IS, use the **show clns interface** command to display the value.

To change the MTU size of IS-IS LSPs, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# lsp-mtu size`	Specifies the maximum LSP packet size, in bytes.

Note If any link in the network has a reduced MTU, all routers must be changed, not just the routers directly connected to the link. This rule applies to all routers in a network.

Enabling Partitioning Avoidance

In ISO CLNS networks using a redundant topology, it is possible for an area to become “partitioned” when full connectivity is lost among a Level 1-2 border router, all adjacent Level 1 routers, and end hosts. In such a case, multiple Level 1-2 border routers advertise the Level 1 area prefix into the backbone area, even though any one router can reach only a subset of the end hosts in the Level 1 area.

When enabled, the **partition avoidance** command prevents this partitioning by causing the border router to stop advertising the Level 1 area prefix into the Level 2 backbone.

Other cases of connectivity loss within the Level 1 area itself are not detected or corrected by the border router, and this command has no effect.

To enable partitioning avoidance, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# partition avoidance`	Causes an IS-IS Level 1-2 border router to stop advertising the Level 1 area prefix into the Level 2 backbone when full connectivity is lost among the border router, all adjacent Level 1 routers, and end hosts.

Changing the Routing Level for an Area

You can change the routing level configured for an area using the **is-type** command. If the router instance has been configured for Level 1-2 area (the default for the first instance of the IS-IS routing process in a Cisco unit), you can remove Level 2 (interarea) routing for the area using the **is-type** command and change the routing level to Level 1 (intra-area). You can also configure Level 2 routing for an area using the **is-type** command, but the instance of the IS-IS router configured for Level 2 on the Cisco unit must be the only instance configured for Level 2.

To change the routing level for an IS-IS routing process in a given area, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)#` **`is-type`** `[`**`level-1`** `|` **`level-1-2`** `|` **`level-2-only`**`]`	Configures the routing level for an instance of the IS-IS routing process.

Modifying the Output of show Commands

To customize display output when the multiarea feature is used, making the display easier to read, use the following command in EXEC mode:

Command	Purpose
`Router#` **`isis display delimiter`** `[`**`return`** *`cnt`* `|`*`char cnt`*`]`	Specifies the delimiter to be used to separate displays of information about individual IS-IS areas.

For example, the following command causes information about individual areas to be separated by 14 hyphens (-) in the display:

```
isis display delimiter - 14
```

The output for a configuration with two Level 1 areas and one Level 2 area configured is as follows:

```
dtp-5# show clns neighbors
--------------
Area L2BB:
System Id      Interface   SNPA                State  Holdtime  Type Protocol
0000.0000.0009 Tu529       172.21.39.9         Up     25        L1L2 IS-IS
--------------
Area A3253-01:
System Id      Interface   SNPA                State  Holdtime  Type Protocol
0000.0000.0053 Et1         0060.3e58.ccdb      Up     22        L1   IS-IS
0000.0000.0003 Et1         0000.0c03.6944      Up     20        L1   IS-IS
--------------
Area A3253-02:
System Id      Interface   SNPA                State  Holdtime  Type Protocol
0000.0000.0002 Et2         0000.0c03.6bc5      Up     27        L1   IS-IS
0000.0000.0053 Et2         0060.3e58.ccde      Up     24        L1   IS-IS
```

Configuring CLNS Static Routing

You need not explicitly specify a routing process to use static routing facilities. You can enter a specific static route and apply it globally, even if you have configured the router for ISO IGRP or IS-IS dynamic routing.

To configure a static route, perform the tasks in the following sections. Only enabling CLNS is required; the remaining tasks are optional, although you might be required to perform them depending upon your specific application.

- Enabling Static Routes (Required)
- Configuring Variations of the Static Route (Optional)

- Mapping NSAP Addresses to Media Addresses (Optional)

Enabling Static Routes

To configure static routing, you must enable CLNS on the router and on the interface. CLNS routing is enabled on the router by default when you configure ISO IGRP or IS-IS routing protocols. NSAP addresses that start with the NSAP prefix you specify are forwarded to the next hop node.

To configure CLNS on the router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# clns routing`	Configures CLNS.
Step 2	`Router(config-if)# clns net {net-address | name}`	Assigns an NSAP address to the router if the router has not been configured to route CLNS packets dynamically using ISO IGRP or IS-IS.
Step 3	`Router(config)# clns route nsap-prefix {next-hop-net | name}`	Enters a specific static route.

Note If you have not configured the router to route CLNS packets dynamically using ISO IGRP or IS-IS, you must assign an address to the router.

You also must enable ISO CLNS for each interface you want to pass ISO CLNS packet traffic to end systems, but for which you do not want to perform any dynamic routing on the interface. ISO CLNS is enabled automatically when you configure IS-IS or ISO IGRP routing on an interface; however, if you do not intend to perform any dynamic routing on an interface, you must manually enable CLNS. You can assign an NSAP address for a specific interface. Assigning an NSAP address allows Cisco IOS software to advertise different addresses on each interface. Advertising different addresses is useful if you are doing static routing and need to control the source NET used by the router on each interface.

To configure CLNS on an interface, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# clns enable`	Enables ISO CLNS for each interface.
Step 2	`Router(config-if)# clns net {nsap-address | name}`	Assigns an NSAP address to a specific interface.

See the "Basic Static Routing Examples," "Static Intradomain Routing Example," and "Static Interdomain Routing Example" sections at the end of this chapter for examples of configuring static routes.

Configuring Variations of the Static Route

You can perform the following tasks that use variations of the **clns route** global configuration command:

- Bind the next hop to a specified interface and media address when you do not know the NSAP address of your neighbor. Note that this version of the **clns route** command is not literally *applied* to a specific interface.
- Discard packets with a specific NSAP prefix that is outside the domain (ISO IGRP) or area (IS-IS) of the router.

- Specify a default prefix.

To enter a specific static route, discard packets, or configure a default prefix, use one or all of the following commands in global configuration mode:

Note To discard or filter packets that have an NSAP prefix within the domain (ISO IGRP) or area (IS-IS) of the router, refer to the "Creating Packet-Forwarding Filters and Establishing Adjacencies" section of this chapter.

Command	Purpose
`Router(config)# clns route nsap-prefix type number [snpa-address]`	Enters a specific static route for a specific interface.
`Router(config)# clns route nsap-prefix discard`	Explicitly tells the software to discard packets with the specified NSAP prefix.
`Router(config)# clns route default type number`	Configures a default prefix rather than specify an NSAP prefix.

Mapping NSAP Addresses to Media Addresses

Conceptually, each ES lives in one area. It discovers the nearest IS by listening to ES-IS packets. Each ES must be able to communicate directly with an IS in its area.

When an ES wants to communicate with another ES, it sends the packet to any IS on the same medium.

1. The IS looks up the destination NSAP address and forwards the packet along the best route. If the destination NSAP address is for an ES in another area, the Level 1 IS sends the packet to the nearest Level 2 IS.
2. The Level 2 IS forwards the packet along the best path for the destination area until it gets to a Level 2 IS that is in the destination area.
3. This IS then forwards the packet along the best path inside the area until it is delivered to the destination ES.

ESs need to know how to get to a Level 1 IS for their area, and Level 1 ISs need to know all of the ESs that are directly reachable through each of their interfaces. To provide this information, the routers support the ES-IS protocol. The router dynamically discovers all ESs running the ES-IS protocol. ESs that are not running the ES-IS protocol must be configured statically.

It is sometimes desirable for a router to have a neighbor configured statically rather than learned through ES-IS, ISO IGRP, or IS-IS.

Note It is necessary to use static mapping only for ESs that do *not* support ES-IS. Cisco IOS software continues to dynamically discover ESs that *do* support ES-IS.

Note If you have configured interfaces for ISO CLNS, ISO IGRP, or IS-IS, the ES-IS routing software automatically turns on ES-IS for those interfaces.

To enter static mapping information between the NSAP protocol addresses and the subnetwork point of attachment (SNPA) addresses (media) for ESs or ISs, use the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# clns es-neighbor nsap snpa`	Configures all end systems that will be used when you manually specify the NSAP-to-SNPA mapping.
`Router(config-if)# clns is-neighbor nsap snpa`	Configures all intermediate systems that will be used when you manually specify the NSAP-to-SNPA mapping.

For more information, see the "Configuring CLNS over WANs" section later in this chapter.

Note The SNPA is a data link layer address (such as an Ethernet address, X.25 address, or Frame Relay DLCI address) used to configure a CLNS route for an interface.

Configuring Miscellaneous Features

To configure miscellaneous features of an ISO CLNS network, perform the optional tasks in the following sections:

- Specifying Shortcut NSAP Addresses (Optional)
- Using the IP Domain Name System to Discover ISO CLNS Addresses (Optional)
- Creating Packet-Forwarding Filters and Establishing Adjacencies (Optional)
- Redistributing Routing Information (Optional)
- Specifying Preferred Routes (Optional)
- Configuring ES-IS Hello Packet Parameters (Optional)
- Configuring DECnet OSI or Phase V Cluster Aliases (Optional)
- Configuring Digital-Compatible Mode (Optional)
- Allowing Security Option Packets to Pass (Optional)

Specifying Shortcut NSAP Addresses

You can define a name-to-NSAP address mapping. This name can then be used in place of typing the long set of numbers associated with an NSAP address.

To define a name-to-NSAP address mapping, use the following command in global configuration mode:

Command	Purpose
`Router(config)# clns host name nsap`	Defines a name-to-NSAP address mapping.

The assigned NSAP name is displayed, where applicable, in **show** and **debug** EXEC commands. However, some effects and requirements are associated with using names to represent NETs and NSAP addresses.

The **clns host** global configuration command is generated after all other CLNS commands when the configuration file is parsed. As a result, you cannot edit the NVRAM version of the configuration to specifically change the address defined in the original **clns host** command. You must specifically change any commands that refer to the original address. These changes affect all commands that accept names.

The commands that are affected by these requirements include the following:

- **net** (router configuration command)
- **clns is-neighbor** (interface configuration command)
- **clns es-neighbor** (interface configuration command)
- **clns route** (global configuration command)

Using the IP Domain Name System to Discover ISO CLNS Addresses

If your router has both ISO CLNS and IP enabled, you can use the Domain Naming System (DNS) to query ISO CLNS addresses by using the NSAP address type, as documented in RFC 1348. This feature is useful for the ISO CLNS **ping** EXEC command and when making Telnet connections. This feature is enabled by default.

To enable or disable DNS queries for ISO CLNS addresses, use the following commands in global configuration mode:

Command	Purpose
`Router(config)#` **`ip domain-lookup nsap`**	Enables DNS queries for CLNS addresses.
`Router(config)#` **`no ip domain-lookup nsap`**	Disables DNS queries for CLNS addresses.

Creating Packet-Forwarding Filters and Establishing Adjacencies

You can build powerful CLNS filter expressions, or access lists. These filter expressions can be used to control either the forwarding of frames through router interfaces, or the establishment of adjacencies with, or the application of filters to, any combination of ES or IS neighbors, ISO IGRP neighbors, or IS-IS neighbors.

CLNS filter expressions are complex logical combinations of CLNS filter sets. CLNS filter sets are lists of address templates against which CLNS addresses are matched. Address templates are CLNS address *patterns* that are either simple CLNS addresses that match just one address, or match multiple CLNS addresses through the use of wildcard characters, prefixes, and suffixes. Frequently used address templates can be given *aliases* for easier reference.

To establish CLNS filters, use the following commands in global configuration mode:

Command	Purpose
`Router(config)#` **`clns template-alias`** *`name template`*	Creates aliases for frequently used address templates.
`Router(config)#` **`clns filter-set`** *`sname`* `[`**`permit`** `|` **`deny`**`]` *`template`*	Builds filter sets of multiple address template permit and deny conditions.
`Router(config)#` **`clns filter-expr`** *`ename term`*	Builds filter expressions, using one or more filter sets.

To apply filter expressions to an interface, use the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# clns access-group name [in | out]`	Applies a filter expression to frames forwarded in or out of an interface.
`Router(config-if)# isis adjacency-filter name [match-all]`	Applies a filter expression to IS-IS adjacencies.
`Router(config-if)# iso-igrp adjacency-filter name`	Applies a filter expression to ISO IGRP adjacencies.
`Router(config-if)# clns adjacency-filter {es | is} name`	Applies a filter expression to ES or IS adjacencies that were formed by ES-IS.

See the "CLNS Filter Examples" section at the end of this chapter for examples of configuring CLNS filters.

Redistributing Routing Information

In addition to running multiple routing protocols simultaneously, Cisco IOS software can redistribute information from one routing process to another. In CLNS routing, there is no redistribution of Level 1 host routes into Level 2. Only Level 1 addresses are advertised into Level 2.

For IS-IS routing, redistribution of all area addresses of all Level 1 areas into Level 2 is implicit, and no additional configuration is required for this redistribution. Explicit redistribution between IS-IS areas cannot be configured. Redistribution from any other routing protocol into a particular area is possible, and is configured per router instance using the **redistribute** and **route map** commands. By default, redistribution is into Level 2.

You can also configure Cisco IOS software to do interdomain dynamic routing by configuring two routing processes and two NETs (thereby putting the router into two domains) and redistributing the routing information between the domains. Routers configured this way are referred to as *border* routers. If you have a router that is in two routing domains, you might want to redistribute routing information between the two domains.

Note It is not necessary to use redistribution between areas. Redistribution only occurs for Level 2 routing.

To configure the router to redistribute routing information into the ISO IGRP domain, use the following commands beginning in global configuration mode:

Command	Purpose
`Router(config)# router iso-igrp [tag]`	Specifies the routing protocol and tag (if applicable) into which you want to distribute routing information.
`Router(config-router)# redistribute iso-igrp [tag] [route-map map-tag]`	Specifies one or more ISO IGRP routing protocol and tag (if applicable) you want to redistribute.
`Router(config-router)# redistribute isis [tag] [route-map map-tag]`	Specifies the IS-IS routing protocol and tag (if applicable) you want to redistribute.
`Router(config-router)# redistribute static [clns | ip]`	Specifies the static routes you want to redistribute.

To configure the router to redistribute routing information into the IS-IS domains, use the following commands beginning in global configuration mode:

Command	Purpose
`Router(config)# router isis [tag]`	Specifies the routing protocol and tag (if applicable) into which you want to distribute routing information.
`Router(config-router)# redistribute isis [tag] [route-map map-tag]`	Specifies the IS-IS routing protocol and tag (if applicable) you want to redistribute.
`Router(config-router)# redistribute iso-igrp [tag] [route-map map-tag]`	Specifies one or more ISO IGRP routing protocol and tag (if applicable) you want to redistribute.
`Router(config-router)# redistribute static [clns | ip]`	Specifies the static routes you want to redistribute.

Note By default, static routes are redistributed into IS-IS.

You can conditionally control the redistribution of routes between routing domains by defining *route maps* between the two domains. Route maps allow you to use tags in routes to influence route redistribution.

To conditionally control the redistribution of routes between domains, use the following command in global configuration mode:

Command	Purpose
`Router(config)# route-map map-tag {permit | deny} sequence-number`	Defines any route maps needed to control redistribution.

One or more **match** command and one or more **set** commands typically follow a **route-map** command to define the conditions for redistributing routes from one routing protocol into another. If there are no **match** commands, everything matches. If there are no **set** commands, nothing is done (other than the match).

Each **route-map** command has a list of **match** and **set** commands associated with it. The **match** commands specify the *match criteria*—the conditions under which redistribution is allowed for the current **route-map command**. The **set** commands specify the redistribution *set actions*—the particular redistribution actions to perform if the criteria enforced by the **match** commands are met. When all **match** criteria are met, all **set** actions are performed

The **match route-map** configuration command has multiple formats. The **match** commands may be given in any order, and *all* defined match criteria must be satisfied to cause the route to be redistributed according to the *set actions* given with the **set** commands.

To define the match criteria for redistribution of routes from one routing protocol into another, use at least one of the following commands in route-map configuration mode:

Command	Purpose
`Router(config-route-map)# `**`match clns address`** *`name`* `[`*`name...name`*`]`	Matches routes that have a network address matching one or more of the specified names (the names can be a standard access list, filter set, or expression).
`Router(config-route-map)# `**`match clns next-hop`** *`name`* `[`*`name...name`*`]`	Matches routes that have a next hop address matching one or more of the specified names (the names can be a standard access list, filter set, or expression).
`Router(config-route-map)# `**`match clns route-source`** *`name`* `[`*`name...name`*`]`	Matches routes that have been advertised by routers matching one or more of the specified names (the names can be a standard access list, filter set, or expression).
`Router(config-route-map)# `**`match interface`** *`type number`* `[`*`type number...type number`*`]`	Matches routes that have the next hop out matching one or more of the specified interfaces.
`Router(config-route-map)# `**`match metric`** *`metric-value`*	Matches routes that have the specified metric.
`Router(config-route-map)# `**`match route-type`** `[`**`level-1`** `|` **`level-2`**`]`	Matches routes that have the specified route type.

To define set actions for redistribution of routes from one routing protocol into another, use at least one of the following commands in route-map configuration mode:

Command	Purpose
`Router(config-route-map)# `**`set level`** `[`**`level-1`** `|` **`level-2`** `|` **`level-1-2`**`]`	Sets the routing level of the routes to be advertised into a specified area of the routing domain.
`Router(config-route-map)# `**`set metric`** *`metric-value`*	Sets the metric value to give the redistributed routes.
`Router(config-route-map)# `**`set metric-type`** `{`**`internal`** `|` **`external`**`}`	Sets the metric type to give the redistributed routes.
`Router(config-route-map)# `**`set tag`** *`tag-value`*	Sets the tag value to associate with the redistributed routes.

See the “Dynamic Interdomain Routing Example” and “TARP Configuration Examples” sections at the end of this chapter for examples of configuring route maps.

Specifying Preferred Routes

When multiple routing processes are running in the same router for CLNS, it is possible for the same route to be advertised by more than one routing process.

If the router is forwarding packets, dynamic routes will always take priority over static routes, unless the router is routing to a destination outside of its domain and area. The router first will look for an ISO IGRP route within its own area, then for an ISO IGRP route within in its own domain, and finally for an IS-IS route within its own area, until it finds a matching route. If a matching route still has not been found, the router will check its prefix table, which contains static routes and routes to destinations outside the area (ISO IGRP), domain (ISO IGRP), and area (IS-IS) routes for that router. When the router is using its prefix table it will choose the route that has the lowest administrative distance.

By default, the following administrative distances are assigned:

- Static routes—10

- ISO IGRP routes—100
- IS-IS routes—110

When you change an administrative distance for a routing process, use the following command in router configuration mode:

Command	Purpose
`Router(config-router)# distance value [clns]`	Specifies preferred routes by setting the lowest administrative distance.

Note The administrative distance for CLNS routes that you have configured by entering the **distance** command will take effect only when routes are entered into the routing prefix table.

If you want an ISO IGRP prefix route to override a static route, you must set the administrative distance for the routing process to be lower than 10 (assigned administrative distance for static routes). You cannot change the assigned administrative distance for static routes.

Configuring ES-IS Hello Packet Parameters

You can configure ES-IS parameters for communication between end systems and routers. In general, you should leave these parameters at their default values.

When configuring an ES-IS router, be aware of the following:

- ES-IS does not run over X.25 links unless the broadcast facility is enabled.
- ES hello packets and IS hello packets are sent without options. Options in received packets are ignored.

ISs and ESs periodically send out hello packets to advertise their availability. The frequency of these hello packets can be configured.

The recipient of a hello packet creates an adjacency entry for the system that sent it. If the next hello packet is not received within the interval specified, the adjacency times out and the adjacent node is considered unreachable.

A default rate has been set for hello packets and packet validity; however, to change the defaults, use the following commands in global configuration mode:

Command	Purpose
`Router(config)# clns configuration-time seconds`	Specifies the rate at which ES hello and IS hello packets are sent.
`Router(config)# clns holding-time seconds`	Allows the sender of an ES hello or IS hello packet to specify the length of time you consider the information in these packets to be valid.

A default rate has been set for the ES Configuration Timer (ESCT) option; however, to change the default, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# clns esct-time seconds`	Specifies how often the end system should send ES hello packet PDUs.

Configuring DECnet OSI or Phase V Cluster Aliases

DECnet Phase V *cluster aliasing* allows multiple systems to advertise the same system ID in end-system hello packets. Cisco IOS software accomplishes cluster aliasing by caching multiple ES adjacencies with the same NSAP address, but different SNPA addresses. When a packet is destined for the common NSAP address, the software splits the packet loads among the different SNPA addresses. A router that supports this capability forwards traffic to each system. You can enable this capability on a per-interface basis.

To configure cluster aliases, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# clns cluster-alias`	Allows multiple systems to advertise the same system ID in end-system hello packets.

If DECnet Phase V cluster aliases are disabled on an interface, ES hello packet information is used to replace any existing adjacency information for the NSAP address. Otherwise, an additional adjacency (with a different SNPA) is created for the same NSAP address.

For an example of configuring DECnet OSI cluster aliases, see the "DECnet Cluster Aliases Example" section at the end of this chapter.

Configuring Digital-Compatible Mode

If you have an old DECnet implementation of ES-IS in which the NSAP address advertised in an IS hello packet does not have the N-selector byte present, you may want to configure Cisco IOS software to allow IS hello packets sent and received to ignore the N-selector byte. The N-selector byte is the last byte of the NSAP address.

To enable Digital-compatible mode, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# clns dec-compatible`	Allows IS hello packets sent and received to ignore the N-selector byte.

Allowing Security Option Packets to Pass

By default, Cisco IOS software discards any packets with security options set. You can disable this behavior. To allow such packets to pass through, use the following command in global configuration mode:

Command	Purpose
`Router(config)# clns security pass-through`	Allows the software to accept any packets it sees as set with security options.

Note The ISO CLNS routing software ignores the Record Route option, the Source Route option, and the QoS option other than when congestion experienced. The Security option causes a packet to be rejected with a bad option indication.

Configuring CLNS over WANs

This section provides general information about running ISO CLNS over WANs.

You can use CLNS routing on serial interfaces with High-Level Data Link Control (HDLC), PPP, Link Access Procedure, Balanced (LAPB), X.25, Frame Relay, dial-on-demand routing (DDR), or SMDS encapsulation. Both incoming and outgoing CLNS packets can be fast switched over PPP.

To use HDLC encapsulation, you must have a router at both ends of the link. If you use X.25 encapsulation, and if IS-IS or ISO IGRP is not used on an interface, you must manually enter the NSAP-to-X.121 address mapping. The LAPB, SMDS, Frame Relay, and X.25 encapsulations interoperate with other vendors.

Both ISO IGRP and IS-IS can be configured over WANs.

X.25 is not a broadcast medium and therefore does not broadcast protocols (such as ES-IS) that automatically advertise and record mappings between NSAP/NET (protocol addresses) and SNPA (media addresses). (With X.25, the SNPAs are the X.25 network addresses, or the X.121 addresses. These addresses are usually assigned by the X.25 network provider.) If you use static routing, you must configure the NSAP-to-X.121 address mapping with the **x25 map** command.

Configuring a serial line to use CLNS over X.25 requires configuring the general X.25 information and the CLNS-specific information. First, configure the general X.25 information. Then, enter the CLNS static mapping information.

You can specify X.25 nondefault packet and window sizes, reverse charge information, and so on. The X.25 facilities information that can be specified is exactly the same as in the **x25 map** interface configuration command described in the "Configuring X.25 and LAPB" chapter in the *Cisco IOS Wide-Area Networking Configuration Guide*.

See the "ISO CLNS over X.25 Example" section at the end of this chapter for an example of configuring CLNS over X.25.

Enhancing ISO CLNS Performance

Generally, you need not change the default settings of the router for CLNS packet switching, but there are some modifications you can make when you decide to make changes in the performance of your network. The following sections describe ISO CLNS parameters that you can change:

- Specifying the MTU Size (Optional)
- Disabling Checksums (Optional)
- Disabling Fast Switching Through the Cache (Optional)
- Setting the Congestion Threshold (Optional)
- Sending Error Protocol Data Units (Optional)
- Controlling Redirect Protocol Data Units (Optional)
- Configuring Parameters for Locally Sourced Packets (Optional)

See the “Performance Parameters Example” section at the end of this chapter for examples of configuring various performance parameters.

Specifying the MTU Size

All interfaces have a default maximum packet size. However, to reduce fragmentation, you can set the MTU size of the packets sent on the interface. The minimum value is 512; the default and maximum packet size depends on the interface type.

Changing the MTU value with the **mtu** interface configuration command can affect the CLNS MTU value. If the CLNS MTU is at its maximum given the interface MTU, the CLNS MTU will change with the interface MTU. However, the reverse is not true; changing the CLNS MTU value has no effect on the value for the **mtu** interface configuration command.

To set the CLNS MTU packet size for a specified interface, use the following command in interface configuration mode:

Command	Purpose
*bytes*Router(config-if)# **clns mtu**	Sets the MTU size of the packets sent on the interface.

Note The CTR card does not support the switching of frames larger than 4472 bytes. Interoperability problems might occur if CTR cards are intermixed with other Token Ring cards on the same network. These problems can be minimized by lowering the CLNS MTU sizes to be the same on all routers on the network.

Disabling Checksums

When the ISO CLNS routing software originates a CLNS packet, by default it generates checksums. To disable this function, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# **no clns checksum**	Disables checksum generation.

Note Enabling checksum generation has no effect on routing packets (ES-IS, ISO IGRP, and IS-IS) originated by the router; it applies to pings and traceroute packets.

Disabling Fast Switching Through the Cache

Fast switching through the cache is enabled by default for all supported interfaces. To disable fast switching, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# **no clns route-cache**	Disables fast switching.

Note The cache still exists and is used after the **no clns route-cache** interface configuration command is used; the software does not support fast switching through the cache.

Setting the Congestion Threshold

If a router that is configured for CLNS experiences congestion, it sets the congestion-experienced bit. You can set the congestion threshold on a per-interface basis. By setting this threshold, you cause the system to set the congestion-experienced bit if the output queue has more than the specified number of packets in it.

To set the congestion threshold, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# clns congestion-threshold number`	Sets the congestion threshold.

Sending Error Protocol Data Units

When a CLNS packet is received, the routing software looks in the routing table for the next hop. If it does not find one, the packet is discarded and an error protocol data unit (ERPDU) is sent.

You can set an interval time between ERPDUs. Setting a minimum interval between ERDPUs can reduce the amount of bandwidth used by ERDPUs. To set a minimum interval between ERPDUs, you configure the **clns erpdu-interval** command on a specified interface and use the *milliseconds* argument to set the minimum time interval between ERPDUs. Cisco IOS software does not send ERPDUs more frequently than one per “x” milliseconds on the specified interface, where “x” is the number you entered for the *milliseconds* argument.

To send ERPDUs and set the minimum interval time between ERPDUs, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# clns send-erpdu`	Sends an ERPDU when the routing software detects an error in a data PDU; this is enabled by default.
Step 2	`Router(config-if)# clns erpdu-interval milliseconds`	Sets the minimum interval time, in milliseconds, between ERPDUs.

Controlling Redirect Protocol Data Units

If a packet is sent out the same interface it came in on, a redirect protocol data unit (RDPDU) can also be sent to the sender of the packet. You can control RDPDUs in the following ways:

- By default, CLNS sends RDPDUs when a better route for a given host is known. You can disable this feature. Disabling this feature reduces bandwidth because packets may continue to unnecessarily go through the router.
- You can set the interval times between RDPDUs.

Note SNPA masks are never sent, and RDPDUs are ignored by Cisco IOS software when the router is acting as an IS.

To control RDPDUs, use either of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)#` **`clns send-rdpdu`**	Sends redirect PDUs when a better route for a given host is known.
`Router(config-if)#` **`clns rdpdu-interval`** *`milliseconds`*	Sets the minimum interval time, in milliseconds, between RDPDUs.

Configuring Parameters for Locally Sourced Packets

To configure parameters for packets originated by a specified router, use either of the following commands in global configuration mode:

Command	Purpose
`Router(config)#` **`clns packet-lifetime`** *`seconds`*	Specifies in seconds the initial lifetime for locally generated packets.
`Router(config)#` **`clns want-erpdu`**	Specifies whether to request ERPDUs on packets originated by the router.

You should set the packet lifetime low in an internetwork that has frequent loops.

Note The **clns want-erpdu** global configuration command has no effect on routing packets (ES-IS, ISO IGRP, and IS-IS) originated by the router; it applies to pings and traceroute packets.

Monitoring and Maintaining the ISO CLNS Network

To monitor and maintain the ISO CLNS caches, tables, and databases, use the following commands in EXEC mode:

Command	Purpose
`Router>` **`clear clns cache`**	Clears and reinitializes the CLNS routing cache.
`Router>` **`clear clns es-neighbors`**	Removes ES neighbor information from the adjacency database.
`Router>` **`clear clns is-neighbors`**	Removes IS neighbor information from the adjacency database.
`Router>` **`clear clns neighbors`**	Removes CLNS neighbor information from the adjacency database.
`Router>` **`clear clns route`**	Removes dynamically derived CLNS routing information.
`Router>` **`ping clns`** `{`*`host`* `|` *`address`*`}`	Invokes a diagnostic tool for testing connectivity.
`Router>` **`show clns`**	Displays information about the CLNS network.
`Router>` **`show clns cache`**	Displays the entries in the CLNS routing cache.

Command	Purpose
Router> **show clns** [*area-tag*] **es-neighbors** [*type number*] [**detail**]	Displays ES neighbor entries, including the associated areas.
Router> **show clns filter-expr** [*name*] [**detail**]	Displays filter expressions.
Router> **show clns filter-set** [*name*]	Displays filter sets.
Router> **show clns interface** [*type number*]	Lists the CLNS-specific or ES-IS information about each interface.
Router> **show clns** [*area-tag*] **is-neighbors** [*type number*] [**detail**]	Displays IS neighbor entries, according to the area in which they are located.
Router> **show clns** [*area-tag*] **neighbors** [*type number*] [**detail**]	Displays both ES and IS neighbors.
Router> **show clns** [*area-tag*] **neighbor areas**	Displays information about IS-IS neighbors and the areas to which they belong.
Router> **show clns** [*area-tag*] **protocol** [*domain* | *area-tag*]	Lists the protocol-specific information for each IS-IS or ISO IGRP routing process in this router.
Router> **show clns route** [*nsap*]	Displays all the destinations to which this router knows how to route CLNS packets.
Router> **show clns** [*area-tag*] **traffic**	Displays information about the CLNS packets this router has seen.
Router> **show isis** [*area-tag*] **database** [**level-1**] [**level-2**] [**l1**] [**l2**] [**detail**] [*lspid*]	Displays the IS-IS link-state database.
Router> **show isis** [*area-tag*] **route**	Displays the IS-IS Level 1 routing table.
Router> **show isis** [*area-tag*] **spf-log**	Displays a history of the shortest path first (SPF) calculations for IS-IS.
Router> **show isis** [*area-tag*] **topology**	Displays a list of all connected routers in all areas.
Router> **show route-map** [*map-name*]	Displays all route maps configured or only the one specified.
Router> **trace clns** *destination*	Discovers the paths taken to a specified destination by packets in the network.
Router> **which-route** {*nsap-address* | *clns-name*}	Displays the routing table in which the specified CLNS destination is found.

Configuring TARP on ISO CLNS

Some applications (typically used by telephone companies) running on SONET devices identify these devices by a target identifier (TID). Therefore, it is necessary for the router to cache TID-to-network address mappings. Because these applications usually run over OSI, the network addresses involved in the mapping are OSI NSAPs.

When a device must send a packet to another device it does not know about (that is, it does not have information about the NSAP address corresponding to the TID of the remote device), the device needs a way to request this information directly from the device, or from an intermediate device in the network. This functionality is provided by an address resolution protocol called TID Address Resolution Protocol (TARP).

Requests for information and associated responses are sent as TARP PDUs, which are sent as Connectionless Network Protocol (CLNP) data packets. TARP PDUs are distinguished by a unique N-selector in the NSAP address. Following are the five types of TARP PDUs:

- Type 1—Sent when a device has a TID for which it has no matching NSAP. Type 1 PDUs are sent to all Level 1 (IS-IS and ES-IS) neighbors. If no response is received within the specified time limit, a Type 2 PDU is sent. To prevent packet looping, a loop detection buffer is maintained on the router. A Type 1 PDU is sent when you use the **tarp resolve** command.
- Type 2—Sent when a device has a TID for which it has no matching NSAP and no response was received from a Type 1 PDU. Type 2 PDUs are sent to all Level 1 and Level 2 neighbors. A time limit for Type 2 PDUs can also be specified. A Type 2 PDU is sent when you use the **tarp resolve** command and specify the option 2.
- Type 3—Sent as a response to a Type 1, Type 2, or Type 5 PDU. Type 3 PDUs are sent directly to the originator of the request.
- Type 4—Sent as a notification when a change occurs locally (for example, a TID or NSAP change). A Type 4 PDU usually occurs when a device is powered up or brought online.
- Type 5—Sent when a device needs a TID that corresponds to a specific NSAP. Unlike Type 1 and Type 2 PDUs that are sent to all Level 1 and Level 2 neighbors, a Type 5 PDU is sent only to a particular router. In addition to the type, TARP PDUs contain the sender NSAP, the sender TID, and the target TID (if the PDU is a Type 1 or Type 2). A Type 5 PDU is sent when you use the **tarp query** command.

TARP can be used for a conventional IS-IS configuration with a single Level 1 and a Level 2 area (or configuration with a single Level 1 area *or* a Level 2 area).

If multiple Level 1 areas are defined, the router resolves addresses using TARP in the following way:

1. The router obtains the NSAP of the Level 2 area, if present, from the locally assigned target identifier.
2. If only Level 1 areas are configured, the router uses the NSAP of the first active Level 1 area as shown in the configuration at the time of TARP configuration (“tarp run”). (Level 1 areas are sorted alphanumerically by tag name, with capital letters coming before lowercase letters. For example, AREA-1 precedes AREA-2, which precedes area-1.) Note that the TID NSAP could change following a reload if a new Level 1 area is added to the configuration after TARP is running.
3. The router continues to process all Type 1 and 2 PDUs that are for this router. Type 1 PDUs are processed locally if the target identifier is in the local TID cache. If not, they are “propagated” (routed) to all interfaces in the *same* Level 1 area. (The same area is defined as the area configured on the input interface.)
4. Type 2 PDUs are processed locally if the specified target identifier is in the local TID cache. If not, they are propagated via all interfaces (all Level 1 or Level 2 areas) with TARP enabled. If the source of the PDU is from a different area, the information is also added to the local TID cache. Type 2 PDUs are propagated via all static adjacencies.
5. Type 4 PDUs (for changes originated locally) are propagated to all Level 1 and Level 2 areas (because internally they are treated as “Level 1-2”).
6. Type 3 and 5 PDUs continue to be routed.
7. Type 1 PDUs are only “propagated” (routed) via Level 1 static adjacencies if the static NSAP is in one of the Level 1 areas in this router.

TARP Configuration Task List

To configure TARP on the router, perform the tasks in the following sections. Only the first task is required; all other tasks are optional.

- Enabling TARP and Configuring a TARP TID (Required)

- Disabling TARP Caching (Optional)
- Disabling TARP PDU Origination and Propagation (Optional)
- Configuring Multiple NSAP Addresses (Optional)
- Configuring Static TARP Adjacency and Blacklist Adjacency (Optional)
- Determining TIDs and NSAPs (Optional)
- Configuring TARP Timers (Optional)
- Configuring Miscellaneous TARP PDU Information (Optional)
- Monitoring and Maintaining the TARP Protocol (Optional)
- Determining the Tunnel Type, page 44 (Optional)
- Configuring GRE/CLNS CTunnels to Carry IPv4 and IPv6 Packets, page 45 (Optional)
- Verifying Tunnel Configuration and Operation, page 48 (Optional)

For several examples of configuring TARP, see the "TARP Configuration Examples" section at the end of this chapter.

Enabling TARP and Configuring a TARP TID

TARP must be explicitly enabled before the TARP functionality becomes available, and the router must have a TID assigned. Also, before TARP packets can be sent out on an interface, each interface must have TARP enabled and the interface must be able to propagate TARP PDUs.

The router will use the CLNS capability to send and receive TARP PDUs. If the router is configured as an IS, the router must be running IS-IS. If the router is configured as an ES, the router must be running ES-IS.

To turn on the TARP functionality, use the following commands in global configuration mode:

Command	Purpose
`Router(config)#` **`tarp run`**	Turns on the TARP functionality.
`Router(config)#` **`tarp tid`** *`tid`*	Assigns a TID to the router.

To enable TARP on one or more interfaces, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)#` **`tarp enable`**	Enables TARP on the interface.

Disabling TARP Caching

By default, TID-to-NSAP address mappings are stored in the TID cache. Disabling this capability clears the TID cache. Reenabling this capability restores any previously cleared local entry and all static entries.

To disable TID-to-NSAP address mapping in the TID cache, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no tarp allow-caching`	Disables TARP TID-to-NSAP address mapping.

Disabling TARP PDU Origination and Propagation

By default, the router originates TARP PDUs and propagates TARP PDUs to its neighbors, and the interface propagates TARP PDUs to its neighbor. Disabling these capabilities means that the router no longer originates TARP PDUs, and the router and the specific interface no longer propagate TARP PDUs received from other routers.

To disable origination and propagation of TARP PDUs, use the following commands in global configuration mode:

Command	Purpose
`Router(config)# no tarp originate`	Disables TARP PDU origination.
`Router(config)# no tarp global-propagate`	Disables global propagation of TARP PDUs.

To disable propagation of TARP PDUs on a specific interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# no tarp propagate [all | message-type {unknowns | type-number} [type-number] [type-number]]`	Disables propagation of TARP PDUs on the interface.

Configuring Multiple NSAP Addresses

A router may have more than one NSAP address. When a request for an NSAP is sent (Type 1 or Type 2 PDU), the first NSAP address is returned. To receive all NSAP addresses associated with the router, enter a TID-to-NSAP static route in the TID cache for each NSAP address.

To create a TID-to-NSAP static route, use the following command in global configuration mode:

Command	Purpose
`Router(config)# tarp map tid nsap`	Enters a TID-to-NSAP static route.

Configuring Static TARP Adjacency and Blacklist Adjacency

In addition to all its IS-IS/ES-IS adjacencies, a TARP router propagates PDUs to all its static TARP adjacencies. If a router is not running TARP, the router discards TARP PDUs rather than propagating the PDUs to all its adjacencies. To allow TARP to bypass routers en route that may not have TARP running, TARP provides a static TARP adjacency capability. Static adjacencies are maintained in a special queue.

To create a static TARP adjacency, use the following command in global configuration mode:

Command	Purpose
`Router(config)# tarp route-static nsap [all | message-type {unknowns | type-number} [type-number] [type-number]]`	Enters a static TARP adjacency.

To stop TARP from propagating PDUs to an IS-IS/ES-IS adjacency that may not have TARP running, TARP provides a blacklist adjacency capability. The router will not propagate TARP PDUs to blacklisted routers.

To blacklist a router, use the following command in global configuration mode:

Command	Purpose
`Router(config)# tarp blacklist-adjacency nsap`	Bypasses a router not running TARP.

Determining TIDs and NSAPs

To determine an NSAP address for a TID or a TID for an NSAP address, use the following commands in EXEC mode:

Command	Purpose
`Router> tarp query nsap`	Gets the TID associated with a specific NSAP.
`Router> tarp resolve tid [1 | 2]`	Gets the NSAP associated with a specific TID.

To determine the TID, the router first checks the local TID cache. If there is a TID entry in the local TID cache, the requested information is displayed. If there is no TID entry in the local TID cache, a TARP Type 5 PDU is sent out to the specified NSAP address.

To determine the NSAP address, the router first checks the local TID cache. If there is an NSAP entry in the local TID cache, the requested information is displayed. If there is no NSAP entry in the local TID cache, a TARP Type 1 or Type 2 PDU is sent out. By default, a Type 1 PDU is sent to all Level 1 (IS-IS and ES-IS) neighbors. If a response is received, the requested information is displayed. If a response is not received within the response time, a Type 2 PDU is sent to all Level 1 and Level 2 neighbors. Specifying the **tarp resolve** *tid* **2** EXEC command causes only a Type 2 PDU to be sent.

You can configure the length of time that the router will wait for a response (in the form of a Type 3 PDU).

Configuring TARP Timers

TARP timers provide default values and typically need not be changed.

You can configure the amount of time that the router waits to receive a response from a Type 1 PDU, a Type 2 PDU, and a Type 5 PDU. You can also configure the lifetime of the PDU based on the number of hops.

You can also set timers that control how long dynamically created TARP entries remain in the TID cache, and how long the system ID-to-sequence number mapping entry remains in the loop detection buffer table. The loop detection buffer table prevents TARP PDUs from looping.

To configure TARP PDU timers, control PDU lifetime, and set how long entries remain in cache, use the following commands in global configuration mode:

Command	Purpose
`Router(config)# tarp t1-response-timer seconds`	Configures the number of seconds that the router will wait for a response from a TARP Type 1 PDU.
`Router(config)# tarp t2-response-timer seconds`	Configures the number of seconds that the router will wait for a response from a TARP Type 2 PDU.
`Router(config)# tarp post-t2-response-timer seconds`	Configures the number of seconds that the router will wait for a response from a TARP Type 2 PDU after the default timer has expired.
`Router(config)# tarp arp-request-timer seconds`	Configures the number of seconds that the router will wait for a response from a TARP Type 5 PDU.
`Router(config)# tarp lifetime hops`	Configures the number of routers that a TARP PDU can traverse before it is discarded.
`Router(config)# tarp cache-timer seconds`	Configures the number of seconds a dynamically created TARP entry remains in the TID cache.
`Router(config)# tarp ldb-timer seconds`	Configures the number of seconds that a system ID-to-sequence number mapping entry remains in the loop detection buffer table.

Configuring Miscellaneous TARP PDU Information

TARP default PDU values typically need not be changed.

You can configure the sequence number of the TARP PDU, set the update remote cache bit used to control whether the remote router updates its cache, specify the N-selector used in the PDU to indicate a TARP PDU, and specify the network protocol type used in outgoing PDUs.

To configure miscellaneous PDU information, use the following commands in global configuration mode:

Command	Purpose
`Router(config)# tarp sequence-number number`	Changes the sequence number in the next outgoing TARP PDU.
`Router(config)# tarp urc [0 | 1]`	Sets the update remote cache bit in all subsequent outgoing TARP PDUs so that the remote router does or does not update the cache.
`Router(config)# tarp nselector-type hex-digit`	Specifies the N-selector used to identify TARP PDUs.
`Router(config)# tarp protocol-type hex-digit`	Specifies the protocol type used in outgoing TARP PDUs. Only the hexadecimal value 0xFE (to indicate the CLNP) is supported.[1]

[1] CLNP = Connectionless Network Protocol

Monitoring and Maintaining the TARP Protocol

To monitor and maintain the TARP caches, tables, and databases, use the following commands in EXEC mode:

Command	Purpose
`Router> clear tarp counters`	Resets the TARP counters that are shown with the **show tarp traffic** command.
`Router> clear tarp ldb-table`	Removes all system ID-to-sequence number mapping entries in the TARP loop detection buffer table.
`Router> clear tarp tid-table`	Removes all dynamically created TARP TID-to-NSAP address mapping entries in the TID cache.
`Router> show tarp`	Displays all global TARP parameters.
`Router> show tarp blacklisted-adjacencies`	Lists all adjacencies that are blacklisted (that is, adjacencies that will not receive propagated TARP PDUs).
`Router> show tarp host tid`	Displays information about a specific TARP router stored in the local TID cache.
`Router> show tarp interface [type number]`	Lists all interfaces on the router that have TARP enabled.
`Router> show tarp ldb`	Displays the contents of the loop detection buffer table.
`Router> show tarp map`	Lists all the static entries in the TID cache.
`Router> show tarp static-adjacencies`	Lists all static TARP adjacencies.
`Router> show tarp tid-cache`	Displays information about the entries in the TID cache.
`Router> show tarp traffic`	Displays statistics about TARP PDUs.

Routing IP over ISO CLNS Networks

The IP over a CLNS Tunnel feature lets you transport IP traffic over Connectionless Network Service (CLNS); for instance, on the data communications channel (DCC) of a SONET ring.

IP over a CLNS tunnel is a virtual interface that enhances interactions with CLNS networks, allowing IP packets to be tunneled through the Connectionless Network Protocol (CLNP) to preserve TCP/IP services.

Configuring an IP over CLNS tunnel (CTunnel) allows you to Telnet to a remote router that has only CLNS connectivity. Other management facilities can also be used, such as Simple Network Management Protocol (SNMP) and TFTP, which otherwise would not be available over a CLNS network.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release.

Configuring IP over a CLNS Tunnel

To configure IP over a CLNS Tunnel (CTunnel), use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface ctunnel interface-number`	Creates a virtual interface to transport IP over a CLNS tunnel and enters interface configuration mode. The interface number must be unique for each CTunnel interface.
Step 2	`Router(config-if)# ctunnel destination remote-nsap-address`	Configures the destination parameter for the CTunnel. Specifies the destination network service access point (NSAP) address of the CTunnel, where the IP packets are extracted.
Step 3	`Router(config-if)# ip address ip-address mask`	Sets a primary or secondary IP address for an interface.

Note To configure a CTunnel between a single pair of routers, you must enter the foregoing commands on each router. The destination NSAP address for Router A would be the NSAP address of Router B, and the destination NSAP address for Router B would be the NSAP address of Router A. Ideally, the IP addresses used for the virtual interfaces at either end of the tunnel should be in the same IP subnet.

Verifying Configuration

To verify correct configuration of the IP over a CLNS Tunnel feature, perform the following steps:

Step 1 On Router A, ping the IP address of the CTunnel interface of Router B.

Step 2 On Router B, ping the IP address of the CTunnel interface of Router A.

Troubleshooting Tips

If the CTunnel does not function, verify correct configuration on both routers as described in the "Verifying Configuration" section.

Monitoring and Maintaining IP over a CLNS Tunnel

To display the status of IP over CLNS tunnels, use the following command in privileged EXEC mode:

Command	Purpose
`Router# show interfaces ctunnel interface-number`	Displays information about an IP over CLNS tunnel.

Determining the Tunnel Type

Before configuring a tunnel, you must determine what type of tunnel you need to create.

SUMMARY STEPS

1. Determine the passenger protocol.
2. Determine the tunnel CLI type.
3. Determine the **tunnel mode** command keyword, if appropriate.

DETAILED STEPS

Step 1 Determine the passenger protocol.

The passenger protocol is the protocol that you are encapsulating.

Step 2 Determine the tunnel CLI type.

Table 3 shows how to determine the tunnel command-line interface (CLI) command required for the transport protocol that you are using in the tunnel.

Table 3 Determining the Tunnel CLI by the Transport Protocol

Transport Protocol	Tunnel CLI Command
CLNS	**ctunnel** (with optional **mode gre** keywords)
Other	**tunnel mode** (with appropriate keyword)

Step 3 Determine the **tunnel mode** command keyword, if appropriate.

Table 4 shows how to determine the appropriate keyword to use with the **tunnel mode** command. In the tasks that follow in this module, only the relevant keywords for the **tunnel mode** command are displayed.

Table 4 Determining the tunnel mode Command Keyword

Keyword	Purpose
dvmrp	Use the **dvmrp** keyword to specify that the Distance Vector Multicast Routing Protocol encapsulation will be used.
gre ip	Use the **gre ip** keywords to specify that GRE encapsulation over IP will be used.
gre ipv6	Use the **gre ipv6** keywords to specify that GRE encapsulation over IPv6 will be used.
gre multipoint	Use the **gre multipoint** keywords to specify that multipoint GRE (mGRE) encapsulation will be used.
ipip [**decapsulate-any**]	Use the **ipip** keyword to specify that IP-in-IP encapsulation will be used. The optional **decapsulate-any** keyword terminates any number of IP-in-IP tunnels at one tunnel interface. Note that this tunnel will not carry any outbound traffic; however, any number of remote tunnel endpoints can use a tunnel configured this way as their destination.
ipv6	Use the **ipv6** keyword to specify that generic packet tunneling in IPv6 will be used.
ipv6ip	Use the **ipv6ip** keyword to specify that IPv6 will be used as the passenger protocol and IPv4 as both the carrier (encapsulation) and transport protocol. When additional keywords are not used, manual IPv6 tunnels are configured. Additional keywords can be used to specify IPv4-compatible, 6to4, or ISATAP tunnels.
mpls	Use the **mpls** keyword to specify that MPLS will be used for configuring Traffic Engineering (TE) tunnels.
rbscp	Use the **rbscp** keyword to specify that RBSCP tunnels will be used.

Configuring GRE/CLNS CTunnels to Carry IPv4 and IPv6 Packets

Perform this task to configure a CTunnel in GRE mode to transport IPv4 and IPv6 packets in a CLNS network.

To configure a CTunnel between a single pair of routers, a tunnel interface must be configured with an IP address, and a tunnel destination must be defined. The destination network service access point (NSAP) address for Router A would be the NSAP address of Router B, and the destination NSAP address for Router B would be the NSAP address of Router A. Ideally, the IP addresses used for the virtual interfaces at either end of the tunnel should be in the same IP subnet. Remember to configure the router at each end of the tunnel.

Tunnels for IPv4 and IPv6 Packets over CLNS Networks

Configuring the **ctunnel mode gre** command on a CTunnel interface enables IPv4 and IPv6 packets to be tunneled over CLNS in accordance with RFC 3147. Compliance with this RFC should allow interoperation between Cisco equipment and that of other vendors in which the same standard is implemented.

RFC 3147 specifies the use of GRE for tunneling packets. The implementation of this feature does not include support for GRE services defined in header fields, such as those used to specify checksums, keys, or sequencing. Any packets received that specify the use of these features will be dropped.

The default CTunnel mode continues to use the standard Cisco encapsulation, which will tunnel only IPv4 packets. If you want to tunnel IPv6 packets, you must use the GRE encapsulation mode. Both ends of the tunnel must be configured with the same mode for either method to work.

Prerequisites

- An IPv4 or IPv6 address must be configured on a CTunnel interface, and manually configured CLNS addresses must be assigned to the CTunnel destination.
- The host or router at each end of a configured CTunnel must support both the IPv4 and IPv6 protocol stacks.
- The CTunnel source and destination must both be configured to run in the same mode.

Restrictions

GRE services, such as those used to specify checksums, keys, or sequencing, are not supported. Packets that request use of those features will be dropped.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface ctunnel** *interface-number*
4. **ip address** *ip-address mask*

   or

   ipv6 address *ipv6-prefix***/***prefix-length* [**eui-64**]
5. **ctunnel destination** *remote-nsap-address*
6. **ctunnel mode gre**
7. **end**
8. **show interfaces ctunnel** *interface-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	**interface ctunnel** *interface-number* **Example:** `Router(config)# interface ctunnel 102`	Creates a virtual interface to transport IP over a CLNS tunnel and enters interface configuration mode. **Note** The interface number must be unique for each CTunnel interface.
Step 4	**ip address** *ip-address mask* or **ipv6 address** *ipv6-prefix***/***prefix-length* [**eui-64**] **Example:** `Router(config-if)# ipv6 address 2001:0DB8:1234:5678::3/126`	Specifies the IPv4 or IPv6 network assigned to the interface and enables IPv4 or IPv6 packet processing on the interface. **Note** See the "Implementing Basic Connectivity for IPv6" module for more information on configuring IPv6 addresses.
Step 5	**ctunnel destination** *remote-nsap-address* **Example:** `Router(config-if)# ctunnel destination 192.168.30.1`	Specifies the destination NSAP address of the CTunnel, where the packets are extracted. • Use the *remote-nsap-address* argument to specify the NSAP address at the CTunnel endpoint.
Step 6	**ctunnel mode gre** **Example:** `Router(config-if)# ctunnel mode gre`	Specifies a CTunnel running in GRE mode for both IPv4 and IPv6 traffic. **Note** The **ctunnel mode gre** command specifies GRE as the encapsulation protocol for the tunnel.

Step 7	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Step 8	**show interfaces ctunnel** *interface-number* **Example:** `Router# show interfaces ctunnel 102`	(Optional) Displays information about an IP over CLNS tunnel. • Use the *interface-number* argument to specify a CTunnel interface. • Use this command to verify the CTunnel configuration.

Verifying Tunnel Configuration and Operation

This optional task explains how to verify tunnel configuration and operation. The commands contained in the task steps can be used in any sequence and may need to be repeated. The following commands can be used for GRE tunnels, IPv6 manually configured tunnels, and IPv6 over IPv4 GRE tunnels. Ths process includes the following general steps (details follow):

Step 1 On Router A, ping the IP address of the CTunnel interface of Router B.

Step 2 Step 2 On Router B, ping the IP address of the CTunnel interface of Router A.

SUMMARY STEPS

1. **enable**
2. **show interfaces tunnel** *number* [**accounting**]
3. **ping** [*protocol*] *destination*
4. **show ip route** [*address* [*mask*]]
5. **ping** [*protocol*] *destination*

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

```
Router> enable
```

Step 2 **show interfaces tunnel** *number* [**accounting**]

Assuming a generic example suitable for both IPv6 manually configured tunnels and IPv6 over IPv4 GRE tunnels, two routers are configured to be endpoints of a tunnel. Router A has Ethernet interface 0/0 configured as the source for tunnel interface 0 with an IPv4 address of 10.0.0.1 and an IPv6 prefix of 2001:0DB8:1111:2222::1/64. Router B has Ethernet interface 0/0 configured as the source for tunnel interface 1 with an IPv4 address of 10.0.0.2 and an IPv6 prefix of 2001:0DB8:1111:2222::2/64.

To verify that the tunnel source and destination addresses are configured, use the **show interfaces tunnel** command on Router A.

```
RouterA# show interfaces tunnel 0

Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.0.0.1 (Ethernet0/0), destination 10.0.0.2, fastswitch TTL 255
  Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
  Tunnel TTL 255
  Checksumming of packets disabled,  fast tunneling enabled
  Last input 00:00:14, output 00:00:04, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
```

```
     4 packets input, 352 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     8 packets output, 704 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Step 3 **ping** [*protocol*] *destination*

To check that the local endpoint is configured and working, use the **ping** command on Router A.

```
RouterA# ping 2001:0DB8:1111:2222::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:0DB8:1111:2222::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/20 ms
```

Step 4 **show ip route** [*address* [*mask*]]

To check that a route exists to the remote endpoint address, use the **show ip route** command.

```
RouterA# show ip route 10.0.0.2

Routing entry for 10.0.0.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
  * directly connected, via Ethernet0/0
      Route metric is 0, traffic share count is 1
```

Step 5 **ping** [*protocol*] *destination*

To check that the remote endpoint address is reachable, use the **ping** command on Router A.

Note The remote endpoint address may not be reachable using the **ping** command because of filtering, but the tunnel traffic may still reach its destination.

```
RouterA# ping 10.0.0.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/21/28 ms
```

To check that the remote IPv6 tunnel endpoint is reachable, use the **ping** command again on Router A. The same note on filtering also applies to this example.

```
RouterA# ping 1::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/20 ms
```

These steps may be repeated at the other endpoint of the tunnel.

For more information on configuring interfaces, refer to the *Cisco IOS Interface Configuration Guide.*

ISO CLNS Configuration Examples

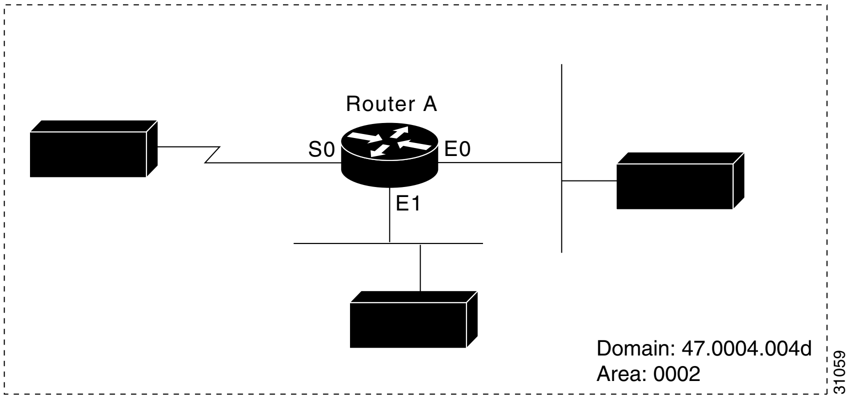
The following sections provide configuration examples of both intra- and interdomain static and dynamic routing using static, ISO IGRP, and IS-IS routing techniques:

- Dynamic Routing Within the Same Area Example
- Dynamic Routing in More Than One Area Example
- Dynamic Routing in Overlapping Areas Example
- Dynamic Interdomain Routing Example
- IS-IS Routing Configuration Examples
- NETs Configuration Examples
- Router in Two Areas Example
- Basic Static Routing Examples
- Static Intradomain Routing Example
- Static Interdomain Routing Example
- CLNS Filter Examples
- Route Map Examples
- DECnet Cluster Aliases Example
- ISO CLNS over X.25 Example
- Performance Parameters Example
- TARP Configuration Examples
- IP over a CLNS Tunnel Example
- Configuring GRE/CLNS CTunnels to Carry IPv4 and IPv6 Packets Example

Dynamic Routing Within the Same Area Example

Figure 5 and the following example show how to configure dynamic routing within a routing domain. The router can exist in one or more areas within the domain. The router named Router A exists in a single area.

Figure 5 CLNS Dynamic Routing Within a Single Area

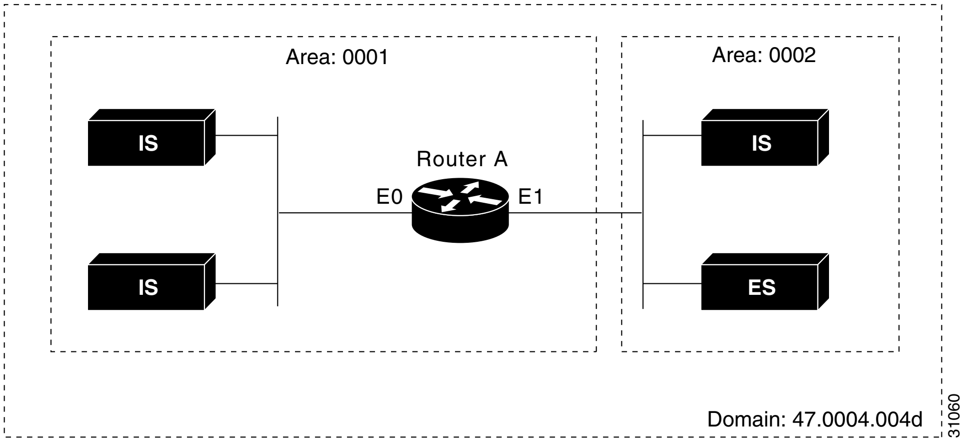


```
! Define a tag castor for the routing process
router iso-igrp castor
! configure the net for the process in area 2, domain 47.0004.004d
 Net 47.0004.004d.0002.0000.0C00.0506.00
! Specify iso-igrp routing using the previously specified tag castor
interface ethernet 0
 clns router iso-igrp castor
 ! Specify iso-igrp routing using the previously specified tag castor
interface ethernet 1
 clns router iso-igrp castor
! Specify iso-igrp routing using the previously specified tag castor
interface serial 0
 clns router iso-igrp castor
```

Dynamic Routing in More Than One Area Example

Figure 6 and the following example show how to configure a router named Router A that exists in two areas.

Figure 6 CLNS Dynamic Routing Within Two Areas



```
! Define a tag orion for the routing process
router iso-igrp orion
! Configure the net for the process in area 1, domain 47.0004.004d
```

```
 net 47.0004.004d.0001.212223242526.00
! Specify iso-igrp routing using the previously specified tag orion
interface ethernet 0
 clns router iso-igrp orion
! Specify iso-igrp routing using the previously specified tag orion
interface ethernet 1
 clns router iso-igrp orion
```

Dynamic Routing in Overlapping Areas Example

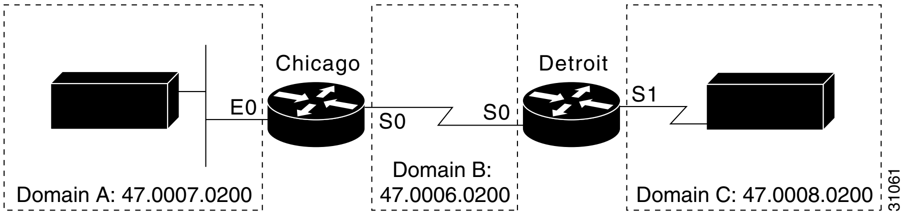
The following example shows how to configure a router with overlapping areas:

```
! Define a tag capricorn for the routing process
router iso-igrp capricorn
! Configure the NET for the process in area 3, domain 47.0004.004d
 net 47.0004.004d.0003.0000.0C00.0508.00
! Define a tag cancer for the routing process
router iso-igrp cancer
! Configure the NET for the process in area 4, domain 47.0004.004d
 net 47.0004.004d.0004.0000.0C00.0506.00
! Specify iso-igrp routing on interface ethernet 0 using the tag capricorn
interface ethernet 0
 clns router iso-igrp capricorn
! Specify iso-igrp routing on interface ethernet 1 using the tags capricorn and cancer
interface ethernet 1
 clns router iso-igrp capricorn
 clns router iso-igrp cancer
! Specify iso-igrp routing on interface ethernet 2 using the tag cancer
interface ethernet 2
 clns router iso-igrp cancer
```

Dynamic Interdomain Routing Example

Figure 7 and the following example show how to configure three domains that are to be transparently connected.

Figure 7 ***CLNS Dynamic Interdomain Routing***



Router Chicago

The following example shows how to configure router Chicago for dynamic interdomain routing:

```
! Define a tag A for the routing process
router iso-igrp A
! Configure the NET for the process in area 2, domain 47.0007.0200
 net 47.0007.0200.0002.0102.0104.0506.00
! Redistribute iso-igrp routing information throughout domain A
redistribute iso-igrp B
```

```
! Define a tag B for the routing process
router iso-igrp B
! Configure the NET for the process in area 3, domain 47.0006.0200
 net 47.0006.0200.0003.0102.0104.0506.00
! Redistribute iso-igrp routing information throughout domain B
redistribute iso-igrp A
! Specify iso-igrp routing with the tag A
interface ethernet 0
clns router iso-igrp A
! Specify iso-igrp routing with the tag B
interface serial 0
clns router iso-igrp B
```

Router Detroit

The following example shows how to configure router Detroit for dynamic interdomain routing. Comment lines have been eliminated from this example to avoid redundancy.

```
router iso-igrp B
 net 47.0006.0200.0004.0102.0104.0506.00
 redistribute iso-igrp C
router iso-igrp C
 net 47.0008.0200.0005.0102.01040.506.00
 redistribute iso-igrp B
interface serial 0
 clns router iso-igrp B
interface serial 1
 clns router iso-igrp C
```

Chicago injects a prefix route for domain A into domain B. Domain B injects this prefix route and a prefix route for domain B into domain C.

You can also configure a border router between domain A and domain C.

IS-IS Routing Configuration Examples

The following examples show the basic syntax and configuration command sequence for IS-IS routing.

Level 1 and Level 2 Routing

The following example shows how to use the IS-IS protocol to configure a single area address for Level 1 and Level 2 routing:

```
! Route dynamically using the is-is protocol
router isis
! Configure the NET for the process in area 47.0004.004d.0001
 net 47.0004.004d.0001.0000.0c00.1111.00
! Enable is-is routing on ethernet 0
interface ethernet 0
 clns router isis
! Enable is-is routing on ethernet 1
interface ethernet 1
 clns router isis
! Enable is-is routing on serial 0
interface serial 0
 clns router isis
```

Level 2 Routing Only

The following example shows a similar configuration, featuring a single area address being used for specification of Level 1 and Level 2 routing. However, in this case, interface serial interface 0 is configured for Level 2 routing only. Most comment lines have been eliminated from this example to avoid redundancy.

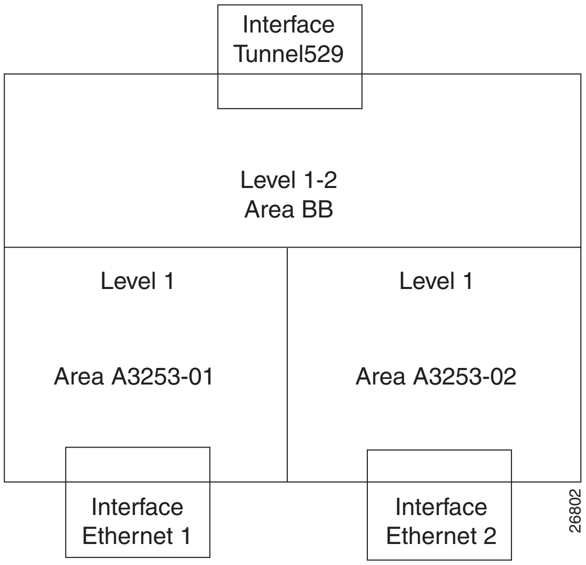
```
router isis
 net 47.0004.004d.0001.0000.0c00.1111.00
interface ethernet 0
 clns router isis
interface ethernet 1
 clns router isis
interface serial 0
 clns router isis
! Configure a level 2 adjacency only for interface serial 0
 isis circuit-type level-2-only
```

Multiarea IS-IS Configuration

The following example shows a mulitarea IS-IS configuration with two Level 1 areas and one Level 1-2 area. Figure 8 illustrates this configuration.

```
clns routing

...

interface Tunnel529
 ip address 10.0.0.5 255.255.255.0
 ip router isis BB
 clns router isis BB

interface Ethernet1
 ip address 10.1.1.5 255.255.255.0
 ip router isis A3253-01
 clns router isis A3253-01
!
interface Ethernet2
 ip address 10.2.2.5 255.255.255.0
 ip router isis A3253-02
 clns router isis A3253-02

...

router isis BB                         ! Defaults to "is-type level-1-2"
 net 49.2222.0000.0000.0005.00
!
router isis A3253-01
 net 49.0553.0001.0000.0000.0005.00
 is-type level-1
!
router isis A3253-02
 net 49.0553.0002.0000.0000.0005.00
 is-type level-1
```

Figure 8 Multiarea IS-IS Configuration with Three Level 1 Areas and One Level 2 Area



OSI Configuration

The following example shows an OSI configuration example. In this example, IS-IS runs with two area addresses, metrics tailored, and different circuit types specified for each interface. Most comment lines have been eliminated from this example to avoid redundancy.

```
! Enable is-is routing in area 1
router isis area1
! Router is in areas 47.0004.004d.0001 and 47.0004.004d.0011
net 47.0004.004d.0001.0000.0c11.1111.00
net 47.0004.004d.0011.0000.0c11.1111.00
! Enable the router to operate as a station router and an interarea router
is-type level-1-2
!
interface ethernet 0
 clns router isis area1
! Specify a cost of 5 for the level-1 routes
 isis metric 5 level-1
! Establish a level-1 adjacency
 isis circuit-type level-1
!
interface ethernet 1
 clns router isis area1
 isis metric 2 level-2
 isis circuit-type level-2-only
!
interface serial 0
 clns router isis area1
 isis circuit-type level-1-2
! Set the priority for serial 0 to 3 for a level-1 adjacency
 isis priority 3 level-1
 isis priority 1 level-2
```

ISO CLNS Dynamic Route Redistribution

The following example shows route redistribution between IS-IS and ISO IGRP domains. In this case, the IS-IS domain is on Ethernet interface 0; the ISO IGRP domain is on serial interface 0. The IS-IS routing process is assigned a null tag; the ISO IGRP routing process is assigned a tag of remote-domain. Most comment lines have been eliminated from this example to avoid redundancy.

```
router isis
 net 39.0001.0001.0000.0c00.1111.00
! Redistribute iso-igrp routing information throughout remote-domain
redistribute iso-igrp remote-domain
!
router iso-igrp remote-domain
 net 39.0002.0001.0000.0c00.1111.00
! Redistribute is-is routing information
redistribute isis
!
interface ethernet 0
 clns router isis
!
interface serial 0
 clns router iso-igrp remote
```

NETs Configuration Examples

The following examples show how to configure NETs for both ISO IGRP and IS-IS.

ISO IGRP

The following example shows how to specify a NET:

```
router iso-igrp Finance
 net 47.0004.004d.0001.0000.0c11.1111.00
```

The following example shows how to use a name for a NET:

```
clns host NAME 39.0001.0000.0c00.1111.00
router iso-igrp Marketing
 net NAME
```

The use of this **net** router configuration command configures the system ID, area address, and domain address. Only a single NET per routing process is allowed.

```
router iso-igrp local
 net 49.0001.0000.0c00.1111.00
```

IS-IS

The following example shows how to specify a single NET:

```
router isis Pieinthesky
 net 47.0004.004d.0001.0000.0c11.1111.00
```

The following example shows how to use a name for a NET:

```
clns host NAME 39.0001.0000.0c00.1111.00
router isis
 net NAME
```

IS-IS Multihoming Example

The following example shows how to assign three separate area addresses for a single router using **net** commands. Traffic received that includes an area address of 47.0004.004d.0001, 47.0004.004d.0002, or 47.0004.004d.0003, and that has the same system ID, is forwarded to this router.

```
router isis eng-area1
!   |       IS-IS Area|      System ID| S|
 net 47.0004.004d.0001.0000.0C00.1111.00
 net 47.0004.004d.0002.0000.0C00.1111.00
 net 47.0004.004d.0003.0000.0C00.1111.00
```

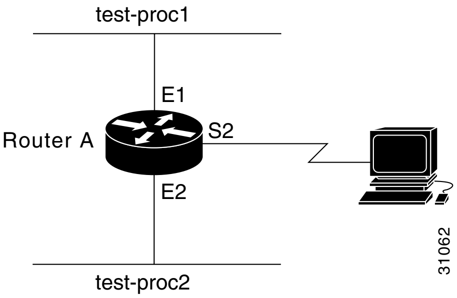
Router in Two Areas Example

The following two examples show how to configure a router in two areas. The first example configures ISO IGRP; the second configures IS-IS.

ISO IGRP

The following example shows the router in domain 49.0001 and having a system ID of aaaa.aaaa.aaaa. The router is in two areas: 31 and 40 (decimal). Figure 9 illustrates this configuration.

Figure 9 ISO IGRP Configuration



```
router iso-igrp test-proc1
! 001F in the following net is the hex value for area 31
 net 49.0001.001F.aaaa.aaaa.aaaa.00
router iso-igrp test-proc2
! 0028 in the following net is the hex value for area 40
 net 49.0001.0028.aaaa.aaaa.aaaa.00
!
interface ethernet 1
 clns router iso-igrp test-proc1
!
interface serial 2
 clns router iso-igrp test-proc1
!
interface ethernet 2
 clns router iso-igrp test-proc2
```

IS-IS

The following example shows how to run IS-IS instead of ISO IGRP. The illustration in Figure 9 still applies. Ethernet interface 2 is configured for IS-IS routing and is assigned the tag of test-proc2.

```
router iso-igrp test-proc1
 net 49.0002.0002.bbbb.bbbb.bbbb.00
```

```
router isis test-proc2
 net 49.0001.0002.aaaa.aaaa.aaaa.00
!
interface ethernet 1
 clns router iso-igrp test-proc1
!
interface serial 2
clns router iso-igrp test-proc1
!
interface ethernet 2
 clns router is-is test-proc2
```

To allow only CLNS packets to pass blindly through an interface without routing updates, use the following configuration:

```
clns routing
interface serial 2
! Permits serial 2 to pass CLNS packets without having CLNS routing turned on
 clns enable
```

Basic Static Routing Examples

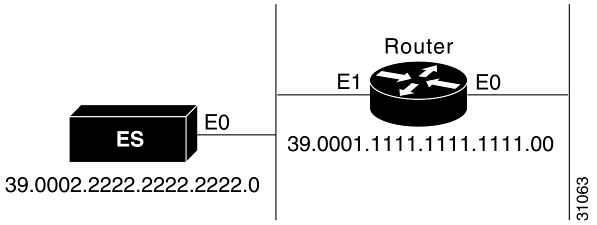
Configuring FDDI, Ethernets, Token Rings, and serial lines for CLNS can be as simple as enabling CLNS on the interfaces. Enabling CLNS on the interfaces is all that is ever required on serial lines using HDLC encapsulation. If all systems on an Ethernet or Token Ring support ISO 9542 ES-IS, then no configuring is required.

The following example shows how to configure an Ethernet and a serial line:

```
! Enable clns packets to be routed
 clns routing
! Configure the following network entity title for the routing process
 clns net 47.0004.004d.0055.0000.0C00.BF3B.00
! Pass ISO CLNS traffic on ethernet 0 to end systems without routing
interface ethernet 0
 clns enable
! Pass ISO CLNS traffic on serial 0 to end systems without routing
interface serial 0
 clns enable
! Create a static route for the interface
 clns route 47.0004.004d.0099 serial 0
 clns route 47.0005 serial 0
```

The following example is a more complete example of CLNS static routing on a system with two Ethernet interfaces. After configuring routing, you define a NET and enable CLNS on the Ethernet 0 and Ethernet 1 interfaces. You must then define an ES neighbor and define a static route with the **clns route** global configuration command, as shown. In this situation, there is an ES on Ethernet 1 that does not support ES-IS. Figure 10 illustrates this network.

Figure 10 Static Routing

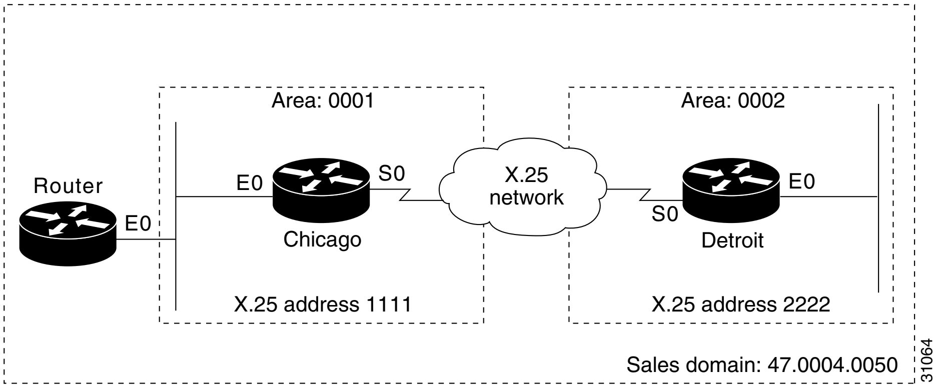


```
 clns host sid 39.0001.1111.1111.1111.00
 clns host bar 39.0002.2222.2222.2222.00
! Assign a static address for the router
 clns net sid
! Enable CLNS packets to be routed
 clns routing
! Pass ISO CLNS packet traffic to end systems without routing them
interface ethernet 0
 clns enable
! Pass ISO CLNS packet traffic to end systems without routing them
interface ethernet 1
 clns enable
! Specify end system for static routing
 clns es-neighbor bar 0000.0C00.62e7
! Create an interface-static route to bar for packets with the following NSAP address
 clns route 47.0004.000c bar
```

Static Intradomain Routing Example

Figure 11 and the following example show how to use static routing inside of a domain. Imagine a company with branch offices in Detroit and Chicago, connected with an X.25 link. These offices are both in the domain named Sales.

Figure 11 CLNS X.25 Intradomain Routing



The following example shows one way to configure the router Chicago:

```
! Define the name chicago to be used in place of the following NSAP
clns host chicago 47.0004.0050.0001.0000.0c00.243b.00
! Define the name detroit to be used in place of the following NSAP
clns host detroit 47.0004.0050.0002.0000.0c00.1e12.00
! Enable ISO IGRP routing of CLNS packets
router iso-igrp sales
! Configure net chicago, as defined above
 net chicago
! Specify iso-igrp routing using the previously specified tag sales
interface ethernet 0
 clns router iso-igrp sales
! Set the interface up as a DTE with X.25 encapsulation
interface serial 0
 encapsulation x25
 x25 address 1111
 x25 nvc 4
! Specify iso-igrp routing using the previously specified tag sales
 clns router iso-igrp sales
```

```
! Define a static mapping between Detroit's nsap and its X.121 address
 x25 map clns 2222 broadcast
```

This configuration brings up an X.25 virtual circuit between the router Chicago and the router Detroit. Routing updates will be sent across this link. This implies that the virtual circuit could be up continuously.

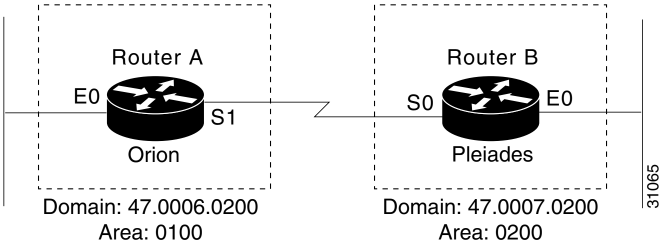
If the Chicago office should grow to contain multiple routers, it would be appropriate for each of those routers to know how to get to router Detroit. Add the following command to redistribute information between routers in Chicago:

```
router iso-igrp sales
 redistribute static
```

Static Interdomain Routing Example

Figure 12 and the following example show how to configure two routers that distribute information across domains. In this example, Router A (in domain Orion) and Router B (in domain Pleiades) communicate across a serial link.

Figure 12 CLNS Interdomain Static Routing



The following example shows how to configure Router A for static interdomain routing:

```
! Define tag orion for net 47.0006.0200.0100.0102.0304.0506.00
router iso-igrp orion
! Configure the following network entity title for the routing process
 net 47.0006.0200.0100.0102.0304.0506.00
! Define the tag bar to be used in place of Router B's NSAP
clns host bar 47.0007.0200.0200.1112.1314.1516.00
! Specify iso-igrp routing using the previously specified tag orion
interface ethernet 0
 clns router iso-igrp orion
! Pass ISO CLNS traffic to end systems without routing
interface serial 1
 clns enable
! Configure a static route to Router B
 clns route 47.0007 bar
```

The following example shows how to configure Router B for static interdomain routing:

```
router iso-igrp pleiades
! Configure the network entity title for the routing process
 net 47.0007.0200.0200.1112.1314.1516.00
! Define the name sid to be used in place of Router A's NSAP
clns host sid 47.0006.0200.0100.0102.0304.0506.00
! Specify iso-igrp routing using the previously specified tag pleiades
interface ethernet 0
 clns router iso-igrp pleiades
! Pass ISO CLNS traffic to end systems without routing
```

```
interface serial 0
 clns enable
! Pass packets bound for sid in domain 47.0006.0200 through serial 0
 clns route 47.0006.0200 sid
```

CLNS routing updates will not be sent on the serial link; however, CLNS packets will be sent and received over the serial link.

CLNS Filter Examples

The following example shows how to allow packets if the address starts with either 47.0005 or 47.0023. It implicitly denies any other address.

```
clns filter-set US-OR-NORDUNET permit 47.0005...
clns filter-set US-OR-NORDUNET permit 47.0023...
```

The following example shows how to deny packets with an address that starts with 39.840F, but allows any other address:

```
clns filter-set NO-ANSI deny 38.840F...
clns filter-set NO-ANSI permit default
```

The following example shows how to build a filter that accepts end system adjacencies with only two systems, based only on their system IDs:

```
clns filter-set ourfriends ...0000.0c00.1234.**
clns filter-set ourfriends ...0000.0c00.125a.**

interface ethernet 0
 clns adjacency-filter es ourfriends
```

Route Map Examples

The following example shows how to redistribute two types of routes into the integrated IS-IS routing table (supporting both IP and CLNS). The first routes are OSPF external IP routes with tag 5, and these are inserted into Level 2 IS-IS LSPs with a metric of 5. The second routes are ISO IGRP derived CLNS prefix routes that match CLNS filter expression “osifilter.” These routes are redistributed into IS-IS as Level 2 LSPs with a metric of 30.

```
router isis
 redistribute ospf 109 route-map ipmap
 redistribute iso-igrp nsfnet route-map osimap
!
 route-map ipmap permit
 match route-type external
 match tag 5
 set metric 5
 set level level-2
!
 route-map osimap permit
 match clns address osifilter
 set metric 30
 clns filter-set osifilter permit 47.0005.80FF.FF00
```

The following example shows how to redistribute a RIP learned route for network 160.89.0.0 and an ISO IGRP learned route with prefix 49.0001.0002 into an IS-IS Level 2 LSP with a metric of 5:

```
router isis
 redistribute rip route-map ourmap
 redistribute iso-igrp remote route-map ourmap
!
 route-map ourmap permit
 match ip address 1
 match clns address ourprefix
 set metric 5
 set level level-2
!
 access-list 1 permit 160.89.0.0 0.0.255.255
 clns filter-set ourprefix permit 49.0001.0002...
```

DECnet Cluster Aliases Example

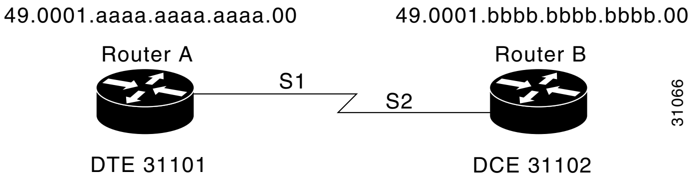
The following example shows how to enable cluster aliasing for CLNS:

```
clns routing
clns nsap 47.0004.004d.0001.0000.0C00.1111.00
router iso-igrp pleiades
! enable cluster aliasing on interface ethernet 0
interface ethernet 0
 clns cluster-alias
! enable cluster aliasing on interface ethernet 1
interface ethernet 1
 clns cluster-alias
```

ISO CLNS over X.25 Example

The following example shows how a serial interface 1 on Router A acts as data terminal equipment (DTE) for X.25. It permits broadcasts to pass through. Router B is an IS, which has a CLNS address of 49.0001.bbbb.bbbb.bbbb.00 and an X.121 address of 31102. Router A has a CLNS address of 49.0001.aaaa.aaaa.aaaa.00 and an X.21 address of 31101. Figure 13 illustrates this configuration.

Figure 13 Routers Acting as DTE and Data Circuit-Terminating Equipment(DCE)



Router A

```
router iso-igrp test-proc
net 49.0001.aaaa.aaaa.aaaa.00
!
interface serial 1
 clns router iso-igrp test-proc
! assume the host is a DTE and encapsulates x.25
 encapsulation x25
! Define the X.121 address of 31101 for serial 1
 X25 address 31101
! Set up an entry for the other side of the X.25 link (Router B)
 x25 map clns 31102 broadcast
```

Router B

```
router iso-igrp test-proc
 net 49.0001.bbbb.bbbb.bbbb.00
!
interface serial 2
 clns router iso-igrp test-proc
! Configure this side as a DCE
 encapsulation x25-dce
! Define the X.121 address of 31102 for serial 2
 X25 address 31102
! Configure the NSAP of Router A and accept reverse charges
 x25 map clns 31101 broadcast accept-reverse
```

Performance Parameters Example

The following example shows how to set ES hello packet and IS hello packet parameters in a simple ISO IGRP configuration, along with the MTU for a serial interface

```
router iso-igrp xavier
 net 49.0001.004d.0002.0000.0C00.0506.00
! Send IS/ES hellos every 45 seconds
clns configuration-time 45
! Recipients of the hello packets keep information in the hellos for 2 minutes
clns holding-time 120
! Specify an MTU of 978 bytes; generally, do not alter the default MTU value
interface serial 2
 clns mtu 978
```

TARP Configuration Examples

The following two sections provide basic and complex examples of TARP configuration.

Basic TARP Configuration Example

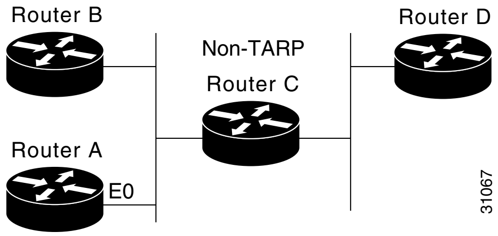
The following example shows how to enable TARP on the router and Ethernet interface 0. The router is assigned the TID myname.

```
clns routing
tarp run
tarp tid myname

interface ethernet 0
 tarp enable
```

Complex TARP Configuration Example

Figure 14 and the following example show how to enable TARP on Router A and on interface Ethernet 0, and assign the TID myname. A static route is created from Router A (49.0001.1111.1111.1111.00) to Router D (49.0004.1234.1234.1234.00) so that Router D can receive TARP PDUs because Router C is not TARP capable. A blacklist adjacency is also created on Router A for Router B (49.001.7777.7777.7777.00) so that Router A does not send any TARP PDUs to Router B.

Figure 14 Sample TARP Configuration



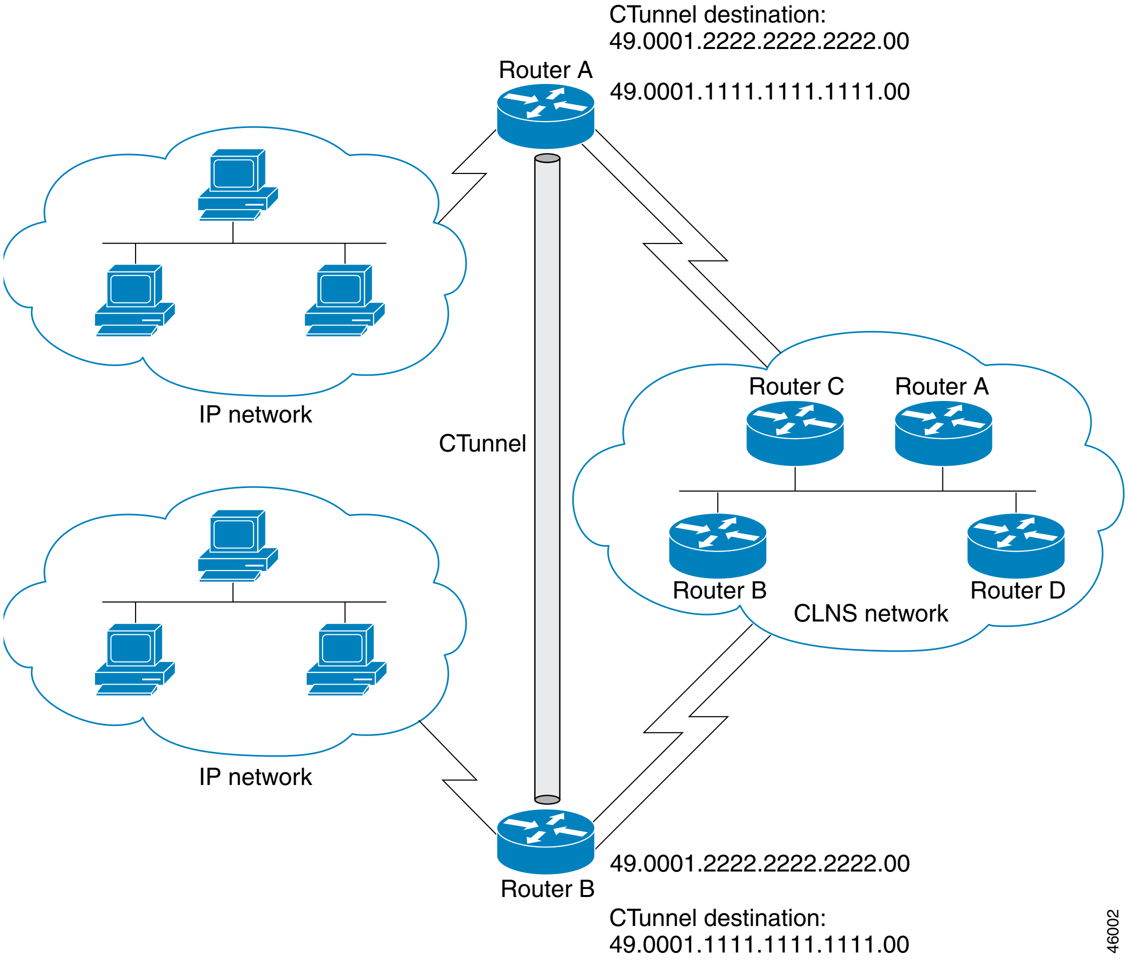
Router A

```
clns routing
tarp run
tarp cache-timer 300
tarp route-static 49.0004.1234.1234.1234.00
tarp blacklist-adjacency 49.0001.7777.7777.7777.00
tarp tid myname
interface ethernet 0
 tarp enable
```

IP over a CLNS Tunnel Example

Figure 15 illustrates the creation of a CTunnel between Router A and Router B, as accomplished in the configuration examples that follow for Router A and Router B:

Figure 15 Creation of a CTunnel



Router A

```
ip routing
clns routing

interface ctunnel 102
 ip address 10.0.0.1 255.255.255.0
 ctunnel destination 49.0001.2222.2222.2222.00

interface Ethernet0/1
 clns router isis

router isis
 net 49.0001.1111.1111.1111.00

router rip
 network 10.0.0.0
```

Router B

```
ip routing
clns routing
```

```
interface ctunnel 201
 ip address 10.0.0.2 255.255.255.0
 ctunnel destination 49.0001.1111.1111.1111.00

interface Ethernet0/1
 clns router isis

router isis
 net 49.0001.2222.2222.2222.00

router rip
 network 10.0.0.0
```

Configuring GRE/CLNS CTunnels to Carry IPv4 and IPv6 Packets Example

The following example configures a GRE CTunnel running both IS-IS and IPv6 traffic between Router A and Router B in a CLNS network. The **ctunnel mode gre** command provides a method of tunneling that is compliant with RFC 3147 and should allow tunneling between Cisco equipment and third-party networking devices.

Router A

```
ipv6 unicast-routing

clns routing

interface ctunnel 102
 ipv6 address 2001:0DB8:1111:2222::1/64
 ctunnel destination 49.0001.2222.2222.2222.00
 ctunnel mode gre

interface Ethernet0/1
 clns router isis

router isis
 network 49.0001.1111.1111.1111.00
```

Router B

```
ipv6 unicast-routing

clns routing

interface ctunnel 201
 ipv6 address 2001:0DB8:1111:2222::2/64
 ctunnel destination 49.0001.1111.1111.1111.00
 ctunnel mode gre

interface Ethernet0/1
 clns router isis

router isis
 network 49.0001.2222.2222.2222.00
```

To turn off GRE mode and restore the CTunnel to the default Cisco encapsulation routing only between endpoints on Cisco equipment, use either the **no ctunnel mode** command or the **ctunnel mode cisco** command. The following example shows the same configuration modified to transport only IPv4 traffic.

Router A

```
ip routing

clns routing

interface ctunnel 102
 ip address 10.2.2.5 255.255.255.0
 ctunnel destination 49.0001.2222.2222.2222.00
 ctunnel mode cisco

interface Ethernet0/1
 clns router isis

router isis
 network 49.0001.1111.1111.1111.00
```

Router B

```
ip routing

clns routing

interface ctunnel 201
 ip address 10.0.0.5 255.255.255.0
 ctunnel destination 49.0001.1111.1111.1111.00
 ctunnel mode cisco

interface Ethernet0/1
 clns router isis

router isis
 network 49.0001.2222.2222.2222.00
```

Additional References

The following sections provide references related to the ISO CLNS feature.

Related Documents

Related Topic	Document Title
ISO CLNS commands	Cisco IOS ISO CLNS Command Reference
Configuring Transparent Bridging	Cisco IOS Bridging and IBM Networking Configuration Guide

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/techsupport

Feature Information for Configuring ISO CLNS

Table 5 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.3(7)T or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp. An account on Cisco.com is not required.

Note Table 5 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 5 Feature Information for Configuring ISO CLNS

Feature Name	Releases	Feature Information
Configuring ISO CLNS	12.3(7)T	This feature was introduced.
CLNS Support for GRE Tunneling of IPv4 and IPv6	12.2(25)S 12.2(33)SRA 15.0(1)M Cisco IOS XE Release 2.6	This feature was integrated with Cisco Release 12.2(25)S. Cisco CTunnels transports IPv4 and IPv6 packets over CLNS-only networks in a manner that allows interoperation between Cisco networking equipment and that of other vendors. The CLNS support to the GRE tunnel mode allows this transportation to happen. The CLNS feature provides compliance with RFC 3147.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.