

# Network Configuration

---

This chapter describes the parameters to set up your network interface. These parameters are in the TCPCFGxx member of the PARM data set. Parameters for modifying protocol configuration are also in this file, but are described in TCP, UDP, RAW and IP Protocol Configuration (TCPCFGxx).

This chapter includes these sections:

- TCP Stack Configuration Member (TCPCFGxx)  
Describes network interface and protocol-specific information
- Defining Physical Medium  
Describes how to use the MEDIA statement
- Defining IP Addressing—NETWORK Statement  
Describes how to use the MEDIA statement
- Driver Statements  
Describes how hardware statements are specified
- CDLC Statement  
Describes how to configure using the CDLC protocol
- CETI Driver Configuration  
Describes how configure to use CETI devices
- CLAW Driver Configuration  
Describes how to use the CLAW driver
- HYPERchannel Configuration  
Describes how to interface to a HYPERchannel network
- LCS Configuration Parameters  
Describes how to use a 3172 Interconnect Controller and the IBM 8232 LAN Channel Station
- IBM 3172 and 8232 Configuration  
Describes how to use the LINK statement for link level adapters such as 3172 or 8232
- ARP Configuration  
Describes how to configure for Address Resolution Protocol

## TCP Stack Configuration Member (TCPCFGxx)

The TCPCFGxx member in the PARM data set specifies the main configuration parameters for the TCP stack.

Generally, the first statement of TCPCFGxx is the MEDIA statement which defines the physical medium that Cisco IOS for S/390 will be physically attached to. The MEDIA statement is followed by one or more NETWORK statements, zero or more driver statements (CETI, CLAW, HYPER, LINK), and zero or more ARP statements.

Multiple MEDIA statements are allowed for multihomed configurations; however, each must be followed by its own NETWORK, driver and ARP statements.

The NETWORK statement defines the IP addressing of the media. If a subnets-are-local configuration is desired, multiple NETWORK statements may follow the MEDIA statement. Multiple Virtual IP Addresses are defined by specifying a MEDIA VIRTUAL statement, followed by one or more NETWORK statements. IP address 127.0.0.1 is reserved for loopback and cannot be coded.

The driver statements follow the MEDIA statement and may be before or after the NETWORK statements. Driver statements are not permitted if the Media type is Virtual. Multiple driver statements are permitted if multiplexing is desired. The LINK statement requires a prior LCS statement. The LCS statement can appear anywhere in TCPCFGxx prior to the LINK statement.

ARP statements follow the driver statements if static ARP resolution is used on this Media. No ARP statements are required if dynamic ARP is being used (recommended).

ROUTE statements may be placed anywhere after the MEDIA statement, or for simplicity, all ROUTE statements may be grouped together if the MEDIANAME keyword is specified on each ROUTE statement.

MEDIANAME links MEDIA, NETWORK, driver, ARP and ROUTE statements. If MEDIANAME is not used, statements are determined by their placement in the TCPCFGxx member. Read MEDIA Statement Usage Notes for more information about using MEDIANAME.

## Defining Physical Medium

The MEDIA statement defines a physical medium that Cisco IOS for S/390 is attached to. The MEDIA statement is followed by one or more NETWORK statements, zero or more driver statements (CETI, CLAW, HYPER, and LINK), and zero or more ARP statements.

Multiple MEDIA statements are allowed for multihomed configurations; however, each must be followed by its own NETWORK, driver, and ARP statements.

## MEDIA Statement

```
MEDIA NAME (media_name)  
      [ARPTIMEOUT (router_time host_time)]  
      [ETHERNET | VIRTUAL | TOKEN4 | TOKEN16 | FDDI | HYPERCHANNEL |  
      CLAW | CDLC]  
      [CHECKSUM | NOCHECKSUM | HOSTCKSUM | OFFLOADCKSUM |  
      NOASSIST | ASSIST]  
      [IDLENET (sec count)]  
      [MSSDEF (mss_value)]
```

---

[MSSOPT (ALWAYS | NET | NEVER | SUBNET)]

[MTU (*mtu\_size*)]

### Syntax Description

**NAME** (*media\_name*)

Specifies a name to be associated with this media. For VIRTUAL, a name of LOOPBACK may be specified.

**ARPTIMEOUT** (*router\_time host\_time*)

Specifies timeouts for the address resolution table (ARP).

*router\_time*—Specifies the number of seconds between ARP attempts for an IP router. The valid range is 1 to 255 seconds. Default: 10

*host\_time*—Specifies the time in seconds that an ARP cache entry is considered valid. The valid range is 1 to 65535 minutes. Default: 1800

**ETHERNET | VIRTUAL | TOKEN4 |  
TOKEN16 | FDDI | HYPERCHANNEL |  
CLAW | CDLC**

Specifies the type of network medium.

Default: ETHERNET

**CHECKSUM | NOCHECKSUM |  
HOSTCKSUM | OFFLOADCKSUM |  
NOASSIST | ASSIST**

Specifies whether checksumming should be performed on the host or, if it is offloaded, to devices that can perform checkless assistance.

CHECKSUM, HOSTCKSUM, and NOASSIST all mean that checksumming for TCP datagrams is performed by the host software.

NOCHECKSUM, OFFLOADCKSUM, and ASSIST all mean that checksumming for TCP packets is performed in the network interface device.

Default: CHECKSUM

**IDLENET** (*sec count*)

Specifies network outage detection thresholds. If no network activity is detected during *sec* seconds, the network is considered idle, and sampling of network components begins. *count* specifies the number of network components to sample for presence on the network. If any respond, the network is considered intact.

Network idle detection is available only for LNILCS and LNICETI configurations.

Fault tolerance must be authorized in all cases.

Default: (300 10) if fault tolerance is authorized;  
(0 0) if fault tolerance is not authorized

### **MSSDEF** (*number*)

Specifies default MTU (Maximum Transmission Unit) on which the maximum TCP segment size is based when the local network MTU cannot be used (for example, MSSOPT(NET) and destination is not on the same network). MSSDEF must not exceed the MTU for local network, so the valid range is 576:MTU. Read the MTU parameter and Maximum Transmission Unit for more information.

Default: MTU

### **MSSOPT** (**NEVER** | **SUBNET** | **NET** | **ALWAYS**)

Specifies when the TCP maximum segment size option will be sent.

**NEVER**—Never send the TCP maximum segment size option.

**SUBNET**—Send the TCP maximum segment size option only if the remote host is on the same subnet as the local host.

**NET**—Send the TCP maximum segment size option only if the remote host is on the same network as the local host. The remote host might be on a different subnet.

**ALWAYS**—Always send the TCP maximum segment size option.

Default: NET

### **MTU** (*number*)

Specifies the maximum transmission unit the destination is capable of receiving.

The range and default values vary according to media type:

**VIRTUAL**—min=576, max=65535, default=8192

**ETHERNET**—min=576, max=1500, default=1500

**4 MB Token Ring**—min=576, max=2002, default=2002

**16 MB Token Ring**—min=576, max=4352, default=4352

**FDDI**—min=576, max=4352, default=4352

**HYPERchannel**—min=576, max=65535, default=32000

**CLAW**—min=576, max=65505, default=4096

**CDLC**—min=576, max=60960, default=60960

## MEDIA Statement Usage Notes

### Using the MEDIA Statement in Fault Tolerant Networks

A typical installation requires only one each of a MEDIA statement, a NETWORK statement, and a driver statement. For fault tolerance, at a minimum, you will need multiple driver statements. If you are supporting multiple network numbers on the same physical medium, you will need multiple NETWORK statements to indicate this. A MEDIA statement should correspond to a physical medium such as Ethernet, token ring, or FDDI, and there should be one MEDIA statement for each physical medium this host is attached to.

---

By default, NETWORK statements following a MEDIA statement are associated with that media. This can be overridden with the MEDIANAME keyword, but it makes the configuration difficult to follow. It is recommended that you keep the NETWORK statements organized under their related MEDIA statement.

Similarly, driver statements are related to a MEDIA statement, and represent the physical related interfaces to the related media. By default, driver statements following a MEDIA statement are associated with that media. As for the NETWORK statements, this can be overridden with the MEDIANAME keyword, but it makes the configuration difficult to follow. It is recommended that you keep your driver statements organized under their related MEDIA statement.

There is no relationship between NETWORK and driver statements. That is, all drivers under a MEDIA statement can respond to all IP addresses for all the NETWORK statements related to that media.

## Maximum Transmission Unit

The Maximum Transmission Unit (MTU) that can be transmitted between two nodes on a local network generally is fixed for the entire network. For Ethernet, this value is 1500 data bytes excluding the local network header, and for DDN and ARPANET, this value is fixed at 1007. For HYPERchannel, there is no limit imposed by the lower-level protocols, and the remote host determines the MTU by how much it is willing to receive (see ARP statement). FDDI networks have an MTU of 4352 as well as 16 MB Token Ring networks. The 4 MB Token Ring networks have an MTU of 2002.

## Maximizing Throughput

Throughput can be maximized by sending data packets that are as large as the MTU for the local network. TCP uses the Maximum Segment Size (MSS) option to regulate the size of TCP segments transmitted on a TCP connection, which in turn determines the maximum size of packets generated by IP. Generally, sending the largest TCP segment possible is more efficient because the total number of segments transmitted is reduced. However, if the communicating hosts are not on the same network (or subnetwork), intervening gateways might need to fragment the packets, which can be less efficient than sending smaller segments in the first place.

The MSSOPT parameter controls when the local TCP uses the MSS TCP option to increase the size of TCP segments it is willing to receive (the default is the MTU). If permitted, TCP increases its receive segment size to a value that lets the remote TCP send segments that are optimal for the local network and will never transmit a segment larger than that advertised by the remote TCP.

Next to NEVER, MSSOPT(SUBNET) is the most restrictive option because both hosts must be on the same subnet. If the network is not a subnet, or all subnets comprising the local network use the same MTU, or fragmentation is not a concern, MSSOPT(NET) should be used. Otherwise, use MSSOPT(ALWAYS).

## Maximum Receive Segment Size

The MSSDEF parameter determines what the maximum receive segment size can be if the MSSOPT parameter does not permit using the optimum value for the local network. This permits using a value larger than the Internet-wide default and smaller than the MTU for the local network. For example, if several local area networks are interconnected by a wide area network whose MTU is smaller than that of the local area networks, MSSDEF should be set to the MTU of the wide area network. When used with the appropriate MSSOPT parameter, TCP can use larger segments while avoiding fragmentation.

The MSSDEF value is specified in terms of the IP packet size and not the actual TCP segment size. In other words, the value specified for MSSDEF should be the largest network packet, excluding the network header but including the normal TCP and IP headers, that can be generated with a maximum size TCP segment.

## Defining IP Addressing—NETWORK Statement

The NETWORK statement defines the IP addressing of the media. If a subnets-are-local configuration is desired, multiple NETWORK statements may follow the MEDIA statement. Multiple virtual IP addresses are defined by specifying a MEDIA VIRTUAL statement, followed by one or more NETWORK statements. IP address 127.0.0.1 is reserved for loopback and cannot be coded.

### NETWORK Statement

```

NETWORK IPADDRESS (ip_address)
    [DEST (destination)]
    [MEDIANAME (name)]
    [METRIC (metric)]
    [NETMASK (network_mask)]
    [SUBNETMASK (subnet_mask)]
    
```

#### Syntax Description

<b>IPADDRESS</b> ( <i>ip_address</i> )	Specifies the full Internet host address in standard dot notation (for example, 128.5.2.17).  Default: None
<b>DEST</b> ( <i>destination</i> )	Specifies the remote IP address for point-to-point links.  Default: None.
<b>MEDIANAME</b> ( <i>name</i> )	Specifies the MEDIA statement this network is associated with.  Default: The most recent MEDIA statement.
<b>METRIC</b> ( <i>metric</i> )	Specifies a hop count for this network, for use by routing protocols.  Default: 1
<b>NETMASK</b> ( <i>network_mask</i> )	Specifies the Network mask to use. The network mask is based on the class of the network.  Default: Class A—255.0.0.0 Class B—255.255.0.0 Class C—255.255.255.0
<b>SUBNETMASK</b> ( <i>subnet_mask</i> )	Specifies the subnet mask in standard dot notation (for example, 225.225.255.0).  Default: Network mask as above.

---

## Network Statement Usage Notes

Consider these usage notes when using the NETWORK statement:

### Local Host Name

The NETWORK statement defines the local Internet address by which remote hosts access this host through the local network.

### Subnet and Network Masks

If the local network is a subnet of a larger network, a subnet mask must be specified with the SUBNET parameter to define which bits of the Internet address contain the subnet number.

If the network is comprised of multiple Class C networks that share the same physical medium, the network mask can be shortened to make a single network.

For example, networks 192.130.0.0, 192.130.1.0, 192.130.2.0, 192.130.3.0 can be made to look like a single Class B/C network by specifying a network mask of 255.255.252.0.

### Configuring a Single LNI to Act as Multiple Hosts

Cisco IOS for S/390 supports multiple NETWORK statements bound to a single interface. When using this feature, Cisco IOS for S/390 can be configured such that a single LNI can act as multiple hosts, each with a unique IP address, on the same or different nets or subnets.

Cisco IOS for S/390 will report its associated IP address as the address specified on the first NETWORK statement using that LNI. Cisco IOS for S/390 will respond to ARPs for any NETWORK statement pointing to it, and attempt to use the correct IP address on ARPs it generates.

### Multihomed Example

```
MEDIA ETHERNET MTU(1500) NAME(HOST1)
.
.
.
NETWORK IPADDRESS(192.130.0.12)
SUBNET(255.255.255.0)
.
.
.
NETWORK IPADDRESS(192.130.1.12)
SUBNET(255.255.255.0)
.
.
.
CETI CUTYPE(3762)
DEVADDR(400)
.
.
.
```

## Multihome/Multiplex

A local network is bound to a particular hardware interface by specifying one or more driver statements. All NETWORK statements associated with that media will be associated with all drivers for that media.

Here is an example for a multiplexed environment.

```
MEDIA ETHERNET MTU(1500) NAME(ETHER)
NETWORK IPADDRESS(192.130.0.12) SUBNET(255.255.255.192)
CETI CUTYPE(3762) DEVADDR(400)
LCS NAME(LAN) DEVADDR(500) CUTYPE(3172)
LINK LCSNAME(LAN) ADAPTER(0)
```

Here is an example for a subnets-are-local and multiplexed environment:

```
MEDIA ETHERNET MTU(1500) NAME(ETHER)
NETWORK IPADDRESS(192.130.0.12) SUBNET(255.255.255.192)
NETWORK IPADDRESS(192.130.0.14) SUBNET(255.255.255.192)
CETI CUTYPE(3762) DEVADDR(400)
LCS NAME(LAN) DEVADDR(500) CUTYPE(3172)
LINK LCSNAME(LAN) ADAPTER(0)
```

Here is an example for a multihomed environment:

```
MEDIA ETHERNET MTU(1500) NAME(ETHER)
NETWORK IPADDR(192.130.0.12) SUBNET(255.255.255.192)
LCS NAME(LAN) DEVADDR(500) CUTYPE(3172)
LINK LCSNAME(LAN) ADAPTER(0)
MEDIA FDDI MTU(4352) NAME(FIBER)
NETWORK IPADDR(192.130.1.12) SUBNET(255.255.255.192)
LINK LCSNAME(LAN) ADAPTER(1)
```

If multiple drivers are specified for the local network, all of them are capable of responding to the same Internet address. In this case, the host is said to be multiplexed onto the local network. On output, packets are distributed on a round-robin basis. On input, the load sharing mechanism (if any) is determined by the network. Multiplexing can be used to increase throughput or to eliminate single points of failure by providing redundancy.

More than one local network can be defined, each with its own unique Internet address within one Cisco IOS for S/390 address space. Also, the local networks can use different transmission technologies and have different characteristics (for example, Ethernet and FDDI). In this case, the host is said to be *multihomed* because there are multiple routes to the local host.

## NETWORK Example

This example shows the usage of the NETWORK statement:

```
NETWORK IPADDRESS(26.131.0.2)
        NETMASK(255.0.0.0)
        SUBNET(0.0.0.0)
        METRIC(1)
```

## Driver Statements

The driver statements follow the NETWORK statements. Driver statements are not permitted if the MEDIA type is VIRTUAL. Multiple driver statements are permitted if multiplexing is desired. The LINK statement requires a previous LCS statement. The LCS statement can appear anywhere in TCPCFGxx prior to the LINK statement.

---

# CDLC Statement

Sending IP directly to the FEP is more efficient, easier to configure, and yields higher performance than encapsulating IP datagrams in SNA Path Information Units (PIUs). IP encapsulation is the approach used in IBM’s SNALINK.

You can use a 3745 FEP to attach channel interfaces directly to an IP network. The channel interface in this situation is configured in the Network Control Program (NCP) as a native IP element. The IP channel interface does not fall under control of VTAM.

Cisco IOS for S/390 uses the IBM CDLC protocol to implement support for the IP channel attachment capability. The 3745 FEP appears as a single device, operating in half-duplex mode. The CDLC protocol uses a single subchannel in half-duplex mode.

IP datagrams are passed from an MVS-based IP application over bus-and-tag or ESCON channels to a 3745 FEP. To Cisco IOS for S/390, the channel-attached NCP running native IP support looks like a channel-attached router.

The 3745 FEP cannot share the same bus-and-tag interface between SNA and IP traffic to the mainframe. The native IP attachment requires a dedicated 3745 channel adapter. If you are using ESCON with the 3746-900 frame, IP and SNA traffic can flow over the same channel connection if you create a separate NCP link station for the IP traffic.

# CDLC Statement

Use the CDLC Statement to configure your CDLC driver.

```
CDLC    DEVADDR (ccuu)
        [IBUF (input_buffer_count)]
        [MEDIANAME (value)]
        [OBUF (output_buffer_count)]
        [RESTART (restart_value)]
        [START | NOSTART | AUTOSTART | NOAUTOSTART]
```

## Syntax Description

DEVADDR (ccuu)	Specifies the first channel <i>cc</i> and unit number <i>uu</i> (channel address) of the specified device.  Default: None
IBUF (input_buffer_count)	Specifies the number (3:30) of input buffers. The number of buffers must be less than or equal to 255. Be aware that these buffers (20K each) are page-fixed in real storage for as long as the device is active. This will affect paging activity.  Default: 10
MEDIANAME (name)	Specifies the MEDIA statement that this driver is associated with.  Default: The most recent MEDIA statement.

**RESTART** (*restart\_time*)

Specifies the time interval, in seconds, for either of these events:

- Restart after an abnormal termination due to an I/O error.
- Sampling frequency when an unreachable device is detected during startup.

Valid values are between 5 and 65535, or 0. If this value is omitted, or if 0 is specified, the default interval will be used.

Default: 60 seconds

**OBUF** (*output\_buffer\_count*)

Specifies the number (3:30) of output buffers. The number of buffers must be less than or equal to 255. Be aware that these buffers (20K each) are page-fixed in real storage for as long as the device is active. This will affect paging activity.

Default: 10

**START | NOSTART |  
AUTOSTART | NOAUTOSTART**

Specifies whether to start the device during initialization. AUTOSTART is an alias for START. NOAUTOSTART is an alias for NOSTART.

Default: START

## CDLC Statement Usage Notes

### Example NCP Coding

This example shows the relevant portions of the NCP used for this configuration.

NCPOPT	OPTIONS	NEWDEFN= (YES, ECHO, NOSUPP)	NDF options
NCP01	BUILD	VERSION=V7R3F, . . . . CNLSQMAX=10000, CNLSQTIM=10, IPPOOL=NCP, IPRATE=(40,50) , IPSNAP=NONE, . . .	Minimum level NCP Version 7 Release 3 IP performance parameters
RINGGRP	GROUP	ECLTYPE=(PHY,ANY) , . . .	Token Ring definition
RINGLINE	LINE	ADDRESS=(1088,ANY) , INTERFACE=IPTOKEN, ADAPTER=TIC2	
RINGPU	PU	ADDR=01, NETWORK=SNA, PUTYPE=1	
RINGIP	PU	ADDR=02, ARPTAB=(1000,20,NOTCANON) , NETWORK=IP, PUTYPE=1	
NCSTGRP	GROUP	NCST=IP, . . .	NCST group
NCSTLINE	LINE		
NCSTPU	PU		
NCSTLU	LU	INTERFACE=IPOWNER, REMLU=SNALINK	
CAGRP	GROUP	LNCTL=CA, . . .	Dedicated channel adapter interface
CALINE	LINE	ADDRESS=01, CATYPE=TYPE6, . . .	
CAPU	PU	PUTYPE=1, INTERFACE=CATCP, LADDR=140.51.100.22, P2PDEST=140.51.100.20, SNETMASK=255.255.255.0, METRIC=1, PROTOCOL=NONE	
	IPOWNER	INTERFACE=IPOWNER, HOSTADDR=140.51.100.21, UDPPORT=580, NUMROUTE=(25,25,25)	
	IPLOCAL	INTERFACE=IPOWNER, LADDR=140.51.100.23, P2PDEST=140.51.100.21, SNETMASK=255.255.255.0, METRIC=1, PROTOCOL=NONE	IPLOCAL definitions for the NCST interface and the Token Ring interface
	IPLOCAL	INTERFACE=IPTOKEN, LADDR=140.51.101.64, SNETMASK=255.255.255.0, METRIC=1, PROTOCOL=NONE	

<b>IROUTE</b>	<b>INTERFACE=IPTOKEN,</b> <b>DESTADDR=140.51.102.0,</b> <b>NEXTADDR=140.51.101.65,</b> <b>METRIC=1,DISP=PERM</b>	<b>IROUTE</b> definition for a static route to a default gateway
---------------	---	--

## CETI Driver Configuration

The CETI statement specifies configuration parameters for a channel adapter that implements the CETI.

These devices are compatible with the Cisco IOS for S/390 3722/3762 High Performance Controller.

### CETI Statement

```
CETI [CUTYPE (3722 | 3762)]
    DEVADDR (ccuu)
    [IBUF (input_buffer_count)]
    [IPARM (buffer_thresh_old gap_time)]
    [MEDIANAME (name)]
    [OBUF (output_buffercount)]
    [OPARM (buffer_threshold gap_time)]
    [RESTART (restart_time)]
    [START | NOSTART | AUTOSTART | NOAUTOSTART]
    [WTIME (wait_time)]
```

#### Syntax Description

<b>CUTYPE</b> (3722   3762)	Specifies the control unit type. Default: 3762
<b>DEVADDR</b> (ccuu)	Specifies the first channel <i>cc</i> and unit number <i>uu</i> (channel address) of the specified device. Default: None
<b>IBUF</b> (input_buffer_count)	Specifies the number (3:255) of input buffers. The number of buffers must be less than or equal to 255. Be aware that these buffers (1500 bytes each) are page-fixed in real storage for as long as the device is active. This will affect paging activity. Default: 90

---

<b>IPARM</b> ( <i>buf_threshold gap_time</i> )	<p>Specifies the input data port parameters consisting of the buffer threshold (0:255) and the message gap time (IBUF-2:65535) expressed in 64 microsecond units. A message gap of 65535 indicates not to interrupt based on time. The buffer threshold must be less than or equal to the number of input buffers. The input values (IPARM) can significantly affect the throughput rate of FTPs, particularly when sending data from MVS. Recommended performance values for IPARM are (4 16), (4 0) or (0 0). The best IPARM values are really determined by the CPU type, type of CETH device, remote host capabilities and network load. Several values should be tested before settling on a specific value.</p> <p>Default: (0 0)</p>
<b>MEDIANAME</b> ( <i>name</i> )	<p>Specifies the MEDIA statement that this driver is associated with.</p> <p>Default: The most recent MEDIA statement.</p>
<b>OBUF</b> ( <i>output_buffer_count</i> )	<p>Specifies the number (3:255) of output buffers. The number of buffers must be less than or equal to 255. Be aware that these buffers (1500 bytes each) are page-fixed in real storage for as long as the device is active. This will affect paging activity.</p> <p>Default: 90</p>
<b>OPARM</b> ( <i>buf_threshold gap_time</i> )	<p>Specifies the output data port parameters consisting of the buffer threshold (0:255) and the message gap time (0:65535) expressed in 64-microsecond units.</p> <p>A message gap of 65535 indicates not to interrupt based on time. The buffer threshold must be less than or equal to the number of output buffers. The message gap should always be set to 65535 because an interrupt based on the amount of time since the last send is never required.</p> <p>Default: ((OBUF-2) 65535)</p>
<b>RESTART</b> ( <i>restart_time</i> )	<p>Specifies the time interval, in seconds, for either of these events:</p> <p>Restart after an abnormal termination due to an I/O error.</p> <p>Sampling frequency when an unreachable device is detected during startup.</p> <p>Valid values are between 5 and 65535, or 0. If this value is omitted, or if 0 is specified, the default interval will be used.</p> <p>Default: 60 seconds</p>

**START | NOSTART |  
AUTOSTART | NOAUTOSTART**

Specifies whether to start the device during initialization. AUTOSTART is an alias for START. NOAUTOSTART is an alias for NOSTART.

Default: START

**WTIME** (*wait\_time*)

Specifies the control port wait time in 64 microsecond units (0:65535). A value of 5092 (325 milliseconds) is recommended if there are other devices on the same channel to prevent those devices from being unable to transfer data over the channel. A value of 0 (0 milliseconds) is recommended if there are no other devices on the same channel to allow optimum throughput of the CETI device.

See the CETI Statement Usage Notes following this table for more information about WTIME.

Note that a value of 5092 allows the very slowest of hardware to share the channel with CETI. For performance, use the default of 512 or a smaller value that works in your environment.

Default: 1563 if CUTYPE is 9750, 512 for all other types

## CETI Statement Usage Notes

### Control Port Wait Time

The control port wait time (WTIME) controls how much of the channel bandwidth is used by the control port. If there is no activity on the data ports, the control unit pauses for the specified amount of time after every CWRITE command to allow other control units access to the channel. Increasing this value lowers channel utilization but increases response time.

The granularity of the control unit's interval timer might limit the usefulness of this parameter. For example, the Intel 9750E currently is unable to measure any interval less than 100 milliseconds. Therefore, any value of WTIME less than 1563 is treated as zero by the control unit. This results in very high channel utilization, and other control units on the channel continue to operate, although with diminished performance. Control units that require a large amount of channel bandwidth should not be configured on the same channel with a CETI control unit.

### Maximum Message Buffering

The maximum number of full-size network messages that can be buffered is 255. Be aware that these buffers are page-fixed in real storage for as long as the device is active. This will affect system paging activity.

### Interrupt Handling

The CETI control unit uses several mechanisms to limit the number of interrupts generated to the CPU. One such mechanism is to delay an attention interrupt after transfer of a message to allow subsequent messages to be transferred as a single burst. Two parameters that control the amount of delay are the buffer threshold and message gap time. The buffer threshold specifies the number of buffers that can be transferred before an attention interrupt is generated. The message gap time

---

specifies the amount of time the control unit waits after a burst of buffers before presenting an attention interrupt, even if the burst contains less than the buffer threshold. Choose the values for these parameters to balance responsiveness with high throughput.

## CETI Usage of ARP

CETI uses Address Resolution Protocol (ARP) to map higher-level protocol addresses into lower-level hardware addresses. A dynamic cache of address mappings is maintained for this purpose.

## CETI Example

This example shows the usage of the CETI statement:

```
CETI CUTYPE(3722)
    DEVADDR(0240)
    IBUF(20)
    IPARM(4 16)
    OBUF(20)
    OPARM(18 65535)
    RESTART(30)
    WTIME(128)
```

## CLAW Driver Configuration

Use the CLAW statement to specify configuration parameters for an interface to an RS/6000, Cisco 7000, or Cisco 7500 series running the Common Link Access to Workstation (CLAW) protocol.

## CLAW Protocol

CLAW is a protocol for communicating between a mainframe and a channel-attached workstation. The protocol solves two major problems:

- Lets multiple applications on the workstation communicate with multiple applications on the host over a single pair of subchannel addresses.
- Lets data be transferred over the channel with a minimal number of I/O interrupts, and does not consume the channel when idle, as CETI does.

Cisco IOS for S/390 currently is a single application (TCP/IP) and it communicates with a single application (TCP/IP) on the workstation (support for multiple applications on either host or workstation is not important for Cisco IOS for S/390 at this time). However, Cisco IOS for S/390 must implement the protocol to initiate and terminate a host application to workstation application connection. This connection process consists of these steps:

- Connect to the workstation and validate connection parameters. Both the workstation and host have eight-character names. These names are exchanged in both the system validate request and response messages. If they do not match, the connection is refused.

The validate request and response also contain the block size of data messages, which must be one of these values: 1024, 2048, 3072, or 4096 (unless PACKED is specified. See the BUFSIZE parameter).

Within Cisco IOS for S/390, the host name, workstation name, and block size are specified on the CLAW statement. On an RS/6000, use the **smit chgcat** command to set these parameters.

- Once the system validate sequence completes, application connection starts. This process includes sending and receiving connection requests, sending and receiving connection confirmations, and sending and receiving disconnects, where appropriate.

Both the connection request and confirm contain the application name, which is hard coded as TCP/IP within both Cisco IOS for S/390 and the RS/6000.

A successful connection results in a path between Cisco IOS for S/390 and the workstation. This path is assigned a link ID that is used as part of Channel Command Word (CCW) operation code for reading and writing data.

Once the connection is established, a point-to-point path exists between TCP/IP on the host and TCP/IP on the workstation. This path is a separate subunit from any attached LANs or WANs. This means the workstation acts as an IP router between any LANs and the host. Therefore, the workstation must have a routing protocol such as GateD or routed running so the rest of the world is able to determine a network path to the host.

Routing protocols such as RIP rely on the presence of inbound broadcasts on each interface to determine operability of a path. Since Cisco IOS for S/390 does not send RIP updates unless GateD is running on the host, any routing protocol on the workstation must be configured so that the point-to-point interface is configured as passive. For GateD on the workstation, gated.conf contains the *ca0 passive;* entry interface.

The maximum transmission unit over the channel is the block size that is specified on the CLAW statement. There are no hardware headers sent or received with each frame, so the entire block size can be used for TCP and IP data. Also, address resolution (ARP) does not occur because this is a point-to-point connection.

The default block size is 4096 (32678, if PACKED). However, the maximum Ethernet packet cannot exceed 1500 bytes (IP and TCP data). This leaves a lot of room in each buffer unused since the TCP/IP application protocol does not permit blocking of data within each buffer. The RS/6000 allows the size of buffers to be 2048, which is more appropriate for Ethernet. However, the Cisco 7000 and 7500 series only support 4096.

---

**Note** The RS/6000 can support only 212,992 bytes of channel buffers. This is 26 input and 26 output buffers at 4096 bytes, 34 input and 34 output buffers at 3072 bytes, 52 input and 52 output buffers at 2048 bytes, and 104 input and 104 output buffers at 1024 bytes. *Do not* specify a number larger than this. The RS/6000 may crash and fail to reboot without removing the channel card. The most efficient block size for sending TCP packets to the RS/6000 is 4096. For other hosts beyond the RS/6000, 2048 is the optimal block size. However, due to propagation delays within the RS/6000, rarely do all the host channel buffers get used. This indicates that 4096 is the better choice in a mixed environment.

---

## CLAW Statement Syntax

**CLAW DEVADDR** (*ccuu*)

[**BUFSIZE** (1024 | 2048 | 3072 | 4096 | 8192 | 12288 | 16384 | 20480 |  
24576 | 28672 | 32768 | 36864 | 40960 | 45056 | 49152 |  
53248 | 57334 | 61440 | 65535)]

[**CHARSET** (*charset*)]

[**HOSTNAME** (*hostname*)]

[**IBUF** (*bufsize each*)]

---

**[MEDIANAME (*name*)]**  
**[OBUF (*outputbuffercount*)]**  
**[PACKED | UNPACKED]**  
**[RESTART (*restarttime*)]**  
**[SINGLENOOP | DOUBLENOOP]**  
**[START | NOSTART | AUTOSTART | NOAUTOSTART]**  
**[WSNAME (*workstation\_name*)]**

## Syntax Description

<b>DEVADDR</b> ( <i>ccuu</i> )	<p>Specifies the first channel <i>cc</i> and unit number <i>uu</i> (channel address) of the specified device.</p> <p>Default: None</p>
<b>BUFSIZE</b> (1024   2048   3072   4096   8192   12288   16384   20480   24576   28672   32768   36864   40960   45056   49152   53248   57334   61440   65535)	<p>Specifies the size of each individual buffer. This parameter applies to both input and output buffers. The sizes above 4096 are valid if PACKED is specified.</p> <p>Default: 4096 (32678 if packed; sizes below 4096 are invalid if PACKED is specified.)</p>
<b>CHARSET</b> ( <i>name</i> )	<p>Specifies the name of the translation character set to be used to convert host, workstation, and application names from EBCDIC to ASCII and vice versa.</p> <p>Default: ENGLISH</p>
<b>HOSTNAME</b> ( <i>name</i> )	<p>Specifies the one to eight-character name by which the host is known as defined in the CLAW definitions on the workstation.</p> <p>Default: HOST</p>
<b>IBUF</b> ( <i>bufsize each</i> )	<p>Specifies the number of input buffers (5:256, 2:16 if PACKED is specified). These buffers are page-fixed in real storage for as long as the device is active. This will affect paging activity.</p> <p>Default: 26, 4 if PACKED is specified.</p>
<b>MEDIANAME</b> ( <i>name</i> )	<p>Specifies the MEDIA statement that this driver is associated with.</p> <p>Default: The most recent MEDIA statement.</p>
<b>OBUF</b> ( <i>bufsize each</i> )	<p>Specifies the number of output buffers (5:256, 2:16 if PACKED is specified). Be aware that these buffers (20 K each) are page-fixed in real storage for as long as the device is active. This will affect paging activity.</p> <p>Default: 26, 4 if PACKED is specified.</p>
<b>PACKED   UNPACKED</b>	<p>Specifies packing of datagrams in a buffer. This can only be specified if the device supports packing.</p> <p>Default: Unpacked</p>

<b>RESTART</b> ( <i>restarttime</i> )	<p>Specifies the time interval, in seconds, for restart after an abnormal termination due to an I/O error.</p> <p>Valid values are between 5 and 65535, or 0. If this value is omitted, or if 0 is specified, the default interval will be used.</p> <p>Default: 60 seconds</p>
<b>SINGLENOOP</b>   <b>DOUBLENOOP</b>	<p>Specifies handling of NOOP channel commands.</p> <p>DOUBLENOOP appends two NOOPs on the end of both the read and write subchannels. SINGLENOOP does not append extra NOOPs.</p> <p>Default: SINGLENOOP</p>
<b>START</b>   <b>NOSTART</b>   <b>AUTOSTART</b>   <b>NOAUTOSTART</b>	<p>Specifies whether to start the device during initialization.</p> <p>AUTOSTART is an alias for START. NOAUTOSTART is an alias for NOSTART.</p> <p>Default: START</p>
<b>WSNAME</b> ( <i>workstation_name</i> )	<p>Specifies the name of the workstation as defined in the CLAW definitions on the workstation.</p> <p>Default: PSCA</p>

## CLAW Statement Usage Notes

### Tracing the Channel Program

Because the channel program is being dynamically updated, the Generalized Trace Facility (GTF) has difficulty tracing it. Use GTF to trace I/O error problems and use TCPEEP or Component Trace to look at data.

### Subunit Considerations

The CLAW interface creates a separate subnet link between TCP/IP running on the host and TCP/IP on the workstation. This is considered to be a separate subnet. The host and the workstation are normally the only hosts on the subnet. Multiple TCP/IP address spaces may share a single channel as long as each TCP/IP address space has its own subchannel pair and if the address is defined in the workstation/router. In this case, all TCP/IP address spaces and the workstation share the subnet.

### Accessing Networks Attached to the Workstation

To access any network attached to the workstation (in other words any Ethernet, Token Ring, or FDDI ring), the workstation must be defined to Cisco IOS for S/390 as an IP router. Use the ROUTE configuration statement to define the workstation as the default router:

```
ROUTE DEST(0.0.0.0) ROUTE(a.b.c.d)
```

where *a.b.c.d* is the Internet address of the CLAW interface on the workstation.

In addition, routing must be permitted on the workstation. Use the **no -a** command to view TCP/IP parameters on an RS/6000. Use the **no -o ipforwarding=1** command to forward IP datagrams on the RS/6000.

---

## Defining a Path to the Host

To make Cisco IOS for S/390 accessible from other networks or subnets, define either a static or dynamic path to the host via the workstation.

This is an example of a static route command on a UNIX system:

```
route add 138.42.136.1
```

Where 138.42.128.234 1, 138.42.136.1 is the Internet address of the host, 138.42.128.234 is the Internet address of the workstation, 1 is the hop count.

---

**Note** Use the LAN (not the CLAW) Internet address of the workstation when defining the first hop gateway.

---

## Dynamic Routing with CLAW

In order to use dynamic routing, the CLAW workstation must be running some type of routing protocol such as RIP and HELLO. GateD (which is preferable) or routed can be run on the workstation to let the workstation announce the presence of the host.

When configuring GateD on the workstation, the CLAW interface must be defined as passive unless GateD is running on the host. In other words, if there is no gateway protocol running on the host, then the GateD on the workstation must not delete the route due to lack of routing responses from the host.

This is what a sample gated.conf file (the configuration file for GateD on the workstation) looks like:

```
interface ca0 passive;
rip supplier {
    interface ca0 noripout noripin;
};
static {
    138.42.136.1 gateway 138.42.136.1 preference 0;
};
```

## LAN Message Headers

IP packets sent and received over the CLAW interface do not have a LAN message header prepended to them as Ethernet, token ring, and FDDI have. This allows the TCP/IP MTU size to be specified up to the size defined by the BUFSIZE parameter on the CLAW statement.

## Maximizing Buffer Space

An RS/6000 supports a buffer pool of up to 212,992 bytes, which is fifty-two 4096-byte buffers. If the majority of IP traffic going over the channel is destined for hosts other than the RS/6000, and those hosts are on an Ethernet, consider reducing the block size being transferred across the channel from 4096 to 2048. This reduces the amount of wasted space in each buffer from 63 percent to 26 percent and at the same time doubles the number of buffers.

Set the block size to 2048 on both the host (BUFSIZE(2048)) and on the workstation (Transmit Buffer Size and Receive Buffer Size). This change also allows the maximum number of buffers on the workstation to be increased from 52 to 104 (52 input/52 output).

### Specifying the Number of Buffers

The number of input and output buffers defined on the host need not match the number of buffers defined on the workstation. The number of buffers required on the host vary by site. Consider these factors when determining the number of buffers:

- The I/O buffers used for CLAW are permanently page-fixed. Defining a buffer pool that is too large reduces the total amount of real storage available in your system and may increase the total paging rate of your system.
- The TCP/IP receive window size limits the amount of data that can be sent or received in a burst. For instance, a window size of 32,768 requires 9 buffers if the remote host is the CLAW workstation (32,768/4056) or 23 buffers if the remote host is attached on the Ethernet (32,768/1460).

If the majority of IP datagram traffic is comprised of small messages (in other words, not equal to the MTU), an increase in buffers allows for quicker transfer to or from the host. Consider reducing the size of the buffers.

---

**Note** Some CLAW devices allow MTU sizes to exceed the buffer size. For example, BUFSIZE(4096) and MTU(32768). While this will result in optimized I/O performance between the mainframe and the device, it will not always translate to improved network activity and may even prove to be detrimental. A 32 K size may be locally agreed on between CLAW and the mainframe, but the actual network limit will probably be substantially less, so fragmentation will result and reduce CLAW's capacity to keep up with the input of the mainframe. This, in turn, may lead to the mainframe overrunning the CLAW device, and excessive retransmissions will negate any I/O improvements.

---

### CLAW Example

This example shows the usage of the CLAW statement:

```
CLAW DEVADDR (0FA0)
    BUFSIZE (4096)
    HOSTNAME (IBMHOST)
    IBUF (26)
    OBUF (26)
    RESTART (120)
    WSNAME (CLAWWS)
```

## HYPERchannel Configuration

Use the HYPER statement to specify configuration parameters for an interface to a 50 MB HYPERchannel network using N220 or EN642 adapters manufactured by Network Systems Corporation.

### HYPER Statement

This is the syntax for the HYPER statement.

```
HYPER DEVADDR (ccuu)
    [CUTYPE (N220 | EN642)]
    [IBUF (input_buffer_count)]
```

---

[**MEDIANAME** (*name*)]  
[**OBUF** (*output\_buffer\_count*)]  
[**RESTART** (*restart\_time*)]  
[**START** | **NOSTART** | **AUTOSTART** | **NOAUTOSTART**]

### Syntax Description

<b>DEVADDR</b> ( <i>ccuu</i> )	<p>Specifies the first channel <i>cc</i> and unit number <i>uu</i> (channel address) of the specified device.</p> <p>Default: None</p>
<b>CUTYPE</b> (N220   EN642)	<p>Specifies control unit type.</p> <p>Default: None</p>
<b>IBUF</b> ( <i>input_buffer_count</i> )	<p>Specifies number of input buffers (2:30). Be aware that these buffers (20 K each) are page-fixed in real storage for as long as the device is active. This will affect paging activity.</p> <p>Default: 6</p>
<b>MEDIANAME</b> ( <i>name</i> )	<p>Specifies the MEDIA statement that this driver is associated with.</p> <p>Default: The most recent MEDIA statement.</p>
<b>OBUF</b> ( <i>output_buffer_count</i> )	<p>Specifies number of output buffers (2:30). Be aware that these buffers (20 K each) are page-fixed in real storage for as long as the device is active. This will affect paging activity.</p> <p>Default: 6</p>
<b>RESTART</b> ( <i>restart_time</i> )	<p>Specifies the time interval, in seconds, for either of these events:</p> <p>Restart after an abnormal termination due to an I/O error.</p> <p>Sampling frequency when an unreachable device is detected during startup.</p> <p>Valid values are between 5 and 65535, or 0. If this value is omitted, or if 0 is specified, the default interval will be used.</p> <p>Default: 60 seconds</p>
<b>START</b>   <b>NOSTART</b>   <b>AUTOSTART</b>   <b>NOAUTOSTART</b>	<p>Specifies whether to start the device during initialization. AUTOSTART is an alias for START. NOAUTOSTART is an alias for NOSTART.</p> <p>Default: START</p>

## HYPER Statement Usage Notes

### ARP Usage

The HYPERchannel LNI does *not* use Address Resolution Protocol (ARP) to map higher-level protocol addresses into lower-level hardware addresses. A static table of address mappings is maintained for this purpose.

The address resolution cache is initialized from a static address resolution table during LNI startup. This table is generated with the ARPTABLE and ARP statements. Static entries in the ARP cache are never removed and never updated. If ARPTABLE has been specified, no action is taken and only hosts defined with ARP statements can be accessed.

## HYPER Example

This example shows the usage of the HYPER statement:

```
HYPER CUTYPE (N220)
      DEVADDR (0400)
      IBUF (20)
      OBUF (12)
      RESTART (60)
```

## LCS Configuration Parameters

The LCS statement in the TCPCFGxx member of the PARM data set specifies LNI configuration parameters for channel attached devices that are compatible with the 3172 Interconnect Controller and the IBM 8232 LAN Channel Station.

These devices are compatible with this interface:

- IBM 3172 Interconnect Controller
- IBM OSA (Open Systems Adapter, Integrated LAN Interface Adapter)
- BTI ELC2 executing 8232 emulation microcode
- Cisco IOS for S/390 3762 FCA (FDDI) and TCA (Token Ring), and ECA (Ethernet) running 8232 emulation
- IBM 2216 NWAYS Multiaccess Connector

Each of these devices supports one or more pairs of device addresses. Each pair of device addresses is represented by an LCS statement. One or more network adapters are associated with each pair of device addresses. Each network adapter is represented in TCPCFGxx with a LINK statement. The NAME operand on the LCS is referenced by the LINK statement to activate the device and network adapters.

## LCS Statement

```
LCS NAME (lcs_device_name)
      DEVADDR (ccuu)
      [CUTYPE (type)]
```

---

[IBUF (*input\_buffer\_count*)]  
[OBUF (*output\_buffer\_count*)]  
[RESTART (*restart\_time*)]

Syntax Description

<b>NAME</b> ( <i>lcs_device_name</i> )	Specifies the name of this device as referenced by the LINK statement, LCSNAME keyword.
<b>DEVADDR</b> ( <i>ccuu</i> )	Specifies the first channel <i>cc</i> and unit number <i>uu</i> (channel address) of the specified device.
<b>CUTYPE</b> ( <i>type</i> )	Specifies the control unit type. This is used only for visual documentation. Type can be any of the following:  3172, 3172-1, 3172-3, 3722, 3762, OSA  Default: None
<b>IBUF</b> ( <i>input_buffer_count</i> )	Specifies the number of input buffers (2:256) for the device. Be aware that these buffers (20 K each) are page-fixed in real storage for as long as the device is active. This will affect paging activity.  Default: 4
<b>OBUF</b> ( <i>output_buffer_count</i> )	Specifies the number of output buffers (2:256) for the device.  Be aware that these buffers (20 K each) are page-fixed in real storage for as long as the device is active. This will affect paging activity.  Default: 4
<b>RESTART</b> ( <i>restart_time</i> )	Specifies the time interval, in seconds, for restart after an abnormal termination due to an I/O error.  Valid values are between 5 and 65535, or 0. If this value is omitted, or if 0 is specified, the default interval will be used.  Default: 60 seconds

## IBM 3172 and 8232 Configuration

Use the LINK statement to define the link level network adapter as IBM 3172 or 8232 and to associate the link level with the LCS device level specification.

### LINK Statement Syntax

**LINK** LCSNAME (*lcs\_device\_name*)  
    [ADAPTER (*number*)]  
    [ALLRT | SINGLERT]  
    [LOCALADDR | NOLOCALADDR]

[**MEDIANAME** (*name*)]

[**PFILTER** | **NOPFILTER**]

[**START** | **NOSTART** | **AUTOSTART** | **NOAUTOSTART**]

### Syntax Description

<b>LCSNAME</b> ( <i>lcs_device_name</i> )	<p>Specifies the name of this LNI as referenced by a previous LCS statement.</p> <p>Default: None</p>
<b>ADAPTER</b> ( <i>number</i> )	<p>Specifies the 3172 adapter for this interface (0:3).</p> <p>Default: 0</p>
<b>ALLRT</b>   <b>SINGLERT</b>	<p>Defines how to route broadcast ARPs for token ring adapters.</p> <p>ALLRT—Use all route broadcasts on ARPs.</p> <p>SINGLERT—Use single route broadcasts on ARPs.</p> <p>Default: ALLRT—Use all route broadcasts on ARPs.</p>
<b>LOCALADDR</b>   <b>NOLOCALADDR</b>	<p>Specifies how to initialize the local hardware address of the driver. LOCALADDR sends a SETADAPTER command to reset the hardware address to its default. NOLOCALADDR suppresses the SETADAPTER command. Some devices do not support SETADAPTER and will fail during startup. Consult with the manufacturer for information about the SETADAPTER command.</p> <p>Default: NOLOCALADDR</p>
<b>MEDIANAME</b> ( <i>name</i> )	<p>Specifies the MEDIA statement that this driver is associated with.</p> <p>Default: The most recent MEDIA statement.</p>
<b>PFILTER</b>   <b>NOPFILTER</b>	<p>Specifies whether to send a PFILTER command to reset hardware filters during startup. Some devices require this command and some devices do not support PFILTER commands and will fail during startup. Consult with the manufacturer for more information about filtering commands.</p> <p>Default: PFILTER</p>
<b>START</b>   <b>NOSTART</b>   <b>AUTOSTART</b>   <b>NOAUTOSTART</b>	<p>Specifies whether to start the device during initialization. AUTOSTART is an alias for START. NOAUTOSTART is an alias for NOSTART.</p> <p>Default: START</p>

---

## LINK Statement Usage Notes

### Statement Order

The LINK statement references a previous LCS statement. The LCS statement must precede the LINK statement.

### Adapter Types

The 3172 supports up to 4 adapters, which may be of different types. The following types are supported:

- ETHERNET 10MBps Ethernet—MTU(1500)
- TOKEN4 4MBps Token Ring—MTU(2002)
- TOKEN16 16MBps Token Ring—MTU(4352)
- FDDI 100MBps Token Ring—MTU(4352)

### ARP Considerations

When Cisco IOS for S/390 needs to broadcast an addressing resolution packet on a token ring network, there are two methods for determining how the packet traverses token ring bridges. When SINGLERT is specified, the ARP is transmitted using a Single Route Broadcast. When ALLRT is specified, the ARP is transmitted using an All Route Broadcast. SINGLERT should be specified only if the token ring bridges are configured to forward packets with the Single Route indicator set. Setting SINGLERT otherwise may cause portions of the token ring network to be unreachable.

## LINK Example

This example shows the usage of the LINK statement and its associated LCS statement:

```
LCS NAME(ETHER) DEVADDR(500)
LINK LCSNAME(ELINK)
    ADAPTER(0) MEDIANAME(ETHER)
```

## ARP Configuration

Address resolution protocol is used to dynamically discover hardware addresses corresponding to IP addresses. This information is stored in an ARP table and aged and refreshed as needed. This protocol is normally used over Ethernet, token ring, and FDDI. It is therefore unusual for these media to need static ARP information. If desired, static ARP entries may be created by coding ARP statements associated with a MEDIA statement.

For HYPERchannel, ARP protocol is no longer supported.

## ARP Statement Syntax

```
ARP PA (internet_protocol_address)
    HA (hardware_address)
    [MAC (hardware_address)]
    [MEDIANAME (name)]
```

### Syntax Description

<b>PA</b> ( <i>internet_protocol_address</i> )	Specifies the Internet protocol address expressed in standard dot notation (for example, 192.34.100.44).  Default: None
<b>HA</b> ( <i>hardware_address</i> )	Specifies the physical hardware address expressed in standard dot notation (read Specifying Addresses).  Default: None
<b>MAC</b> ( <i>hardware_address</i> )	MAC is an alias for HA.
<b>MEDIANAME</b> ( <i>name</i> )	Specifies the MEDIA statement that this driver is associated with.  Default: The most recent MEDIA statement.

## ARP Statement Usage Notes

### Specifying Addresses

Standard dot notation is used to specify protocol and hardware addresses where each byte of the address is coded as a decimal number (0:255) and separated from each other by periods.

The specification of the physical hardware address must be consistent with the type of hardware specification on the ARPTABLE statement that begins the table. The hardware address for Ethernet is 48 bits (6 bytes) long, and the hardware address for HYPERchannel is 32 bits (4 bytes) long. The 16-bit HYPERchannel addresses should be coded as 32-bit addresses by putting two bytes of zero in front of the address.

### Subnet Numbers

The MEDIANAME will reference a MEDIA statement and its associated statements. The network and subnetwork numbers contained in the protocol address in the ARP statement must be consistent with the local Internet address defined with those NETWORK statements.

## ARP Examples

This example shows the usage of the ARP statement:

```
ARP PA(27.132.0.1) HA(0.0.17.0)
ARP PA(27.132.0.2) HA(0.0.34.0)
```