

IBM Systems Network Architecture (SNA) Protocols

Background

IBM networking today consists of essentially two separate architectures that branch, more or less, from a common origin. Before contemporary networks existed, IBM's *Systems Network Architecture (SNA)* ruled the networking landscape, so it often is referred to as traditional or legacy SNA.

With the rise of personal computers, workstations, and client/server computing, the need for a peer-based networking strategy was addressed by IBM with the creation of *Advanced Peer-to-Peer Networking (APPN)* and *Advanced Program-to-Program Computing (APPC)*.

Although many of the legacy technologies associated with mainframe-based SNA have been brought into APPN-based networks, real differences exist. This chapter discusses each branch of the IBM networking environment, beginning with legacy SNA environments and following with a discussion of APPN. The chapter closes with summaries of the IBM Basic-Information Unit (BIU) and Path-Information Unit (PIU).

Note IBM-based routing strategies are covered in a separate chapter. Refer to Chapter 37, “IBM Systems Network Architecture (SNA) Routing” for details about IBM routing protocols.

Traditional SNA Environments

SNA was developed in the 1970s with an overall structure that parallels the OSI reference model. With SNA, a mainframe running *Advanced Communication Facility/Virtual Telecommunication Access Method (ACF/VTAM)* serves as the hub of an SNA network. ACF/VTAM is responsible for establishing all sessions and for activating and deactivating resources. In this environment, resources are explicitly predefined, thereby eliminating the requirement for broadcast traffic and minimizing header overhead. The underlying architecture and key components of traditional SNA networking are summarized in the sections that follow.

IBM SNA Architecture

IBM SNA-model components map closely to the OSI reference model. The descriptions that follow outline the role of each SNA component in providing connectivity among SNA entities.

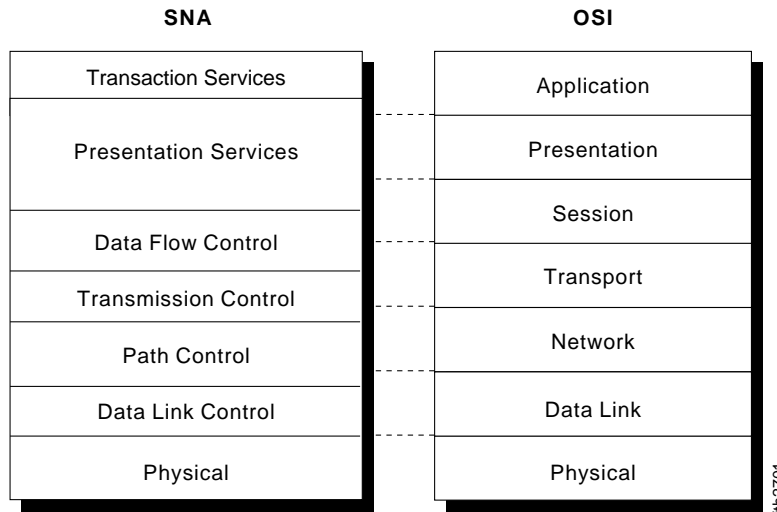
- *Data-link control (DLC)*—Defines several protocols, including the *Synchronous Data Link Control (SDLC)* protocol for hierarchical communication, and the *Token Ring Network* communication protocol for LAN communication between peers.

- *Path control*—Performs many OSI network-layer functions, including routing and datagram segmentation and reassembly (SAR).
- *Transmission control*—Provides a reliable end-to-end connection service, as well as encrypting and decrypting services.
- *Data flow control*—Manages request and response processing, determines whose turn it is to communicate, groups messages together, and interrupts data flow on request.
- *Presentation services*—Specifies data-transformation algorithms that translate data from one format to another, coordinate resource sharing, and synchronize transaction operations.
- *Transaction services*—Provides application services in the form of programs that implement distributed processing or management services.

Note SNA does not define specific protocols for its physical control layer. The physical control layer is assumed to be implemented via other standards.

Figure 29-1 illustrates how these elements of the IBM SNA model map to the general ISO OSI networking model.

Figure 29-1 IBM SNA maps to all seven levels of the OSI model.



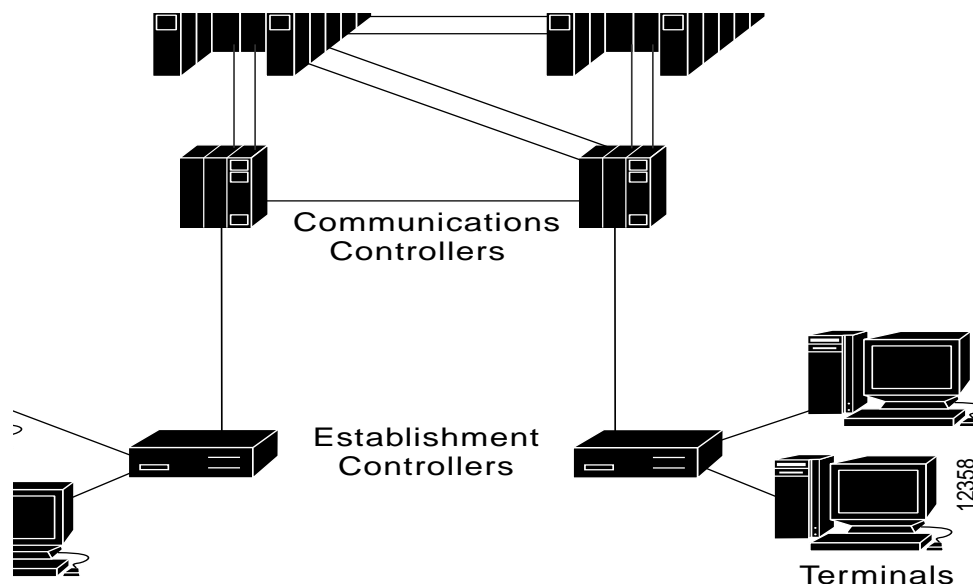
A key construct defined within the overall SNA network model is the *path control network*, which is responsible for moving information between SNA nodes and facilitating internetwork communication between nodes on different networks. The path control network environment uses functions provided by the path control and data-link control (DLC). The path control network is a subset of the IBM transport network.

IBM SNA Physical Entities

Traditional SNA physical entities assume one of the following four forms: *hosts*, *communications controllers*, *establishment controllers*, or *terminals*. Hosts in SNA control all or part of a network and typically provide computation, program execution, database access, directory services, and network management. (An example of a host device within a traditional SNA environment is an S/370 mainframe.) Communications controllers manage the physical network and control

communication links. In particular, communications controllers—also called *front-end processors* (FEPs)—are relied upon to route data through a traditional SNA network. (An example of a communications controller is a 3745.) Establishment controllers are commonly called *cluster controllers*. These devices control input and output operations of attached devices, such as terminals. (An example of an establishment controller is a 3174.) Terminals, also referred to as workstations, provide the user interface to the network. (A typical example would be a 3270. Figure 29-2 illustrates each of these physical entities in the context of a generalized SNA network diagram.)

Figure 29-2 SNA physical entities can assume one of four forms.



IBM SNA Data-Link Control

The SNA *data-link control* (DLC) layer supports a number of media, each of which is designed to provide access to devices and users with differing requirements. SNA-supported media types include mainframe channels, SDLC, X.25, and Token Ring, among other media.

A standard SNA mainframe channel attachment provides a parallel-data channel that uses *direct memory access* (DMA) data-movement techniques. A mainframe channel connects IBM hosts to each other and to communications controllers via multiwire cables. Each cable can be up to several hundred feet in length. A standard mainframe channel can transfer data at rate of 3 to 4.5 Mbps.

IBM's Enterprise Systems Connection (ESCON) mainframe attachment environment permits higher throughput and can cover greater physical distances. In general, ESCON transfers data at 18 Mbps and supports a point-to-point connection, ranging up to several kilometers, and transfers. To allow higher data rates and longer distances, ESCON uses optical fiber for its network medium.

SDLC has been widely implemented in SNA networks to interconnect communications and establishment controllers and to move data via telecommunications links.

X.25 networks have long been implemented for WAN interconnections. In general, an X.25 network is situated between two SNA nodes and is treated as a single link. SNA implements X.25 as the access protocol, and SNA nodes are considered adjacent to one another in the context of X.25

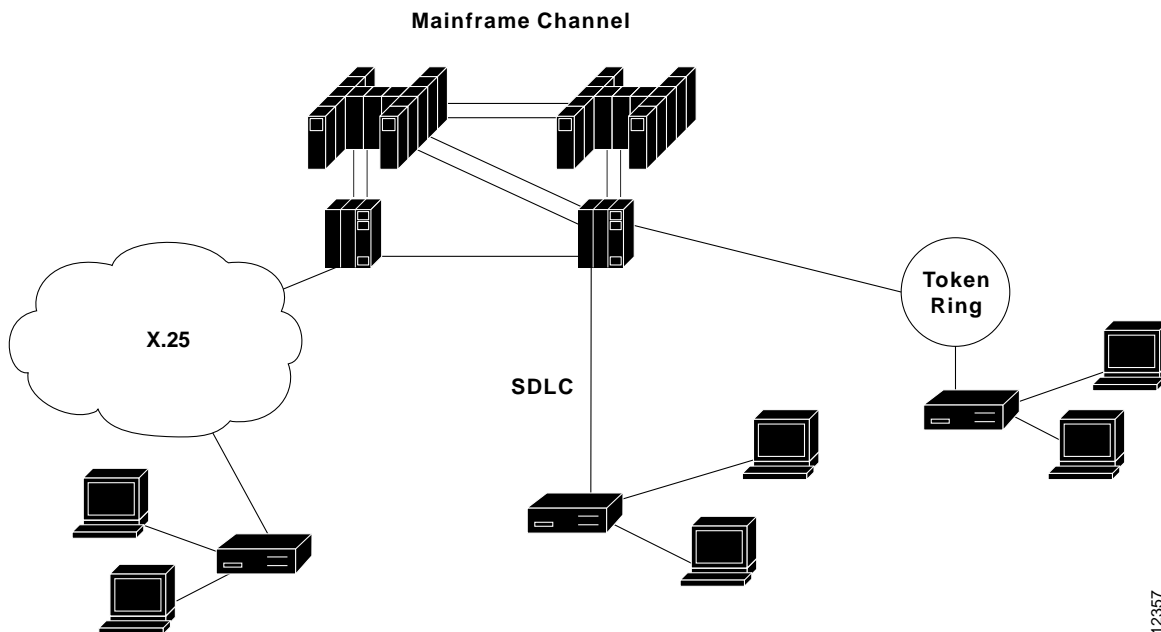
networks. To interconnect SNA nodes over an X.25-based WAN, SNA requires DLC-protocol capabilities that X.25 does not provide. Several specialized DLC protocols are employed to fill the gap, such as the physical services header, Qualified Logical Link Control (QLLC) and Enhanced Logical Link Control (ELLC).

Token Ring networks are the primary SNA DLC method for providing media access to LAN-based devices. Token Ring, as supported by IBM, is virtually the same as the IEEE 802.5 link-access protocol running under IEEE 802.2 Logical Link Control Type 2 (LLC2).

In addition to the basic suite of media types, IBM added support for several other widely implemented media, including IEEE 802.3/Ethernet, Fiber-Distributed Data Interface (FDDI), and Frame Relay.

Figure 29-3 illustrates how the various media generally fit into an SNA network.

Figure 29-3 SNA has evolved to support a variety of media.



12357

IBM Network Addressable Units (NAUs)

SNA defines three essential Network Addressable Units (NAUs): *logical units*, *physical units*, and *control points*. Each plays an important role in establishing connections between systems in an SNA network.

Logical units (LUs) function as end-user access ports into an SNA network. LUs provide users with access to network resources, and they manage the transmission of information between end users.

Physical units (PUs) are used to monitor and control attached network links and other network resources associated with a particular node. PUs are implemented on hosts by SNA access methods, such as the virtual telecommunication access method (VTAM). PUs also are implemented within communications controllers by network control programs (NCPs).

Control points (CPs) manage SNA nodes and their resources. CPs generally are differentiated from PUs in that CPs determine which actions must be taken, while PUs cause actions to occur. An example of a CP is the SNA system services control point (SSCP). An SSCP can be the CP residing in a PU 5 node or an SSCP as implemented under an SNA access method, such as VTAM.

IBM SNA Nodes

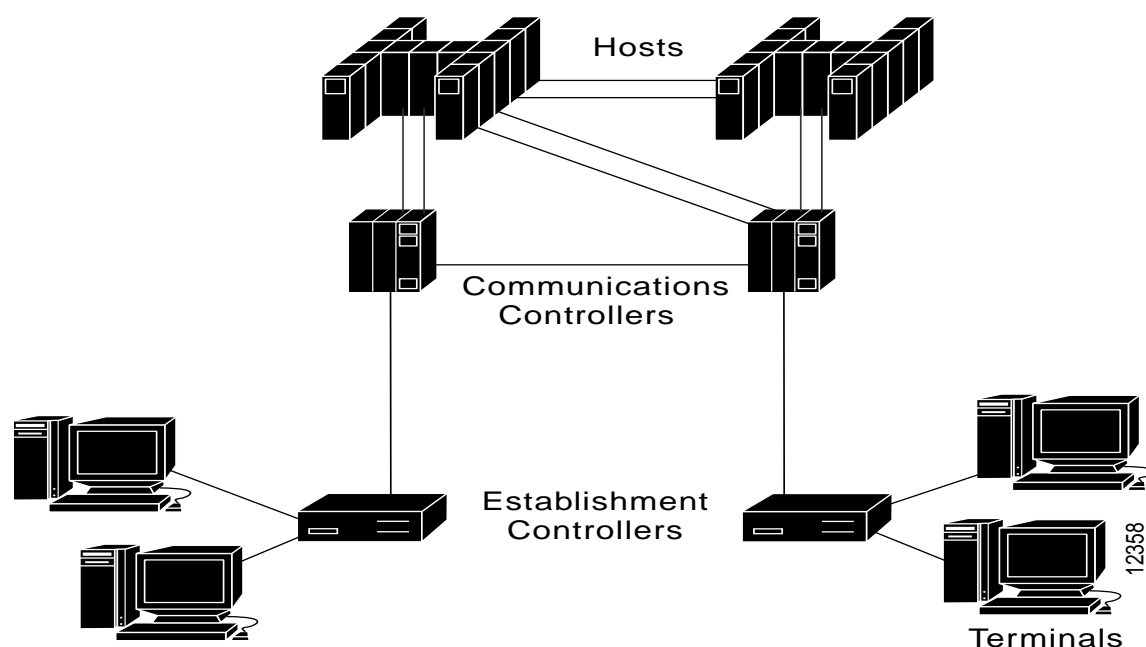
Traditional SNA nodes belong to one of two categories: *subarea nodes* and *peripheral nodes*. SNA subarea nodes provide all network services, including intermediate node routing and address mapping between local and network-wide addresses. No relationship exists between SNA node types and actual physical devices. Two subarea nodes are of particular interest: *node type 4* and *node type 5*.

Node type 4 (T4) usually is contained within a communications controller, such as a 3745. An example of a T4 node is an NCP, which routes data and controls flow between a front-end processor and other network resources.

Node type 5 (T5) usually is contained in a host, such as a S/370 mainframe. An example of a T5 node is the VTAM resident within an IBM mainframe. A VTAM controls the logical flow of data through a network, provides the interface between application subsystems and a network, and protects application subsystems from unauthorized access.

SNA peripheral nodes use only local addressing and communicate with other nodes through subarea nodes. Node type 2 (T2) is generally the peripheral node type of interest, although SNA does specify a node type 1 peripheral node. T2 typically resides in intelligent terminals (such as a 3270) or establishment controllers (such as a 3174). Node Type 1 (T1) is now obsolete, but when implemented, it resided in unintelligent terminals. Figure 29-4 illustrates the various SNA nodes and their relationships to each other.

Figure 29-4 Peripheral nodes communicate with other nodes through subarea nodes.



IBM Peer-Based Networking

Changes in networking and communications requirements caused IBM to evolve (and generally overhaul) many of the basic design characteristics of SNA. The emergence of peer-based networking entities (such as routers) resulted in a number of significant changes in SNA. Internetworking among SNA peers hinges on several IBM-developed networking components.

Advanced Peer-to-Peer Networking (APPN) represents IBM's second-generation SNA. In creating APPN, IBM moved SNA from a hierarchical, mainframe-centric environment to a peer-based networking environment. At the heart of APPN is an IBM architecture that supports peer-based communications, directory services, and routing between two or more APPC systems that are not directly attached.

APPN Components

In addition to the APPN environment, peer-based SNA networking specifies three additional key networking concepts: *logical units* (LUs), *Advanced Program-to-Program Computing* (APPC), and *node type 2.1*. Each plays an important role in the establishment of communication among SNA peers within the context of an SNA-based peer internetwork.

Logical Unit (LU) 6.2 governs peer-to-peer communications in an SNA environment. In addition, LU 6.2 supports general communication between programs in a distributed processing environment and between both similar and dissimilar node types. APPC enables SNA applications to communicate directly with peer SNA applications, and it provides a set of programming conventions and protocols that implement LU 6.2. Node type 2.1s (T2.1) are logical entities that permit direct communication among peripheral nodes capable of supporting T2.1. The T2.1 entity facilitates point-to-point communications by providing data-transport support for peer-level communications supported by APPC. In addition, a T2.1 contains a Peripheral Node Control Point (PNCP) that combines the traditional functions of a physical unit (PU) and control point (CP).

IBM APPN Node Types

Under APPN, peer-based communication occurs among several well-defined node types. These nodes can be broken down into three basic types: *low-entry nodes* (LENs), *end nodes* (ENs), and *network nodes* (NNs).

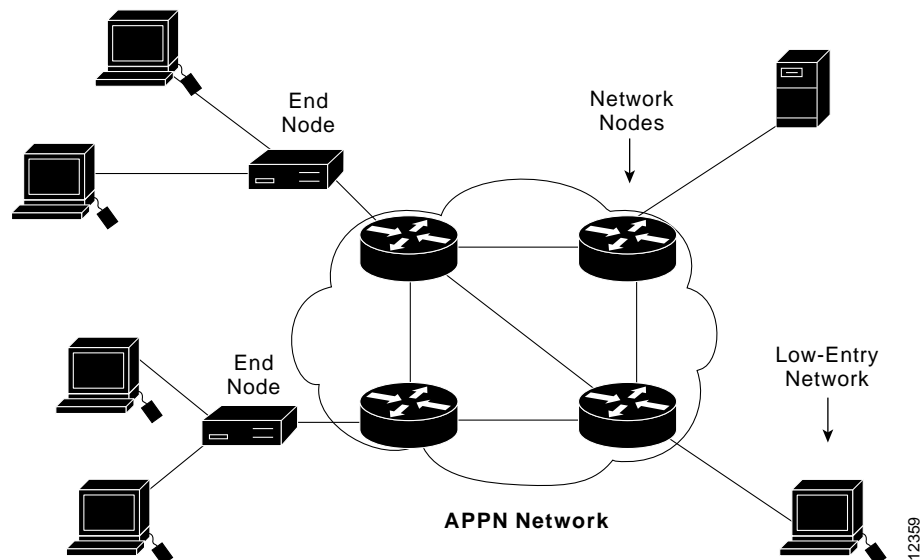
The low-entry network (LEN) node is a pre-APPN era peer-to-peer node. A LEN node participates in APPN networking by taking advantage of services provided by an adjacent network node (NN). The CP of the LEN node manages local resources but does not establish a CP-to-CP session with the adjacent NN. Before a session can start, session partners must be defined for a LEN node, and the LEN node must be defined for its service-providing adjacent NN.

An end node EN contains a subset of full APPN support. An end node accesses the network through an adjacent NN and uses the routing services of the same adjacent NN. To communicate on a network, an EN establishes a CP-to-CP session with an adjacent NN and uses the CP-to-CP session to register resources, request directory services, and request routing information.

A network node (NN) contains full APPN functionality. The CP in an NN manages the resources of the NN, as well as the attached ENs and LEN nodes. In addition, the CP in an NN establishes CP-to-CP sessions with adjacent ENs and NNs and maintains the network topology and directory databases created and updated by gathering information dynamically from adjacent NNs and ENs.

Figure 29-5 illustrates where each of these peers types might be found in a generalized APPN environment.

Figure 29-5 APPN supports several well-defined node types.



IBM APPN Services

Basic APPN services fall into four general categories: configuration, directors, topology, and routing and session services. Each is summarized in the sections that follow.

IBM APPN Configuration Services

APPN configuration services are responsible for activating connections to the APPN network. Connection activation involves establishing a connection, establishing a session, and selecting an adjacency option.

The connect phase of connection activation enables the initial establishment of communications between nodes. This initial communication involves exchanging characteristics and establishing roles, such as primary versus secondary. Connection establishment is accomplished by the transmission of *exchange identification type 3 (XID3)* frames between nodes.

During session establishment, CP-to-CP sessions are established with an adjacent EN or NN. Minimally, each node must establish at least one pair of CP-to-CP sessions with one adjacent node. An EN can establish a maximum of one pair of CP-to-CP sessions but can be attached to more than one NN. Between NNs, pairs of CP-to-CP sessions with all adjacent nodes or a subset of adjacent nodes can be established. The minimum requirement is a single pair of sessions to one adjacent NN, which ensures proper topology updating.

Adjacency among APPN nodes is determined by using CP-to-CP sessions. Two configurable options are available for determining node adjacency. A node can be specified as adjacent to a single node or as logically adjacent to every possible adjacent node. Selecting an adjacency option for a specific situation depends on a given network's connectivity requirements. The reduction in CP-to-CP sessions associated with single-node adjacency can reduce network overhead associated with topology updates, as well as the number of buffers required to distribute topology updates. Reducing the number of adjacent nodes, however, also increases the time required to synchronize routers.

IBM APPN Directory Services

Directory services are intended to help network devices locate service providers. These services are essential to establishing a session between end users. Directory services in APPN call for each NN to maintain a directory of local resources and a network directory that associate end users with NNs providing service. A distributed directory service then is formed from the collection of individual NN network directories. This section summarizes the nature of APPN databases, node-directory service handling, and the role of a centralized directory service.

The local and network directory databases support three service-entry types: *configured entries*, *registered entries*, and *cached entries*. Configured database entries usually are local low-entry network nodes that must be configured because no CP-to-CP session can be established over which to exchange information. Other nodes might be configured to reduce broadcast traffic, which is generated as part of the discovery process. Registered entries are local resource entries about which an end node has informed its associated network node server when CP-to-CP sessions are established. A registered entry is added by an NN to its local directory. Cached database entries are directory entries created as session requests and received by an NN. The total number of cached entries permitted can be controlled through user configurations to manage memory requirements.

The end-node directory service-negotiation process involves several steps. An EN first sends a LOCATE request to the NN providing network services. The local and network directory databases next are searched to determine whether the destination end user is already known. If the destination end-user is known, a single directed LOCATE request is sent to ensure its current availability. If the destination end-user is not found in the existing databases, the NN sends a LOCATE request to adjacent ENs to determine whether the destination end user is a local resource. If the destination is not local, the NN sends a broadcast LOCATE request to all adjacent NNs for propagation throughout the network. A message is sent back to the originating NN indicating that the destination is found when the NN providing network services for the destination end-user locates the end-user resource. Finally, both origin and destination NNs cache the information.

Directory services for LEN nodes are handled by a proxy service process. First, a LEN node sends a bind session (BIND) request for attached resources. This contrasts with the LOCATE request sent by ENs. To receive any directory services, an NN must provide proxy services for a LEN node. When a proxy service NN is linked to the LEN node, the NN broadcasts LOCATE requests as needed for the LEN node.

A central directory service usually exists within an ACF/VTAM and usually is implemented to help minimize LOCATE broadcasts. This kind of database can be used to maintain a centrally located directory for an entire network because it contains configured, registered, and cached entries. Under a centralized directory service process, an NN sends a directed LOCATE broadcast to the central directory server, which then searches the central database and broadcasts when necessary.

IBM APPN Topology and Routing Services

In an APPN network topology, network nodes are connected by *transmission groups (TGs)*. Each TG consists of a single link, and all NNs maintain a network-topology database that contains a complete picture of all NNs and TGs in the network. Transmission groups are discussed in Chapter 37, "IBM Systems Network Architecture (SNA) Routing."

A network's topology database is updated by information received in a *topology-database update (TDU)* message. These TDU messages flow over CP-to-CP sessions whenever a change occurs in the network, such as when a node or link becomes active or inactive, when congestion occurs, or when resources are limited.

The network-topology database contains information used when calculating routes with a particular *class of service (COS)*. This information includes NN and TG connectivity and status, and NN and TG characteristics, such as TG capacity.

APPN's routing services function uses information from directory and topology databases to determine a route based on COS. Route determination starts when an end node first receives a session request from a logical unit. A LOCATE request is sent from the EN to its NN to request destination information and to obtain a route through the network. The NN then identifies the properties associated with the requested level of service. The identified properties are compared to properties of each TG and NN in the network, and all routes that meet the specified criteria are cached as applicable. Each EN, NN, and TG in the network is assigned a weight based on COS properties, such as capacity, cost, security, and delay. Properties also can be user-defined. Finally, a least-cost path is determined by totaling the weights on the paths that meet the routing criteria.

IBM APPN Session Services

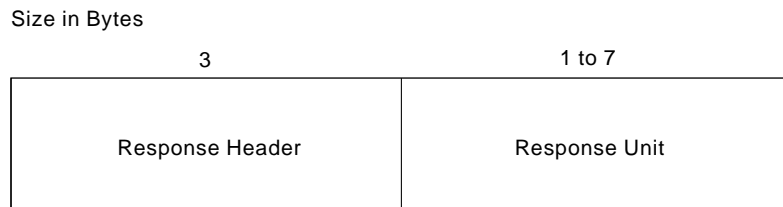
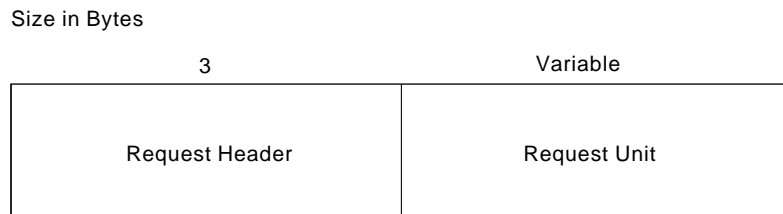
Following route establishment, the APPN session-establishment process varies depending on the node type. If the originating end user is attached to an EN, a LOCATE reply containing the location of the destination and route is returned to the originating EN by the NN adjacent to the destination EN. The originating EN then sends a BIND on a session route. If originating, the end user is attached to a LEN node that sends a BIND to its adjacent NN. The adjacent NN converts the LEN BIND to APPN BIND and sends a BIND on the session path.

Note A BIND is a specific type of request message sent from one LU to another LU. A BIND carries the route being used for a session. It specifies NNs and TGs, a unique session identifier for each TG, the transmission priority for the session, and window information to support adaptive pacing to limit traffic on the network.

Basic Information Unit (BIU) Format

IBM SNA NAUs employ basic information units (BIUs) to exchange requests and responses. Figure 29-6 illustrates the BIU format.

Figure 29-6 A basic information unit (BIU) can be either a request or response.



ith2706

BIU Fields

The following field descriptions summarize the content of the BIU, as illustrated in Figure 29-6:

- *Request Header*—Identifies the type of data in the associated request units. This header provides information about the format of the data and specifies protocols for the session. Only NAUs use Request Header information.
- *Request Unit*—Contains either end-user data or SNA commands. End-user data is sent in data-request units. SNA commands are sent in command-request units that control the network and contain information exchanged between end users.
- *Response Header*—Identifies the type of data associated with the response unit. The request/response indicator bit distinguishes a response header from a request header. A receiving NAU indicates whether the response being returned to the request sender is positive or negative by setting the *response-type indicator* (RTI) bit in the response header.
- *Response Unit*—Contains information about the request indicating either a positive or negative response. Positive responses to command requests usually contain a 1- to 3-byte response unit that identifies the command request. Positive responses to data requests contain response headers but no response unit.

Negative response units are 4 to 7 bytes long and always are returned with a negative response. A receiving NAU returns a negative response to the requesting NAU under one of three conditions:

- Sender violates SNA protocol
- Receiver does not understand the transmission
- Unusual condition, such as a path failure, occurs

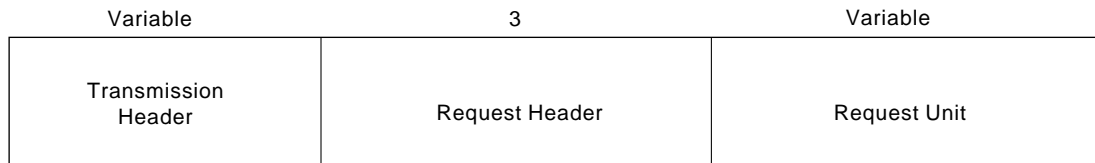
When a negative response is transmitted, the first 4 bytes of a response unit contain data that explains why the request is unacceptable. The receiving NAU sends up to 3 additional bytes that identify the rejected request.

Path Information Unit (PIU) Format

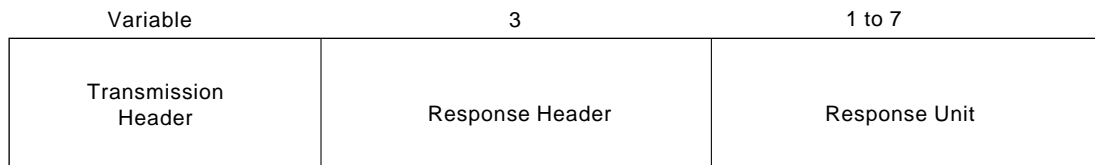
The path information unit (PIU) is an SNA message unit formed by path-control elements by adding a transmission header to a BIU. Figure 29-7 illustrates the PIU format.

Figure 29-7 The path information unit (PIU) requests and responses each consist of three fields.

Size in Bytes



Size in Bytes



itn2707

PIU Fields

The following field descriptions summarize the content of the PIU, as illustrated in Figure 29-7:

- *Transmission Header*— Routes message units through the network. This header contains routing information for traditional SNA subarea networking. Transmission-header formats are differentiated by the *format identification* (FID) type. Path control uses the FID types to route data among SNA nodes.

Three FID types are implemented in PIUs:

- FID0 is used to route data between adjacent subarea nodes for non-SNA devices. FID0 generally is rendered obsolete by the FID4 bit set to indicate whether a device is an SNA or non-SNA device.
- FID1 is used to route data between adjacent subarea nodes when one or both of the nodes do not support explicit and virtual route protocols.
- FID2 is used to route data between a subarea boundary node and an adjacent peripheral node, or between adjacent type 2.1 nodes.
- In general, the transmission header is used to route data between adjacent subarea nodes when both the subarea nodes support explicit and virtual route protocols.

- *Request Header*— Identifies the type of data in the associated request units. This header provides information about the format of the data and specifies protocols for the session. Only NAUs use request header information.
- *Request Unit*— Contains either end-user data or SNA commands. End-user data is sent in data-request units. SNA commands are sent in command-request units that control the network and contain information exchanged between end-users.

- *Response Header*—Identifies the type of data associated with a response unit. The request/response indicator bit distinguishes a response header from a request header. A receiving NAU indicates whether the response being returned to the request sender is positive or negative by setting the RTI bit in the response header.
- *Response Unit*—Contains information about the request indicating either a positive or negative response. Positive responses to command requests usually contain a 1- to 3-byte response unit that identifies the command request. Positive responses to data requests contain response headers but no response unit.

Negative response units are 4 to 7 bytes long and always are returned with a negative response. A receiving NAU returns a negative response to the requesting NAU under one of three conditions: The sender violates SNA protocol, a receiver does not understand the transmission, or an unusual condition, such as a path failure, occurs.

When a negative response is transmitted, the first 4 bytes of a response unit contain data that explains why the request is unacceptable. The receiving NAU sends up to 3 additional bytes that identify the rejected request.