

IBM i
7.4

*Networking
APPC, APPN, and HPR*



Note

Before using this information and the product it supports, read the information in [“Notices” on page 55.](#)

This document may contain references to Licensed Internal Code. Licensed Internal Code is Machine Code and is licensed to you under the terms of the IBM License Agreement for Machine Code.

© **Copyright International Business Machines Corporation 1998, 2018.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

APPC, APPN, and HPR.....	1
PDF file for APPC, APPN, and HPR.....	1
Planning APPN and HPR network.....	1
Considerations in selecting APPC networking protocol.....	1
Considerations in designing an APPN and HPR network.....	2
Configuring APPC, APPN, and HPR.....	3
Manual configuration for APPN and HPR.....	3
Changing network attributes.....	3
Creating controller descriptions for APPC connections.....	5
Configure Enterprise Extender to perform switched line disconnect.....	5
Creating device descriptions for APPC connections.....	7
Creating APPN location lists.....	7
Creating mode descriptions.....	7
Creating class-of-service descriptions.....	8
Configuring Branch Extender support.....	8
Configuration considerations used to optimize error recovery performance.....	9
Considerations for the ONLINE parameter that can affect error recovery.....	9
Considerations for communications-related system values.....	9
Considerations for network attributes that can affect APPC error recovery.....	10
Considerations for line configuration settings that can affect error recovery.....	11
Considerations for the automatic deletion of APPC device descriptions.....	11
Link-level timers and retries.....	11
Considerations for controller configuration descriptions that can affect error recovery.....	11
Considerations for automatic delete device (AUTODLTDEV) parameter for error recovery... ..	12
Considerations for the INLCNN parameter that can affect error recovery.....	12
Considerations for the MINSWTSTS parameter that can affect error recovery.....	12
APPC controller recovery summary.....	13
Considerations for disconnect timer (DSCTMR) parameter for error recovery.....	14
Considerations for modes that can affect error recovery.....	14
Considerations for jobs that can affect error recovery.....	14
Considerations for the CMNRCYLMT parameter that can affect error recovery.....	15
Considerations for prestart job entries that can affect APPC error recovery.....	15
Considerations for job logs that can affect communications error recovery.....	16
The Change System Job (CHGSYSJOB) command.....	17
Configuring APPC with VTAM.....	17
Examples: APPC, APPN, and HPR configuration	18
Examples: APPN configuration.....	18
Example: Two systems as end nodes using APPN.....	18
Configuring system A (New York) as an end node.....	18
Configuring system B (Los Angeles) as an end node.....	19
Example: Three systems using APPN.....	20
Configuring system A (New York).....	20
Configuring system B (Los Angeles).....	21
Configuring system C (Chicago).....	22
Example: Two APPN networks with different network IDs linked together.....	23
Configuring system A (New York).....	23
Configuring system B (Detroit).....	24
Configuring system NN1 (Chicago).....	25
Configuring NN2 (Minneapolis).....	26
Optimizing APPN and HPR communication performance.....	27
Performance considerations for APPN and HPR.....	27

Communications optimization using high-performance routing.....	28
Communications optimization using APPN virtual controllers.....	30
Configuration parameters for fine-tuning APPC performance.....	31
Maximum length request/response unit size (MAXLENRU) parameter.....	31
Maximum frame size (MAXFRAME) parameter	31
Pacing (INPACING, OUTPACING, MAXINPACING) parameters.....	31
Transmission priority (TMSPTY) parameter.....	32
Wait time (QACRETRY and QACINTERVL) data areas.....	32
APPC, APPN, and HPR security.....	33
Session-level security for APPN and HPR.....	34
Protecting your system in an APPN and HPR environment.....	34
APPN filtering support.....	34
Creating a session endpoint filter.....	35
Class of service routing.....	36
Troubleshooting APPN and HPR	37
Solving remote communication problems using STRPASTHR.....	37
Solving communication problems using DSPAPPNINF.....	37
Solving communication problems using WRKAPPNSTS.....	38
Solving communications problems using communications trace.....	39
Solving communication problems using session activity.....	39
Systems Network Architecture sense codes.....	39
APPN error log data.....	39
Standard APPN diagnostic data.....	40
APPN session setup states.....	43
Optional APPN diagnostic data.....	47
Search-sent elements.....	47
Regular Route Selection control vector (RSCV) 46.....	48
Regular Route Selection control vector (RSCV) 0E.....	49
Single hop route failure element.....	49
Ineligible destination network nodes elements.....	50
Destination node list.....	50
User class-of-service with inactive transmission groups RSCV.....	51
Any class-of-service with active transmission groups RSCV.....	52
Notices.....	55
Programming interface information.....	56
Trademarks.....	56
Terms and conditions.....	57

APPC, APPN, and HPR

Systems Network Architecture (SNA) includes the layered logical structure, formats, protocols, and operational sequences that are used for transmitting information units through networks. Using APPC, APPN, and HPR is an example of implementing SNA.

You can use APPC, APPN, and HPR to connect the IBM® i server with other systems.

Enterprise Extender is a networking architecture that allows Systems Network Architecture (SNA) applications to run over Internet Protocol (IP) networks using High Performance Routing (HPR). This is the only way to run SNA applications over IP networks with communications input/output adapters (IOAs), such as Gigabit Ethernet. Gigabit Ethernet adapters do not automatically support SNA traffic. Enterprise Extender is required to allow SNA data to flow over a Gigabit adapter.

Note: By using the code examples, you agree to the terms of the [“Code license and disclaimer information”](#) on page 53.

Related concepts

[Migration from AnyNet to Enterprise Extender](#)

Related reference

[APPC Programming PDF](#)

PDF file for APPC, APPN, and HPR

You can view and print a PDF file of this information.


To view or download the PDF version of this document, select [APPC, APPN, and HPR](#).

Saving PDF files

To save a PDF on your workstation for viewing or printing:

1. Right-click the PDF link in your browser.
2. Click the option that saves the PDF locally.
3. Navigate to the directory in which you want to save the PDF.
4. Click **Save**.

Downloading Adobe Reader

You need Adobe Reader installed on your system to view or print these PDFs. You can download a free copy from the [Adobe Web site](http://www.adobe.com/products/acrobat/readstep.html) (www.adobe.com/products/acrobat/readstep.html) .

Planning APPN and HPR network

Before you set up and configure your APPN and HPR network, keep in mind these considerations when planning the network.

Considerations in selecting APPC networking protocol

When choosing the advanced program-to-program communications (APPC) networking protocol for your business, you must understand some of the operational characteristics for APPN and HPR. These operational characteristics can affect the communication performance on your system.

Notes about the APPC networking protocol:

- HPR provides a significant enhancement over APPN in terms of network availability by establishing and maintaining end-to-end connections and the ability to switch paths transparently. For HPR, segmentation and reassembly are accomplished in the central processing unit (CPU).
- Enterprise Extender is a networking architecture that allows Systems Network Architecture (SNA) applications to run over Internet Protocol (IP) networks using High Performance Routing (HPR). This is the required way to run SNA applications over IP networks with communications input/output adapters (IOAs), such as Gigabit Ethernet.

Related concepts

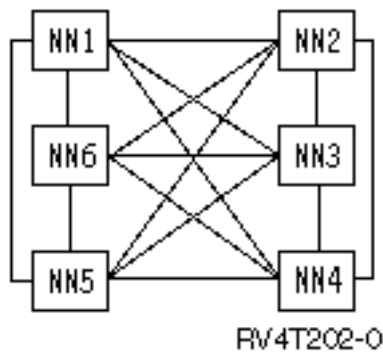
[Migration from AnyNet to Enterprise Extender](#)

Considerations in designing an APPN and HPR network

When you design your network, consider these factors to optimize performance.

- Avoid mesh connectivity

The number of control program-to-control program (CP-CP) sessions that are configured for each network node (NN) has a direct impact on the performance of a network. Network control information such as topology updates and location searches flow over CP-CP sessions. A consequence of too many CP-CP sessions is that information is sent out to more nodes and the same node multiple times. This increases the network processing that is done. In a mesh-connected network, every NN has a CP-CP session with every other NN, increasing the number of CP sessions in this network. The number of CP-CP sessions in the network should be kept to a minimum while still providing necessary connectivity.



- Consider backup CP-CP sessions where appropriate

A CP-CP spanning tree is a term that is used to describe the contiguous path of CP-CP sessions between nodes throughout the network. CP-CP sessions carry necessary control information and are required between NNs in order to participate in the APPN network. Careful analysis to determine the minimal set of links necessary to support CP-CP sessions is important. Once these links are identified, it is recommended that back-up links providing alternate CP-CP sessions are added to the network. These backup links ensure availability of the CP-CP spanning tree and are needed if the critical links fail.

- Consider using border nodes

APPN architecture does not allow two adjacent APPN NNs to connect and establish CP-CP sessions unless they have the same network identifier (NETID). Border nodes overcome this restriction. Border nodes enable NNs with different NETIDs to connect and allow session establishment between logical units (LU) in different NETID subnetworks. Border nodes prevent topology information from flowing across different NETID subnetworks. Use border nodes to subdivide a large APPN network into smaller and more manageable subnetworks. IBM i provides this border node capability for only adjacent networks.

- Processing reduction for entry nodes (ENs) and low-entry networking (LEN) nodes

The amount of processing is reduced when the IBM i is an EN or LEN node, as opposed to an NN for the following reasons:

- ENs and LEN nodes do not receive most network topology and directory search information flows, while NNs do.
- Reduction of network flows resulting from fewer NNs

In addition, topology information about ENs and LEN nodes does not flow through the network. NN topology does flow to the entire network that causes the other NNs to process information about every other NN.

The NNs perform route calculation for themselves and other ENs and LEN nodes. (This function flows from the EN or LEN node to the NN.)

- Use Branch Extender

Branch Extender is an extension to the APPN network architecture. It appears as a NN to the local area network (LAN), and as an EN to the wide area network (WAN). This reduces topology flows about resources in the LAN from being disconnected from the WAN. The only topology flows necessary are for network management that identify the types of links.

Related concepts

[Optimizing APPN and HPR communication performance](#)

If you are responsible for network administration, you might be concerned with the speed at which computers throughout that network can exchange data.

[Communications optimization using APPN virtual controllers](#)

An *APPN virtual controller* is a controller description that Advanced Peer-to-Peer Networking (APPN) can use and that high-performance routing (HPR) support uses.

Related tasks

[Changing network attributes](#)

Network attributes describe the local system name, the default local location name, the default control point name, the local network identifier, and the network node type.

Configuring APPC, APPN, and HPR

You can have APPC, APPN, and HPR configured automatically or manually on your system.

Related concepts

[Communications optimization using high-performance routing](#)

High-performance routing (HPR) is the next evolution of Advanced Peer-to-Peer Networking (APPN). HPR differs from APPN in the areas of transport, intermediate session routing, congestion control, and error recovery.

Manual configuration for APPN and HPR

After you configure an advanced program-to-program communications (APPC) environment, you need to change network attributes, the first step in configuring APPN and HPR.

Related tasks

[Creating a line description](#)

Changing network attributes

Network attributes describe the local system name, the default local location name, the default control point name, the local network identifier, and the network node type.

If the machine is an end-node, the attributes also contain the names of the network servers that are used by this IBM i system. Network attributes also determine whether the system uses HPR, or whether you want to use virtual controllers for APPN.

To change the network attributes, follow these steps:

1. Vary off all APPC and host controllers.

The easiest way to do this is to use:

```
VRYCFG CFGOBJ(*APPN) CFGTYPE(*CTL) STATUS(*OFF) RANGE(*NET)
```

2. Type the Change Network Attributes (CHGNETA) command on any IBM i command line and press F4.
3. Use the online help information to complete the parameter values.
4. Press the Enter key.

The network attributes are changed.

5. Vary on all the controllers that were varied off in the first step.

Use the following example:

```
VRYCFG CFGOBJ(*PRVCFGTYPE) CFGTYPE(*CTL) STATUS(*ON) RANGE(*NET)
```

Note: The VRYCFG of *APPN will find *all* APPN controllers and devices on the system and will try to vary them off. The VRYCFG with *PRVCFGTYPE will then try to vary them *all* on.

Configuring APPN virtual controllers

On the IBM i, local applications that need to establish LU 6.2 sessions to other locations in the APPN network require an APPC device description that specifies APPN(*YES). For simplicity, these devices are referred to as APPN devices. Multiple device descriptions can be created and used simultaneously to communicate between the same local location and the remote location pair. After a session is established, the controller description continues to use the same APPN device description for the life of that session.

Configuration to use virtual controllers:

- Set the ALWVRTAPPN network attribute to (*YES)

Existing APPN device descriptions (attached to a real controller description) will no longer be used.

Note: This does not affect HPR since it always uses virtual APPN support.

If you are using the HPR tower option (RTP):

1. Vary off all APPN controllers. Use:

```
VRYCFG CFGOBJ(*APPN) CFGTYPE(*CTL)
STATUS(*OFF) RANGE(*NET)
```

2. Set the Allow HPR transport tower support (ALWHPRTWR) parameter to (*YES)
3. Vary on all your APPN controllers. Use:

```
VRYCFG CFGOBJ(*PRVCFGTYPE) CFGTYPE(*CTL)
STATUS(*ON) RANGE(*NET)
```

Configuring APPN using Branch Extender

To use Branch Extender, see the topic Configuring Branch Extender support.

Considerations for system names

Use caution when you use names with the special characters # (X'7B'), \$ ('5B'), and @ ('7C'). These special characters might not be on the keyboard of the remote system. These special characters are not supported for APPC over TCP/IP (network IDs and location names only). The use of these symbols should be limited to migration of the operating system. Do not use these characters for newly created names.

If you are using a national language keyboard that does not have the #, \$, or @ symbols, see the information about National language keyboard types and SBCS code page in the information center.

The names that may be exchanged with remote systems include the following:

- Network IDs
- Location names
- Mode names

- Class-of-service names
- Control-point names
- Connection network names

Related concepts

[Considerations in designing an APPN and HPR network](#)

When you design your network, consider these factors to optimize performance.

Related tasks

[Configuring Branch Extender support](#)

A Branch Extender is an extension to the APPN network architecture that appears as a network node (NN) to the local area network (LAN) and as an end node (EN) to the wide area network (WAN). Branch Extender reduces topology exchange messages between NNs when a link in the LAN disconnects from the WAN.

Related reference

[National language keyboard types and SBCS code pages](#)

Creating controller descriptions for APPC connections

A controller description defines the adjacent systems in the network. To create controller descriptions, follow these steps.

1. Type one of these commands on any IBM i command line for the type of controller you need to define and press F4.
 - Create Controller Description (APPC) (CRTCTLAPPC)
 - Create Controller Description (Systems Network Architecture (SNA) HOST) (CRTCTLHOST)
2. Use the online help information to choose the correct parameter values.
3. Press Enter.

The controller description is created.

To specify Advanced Peer-to-Peer Networking (APPN) support, specify *YES on the APPN parameter of the CRTCTLHOST command.

To specify Enterprise Extender support (SNA over IP using HPR), specify *HPRIP on the LINKTYPE parameter of the CRTCTLAPPC command.

Enterprise Extender support can be configured to run over Point-to-Point Protocol (PPP) lines and perform switched line disconnect behavior after periods of inactivity.

Configure Enterprise Extender to perform switched line disconnect

The SWITCHED parameter can be specified with *YES on the **CRTCTLAPPC** command when *HPRIP is specified for the LINKTYPE. This configuration allows Enterprise Extender support to run over Point-to-Point Protocol (PPP) lines and perform switched line disconnect behavior after periods of inactivity.

The period of inactivity is specified by the Disconnect timer (DSCTMR) parameter. Configure the Disconnect timer (DSCTMR) values in the APPC controller description to be less than the Line inactivity timeout value specified in the PPP connection profile for the switched PPP line. This configuration allows the Enterprise Extender support on the APPC controller to disconnect cleanly before the switched PPP line detects a line inactivity timeout, which would cause the modem or line to drop. See [Configuring PPP](#) for information on configuring a PPP connection profile. Only specify the SWITCHED parameter as *YES for Enterprise Extender support when it is running over an actual switched PPP line. The Initial connection (INLCNN) and Dial initiation (DIALINIT) parameters on the APPC controller description must be compatible with the dial and answer mode parameters of the PPP switched line. For example, if the APPC controller is configured with INLCNN *DIAL then also configure the PPP connection profile for the switched line to support both dial and answer operations. If the APPC controller is configured with INLCNN*ANS, then the PPP connection profile for the switched line only needs to be configured for answer operations. The remote phone number for dial operations is only specified in the PPP connection profile. Also, no attached line is specified in the switched line list (SWTLINLST) parameter of the APPC

controller since the local and remote IP addresses that are configured in the controller select the switched PPP line to use. If the Switched disconnect parameter is SWTDSC(*YES), the LDLC liveness timer is ignored to allow the connection to drop after periods of inactivity.

Configuration examples for dial operation:

```
CRTCTLAPPC CTLD(DIAL_IMMED) LINKTYPE(*HPRIP) SWITCHED(*YES) RMTINTNETA('10.44.58.2')
LCLINTNETA(*SYS) RMTNETID(APPN) RMTCPNAME(SYSTEM_A) DIALINIT(*IMMED)
USRDFN1(128) USRDFN2(128) USRDFN3(128) DSCTMR(20 10)
```

In this immediate dial example, the APPC Controller initiates a DIAL at vary on. The controller will stay active for minimum of 20 seconds after vary on and will disconnect the line 10 seconds after the last SNA session is unbound when the user application completes.

```
CRTCTLAPPC CTLD(DIAL_DELAY) LINKTYPE(*HPRIP) SWITCHED(*YES) RMTINTNETA('10.44.58.2')
LCLINTNETA(*SYS) RMTNETID(APPN) RMTCPNAME(SYSTEM_A) DIALINIT(*DELAY)
USRDFN1(128) USRDFN2(128) USRDFN3(128) DSCTMR(20 10)
```

In this delayed dial example, the APPC Controller does not dial until a user application sends data. The controller will stay active for a minimum of 20 secs and will disconnect the line 10 secs after the last SNA session is unbound for the user application.

Corresponding dial PPP profile configuration that uses internal modem:

- Protocol type: PPP
- Mode type: Dial on demand (answer enabled dedicated peer)
- Type of line service: Single line
- Line name: QPPPCMNO7
- Remote phone number: 3331234
- Line inactivity timeout: 30

Note: This value must be greater than the disconnect timer value in the APPC controller.

- Local IP address: Use fixed IP address 10.44.58.3 (*VIRTUALIP)
- Remote IP address: Use fixed IP address 10.44.58.2

Configuration example for answer operation:

```
CRTCTLAPPC CTLD(ANS_CALL) LINKTYPE(*HPRIP) SWITCHED(*YES) RMTINTNETA('10.44.58.3')
LCLINTNETA(*SYS) RMTNETID(APPN) RMTCPNAME(SYSTEM_B) INLCNN(*ANS) USRDFN1(128)
USRDFN2(128) USRDFN3(128) DSCTMR(20 10)
```

In this answer example, the APPC Controller accepts only incoming packets to establish a data link connection. The controller will stay active for minimum of 20 seconds after vary on and will disconnect the line 10 seconds after the last SNA session for user application is unbound.

Corresponding answer PPP profile configuration that uses internal modem:

- Protocol type: PPP
- Mode type: Switched line-answer
- Type of line service: Single line
- Line name: QPPPCMNO8
- Line inactivity timeout: 30

Note: This value must be greater than the disconnect timer value in the APPC controller.

- Local IP address: Assign fixed IP address 10.44.58.2 (*VIRTUALIP)
- Remote IP address: Use fixed IP address
- Starting IP address: 10.44.58.3

Creating device descriptions for APPC connections

A device description for APPC connections describes the characteristics of the physical or program device that is to communicate with the local system.

Device descriptions can describe a physical device (such as an Advanced Function Printing device), or logically represent a communications session or a program running on another system.

Note: The device description is typically created after the controller description. Device descriptions for Advanced Peer-to-Peer Networking (APPN), Transmission Control Protocol/Internet Protocol (TCP/IP), and user-defined communications are typically created automatically. When the Create Device Description (APPC) command is used to create APPN devices, the APPN parameter must be set to *YES.

The system automatically creates devices for APPN communications.

If you need to create a device description, follow these steps:

1. Type one of these commands on the IBM i command line for the type of device you are creating and press F4.
 - Create Device Description (APPC) (CRTDEVAPPC)
 - Create Device Description (Display) (CRTDEVDSP)
 - Create Device Description (Host) (CRTDEVHOST)
 - Create Device Description (Printer) (CRTDEVPRT)
 - Create Device Description (SNA Pass-through) (SNPT) (CRTDEVSNTPT)
 - Create Device Description (SNA upline facility (SNUF)) (CRTDEVSNUF)
2. Use the online help information to choose the parameter values.
3. Press Enter.

The device description is created.

Creating APPN location lists

An APPN locations list defines special characteristics of remote locations for APPN.

Special characteristics of remote locations include whether the remote location is in a different network from the local location and the security requirements for both. If special characteristics of remote locations exist, an APPN remote location list is required.

One local location name is the control point name that is specified in the network attributes. If you need additional locations for the IBM i system, an APPN local location list is required.

Note: QAPPNSSN and QAPPNDIR are two special configuration lists that can be manually configured to make your system secure.

To create APPN location lists, do the following:

1. Type the Create Configuration List (CRTCFGL) command on any IBM i command line and Press F4.
2. Specify *APPNLCL for the configuration list type (Type parameter).
3. Use the online help information to choose the correct parameter values.
4. Press Enter.

The APPN location list is created.

Creating mode descriptions

A mode description describes the session characteristics (including the number of sessions) that are used to negotiate the allowable values between the local and remote locations. The IBM i mode descriptions are used only by APPC, APPN, and HPR support.

Note: The system ships with several mode descriptions. You probably will not need to create one. You can use the Work with Mode Descriptions (WRKMODD) command to find out which mode descriptions already exist on your system.

The mode description also specifies a class of service description (COSD) that is used if and when this mode is used to cross an APPN network.

If you need to create a mode description, do the following:

1. Type the Create Mode Description (CRTMODD) command on any IBM i command line and press F4.
2. Use the online help information to choose the parameter values.
3. Press Enter.

The mode description is created.

In order for APPN and HPR to choose an optimal route at any given point in time, the pre-established sessions and locally controlled parameters should be set to zero.

Notes:

1. If pre-established sessions are not set to zero, the first time the mode is started (through session establishment or by using the STRMOD command) APPN and HPR will establish the number of sessions specified. These sessions remain up even if conversations are not active.
2. If locally controlled sessions are not set to zero, (through session establishment, or by using the STRMOD command) APPN and HPR establish one session that will not be taken down at conversation end.

Creating class-of-service descriptions

Class-of-service descriptions are used only by APPN and HPR. A class-of-service description tells the system which network nodes and transmission groups are acceptable and, of those acceptable, which are preferred during route selection.

The descriptions can include information such as transmission priority, link speed, cost-per-connection time, and security.

To create a Class-of-service description, do the following:

1. Type the Create Class-of-Service Description (CRTCOSD) command on any IBM i command line and press F4.
2. Use the online help information to choose the parameter values.
3. Press Enter.

The class-of-service description is created.

Related concepts

Class of service routing

Class-of-service (COS) descriptions are used to calculate the route that a session takes. COS descriptions also specify the transmission priority that governs the rate of data transfer after a session is established.

Configuring Branch Extender support

A Branch Extender is an extension to the APPN network architecture that appears as a network node (NN) to the local area network (LAN) and as an end node (EN) to the wide area network (WAN). Branch Extender reduces topology exchange messages between NNs when a link in the LAN disconnects from the WAN.

The only topology flows necessary are for network management identifying the types of links.

To configure Branch Extender, follow the actions below:

1. Set the NODETYPE parameter in the network attribute to *BEXNODE
2. Set the BEXROLE controller parameter.

This specifies the role of the local system in an APPN network for the remote controller being configured. The two options for BEXROLE are:

- *NETNODE: The local system takes the role of a network node for the remote controller.
- *ENDNODE: The local system takes the role of an end node for the remote controller.

Related tasks

[Changing network attributes](#)

Network attributes describe the local system name, the default local location name, the default control point name, the local network identifier, and the network node type.

Configuration considerations used to optimize error recovery performance

How your system is configured makes a significant difference in its performance during communications error recovery.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Related reference

[Communications Management PDF](#)

Considerations for the ONLINE parameter that can affect error recovery

Careful use of the ONLINE configuration parameter helps avoid unnecessary communications error recovery.

Most communications configuration objects are created with the ONLINE parameter default set to *YES (except for PPP lines, the ONLINE parameter is set to *NO).

Consider the setting of the ONLINE parameter in any of the following commands:

- CRTCTLxxx commands
- CRTDEVxxx commands
- CRTLINxxx commands

When choosing how to set the ONLINE parameter, consider the following:

- Limit the configuration objects that vary on during IPL with the ONLINE parameter set to *YES. These objects should be only those like tape drives, CD-ROM drives, and selected local workstations that are critical to getting your applications up and available for general system use.
- Place critical users in a subsystem group, and vary on configuration objects for this group by using the ONLINE parameter that is set to *YES. This allows critical users to get back online sooner.
- For noncritical users, vary on configuration objects at a later point by setting the ONLINE parameter to *NO. Use a CL program or change the system startup program to manage vary on of the remaining configuration objects.
- Whenever possible, avoid varying on any configuration that would fail in attempts to connect to remote systems.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Related reference

[Communications Management PDF](#)

Considerations for communications-related system values

System values, such as system date and library list, control information for the operation of certain parts of the system. You can change the system value to define your working environment.

The following information explains more about each of the system values for communications error recovery.

- *QCMNARB* (communications arbiters): controls the number of communication arbiter system jobs that are available to process communications functions.

- Do not set this value to zero unless directed by software service. If this system value is set to zero, the work is performed in the QSYSARB and QPLUS system jobs as opposed to being performed by the communication arbiters.
- The QCMNARB system value supports the following values: *CALC, 0-99.
- *CALC is the default setting for this system value. The system determines the number of jobs based on the system's hardware configuration.
- Consider having more than one QCMNARB job if there is an excessive amount of these system activities.
- A change to this value requires an initial program load (IPL) of the system for it to take effect.
- *QPASTHRSVR* (pass-through servers): controls how many pass-through server jobs are available for processing display station pass-through requests.
 - The default setting of this system value is calculated based on the hardware configuration of your system.
 - Consider multiple pass-through server jobs in error recovery situations to make the system quicker.
- *QCMNRCYLMT* (communications recovery limit): controls the number of automatic recovery attempts to make. It also controls when an inquiry message is sent to the system operator when the specified number of recovery attempts has been reached.
 - If the CMNRCYLMT parameter value is specified as *SYSVAL for a network interface description, a line description, or a controller description, then the QCMNRCYLMT value is also used. These parameter values also contain a count limit and time interval.

The count limit can be 0 (no recovery attempted) to 99. The time interval can be 0, or a value from 1 to 120 (minutes). A count limit of 0, and a time interval of more than 0 effectively disables automatic second-level error recovery. This may cause the devices and controllers to go into recovery pending (RCYPND) state, and require operator intervention. A count limit of more than 0, and a time interval of 0 allows automatic second-level error recovery continuously. However, this is not recommended.

Note: To avoid looping recovery, keep the number of retries small; you do not want time to expire before the number of retries are exceeded. Otherwise, you will end up in infinite recovery.
- *QDEVRCYACN* (device I/O recovery action): Controls the recovery action to take for the job when a device error is encountered on a reading and writing operation on the *REQUESTER device for interactive jobs.

Related concepts

[Work management](#)

Related reference

[Communications Management PDF](#)

[Considerations for the CMNRCYLMT parameter that can affect error recovery](#)

The QCMNRCYLMT system value or the recovery limits (CMNRCYLMT) parameter on the configuration object controls automatic communications error recovery.

[Considerations for job logs that can affect communications error recovery](#)

You need to consider whether to generate job logs when an error condition occurs and the active jobs are ended.

Considerations for network attributes that can affect APPC error recovery

Network attributes control information about the communications environment. Allow APPN virtual controller support (ALWVRTAPPN) and Virtual controller autcreate APPC device limit (VRTAUTODEV) are network attributes that play a role when a communications error occurs.

The following information explains more about each network attribute and how the attribute affects system performance during error recovery.

- Allow APPN virtual controller support (ALWVRTAPPN) controls whether APPN devices should be attached to real APPN controllers, or to a virtual controller.

- The default value is *NO.
- Use virtual APPN controllers to limit the number of devices that go into error recovery when a failure occurs.
- ALWVRTAPPN can be used to eliminate multiple device descriptions that can get created when multiple routes through the APPN network exist.
- Virtual controller autcreate APPC device limit (VRTAUTODEV) indicates the maximum number of automatically created APPC devices for each virtual controller when the following is true:
 - The Allow APPN Virtual Controller (ALWVRTAPPN) network attribute is *YES.
 - The Allow HPR Transport Tower (ALWHPRTWR) network attribute is *YES.

The VRTAUTODEV network attribute specifies the upper limit for the number of automatically created APPC devices on virtual controllers. The more APPC devices created, the longer it takes the system to do error recovery processing on the controller. The default value on this network attribute is 100. For every 100 new APPN locations that your system communicates, a new virtual APPN controller is created.

Note: Manually created devices may still be created if the VRTAUTODEV parameter value is less than the limit of 254.

Related concepts

[Work management](#)

Related reference

[Communications Management PDF](#)

Considerations for line configuration settings that can affect error recovery

Line descriptions describe the physical line connection and the data link protocol to be used between the IBM i server and the network.

Considerations for the automatic deletion of APPC device descriptions

The system is set to automatically delete APPC devices that were automatically created.

For virtual APPN devices, the default is 31,680 minutes.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Related concepts

[Considerations for controller configuration descriptions that can affect error recovery](#)

A controller description defines the adjacent systems in the network.

Related reference

[Communications Management PDF](#)

Link-level timers and retries

The configuration of link-level timers and retries can have a significant effect on network performance.

For a complete list of link-level timers and retries, see the appropriate protocol-specific publication.

Considerations for controller configuration descriptions that can affect error recovery

A controller description defines the adjacent systems in the network.

Related concepts

[APPC controller recovery summary](#)

The action the system takes when advanced program-to-program communications (APPC) controller descriptions go into recovery depends on the setting of many parameters. These tables can help you understand and select the appropriate configuration parameters to optimize system behavior when APPC controllers representing personal computer clients go into error recovery.

Related reference

[Considerations for the automatic deletion of APPC device descriptions](#)

The system is set to automatically delete APPC devices that were automatically created.

[Considerations for disconnect timer \(DSCTMR\) parameter for error recovery](#)

The disconnect timer (DSCTMR) parameter controls the amount of time to wait before a connection without activities is dropped, or the amount of time to delay the automatic disconnection.

Considerations for automatic delete device (AUTODLTDEV) parameter for error recovery

Device descriptions that are automatically created by the system can also be automatically deleted by the system. The default is to delete devices that were automatically created after one day (1440 minutes) of inactivity.

Specifying the default has the potential side-effect of having device descriptions deleted over the weekend. This can cause a system slow-down. For example, when users reconnect on monday morning (after a 48-hour period of system inactivity), they may discover that they need to re-create their device descriptions.

You might want to have the AUTODLTDEV parameter default to a value larger than 24 hours; 72 hours may be more appropriate to cover weekends. Use a model controller to change this value for an autogenerated controller description.

The default value for devices that are attached to automatically created APPN virtual controllers is 10,000 minutes.

Note: Using HPR or turning on the ALWVRTAPPN network attribute can also solve the problem of having multiple sets of configuration objects because HPR prevents multiple objects from being configured.

Considerations for the INLCNN parameter that can affect error recovery

During error recovery, the actions you take to recover the controller depend on whether the controller description was created with *DIAL or *ANS specified for the initial connection (INLCNN) parameter. You might want to change this parameter for error recovery.

The INLCNN parameter is on the CHGCTLxxx or the CRTCTLxxx commands.

When configuring the INLCNN parameter, use the following information:

- For IBM i-to-IBM i connections when either system might initiate a connection with the other, specify *DIAL for the INLCNN parameter.

Note: Whether the system actually attempts to dial depends on the setting of the Advanced Peer-to-Peer Networking (APPN), DIALIMMED, MINSWTSTS, and CTLOWN parameters along with the INLCNN parameter.

- For IBM i-to-PC connections, specify *ANS for the INLCNN parameter to avoid unnecessary recovery attempts when the personal computer shuts down.

Note: If the remote system never answers the dial attempt, consider changing the configuration to *ANS to avoid dial failures.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Related reference

[Communications Management PDF](#)

Considerations for the MINSWTSTS parameter that can affect error recovery

The default value for the Advanced Peer-to-Peer Networking (APPN) minimum switched status (MINSWTSTS) parameter is set to *VRYONPND. Specifying this parameter makes APPN controllers in vary

on pending status available for APPN route selection. You can consider changing this parameter value for error recovery.

The MINSWTSTS parameter is on the CHGCTLAPPC, the CHGCTLHOST, the CRTCTLAPPC, or the CRTCTLHOST commands.

When configuring the MINSWTSTS parameter, consider the following factors:

- Set the MINSWTSTS parameter to *VRYON to limit the routes that APPN recognizes as available. This prevents APPN from selecting routes that have a controller in varied on or pending status on one system, but in varied off or inoperative status on an adjacent system.
- Set the switched disconnect (SWTDSC) parameter to *NO when the MINSWTSTS parameter is set to *VRYON. This makes the connection appear like a leased connection. If you have a switched line, do not use the MINSWTSTS(*VRTON) parameter.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Related reference

[Communications Management PDF](#)

APPC controller recovery summary

The action the system takes when advanced program-to-program communications (APPC) controller descriptions go into recovery depends on the setting of many parameters. These tables can help you understand and select the appropriate configuration parameters to optimize system behavior when APPC controllers representing personal computer clients go into error recovery.

Table 1. When does the IBM i attempt to connect the remote system?

MINSWTSTS	INLCNN	APPN	CTLOWN	Power PC off (recovery)	Manual vary on
*VRYONPND	*DIAL	*YES	*SYS	Dial is attempted	Dial is attempted
*VRYONPND	*DIAL	*YES	*USER	Dial not attempted	Dial is attempted
*VRYONPND	*DIAL	*NO	*SYS	Configuration not allowed	
N/A	*DIAL	*NO	*USER	Dial not attempted	Dial is attempted
*VRYONPND	*ANS	*YES	*SYS	Dial not attempted	Dial not attempted
*VRYONPND	*ANS	*YES	*USER	Dial not attempted	Dial not attempted
*VRYONPND	*ANS	*NO	*SYS	Configuration not allowed	
N/A	*ANS	*NO	*USER	Dial not attempted	Dial not attempted

*Table 2. MINSWTSTS(*VRYON) affects on IBM i attempts to connect to the remote system*

APPN	INLCNN	CTLOWN	SWTDSC	Power PC off (recovery)	Manual vary on
*YES	*DIAL	*SYS	*YES	Configuration not allowed	
*YES	*DIAL	*SYS	*NO	Dial is attempted	Dial is attempted

<i>Table 2. MINSWTSTS(*VRYON) affects on IBM i attempts to connect to the remote system (continued)</i>					
APPN	INLCNN	CTLOWN	SWTDSC	Power PC off (recovery)	Manual vary on
*YES	*DIAL	*USER	*YES	Configuration not allowed	
*YES	*DIAL	*USER	*NO	Dial is attempted	Dial is attempted

Note: In all cases when a dial is attempted, when the remote system is using a PC with Client Access for Windows or IBM i Access for Windows installed, that dial attempt fails with the following message:

```
CPA57EF to QSYSOPR (Controller contact not successful)
```

Related concepts

[Considerations for controller configuration descriptions that can affect error recovery](#)

A controller description defines the adjacent systems in the network.

Considerations for disconnect timer (DSCTMR) parameter for error recovery

The disconnect timer (DSCTMR) parameter controls the amount of time to wait before a connection without activities is dropped, or the amount of time to delay the automatic disconnection.

The default value is 170 seconds. The range is 0 - 65536 seconds.

The DSCTMR parameter is on the CHGCTLxxx and CRTCTLxxx commands.

Related concepts

[Considerations for controller configuration descriptions that can affect error recovery](#)

A controller description defines the adjacent systems in the network.

Considerations for modes that can affect error recovery

A *mode description* is a system object that is created for communications devices to describe the session limits and the characteristics of the session. You can view, create, change, and work with mode descriptions using the Work with Mode Descriptions (WRKMODD) command.

Here are the session characteristics:

- The maximum number of sessions allowed
- The maximum number of conversations allowed
- The pacing value for incoming request
- The maximum size of request units
- Other controlling information for the session

The QPCSUPP (PC support) mode and QSERVER (server) modes are used by the IBM i Access for Windows licensed program.

Considerations for jobs that can affect error recovery

When a line or controller fails, the application programs are notified. Often you must end those jobs that were running over the line and controller. You must start those jobs again after the communications resource is recovered. The ending of jobs (particularly abnormal job ending) is an extremely complex transaction regarding performance.

Related reference

[Considerations for job logs that can affect communications error recovery](#)

You need to consider whether to generate job logs when an error condition occurs and the active jobs are ended.

Considerations for the CMNRCYLMT parameter that can affect error recovery

The QCMNRCYLMT system value or the recovery limits (CMNRCYLMT) parameter on the configuration object controls automatic communications error recovery.

The CMNRCYLMT parameter is on the CHGCTLxxx, CHGLINxxx, CRTCTLxxx, and CRTLINxxx commands.

These parameter values contain two related numbers that you can set:

- The number of second-level recovery attempts automatically performed by the system (count limit)
- The length of time (time interval) in which the specified number of second-level recoveries can occur.

The default value on CMNRCYLMT on lines and controllers is two retries within 5 minutes (2 5).

To configure the CMNRCYLMT parameter, consider the following factors:

- If automatic communication fails for personal computers on a local area network (LAN) and the IBM i attempts to recover the connection, this places unnecessary work on the system.

Note: If automatic communications error recovery is not used, manual recovery is necessary, which requires operator intervention. A good compromise is to set the automatic recovery limits to just one retry.

- Use a count limit 0 and a time interval of more than 0 to turn off second-level error recovery. Turning off second-level recovery may cause the devices and controllers to go into recovery pending (RCYPND) state. A message is sent to QSYSOPR, or the configured message queue, that requires operator intervention. Use manual recovery to either respond to the message in QSYSOPR, or the configured message queue, or vary the objects off and back on.

Note: First-level error recovery is still done.

- Write applications that can determine if a failure has occurred, and then handle the errors.
 - Monitor the error messages in QSYSOPR, or the configured message queue, when they occur and handle the condition.
 - Monitor the status of the configuration objects by using Retrieve Configuration Status (QDCRCFGS) and List Configuration Descriptions (QDCLCFGD) application programming interfaces (API).

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Related reference

[Communications Management PDF](#)

[Considerations for communications-related system values](#)

System values, such as system date and library list, control information for the operation of certain parts of the system. You can change the system value to define your working environment.

Considerations for prestart job entries that can affect APPC error recovery

Using prestart jobs reduces the startup time of the connection. You might want to change the prestart job entries. The prestart job entries depend on the usage of your system and the servers during error recovery situations.

Prestart jobs are reused rather than ended. After an error, users are able to reconnect more quickly. Prestart job entries are included with the system for QCMN, QBASE, and QSERVER for these server jobs.

Change the prestart jobs entries as appropriate for your environment.

- Consider the following parameters and their values:
 - STRJOBS(*YES and *NO)
 - INLJOBS
 - THRESHOLD

- ADLJOBS
- MAXJOBS
- Use the INLJOB parameter to make a larger number of jobs available for the following reasons:
 - You have many users who will connect to the system.
 - The connect processing is to be done as quickly as possible.
- Make sure that the THRESHOLD value is higher than the total number of active users.
- Make sure that the ADLJOBS value is higher than the number of jobs that is used.

Note: As user applications are developed, consider using prestart jobs to reduce program start request startup processing.

Tip: Displaying the inactive prestart jobs

To display the inactive prestart jobs, press F14 on the WRKACTJOB display.

This display can be used to include jobs that are typically not shown on the WRKACTJOB displays. Inactive prestart jobs show a status of PSRW (program start request wait).

Work entries

In a subsystem description, work entries are defined to identify the sources from which jobs can be started in that subsystem.

The types of work entries are as follows:

Autostart job entry

Specifies a job that is automatically started when the subsystem is started.

Workstation entry

Specifies one workstation or a group of workstations from which interactive jobs can be started.

Job queue entry

Specifies one of the job queues from which the subsystem can select batch jobs. A batch job is a job that can run independently of a user at a workstation.

Communications entry

Specifies one or a group of communications device descriptions from which communications batch jobs can be started. Communications batch jobs do not use job queues.

Prestart job entry

Identifies an application program to be started that waits for incoming allocation requests.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Related reference

[Communications Management PDF](#)

[IBM i Access Client Solutions](#)

[Considerations for job logs that can affect communications error recovery](#)

You need to consider whether to generate job logs when an error condition occurs and the active jobs are ended.

Considerations for job logs that can affect communications error recovery

You need to consider whether to generate job logs when an error condition occurs and the active jobs are ended.

Producing job logs can use considerable system resources, especially during error recovery in which many jobs end at one time. In this case, it might be better to not generate job logs. However, if you do not produce job logs, you will not have any data for analysis if something goes wrong. There is a trade-off to be made.

To make your system generate less or none job logs, configure the system as follows:

- Set the DEVRCYACN parameter to *ENDJOBNO LIST. There is also a QDEVRCYACN system value for ease of configuration.

Note: The QDSCJOBITV system value determines when the unused disconnected jobs are ended.

- Change the job description (or the job itself through an initial program for the user profile) to LOGLVL(4 0 *NOLIST). With this description, the job logs are not generated if jobs end normally, but are generated if jobs end abnormally.

Note: Disconnected jobs also use resources. The System Work Control Block Table can grow, which has other side effects. Do not disconnect from jobs to which you will never reconnect.

If some of your users reconnect after a failure, however, the disconnect options can offer increased performance for them.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Related concepts

[Considerations for prestart job entries that can affect APPC error recovery](#)

Using prestart jobs reduces the startup time of the connection. You might want to change the prestart job entries. The prestart job entries depend on the usage of your system and the servers during error recovery situations.

[Work management](#)

Related reference

[Communications Management PDF](#)

[Considerations for jobs that can affect error recovery](#)

When a line or controller fails, the application programs are notified. Often you must end those jobs that were running over the line and controller. You must start those jobs again after the communications resource is recovered. The ending of jobs (particularly abnormal job ending) is an extremely complex transaction regarding performance.

[Considerations for communications-related system values](#)

System values, such as system date and library list, control information for the operation of certain parts of the system. You can change the system value to define your working environment.

The Change System Job (CHGSYSJOB) command

You can change the run priority of a system job by using the Change System Job (CHGSYSJOB) command.

The following are system jobs of interest for communications recovery.

- QCMNARB01 through QCMNARB99
- QSYSCOMM1

In general, these system jobs should be run at their default, system-provided priority. However, if one of these jobs begins using large amounts of CPU, and is affecting other work on the system, it is possible to lower its priority. Note that this might result in queued-up work for that job.

Configuring APPC with VTAM

You need to coordinate the Virtual Telecommunications Access Method (VTAM®) and advanced program-to-program communication (APPC) configuration objects when configuring APPC with VTAM.

1. The controller description is equivalent to the IBM Network Control Program and Virtual Telecommunications Access Method (NCP/VTAM) PU macros. You can find the information in a controller description in the Extended Services Communication Manager Partner LU profile.
2. The device description is equivalent to the NCP/VTAM logical unit (LU) macro. You can find the information of a device description in Extended Services Communications Manager Partner LU and LU profiles.

- The mode description is equivalent to the NCP/VTAM mode tables. You can find the information in a mode description in the Extended Services Communications Manager Transmission Service Mode profile and Initial Session Limits profile.

Examples: APPC, APPN, and HPR configuration

When configuring APPC, APPN and HPR, you might want to see these examples, which illustrate the configuration of APPN and HPR in different scenarios.

Note: By using the code examples, you agree to the terms of the [“Code license and disclaimer information”](#) on page 53.

Examples: APPN configuration

These examples illustrate the APPN configuration in different scenarios.

Note:

- In all of the examples, default values are used for all parameters not explicitly defined.
- The name assigned to each description that is created is the same as the name of the destination that is being defined in that description. For example, the line description configured in New York for the connection to Los Angeles is LOSANGEL.
- Names (such as location names), telephone numbers, exchange identifiers, and other values that shown in the examples are for illustration only. The values you assign to your configuration are dependent on your network requirements.

Example: Two systems as end nodes using APPN

In Figure 4, systems A and B are both configured as end nodes in the network attributes. The only APPN-specific parameter that must be configured is the remote control-point name in the controller description. A device description is not a requirement for an APPN configuration.

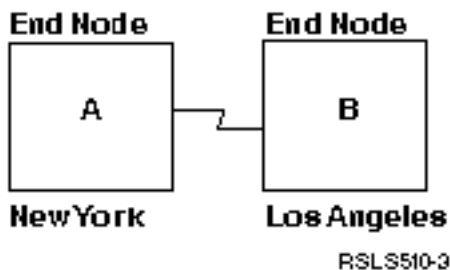


Figure 1. Two-system APPN network

See the following information to determine the configuration requirements for each system in Figure 4.

Configuring system A (New York) as an end node

These CL commands are used to define the configuration for system A (NEWYORK).

The example shows the commands as used within a CL program. You can also perform the configuration using the configuration menus.

Note: By using the following code examples, you agree to the terms of the [“Code license and disclaimer information”](#) on page 53.

```

/*****/
/*
/* MODULE: NYLAAPPN LIBRARY: PUBSCFGS */
/* */
/* LANGUAGE: CL */
/* */
/* FUNCTION: CONFIGURES APPN ENDNODES AS FOLLOWS: */

```

```

/*
/*          NEWYORK  \-----\  LOSANGEL
/*          \-----/
/*
/*          (THIS IS NEWYORK TO LOSANGEL)
/*
/*****
/*****
/*          NEWYORK TO LOSANGEL
/*****
/*****
/* Change network attributes for NEWYORK */
CHGNETA  LCLNETID(APPN) LCLCPNAME(NEWYORK)
          LCLLOCNAME(NEWYORK) NODETYPE(*ENDNODE)
          ALWVRTAPPN(*YES) ALWHPRTWR(*YES)
/* Create line description for NEWYORK */
CRTLINETH LIND(NEWYORK) RSRNAME(CMN11)
/* Add TCP interface to line description NEWYORK */
ADDTCPIFC INTNETADR('192.168.41.11') LIND(NEWYORK)
          SUBNETMASK('255.255.255.0')
/* Create controller description for NEWYORK to
          LOSANGEL */
CRTCTLAPPC CTLD(LOSANGEL) LINKTYPE(*HPRIP)
          RMTNETID(APPN) RMTCPNAME(LOSANGEL)
          NODETYPE(*CALC)
          RMTINTNETA('192.168.31.42')
          LCLINTNETA('192.168.41.11')
          LDLCLNKSPD(*MAX)

```

Note: The 192.168.x.x addresses are private IP addresses and cannot be routed over a public network unless NAT/proxy servers are used.

Related reference

[Change Network Attributes \(CHGNETA\)](#)

[Create Line Desc \(Ethernet\) \(CRTLINETH\)](#)

[Add TCP/IP Interface \(ADDTCPIFC\)](#)

[Create Ctl Desc \(APPC\) \(CRTCTLAPPC\)](#)

Configuring system B (Los Angeles) as an end node

These CL commands define the configuration for System B (LOSANGEL).

The example shows the commands as used within a CL program. You can also perform the configuration using the configuration menus.

Note: By using the following code examples, you agree to the terms of the [“Code license and disclaimer information”](#) on page 53.

```

/*****
/*
/* MODULE:  LANYAPPN          LIBRARY:  PUBSCFGS
/*
/* LANGUAGE:  CL
/*
/* FUNCTION:  CONFIGURES APPN ENDNODES AS FOLLOWS:
//*/
/*          NEWYORK  \-----\  LOSANGEL
/*          \-----/
/*
/*          (THIS IS LOSANGEL TO NEWYORK)
/*
/*****
/*****
/*          LOSANGEL TO NEWYORK
/*****
/*****
/* Change network attributes for LOSANGEL */
CHGNETA  LCLNETID(APPN) LCLCPNAME(LOSANGEL)
          LCLLOCNAME(LOSANGEL) NODETYPE(*ENDNODE)
          ALWVRTAPPN(*YES) ALWHPRTWR(*YES)
/* Create line description for LOSANGEL to NEWYORK */
CRTLINETH LIND(LOSANGEL) RSRNAME(CMN11)
/* Add TCP interface to line description to NEWYORK */
ADDTCPIFC INTNETADR('192.168.31.42') LIND(NEWYORK)
          SUBNETMASK('255.255.255.0')
/* Create controller description for LOSANGEL to
          NEWYORK */
CRTCTLAPPC CTLD(NEWYORK) LINKTYPE(*HPRIP)
          RMTNETID(APPN) RMTCPNAME(NEWYORK)

```

```

NODETYPE(*CALC)
RMTINTNETA('192.168.41.11')
LCLINTNETA('192.168.31.42')
LDLCLNKSPD(*MAX)

```

Note: The 192.168.x.x addresses are private IP addresses and cannot be routed over a public network unless NAT/proxy servers are used.

Related reference

- [Change Network Attributes \(CHGNETA\)](#)
- [Create Line Desc \(Ethernet\) \(CRTLINETH\)](#)
- [Add TCP/IP Interface \(ADDTCPIFC\)](#)
- [Create Ctl Desc \(APPC\) \(CRTCTLAPPC\)](#)

Example: Three systems using APPN

In Figure 6, A and B are end nodes. The network node must configure its network attributes to reflect that it is a network node.

Each system must configure the remote control-point name in the controller description that represents the adjacent system. Also, A and B must indicate in the controller description for the network node that it can be a network node. A and B must add the network node to the server list in network attributes so that the network node might act as a network server for both end nodes.

Note: Neither end node needs to configure any information about the other end node.

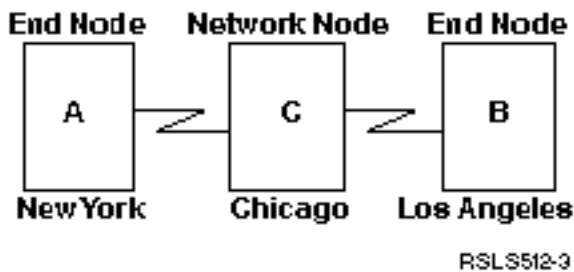


Figure 2. Three-system APPN network

Configuring system A (New York)

These CL commands define the configuration for the system that is identified as New York (system A). The examples show these commands as used within a CL program. You can also perform the configuration using the configuration menus.

Note: By using the following code examples, you agree to the terms of the [“Code license and disclaimer information”](#) on page 53.

```

/*****
/*
/*  MODULE:  NYCHENNN                LIBRARY:  PUBSCFGS          */
/*
/*  LANGUAGE:  CL                    */
/*
/*  FUNCTION:  CONFIGURES APPN EN-NN-EN AS FOLLOWS:          */
/*
/*
/*           NEWYORK {-----} CHICAGO {-----} LOSANGEL    */
/*
/*           (THIS IS NEWYORK TO CHICAGO)                    */
/*
/*
/*
/*****
/*****
/*
/*           NEWYORK TO CHICAGO
/*****
/*
/*  Change network attributes for NEWYORK */
CHGNETA  LCLNETID(APPN) LCLCPNAME(NEWYORK)

```



```

        LCLLOCNAME(NEWYORK) NODETYPE(*ENDNODE)
        NETSERVER((APPN CHICAGO))
        ALWVRTAPPN(*YES) ALWHPRTWR(*YES)
/* Create remote configuration list for NEWYORK
   where '3BD29F' is the location password used
   when establishing sessions */
CRTCFGL TYPE(*APPNRMT) APPNRMTE((LOSANGEL APPN
        NEWYORK LOSANGEL APPN 3BD29F *YES *NO *NO *NO
        'RMT LOC of NEWYORK'))
/* Create line description for NEWYORK to CHICAGO */
CRTLINETH LIND(NEWYORK) RSRNAME(CMN11)
/* Add TCP interface to line description NEWYORK */
ADDTCPIFC INTNETADR('192.168.1.42') LIND(NEWYORK)
        SUBNETMASK('255.255.255.0')
/* Create controller description for NEWYORK to
        CHICAGO */
CRTCTLAPPC CTLD(CHICAGO) LINKTYPE(*HPRIP)
        RMTNETID(APPN) RMTCPNAME(CHICAGO)
        NODETYPE(*NETNODE)
        RMTINTNETA('192.168.2.11')
        LCLINTNETA('192.168.1.42')
        LDCLLNKSPD(*MAX)

```

Note: The 192.168.x.x addresses are private IP addresses and cannot be routed over a public network unless NAT/proxy servers are used.

Related reference

[Change Network Attributes \(CHGNETA\)](#)

[Create Configuration List \(CRTCFGL\)](#)

[Create Line Desc \(Ethernet\) \(CRTLINETH\)](#)

[Add TCP/IP Interface \(ADDTCPIFC\)](#)

[Create Ctl Desc \(APPC\) \(CRTCTLAPPC\)](#)

Configuring system B (Los Angeles)

These CL commands are used to define the configuration for the system that is identified as LOSANGEL (system B). The examples show these commands as used within a CL program. You can also perform the configuration using the configuration menus.

Note: By using the following code examples, you agree to the terms of the [“Code license and disclaimer information”](#) on page 53.

```

/*****
/*
/* MODULE: LACHENNN LIBRARY: PUBSCFGS */
/*
/* LANGUAGE: CL */
/*
/* FUNCTION: CONFIGURES APPN EN-NN-EN AS FOLLOWS: */
/*
/*
/* NEWYORK {-----} CHICAGO {-----} LOSANGEL */
/*
/* (THIS IS LOSANGEL TO CHICAGO) */
/*
/*
/*
/*****
/*****
/* LOSANGEL TO CHICAGO */
/*****
/* Change network attributes for LOSANGEL */
CHGNETA LCLNETID(APPN) LCLCPNAME(LOSANGEL)
        LCLLOCNAME(LOSANGEL) NODETYPE(*ENDNODE)
        NETSERVER((APPN CHICAGO))
        ALWVRTAPPN(*YES) ALWHPRTWR(*YES)
/* Create remote configuration list for LOSANGEL to
   New York where '3BD29F' is the location password
   used when establishing sessions */
CRTCFGL TYPE(*APPNRMT) APPNRMTE((NEWYORK APPN
        LOSANGEL NEWYORK APPN 3BD29F *YES *NO *NO *NO
        'RMT LOC of LOSANGEL'))
/* Create line description for LOSANGEL to CHICAGO */
CRTLINETH LIND(LOSANGEL) RSRNAME(CMN41)

```

```

/* Add TCP interface to line description LOSANGEL */
ADDTCPIFC INTNETADR('192.168.3.22') LIND(LOSANGEL)
          SUBNETMASK('255.255.255.0')
/* Create controller description for LOSANGEL to
          CHICAGO */
CRTCTLAPPC CTLD(CHICAGO) LINKTYPE(*HPRIP)
          RMTNETID(APPN) RMTCPNAME(CHICAGO)
          NODETYPE(*NETNODE)
          RMTINTNETA('192.168.2.11')
          LCLINTNETA('192.168.3.22')
          LDLCLNKSPD(*MAX)

```

Note: The 192.168.x.x addresses are private IP addresses and cannot be routed over a public network unless NAT/proxy servers are used.

Related reference

[Change Network Attributes \(CHGNETA\)](#)

[Create Configuration List \(CRTCFGL\)](#)

[Create Line Desc \(Ethernet\) \(CRTLINETH\)](#)

[Add TCP/IP Interface \(ADDTCPIFC\)](#)

[Create Ctl Desc \(APPC\) \(CRTCTLAPPC\)](#)

Configuring system C (Chicago)

These CL commands define the configuration for the system that is identified as CHICAGO (system C). The example shows the commands as used within a CL program. You can also perform the configuration using the configuration menus.

Note: By using the following code examples, you agree to the terms of the [“Code license and disclaimer information”](#) on page 53.

```

/*****/
/*
/* MODULE:  CHNYCHLA                LIBRARY:  PUBSCFGS                */
/*
/* LANGUAGE:  CL                    */
/*
/* FUNCTION:  CONFIGURES APPN NETWORK:                */
/*
/*          NEWYORK  \-----\  CHICAGO  \-----\  LOSANGEL
/*          \-----/          \-----/
/*
/*          (THIS IS CHICAGO TO NEWYORK AND LOSANGEL)
/*
/*
/*
/*
/*****/
PGM

          /* Change network attributes for CHICAGO */
          CHGNETA  LCLNETID(APPN) LCLCPNAME(CHICAGO)
                  LCLLOCNAME(CHICAGO) NODETYPE(*NETNODE)
                  ALWVRTAPPN(*YES) ALWHPRTWR(*YES)
/*****/
/*          CHICAGO TO NEWYORK                */
/*****/
          /* Create line description for CHICAGO to NEWYORK */
          CRTLINETH LIND(CHICAGO) RSRCPNAME(CMN12)
          /* Add TCP interface to line description CHICAGO */
          ADDTCPIFC INTNETADR('192.168.3.22') LIND(CHICAGO)
                  SUBNETMASK('255.255.255.0')
          /* Create controller description for CHICAGO to
                  NEWYORK */
          CRTCTLAPPC CTLD(NEWYORK) LINKTYPE(*HPRIP)
                  RMTNETID(APPN) RMTCPNAME(NEWYORK)
                  NODETYPE(*ENDNODE)
                  RMTINTNETA('192.168.1.42')
                  LCLINTNETA('192.168.2.11')
                  LDLCLNKSPD(*MAX)
/*****/
/*          CHICAGO TO LOSANGEL                */
/*****/
          /* Create controller description for CHICAGO to
                  LOSANGEL */
          CRTCTLAPPC CTLD(LOSANGEL) LINKTYPE(*HPRIP)

```

```
RMTNETID (APPN) RMTCPNAME (LOSANGEL)
NODETYPE (*ENDNODE)
RMTINTNETA ('192.168.3.22')
LCLINTNETA ('192.168.2.11')
LDLCLNKSPD (*MAX)
```

Note: The 192.168.x.x addresses are private IP addresses and cannot be routed over a public network unless NAT/proxy servers are used.

Related reference

[Change Network Attributes \(CHGNETA\)](#)

[Create Line Desc \(Ethernet\) \(CRTLINETH\)](#)

[Add TCP/IP Interface \(ADDTCPIFC\)](#)

[Create Ctl Desc \(APPC\) \(CRTCTLAPPC\)](#)

Example: Two APPN networks with different network IDs linked together

Figure 7 shows two APPN networks that are linked together by network nodes.

The network with the LCLNETID of NEWNET is a simple connection of one end node to one network node. Network node B might act as a network server providing routing services for node A. Although there are no other nodes in the NEWNET network, there is a need for nodes A and B to communicate with the nodes in network APPN. To accomplish this, network node B is connected to network node NN1 in the APPN network. Node B must have a line description and a controller description created to identify node A, and a line description and a controller description to identify node NN1.

The network with the LCLNETID of APPN is similar to NEWNET; with the exception that NN2 is a network node instead of an end node. In order for NN1 and NN2 to communicate with the nodes in NEWNET, NN1 must have two line descriptions, and two controller descriptions created. These identify both node B and node NN2.

After node B and node NN1 are identified to each other as adjacent nodes, all nodes in either network can communicate through nodes B and NN1.

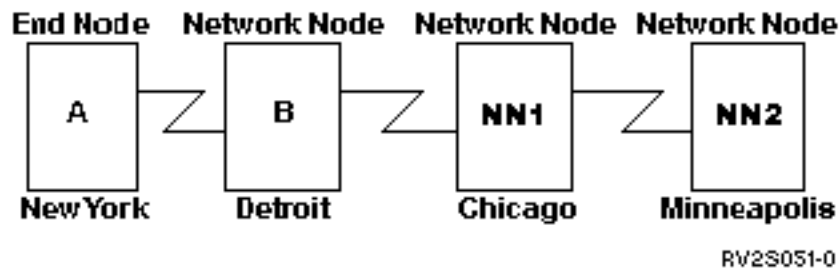


Figure 3. Two APPN networks linked by network nodes

See the following to determine the configuration requirements for each system in Figure 7.

Configuring system A (New York)

These CL commands are used to define the configuration for the system identified as NEWYORK (system A). The examples show these commands as used within a CL program. You can also perform the configuration using the configuration menus.

Note: By using the following code examples, you agree to the terms of the [“Code license and disclaimer information”](#) on page 53.

```

/*****/
/*
/* MODULE: NYCINT LIBRARY: PUBSCFGS */
/*
/* LANGUAGE: CL */
/*
/* FUNCTION: CONFIGURES APPN EN-NN AS FOLLOWS: */
/*
/*
/*****/
```

```

/*          NEWYORK /-----\ DETROIT          */
/*          \-----/                          */
/*                                              */
/*          (THIS IS NEWYORK TO DETROIT)        */
/*                                              */
/*                                              */
/*                                              */
/*****
/*****
/*          NEWYORK TO DETROIT          */
/*****
/*****
/* Change network attributes for NEWYORK */
CHGNETA  LCLNETID(NEWNET) LCLCPNAME(NEWYORK)
         LCLLOCNAME(NEWYORK) NODETYPE(*ENDNODE)
         NETSERVER((NEWNET DETROIT))
         ALWVRTAPPN(*YES) ALWHPRTWR(*YES)
/* Create line description for NEWYORK */
CRTLINETH LIND(NEWYORK) RSRCPNAME(CMN11)
/* Add TCP interface to line description NEWYORK */
ADDTCPIFC INTNETADR('192.168.1.43') LIND(NEWYORK)
         SUBNETMASK('255.255.255.0')
/* Create controller description for NEWYORK to
         DETROIT */
CRTCTLAPPC CTLD(DETROIT) LINKTYPE(*HPRIP)
         RMTNETID(NEWNET) RMTCPNAME(DETROIT)
         RMTINTNETA('192.168.4.2')
         LCLINTNETA('192.168.1.43')
         LDLCCLKSPD(*MAX)
         NODETYPE(*NETNODE)

```

Note: The 192.168.x.x addresses are private IP addresses and cannot be routed over a public network unless NAT/proxy servers are used.

Related reference

[Change Network Attributes \(CHGNETA\)](#)

[Create Line Desc \(Ethernet\) \(CRTLINETH\)](#)

[Add TCP/IP Interface \(ADDTCPIFC\)](#)

[Create Ctl Desc \(APPC\) \(CRTCTLAPPC\)](#)

Configuring system B (Detroit)

These CL commands define the configuration for the system that is identified as DETROIT (system B). The example shows the commands as used within a CL program. You can also perform the configuration using the configuration menus.

Note: By using the following code examples, you agree to the terms of the [“Code license and disclaimer information”](#) on page 53.

```

/*****
/*
/*          MODULE:  DETRINT          LIBRARY:  PUBSCFGS          */
/*
/*          LANGUAGE:  CL          */
/*
/*          FUNCTION:  CONFIGURES APPN NETWORK:          */
/*
/*
/*          NEWYORK /-----\ DETROIT /-----\ CHICAGO          */
/*          \-----/          \-----/          */
/*
/*          (THIS IS DETROIT TO NEWYORK AND CHICAGO)        */
/*
/*
/*
/*****
/* Change network attributes for DETROIT */
CHGNETA  LCLNETID(NEWNET) LCLCPNAME(DETROIT)
         LCLLOCNAME(DETROIT) NODETYPE(*NETNODE)
         ALWVRTAPPN(*YES) ALWHPRTWR(*YES)
/*****
/*          DETROIT TO NEWYORK          */
/*****
/*****
/* Create line description for DETROIT to NEWYORK */
CRTLINETH LIND(DETROIT) RSRCPNAME(CMN12)
/* Add TCP interface to line description DETROIT */
ADDTCPIFC INTNETADR('192.168.4.2') LIND(DETROIT)

```

```

                SUBNETMASK('255.255.255.0')
/* Create controller description for DETROIT to
                                NEWYORK */
CRTCTLAPPC CTLD(NEWYORK) LINKTYPE(*HPRIP)
                RMTNETID(NEWNET) RMTCPNAME(NEWYORK)
                NODETYPE(*ENDNODE)
                RMTINTNETA('192.168.1.43')
                LCLINTNETA('192.168.4.2')
                LDLCCLKSPD(*MAX)
/*****
/*
                DETROIT TO CHICAGO
                */
/*****
/* Create controller description for DETROIT to
                                CHICAGO */
CRTCTLAPPC CTLD(CHICAGO) LINKTYPE(*HPRIP)
                RMTNETID(APPN) RMTCPNAME(CHICAGO)
                NODETYPE(*NETNODE)
                RMTINTNETA('192.168.2.12')
                LCLINTNETA('192.168.4.2')
                LDLCCLKSPD(*MAX)

```

Note: The 192.168.x.x addresses are private IP addresses and cannot be routed over a public network unless NAT/proxy servers are used.

Related reference

[Change Network Attributes \(CHGNETA\)](#)

[Create Line Desc \(Ethernet\) \(CRTLINETH\)](#)

[Add TCP/IP Interface \(ADDTCPIFC\)](#)

[Create Ctl Desc \(APPC\) \(CRTCTLAPPC\)](#)

Configuring system NN1 (Chicago)

These CL commands are used to define the configuration for the system that is identified as CHICAGO (system NN1). The examples show these commands as used within a CL program. You can also perform the configuration using the configuration menus.

Note: By using the following code examples, you agree to the terms of the [“Code license and disclaimer information”](#) on page 53.

```

/*****
/*
/* MODULE:  CHICINT                LIBRARY:  PUBSCFGS                */
/*
/* LANGUAGE:  CL                    */
/*
/* FUNCTION:  CONFIGURES APPN NETWORK:                */
/*
/*           THIS IS: CHICAGO TO MPLS                */
/*           CHICAGO TO DETROIT                    */
/*
/*
/*
/*
/*
/*
/*****
PGM

/* Change network attributes for CHICAGO */
CHGNETA  LCLNETID(APPN) LCLCPNAME(CHICAGO)
                LCLLOCNAME(CHICAGO) NODETYPE(*NETNODE)
                ALWVRTAPPN(*YES) ALWHPRTWR(*YES)
/*****
/*
                CHICAGO TO MPLS
                */
/*****
/* Create line description for CHICAGO to MPLS */
CRTLINETH LIND(CHICAGO) RSRCPNAME(CMN21)
/* Add TCP interface to line description CHICAGO */
ADDTCPIFC INTNETADR('192.168.2.12') LIND(CHICAGO)
                SUBNETMASK('255.255.255.0')
/* Create controller description for CHICAGO to MPLS */
CRTCTLAPPC CTLD(MPLSL) LINKTYPE(*HPRIP)
                RMTNETID(APPN) RMTCPNAME(MPLS)
                NODETYPE(*NETNODE)
                RMTINTNETA('192.168.5.20')
                LCLINTNETA('192.168.2.12')
                LDLCCLKSPD(*MAX)
/*****
/*
                CHICAGO TO DETROIT
                */

```

```

/*****
/* Create controller description for CHICAGO to DETROIT */
CRTCTLAPPC CTLD(DETROIT) LINKTYPE(*HPRIP)
RMTNETID(NEWNET) RMTCPNAME(DETROIT)
STNADR(01) NODETYPE(*NETNODE)
RMTINTNETA('192.168.4.2')
LCLINTNETA('192.168.2.12')
LDLCLNKSPD(*MAX)

```

Note: The 192.168.x.x addresses are private IP addresses and cannot be routed over a public network unless NAT/proxy servers are used.

Related reference

[Change Network Attributes \(CHGNETA\)](#)

[Create Line Desc \(Ethernet\) \(CRTLINETH\)](#)

[Add TCP/IP Interface \(ADDTCPIFC\)](#)

[Create Ctl Desc \(APPC\) \(CRTCTLAPPC\)](#)

Configuring NN2 (Minneapolis)

This example program shows the CL commands used to define the configuration for the system that is identified as MPLS (NN2). The example shows these commands used within a CL program. You can also set the configuration using the configuration menus.

Note: By using the following code examples, you agree to the terms of the [“Code license and disclaimer information”](#) on page 53.

```

/*****
/*
/* MODULE: MPLSINT LIBRARY: PUBSCFGS */
/*
/* LANGUAGE: CL */
/*
/* FUNCTION: CONFIGURES APPN NETWORK: */
/*
/* THIS IS: MPLS TO CHICAGO */
/*
/*
/*
/*****
/* Change network attributes for MPLS */
CHGNETA LCLNETID(APPN) LCLCPNAME(MPLS)
LCLLOCNAME(MPLS) NODETYPE(*NETNODE)
ALWVRTAPPN(*YES) ALWHPRTWR(*YES)
/*****
/* MPLS TO CHICAGO */
/*****
/* Create line description for MPLS */
CRTLINETH LIND(MPLS) RSRCPNAME(CMN22)
/* Add TCP interface to line description MPLS */
ADDTCPIFC INTNETADR('192.168.2.12') LIND(MPLS)
SUBNETMASK('255.255.255.0')
/* Create controller description for MPLS to CHICAGO */
CRTCTLAPPC CTLD(CHICAGO) LINKTYPE(*HPRIP)
RMTNETID(APPN) RMTCPNAME(CHICAGO)
NODETYPE(*NETNODE)

```

Note: The 192.168.x.x addresses are private IP addresses and cannot be routed over a public network unless NAT/proxy servers are used.

Related reference

[Change Network Attributes \(CHGNETA\)](#)

[Create Line Desc \(Ethernet\) \(CRTLINETH\)](#)

[Add TCP/IP Interface \(ADDTCPIFC\)](#)

[Create Ctl Desc \(APPC\) \(CRTCTLAPPC\)](#)

Optimizing APPN and HPR communication performance

If you are responsible for network administration, you might be concerned with the speed at which computers throughout that network can exchange data.

Fortunately, you can manage the ability of a network to perform work and remain robust. The higher the performance, the more jobs a network can handle.

In addition, you must consider the individual components that make up the systems in your network, in relation to the environment in which that system is running. If you have decided to configure an APPN and HPR network, review the following topics:

Related concepts

Considerations in designing an APPN and HPR network

When you design your network, consider these factors to optimize performance.

Performance considerations for APPN and HPR

Here are the factors that can affect the performance of the APPN and HPR protocols.

- Transmission priority

When you create a class-of-service description, you can define one of three transmission priorities for each class of service. The transmission priority (TMSPTY) parameter specifies the transmission priority for the class of service to be high, medium, or low.

The transmission priority you specify is carried in the session activation request at session establishment. The transmission priority allows each logical unit on the session and each routing entry along the session path to store the same transmission priority. By assigning an appropriate mode (which includes a class-of-service) at session activation time, you can ensure a better response time for the applications that require it. Generally, interactive traffic should have a high priority and batch traffic a low priority.

- Route addition resistance

Route addition resistance (RAR) is a relative value that indicates how desirable one network node is, as compared to other network nodes, for having intermediate sessions routed through it.

Changing this value and working with the different class-of-service descriptions can control route sessions.

The RAR value is defined in the network attributes for the local IBM i system.

- Pacing values: See the Pacing (INPACING, OUTPACING, MAXINPACING) parameters information for pacing considerations.

- Session activation considerations

When a session is requested to a remote location that matches a network node control-point name, a directory search is not performed by the node that calculates the route. This is true if the session request is being started by a user on the network node, or on an end node that the network node is providing services for. Session start requests for remote locations in end nodes and remote locations in network nodes that do not match the control-point name of the network nodes take longer. These session start requests take longer because the directory search needs to be sent and the replies need to be received.

- Maximum intermediate sessions

The Change Network Attributes (CHGNETA) command specifies the maximum number of intermediate sessions that are allowed on a network node. When the number of intermediate sessions reaches 90% of the maximum value, the node is marked as congested. A node that is congested may or may not be used for intermediate sessions that depend on the class-of-service definition. The node is not congested when the number of intermediate sessions drops below 80% of the configured value. Also, if the maximum number of intermediate sessions is reached (100%), then intermediate sessions will not be allowed through this network node until the value drops. You can limit the effect of intermediate sessions on local processing by setting an appropriate value.

- Segmentation and reassembly

Note: Communications input/output adapters (IOAs), such as Gigabit Ethernet, perform segmentation in the server CPU. Gigabit Ethernet adapters do not automatically support SNA. Enterprise Extender is required to allow SNA data to flow over a Gigabit adapter.

With APPN, any network congestion control is handled on a hop-by-hop basis by using pacing values. It is possible to overload connections in an APPN environment. A particular system might receive more data over a communications link than it can handle based on buffer space. The system requires the node that sends the data to retransmit all of the frames that were sent following the last successfully acknowledged frame. This retransmission occurs at the data link control (DLC) layer.

Note: HPR has little IOP assistance. Much of the segmentation and reassembly is done in the server CPU.

- Error recovery

APPN requires link-level error recovery to cause retransmission of lost frames. This link-level error recovery can only survive short and temporary outages (several seconds). If a link outage or node outage occurs, that is of longer duration, APPN has no recovery mechanisms for keeping the affected sessions active. The applications must handle any session recovery.

The following matrix shows how HPR traffic is supported between two systems that are based on their HPR link-level error recovery settings. The HPR link-level error settings are exchanged between the systems:

System 1	System 2		
	No link-level ERP allowed	Link-level ERP required	Prefer no link-level ERP but might run using link-level ERP
No link-level ERP allowed	HPR supported (no ERP)	HPR not used	HPR supported (no ERP)
Link-level ERP required	HPR not used	HPR supported (uses ERP)	HPR supported (uses ERP)
Prefer no link-level ERP but might run using link-level ERP	HPR supported (no ERP)	HPR supported (uses ERP)	HPR supported (no ERP)

Related concepts

Communications optimization using high-performance routing

High-performance routing (HPR) is the next evolution of Advanced Peer-to-Peer Networking (APPN). HPR differs from APPN in the areas of transport, intermediate session routing, congestion control, and error recovery.

Pacing (INPACING, OUTPACING, MAXINPACING) parameters

Pacing is required if there is a possibility of overflowing data buffers internal to the controller or to the host system. This typically occurs if the controller or host must pass the data to a device that operates at a slow speed. If the host system receives a pacing response, it sends more data frames, up to the window size, to the controller.

Communications optimization using high-performance routing

High-performance routing (HPR) is the next evolution of Advanced Peer-to-Peer Networking (APPN). HPR differs from APPN in the areas of transport, intermediate session routing, congestion control, and error recovery.

HPR has many functional aspects common with APPN, such as configuring adjacent stations, search processing, and route computation.

HPR supports a key availability enhancement that is called non-disruptive path switching. This function provides the ability to recover from link or node outages without having session failures. This makes the outage transparent to the application. The application may experience a response time delay while data traffic is being rerouted. On IBM i, the amount of time the system takes to establish a new path, or re-establish the original failed route path is configurable. This error recovery feature is the key difference between APPN and HPR.

HPR can support the non-disruptive path switching feature because of an enhanced data transport mechanism that is called Rapid Transport Protocol (RTP). RTP is the data transport protocol that is used between a pair of systems that support the HPR RTP tower. This pair of systems establishes an RTP connection which carries out APPN sessions (multiple APPN sessions can be multiplexed over a single RTP connection). In order to establish an RTP connection between a pair of HPR RTP tower systems, the following must be true:

- The set of nodes must support the HPR intermediate routing function.
- The transmission groups (TGs) that exist between the two HPR RTP tower systems must support the HPR intermediate routing function.

This routing is known as Automatic Network Routing (ANR).

When an RTP node sends packets of data, it must keep those buffers until the RTP node receives acknowledgement that its RTP partner has successfully received the data. Maintaining detailed knowledge of data sent and received is necessary in order to provide the additional value provided by HPR, the non-disruptive path switching function. HPR does not rely on the data link layer to provide data retransmission functions. HPR supports a function that is called selective retransmission. With selective retransmission, only the data which has not been acknowledged gets retransmitted. For example, if an RTP node sends eight packets and all but the fourth packet is successfully acknowledged, then only the fourth would retransmit. This differs from other retransmission algorithms in which the first unsuccessful packet and all the subsequent packets would transmit.

Nodes performing intermediate routing of HPR traffic or ANR have no session awareness. HPR uses source routing. The nodes performing ANR examine packets as they receive them and determine the next hop of the route. The next hop is based on something that is called the ANR label. All HPR packets contain an ANR label. Any ANR that a network node is performing does **not** count as an APPN intermediate session. The maximum intermediate sessions parameter that is configured by means of the Change Network Attribute (CHGNETA) command has no effect on the ANR capacity of a system. Controlling the amount of ANR that different systems will perform in a network is completely dependent on the route selection phase of APPN session establishment.

When sessions are carried over RTP connections, any segmentation or reassembly is performed within the IBM i central processing unit (CPU).

HPR uses a function called Adaptive Rate Based (ARB) congestion control. ARB regulates the flow of traffic by predicting congestion in the network and reducing the sending rate of a node into the network. ARB then attempts to prevent congestion rather than reacting to it after it has occurred. If all of the traffic occurring over a network was HPR, then ARB would be a fair way of sharing the bandwidth of a network. ARB also allows high utilization of the networking resources. When HPR traffic mixes with straight APPN or TCP/IP traffic, the HPR throughput may suffer because the other protocols do not practice similar congestion control techniques.

Related concepts

[Performance considerations for APPN and HPR](#)

Here are the factors that can affect the performance of the APPN and HPR protocols.

[Configuring APPC, APPN, and HPR](#)

You can have APPC, APPN, and HPR configured automatically or manually on your system.

Communications optimization using APPN virtual controllers

An *APPN virtual controller* is a controller description that Advanced Peer-to-Peer Networking (APPN) can use and that high-performance routing (HPR) support uses.

An *APPN virtual controller* can be used to attach and manage advanced program-to-program communications (APPC) device descriptions. This type of controller does not represent a connection to a remote system.

On IBM i, local applications that need to establish LU 6.2 sessions to other locations in the APPN network require an APPC device description that specifies APPN(*YES). For simplicity, these devices are referred to as APPN devices.

The Allow APPN virtual support (ALWVRTAPPN) parameter is on the Change Network Attributes (CHGNETA) command. If the ALWVRTAPPN parameter is *YES, any existing APPN devices that are attached to the real APPN controller description will not be allowed to vary on. Message CPDB157 will be issued. If migrating to this new APPN object model, you may want to delete any existing APPN devices because they will no longer be used. You may also want to delete the devices if you have no intention of resetting the ALWVRTAPPN parameter to *NO.

APPN virtual controllers provide the following benefits:

- Virtual controllers can reduce the number of device descriptions

Prior to supporting APPN virtual controllers, multiple APPN device descriptions can be created and used simultaneously to communicate between the same local location and remote location pair. This situation was possible because alternate paths exist through the network. The first hop out of the local system (that is represented by a controller description) can be different for the two paths. After a session is established, the same APPN device description is used for the life of that session. With APPN virtual controller support, you can accomplish all communications between the same local location and remote location pair using a single device description. This single device description can be used, even if multiple paths to that remote location exist in the network.

- Virtual controllers bypass 254 device limit

IBM i allows a maximum of 254 devices to attach to a controller description. In some environments, there may be a requirement to access more than 254 different locations (that are each represented by devices) through a single system. For example, an IBM i may attach to an IBM Z, connected to hundreds of systems that the local IBM i would like to communicate with (through IBM Z). Without APPN virtual controller support, this communication requires the definition of parallel transmission groups (multiple controller descriptions) between the local system and IBM Z. Using multiple real controller descriptions can be costly in regards to both the line costs and managing multiple connections. With APPN virtual controller support, you use one real controller description, but attach more than 254 devices that are spread across more than one virtual controller.

- Error recovery minimized

APPN virtual controller descriptions do not associate with any communication lines or adjacent system. Thus, there are no *communication failures* to associate with these controller descriptions. This situation highlights some key points regarding error recovery:

When APPN virtual controller descriptions are not used, device descriptions attach to APPN controller descriptions that represent connections to adjacent systems. When communication failures occur, applications that are affected by the session outages must be notified. The system also performs error recovery on the controller description and all of the attached device descriptions. In some large environments, device error recovery can take a lot of time.

When APPN virtual controller descriptions are used, the APPN controller descriptions that represent connections to adjacent systems do not have device descriptions attached. When communication fails (for example, a line failure), the applications affected by the session outages are notified. The system recovers errors on the controller description. No error recovery is required on the device descriptions if each of the following are true:

- The device descriptions are attached to the APPN virtual controller description.
- The APPN virtual controller descriptions are not marked as inoperative.

Eliminating error recovery at the device level can help decrease the amount of time IBM i requires to recover errors after some communication failures.

Related concepts

[Considerations in designing an APPN and HPR network](#)

When you design your network, consider these factors to optimize performance.

Configuration parameters for fine-tuning APPC performance

The setting of certain parameters affects the communication performance of the IBM i. To fine-tune your advanced program-to-program communications (APPC) performance, you can change the values for the following parameters.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Maximum length request/response unit size (MAXLENRU) parameter

The maximum length of a Systems Network Architecture (SNA) request/response unit (RU) can be specified with the MAXLENRU parameter in a mode description for APPC, APPN, and HPR. In most situations, the use of the *CALC value for the MAXLENRU parameter gives you the optimal RU size.

If you select a value of *CALC for the MAXLENRU parameter, the system selects an efficient size that is compatible with the frame size that you choose. (The frame size is on the line description command.) Changing the RU size to a value other than *CALC might negate this performance feature.

If *CALC is not used, you should most often choose an RU size that is slightly less than the maximum frame size or a multiple of the maximum frame size. This setting ensures that the largest possible frame size transmits all the time.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Maximum frame size (MAXFRAME) parameter

The maximum frame size is specified in the MAXFRAME parameter in the line and controller descriptions. Larger frame sizes generally supply better performance. Large frame sizes might not work well for error-prone lines or networks because longer times are required to transmit large frames when errors occur.

For each line type, set these maximum frame sizes in the line description.

To take advantage of these large frame sizes, these values must be configured correctly. The MAXFRAME parameter on the line description and controller description must reflect the maximum value.

Having configured a large frame size does not negatively affect performance for small transfers. Note that both the server and the other link station must be configured for large frames. Otherwise, the smaller of the two maximum frame size values are used in transferring data. Bridges may also limit the frame size.

Note: In order to run HPR, MAXFRAME must be set to at least 768.

The Communications Configuration manual might be a useful reference to you. It is available from the [IBM Publications Center](#) in an online format that you can download.

Pacing (INPACING, OUTPACING, MAXINPACING) parameters

Pacing is required if there is a possibility of overflowing data buffers internal to the controller or to the host system. This typically occurs if the controller or host must pass the data to a device that operates at

a slow speed. If the host system receives a pacing response, it sends more data frames, up to the window size, to the controller.

- Pacing determines how many message units (SNA RUs) can be transferred over a session before receiving an acknowledgement from the receiving system. An excessive number of pacing responses may adversely affect network performance. However, the absence of pacing can cause network congestion and unfair utilization of IBM i resources (buffers and central processing units). The values that you can use in negotiating the pacing values with the adjacent system are determined from the INPACING and OUTPACING values on the mode description. The IBM i will not allow these values to be negotiated to a higher value. If necessary, the receive pacing value will negotiate to a lower value, matching the INPACING value.
- The pacing value is determined at session establishment time and is not changed for the duration of the session for the following reasons:
 - The adjacent system does not support adaptive pacing
 - The transmission priority is low priority
- If the adjacent system does support adaptive pacing, the minimum pacing value is set at session establishment time using the INPACING and OUTPACING values. The location that starts the session establishment (BIND request) is responsible for setting the values. No negotiation of the values is performed. However, support is provided by the system to change or adapt the pacing values based on the system's buffer resources and traffic patterns in the network. The system can now allocate its session buffers automatically to efficiently use its available resources. The MAXINPACING parameter defines the upper limit on the number of session buffers. The default value of *CALC sets an upper limit of 2 for the INPACING value.
- The IBM i system also has the ability to slow down the transfer of data or even stop receiving at any node of any session. This allows for more equity in the network by dynamically turning the flow of messages on to any hop for any session that may be contributing to congestion problems. In general, the value of the INPACING, OUTPACING, and MAXINPACING parameters on the mode description can affect the data rate, network congestion, buffer utilization, and central processing unit (CPU) utilization.

Related concepts

Performance considerations for APPN and HPR

Here are the factors that can affect the performance of the APPN and HPR protocols.

Transmission priority (TMSPTY) parameter

The transmission priority (TMSPTY) parameter is on the class-of-service (COS) description. When you create a class-of-service description, you can specify that the transmission priority for the class of service is high, medium, or low using the transmission priority (TMSPTY) parameter.

The session activation request carries the transmission priority you specify when a session is established. This allows each logical unit on the session and each routing entry along the session path to store the same transmission priority. You can ensure better response time for the applications that require it by assigning an appropriate mode (which includes a class of service) at session activation time.

Note: Generally, interactive traffic should have a high priority and batch traffic a low priority.

Wait time (QACRETRY and QACINTERVL) data areas

To prevent the data integrity problem, advanced program-to-program communications (APPC) waits for an acknowledgement from the remote system after sending the DETACH signal to end the transaction. You can configure the wait time using the QACRETRY and QACINTERVL data areas in the QGPL library.

You can use the QACRETRY data area to specify the number of retries to receive the acknowledgement, and use the QACINTERVL data area to specify the interval for each retry.

You can create these two data areas and set the values by using the Create Data Area (CRTDTAARA) command.

- Set the type of values that are contained in the two data areas to *DEC using the TYPE parameter.

- Set the length for these two data areas to 2 using the LEN parameter.
- Set the initial value for these two data areas using the VALUE parameter, with 99 as the maximum value.

For example, the following command creates the QACRETRY data area and sets the number of retries to 99:

```
CRTDTAARA DTAARA(QGPL/QACRETRY) TYPE(*DEC) LEN(2) VALUE(99)
```

Notes:

- If you do not create either or both of these two data areas, the default values are used: 8 for the number of retries and 15 seconds for the interval for each retry.
- For either of the two data areas, if you create the data area but do not specify the initial value for it, the data area is initialized to a value of 0.

Set appropriate values for the two data areas based on the specific situation:

- For network connections with slow pacing responses from the remote application and a slow-speed switched line, set a relatively long wait time. By doing so, you can ensure that the data and the DETACH signal can flow to the remote system within the wait time.
- For those remote applications that do not send acknowledgements when they are finished, set an appropriate wait time. By doing so, you can prevent the local system from waiting longer than necessary before sending the UNBIND signal to complete the flow of the conversation between the two systems.

Related reference

[Create Data Area \(CRTDTAARA\) command](#)

APPC, APPN, and HPR security

For IBM i systems that communicate with each other using APPC, APPN, and HPR, consider the following security aspects.

- General security considerations:

Consider the following measures when securing your network:

Note: The following password considerations only apply if password protection is not active.

1. When application program security is used, specify SECURELOC(*VFYENCPWD). This means that you only get to log on if BOTH your user profile name AND password are the same on both systems.
 2. The person responsible for network security ensures that each user has a unique user ID throughout the network.
 3. Have your system administrator set a limit on the number of consecutive password attempts that are not valid for a given display device. When this limit is reached, the device is then varied off. Set the limit with the system value QMAXSIGN. This is only true for Display devices, not for APPC devices.
 4. Users can sign on to more than one IBM i system with the same profile. To limit the user profile to one sign-on, Set the system value (*SYSVAL) for LMTDEVSSN parameter on either the Create User Profile (CRTUSRPRF) or Change User Profile (CHGUSRPRF) command.
- Physical security considerations:

You are responsible for the physical security of your system when you specify *NONE for the location password (LOCPWD) parameter during APPC configuration. In this case, the IBM i system does not validate the identity of a remote system when a session is being established. However, you can still use application-level security if the remote system supports it. For example, if the remote system is an IBM i system with security level 20 or above. Security needs to be consistent across all the systems in a network if intersystem access is to be controlled and yet not unnecessarily restricted.

Related concepts

[Planning and setting up system security](#)

Session-level security for APPN and HPR

Session-level security is achieved by specifying a password on the LOCPWD parameter during configuration. The IBM i system uses the password to validate the identity of the remote system during session establishment. If the password does not match the password specified on the remote system, the connection is not allowed.

If the remote system does not support session level security (Series/1 RPS version 7.1, CICS/VS release 1.6), specify LOCPWD(*NONE) to establish the connection and provide the necessary physical security.

A security concern exists when you create device descriptions with APPN(*YES) and APPN automatically creates and varies on a device description with the same remote network ID, location name, and local location name as the APPN remote location configuration list entry. To compensate for remote locations using an independent device description with APPN(*YES), add an entry to the APPN remote location configuration list that includes security information.

Note: In order to avoid connection problems, ensure that all the device descriptions, as described previously, contain exactly the same security information.

Protecting your system in an APPN and HPR environment

APPN reduces the physical, configuration barriers to communications. However, you might want to build some logical barriers between systems in the network for security reasons. This ability to control which systems can connect to yours is often called *firewall support*.

APPN networks provide open connectivity and require minimal configurations by each system in the network. When a system has a connection into an APPN network, it can establish sessions with other systems that are connected within that APPN network.

With firewall support, network administrators might use a variety of node types to specify which connections between APPC locations are allowed. For example, you might want to allow SYSTEMB to communicate with SYSTEMA and SYSTEMD, but not with SYSTEMC.

APPN filtering support

APPN filtering support provides the ability to create a firewall that is based on APPC location names. *Session endpoint filter* and *directory search filter* are the two types of filter lists available.

Before we discuss APPN filtering support, an explanation of node types in an APPN network is needed:

- A *peripheral node* is at the edge of a network. It can participate in the network, but it cannot provide intermediate routing to other systems in the network. A peripheral node can be an *end node (EN)* such as MADISON and PARIS in the figure below. A peripheral node can be a *low-entry networking node (LEN)*, such as CHICPC1 and CHICPC2. A peripheral node can also be a network node in a different network (NETID). From CHICAGO's perspective, LONDON is a peripheral node.
- A *network node (NN)* provides routing services among systems in the network. In CHICAGO, and ATLANTA are examples of network nodes.
- A *Branch Extender* node is an extension to the APPN network architecture that appears as a network node (NN) to the local area network (LAN), and as an end node (EN) to the wide area network (WAN). This reduces topology flows about resources in the LAN from being disconnected from the WAN.

The two types of APPN filtering support are explained as follows:

- A *session endpoint filter* controls access to and from a location. For example, in the session endpoint filter on the CHICAGO system in the figure below, it specifies which locations can establish a session with CHICAGO or with PAYROLL. CHICAGO and PAYROLL are two different locations on the CHICAGO system.

Similarly, the session endpoint filter on the MADISON system specifies which locations can establish a session with the MADISON location.

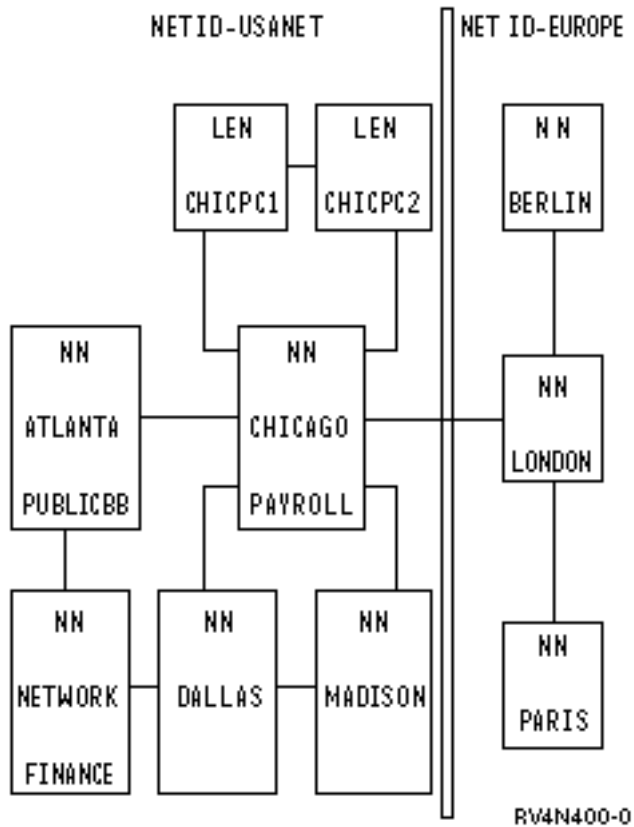


Figure 4. Two connected APPN networks

To create a session endpoint filter, use the QAPPNSSN configuration list by itself, or in conjunction with the QAPPNRMT configuration list.

- A *directory search filter* on a network node determines the following for its associated peripheral nodes:
 - Access *from* the peripheral node (when the peripheral node is the requester). For example, in you can use the directory search filter on LONDON to control the possible destinations for users on the PARIS system. Similarly, you can use the directory search filter on CHICAGO to control the possible destinations for users on CHICPC1 and CHICPC2.
 - Access *to* the peripheral node (when the peripheral node is the destination). In for example, you can use the directory search filter on CHICAGO to determine which locations can access CHICPC1. Because both CHICAGO and DALLAS provide connections to MADISON, you must set up the directory search filters on both CHICAGO and DALLAS to restrict connections to MADISON.

Similarly, you can use the directory search filter on CHICAGO to specify which USANET locations are permissible destinations for EURONET users.

To create a directory search filter, use the QAPPNDIR configuration list.

Creating a session endpoint filter

You can create a session endpoint filter by using the QAPPNSSN and QAPPNRMT configuration lists together, or using the QAPPNSSN configuration list by itself. Use the QAPPNSSN and QAPPNRMT configuration lists together for a more secure method.

The following example describes the two methods that can be used to create a session endpoint filter on the CHICAGO system, as is shown in the following figure. The session endpoint filter on the CHICAGO system must satisfy the following requirements:

- Only the FINANCE location can establish a session with the PAYROLL location.
- The CHICAGO location can communicate with any USANET location except PAYROLL.
- The CHICAGO location can communicate with LONDON.

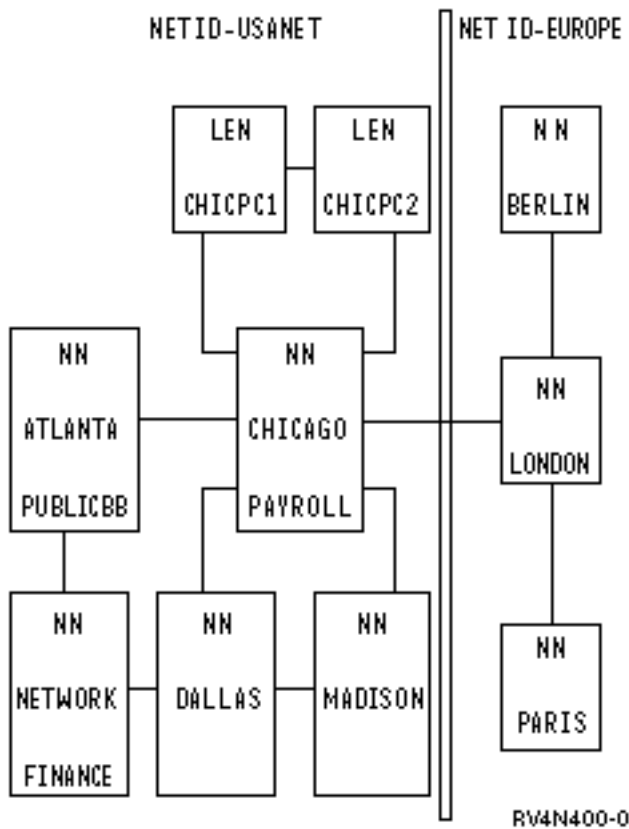


Figure 5. Two connected APPN networks

- **Using the QAPPSSN and QAPPNRM configuration lists together:**

The more secure method for creating a session endpoint filter is to use the QAPPSSN configuration list and the QAPPNRM configuration list together. The QAPPNRM configuration list provides password security between systems, which helps to protect from an imposter system (a system or user that is pretending to be another system).

When you use this method, you create the QAPPSSN configuration list that does not specify any remote locations. It points to the QAPPNRM configuration list.

The drawback to this method is that you must explicitly define each location pair on the QAPPNRM configuration list. If you want the CHICAGO location (which is on the same system as the PAYROLL location) to communicate with other locations, you need to add an entry for each pair.

- **Using the QAPPSSN configuration list by itself:**

When you specify remote locations in the QAPPSSN configuration list, your configuration task is simpler because you can use generic names and wildcard entries. However, when you use this method, you do not have the protection of password verification between locations. In addition, when you use generic names and wildcards, the system might accept or reject requests in a different way than you intended.

Class of service routing

Class-of-service (COS) descriptions are used to calculate the route that a session takes. COS descriptions also specify the transmission priority that governs the rate of data transfer after a session is established.

The following COS descriptions are shipped with the IBM i system:

- #CONNECT: the default class of service
- #BATCH: a class of service that is tailored for batch communications

- #BATCHSC: is the same as #BATCH except that a data link security level of at least *PKTSWTNWK is required
- #INTER: a class of service that is tailored for interactive communications
- #INTERSC: is the same as #INTER except that a data link security level of at least *PKTSWTNWK is required

If you want to force a particular route to be selected, create a user COS description.

Related tasks

Creating class-of-service descriptions

Class-of-service descriptions are used only by APPN and HPR. A class-of-service description tells the system which network nodes and transmission groups are acceptable and, of those acceptable, which are preferred during route selection.

Troubleshooting APPN and HPR

If you have a communication problem related to APPN or HPR, review the following topics to help troubleshooting.

Solving remote communication problems using STRPASTHR

When you encounter communication problems indicating that the route to the remote location cannot be found, you can attempt to make the connection again using the Start Pass-Through (STRPASTHR) command.

The STRPASTHR command has built in diagnostic capabilities that exceed those provided by other interfaces that utilize APPN networks. These diagnostic capabilities include problem analysis, problem logging, and error logging functions.

When the STRPASTHR command fails to contact a remote location in an APPN network, a record is written to the problem log. The logging action only occurs if a problem analysis is associated with the STRPASTHR command for analyzing the data. You can use the Work with Problems (WRKPRB) and the Analyze Problem (ANZPRB) commands to examine and interpret the problem log to isolate the problem.

When the STRPASTHR command fails to contact a remote location in an APPN network, an error log is also issued. You can also use the error log to assist in troubleshooting your communication problem.

Related concepts

APPN error log data

APPN error log data defines the setup data of an APPN session that is supplied when an error log is issued for a start pass-through failure. A start pass-through failure results in message CPF8933 (Route to specified location not found) being issued to the user's workstation. Use the following information for error log entries with reference codes 7100 and 7101.

Solving communication problems using DSPAPPNINF

Isolating a routing problem in an Advanced Peer-to-Peer Networking (APPN) network can be challenging. To view the APPN information to assist you in understanding the topology of network nodes and some of their locations, use the Display APPN Information (DSPAPPNINF) command.

To display APPN Information, type the command DSPAPPNINF on a command line and press F4. You can also select option 6 (Display APPN information) on the Network Management menu.

The information that the system displays, prints, or stores depends on the options you select. The system displays additional options that are based on the previous options you selected.

See the following scenarios for help when using the DSPAPPNINF command:

- Type DSPAPPNINF *TOPOLOGY on System A.
 1. Type 5 next to System A to display the show Link Destination Nodes display.

The Display Link Destination Nodes display identifies how this node's topology database looks. The Link Active column identifies whether APPN will consider the link in route computation. If a Link Active column value is No, this indicates the link will not be included in APPN route selection.

2. Type 5 to display link characteristics. This information along with the information from the Display Network Attributes (DSPNETA) command identifies the transmission group (TG) values and node values.

With this information, you can determine why a path is taken or why one is not being taken for the class of service (COS).

- Type DSPAPPNINF *LCLNODE on System A.

This allows you to determine what locations are known by the local node. It shows locations that are configured on the local node and locations that are found by previous searches.

- Type DSPAPPNINF *SSN on System A.

This allows you to look at up to 200 endpoint sessions that were successfully established since the last IPL. You can also see the route that was taken by that session, error data, session start BIND, end time, pacing that is used, and others.

- Type DSPAPPNINF *SSN SSNTYPE(*INMSSN) on System A.

This allows you to determine whether active sessions are routed through the local system. You may, for example, want to vary off a controller, but need to know whether it is being used for intermediate sessions. You can also see which controller descriptions are associated with intermediate sessions.

Solving communication problems using WRKAPPNSTS

The Work with APPN Status (WRKAPPNSTS) command provides session-related information for advanced program-to-program communications (APPC) controller descriptions running Advanced Peer-to-Peer Networking (APPN) or high-performance routing (HPR).

You can use the WRKAPPNSTS command to get the following information about APPN controller descriptions:

- All the location pairs that have one or more sessions over the controller description. The session activity is not limited to those sessions in which the local system is the source or target of the session. Information about APPN intermediate sessions and cases when the local system is performing the APPN, or HPR boundary function, are also provided.

Note: Automatic Network Routing (ANR) traffic is not shown.

- Session information for the location pairs that are associated with a controller. The session information provides a tie between a particular session and the device description the system uses. For example, a device description attached to a real controller description or to an APPN virtual controller description is shown.
- Rapid Transport Protocol (RTP) connections that either originate or end on the local system. It is also possible to see the location pairs and session information associated with the sessions that are carried over the RTP connection.
- The route that a particular RTP connection has taken through an HPR subnet.

You can also use the WRKAPPNSTS command to request the system to perform some operations against RTP connections. These operations include performing a nondisruptive path switch and ending an RTP connection that is currently active. You can issue these operations against the following RTP connections:

- A single RTP connection
- All of the RTP connections that have the first hop of their route across the controller description being displayed

Related concepts

[Solving communication problems using session activity](#)

Session activity, or actual work that occurs between the local system and adjacent systems provides you with the information about network attributes, mode, class of service (COS), and topology information.

Solving communications problems using communications trace

Communications trace can capture HPR data running over IP networks for analysis. You can use communications trace to troubleshoot Enterprise Extender communications problems.

Related reference

[Communications trace](#)

Solving communication problems using session activity

Session activity, or actual work that occurs between the local system and adjacent systems provides you with the information about network attributes, mode, class of service (COS), and topology information.

You might want to view the session activity for any of the following reasons:

- Activity occurs over the controller descriptions to adjacent systems.
- Certain sessions are established over a connection that the operator did not expect.
- The optimal route is no longer operating.

To find another route for the session, you might need to know what location pairs are used for the particular connection. If you want to change the route for any session, you might need to vary off the controller description. Before varying off controller descriptions, determine whether any active sessions are using that connection (so you can notify the affected users of the upcoming outage). Delay varying off the controller description with active sessions.

Related concepts

[Solving communication problems using WRKAPPNSTS](#)

The Work with APPN Status (WRKAPPNSTS) command provides session-related information for advanced program-to-program communications (APPC) controller descriptions running Advanced Peer-to-Peer Networking (APPN) or high-performance routing (HPR).

Systems Network Architecture sense codes

Systems Network Architecture (SNA) sense codes contain additional information for system programmers and system support personnel about the error or problem that has occurred on the network.

Related reference

[SNA Formats](#)

APPN error log data

APPN error log data defines the setup data of an APPN session that is supplied when an error log is issued for a start pass-through failure. A start pass-through failure results in message CPF8933 (Route to specified location not found) being issued to the user's workstation. Use the following information for error log entries with reference codes 7100 and 7101.

Note: Use the Work with Problems (WRKPRB) command for error log entries with the reference code 7102.

Related concepts

[Solving remote communication problems using STRPASTHR](#)

When you encounter communication problems indicating that the route to the remote location cannot be found, you can attempt to make the connection again using the Start Pass-Through (STRPASTHR) command.

Standard APPN diagnostic data

This table defines the format of APPN error log entries. The information available in the error log depends on how far the session initiation attempt has proceeded when the failure or timeout occurs.

Table 3. APPN error log data

Byte	Bit	Content
Session setup control information		
0-3		Length of entire APPN error log structure
4-15		Reserved
16-17		Reserved
18-19		Time out session setup state (available if session failed due to time-out)
1A-21		Reserved
22		Flag bits
	0	Local system node type (0 = end node and 1 = network node)
	1	Session setup request should no longer be tracked
	2	A final session state has been reached
	3-7	Reserved
Pre-search phase data		
23		Pre-search phase data measurements
	0	Pre-search phase data is valid to look at because some of the fields have been filled in
	1-7	Reserved
24-2B		Local location name
2C-33		Remote location name
34-3B		Remote network identifier
3C-43		Mode name
44-4D		Device description name
4E-57		Controller description name
58-71		PCID (procedure correlation identifier)
72-79		Class-of-service name
Common information during search phase		
7A		Common information during search phase
	0	Common information data is valid to look at because some of the fields have been filled in
	1	Wildcard entry was used to satisfy search
	2-7	Reserved

Table 3. APPN error log data (continued)

Byte	Bit	Content
7B-82		Network identifier for the destination node
83-8A		Control-point name for the destination node
8B-92		Network identifier for the network node server of the destination node
93-9A		Control-point name for the network node server of the destination node
9B-9E		Reserved
9F-A6		The network identifier of the remote location that was found using the *ANY directory entry
A7-AE		The control-point name of the remote location that was found using the *ANY directory entry
AF-B6		The network identifier of the network node server of the remote location that was found using the *ANY directory entry
B7-BE		The control-point name of the network node server of the remote location that was found using the *ANY directory entry
Directory search summary information - end node		
BF		Directory search summary information - end node
	0	End node search information is valid to look at because some of the fields have been filled in
	1	Search type (0 = local only search and 1 = distributed search)
	2	Real indicator supplied by the network node server
	3	Default indicator supplied by the network node server - note that the real and default server-supplied indicators are mutually exclusive
	4-7	Reserved
C0-C7		Network identifier of the network node server for the local system
C8-CF		Control-point name of the network node server for the local system
Directory search summary information - network node		
D0		Network node directory steps processed indicators
	0	Network node search information is valid to look at because some of the fields have been filled in
	1	Query topology database for a network node control-point name
	2	Location found in local directory database
	3	One hop search sent to an attached end node
	4	Route selection attempted for a directed search to a network node
	5	Directed search sent to network node
	6	Reserved
	7	Reserved
D1	0	Domain broadcast sent
	1	Broadcast search sent

Table 3. APPN error log data (continued)

Byte	Bit	Content
	2	Reserved
	3	Reserved
	4-7	Reserved
D2-D9		Directed search target network identifier
DA-E1		Directed search target control-point name
E2-E9		Reserved
EA-F1		Reserved
F2-F9		Reserved
FA-101		Reserved
Switched link activation		
102	0	Link activation data is valid to look at because some of the fields have been filled in
	1-7	Reserved
103-10A		First hop of the route network identifier (real node)
10B-112		First hop of the route control-point name (real node)
113-11A		First hop of the route network identifier (virtual node)
11B-122		First hop of the route control-point name (virtual node)
123		Transmission group number for first hop of route
124-12D		Line description name
12E-131		Reserved
132-133		Reason code for error
Generic session setup information		
134-137		Sense code returned
138-15D		Past session setup states
15E-15F		Current session setup state
160-17F		Reserved
180		Variable data area (see “Optional APPN diagnostic data” on page 47)

Note: 0=false and 1=true on the bit fields, unless otherwise specified.

APPN session setup states

The following table presents the possible session setup states for APPN when it processes a session initiation request. One of these values always resides in the current session setup state.

Table 4. APPN session setup states

State	Reason
1000	Session setup complete. An existing session will be used; therefore, APPN control point functions will not be called.
1015	Session setup request has failed. Refer to sense code for details.
1020	Session setup rejected. The local location name chosen is not defined in either the network attributes or as a local location list entry.
1025	Session setup rejected. Mode name specified is not defined on the system.
1030	Session setup request has been sent by location manager to resource manager to obtain a device.
1032	Session setup request cannot be satisfied with a non-APPN device or with an existing APPN session. The APPN control point has been called to establish a new session.
1035	Session setup has been pended because of a previous request that is waiting for the request transmission group vectors processing to complete.
1040	Session setup has been pended because of a previous request that is still waiting for the route selection phase (request single hop route - end node) to complete.
1050	Session setup has been pended because of a previous request that is still waiting for the route selection phase (request route - network node) to complete.
1060	Session setup has been pended because of a previous request that is still waiting for the switched link activation phase to complete.
1070	Session setup has been pended because of a previous request that is still waiting for the location search phase to complete.
1080	A request transmission group vectors request is outstanding to topology routing services component.
1082	A request transmission group vectors request is being processed by topology routing services component.
1084	The request transmission group vectors response was returned by the topology routing services component.
1086	The request transmission group vectors request has been received by session services.
1090	Location search phase request is outstanding, but has not yet been received by the local system's directory services function.
End node location search phase (2000 - 2999) states	
2000	The local system's directory services has received the search request and has begun its processing.
2010	There is a one hop search request outstanding to the local system's network node server.
2020	Location search processing has been completed by the local system's directory services.
2025	Session services has received the locate message response from directory services.
2030	Location search phase has failed. The owning control point for the remote location can not be determined during the search phase. In this case, the search was forwarded to our network node server and the location can not be found.

Table 4. APPN session setup states (continued)

State	Reason
2040	Location search phase has failed. The owning control point for the remote location can not be determined during the search phase. In this case, the search was not sent out of the local system because of no network node server, and there was no network node that the local system can forward the bind to.
2050	Location search phase has failed. The network node server has sent an SNA negative response indicating that the route selection control vector (RSCV) required is larger than 255 bytes.
2060	Location search phase has failed. The network node server has sent an SNA negative response indicating that the class-of-service is not valid.
2070	Location search phase has failed. The network node server has sent an SNA negative response indicating a route-not-available condition.
Network node location search phase (3000 - 3999) states	
3000	The local system's directory services has received the search request and has begun its processing.
3010	Query control-point name outstanding. A request to determine if the remote location is the name of a network node control point in the topology database is outstanding.
3012	Query control-point name request is being processed by topology routing services.
3014	Query control-point name response has been sent by topology routing services.
3016	Query control-point name response has been received by directory services.
3020	One hop search request is outstanding to an attached end node.
3030	A request for a route is outstanding to topology routing services so that a directed search can be sent to another network node.
3032	Request route for directed search is being processed by topology routing services.
3034	Request route response for directed search has been sent by topology routing services.
3036	Request route response for directed search received by directory services.
3040	A directed search request is outstanding to another network node.
3050	A request for a route is outstanding to topology routing services for a remote search.
3052	Request route for a remote search is being processed by topology routing services.
3054	Request route response for a remote search has been sent by topology routing services.
3056	Request route response for a remote search was received by directory services.
3060	A routed search request is outstanding to a network node.
3070	A domain broadcast is currently being run. This involves querying attached end nodes or network nodes in another network to determine if the location is known by that system.
3080	A broadcast search is outstanding to one or more directly attached network nodes (this might involve attached network nodes that have access to multiple networks).
3090	A request for a route is outstanding to topology routing services to determine if a node that can access multiple networks exists so that it can determine where the remote location exists.
3092	Request route for a node that can access multiple networks is being processed by topology routing services.

Table 4. APPN session setup states (continued)

State	Reason
3094	Request route response for a node that can access multiple networks has been sent by topology routing services.
3096	Request route response for a node that can access multiple networks was received by directory services.
3100	A search request is outstanding to a node that can access multiple networks.
3110	A request is outstanding to the session services component in order to perform functions necessary to send searches into a different APPN network.
3120	The location search phase has completed and a response has been returned by directory services.
3125	The location search phase has completed and the response has been received by session services.
3130	The location search phase has failed.
Route selection phase (4000 - 4999) states	
4000	A request for a single hop route is outstanding to the topology routing services component.
4002	The request for a single hop route is being processed by topology routing services.
4004	The request single hop route response has been returned by topology routing services.
4006	The request single hop route response has been received by session services.
4010	A request single hop route failure has occurred.
4030	A request for a route is outstanding to the topology routing services component.
4032	The request for a route is being processed by topology routing services.
4034	The request route response has been returned by topology routing services.
4036	The request route response has been received by session services.
4040	The request for a route has failed. The class-of-service name being used is not defined on the local system.
4050	The request for a route has failed. The route selection control vector required to satisfy the end-to-end route is larger than the architected limit (255 bytes).
4060	The request for a route has failed. Route-not-available condition has been detected. No destination network nodes or virtual nodes are available for intermediate routing.
4062	The request for a route has failed. Route-not-available condition has been detected. A route that satisfies the user class-of-service but which uses inactive transmission groups does exist.
4064	The request for a route has failed. Route-not-available condition has been detected. A route that has active transmission groups but which does not meet the class-of-service requirements does exist.
4066	The request for a route has failed. Route-not-available condition has been detected. A route that has active transmission groups but which does not meet the class-of-service requirements does exist. A route that satisfies the user class-of-service but which uses inactive transmission groups also exists.
4068	The request for a route has failed. Route-not-available condition has been detected. Destination intermediate routing nodes exist, but no route of any type can be calculated.
4080	Session setup failure. The controller description that represents the first hop of the route is unknown by the local system.

Table 4. APPN session setup states (continued)

State	Reason
Switched link activation phase (5000 - 5199) states	
5000	The request to activate the switched link is currently outstanding from session services.
5005	Configuration services has begun processing the activate route request but has not yet completed its processing.
5010	The activate route has completed, but some failure has occurred. Details are based on sense code.
5020	The request to activate the switched link has been pended. A controller description is being created or varied on for establishing a link using the connection network.
5030	The request to activate the switched link has been pended. The controller is not allowed to make a connection in this state. The probable cause is a message outstanding for this controller description.
5040	The request to activate the switched link has been pended. Configuration services is waiting for the operating system to issue the command to activate the switched connection.
5050	The request to activate the switched link has been pended. The attempt to select an eligible line description for this request has failed. The probable cause is a message that is outstanding requiring operator intervention.
5070	The request to activate the switched link has been pended. The system is currently in the process of establishing an outgoing connection.
5080	The request to activate the switched link has been pended. The outgoing connection has been made, but the exchange identification phase is in progress.
5090	The request to activate the switched link has been pended. The outgoing connection or exchange identification phase has failed. The system is waiting for the operator to respond to a message.
5100	The switched link activation has completed successfully.
5110	The session services component has received its response to its switched link activation request.
Non-switched link activation phase (5200 - 5299) states	
5200	Session services is waiting for configuration services to complete the non-switched link activation.
5210	The non-switched link activation phase has completed successfully.
HPR route setup phase (5300 - 5399) states	
5300	A request to determine whether a session should be carried over an RTP connection is outstanding.
5310	The request to determine if an RTP connection should be used for the session has detected an error.
5315	An HPR Route Setup request is outstanding.
5320	The HPR Route Setup request was returned with a good completion.
5325	The HPR Route Setup request has failed.
5330	The HPR Route Setup phase has completed successfully.
APPN virtual controller selection phase (5400 - 5499) states	
5400	The request is outstanding to the virtual controller manager component to find the APPN virtual controller description.
5490	The request to find the APPN virtual controller description has failed.

Table 4. APPN session setup states (continued)

State	Reason
5495	The request to find the APPN virtual controller description has completed successfully.
Device selection phase (6000 - 6999)	
6000	A request is outstanding to the T2 station input/output manager (IOM) task to select a device.
6005	The T2 station input/output manager (IOM) task has begun processing the get device request.
6010	Device selection pended. The device was found, but it is in the process of being automatically varied on.
6020	Device selection pended. No device was found; therefore, a new device is in the process of being created and varied on.
6025	Device selection request pended. A dynamic device creation or vary on is already in progress for a previous get device request or for a received bind request.
6030	The device selection has failed. Refer to the sense data returned for an explanation of this failure.
6040	The device selection phase was completed successfully by the T2 station input/output manager (IOM) task.
6045	The device selection response was received by session manager.
6050	APPN session manager processing is complete.
6060	The session setup has completed successfully.

Optional APPN diagnostic data

The optional APPN diagnostic data is presented in a format similar to the format of a control vector. This data is located after the standard APPN diagnostic data. More than one type of variable data can be presented.

The type of optional data that is contained in the error log depends on the current session setup state at the time the error or time-out occurred. This data begins at offset X'0312' from the start of the error log entry.

Header information exists at the beginning of each variable data element. Header information provides the length and key value of the data area element (similar to the way that control vectors are structured).

Search-sent elements

This table defines the structure of a search-sent information element. Multiple elements might be supplied. The header information length is used to determine the length of a single element.

At times, only specific search types and search results are supplied. These are performed for the session setup state domain broadcast (3070) and broadcast search outstanding (3080). At other times, all the searches sent and their results are supplied in the search failure (3130) session setup state.

Table 5. Search-sent information elements

Byte	Hex value	Content
Header information for variable data		
0		Length of this type of variable data
2	X'01'	Key value for a search-sent element
Variable data		
3		Network identifier of the system searched

Table 5. Search-sent information elements (continued)

Byte	Hex value	Content
0B		Control-point name of the system searched
13		Search type
	X'00'	No search sent
	X'01'	Search type is single hop
	X'02'	Search type is directed to a network node control point
	X'03'	Domain broadcast
	X'04'	Network broadcast
	X'05'	Directed for remote search
	X'06'	Directed to a node that can access multiple networks
14		Node type
	X'01'	End node
	X'02'	Network node
	X'03'	Control point resides in a network with a different network identifier
15		Search results
	X'00'	Search response not received
	X'01'	Positive explicit response
	X'02'	Positive *ANY response
	X'03'	Negative response
16		Sense code

Regular Route Selection control vector (RSCV) 46

This regular structure is used for a Route Selection control vector (RSCV) consisting of X'46' control vectors. This structure is used in BIND processing.

RSCV 46 is carried in BIND, RSP(BIND), and other RUs. It describes the path through an APPN network that a session is to take or has taken. RSCV 46 is sent and received by APPN nodes, but not by LEN nodes.

Table 6. Routing information RSCV 46

Byte	Hex value	Content
Header information for variable data		
0		Length of this type of variable data
2	X'02'	Key value for routing information (RSCV 46) variable data
Variable data		
3		RSCV length
4		RSCV key = X'2B'
5		Maximum hop count: the binary number of the transmission group descriptor or network name.

Table 6. Routing information RSCV 46 (continued)

Byte	Hex value	Content
6		Current hop count: the binary index number of the last transmission group descriptor control vector.
7-n		Control vectors
	X'46'	Transmission group descriptor control vector: one for each transmission group on the session path (present when the RSCV is carried on a BIND or RSP(BIND)).

Regular Route Selection control vector (RSCV) 0E

This regular structure is used for a Route Selection control vector (RSCV) consisting of X'0E' control vectors. This structure is used in search processing.

RSCV 0E is carried in search requests through an APPN network. RSCV 0E can be sent and received by APPN network nodes.

Table 7. Routing information RSCV 0E

Byte	Hex value	Content
Header information for variable data		
0		Length of this type of variable data
2	X'03'	Key value for routing information (RSCV 0E) variable data
Variable data		
3		RSCV length
4		RSCV key = X'2B'
5		Maximum hop count: the binary number of the transmission group descriptor or network name.
6		Current hop count: the binary index number of the last transmission group descriptor control vector.
7-n		Control vectors
	X'0E'	Control-point name control vector: one for each control point on the search path

Single hop route failure element

The single hop route element explains why the entries cannot be used. The structure of the single hop route element consists of the partner node for the single-hop-route request and an array of 255 entries that represents the status of the particular transmission groups.

Table 8. Single hop route information

Byte	Bits	Content
Header information for variable data		
0		Length of this type of variable data
2	X'04'	Key value for routing information variable data
Variable data		
3		Network identifier of the partner node

Table 8. Single hop route information (continued)

Byte	Bits	Content
B		Control-point name of partner node
13		The 255 entries (1 byte each) that represent the state of the transmission group
	X'00'	Transmission group number not defined
	X'01'	Transmission group is active but does not have the correct class-of-service characteristics
	X'02'	Transmission group is inactive but does have correct class-of-service characteristics
	X'03'	Transmission group is inactive and does not have correct class-of-service characteristics

Ineligible destination network nodes elements

Ineligible destination network nodes elements specify the reason why a particular transmission group that is returned by an end node is ineligible for providing access to the APPN network.

Note: There might be multiple elements supplied. The header information length should be used to determine when all elements have been processed. This information may be available for state 4060.

Table 9. No destination network nodes eligible information

Byte	Hex value	Content
Header information for variable data		
0		Length of this type of variable data
2	X'05'	Key value for routing information variable data
Variable data		
3		Network identifier of the ineligible destination network node
B		Control-point name of the ineligible destination network node
13		Transmission group number of ineligible destination node
14		Reason why transmission group is ineligible
	X'00'	Transmission group number not defined
	X'01'	Transmission group is active but does not have the correct class-of-service characteristics
	X'02'	Transmission group is inactive but does have correct class-of-service characteristics
	X'03'	Transmission group is inactive and does not have correct class-of-service characteristics

Destination node list

The destination node list structure specifies a single network-qualified control point name that represents one of the possible destinations (network nodes or virtual nodes) that, during route selection, cannot be reached.

Note: There may be multiple elements supplied. The header information length should be used to determine when all elements have been processed. This information may be available for states 4062, 4064, 4066, and 4068.

Table 10. Destination node list

Byte	Hex value	Content
Header information for variable data		
0		Length of this type of variable data
2	X'06'	Key value for routing information variable data
Variable data		
3		Network identifier of the destination node
B		Control-point name of destination node
13		Node type
	X'02'	Network node
	X'04'	Virtual node

User class-of-service with inactive transmission groups RSCV

This structure is used to represent a Route Selection control vector (RSCV) that allows inactive transmission groups. This structure has the same class-of-service characteristics as the class-of-service that is given by the user.

This RSCV is carried in BIND, RSP(BIND), and other RUs. It describes the path through an APPN network that a session is to take or has taken. This RSCV can be sent and received by APPN nodes, but not by LEN nodes.

Table 11. User class-of-service with inactive transmission groups

Byte	Hex value	Content
Header information for variable data		
0		Length of this type of variable data
2	X'07'	Key value for routing information variable data
Variable data		
3-4		RSCV length
5		RSCV key = X'2B'
6		Maximum hop count: the binary number of the transmission group descriptor or network name
7		Current hop count: the binary index number of transmission group the last descriptor control vector
8-n		Control vectors
	X'46'	Transmission group descriptor control vector: one for each transmission group on the session path
	X'47'	Transmission group characteristics of control vector: one for each transmission group on the session path (present when the RSCV is carried on a BIND or RSP(BIND))

Any class-of-service with active transmission groups RSCV

This structure represents a route selection control vector (RSCV) that allows active transmission groups, and also any class-of-service characteristics.

BIND, RSP(BIND), and other RUs carry this RSCV. It describes the path through an APPN network that a session is to take or has taken. APPN nodes send and receive this RSCV, but LEN nodes do not.

Table 12. User class-of-service with active transmission groups

Byte	Hex value	Content
Header information for variable data		
0		Length of this type of variable data
2	X'08'	Key value for routing information variable data
Variable data		
3-4		RSCV length
5		RSCV key = X'2B'
6		Maximum hop count: the binary number of the transmission group descriptor or network name
7		Current hop count: the binary index number of the last transmission group descriptor control vector
8-n		Control vectors
	X'46'	Transmission group descriptor control vector: one for each transmission group on the session path
	X'47'	Transmission group characteristics control vector: one for each transmission group on the session path (present when the RSCV is carried on a BIND or RSP(BIND))

Code license and disclaimer information

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This APPC, APPN, and HPR publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.



Product Number: 5770-SS1