Release Notes Instructions

The following new functions are documented in these Release Notes for PTF01 for Version 3.2 of the IBM 2216.

NOTE: Customers using new function supported by this PTF01 for Version 3.2 need to upgrade their Configuration Program. The new Configuration Program is available on the PTF01 CD-ROM or on the Internet.

- ESCON Channel Enhancements
- 3172 Emulation Using LCS Passthru
- TN3270E Enhancements

  The following TN3270E enhancements are available with this PTF. Documentation for TN3270E Enhancements is contained in the Version 3.2 IBM 2216 software documentation. TN3270E enhancements include:

  – TN3270E LU Pooling

    LU pooling is an enhancement to the TN3270E Server function that makes it easier to configure some TN3270E Server networks. This function allows SNA LUs to be grouped into named "pools". TN3270E clients can then request a connection using the pool's name as an LU name. The TN3270E Server will then choose an LU from the specified pool to service the client's request.

  – TN3270E Server Client IP Address to LU Name Mapping

    The TN3270E Server Client IP Address to LU Name Mapping function provides a mechanism for administrators to control client access to the TN3270E Server's resources (ie LUs).

    Mapping enhances central administration by allowing the administrator to configure which SNA resources(LUs/Pool) client IP address/subnets will map to and use without modifying client configurations.

  – TN3270E Server and DDDLU

    The dynamic definition of dependent LUs (DDDLU) is a VTAM facility that allows the logical units to be known by VTAM when they connect to VTAM, rather than during the major node activation of the related PU.

    If prompted by VTAM, the TN3270E Server function will use DDDLU to create its local LUs in VTAM. Instead of sending all of the Reply PSID's when the ACTPU is received, the server will wait until the LU actually needs to be defined. The LU definition will occur when a TN3270 client connects in and needs an LU that has not been defined to VTAM.

  – TN3270E Support for Multiple TCP ports.

    This enhancement allows you to define multiple TCP ports for the TN3270E Server to "listen" on. This support allows clients to specify the SNA resource they want using a port number.

  – Ability to Disable TN3270E Negotiation.

    This enhancement allows you to specify whether the added port will negotiate to be a TN3270E server. If it is not an “E” Server, it will not support printing or system requests.

The following functional enhancement is included in this PTF:

- Changes to TFTP

The following documentation updates are included in this PTF:

- Configuring and Monitoring Layer 2 Tunneling Protocol
- Using the 10/100 Mbps Ethernet Interface
- Configuring and Monitoring the 10/100 Mbps Ethernet Interface

Escon Channel Enhancements

The following topics contain changes to the current configuration documentation to support this enhancement. The changes for this enhancement occur in Chapters 30 and 31 of the *Nways Multiprotocol Access Services Software User's Guide*. Changes to the existing text are noted in underlined type. The headings in bold type match the headings found in the printed publication and in online versions of the publication.

Channel Adapter Overview

The ESCON Channel Adapter provides the 2216 with access to SNA and TCP/IP host applications from LANs, WANs, and ATM.

Figure 1 shows a 2216 connected to a VTAM host through a channel adapter. Each ESCON channel adapter provides up to 64 subchannels and up to 32 associated virtual network handlers that can support LAN Channel Station (LCS), Link Services Architecture (LSA), and Multi-Path Channel (MPC+) protocols. Each 2216 can contain up to four channel adapters.

Each ESCON Channel Adapter can provide connections to up to:

- 32 hosts or logical host images using LCS or MPC+ (when used with an ESCON Director)
- 64 hosts or logical host images using LSA (when used with an ESCON Director)
- 15 logical host images in EMIF-capable processors in logically partition mode when no ESCON director is present.

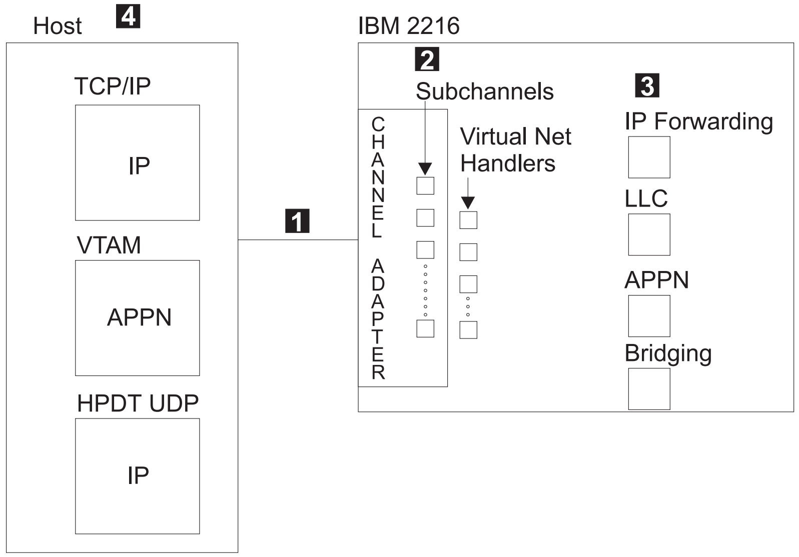

Figure 1. 2216 Connected to a Host through an ESCON Channel Adapter - Logical View

1 At the physical level, the ESCON Channel Adapter provides a flexible fiber optic connection to communication channels at the host processor.

2 At the logical level, the ESCON channel adapter provides up to 64 *subchannels* and up to 32 associated *virtual network handlers*.

Each virtual network handler supports one of the following protocols:

LCS
LAN Channel Station

LSA
Link Services Architecture

MPC+
Multi-Path Channel+

For each LCS virtual network handler, you must define two subchannels, one for read and one for write; you can define up to 32 LCS virtual network handlers for each channel adapter.

For each LSA virtual network handler, you must define at least one subchannel up to a maximum of 64 subchannels. You can define up to 32 LSA virtual network handlers for each channel adapter.

For MPC+, you can define up to 64 subchannels. You must have at least one read subchannel and at least one write subchannel. You can define up to 32 MPC+ virtual network handlers for each channel adapter.

Notes:

1. You can configure LCS and LSA on the same channel adapter.
2. For migration purposes, MPC+ may be configured on the same channel adapter as LCS/LSA. This is not recommended as a long term solution. MPC+ combined with another type of virtual interface (LCS/LSA) on the same adapter could impact the performance benefits provided by the MPC+ interface.

3 The 2216 channel adapter provides services for IP Forwarding, Logical Link Control (LLC), Advanced Peer-to-Peer Networking (APPN), and bridging.

4 The virtual net handlers provide connections for transmitting and receiving packets of information for host applications as shown in Figure 2 on page 5 and Figure 3 on page 5.

Once the channel adapter is installed and configured for LCS, LSA, and MPC+, it can provide:

- Hierarchical SNA, including DLSw traffic, and APPN ISR and HPR traffic running over LSA connections. (DLSw and APPN require LLC loopback.)
- TCP/IP traffic running over LCS and MPC+.
- APPN HPR traffic running over MPC+.
- HPDT UDP traffic running over MPC+.

Note: UDP+ is not supported on a Parallel Channel Adapter (PCA).

Figure 2 on page 5 shows the basic flow for a channel adapter with LCS and LSA configured, and Figure 3 on page 5 shows the basic flow for a channel adapter for which MPC+ is configured.

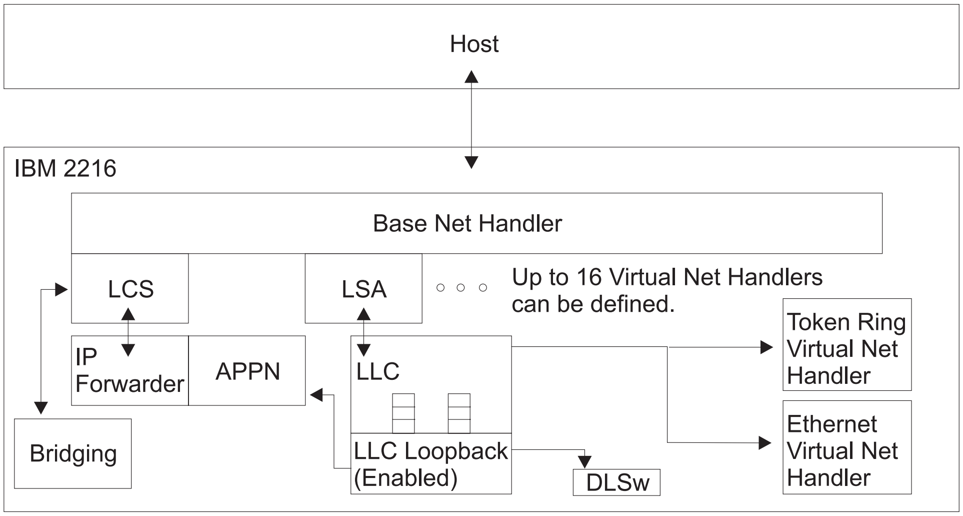
Host
IBM 2216
Base Net Handler
LCS
LSA
○ ○ ○
Up to 16 Virtual Net Handlers can be defined.
Token Ring Virtual Net Handler
IP Forwarder
APPN
LLC
Bridging
LLC Loopback (Enabled)
DLSw
Ethernet Virtual Net Handler

Figure 2. 2216 Virtual Net Handlers for LCS and LSA

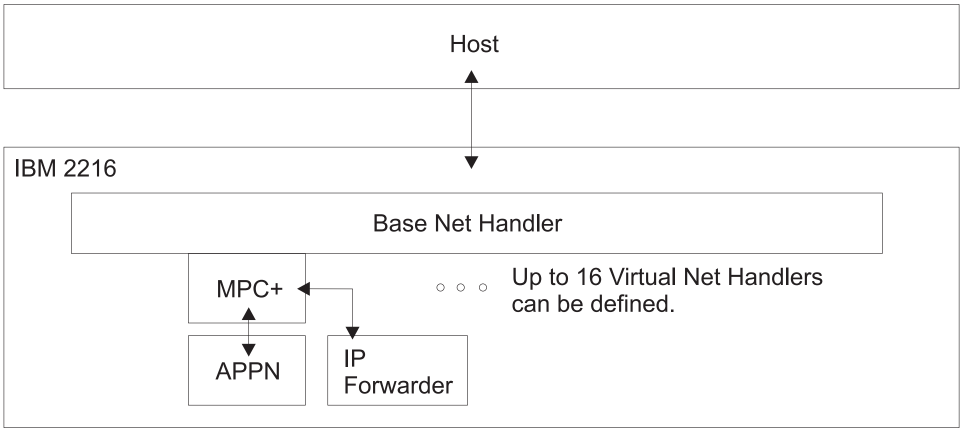

Figure 3. 2216 Virtual Net Handlers for MPC+

Example IOCP Definition for the ESCON Channel

Figure 4 on page 6 shows an example of an ESCON configuration. The S/390 host is divided into two logical partitions (LP): LPA and LPB. A connection on path 30 is configured between LPA and 2216A through ESCD switch 00. LPA is attached to ESCD port C0 and 2216A is attached to port C1. The connection between port C0 and C1 is dynamic.

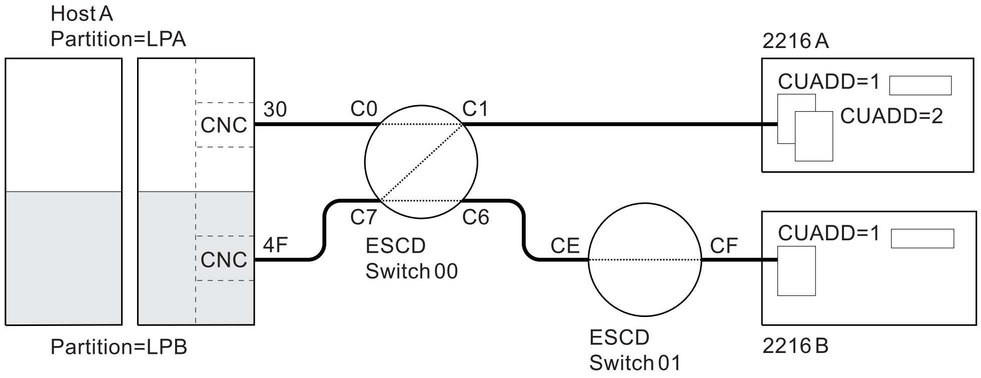

Figure 4. ESCON Channel Configuration Example

LPB on path 4F has a connection with 2216A through ESCD switch 00, and a connection with 2216B through ESCD switches 00 and 01. The connection between ports C7 and C6 is dynamic; the connection between ESCD ports CE and CF is dedicated.

The following example definitions match Figure 4:

Channel path definitions:

```
CHPID      PATH=((30)),TYPE=CNC,PART=(LPA),SWITCH=00
CHPID      PATH=((4F)),TYPE=CNC,PART=(LPB),SWITCH=00
```

Control unit and device definition for the 2216, with logical addressing = 1 for 2216A:

```
CNTLUNIT   CUNUMBR=500,PATH=30,UNIT=3172,LINK=C1,         X
           UNITADD=(00,32),CUADD=1
IODEVICE   ADDRESS=(500,32),CUNUMBR=500,UNIT=3172,        X
           UNITADD=00
```

Control unit and device definition for the 2216 with logical addressing = 2 for 2216A:

```
CNTLUNIT   CUNUMBR=600,PATH=4F,UNIT=3172,LINK=C1,        X
           UNITADD=(00,32),CUADD=2
IODEVICE   ADDRESS=(600,32),CUNUMBR=600,UNIT=3172,       X
           UNITADD=00
```

Control unit and device definition for the 2216, with logical addressing = 1 for 2216B:

```
CNTLUNIT   CUNUMBR=620,PATH=4F,UNIT=3172,LINK=C6,         X
           UNITADD=(20,32),CUADD=1
IODEVICE   ADDRESS=(620,32),CUNUMBR=620,UNIT=3172,        X
           UNITADD=20
```

The IOCP macroinstructions in the example:

- Assign a CHPID to logical partitions LPA and LPB.
- Define channel path 30 to the 2216 for partition LPA and channel path 4F for partition LPB.
- Identify channel type as an ESCON channel (CNC).
- Assign the two CHPIDs to ESCD switch number 00.
- Associate control unit numbers 500 and 600 to logical addresses 1 and 2 on 2216A and control unit number 620 to logical address 1 on 2216B.

- Assign link address C1 to control units 500 and 600 and link address C6 to control unit 620.
- Define unit addresses (subchannels) 00 through 1F to control units 500 and 600 and unit addresses 20 through 3F to control unit 620.
- Identify each control unit as a 3172 device.

Considerations:

1. The allowable device address range is 00 through FF. The 2216 address range is limited to 64 addresses, and only requires that the addresses defined at the host computer map to the address or addresses configured in the 2216. The address range can extend beyond the addresses actually used, but cannot overlap addresses of other control units cabled to the same CHPID or channel.
2. The ESCON channel mode of operation can be type CNC for basic ESCON channel mode or CVC if there is an ESCON Converter attached.
3. The IODEVICE UNIT parameter should be set to 3172.
4. The LINK number specifies the link address (ESCD port number) to which the 2216 is connected. When two ESCDs are connected in series, the link address must be the port number of the ESCD that has the dynamic connection and to which the 2216 is attached.

Example IOCP Definition for the EMIF Host

Figure 5 shows an example of an ESCON configuration using the ESCON Multiple Image Facility (EMIF). The S/390 host is divided into two logical partitions (LP): LPA and LPB. Both LPA and LPB are connected on path 30 to 2216 A through switch 00.

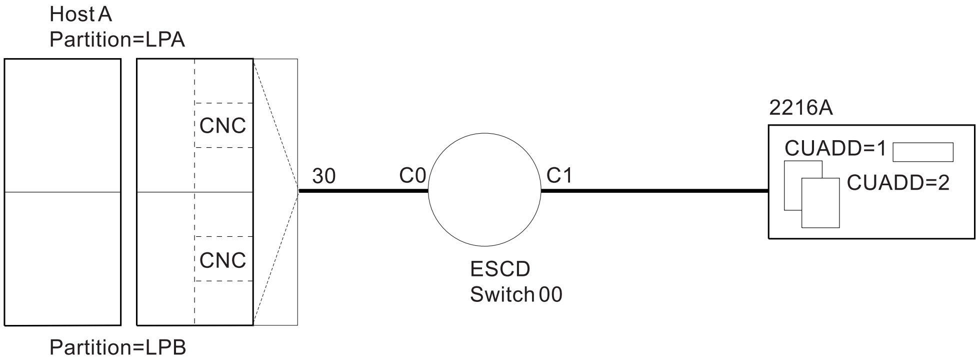

Figure 5. EMIF Host Configuration Example

The following example definitions match Figure 5:

Channel path definitions:

```
CHPID      PATH=((30)),TYPE=CNC,PART=(LPA,LPB),SWITCH=00
```

Control unit and device definition for the 2216, with logical addressing = 1 for 2216A:

```
CNTLUNIT   CUNUMBR=500,PATH=30,UNIT=3172,LINK=C1,         X
           UNITADD=(00,32),CUADD=1
IODEVICE   ADDRESS=(500,32),CUNUMBR=500,UNIT=3172,        X
           UNITADD=00
```

Control unit and device definition for the 2216, with logical addressing = 2 for 2216A:

```
CNTLUNIT   CUNUMBR=620,PATH=30,UNIT=3172,LINK=C1,         X
           UNITADD=(20,32),CUADD=2
IODEVICE   ADDRESS=(620,32),CUNUMBR=620,UNIT=3172,        X
           UNITADD=20
```

The IOCP macroinstructions in the example:

- Assign a CHPID to logical partitions LPA and LPB
- Define channel path 30 to the 2216 to be shared by partition LPA and partition LPB.
- Identify channel type as an ESCON channel (CNC)
- Assign the CHPID to ESCD switch number 00
- Associate control unit numbers 500 to logical address 1 and 620 to logical address 2 on 2216A
- Assign link address C1 to control units 500 and 620
- Define unit addresses (subchannels) 00 through 1F to control unit 500 and 20 through 3F to control unit 620
- Identify each control unit as a 3172 device.

Considerations:

1. The allowable device address range is 00 through FF. The 2216 address range is limited to 64 addresses, and only requires that the addresses defined at the host computer map to the address or addresses configured in the 2216. The address range can extend beyond the addresses actually used for the 2216, but cannot overlap addresses of other control units cabled to the same CHPID or channel.
2. The ESCON channel mode of operation can be type CNC for basic ESCON channel mode or CVC if there is an ESCON Converter attached.
3. The IODEVICE UNIT parameter should be set to 3172.
4. The LINK number specifies the link address (ESCD port number) to which the 2216 is connected. When two ESCDs are connected in series, the link address must be the port number of the ESCD that has the dynamic connection and to which the 2216 is attached.

3172 Emulation Using LCS Passthru

LCS Passthru allows the IBM 2216 to function as a 3172 replacement in TCP/IP networks. Frames received from a TCP/IP host are passed directly to a downstream LAN adapter, bypassing the IP router and bridging functions of the IBM 2216. IP and ARP frames received by a LAN adapter associated with the LCS passthru function are passed directly to the LCS net handler for delivery to the TCP/IP host. The IBM 2216 replaces the 3172 LCS function without requiring changes in IP network topology or adding additional bridge hops.

The following example describes LCS Passthru configuration:

```
*t 6
Gateway user configuration
Config>add dev esc
Device Slot #(1-8) [1]?3
Adding ESCON Channel device in slot 3  port 1 as interface #4
Use "net 4" to configure ESCON Channel parameters
Config>net 4
ESCON Config>add lcs
ESCON Add Virtual>?
LANtype
MAC address
MAXdata
BLKtimer
ACKlen
SUBchannels
ENable 3172 Emulation
Exit
ESCON Add Virtual>enable
Enabling LCS 3172 Emulation for network 5.
Please set the Network link using the "NET" command.
ESCON Add Virtual>?
BLKtimer
ACKlen
SUBchannels
DISable 3172 Emulation
NET link
Exit
ESCON Add Virtual>net 0
ESCON Add Virtual>sub add
Please add or configure one subchannel for an LCS virtual interface.
Although LCS requires two subchannels, it is only necessary to specify
one subchannel.  An adjacent subchannel will be chosen such that the two
subchannels will form a sequential pair with the write subchannel (device
address is even) before the read subchannel (device address is odd).
ESCON Config LCS Subchannel>?
LINk address (ESCD Port)
LPAR number
CU logical address
Device address
Exit
ESCON Config LCS Subchannel>link f7
ESCON Config LCS Subchannel>lpar 0
ESCON Config LCS Subchannel>cu 0
ESCON Config LCS Subchannel>dev 20
ESCON Config LCS Subchannel>ex
ESCON Add Virtual>ex
ESCON Config>list
Net:  5   Protocol: LCS    LAN type: Token Ring      LAN number:  0
          3172 Emulation is enabled.
          MAC address: Obtained from net 0
          Block Timer:    5 ms   ACK length:   10 bytes
ESCON Config>list all
Net:  5   Protocol: LCS    LAN type: Token Ring      LAN number:  0
          3172 Emulation is enabled.
          MAC address: Obtained from net 0
          Block Timer:    5 ms   ACK length:   10 bytes
          Read Subchannels:
          Sub  0   Dev addr: 21  LPAR: 0  Link addr: F7  CU addr: 0
          Write Subchannels:
          Sub  1   Dev addr: 20  LPAR: 0  Link addr: F7  CU addr: 0

ESCON Config>
```

List example from the `t 5` prompt with 3172 Emulation enabled

```
LCS> list all

LCS Virtual Adapter
LCS Information for Net 5
--- ----------- --- --- --
LAN Type: Token-Ring           LAN Number: 0
Local Read Subchannel number:  1
Local Write Subchannel number: 0
MAC Address: 08005AFE0144
LCS 3172 Emulation to net 0
Status: Down
```

L2TP Documentation Changes

The following text provides information about the commands available at the L2T interface configuration (`L2TP n>`) command prompt and which parameters can be set using the **set** command at the `Layer-2-Tunneling Config>` command prompt.

Accessing the L2TP Interface Configuration Prompt

The **add dev layer-2-tunneling** command must be followed by a **net** *x* command where the variable *x* represents the number of the L2TP interface on the router, for example 8. In this example, the command prompt becomes `L2TP 8>`.

To access the L2T interface configuration prompt:

1. Enter **talk 6** at the OPCON (*) prompt.
2. Enter **add dev layer-2-tunneling** at the `Config>` prompt.
3. Enter **n** *interface#* at the `Config>` prompt.

L2TP Interface Configuration Commands

Table 1 summarizes the L2TP interface configuration commands. Enter these commands at the `L2TP n>` prompt, where *n* is the interface number.

Table 1. L2TP Interface Configuration Commands

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See FORM='FULL'..
Disable	Disables interface attributes.
Enable	Enables interface attributes.
List	Displays information about the L2TP interface.
Set	Allows you to set various L2TP interface parameters.
Exit	Returns you to the previous command level. See FORM='FULL'..

Disable

Use the **disable** command to disable outbound calls from the L2TP access concentrator (LAC).

Syntax: disable outbound-calls-from-lac

outbound-calls-from-lac
Prevents a LAC from placing a call to initiate an L2TP tunnel.

Enable

Use the **enable** command to disable outbound calls from the L2TP access concentrator (LAC).

Syntax:

enable outbound-calls-from-lac

outbound-calls-from-lac
Allows a LAC to place a call to initiate an L2 tunnel. The software prompts you for session parameters.

Example:

```
L2TP 8> enable outbound-call-from-lac
        Outbound Call Type (ISDN, V34)? [ISDN]
        Outbound calling address: 1234
        Outbound calling subaddress:
L2TP 8>
```

List

Use the **list** command to display the state of the various L2T interface configuration parameters.

Syntax: list

```
L2TP 8>list
CONNECTION TYPE
---------- -----
  Connection Direction            INBOUND
  Remote Tunnel Hostname           *ANY*
```

Set

Use the set command to configure the L2T interface operational parameters.

Syntax: set any-remote-hostname
connection-direction
idle
remote-hostname

any-remote-hostname
Clears the outbound remote hostname and disables inbound remote host name matching on this net.

connection-direction [inbound] or [outbound] or [both]
Specifies whether the connection can be initiated by the peer (inbound), the LAC (outbound) or either the peer or the LAC (both) on this net. If you specify both, you cannot specify 0 for the idle time.

Default value: inbound

idle-time *seconds*
Specifies the number of seconds of inactivity after which L2 tunneling will disconnect the tunnel on this net. A value of 0 indicates that the tunnel is fixed and should not be disconnected.

Valid values: 0 to 1024

Default value: 0

remote-hostname *hostname*
Specifies the remote-hostname used on this tunnel.

For an outbound tunnel, the hostname is sent to the peer when placing a call. The peer uses this hostname to determine whether the call should be completed. This hostname must be configured in the authentication subsystem for calls to complete successfully. See . for more information.

For an inbound tunnel, the hostname is used to verify whether a call received from a peer on this tunnel should be completed.

Valid values: Any name from 1 to 64 ASCII characters

Default value: None

Accessing the L2 Tunneling Configuration Prompt

The following **set** command is available at the `Layer-2-Tunneling Config>` command prompt.

To access the Layer-2- tunneling configuration command prompt:

1. Enter **talk 6** at the OPCON (*) prompt.
2. Enter **feature layer-2-tunneling** at the `Config>` prompt.

Set

Use the set command to configure the L2 tunneling operational parameters.

Syntax: set buffers
call-rcv-window
max-calls
max-tunnels
transmit-retries
tunnel-rcv-window

Descriptions of these parameters are in the IBM 2216 V3.2 documentation.

10/100 Mbps Ethernet

Using the 10/100 Mbps Ethernet Network Interface

This chapter describes how to use the 10/100 Mbps Ethernet interface. It includes the following section:

- "Displaying 10/100 Mbps Ethernet Statistics"
- "Auto-negotiation on the 10/100 Mbps Ethernet Interface" on page 21
- "Configuring Values Other than Auto for Duplex" on page 21
- "Configurations That May Result in Link Activation Failure on the IBM 2216" on page 21
- "Configurations That May Result in Mismatched Duplex Modes During Operation" on page 22

Displaying 10/100 Mbps Ethernet Statistics

You can use the **interface** command from the GWCON environment to display the following statistics.

```
+i 0
                                             Self-Test  Self-Test Maintenance
Net   Net'   Interface   Slot-Port              Passed     Failed      Failed
0     0      Eth/0       Slot: 1   Port: 1           1          0         0

  Ethernet/IEEE 802.3 MAC/data-link on 10/100 Ethernet interface

  Physical address         002035030008
  PROM address             002035030008
  Actual address           002035030008
  Adapter Level            0
  Configured Duplex:       Auto-Negotiation
  Actual Duplex:           Half Duplex***
  Configured Speed:        Auto-Negotiation
  Actual Speed:            100 Mbps

  Input statistics:
   failed, packet too long        0   failed, CRC error            0
   failed, alignment error        0   failed, receive overflow     0
  *receive collision              0  *missed  frame                0
 **frames  filtered               0   receive underrun             0

   Output statistics:
   one retry                      0   single collision             0
   multiple collisions            0   failed, transmit underflow   0
   failed, excess collisions      0   failed, loss of carrier      0
   late collisions                0   more than one retry          0
   buffer error                   0   total collisions             0
   excessive deferral             0   deferred                     0
   memory error                   0

*  cannot be cleared.
** cleared automatically when read.
*** CAUTION: MAY BE DIFFERENT ON THE SWITCH/HUB PORT.
```

These statistics have the following meaning:

Nt

Global network number.

Nt'

This field is for the serial interface card. Disregard the output.

Interface
Interface name and its instance number.

Self-Test: Passed
Number of self-tests that succeeded.

Self-Test: Failed
Number of self-tests that failed.

Maintenance: Failed
Number of maintenance failures.

Physical address
The Ethernet address of the device currently in use. This may be the PROM address or an address overwritten by some other protocol.

PROM address
The permanent unique Ethernet address in the PROM for this Ethernet interface.

Actual address

Adapter level

Configured duplex
The value configured for duplex. Values can be Half Duplex, Full Duplex, or Auto-Negotiation.

Actual duplex
The value at which the adapter is presently operating. It might be different from the value configured, depending on the switch capability. If the adapter is not Up, the value displayed will be *Unknown*. Otherwise the value can be Half Duplex or Full Duplex.

Whenever the link partner (switch or hub) does not participate during the negotiation phase, “***” will follow the actual duplex mode value. When “***” is indicated the operational duplex value should be verified on the switch or hub for consistency.

Most hubs (unlike switches) can only support half-duplex mode, and are not capable of negotiation. As such the “***” indication will usually be displayed when the interface is connected to a hub.

A message will also be logged via the ELS system whenever a possibilty of a mis-match in duplex mode exists.

Note: If the link partner (switch or hub) to which the interface is connected does not respond during the negotiation phase, the two may result in operating in different duplex modes. That is, the interface may be operating in half-duplex, while the switch port is operating in full duplex mode. A mis-match in the duplex mode can result in severe performance degradation. See “10/100 Mbps Ethernet Configuration Commands” on page 24 for important information regards speed and duplex configurations.

Configured speed
The value configured for speed. Values can be 10 Mbps, 100Mbps, or Auto-Negotiation.

Actual speed

The speed at which the adapter is presently operating. If the adapter is not Up, the value displayed will be *Unknown*. Otherwise the value can be 10 Mbps or 100 Mbps.

Input statistics:

failed, packet too long or failed, frame too long

The Failed, Packet Too Long counter increments when the interface receives a packet that is larger than the maximum size of 1518 bytes for an Ethernet frame. This data is exported via SNMP as the dot3StatsFrameTooLongs counter.

failed, CRC error or failed, FCS (Frame Check Sequence) error

The Failed, CRC (Cyclic Redundancy Check) Error counter increments when the interface receives a packet with a CRC error. This data is exported via SNMP as the dd3StatsFCSErrors counter.

failed, alignment error

The Failed, Framing Error counter increments when the interface receives a packet where the length in bits is not a multiple of eight.

failed, receive overflow

Overflow error indicates that the receiver has lost all or part of the incoming frame, due to an inability to move data from the receive FIFO into memory buffer before the internal FIFO overflowed.

receive collision

Indicates the total number of collisions encountered by the receiver support on the adapter.

Note: This counter cannot be cleared by the **clear statistics** command because it is maintained on the adapter. The **test network** command is the only way to reset this counter.

missed frame

Indicates the number of incoming receive frames lost due to unavailability of a receive buffer in the system. This error indicates that the system is not processing received frames as fast as they are being received from the local network.

Note: This counter cannot be cleared by the **clear statistics** command because it is maintained on the adapter. The **test network** command is the only way to reset this counter.

frames filtered

Indicates the number of incoming frames that were discarded by the adapter. This counter is updated only when bridging is enabled.

Note: This counter is maintained on the adapter, and is cleared every time it is read. This counter will be cleared by the **interface statistics** and the **test network** commands.

receive underrun

Indicates the number of times the adapter did not have a second buffer to store a long frame (requiring more than one buffer).

Output statistics:

one retry

Indicates that exactly one retry was needed to transmit a frame. This data is exported via SNMP as the dot3StatsDeferredTransmissions counter.

single collision

The Single Collision counter increments when a packet has a collision on the first transmission attempt, and then successfully sends the packet on the second transmission attempt. This data is exported via SNMP as the dot3StatsSingleCollisionFrames counter.

multiple collisions

The Multiple Collisions counter increments when a packet has multiple collisions before being successfully transmitted. This data is exported via SNMP as the dot3MultipleCollisionFrames counter.

failed, transmit underflow

Transmit underrun indicates that transmitter has truncated a message because it could not read data from the memory fast enough. It also indicates that the FIFO on the adapter has emptied out before the end of the frame was reached. IFO into memory buffer before the internal FIFO overflowed.

failed, excess collisions

The Failed, Excess Collisions counter increments when a packet transmission fails due to 16 successive collisions. This error indicates a high volume of network traffic or hardware problems with the network. This data is exported via SNMP as the dot3StatsExcessiveCollisions counter.

failed, loss of carrier

Loss of carrier is set when the carrier is lost during transmission. The adapter does not retry upon loss of carrier. It will continue to transmit the whole frame until done.

late collisions

A late collision indicates that a collision has occurred after the first channel slot time has elapsed. The adapter does not retry on late collisions.

more than one retry

More than one retry indicates that more than one retry was needed to transmit a frame.

buffer error

Buffer error occurs if there is a memory corruption problem in the system, or under certain FIFO underflow conditions on the adapter.

total collisions

The Total Collisions counter increments by the number of collisions a packet incurs.

excessive deferral

Excessive deferral indicates that the transmitter on the adapter has experienced Excessive Deferral on this a transmit frame, where Excessive Deferral is defined in the ISO 8802-3 (IEEE/ANSI 802.3) standard.

deferred

Deferred indicates the number of times the adapter had to defer while trying to transmit a frame. This condition occurs if the DMA channel is busy when the adapter is ready to transmit.

memory error

Memory errors occur when the adapter is not given access to the system interface bus within the programmable length of time. This error will normally occur during transmit operations, indicating transmit underrun.

Auto-negotiation on the 10/100 Mbps Ethernet Interface

Specifying values other than *AUTO* for speed and duplex on the 10/100 Ethernet interface or its link partner (switch port) can result in duplex mode mismatch or link activation failures.

Link activation failures due to configuration mismatches will occur on the IBM 2216 whenever the speed configured at both ends are not identical.

When a value other than *auto-negotiate* is specified for duplex (on 10/100 Ethernet devices such as switches, hubs, adapters), the device will generally not transmit its configured values to its link partner. The lack of information regarding it's link partners capability is the main cause of a mismatch in duplex mode.

A mis-match of the duplex mode between the interface and the switch port can lead to severe performance degradation.

Note: IT IS STRONGLY RECCOMENDED THAT -AUTO- BE CONFIGURED FOR SPEED AND DUPLEX ON BOTH THE SWITCH PORT AND THE IBM 2216 interface.

Configuring Values Other than Auto for Duplex

The IBM 2216 10/100 Mbps Ethernet interfaces will alert you whenever there is a possibility of a mis-match in the duplex mode between the switch and the router. When auto-negotiation is not configured on both ends, there is no definite way for either end to determine the duplex mode in use at the remote end, and the switch and the router interface can be operating with non-identical duplex modes.

Depending on the switch implementation, the switch port may operate at half duplex, when you have configured full duplex. As such, there is always a possibility of a mis-match to occur between the switch port and the router interface. The IBM 2216 will alert you whenever there is such a possibility, but some switches do not give such an indication.

Configurations That May Result in Link Activation Failure on the IBM 2216

The primary cause of the link activation failure is mis-matched speeds.

To avoid link activation failure, configure *auto* for speed and duplex on the IBM 2216 and the switch port.

Note: These results may vary depending on the switch manufacturer and model.

Table 2. Configurations That May Result in Link Failure on the IBM 2216

IBM 2216	Hub/Switch
Auto 10	HDX* 100
Auto 10	FDX* 100
Auto 100	HDX 10
Auto 100	FDX 10
HDX 10	HDX 100
HDX 10	FDX 100
HDX 100	HDX 10
HDX 100	FDX 10
FDX 10	HDX 100
FDX 10	FDX 100
FDX 100	HDX 10
FDX 100	FDX 10

* HDX = Half Duplex   FDX = Full Duplex

Configurations That May Result in Mismatched Duplex Modes During Operation

The primary cause of the mis-matched duplex modes is the disabling of auto-negotiation on the switch port and/or the IBM 2216 interface.

To avoid mis-matches in duplex mode, configure *auto* for speed and duplex on the IBM 2216 and the switch port.

Note: These results may vary depending on the switch manufacturer and model.

Table 3. Configurations That May Result in Mismatched Duplex Modes During Operation

Configuration		Result	
IBM 2216	**Hub/Switch**	**IBM 2216**	**Hub/Switch**
Auto Auto	FDX* 10	HDX* 10	FDX 10
Auto Auto	FDX 100	HDX 100	FDX 100
HDX Auto	FDX 10	HDX 10	FDX 10
HDX Auto	FDX 100	HDX 100	FDX 100
Auto 10	FDX 10	HDX 10	FDX 10
Auto 100	FDX 100	HDX 100	FDX 100
HDX 10	FDX 10	HDX 10	FDX 10
FDX Auto	HDX 10	FDX 10	HDX 10
FDX Auto	HDX 100	FDX 100	HDX 100
HDX 100	FDX 100	HDX 100	FDX 100
FDX 10	HDX 10	FDX 10	HDX 10
FDX 100	HDX 100	FDX 100	HDX 100

* HDX = Half Duplex   FDX = Full Duplex

Configuring and Monitoring the 10/100 Mbps Ethernet Network Interface

This chapter describes the 10/100 Mbps Ethernet interface configuration and operational commands. It includes the following sections:

- "Accessing the Interface Configuration Process" on page 24
- "10/100 Mbps Ethernet Configuration Commands" on page 24
- "Accessing the 10/100 Mbps Interface Monitoring Process" on page 26
- "10/100 Mbps Ethernet Interface Monitoring Commands" on page 27

Accessing the Interface Configuration Process

Use the following procedure to access the configuration process. This process gives you access to an Ethernet interface's *configuration* process.

1. At the OPCON prompt, enter **talk 6**. (For more detail on this command, refer to FORM='NORMAL'..) For example:

   ```
   * talk 6
   Config>
   ```

   The CONFIG prompt (`Config>`) displays on the console. If the prompt does not appear when you first enter configuration, press **Return** again.

2. At the CONFIG prompt, enter the **list devices** command to display the network interface numbers for which the router is currently configured.

3. Record the interface numbers.

4. Enter the **network** command and the number of the Ethernet interface you want to configure. For example:

   ```
   Config> network 0
   Ethernet 100 interface configuration
   ETH100 Config>
   ```

   The 10/100 Mbps Ethernet configuration prompt (`ETH100 Config>`), is displayed.

10/100 Mbps Ethernet Configuration Commands

This section describes the 10/100 Mbps Ethernet configuration commands. Enter the commands at the `ETH config>` prompt.

Table 4. 10/100 Mbps Ethernet Configuration Command Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See FORM='FULL'..
Duplex	Sets the duplex mode.
IP-Encapsulation	Sets the IP encapsulation as Ethernet (type X'0800') or IEEE (802.3 with SNAP).
List	Displays the current connector-type, and IP encapsulation.
Physical-Address	Sets the physical MAC address.
Speed	Sets the link speed.
Exit	Returns you to the previous command level. See FORM='FULL'..

Duplex

Use the **duplex** command to set the duplex mode.

| **Note:** The default value of *auto* is recommended. The value **half-duplex** or **full-duplex** should be specified only if auto-negotiation does not result in successful activation of the interface or desired duplex mode.
|
|

| If a value other than *auto* is specified, ensure that the same value is configured on the switch port. After configuring the switch port to match the duplex specified on the 10/100 Mbps Ethernet interface, disable and test the interface.
|
|
|

Verify that the actual duplex mode shown on the interface status panel matches the operational value on the switch port.

The interface may enter the Up state with mis-matched duplex mode. Operating with mis-matched duplex modes on the interface and switch port can cause severe performance degradation.

See “Configuring Values Other than Auto for Duplex” on page 21 for information about duplex mode.

Syntax:

duplex half_duplex

full_duplex

auto

Half_duplex
The interface will not transmit while receiving or receive while transmitting.

Full_duplex
The interface will transmit and receive simultaneously.

Auto The interface will automatically select half-duplex or full duplex depending on the link partner's capability.

IP-Encapsulation

Use the **IP-encapsulation** command to select Ethernet (Ethernet type X'0800') or IEEE 802.3 (Ethernet 802.3 with SNAP). Enter **e** or **i** for the type.

Syntax:

IP-encapsulation *type*

Example: `IP-encapsulation e`

List

Use the **list** command to display the current configuration for the 10/100 Mbps Ethernet interface.

Syntax:

list all

Example: `list all`

```
The duplex is   HALF DUPLEX
The speed is   100Mb
IP Encapsulation:   Ether

MAC Address:        023456789A56
```

Physical-Address

Use the **physical-address** command to set the physical (MAC) address.

Syntax:

physical-address ***address***

physical-address

This command lets you indicate whether you want to define a locally administered address for the Ethernet interface’s MAC sublayer address, or use the default burned-in address (indicated by all zeros). The MAC sublayer address is the address that the Ethernet interface uses to receive and transmit frames.

Note: Pressing **Enter** leaves the value the same. Entering **0** causes the router to use the burned-in address. The default is to use the burned-in address.

Valid Values: Any 12-digit hexadecimal address.

Default Value: burned-in address (indicated by all zeros).

Example:

```
physical-address
MAC address in 00:00:00:00:00:00 form []? 12:15:00:FA:00:FE
```

Speed

Use the **speed** command to set the speed used by this interface.

Note: The default value of *auto* is recommended. The values of **ten** and **hundred** should be specified only if auto-negotiation does not result in sucessful activation of the interface or desired speed.

If a value other than *auto* is specified, ensure that the same value is configured on the switch port. After configuring the switch port to match the speed specified on the 10/100 Mbps Ethernet interface, disable and test the interface.

If the interface and switch (or hub) port are not configured for identical speed, the interface will not attain the Up state.

See “Auto-negotiation on the 10/100 Mbps Ethernet Interface” on page 21 for information about auto-negotiation.

Syntax:

speed ten

hundred

auto

Ten The interface will operate at 10 Mbps.

Hundred The interface will operate at 100 Mbps

Auto The interface will automatically select the speed (10 Mbps or 100 Mbps) depending on the link partner's capability.

Accessing the 10/100 Mbps Interface Monitoring Process

To monitor information related to the 10/100 Mbps Ethernet Network Interface, access the interface monitoring process by doing the following:

1. At the OPCON prompt, enter **talk 5**. For example:

   ```
   * talk 5
   ```

   The GWCON prompt (+) is displayed on the console. If the prompt does not appear when you first enter GWCON, press **Return** again.

2. At the GWCON prompt, enter the **configuration** command to see the protocols and networks for which the router is configured. For example:

   ```
   + configuration
   ```

   See FORM='NORMAL'. for sample output of the **configuration** command.

3. Enter the **network** command and the number of the Ethernet interface. In this example:

   ```
   + network 0
   ETH100>
   ```

   The 10/100 Mbps Ethernet monitoring prompt is displayed. You can now view information about the 10/100 Mbps Ethernet interface by entering monitoring commands.

10/100 Mbps Ethernet Interface Monitoring Commands

This section summarizes the 10/100 Mbps Ethernet monitoring commands. Enter commands at the `ETH100>` prompt. Table 5 lists the monitoring commands.

Table 5. Ethernet Monitoring Command Summary

Command	Function
? (Help)	Displays all the commands available for this command level or lists the options for specific commands (if available). See FORM='FULL'..
Collisions	Displays collision statistics for the specified Ethernet interface.
Exit	Returns you to the previous command level. See FORM='FULL'..

Collisions

This command displays the counts of transmissions for packets that incurred collisions before successful transmission. Counters are displayed for packets sent after 15 collisions. An increased number of packets transmitted with collisions and higher numbers of collisions per packet are signs of transmitting onto a busy Ethernet.

These counters are cleared by the OPCON **CLEAR** command. This data is exported via SNMP as the dot3CollTable counter.

Syntax:

coll**isions**

Example:

```
Eth100> coll

Transmitted with  1 collisions:0
Transmitted with  2 collisions:0
Transmitted with  3 collisions:0
Transmitted with  4 collisions:0
Transmitted with  5 collisions:0
Transmitted with  6 collisions:0
Transmitted with  7 collisions:0
Transmitted with  8 collisions:0
Transmitted with  9 collisions:0
Transmitted with 10 collisions:0
Transmitted with 11 collisions:0
Transmitted with 12 collisions:0
Transmitted with 13 collisions:0
Transmitted with 14 collisions:0
Transmitted with 15 collisions:0
```

Changes to TFTP

The following functional changes have been made to TFTP:

- Increased the default amount of time that must pass before a transfer times out to approximately 5 minutes.
- Provided two new patch variables that allow you to control the retransmission interval and the time for a transfer to timeout.

  tftp-max-rxto-time is a new patch variable you can use to specify the maximum time to wait for a response from the partner before failing the transfer. **Default value**: 300 seconds

  tftp-min-rexmtime allows you to specify the smallest interval of time to wait for a response from the partner before retransmitting the last packet sent. **Default value**: 1 second.

  The units for both of these patch variables are seconds.
- Transferring large amounts of data to multiple files if necessary.

  This function is important for situations where the receiving TFTP server has a bug handling the block count. The TFTP protocol requires that a block count be transmitted with every data block. The acknowledgement for that data block carries the block number that was in the data block being acknowledged. The transmitter of the data won't send any more data until it receives an acknowledgement for the last data block sent. Once the receiver of the data sends the acknowledgement it expects to receive a data block with a block count that is one greater than the block count it previously received. This block count is two bytes long.

  Some TFTP servers have improperly implemented this as a signed short word (two-byte variable where the high order bit being 1 indicates a negative value) and others as an unsigned long word (four byte variable).

  If the amount of data to be transferred is so great that the block count wraps, then depending on how the receiver verifies the block count, it may or may not acknowledge the data. If the receiver uses a signed short, the problem will be experienced when the block count goes from 0x7fff to 0x8000. If the receiver uses an unsigned long, the problem will be experienced when the block count goes from 0xffff to 0x0000. In both cases the block count in the data block will appear to be less than the previously received block count and the receiver will not acknowledge.

  The transmitting TFTP on the router will either receive an error packet or time out waiting for the receiver to respond. When this happens, TFTP on the router will realize that the block count had wrapped and will automatically recover by making a write request to the receiver for a new file. The new file name is derived from the original file name. The new file name is derived by overlaying the last two characters of the original file name with two decimal digits. Every time the block count wraps, a new file will be written until all the data has been transferred. Tools like **cat** can be used at the receiver to concatenate the files.
- The patch variable **tftp-blocks-per-file** was added so that you can specify the maximum number of blocks to transfer to a file at the receiver. This allows you to tell the router to automatically do a write request for a new file once the number of blocks specified has been sent. Doing this circumvents the auto-

matic recovery described above, speeding up the transfer by avoiding the 5 minute timeout period.

The only values that may be specified for this patch variable are: 0xffff (65535) and 0x7fff (32767).

This patch variable is useful if you know that the receiving server has problems handling the wrap of the block count.