CHAPTER 1

Introduction to Cisco IOS for S/390

The *Cisco IOS for S/390 User's Guide* provides users with guidelines for using communications programs on Multiple Virtual System (MVS) operating systems on which Cisco IOS for S/390 is installed.

This guide describes command and program usage instructions for users working on an MVS system running Cisco IOS for S/390. Users who need to send mail, use Telnet to log on to remote computers, or use File Transfer Protocol (FTP) to transfer files among connected computers will find instructions for the necessary commands and their options in this guide.

This chapter describes Cisco IOS for S/390, its capabilities, and the programs it provides. In it, you will find these chapters:

- Overview

  Describes Cisco IOS for S/390, its functions within a network, and the role Cisco IOS for S/390 plays in your network communications.

- Software Components

  Describes the Cisco IOS for S/390 internal components and how they provide their services to users.

- Using Host Name Strings

  Describes how to make connections to hosts.

Overview

Cisco IOS for S/390 is a communication subsystem for the Department of Defense internet protocols. It runs on an IBM 370-style mainframe under the MVS operating system and provides

Cisco IOS for S/390 includes this Internet-specific protocol code:

- The local network I/O driver
- Code for the host-to-host protocol TCP/IP
- Programs for the user-level protocols

Cisco IOS for S/390 supports these standard user-level protocols:

- Telnet for remote login
- File Transfer Protocol (FTP) for file transfer
- Simple Mail Transfer Protocol (SMTP) for electronic mail

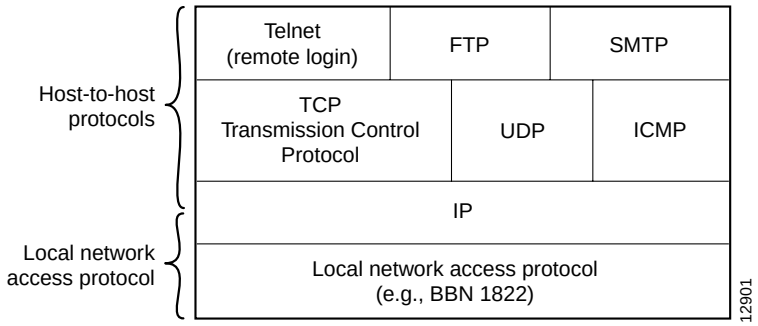
You can install Cisco IOS for S/390 on MVS/ESA with no operating system modifications; inter-process communication is processed with IBM's ACF/VTAM and Cross-Memory Services.

Software Components

The Cisco IOS for S/390 software provides several different protocols. (See Figure 1-1).

- Internet Protocol (IP)
- Internet Control Message Protocol (ICMP)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)
- The user level protocol, including Telnet, File Transfer Protocol (FTP), and Simple Mail Transfer Protocol (SMTP)

Figure 1-1 Protocol Hierarchy within Cisco IOS for S/390



Internet Protocol

IP provides datagram service in an inter-network environment. It sends datagrams between hosts that may be on different networks linked by packet-switching hosts called gateways. A datagram consists of an IP header followed by data.

From a host viewpoint, IP provides the principal functions of host addressing, fragmentation, and reassembly of packets, enabling communication between networks with different packet sizes. IP does not provide error detection and recovery.

Internet Control Message Protocol

ICMP is an extension of IP and carries routing, congestion control, and error reports to hosts.

Transmission Control Protocol

Transmission Control Protocol (TCP) is a reliable end-to-end protocol for transmitting data between processes over connections or virtual circuits. TCP uses IP to carry data packets and is the transport service protocol that most user-level protocols use. These are some characteristics of TCP:

- Segments

  TCP sends data in messages called segments, each of which begins with a TCP header and is sent as a datagram using IP. TCP delivers data segments to a user reliably and in order. The data in these ordered segments logically form an uncommitted stream of 8-bit data bytes or octets.

- Flow Control

  TCP provides flow control on each connection using a windowing mechanism in a fine-grain sequence space — in other words, a single octet of the data stream.

- Checksums

  TCP uses checksums to ensure end-to-end reliability. The receiver sends acknowledgments of correctly received data, and the sender does timer-based retransmission of unacknowledged data.

- Full-duplex Connections

  TCP creates full-duplex connections whose ends are labeled with 16-bit numbers called ports. A TCP connection is defined by a set of four address parameters:

  (*local_host_IP_address,local_port remote_host_IP_address,remote_port* **)**

  TCP allows the same local port on a host to participate in any number of connections whose remote ends have differing pairs (remote_host_IP_address, remote_port). Thus, a server host's well-known port (wkport) can participate in multiple TCP connections as long as the user host's pairs (remote_host_IP_address, remote_port) are each unique.

- Connection Management

  A TCP connection is full-duplex, even if an application needs only a simple connection. Furthermore, a TCP connection is allowed to be half-open indefinitely. A close request (FIN) signals the end of data transmission in only one direction; data flow can continue in the other direction until a matching FIN is sent. The connection is fully closed and deleted only when both ends send close requests.

User Datagram Protocol (UDP)

UDP provides datagram service between two processes. UDP does not define connections as TCP does. UDP does, however, have 16-bit port numbers like TCP; sending or receiving a UDP datagram requires the same set of four address parameters that define a TCP connection. As a result, UDP implementation is much like that of TCP, except UDP is simpler.

Using Host Name Strings

Both User Telnet and User FTP programs require you to identify the remote host to which a connection is to be established. Depending on the service to which the connection is established, you might also want to specify optional parameters, such as port number.

Host name strings comprise three sections as illustrated in the following:

host<route>,port

The only required portion of the host string is the host section.

Specifying the Host

Each host is assigned an Internet host number, or Internet address. Because Internet numbers are hard to remember and keep track of these host numbers, a Domain Name System tracks Internet addresses and correlates them to host names. So you can use names instead of numbers to reference host computers.

host_name.network_name.

Example

In the following example, the name identifies a particular host, UNIX, (host number 123.196.222.160) in the COMPANY.NAME.COM system. Notice that the name is terminated with a period. This identifies the name as being fully qualified.

UNIX.*COMPANY.NAME***.COM.**

When you enter a fully qualified host string, that is, a host string with the trailing period, Cisco IOS for S/390 processes it exactly as it was entered. When the name is not fully qualified, Cisco IOS for S/390 tries to resolve it using a search list set up by the system administrator. If the search list is properly set up, a host string such as HOST1 can be entered and Cisco IOS for S/390 processes it as if HOST1.COMPANY.COM. had been entered.

Sometimes a host name has an alternate name, or an alias, defined for it. If the target host has an alias, you can enter the alias. HOST1.COMPANY.COM. has the alias UNIX. Entering either of these host strings causes a connection to host 123.196.222.160.

As an option, you can enter the Internet host number instead of the host name. Internet host numbers consist of four integers (between 0 and 255) separated by periods. This is known as dotted decimal notation. Entering 26.0.0.73 specifies the host name NIC.DDN.MIL.

Note The host parameter of the host string is required.

Specifying the Port

The port option lets you select the port number on the destination host. By default, Telnet connects to port 23 and FTP connects to port 21. If a port variable is used, it must have a leading comma to separate it from the host or <*route*> options.

The port number is an integer with a value between 1 and 65535. If the port number is greater than 999, do not use a comma within the port number. For example, enter 1023, not 1,023.

Note The port option of the host string is optional.

Example

The following command connects to port 1023 at host hostA.

hostA,1023

Specifying Internal VTAM Applications

In addition to connecting to remote hosts through TCP/IP, you can connect to applications internal to Cisco IOS for S/390. These applications are:

- **VTAMTEST**

  Invokes **ACTEST**, but instead of communicating over TCP/IP, the communication is through VTAM.

- **SNDMSG**

  Allows a user to send mail messages to other internet users.

- **STELWHO**

  Invokes a **WHOIS** client function.

- **BCAST**

  Invokes a broadcast facility to send messages to all or selective Cisco IOS for S/390 users.

Examples

Specify internal VTAM applications by entering a semi-colon (**;**) followed by the name of the application, as in the following example:

;SNDMSG

Some internal applications, when invoked, can be passed optional parameters. The first blank in the optional parameter field terminates the parameter. Optional parameters can follow the application name and are separated from it by a slash (/), as in the following example:

;stelwho/nic

Example Host Name Strings

The following example host name strings all perform the same function; each string specifies a connection to the Telnet Server at host NIC.DDN.MIL., which has an ARPANET address (host number) of 10.0.0.51:

nic.ddn.mil.	Fully qualified name
sri-nic.arpa	Alias of NIC.DDN.MIL.
10.0.0.51	Host number NIC.DDN.MIL
nic	Only if ddn.mil. is in search list.

The following examples are more complex:

nic.ddn.mil.,9	Telnet connection to port 9 (discard)
128.16.9.3<10.0.0.51>,17	Suggest gateway route; use port 17

The following are examples for using internal applications:

;sndmsg	Invoke Send Mail
;vtamtest	Invoke ACTEST using VTAM interface
;stelwho/nic	Invoke WHOIS client
;bcast	Invoke Broadcast function