CIT Quick Reference Sheets

Network Troubleshooting and Baselining

The General Troubleshooting Process

You can characterize troubleshooting by using a three-stage process. Unfortunately, you might need to repeat this process until a resolution is found. The three steps of the process are

1. **Gather symptoms**—During this step, you must gather and document symptoms of the problem. It is also during this step that problem ownership is determined.
2. **Isolate the problem**—At this point in the process, you must choose and apply the correct troubleshooting strategy.
3. **Correct the problem**—Use the information that you have gained in the previous steps to correct the problem. It is important that the operational status of the network be assessed once this step is complete to ensure that the changes have not had a negative impact.

Gathering Symptoms

The first step in the troubleshooting process, which is to gather the symptoms, has five steps itself:

1. **Analyze existing symptoms**—During this step, you must examine the currently documented symptoms and document any others that are known, but undocumented.
2. **Determine ownership**—At this point, you need to leverage your understanding of the technical and organizational environment to determine ownership of the problem. If the problem is within a system you are responsible for, continue the process. If not, contact and work with the responsible party.
3. **Narrow the scope**—Combining the information from the first two steps, narrow the scope of the problem by eliminating symptoms of symptoms and focusing on the root cause. In general, use this step to focus on what you know and eliminating superfluous information.
4. **Determine symptoms**—If necessary, gather additional symptoms attempting to gain new insight into the problem.
5. **Document symptoms**—Simply document the symptoms you know.

Types of Troubleshooting Methodologies

Three general troubleshooting methodologies help you in the fault-finding process:

- **Top-down**—This approach begins the troubleshooting process at Layer 7 of the OSI model and works down.
- **Bottom-up**—This methodology starts at the physical layer, Layer 1 of the OSI model, and works up.
- **Divide and conquer**—Leverages past experiences to begin troubleshooting in the middle of the OSI model. This model initially troubleshoots in both directions and is considered to be a more advanced technique when compared to the top-down or bottom-up methodologies.

Guidelines for Network Documentation

- **Determine your scope**—Determine which portion of the network or what aspects of the network this documentation covers. Without this determination, it is likely that the resultant documentation will not cover the appropriate material at an appropriate depth.
- **Know your objective**—Understand what your document is trying to accomplish. Then, make sure that each component works towards accomplishing that goal.
- **Be consistent**—Being consistent aids others in understanding your documentation. Consistent conventions also help troubleshooters cross reference information they need.
- **Keep the document accessible**—What good is the network documentation if you can't access it during a network outage? It is critical to keep documentation accessible to the people who need it. Otherwise, its creation was just an exercise.
- **Maintain the documentation**—The only thing that is possibly worse than no documentation is old incorrect documentation. It is important that troubleshooters trust the documentation and do not have to waste time validating or correcting it.

Reasons to Baseline a Network

- Give yourself or others an understanding of the network.
- Document the normal operation for comparison during troubleshooting and fault isolation.

Steps in the Network Discovery Process

The network discovery process is made up of four steps. By using these steps consistently, your network documentation is more accurate and complete:

1. **Document the current device name and addresses**—In this step, you must detail the network layer properties of the current device.
2. **Document all active interfaces**—Not only is this information important to the present device, it is used in the discovery process, which happens next.
3. **Discover the directly connected devices**—Use the network layer addressing and interface information documented in the previous steps to discover adjacent network devices. Several **show** commands can help you in discovering the network; they are detailed in the section, "Helpful Commands in Network Discovery and Documentation."
4. **View details about connected devices**—Leverage available tools to view and document information about connected devices.

Network Topology Diagrams

Network topology diagrams greatly aid the troubleshooting process. Once created, they give the troubleshooter the following information:

- A quick overview of the network
- A visual representation of how devices are connected
- A list the device names and locations
- An overview of the routing protocol configurations, such as area definitions and points of summarization

Components of a Network Configuration Table

A vital component of any set of network documentation, the network configuration tables need to capture all-important aspects of a router or switch configuration. The following critical points need to be included in network configuration tables for routers and switches alike:

- System name
- Software version and filename
- Interface types
- Network layer addresses
- Data-link addresses
- Router memory and central processing unit (CPU)
- Routing protocol configuration
- Any other critical configuration or hardware component

Important Points to Document for Ethernet Switches

Specific to an Ethernet switch, the following items allow the reader to quickly understand the switch configuration, and if needed, quickly recreate it:

- Switch name
- IP address
- STP bridge priority
- VLAN Trunking Protocol (VTP) domain name
- VTP role
- Per-port information
  —STP state
  —Speed and duplex configuration
  —VLAN assignment
  —Trunking configuration
  —EtherChannel configuration

Important Points to Document for Routers

The following items are critical to the operation of a router. As such, it is important that these details be found in any relevant network configuration tables:

- Router name
- Network layer addresses
- Interface types
  —Per interface settings
  —Speed and duplex
- Routing protocol configuration
  —Router ID
  —Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS) area definition
  —Summarization
  —Autonomous system number

Helpful Commands in Network Discovery and Documentation

This table lists some EXEC commands that are useful when discovering or documenting a network.

Command	Description
show cdp neighbors [**detail**]	Used to display the adjacent Cisco devices. The **detail** option displays additional information, including any configured network layer addresses.
show ip ospf neighbor	Displays all active OSPF neighbors.
show ip eigrp neighbors	Displays the Enhanced Interior Gateway Routing Protocol (EIGRP) adjacencies.
show ip route	Displays the IP routing table.
show ip arp [**interface** *type number*]	Shows the Address Resolution Protocol (ARP) cache. The output of this command can be displayed per-interface using the interface argument.
show ip bgp summary	Displays the Border Gateway Protocol (BGP) peers configured on this router.
show clns neighbor	Used to display information about neighboring IS-IS routers.

Troubleshooting TCP/IP

Static Routes

Static routes can be good and bad. They are often the source of network problems, but at the same time, understanding their operation and benefits can aid in troubleshooting. To create a static route, use the following global configuration command:

```
ip route prefix mask {ip-address | interface-type interface-number
  [ip-address]} [distance] [permanent]
```

This command has few, but nonetheless important, options. The following table explains these options.

Option	Description
prefix	The destination network.
mask	The mask to apply to the entered prefix.
ip-address	The next hop for the static route.
interface-type interface-number	Configures a specific output interface for the static route; should only be used on point-to-point interfaces. When used on multi access interfaces, this causes the router to ARP for any destination included in that route.
distance	The administrative distance for the route.
permanent	Specifies that the route should not be removed from the routing table when the associated interface is down.

Administrative Distance

The router assigns a value called administrative distance (AD) to sources of routing information. This value is used to select a route when multiple routes to the same destination are available from different sources. A lower AD indicates a higher level of preference. The following table shows the default AD for each available source of IP routing information.

0	Connected routes
1	Static routes
5	EIGRP summary route
20	External BGP (EBGP)
90	Internal EIGRP

100	IGRP
110	OSPF
115	IS-IS
120	Routing Information Protocol (RIP)
140	Exterior Gateway Protocol (EGP)
160	On-Demand Routing (ODR)
170	External EIGRP
200	Internal BGP (IBGP)
255	Unknown

The AD can be changed for any routing protocol using the router configuration command:

```
distance distance {ip-address {wildcard-mask}} [ip-standard-list]
  [ip-extended-list]
```

For EIGRP and BGP, the commands are slightly different. To change the AD for EIGRP, use the router configuration command:

```
distance eigrp internal-distance external-distance
```

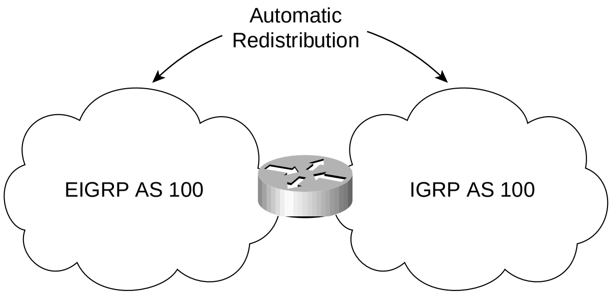
For BGP, use the router configuration command:

```
distance bgp external-distance internal-distance local-distance
```

Route Redistribution

Redistribution is the process of importing routes from an outside source. Examples of redistribution include the advertising of static routes to OSPF neighbors, and the broadcasting of OSPF routes to other routers running RIP.

Within the context of IP, redistribution takes place automatically in one instance, when a router is running both EIGRP and IGRP with the same autonomous system numbers.



For all other cases of redistribution, manual administrator intervention is required. When performing redistribution, the following items must be taken into account:

- **Incompatible metrics**—The metrics used by different routing protocols are not compatible and must be manually configured.
- **Routing loops**—It is possible to introduce routing loops when multiple routers are performing redistribution; to prevent this, configure filtering to limit the routes that are redistributed.

Route redistribution is configured with the following router configuration command:

```
redistribute protocol [process-id] {level-1 | level-1-2 | level-2} [metric
  metric-value] [metric-type type-value] [match {internal | external 1 |
  external 2}] [route-map map-tag] [subnets]
```

The following table explains the many options to the **redistribute** command.

Option	Description
protocol	The routing protocol to redistribute routes from.
process-id	The process ID of the routing process to import from.
level-1	For IS-IS, redistribute only L1 routes.

level-1-2	For IS-IS, redistribute both L1 and L2 routes.
level-2	For IS-IS, redistribute only L2 routes.
metric *metric-value*	Set the default metric in this routing protocol for the redistributed routes.
metric-type *type-value*	For OSPF, redistributed routes can be either external type 1 or external type 2. Enter either 1 or 2 to select the external route type for redistributed routes. For IS-IS, redistributed routes can be either internal or external. Enter either internal or external to configure the IS-IS type for redistributed routes.
match {**internal** | **external 1** | **external 2**}	Redistribute OSPF routes only of the specified type.
route-map *map-tag*	Use the specified route map to filter or alter redistributed routes.
subnets	For OSPF, redistribute networks using masks longer than the appropriate natural mask.

If you decide not to set the default metric for redistributed routes on the **redistribute** command, you can use one of the following.

BGP	**default-metric** *metric*
EIGRP	**default-metric** *bandwidth delay reliability loading mtu*
IGRP	**default-metric** *bandwidth delay reliability loading mtu*
OSPF	**default-metric** *metric*
RIP	**default-metric** *metric*

IOS TCP/IP Troubleshooting Commands

Many IOS commands can aid in the troubleshooting of TCP/IP problems. The following table explains these commands and their usefulness.

Command	Description
clear arp	Removes all entries in the router's ARP cache.
clear ip route {* | *ip-address* [*mask*]}	Removes a specific route from the router's routing table.
ping *ip-address*	Tests ICMP connectivity between the router and a remote IP address.
ping (Extended)	An extended ping allows you to specify the following items: • Repeat count • Datagram size • Timeout • Source address or interface • IP type of service • Don't Fragment bit • Validate reply data • Data pattern • Source routing • Sweep packet sizes
show ip access-list [*access-list*]	Shows the configured access lists and how many times each access list entry (ACE) has been matched; the output can be limited by specifying a particular access list.
show interfaces [*type number*]	Shows the operational status of a router interface and the IP address and mask in use.

show ip arp [*ip-address*] [**interface** *type number*]	Shows the contents of the router ARP cache; can be limited to a specific IP address or interface.
show ip interfaces [*type number*]	Displays detailed information about the IP operation of all interfaces or a specific interface. The information includes the IP address, mask, routing protocol metrics, route cache flag, MTU, and applied access lists.
show ip protocols	Displays the configuration and status of any configured routing protocols.
show ip route [*source*]	Displays the contents of the routing table; output can be limited to a specific routing protocol.
trace *ip-address*	Displays the path that UDP packets take to the remote IP address.
trace (Extended)	Traces the path through the network that UDP packets take from the router to a remote destination. Using the extended form of this command, you can set: • Target IP address • Source addresses • Numeric display • Timeout • Probe count • Minimum Time-To-Live (TTL) • Maximum TTL • UDP port number • Source routing options

Workstation Troubleshooting Commands

Several commands can be issued on Windows and UNIX workstations to aid in the IP troubleshooting process.

The following command, common to Windows and UNIX, can help determine if an IP address is reachable from the designated workstation:

```
ping ip-address
```

The following commands display the IP configuration of all configured network interfaces:

```
(Windows) ipconfig /all
(UNIX) ipconfig -a
```

The following commands output the routing table of the workstation they are issued on. Just like a router's routing table, these commands display the network, mask, and next hop information for all routes in the table:

```
(Windows) route print
(UNIX) netstat -rn
```

The following commands display an approximation of the path packets between the workstation and specified IP address take:

```
(Windows) tracert ip-address
(UNIX) traceroute ip-address
```

NOTE: It is important to use the output of any **traceroute** command (from a router, Windows, or UNIX workstation) with a grain of salt. Because of asymmetric routing, policy routing, and other factors, the path reported by traceroute might or might not reflect the actual path of user traffic.

TCP/IP Debugging Commands

The following commands can be used to debug IP-related problems.

This command displays ARP requests sent and answered by the router:

```
debug arp
```

This command displays information about EIGRP packets sent and received by the router:

```
debug ip eigrp
```

The **debug** command:

```
debug ip ospf events
```

displays information about the following OSPF events:

- Adjacency management
- Link State Advertisement (LSA) flooding
- Designated router elections
- SPF calculations

This command displays real-time information about OSPF packets sent and received by the router. The output can be limited by specifying an access list:

```
debug ip ospf packet [access-list]
```

The following command displays information about packets routed by the router. The output can be limited by specifying an access list:

```
debug ip packet [access-list]
```

The following command displays information about the operation of the Routing Information Protocol (RIP):

```
debug ip rip
```

Troubleshooting Switched Ethernet Networks

Protocols Used on a Switched Network

- **VLAN Trunking Protocol** (VTP)—Used to disseminate information about VLANs throughout the network. Switches are grouped into a VTP domain. Allows for simple security using VTP domain passwords. VTP pruning, which is disabled by default, automatically removes unneeded VLANs from trunk ports.

  Switches can be configured for one of these three VTP roles:

  —**Server**—Can make changes to the VTP database. Servers multicast changes to other VTP servers and VTP clients in the domain.

  —**Client**—Cannot make changes to the VTP database. Clients listen for VTP database announcements from servers.

  —**Transparent**—Can configure VLANs locally. Although transparent switches do not advertise VTP messages, they forward those VTP messages heard unaltered.

- **Dynamic Trunking Protocol** (DTP)—Used to automatically configure VLAN trunking on switch to switch links. Available modes include on, off, desirable, and auto. Should be disabled on workstation ports.
- **Spanning Tree Protocol** (STP)—Used to remove loops from the network. Defined as the 802.1d standard. Port Fast reduces edge port enablement time from 50 to about 5 seconds. STP has several available modes of operation, such as

  —Per VLAN Spanning Tree Plus (PVST+)

  —Multiple Spanning Tree Protocol (MST)

  —Common Spanning Tree Protocol (CST)

- **Inter-Switch Link** (ISL)—This Cisco proprietary protocol allows multiple VLANs to traverse one link. Adds VLAN tag to each packet. Supports VLANs numbered 1 through 1005.
- **802.1Q Trunk Protocol**—Standards-based protocol. Allows multiple VLANs to travel across one link. Adds VLAN tag to each packet. Traffic on Native VLAN does not have 802.1Q header added. 802.1Q header includes 802.1P class of service bits. Supports VLANs number from 1 to 4094.
- **EtherChannel**—Provides additional bandwidth through the aggregation of multiple ports. Appears as one port to the Spanning Tree Protocol. Can support up to eight interfaces. (Interfaces can be added or removed seamlessly.) EtherChannel can use the following load distribution methods: per source MAC, per destination MAC, per source IP address, per destination IP address, per source IP port, and per destination IP port. EtherChannel has three current flavors: EtherChannel, Fast EtherChannel (FEC), and Gigabit EtherChannel (GEC).
- **Cisco Discovery Protocol** (CDP)—Allows devices and administrators to learn information from neighboring network devices. The information carried in CDP includes

  —Model

  —Capabilities

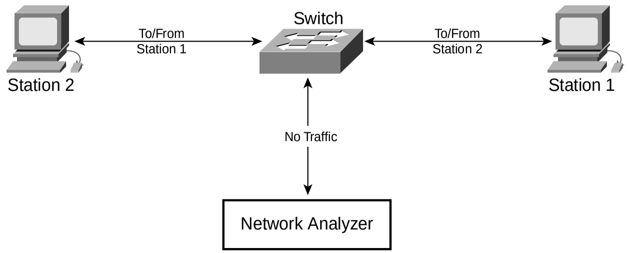
  —Software revision

  —Layer 3 addresses

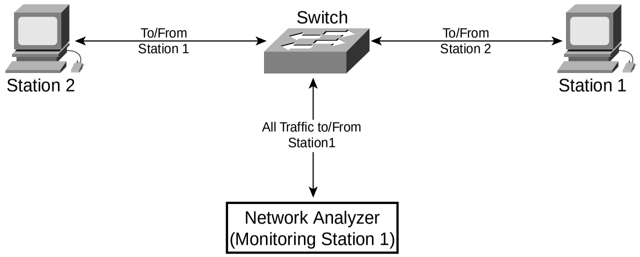
  —Interface

Using SPAN and RSPAN

Switched networks inherently limit the visibility of network analyzers because traffic is directed to specific switch ports and not simply replicated to all ports.



To gain visibility into network traffic when troubleshooting must take place, a Switch Port ANalyzer (SPAN) port replicates traffic to or from a switch port or VLAN to the port connected to a network analyzer.



The syntax to enable a SPAN differs between CatOS and IOS switches; both syntaxes are shown as follows:

CatOS

```
set span {src_mod/src_ports | src_vlans | sc0} {dest_mod/dest_port}
  [rx | tx | both]
```

IOS

```
monitor session session source {interface type | vlan vlan-id [rx | tx |
  both]}
```

and

```
monitor session session destination {interface type | vlan vlan-id}
```

Troubleshooting Error Disabled Ports

A port might become error disabled when one of the following occurs:

- Excessive collisions.
- Late collisions.
- EtherChannel misconfiguration.
- BPDU port-guard is violated.
- A unidirectional link is detected via Unidirectional Link Detection (UDLD).

To reenable an error disabled port, one of the following commands must be used:

```
(CatOS) set port enable mod/port
(IOS) shutdown
```

then

```
no shutdown
```

Alternatively, error disabled ports can be automatically enabled after a period of time using the commands:

```
(CatOS) set errdisable-timeout interval timeout
(IOS) errdisable recovery interval interval
```

Important Troubleshooting Commands

Although not categorized as debug commands, these commands can be of great assistance when troubleshooting switched networks.

The following CatOS command configures a proven set of parameters for workstation ports. Specifically, it turns on STP Port Fast and disables trunking and channeling:

```
set port host mod/port
```

Although the following command is not a **debug** command, it does allow direct access to intelligent Catalyst 6500 modules. Such modules contain their own configuration and must be accessed directly to be configured:

```
(CatOS) session slot module processor proc
(IOS) session slot
```

The following commands are used to view the switch's Layer 2 forwarding table:

```
(CatOS) show cam
(IOS) show mac-address-table
```

This CatOS command displays counter information for all or the specified ports. These counters include the number of bytes transmitted and received, as well as the number and types of errors received:

```
show port [mod/port]
```

The following commands display information about the state of STP on the specified VLAN. This information includes the root bridge ID and the switch's designated and root ports:

```
(CatOS) show spantree number
(IOS) show spanning-tree vlan number
```

These commands display information about ports that have been blocked by the STP:

```
(CatOS) show spantree blockedports
(IOS) show spanning-tree blockedports
```

Debugging Commands

The Cisco IOS Software allows for extensive debugging facilities. The commands detailed as follows give you insight into the operation of your network, and aid in the troubleshooting process. These commands are not available on CatOS devices.

The following debug command displays information about the establishment and maintenance of EtherChannel groups:

```
debug etherchnl
```

The following debug command displays information about interface state changes on the designated interface:

```
debug interface mod/port
```

The following debug command displays real-time information about the operation of multilayer switching (MLS) for switches that are acting as an MLS Route Processor (RP):

```
debug mls rp {all | error | events | ip | ipx | locator | packets | verbose
  packets}
```

The following debug command displays information about the operation of any configured SPAN sessions:

```
debug monitor
```

The following command provides real-time information into the Spanning Tree Protocol. With many options, the output can be granular:

```
debug spanning-tree {all | bpdu | bpdu-opt | etherchannel | config |
  events | exceptions | general | pvst+ | root | snmp}
```

Troubleshooting PPP

Components of PPP

The Point-to-Point Protocol (PPP) is modular and is made up of many components. The following sections attempt to detail many of the most used protocol components.

- The Link Control Protocol (LCP) establishes and tests Layer 2 connectivity.
- Authentication can take place in one or both directions and takes place before any Network Control Protocols (NCPs) are run.
  —Challenge Handshake Authentication Protocol (CHAP) uses a three-way handshake in which passwords are never sent over the network.
  —Password Authentication Protocol (PAP) sends username and password information unencrypted.
- Network Control Protocols (NCPs)
  —A NCP exists for each protocol.
  —Negotiates protocol specific parameters.
  —Example NCPs include IPCP, IPXCP, ATALKCP, CDPCP, and BridgeCP.
- Link Quality Monitoring (LQM)
  —Measures link quality
  —Disables unreliable links
  —Replaces regular keepalives
  —Can be enabled on one or both PPP endpoints
- Multilink PPP
  —Allows bonding of multiple physical interfaces into a single logical interface
  —Packet fragmentation
  —Fragment interleaving
  —Works on synchronous and asynchronous connections

Debugging PPP Connections

A suite of commands can help you troubleshoot PPP connection establishment problems. The following table details these commands.

debug ppp authentication	Displays information relevant to the operation of PAP and CHAP.
debug ppp compression	Displays information about the negotiation of compression.
debug ppp errors	Outputs errors and error statistics during connection establishment.
debug ppp multilink events	Displays information about system events affecting the multilink PPP connection.
debug ppp multilink fragments	Displays information about multilink PPP fragments sent and received over the connection.
debug ppp multilink negotiation	Outputs information about the negotiation of multilink PPP during connection establishment.
debug ppp negotiation	Displays information about the negotiation of parameters via LCP and the NCPs.
debug ppp packet	Outputs the details of keepalives and Link Quality Reports (LQRs) as they traverse the connection.

Client IP Address Assignment

A PPP client can be assigned an IP address in several ways. The following is a prioritized list of the available methods:

- AAA server IP pool
- Local IP pool or Dynamic Host Configuration Protocol (DHCP) configuration
- Dialer map address
- Address specified via the **ppp** or **slip** EXEC commands
- The **peer default ip address** *address* interface configuration command
- Peer provided IP address

The default address allocation method can be controlled with the following global configuration command:

```
ip address-pool {dhcp-proxy-client | local }
```

An IP address pool is configured using the following command:

```
ip local pool pool-name start-address end-address
```

After the pool is created, it can be applied with the interface configuration command that follows:

```
peer default ip address pool pool-name
```

To configure the access server to use DHCP to determine addresses available for client assignment, use the global configuration command that follows:

```
ip dhcp-server {ip-address | name}
```

After the DHCP servers are configured and the default address allocation method set to DHCP, the following interface configuration command enables IP address assignment from the DHCP retrieved addresses:

```
peer default ip address pool dhcp
```

A specific address can be assigned to user dialing into a particular access server interface with the following interface configuration command:

```
peer default ip address ip-address
```

Configuring and Tuning Multilink PPP

The following commands configure and customize the implementation of multilink PPP.

The following command allows you to determine at what point additional connections are established using other rotary group members:

```
dialer load-threshold load [inbound | outbound | either]
```

The following command creates a virtual multilink interface. Member interfaces are linked to this interface using the **multilink-group** interface configuration command:

```
interface multilink number
```

The following global configuration command links all synchronous interfaces configured for multilink PPP to a single virtual-template interface:

```
multilink virtual-template number
```

The following interface configuration command links physical interfaces to a single multilink virtual interface. The virtual interface is configured with the global configuration command **interface multilink** *number*:

```
multilink-group number
```

The following interface configuration command enables multilink PPP on an interface:

```
ppp multilink
```

The following command enables multilink PPP fragmentation on an interface. This is enabled by default and can be disabled using the command **ppp multilink fragment disable**:

```
no ppp multilink fragment disable
```

The following interface configuration command enables interleaving of multilink PPP fragments. This is disabled by default for interfaces using multilink PPP encapsulation:

```
ppp multilink interleave
```

Verification of a PPP Configuration

The following commands can be used to view the status of a configured PPP configuration:

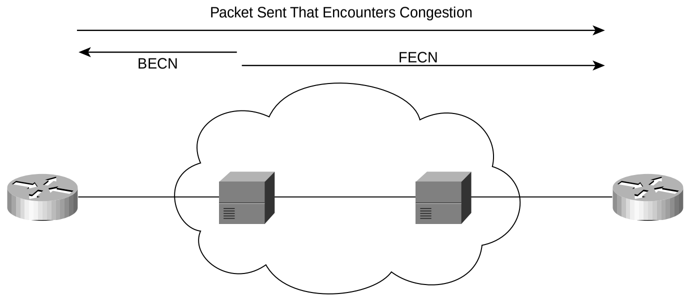
- **ping** *ip-address*
- **show compress**
- **show interfaces** [*type number*]
- **show ppp multilink**
- **show users** [all]

Supporting Frame Relay

Key Frame Relay Troubleshooting Terminology

Frame Relay is an extremely widespread protocol. Understanding its mechanisms greatly increases your effectiveness in troubleshooting its problems.

- **Backwards Explicit Congestion Notification** (BECN)—Sent by the Frame Relay network to indicate congestion on a virtual circuit (VC) in the opposite direction. If a router receives a BECN, it is an indication of congestion from that router toward the other end of the VC.
- **Forward Explicit Congestion Notification** (FECN)—Sent by the Frame Relay network to indicate congestion on a VC in the same direction as the FECN traveled.



- **The Discard Eligible** (DE) **bit**—Set in the Frame Relay header of packets that are in excess of the Committed Information Rate (CIR) on a VC. The DE bit can be set by the network or transmitting Frame Relay endpoint.
- **The Local Management Interface** (LMI)—A means of communication between Frame Relay switch and Frame Relay endpoint. The status of VCs is communicated using LMI.

Configuring a Router to Interact with a Congested Frame Relay Network

The map class configuration command

```
frame-relay adaptive-shaping becn
```

instructs the router to slow transmission into the network when a BECN is received. The slower transmission rate is applied on a per-VC basis and is by default half of the configured CIR. This value, called MinCIR, can be explicitly configured with the map class configuration command:

```
frame-relay mincir mincir
```

Using the **frame-relay de-group** and **frame-relay de-list** commands, it is possible to selectively set the DE bit in outgoing packets. The following example sets the DE bit in all outgoing IPX packets and IP packets larger than 512 bytes; all other packets are unchanged:

```
Router(config)# frame-relay de-list 1 protocol ipx
Router(config)# frame-relay de-list 1 protocol ip gt 512
Router(config)# interface Serial 0.16
Router(config-subif)# frame-relay interface-dlci 16
Router(config-fr-dlci)# frame-relay de-group 1 16
```

Viewing the Status of Frame Relay Connections

The following commands help you view the status of any configured Frame Relay connections.

The following command displays configuration and statistic information about the Frame Relay Local Management Interface (LMI). The important troubleshooting information includes the LMI type, interface type, and the number of LMI status messages sent and received:

```
show frame-relay lmi [interface type number]
```

The output of the following command includes the mappings of data-link connection identifiers (DLCIs) to Layer 3 addresses, as well as other characteristics of the connection, such as compression and LMI type:

```
show frame-relay map
```

When the following command displays information about an interface configured with Frame Relay encapsulation, it displays the Frame Relay specific information such as LMI and encapsulation types, LMI statistics, and DLCI used for LMI:

```
show interfaces [type number]
```

The following command displays the status and statistics for each VC. Statistics displayed include the number of BECN, FECN, and DE packets received for each VC. After LMI is configured, any VCs configured on the Frame Relay switch are display, including those not configured on the router:

```
show frame-relay pvc [interface name number] [dlci]
```

The following command shows the number of Frame Relay ARP request and replies sent and received:

```
show frame-relay traffic
```

The following command shows the configuration of traffic shaping on each VC. The output includes the target and peak rates, time interval (Tc), and the status of adaptive shaping:

```
show traffic-shape [type number]
```

The output of the following command includes statistics relative to the operation of traffic shaping on each VC. These statistics include the number of bytes and packets shaped, the queue depth, and if adaptive shaping is enabled:

```
show traffic-shape statistics [type number]
```

Debugging Frame Relay Connections

These debug commands provide real-time insight into the operation of Frame Relay.

The following command displays Frame Relay LMI messages as they are sent and received by the router. Once every 60 seconds, your router receives an LMI message including the configured permanent virtual circuits (PVCs) and their CIRs from the Frame Relay switch:

```
debug frame-relay lmi [interface type number]
```

Not restricted to just Frame Relay debugging, the following command displays keepalive messages sent and received:

```
debug serial interface
```

The following command displays real-time information about Frame Relay ARP messages sent and received:

```
debug frame-relay events
```

The following command displays information about each outgoing packet for all interfaces or the selected interface. The information displayed includes the DLCI, packet type, and packet size:

```
debug frame-relay packet [interface type number]
```