Advances in APPN architecture

by R. Bird
C. Brotman
R. Case
G. Dudley
R. E. Moore
M. Peters

In this paper, we discuss the evolving environment and requirements for the Advanced Peer-to-Peer Networking™ (APPN®) architecture and the accommodation of these changes in basic directory, topology, and configuration services, as well as in application transport capabilities. We use high-performance routing, a recent APPN extension, as an example of the adaptation of the architecture to emerging high-speed communication facilities and the increasing trend to multiprotocol networks. Finally, we discuss some extensions to switched support and network management in APPN and speculate on possible future considerations.

During the 1970s, both engineering and economics dictated the off-loading of communications functions from mainframes to specialized front-end processors. By the early 1980s, the steady decline in computing price and performance led to the increasing popularity and ubiquity of powerful midrange and small systems. IBM architects foresaw the need for new networking support for an emerging class of applications that would eventually dominate computing: distributed applications using program-to-program communication (for example, client/server—a trend that, over ten years later, is increasingly pervasive, but still not ubiquitous). This programming style is exemplified by Systems Network Architecture (SNA) logical unit type 6.2 (LU 6.2), also called Advanced Program-to-Program Communication (APPC). This new programming style contrasted sharply with the previous program-to-device networking model, which maintained data at central sites and provided access via centrally controlled display terminals and printers.

Extrapolating from economic and technology trends, the architects envisioned a network of autonomous systems that would complement the centrally managed, centrally administered, mainframe-centric model of computing. These trends took shape in the form of personal computer-based local area networks (LANs). But the protocols that were emerging for the LAN-only environment—such as Network Basic Input/Output System (NetBIOS)—had several shortcomings that made them unsuitable for use in a heterogeneous network that included wide-area connectivity. Harnessing the distributed computing power inherent in the new computing systems, and leveraging the new program-to-program communication, the IBM architects defined the Advanced Peer-to-Peer Networking* (APPN*) architecture. Its new distributed algorithms for automating network control and maintenance have resulted in self-managing, adaptable networks, capable of dynamically learning everything needed for their own efficient operation. APPN architecture was specifically designed to

0018-8670/95/$3.00 © 1995 IBM

avoid the limitations of existing LAN-based protocols. For example, a limitation of NetBIOS is its use of LAN broadcasts for address discovery. While broadcasts provide good function at a low cost in small LANs, they severely limit the ability to increase the size of the LAN internetwork. By contrast, APPN uses distributed algorithms for address resolution and route selection that do not rely on LAN broadcasts, and hence are more scalable. APPN is not dependent on any functions that are available only on LANs, nor does it assume, or require, any particular configuration such as hub-and-spokes, mesh, or logical ring, as do some other protocols. As we shall see, this independence from the underlying transmission medium gives APPN today a powerful head start toward exploiting new link technologies such as asynchronous transfer mode (ATM).

The first steps toward APPN architecture were concerned with providing rudimentary networking support for autonomous computers that offered the new program-to-program communication capability. This early level of networking support for APPC, called low-entry networking (LEN), assumed that a small computer could either establish a direct communication link to any partner it needed to reach (when all communicating systems were adjacent) or be adjacent to a traditional subarea SNA backbone network[1] capable of routing its traffic using connections, or hops, through adjacent intermediate nodes.

More sophisticated networking support—APPN—was on the drawing board as the first LEN products were installed. It was assumed that these small and midrange computers would often be located in small businesses or decentralized sites that did not need, or could not afford, a dedicated networking support staff. Every computer was theoretically capable of communicating with any other as a peer; ad hoc patterns of connection and disconnection would occur. With these assumptions, several new requirements for networks emerged as critical:[2]

- Networks should be easy to use, change, manage, and grow.
- Network control should be decentralized.
- Any topology should be possible.
- Networks should allow broad flexibility for physical-level attachments.
- Internetworking with subarea SNA should be supported.
- Simplicity and low cost are important.
- Continuous operation is essential.

Some of the technical solutions that emerged to solve these requirements include the following.

- *Autonomy*—The type of networking definitions requiring coordination at multiple sites was minimized or eliminated. For example, most parameters required to activate a link and establish addressability between a pair of computers are negotiated, rather than predefined.
- *Client/server model*—To optimize overall cost and performance, networking functions were divided between *end nodes* and *network nodes*. Rather than devote storage and machine cycles everywhere to routine network maintenance chores, such as maintaining a replicated topology database used for route selection and a distributed directory used to learn the locations of application programs and other network resources, end node *clients* use the services of a network node *server*. This division of function allows flexibility while reducing the cost of the majority of nodes in the network—the application processors.
- *Arbitrary topology*—The overall network topology is no longer assumed to be preplanned or centrally administered. Each machine user controls the membership of that machine in the network, reflecting the way small and intermediate systems are often used. This results in a distributed topology management algorithm enabling nodes to join and leave the network at will, while preserving the ability to route traffic over currently available paths. The topology of network nodes and their interconnecting links is replicated at all network nodes, which update each other as changes occur. The topology of end nodes is known at their individual network node servers; this information is reported as needed during directory searches to allow selection of optimal session routes between end users.
- *Flexibility*—It was unreasonable to think that each computer user would be capable of predefining everything needed to route messages to a given partner; users might not even know in advance which partners they would contact. A distributed directory database with a query algorithm enabled users and programs to locate each other and obtain necessary routing information without any predefinitions other than partners' names.
- *Adaptive buffer management*—The adaptive session-level pacing function developed for APPC,

which worked so well to manage buffers and prevent storage-related deadlocks between communicating programs, was applied to hop-by-hop flow control for intermediate nodes, providing a measure of distributed self-tuning in the face of unpredictable traffic patterns.

In APPN, network nodes perform intermediate routing and exchange topology information among themselves, while end nodes do not. An end node establishes a special connection with one of the adjacent network nodes, which provides services for resource registration, discovery of remote network resources, route selection, and session setup procedures on behalf of application programs in the end node. Additional control connections are established between logically adjacent network nodes to exchange and propagate topology information, to support resource searches, and more.

Changing times have brought new technologies, new user environments, and new user requirements.

The overall APPN design proved so successful, first in a prototype version on the System/36*,[3] that it was later deployed on all major IBM networking platforms, from application hosts ranging from the System/390* mainframe family to the Personal System/2* with Operating System/2* and Communications Manager/2, to specialized communication nodes including the 3174 terminal controller, the 3746 Nways* Multinetwork Controller, and the 2217 Multiprotocol Concentrator. Today APPN is the vehicle for meeting customer requirements for 100 percent host availability and exploiting the powerful capabilities of the newest parallel sysplex mainframe hardware from IBM. APPN has come a long way since IBM researchers first envisioned it as the "glue" in networks of small systems.[4]

New environments, new requirements

Changing times have brought new technologies, new user environments, and, consequently, new user requirements. While the original design points for APPN still hold, additional ones were needed to address these changes.

For networking technologies, the driving forces of change have been in two major areas: processing technologies and link technologies.

Processing capability has increased and become less expensive, thus moving the balance of computing power away from the data center and out to the fringes of the network. This further drives the need for peer-to-peer networks of increasing size, as well as for paradigms such as client/server and distributed computing. Today APPN networks may contain hundreds of network nodes, which replicate a common network-node topology and coordinate directory searches among themselves. Continued growth of a given network may be fueled by the growth of the using enterprise, consolidation of multiple enterprises, or addition of new applications and data traffic. As the size of an APPN network continues to increase, the increase in network resources required to replicate and maintain the network-node topology among all network nodes and to do required broadcast directory searches to find communication partners may become burdensome. As a result, requirements have emerged for ways to keep networks manageable and efficient, and to isolate network topology information within logically or corporately distinct subnetworks. The APPN border node provides for topology isolation and growth, while the APPN central directory server helps reduce directory search overhead. These facilities are discussed later in this paper.

Links, incorporating the physical media across which the data flow, have seen orders of magnitude improvement in both capacity and reliability. Typical links in the early 1980s, made of copper, had a typical performance of 9.6–56 kilobits per second (Kbps) and bit-error rates of 10^{-5}. Contrast this with modern fiber-optic links, which may support gigabyte bandwidths with bit-error rates on the order of 10^{-12}.

Improvements in link and processing technologies have driven changes in networking protocols. For example, due to the relatively high error rates, earlier networking protocols required significant processing overhead for error detection and recovery, typically at each node of the network. The im-

proved reliability of today's links means that error checking need not be done link by link. High-performance routing (HPR),[5] an enhancement to APPN, allows checking and recovery to be eliminated from the interior nodes of a network and performed only at the endpoints. This reduction of checking overhead, coupled with intrinsically lower error rates, can significantly lessen the overall time the distributed processing resource is used for reliability purposes.

While the lower error rates of today's links reduce processing for error checking and recovery, their increased bandwidth reduces the time a node can spend, per byte of information, performing functions such as routing and pacing. It also means that more data can be physically on the link ("in the pipe") at any given instant. This can drive buffer requirements in the nodes higher, especially using protocols designed for slower links. With cross-country or even intercontinental fiber-optic links running at gigabit speeds, megabytes of data can be in flight from the sender before the first byte of data arrives at the receiver. If it takes tens of milliseconds for data to traverse a link and megabytes of data could be flowing during that time, a protocol that requires frequent responses will not use the link efficiently. The Rapid Transport Protocol of HPR addresses this.

One result of increased link bandwidth is the potential for higher numbers of messages or circuits to traverse a node, again driving up processing needs. So while the improved reliability of modern link technologies reduces error processing, the increased bandwidth may consume more computation than before. This could limit the ability of networks to make full and efficient use of the promised bandwidth gains. The automatic network routing function of HPR addresses this problem.

Changes in the way networks are used also drive new requirements. One such change is the trend toward networks that are more heterogeneous in protocols, nodes, and links. Whereas, in the early days of SNA, an IBM customer's network would typically have been homogeneous—using SNA subarea protocols, IBM front-end processors, IBM display controllers, and Synchronous Data Link Control (SDLC) twisted-pair copper links—today's networks might consist of many different protocols (for example, SNA subarea, APPN, Transmission Control Protocol/Internet Protocol [TCP/IP], NetBIOS, IPX**, AppleTalk**, etc.), many different computation nodes (controllers, routers, hubs, bridges—all potentially from different companies), and a wide variety of link types and speeds—from 2400-bits per second (bps) modem phone lines, to 16-megabits per second (Mbps) token-ring LANs, to 1.2-gigabits per second (Gbps) fiber-optic links. Such diversity, particularly in this era of network consolidation, has become the norm. All these diverse features must coexist and interoperate, sometimes in neatly partitioned subnetworks, other times in an undisciplined mesh of nodes. The dependent LU requester enhancement allows APPN to support traditional program-to-device applications along with newer program-to-program applications. APPN with HPR is capable of running efficiently on a wide variety of platforms, coexisting with other protocols, and exploiting existing and newly emergent technologies, such as integrated services digital network (ISDN), frame relay, and ATM. A later section discusses the ongoing evolution of APPN in support of these switched technologies.

In addition to consolidation of different networking technologies, today's users and network managers want portability, "openness," and interoperability. Portability means that the technology is available on a wide variety of platforms. APPN, for example, runs on personal computers, Application System/400* (AS/400*) systems, UNIX** workstations, and specialized networking hardware, as well as on mainframes. Openness means that the technology is readily available and not exclusive to one manufacturer or vendor. Interoperability means that the different products available in the marketplace will coexist and work correctly together. IBM has made APPN an open standard. The *APPN Implementers' Workshop* (AIW), initiated by IBM in 1993, is a consortium of over 40 networking product manufacturers dedicated to ensuring the interoperability of high-quality APPN products from a wide variety of vendors. The AIW exists not only for openly sharing APPN specifications and implementation experiences, but for cooperatively developing extensions and modifications to the architecture.

As networks get larger, customers face the daunting task of managing increasingly complex configurations. New networking tools and techniques are necessary to deal with this challenge. In a later section, we describe how APPN has expanded its scope to include new management disciplines and to em-

brace the two major open management architectures in the marketplace.

The remainder of the paper takes up APPN advances in the following order. First, we look at extensions improving APPN basic directory, configuration, and application-transport capabilities. Then we examine some details of HPR and describe recent accommodation by APPN of new switching technologies. After discussing the expanded management services, we end by considering some possible future extensions.

Central directory

The APPN directory uses a partially distributed database. The directory entries are distributed using a two-level hierarchy. All nodes keep a directory of LU names that are local to the node. In addition, end nodes register their directory entries with a network node server. A network node server maintains a second-level directory of all LUs on all end nodes it serves. To establish sessions between LUs on nodes served by different network nodes, directories exchange information using *Locate* flows. Locate flows may be broadcast to all network nodes or directed to a specific node based on directory entries cached from previous flows.

Central directory services add a third level to the APPN directory services hierarchy. Network nodes may register resources with a central directory, much as end nodes may register resources with their network node servers. Network nodes also use the services of a central directory to locate resources for session initiation. Network nodes rely on a central directory server, if one is present, to resolve unknown LU names. The central directory server takes over all directory broadcast responsibilities. This allows it to expand its cache of LU names and locations. Peer central directories can cooperate to divide workload and provide backup for one another.

This extension is an example of how the architecture is readily enhanced. Locate flows are used to centrally register resources in much the same way that they are used to locate resources. A central directory server need not be adjacent to the network nodes using its services. It is identified in the topology database, so all network nodes can find it.[6] This architecture allows a specialized network node acting as a central directory to provide services that can dramatically reduce the number of broadcasts in an APPN network, without compromising any of the dynamics or decentralized control.

Border node

APPN network nodes and the links connecting them are represented in a topology database. The information in this database is used to select optimal routes for sessions, based on the requirements of the data and the characteristics of the links. The overhead of maintaining a complete and accurate replicated database is minimized by sending topology updates only when changes occur. As a network grows, the size of the database grows with the number of network nodes and the number of links connecting those nodes. This places a growing demand on the storage at the network nodes. Since topology database updates are sent to every network node for every change to the network-node topology, the demand placed on the processing resources at network nodes and on the links connecting them increases with larger networks as well.

These scaling constraints impose an effective limit on the size of an APPN network that can be built with nodes and links of a given capacity. Since the topology data and algorithm are fully replicated, this limit is based on the storage and processing capabilities of the least capable node in the network.

A *border node* provides an extension to APPN architecture to allow networks to be partitioned into separate topology domains, or *subnets*. Networks can be divided into topology subnets according to any policies or criteria. A topology subnet is simply a group of APPN network nodes that share a common topology database. A border node provides a service (topology subnet connectivity) to all LUs without requiring this service in all nodes. Border nodes allow directory searches and sessions to span interconnected topology subnets while limiting topology flows.

In practice, there are many reasons to partition networks long before the scalability limits are reached. Although APPN supports networks of arbitrary topology, network designs are rarely arbitrary. Network designs are constrained by geography, availability of connection services, and expected traffic. For example, LANs at many remote sites may be connected via leased lines to a central office. In

this case, it is not necessary for network nodes to know the topology of remote nodes in all other sites, since all routes are concentrated through the central office.

When two different enterprises are connected, a network administrator may want to prevent one network from having to process topology updates from another network. An administrator may also wish to prevent administrators from other networks from being able to view his or her network topology.

Configuration flexibility. A border node is an enhanced APPN network node; therefore it supports all of the attachment types that a network node supports. It may be directly connected to other network nodes, to end nodes, or to LEN nodes in its own topology subnet. A border node is also connected to a network node or a border node in a different topology subnet.

Topology isolation. A border node connects different topology subnets, but it is a member of only one subnet (its *native* subnet). It assumes an end-node role for all directory and topology flows to adjacent (*nonnative*) subnets. This takes advantage of the distribution of function between end nodes and network nodes in a way that was not originally included in the architecture. Topology information does not flow between network nodes and end nodes, but directory requests do, and end-user (LU-LU) sessions may be established. This same limited connectivity is desired between APPN topology subnets, and it is accomplished by the presence of the border node. Control connections are established and capabilities are negotiated as between basic end nodes and network nodes.

A border node participates in topology algorithms in its native topology subnet as a normal network node. Links connecting the border node to other topology subnets are defined as *intersubnet links*. No topology updates are sent over an intersubnet link, even though control connections are active.

Directory requests. Locate flows may be sent over intersubnet links, so the APPN distributed directory algorithm may span multiple topology subnets. A border node may forward a broadcast Locate flow to all adjacent topology subnets or to selected subnets, based on the name of the LU to be located.

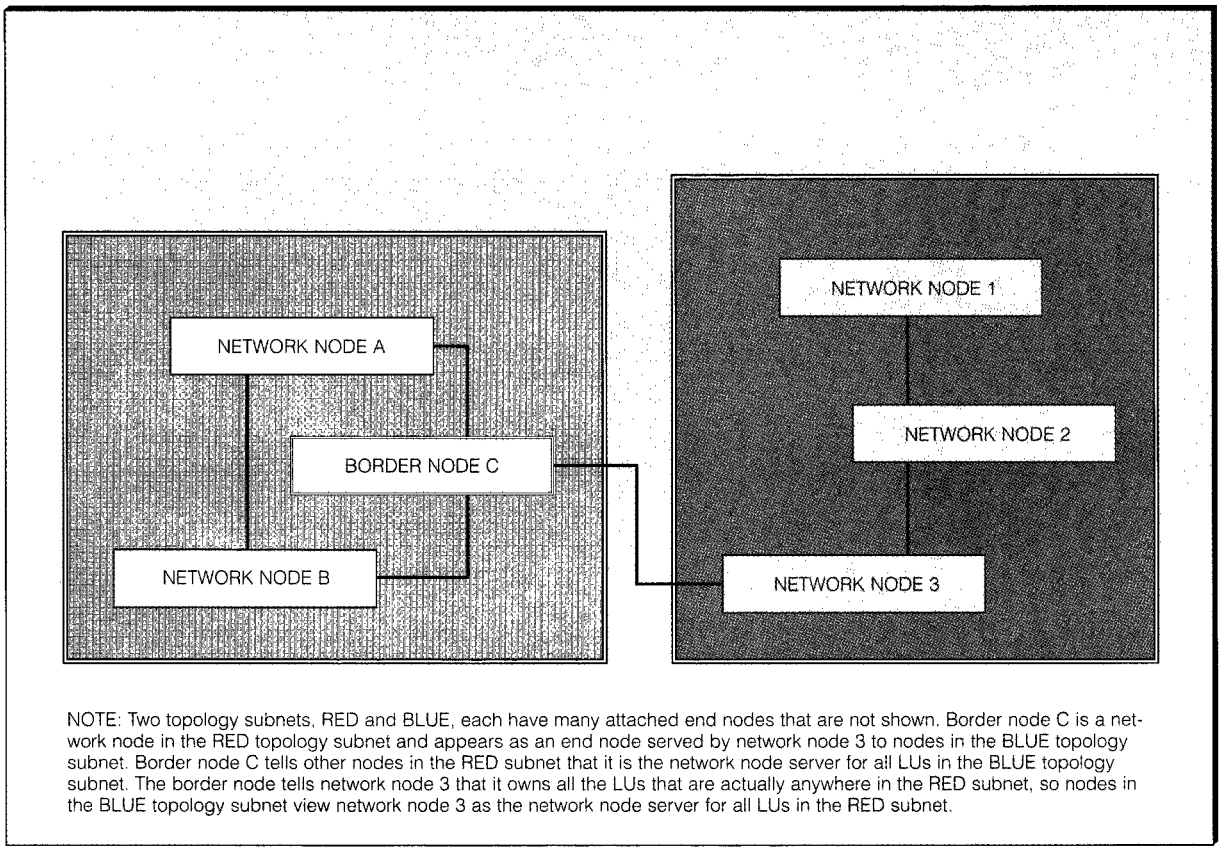
SNA names are hierarchically structured. Fully qualified LU names are represented as a *net identifier* and an *LU name*. All network nodes in a topology subnet share the same net identifier for their control points, or *CP*s, which provide network services and communicate with each other using control connections, or *CP-CP sessions*. While not architecturally required, it is common for all LUs in the domain of a network node (all LUs on nodes served by that network node) to have the same net identifier. Since all LUs in a topology subnet are likely to have the same net identifier, the net identifier can be useful at a border node to decide whether to forward Locate flows into a particular topology subnet or not.

Since intersubnet sessions will always traverse intersubnet links, border nodes that own these links forward a broadcast Locate flow across them only if the links are suitable for the requested class of service for the session.

Intersubnet routing. APPN routes are calculated based on information stored in the topology database. Since topology information is limited to nodes and links in the native topology subnet, routes can be calculated only within the native subnet. As noted earlier, a border node handles Locate flows as an end node served by a network node in the adjacent topology subnet. It also provides network node services in its own topology subnet on behalf of LUs whose real network node server is in a different topology subnet, as shown in Figure 1. This allows the border node to perform all the functions required to establish intersubnet routes without affecting the route-selection algorithm in any other network nodes. However, this also means that there is no way to select a globally optimal route for a session connecting LUs in different topology subnets. The route is only "piecewise optimal." That is, it is optimal within each topology subnet on the path, but requires good network design and placement of border nodes to ensure near-optimal routes end-to-end.

A border node alters information in the Locate flows as they pass through it. It replaces the real network node server of the partner LU with itself for all flows into its native topology subnet. This gives the appearance that the LUs have been located on the border node. When a session is to be set up with an LU in another topology subnet, the native piece of the route is calculated and saved. When a session initiation request, or *BIND*, arrives, the border node correlates the BIND with the route calculated for the previous Locate procedure.

Figure 1 A simple border node configuration



NOTE: Two topology subnets, RED and BLUE, each have many attached end nodes that are not shown. Border node C is a network node in the RED topology subnet and appears as an end node served by network node 3 to nodes in the BLUE topology subnet. Border node C tells other nodes in the RED subnet that it is the network node server for all LUs in the BLUE topology subnet. The border node tells network node 3 that it owns all the LUs that are actually anywhere in the RED subnet, so nodes in the BLUE topology subnet view network node 3 as the network node server for all LUs in the RED subnet.

Since a BIND might not occur for a given Locate procedure, the border node discards this information if it is not used in a timely fashion.

When the BIND arrives, it is also altered by replacing the routing information for the piece of the route in one topology subnet with the information for the piece of the route in the next subnet. This hides the topology information in Locate and BIND flows from the adjacent topology subnet. It also limits the amount of routing information by not including all hops of the entire route.

Since a border node provides these functions for sessions and alters flows as they pass from one topology subnet to the next, routes between endpoints in different topology subnets must traverse border nodes on the path.

Dependent LU requester

While its initial designers planned for the interoperability of APPN with SNA subarea networks and designed APPN packet formats to support routing of the older program-to-device traffic,[7] full support was not completed quickly, perhaps because of overly optimistic predictions about how quickly the client/server computing model would replace the mainframe-interactive model. One relatively recent significant addition to APPN, therefore, was the set of functions designed to overlay and integrate the subarea SNA hierarchical network model required by system services control point (SSCP)-dependent LUs on the flexible, nonhierarchical, distributed APPN infrastructure, without reducing APPN ease-of-use.[8] One challenge came in migrating the older SNA session-setup flows required by dependent LUs to a pure APPN environment. These protocols, once

conducted only on SSCP-based sessions involving mainframes, support sophisticated session-management programs written to the Virtual Telecommunications Access Method (VTAM*) application programming interface. Such programs incorporate functions for queuing session-establishment requests for busy single-session-only devices, and for initiation of sessions by a secondary LU (e.g., a terminal is powered on and automatically receives a logon screen), or by a third party (e.g., a logon-management application program). Such functions could not simply be declared obsolete, since they were totally integrated into critical customer business processes in the form of many millions of lines of application code; also they provide an unparalleled degree of network control that cannot be replaced easily. This challenge is being addressed by *Session Services Extensions*, an extension to APPN protocols in 1993 supported in VTAM V4R1 now being introduced to the APPN Implementers' Workshop.

Even with V4R1—barring gateways or bridging—dependent devices still had to be adjacent to the host providing them SSCP services, that is, adjacent from the perspective of the SNA network routing layer, called "path control" (Open Systems Interconnection [OSI] layer 3).[9] The second challenge, therefore, was to remove restrictions on where dependent LUs could be attached in an APPN network without introducing poorly performing schemes such as protocol encapsulation, or the indirection and limited scalability of bridging. This was accomplished in 1994 by the *dependent LU requester* (DLUR) architecture[10] that IBM opened to the industry via the APPN Implementers' Workshop during its development. Unlike other methods (and there are many) to support dependent LUs in an internetwork by circumventing SNA restrictions, DLUR simply lifts the restriction, solving this challenge in a straightforward and flexible manner.

The flows between dependent devices and the application host are managed in the subarea network by an entity called the SNA *subarea boundary function* (BF), often implemented in a communications controller such as the 3745 with Network Control Program (NCP) software. VTAM itself provides such a boundary function for channel-attached devices and when no front-end processor is used for communication lines. DLUR architecture provides a boundary function remote from the SSCP across an APPN network. By adding a small piece of client logic (the DLUR) to existing APPN nodes that also support dependent devices, SSCP-physical unit (PU) and SSCP-LU support are "piped" over an arbitrary number of APPN hops, from an existing SSCP to wherever they are needed. A corresponding piece of server logic (the *dependent LU server*) in the host complements the DLUR client by managing the BF-SSCP relationship.

By contrast, other techniques to support dependent LUs in a multiprotocol internetwork include:

- *Remote bridging*[11]—a means of extending transport services over wide area networks to remote nodes, which operates at the data link layer (OSI layer 2). However, remote bridging lacks alternative path capability and scalability, does not support priority, and has hop-count limitations.
- *Single-stack tunneling (encapsulation)*—for example, data link switching,[12] an extension of bridging that maps layer-2 data link connections to Transmission Control Protocol (TCP) connections. The link-layer protocols are divided into three concatenated path segments, with link-level procedures performed in each segment to avoid timeouts, but linking all three segments together to form a logical link.
- *Protocol conversion*—for example, TN3270, a TCP/IP-based 3270 terminal emulator program that was distributed with the Berkeley Software Distribution UNIX 4.3 and is supported by most TCP/IP networking vendors. While it provides 3270 access in a native TCP/IP environment, it omits some important support for 3270 functions and attachments and makes heavy demands on System/390 host processing. Many mission-critical applications that run on Multiple Virtual Storage (MVS) depend heavily on functions that TN3270 does not support.
- *APPC3270*—an IBM-developed application program that uses APPC protocols to carry the 3270 data stream to a partner APPC3270 program. Unlike TN3270, it conveys control information using native SNA headers; thus it has better performance. But like TN3270, it requires a server running on the host, and it does not support real 3270 devices.

DLUR brings added benefits to dependent devices. DLUR not only relaxes the ties of dependent LUs to mainframe hosts and preserves customer investments in existing devices, such as 3279 terminals, 3270 emulators, and 3274 control units, and their expensive-to-update program-to-device applica-

tions, but it brings added benefits over the subarea network support.

Along with DLUR comes dynamic definition for dependent LUs, easing their administration. DLUR improves availability as a result of two factors. First, APPN routing provides multiple and alternative paths between a device and an application program. Second, DLUR decouples the routes for SSCP control sessions from routes for LU-LU sessions. Consequently, a link failure disrupting a control session does not necessarily disrupt the LU-LU sessions it manages, and control sessions can be recovered dynamically over different paths, even to different SSCPs. DLUR improves performance through the ability to route across APPN/HPR paths, to have dynamically selected alternative routes, and by no longer consuming costly mainframe cycles or 3745 control block storage for pure routing functions. DLUR brings added security, since LU 6.2 authentication protocols can be used on the DLUR-DLUS "pipe." DLUR greatly improves network manageability, since protocol-neutral devices such as gateways, bridges, and routers between dependent devices and hosts, formerly invisible to SNA network managers, become visible and manageable when their bridging function is replaced by APPN DLUR support.

DLUR also offers simple migration, since it can be installed virtually anywhere in an APPN network.

The fundamental benefit is the access to APPN flexibility that DLUR confers on traditional SNA applications. Because all the cited benefits enable more cost-effective network designs, DLUR lets users adapt the network to the needs of a changing organization, rather than dictating the organizational structure around the needs of an inflexible network.

High-performance routing

High-performance routing (HPR)[13] is a recent APPN feature that enhances data routing performance and reliability, especially when high-speed links are used. HPR was introduced into the APPN Implementers' Workshop in 1993 and gained final approval in May 1995.

To support the emerging high-speed communications facilities, routing in intermediate nodes using HPR is done at a lower layer and is much faster than in the existing base-APPN intermediate session routing protocol. The HPR intermediate routing protocol minimizes both storage and processing requirements in intermediate nodes. The hop-by-hop error recovery and flow control used in base APPN for the older, slower-speed links is unnecessary for reliable high-speed links. HPR provides error recovery and flow control at the session endpoints that eliminates the need for performing these functions on each link along the session path.

To provide greater session reliability, HPR uses a nondisruptive path switch function that automatically switches session paths around failed links or nodes.

The two main components of HPR are the Rapid Transport Protocol (RTP) and automatic network routing (ANR).

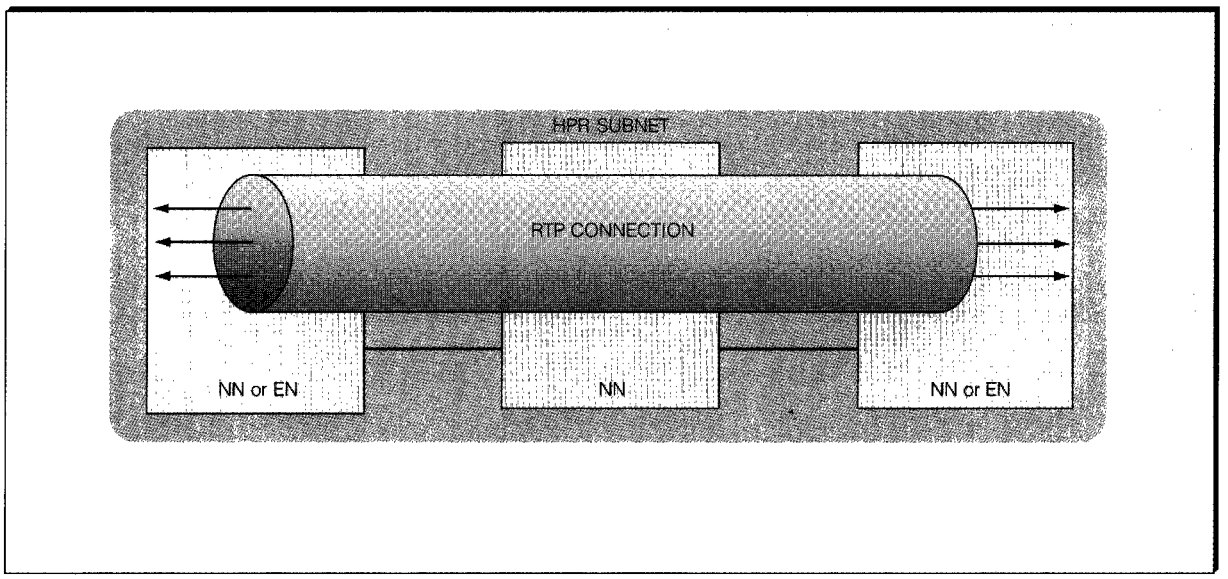
Rapid Transport Protocol. RTP is a connection-oriented, full-duplex protocol designed to support data transfer in high-speed networks. *RTP connections* are established within an HPR subnet and are used to carry session traffic. These connections can be thought of as "transport pipes" over which session traffic is carried. RTP connections can carry data at very high speeds by using low-level intermediate routing and by minimizing the overhead on individual links for error recovery and flow control.

The RTP functions include the following.

End-to-end error recovery. In base APPN, error recovery is done on every link in the network. To better serve the emerging high-speed links with their lower bit-error rates, HPR eliminates link-level error recovery and instead does error recovery only at the endpoints of the RTP connection. This improves performance by avoiding error recovery flows and processing on every link. RTP also supports *selective* retransmission, where missing or corrupted packets are re-sent, but not the good packets after the failed one as in "go-back-N" schemes.

End-to-end flow/congestion control. Flow control in base-APPN networks is done on each stage, or hop, of the session path, using adaptive session-level pacing. This method provides good performance for networks comprising a mixture of link types, with differing speeds and quality. However, for high-speed networks, adaptive session-level pacing is not desirable because of the amount of

Figure 2 RTP connection supporting APPN sessions. Multiple sessions using the same class of service can share the connection.



processing time required in each intermediate node.

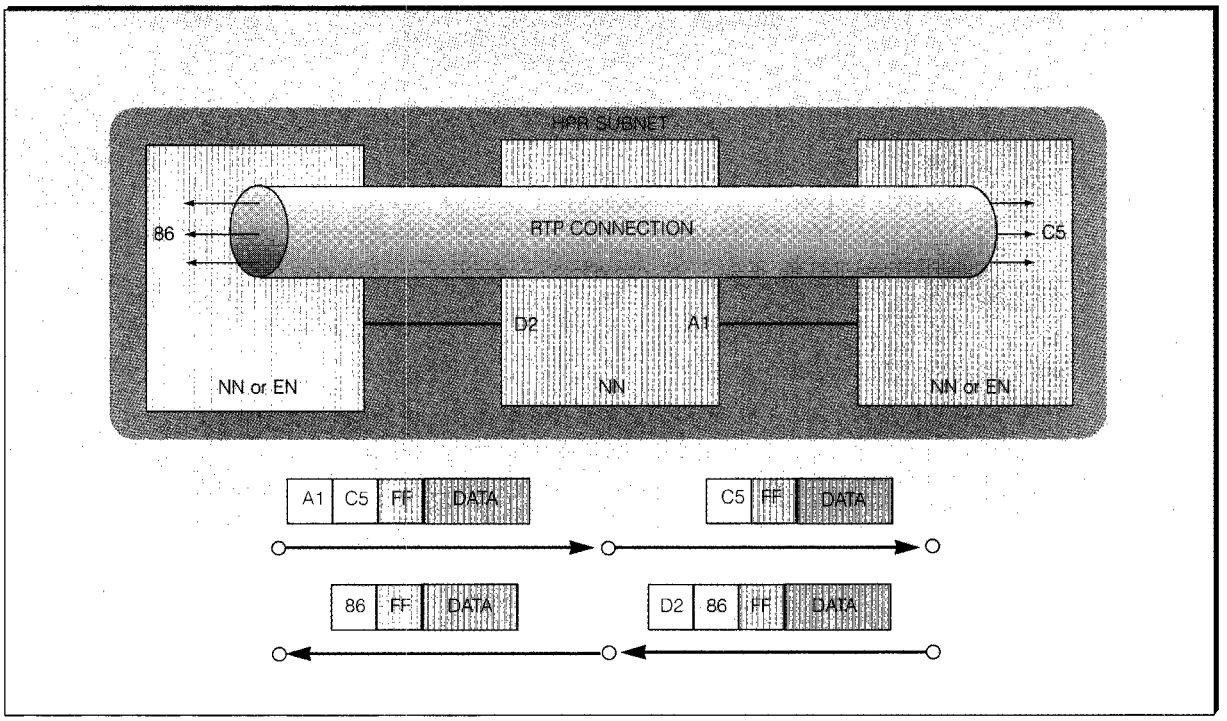
RTP provides a new protocol at the connection endpoints called *adaptive rate-based* (ARB) flow/congestion control. ARB control ensures that the receiving RTP endpoint is not flooded. It prevents congestion by constantly monitoring the amount of data flowing over the RTP connection and reducing it when necessary. ARB control also maximizes link utilization by sending data into the network in measured amounts rather than uncontrollable bursts.

Nondisruptive path switch. The physical path of an RTP connection can be switched automatically to reroute sessions around a failure in the network. The flow of data on the sessions is resumed on the RTP connection using the new path without disrupting the rerouted sessions. Any data that were in the network at the time of the failure will be recovered automatically using RTP end-to-end error recovery.

Figure 2 shows an RTP connection that is carrying multiple sessions. Traffic from many sessions requesting the same class of service can be routed over the same RTP connection. If an HPR node is an intermediate node on a session path, then it must be a network node (NN), just as in base APPN; otherwise, it can be either an end node (EN) or a network node.

Automatic network routing. ANR is a new routing protocol that minimizes storage and processing (CPU cycles) requirements for routing packets through intermediate nodes. Because ANR takes place at a lower layer than APPN intermediate session routing, packets can be switched very fast. It is expected that packets may be switched 10 times faster than in base APPN when ANR is done in software, and up to 100 times faster when done in hardware. Functions such as link-level error recovery, segmentation and reassembly, and flow control are no longer performed in the intermediate nodes but only at the RTP connection endpoints. An HPR intermediate node is not aware of SNA sessions or the RTP connections that are established through it, and therefore does not require the memory for them. This saving of intermediate storage is essential for the future, when HPR nodes supporting high-speed links will be carrying many more intermediate sessions than APPN nodes carry today. Also, the connectionless property of ANR allows a more efficient and equitable sharing of resources among APPN and other protocol stacks in multiprotocol

Figure 3 ANR routing. Intermediate nodes strip routing information from the header at every hop along the path.



routers, which can improve overall performance and reduce network costs.

Source routing. ANR is a *source-routing* protocol and carries the routing information for the entire path in a network header in the packet. Each intermediate node strips off information from this network header before forwarding the packet to the outbound link, so the next node can easily find its routing information at a fixed place in the header.

Transmission priority. The network header contains a transmission priority field that is used during intermediate ANR by HPR nodes. The transmission priority field specifies one of four values: network, high, medium, or low. The *network* priority value is reserved for control traffic such as topology database updates and directory searches. The value of the priority field for LU-LU sessions comes from the class of service selected by the LU that originated the session.

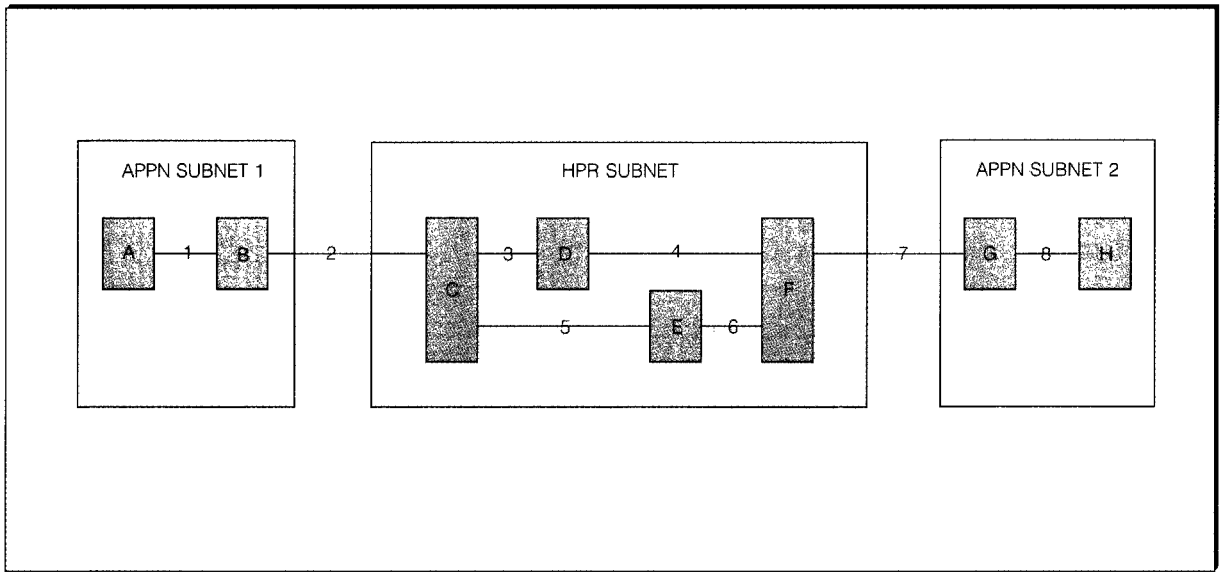
HPR nodes keep queues for each priority on every link, and therefore higher priority packets can overtake lower priority ones. Aging mechanisms ensure that lower priority packets are not permanently held in queues while only higher priority traffic is serviced.

Figure 3 shows the principle of ANR. The intermediate NN strips the first *routing label* (A1) from the network header before forwarding the packet on link A1. The address of C5 represents the endpoint in the last HPR node. With no need to reserve storage or to do link-level error recovery, the intermediate NN can route packets very quickly.

General HPR/APPN network operation. Figure 4 shows a network where an HPR subnet connects two base-APPN subnets. All nodes within the APPN subnets are base-APPN network nodes, and all nodes within the HPR subnet are HPR network nodes. The following sections describe the basic operation of this combined network.

Topology. When all the links and nodes are active, every node has the topology for the entire network, shown in Figure 4, stored in its topology database.

Figure 4 APPN/HPR network



Nodes in the APPN subnets view nodes and links in the HPR subnet as base APPN. Nodes in the HPR subnet can distinguish between the base-APPN and HPR links and nodes.

Session activation. If an LU in node A wishes to establish a session with an LU in node H, the following events occur:

- Node A initiates a directory search to locate the target LU. The directory search protocols are exactly the same as in base APPN.
- When the search completes, node A computes a route (1-2-3-4-7-8) to node H and sends the BIND over the first hop of the route (link 1). The BIND is sent using the base-APPN format.
- When node B receives the BIND, it creates a *session connector* that is used for intermediate session routing and adaptive-pacing flow control.
- Node B forwards the BIND over link 2 to node C, which contains an *APPN/HPR boundary function* (BF). The BF obtains ANR routing information for RTP path 3-4 and establishes an RTP connection to node F over path 3-4. The BF also creates a session connector that connects the base-APPN session being established with the RTP connection. The BIND is sent over the RTP connection in a *network layer packet* (NLP).[14]
- Since data sent on RTP connections are ANR-routed through intermediate nodes, node D, after stripping off the link-4 ANR label, simply forwards the NLP to node F over link 4 using the fast ANR protocols. Node D maintains no memory for ANR-routed NLPs (even when one contains a BIND).
- Node F also contains a BF. When it receives the NLP containing the BIND, it creates a session connector that connects the RTP connection and the base-APPN session that is being established over link 7. The BIND is sent over link 7 to node G using base-APPN format.
- Node G performs normal APPN intermediate processing (just as in node B) and forwards the BIND to node H, the final destination.
- Node H sends back a BIND response, which flows using base-APPN protocols over links 8, 7, 2, and 1, and HPR/NLP protocols over the RTP connection that was established over links 3 and 4.

Session traffic. After the session has been established, the session traffic flows over links 1, 2, 7, and 8 using base-APPN protocols. It flows over the RTP connection for links 3 and 4 where HPR/NLP protocols are used. Over the RTP connection, error recovery and flow control are at nodes C and

F; intermediate node D does not become involved in these protocols—it handles only ANR protocols.

Path switching. If a link fails along the HPR portion of the path, it may be possible to switch paths. For example, if link 4 fails, the path for the RTP connection will be switched from path 3-4 to path 5-6 (assuming path 5-6 satisfies the class of service associated with the sessions). Switching the path involves recalculating a new RTP path and obtaining ANR information about it. Once this is done, traffic on the RTP connection is ANR-routed over the new path 5-6 through node E. Node E does not know anything about the RTP connections or sessions, it just performs ANR protocols. Any data that might have been lost due to the link failure will be recovered by the RTP connection endpoints (in nodes C and F).

HPR migration. There are no configuration restrictions on migrating an APPN network to support HPR. As soon as an HPR subnet is formed, the benefits of HPR are achieved. New HPR nodes may be added or existing base-APPN nodes upgraded to HPR in any manner.

To take full advantage of the HPR function, however, HPR subnets need to be formed. High-speed links carrying heavy traffic, such as backbone trunks, benefit from the performance enhancements of HPR.

As soon as two adjacent base-APPN nodes migrate to HPR, they can achieve nondisruptive path switching and adaptive rate-based flow/congestion control. They can also reduce traffic flows for error recovery by using RTP selective retransmission protocol. When additional nodes migrate to HPR so that an intermediate HPR node with at least two HPR links performs ANR only, then the other benefits of HPR can be obtained: fast intermediate node routing with priority and reduction in intermediate node storage.

As HPR is an extension to APPN, it uses the base-APPN directory protocols, topology protocols, and route-selection algorithm. Because HPR nodes and links appear as base-APPN nodes and links in the topology databases of base-APPN nodes, an orderly migration to HPR that will not affect existing base-APPN nodes in the network is possible.

The desirability of HPR links is reflected in terms of their characteristics. If HPR links have higher speed than the existing links, then their route selection "weights" reflect that. HPR nodes will not automatically select HPR links in preference to base-APPN links with the same weights, because that would negate the base-APPN load-balancing feature.[15] If the selection of HPR links in preference to APPN links is required, it can be done by defining their characteristics appropriately. Such characteristics may be user-defined.

HPR insulates the upper layers from any awareness of the RTP connections and ANR protocols in the network. The LU-LU sessions will benefit from the improved performance in the network without having to make any changes to support HPR. Any existing applications supported by the LUs are supported by HPR networks. For example, dependent LU sessions may be carried over an RTP connection to an HPR node containing the partner LU.

Exploiting switched technology

A switched transmission facility allows connections to be established as needed, with connection charges incurred only while the connection is active. For that reason, switched facilities are often used for home and mobile data applications. Despite the longtime ubiquity of switched transmission facilities, in the commercial environment the predominant transmission facility for data, unlike voice, across wide area networks has been leased or otherwise privately-held telecommunication lines. Mainframe-centric networks, featuring communication between any end-user equipment and any mainframe-based application within the network, have typically been star-structured configurations with nonswitched lines radiating from a data center to end-user sites. In the current era of powerful personal computers and workstations, end-user sites need to communicate among themselves without using a predictable configuration. As a result, switched facilities are rapidly replacing nonswitched lines for data transmission. The high cost of private lines, especially in Europe, is hastening this change. Carriers are offering new switched link-layer technologies such as ATM and switched frame relay, as well as new switched services such as LAN emulation.

The following sections describe these new switched facilities as well as new APPN features that are being developed to exploit them.

Survey of switched technologies. The most common switched facilities use analog modems to access dial services on public switched telephone networks. Typical data rates offered by such modems include 14.4 and 28.8 Kbps.

Carrier offerings for narrow-band *integrated services digital network* (ISDN)[16] are becoming prevalent in Europe, especially Germany, and in Japan. ISDN provides a "basic-rate" interface of 144 Kbps to the customer premises, with "primary-rate" interfaces of 1.544 Mbps in the United States and 2.048 Mbps in Europe also available. ISDN provides both circuit-switched and packet-switched connections.

The basic-rate interface consists of two 64-Kbps *B channels* and one 16-Kbps *D channel*. The primary function of the D channel is to carry signaling information for circuit switching of the B channels, but it can also provide access to an X.25 packet-handling service. In circuit-switching mode, an entire B channel is switched to a single destination.

A *frame-relay*[16] network, via a standard user-to-network interface (UNI), routes variable-length frames, typically over carrier-provided subnets through which many remote stations can be accessed in a point-to-point fashion. Frame-relay networks provide a connection-oriented service using link-layer addressing. Different user data streams may be multiplexed at a physical port over distinct logical connections. The core services provided by a frame-relay network are frame delimiting, congestion notification, and error checking. Frame-relay terminal equipment often requires only a software upgrade to existing equipment, and as a result is already widely deployed.

Carriers currently have primarily permanent virtual circuit frame-relay offerings. Only a few carriers currently offer a switched frame-relay service, but other carriers are sure to follow.[17] Frame-relay signaling is defined in International Telecommunication Union (ITU) recommendation Q.933. For data transfer, switched frame relay provides guaranteed throughput and is a good first step into the switched environment.

An *ATM bearer service*[16] provides a sequence-preserving, connection-oriented, cell-transfer service between source and destination with a guaranteed quality of service (QOS) and throughput. Virtual connections can be either point-to-point bidirectional or point-to-multipoint unidirectional. An ATM bearer service also supports signaling protocols if it provides a switched virtual circuit (SVC)

IBM's Networking BroadBand Services architecture and Nways products provide frame relay and ATM bearer services.

connection service. Virtual connections are established and released via the signaling protocol or by subscription.

ATM networks are designed to transport a variety of traffic classes satisfying a range of transfer capacity needs and network performance objectives. The higher-layer user of ATM indicates its throughput requirements for an SVC by specifying a subset of the following connection traffic parameters at the UNI defined by the ATM Forum: peak cell rate, sustainable cell rate (i.e., the maximum average rate), and maximum burst size at the peak cell rate.

A traffic contract specifies the negotiated characteristics in one direction of flow on a virtual connection at the UNI. A traffic contract consists of a requested QOS class and a set of connection traffic parameters. A QOS class specifies a set of ATM performance parameters such as cell loss ratio, mean cell-transfer delay, and cell delay variation. The ATM subnet commits to meet the requested QOS as long as the user complies with the traffic contract.

IBM's Networking BroadBand Services architecture[18] and Nways products provide frame relay and ATM bearer services.

LAN emulation. *LAN emulation* enables higher layer protocols (such as APPN and TCP/IP) to access ATM subnets as if they were running over a *legacy* LAN (e.g., IEEE 802.3 and IEEE 802.5). Thus, LAN emulation allows a higher layer protocol to have access to ATM subnetworks with no changes required for the higher layer protocol. When an APPN

node implements LAN emulation, its data link control (DLC) function must have LAN-emulation software. LAN emulation requires one node on the ATM subnet to act as a LAN-emulation server responsible for mapping legacy LAN addresses to ATM addresses.

LAN emulation hides the underlying switched subnet from higher layer protocols; as a result, higher layer protocols cannot use some features, such as guaranteed throughput and QOS, of the ATM subnet. Therefore, a higher layer protocol needs a "native" interface to the ATM subnet to exploit all its features.

APPN currently has the capability to define a connection network on legacy LANs. With LAN emulation, APPN views an ATM network as a LAN; therefore, LAN emulation allows a connection network to be defined on an ATM network.

Connection network model. APPN links are logical, point-to-point connections between a pair of adjacent APPN nodes. A node requires a system definition for each such link. When N APPN nodes can communicate with each other across a shared transport facility such as a LAN, the required number of APPN link definitions (which includes link-level signaling information required for routing to partner nodes) would be $N \times N$. For that reason, APPN uses a *connection network* model to reduce the system definition to essentially a $1 \times N$ problem.

In the connection network model, a *virtual routing node* (VRN) represents the shared facility. Each node attached to the facility defines a single link to the VRN rather than links to all other attached nodes.[19]

Both APPN end nodes and network nodes make use of this model. Nodes of both types know their own link-level attachment information, which is needed for DLC signaling by partner nodes when the links between them are to be activated. Nodes differ in how they present this information to the network. End nodes return information about their VRN connections and related DLC signaling information as part of the normal directory services process to find a destination node. Network nodes store all their VRN connection and allied DLC signaling information in the distributed topology database that all network nodes maintain and access. Thus, a network node calculating the appropriate route for a requested session between two nodes can not only determine that the paired nodes share the same connection network, but can also provide the needed DLC signaling information to the node that will initiate the session over a selected route using that facility.

Extensions to the LAN connection network model are required to exploit the features of switched facilities. For LANs, the signaling information, which consists of the medium access control (MAC) address and the logical link control (LLC) service access point (SAP) address, is sufficient to establish a connection; however, additional signaling information is required for switched facilities. For example, the signaling information for an ATM connection network includes the bandwidth and QOS requirements. The LAN connection network model assumes the same characteristics for each connection crossing the LAN; however, parameters for switched connections may vary call by call.

Multiple connection networks may be defined on a shared facility. Normally, a single connection network is defined on a LAN, but separate connection networks may be defined on switched subnets for local- and wide-area connections. The connection network model for LANs allows only one link to be defined between a physical port and a VRN. The definition of multiple links between a port and the VRN for a switched subnet allows the establishment of multiple switched connections between nodes, with each connection having its own parameters; in addition, multiple connections can be used for separation of traffic for different classes of service.

Use by APPN of the connection network model compares favorably with mechanisms for transport of Internet Protocol (IP) traffic over ATM networks defined by Internet Engineering Task Force (IETF) RFC 1577. In the LAN shared-transport environment, IP uses the Address Resolution Protocol (ARP) and the Inverse Address Resolution Protocol to establish connectivity between IP end systems (hosts or routers). RFC 1577 describes the use of an ARP server to duplicate these functions in the ATM environment. One or more logical IP subnets (LISs) is defined on an ATM network: within an LIS, all IP hosts and routers share the same IP network number, subnet number, and address mask. The IP hosts and routers of an LIS register their IP addresses and ATM addresses with the server and use the server to discover the addresses of other IP hosts and routers. This approach precludes direct

ATM connections between IP end systems in different LISs; thus, two ATM connections are required with an intermediate IP router. By contrast, direct ATM connections are allowed between APPN end nodes with different network node servers. Also, for each LIS, RFC 1577 specifies a single ARP server, which is a potential single point of failure; APPN allows an end node to define alternative network node servers.

Connection network link definition. APPN transmission priority and class of service allow APPN links to be highly utilized for batch traffic without impairing delay-sensitive interactive traffic.[20] Aging algorithms guarantee that lower priority traffic is not completely blocked. The result is predictable performance: interactive traffic obtains its required response time while sharing the transport with batch applications; the transport service can discriminate among batch applications while ensuring that the lowest priority traffic has adequate service.

When selecting a route for a requested session, APPN uses the associated class of service of the session to choose the most appropriate sequence of links (based on their characteristics) for the session to traverse. The connection network model allows multiple, parallel links—each with its own distinct characteristics, throughput parameters, and QOS class—to be defined to a VRN. APPN topology and routing services select the appropriate link for the traffic of each session; as a result, a switched connection with appropriate throughput parameters and QOS class is used. In using the APPN support, the customer's objectives in designing the number of such links and their characteristics, throughput parameters, and QOS classes may include:

- No performance regression: continued good response time for interactive data traffic without blocking lower priority traffic
- Fairness: unbiased service for the multiple APPN traffic streams of a given priority traversing a switched network
- Efficiency: high utilization of the links within the switched network and good resulting cost performance

Tariff structures for switched services may vary over time, as well as by technology and carrier. Thus, the APPN enhancements for switched technology are designed to be robust enough to provide line cost savings in dissimilar environments. The enhancements allow customers to have different strategies in defining link characteristics and throughput parameters to improve performance or reduce line costs.

One customer strategy is to define a link to a VRN from each node attached to the switched subnetwork. A single connection is established as needed between pairs of nodes attached to the connection network. If such a connection has traffic throughput guarantees, it has the appearance of an APPN link of fixed bandwidth. Alternatively, a best-effort connection has lower associated line costs but degraded response time for interactive traffic during periods of network congestion.

Another strategy is to define links to two VRNs from each node; this allows the establishment of two connections between nodes attached to a switched subnetwork. For interactive and network control traffic, where response time is important, a connection with a throughput guarantee would be established; a best-effort connection would be used for batch traffic.

APPN network management. While APPN has been evolving, so has its systems and network management. Originally, APPN networks were managed solely by the IBM-defined *SNA management services* (SNA/MS).[21] To meet customer needs for unified management of multiprotocol networks using open techniques, APPN management has been brought under both open management architectures currently in the marketplace: the Simple Network Management Protocol (SNMP) and the Common Management Information Protocol (CMIP).[22] While retaining the proven SNA/MS support for areas such as problem management[23] and change management,[24] IBM has focused on using the open architectures for management disciplines not previously covered for APPN resources: topology management, accounting, and performance management.

As a consequence of covering so much of the management landscape for APPN resources, IBM management functions are spread out over a number of platforms and realized in a number of distinct products. Work is under way to unify all of these functions within IBM SystemView*. By encapsulating existing SNA/MS, SNMP, and CMIP management inside objects that represent management applications, SystemView will hide the different

management protocols and provide a single interface to a management application program needing any of the functions that these protocols provide.

The open management protocols also make it possible for APPN networks and resources to be managed from locations outside the APPN networks themselves. CMIP or SNMP traffic may, for example, use IP as its transport between a centralized manager on an IP host and a managed system that is both an IP host and a node in an APPN network. As described below, mechanisms for transport of SNMP and CMIP traffic over SNA sessions are provided, so that a foreign transport such as IP is not *required* for managing APPN resources.

Management models for APPN resources. The CMIP managed-object model for APPN topology takes advantage of the fact that the APPN *network-node* topology database is fully replicated in all the network nodes in a topology subnet: a CMIP *manager* can retrieve the network-node topology of an entire subnet by interacting with only one of its network nodes. Since *local* topology information (regarding end nodes and the links attached to them) is not replicated, a manager must interact directly with the network node or end node whose local topology it desires to see. A typical mode of operation for a CMIP manager is to monitor network-node topology continuously, using the results to maintain an accurate map of current network-node topology, but to request local topology only when an operator asks for it, perhaps by selecting a representation (icon) for an APPN node on a graphical user interface.

The managed-object model for APPN topology is not limited to monitoring. CMIP actions are defined that allow an operator to activate and deactivate both physical adapter ports and links.

A second CMIP managed-object model supports the gathering of accounting data for APPC (LU 6.2) sessions and conversations. Information about a conversation is captured by a CMIP *agent* at the session endpoints, while session information can be gathered by an agent either at the session endpoints or at intermediate nodes in the session path.[25] The accounting data captured include the following:

- For sessions—information about the session, its endpoint LUs, and its route; session start and end times; and various session traffic counters
- For conversations—information about the conversation and its underlying session; information about the LUs and the target transaction program; conversation start and end times; and counts of the request/response units sent on the conversation in each direction.

Accounting information is saved, at the node that captures it, for retrieval later by the manager. An agent may notify a manager that its stored accounting data are approaching the capacity of the agent, so that the manager can retrieve the data before any are lost. Agents receive explicit acknowledgments when a manager receives a batch of accounting records, freeing them to erase the records from their own local storage.

The SNMP model for APPN management also provides for APPN topology management and APPC accounting. In addition it includes managed objects that support certain types of performance management for APPN networks. As in the case of the CMIP model, an SNMP manager wishing to monitor the network-node topology of an APPN topology subnet needs to interact with an SNMP agent in only one of the network nodes in the subnet.

A feature of the model is its use of the *flow-reduction sequence numbers* (FRSNs), which are carried on the APPN topology flows, as a means of reducing the overhead associated with SNMP polling. (A network node increments its FRSN each time it sends out updated topology information. The current FRSN is carried on the message transporting the updates and stored in all the records in the node topology database that were included in the particular update.) Since SNMP has only limited support for asynchronous notifications, an SNMP manager maintaining a topology map of network nodes in an APPN subnet must *poll* the SNMP agent in one of the network nodes in order to detect topology changes. Traditionally, when SNMP managers poll, they retrieve all the information present at an agent and use it to reconstruct their view of the current situation. This approach would have been very costly for the volumes of data associated with the topology of an APPN subnet.

The SNMP model for APPN has eliminated most of this polling by using the FRSN as the primary index into APPN network-node topology tables maintained at the agents. An agent has two tables for network node topology: one has information about network nodes, the other information about the

links connecting network nodes. Each agent derives these tables from the topology database in its own node. Using the FRSN as the index for ordering the tables has the effect of moving new topology information to the end of the tables, since new information always has a greater FRSN value than old information. A manager simply remembers the FRSN for the last row it has retrieved from each table, and then asks for the row after the one indexed by that FRSN. In this way, it can get all the updated information without ever having to retrieve rows remaining unchanged.

To understand better the benefits of FRSN-based polling, consider a hypothetical topology subnet with N network nodes and L links connecting them. (A representative APPN configuration might have 100 network nodes and 400 links in a topology subnet.) There are three Management Information Base (MIB) designs, with three corresponding polling strategies, that we contrast here:

- *No optimization:* No attempt is made to reduce the polling overhead. At each polling interval, an SNMP manager must direct $N + L + 2$ SNMP operations to the agent: N operations to get information on all the network nodes, one more operation to determine that the last network node has been covered, and similarly for the links.[26]
- *Last-update optimization:* This technique has been used for some SNMP MIBs. The idea is to include with a table a separate MIB variable indicating when the table was last updated: either a timestamp or a sequence number will do. At each polling interval, an SNMP manager needs to retrieve only the two MIB variables for the network node and link tables, since they will indicate whether there is new information in either of them. If either table has been updated, the manager must retrieve the entire table to get an accurate picture of the new topology. Otherwise, it simply waits until its next poll. Numerically, a manager will need to send two SNMP operations when neither table has changed, $N + 2$ or $L + 2$ when one has changed, and $N + L + 4$ when both have changed.
- *FRSN-based polling:* In this case, a manager needs to send two SNMP operations to detect that the network topology has not changed, just as with the last-update optimization. When the topology has changed, however, the manager needs to send only $\Delta N + 2$ operations (if only the network node table has changed), $\Delta L + 2$ operations (if only the link table has changed), or $\Delta N + \Delta L + 2$ operations (if both have changed), where the "Δ" notation represents just the number of changed entries in the corresponding table. For numbers in the range of $N = 100$ and $L = 400$, this can represent a considerable savings over the last-update approach.

Transport of SNMP and CMIP data in APPN networks. Obviously, it would be undesirable if, to manage APPN resources with SNMP or CMIP, a customer had to install an actual IP or OSI network alongside the APPN network to be managed. Thus, CMIP and SNMP flows between a manager and an agent on an APPN node use the APPN network itself as their transport vehicle. CMIP does this using the *multiple-domain support transport* structure already in place for SNA/MS.[27]

APPN will continue to evolve in the light of new technologies and requirements.

The initial design for SNMP-based management of APPN resources had the SNMP traffic flowing over its native transport: User Datagram Protocol (UDP). This was appropriate for the initial target product environment—that of the 6611 Network Processor—in which APPN was simply one protocol stack among several in an SNMP-managed system. As SNMP capabilities were extended into other APPN platforms, however, a native transport for SNMP was needed. We considered a solution based on the AnyNet sockets-over-SNA mapping,[28] since this would allow existing IP-based SNMP managers and agents to communicate across an APPN network. This solution was unsatisfactory, however, because while it eliminated the need for an actual IP network for the SNMP flows, it still required the definition of a virtual IP network, with each managed APPN node requiring an IP address.

We chose instead *native* transport of SNMP data over APPC sessions. To minimize the impact on existing SNMP applications, this architecture does not

use most of the capabilities available with APPC: it simply uses APPC conversations as a way to emulate the services provided by UDP datagrams, the native transport of SNMP.[29] Other industry transports, such as IPX, NetBIOS, and Appletalk, incorporate similar schemes for transport of SNMP data. The architecture is also sufficiently general to handle both SNMP version 1 and SNMP version 2.

Management of APPN extensions. Each of the APPN extensions discussed earlier in this article has brought with it additional management requirements. In most cases these requirements include new, product-independent, generic alerts for reporting new types of problems, such as loss of communication between a dependent LU requester and its server. Also common are updates to the configuration models, so that a manager can, for example, distinguish an HPR-capable node from a base-APPN node, and thus represent them graphically to an operator by different icons.

Future considerations

APPN will continue to evolve in the light of new technologies and requirements. We expect a number of important extensions as a consequence of the emergence of new switched services, particularly ATM.

Bandwidth reduction strategies. The key challenge in using switched services is to minimize their cost by exploiting their intermittent nature, and to do this flexibly in order to adapt to new and perhaps unforeseeable tariff structures. To minimize switched circuit costs, new APPN support could include support for three new functions: dynamic bandwidth modification, short-hold mode, and dynamic modification via subnet-provided adaptation.

Dynamic bandwidth modification. The strategy here is to reserve only the amount of bandwidth actually needed, provided that additional bandwidth can be added and released on demand using dynamic "triggers." Triggers would include operator-initiated modification of switched connection bandwidth and modification based on algorithms to determine when more or less bandwidth is needed. These techniques require the switched facility to support bandwidth modification of active connections. Future ATM standards, and products based on Networking BroadBand Services (NBBS), could allow APPN to dynamically signal changes in its required bandwidth to the subnet. The link capacity and cost would grow or shrink based on the changing needs of APPN.

Short-hold mode. This is a method to disconnect and reconnect a switched connection transparently without the knowledge of the higher layers, which would regard the circuit as being continuously active. This method is useful when tariffs are biased toward connect time.

APPN already includes the ability to disconnect links when there are no active sessions. However, sessions may be long-lived and may remain active even when there are no data being sent. In particular, the APPN requirement for persistent CP-CP sessions (for control traffic) is costly in environments where switched virtual circuits (SVCs) are prevalent. Therefore, it is desirable for APPN to allow CP-CP session traffic to cross SVCs without requiring that the SVCs be kept active for the duration of the CP-CP session. Using short-hold mode, the CP-CP session between two APPN nodes could remain active, while the SVC over which the session runs is disconnected during lulls in the APPN control traffic. The SVC could be reestablished whenever the control traffic resumes. In this way, tariff charges for control traffic could be significantly reduced. The interruption of the SVC would not be detected by the CP-CP session endpoints but simply appear as a delay in resuming data transmission.

Dynamic modification via subnet-provided adaptation. NBBS networks allow users to specify minimum and maximum acceptable bandwidth parameters. The network then monitors the offered load and modifies the reserved bandwidth, within the limits, to meet the needs of the user. Providing this interface to NBBS would enhance APPN switched support in an important way.

Real-time transport. ATM facilities provide the *real-time* data transmission function required for applications such as multimedia. Real-time data transfer requires data transmission rates (or *bandwidth*) equal to the source generation rate, and guaranteed values for mean cell-transfer delay and cell delay variation. APPN enhancements would be desirable to exploit this function. Traffic requiring real-time service would be transported only between APPN nodes attached to ATM subnetworks, or to other subnetworks providing such services. (More extensive APPN enhancements would be re-

quired to transport such traffic across subnetworks that do not provide these services.)

The interface of the APPN node to the LU would need to allow a high-function LU to provide throughput and QOS parameters specified by the application. A new "demand" type link to the VRN of a connection network would be needed. If a demand link were selected by topology and routing services, a dedicated connection would be established using the application-specified parameters for the session. Thus, the session traffic would receive guaranteed throughput and QOS in the switched subnetwork. An APPN node would also need to minimize the delay through the APPN stack (perhaps by implementing in an operating environment with guaranteed CPU and buffer utilization).

Address resolution. Switched subnetworks are expected to offer *address-resolution* services. Such services would map higher layer addresses to switched subnet addresses. For example, an IP address-resolution service of an ATM subnet would map IP addresses to ATM addresses.

Address-resolution services could be exploited by APPN to relieve the customer from manually defining the switched subnet addresses of network nodes to which control connections are desired. Network nodes attached to the subnetwork would register their subnet addresses with the network service. Any end node or network node wishing to connect to a network node would request the service to map the CP name of the desired network node to the associated subnet address.

Internetwork routing. As APPN networks grow and are connected to each other using border nodes, there will be an increased need to be able to select session routes that are globally optimal. Today, border nodes constrain routes to be only "piecewise optimal" within a topology subnet. To find globally optimal routes, internetwork topology might be abstracted and shared among nodes in different topology subnets.

Conclusions

We have reviewed the motivations for APPN and discussed how the requirements have evolved since its introduction in the mid-1980s. We have described some changes and enhancements that allow APPN to meet these new challenges and continue to be an important and viable networking technology. The architecture is versatile and flexible, and it is constantly evolving to meet new requirements.

The way APPN is developed is also going through a transition. Although it was originally developed by IBM, new development is coming more and more from industry cooperation in the APPN Implementers' Workshop. The goals of this group are to enable the development and availability of APPN on a wide variety of platforms from many vendors. Its members include many prominent vendors of networking equipment and software.

As customers install more APPN networks and take advantage of the advances described in this paper, the importance of APPN will continue to grow. Along with increased investment and development of features by many networking vendors, this assures a healthy future for the APPN architecture.

Acknowledgments

Of course we are deeply indebted to all the people who invented, developed, implemented, and delivered the advances we have described. The sheer number of people from many divisions of IBM and from other companies prevents our attempting to list them all here.

We would like to thank the referees of this paper for their comments and suggestions that helped us to focus on the most important points. We would also like to thank our colleagues, especially Gary Schultz, for their careful review and help in organizing the material so that it might be useful to readers regardless of their prior knowledge of APPN.

Cited references and notes

1. *SNA Format and Protocol Reference: Architectural Logic*, SC30-3112, IBM Corporation (1980); available through IBM branch offices.
2. P. E. Green, R. J. Chappuis, J. D. Fisher, P. S. Frosch, and C. E. Wood, "A Perspective on Advanced Peer-to-Peer Networking," *IBM Systems Journal* **26**, No. 4, 414–428 (1987).
3. R. A. Sultan, P. Kermani, G. A. Grover, T. P. Barzilai, and A. E. Baratz, "Implementing System/36 Advanced Peer-to-Peer Networking," *IBM Systems Journal* **26**, No. 4, 429–452 (1987).

4. A. E. Baratz et al., "SNA Networks of Small Systems," *IEEE Journal on Selected Areas in Communications* **SAC-3**, No. 3, 416–426 (May 1985).
5. J. P. Gray and M. L. Peters, "A Preview of APPN High Performance Routing," *Local Area Network Interconnection*, Plenum Press, NY (1993).
6. Since the topology database is fully replicated only within a topology subnet, a central directory may be used only within its native topology subnet.
7. E.g., the device-oriented 3270 data stream generated by display terminals and printers such as SNA logical units type 1, 2, and 3. These LU types exemplify a class of older SNA end systems called "dependent LUs" that also includes LU type 0 and a dependent version of LU 6.2. Their "dependency" is on the session services provided by the mainframe-based SSCP.
8. M. Peters et al., "APPN and Extensions: The New Industry Standard for SNA Internetworking," *Business Communications Review*, Supplement, 20–27 (February 1994).
9. In SNA's early years, physical and logical adjacency were equivalent. Over time, PC LAN technology was extended by products such as LAN bridges that interconnect devices at the data link layer (OSI layer 2). Bridges and related layer-2 techniques are "invisible" to SNA path control as well as to the APPN routing layer. An SNA session carried over a layer-2 technology is logically, but not physically, adjacent.
10. *APPN Dependent LU Requester Reference*, SV40-1010, IBM Corporation (1994); available via ftp from networking.raleigh.ibm.com/pub/standards/aiw/dlur and through IBM branch offices.
11. K. J. Christensen, L. C. Haas, F. E. Noel, and N. C. Strole, "Local Area Networks—Evolving from Shared to Switched Access," *IBM Systems Journal* **34**, No. 3, 347 (1995, this issue).
12. P. W. Gayek, "Data Link Switching: Present and future," *IBM Systems Journal* **34**, No. 3, 409 (1995, this issue).
13. *APPN High-Performance Routing Reference* (draft), available via ftp from networking.raleigh.ibm.com/pub/standards/aiw/appn/hpr.
14. All traffic flowing over RTP connections is carried in an NLP. An NLP contains headers for both ANR and RTP protocols.
15. Base APPN randomly chooses among routes of equal weight to distribute traffic evenly over them.
16. R. J. Cypser, *Communications for Cooperating Systems: OSI, SNA, and TCP/IP*, Addison-Wesley Publishing Co., Reading, MA (1991).
17. *Netwatcher* **12.6**, CIMI Corporation (June 1994).
18. P. F. Chimento, J. E. Drake, L. Gun, E. A. Hervatic, C. P. Immanuel, G. A. Marin, R. O. Onvural, S. A. Owen, and T. E. Tedijanto, "Broadband Network Services for High-Speed Multimedia Networks," unpublished paper, available on request from appn@vnet.ibm.com.
19. Other *regular* links may be defined for the shared facility that are not part of the connection network.
20. APPN networks typically use high priority for interactive traffic and low priority for batch traffic. Networks may vary in using medium priority for less important interactive traffic or for more important batch traffic.
21. M. O. Allen and S. L. Benedict, "SNA Management Services Architecture for APPN Networks," *IBM Systems Journal* **31**, No. 2, 336–352 (1992).
22. The SNMP standards are a product of the Internet Engineering Task Force (IETF). The CMIP standards are a product of a joint committee of the International Telecommunication Union (ITU) on the one hand, and the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) on the other.
23. R. E. Moore, "Utilizing the SNA Alert in the Management of Multivendor Networks," *IBM Systems Journal* **27**, No. 1, 15–31 (1988).
24. C. P. Ballard, L. Farfara, and B. J. Heldke, "Managing Changes in SNA Networks," *IBM Systems Journal* **28**, No. 2, 260–273 (1989).
25. Obviously, capturing session accounting information at an intermediate node requires that the node have knowledge of the session. Thus, intermediate nodes in an HPR RTP connection are not able to capture these data.
26. In SNMP, one operation, involving one exchange of line flows, is needed for each table entry. An additional operation detects that the table has been exhausted.
27. For details on how CMIP data are packaged for transport by multiple-domain support, see *SystemView Mapping of OSI Upper Layers to MDS for MIP over SNA*, SC31-7137, IBM Corporation (1994); available through IBM branch offices.
28. D. Pozefsky, R. Turner, A. K. Edwards, S. Sarkar, J. Mathew, G. Bollella, K. Tracey, D. Poirier, J. Fetvedt, W. S. Hobgood, W. A. Doeringer, and D. Dykeman, "Multiprotocol Transport Networking: Eliminating Application Dependencies on Communications Protocols," *IBM Systems Journal* **34**, No. 3, 472 (1995, this issue).
29. We *could* have used more of the capabilities of APPC in the architecture. For example, SNMP requests could have been confirmed, or the conversation for a request could have been held open for its response. Such choices, however, would have complicated the design of the SMNP applications using the architecture by introducing a whole new category of error cases to be handled. For example, if responses flowed on the same conversations as their requests, a managing application would be responsible for determining that a response to one of its requests was never going to come, so that it could take steps to release the conversation.

General references

J. Nilausen, *APPN Networks*, John Wiley & Sons Ltd., England (1994).

SNA Advanced Peer-to-Peer Architecture Reference, SC30-3422, IBM Corporation (1993); available through IBM branch offices.

SNA Management Services Reference, SC30-3346, IBM Corporation (1993); available through IBM branch offices.

Accepted for publication March 15, 1995.

Ray Bird *IBM Networking Hardware Division, 800 Park Offices Drive, Research Triangle Park, North Carolina 27709 (electronic mail: raybird@vnet.ibm.com).* Mr. Bird is a senior engineer in the Networking Architecture group at IBM, RTP, North Carolina. He joined IBM in 1967 after obtaining a B.S. in mathematics from Rutgers University. From 1967 to 1977 he worked on the OS/360 operating system and the Telecommunications Access Method (TCAM). From 1977 to the present he has worked on Systems Network Architecture protocols in-

cluding subarea, LU 6.2, and Advanced Peer-to-Peer Networking (APPN). He is currently working on future enhancements to APPN.

Chuck Brotman *IBM Networking Hardware Division, 800 Park Offices Drive, Research Triangle Park, North Carolina 27709 (electronic mail: brotman@vnet.ibm.com).* Mr. Brotman is a senior engineer in the APPN Architecture group at IBM. In addition to working on APPN architecture and strategy, he currently serves as the Chair of the APPN Implementers' Workshop (AIW). Before coming to RTP he worked on advanced processor technologies, architectures, and designs. He holds two patents for highly parallel database machines and has worked on hardware and software architectures for artificial intelligence and logic programming. He joined IBM in 1977 to work on System/370™ I/O Channel development. He holds an M.S. in electrical engineering from Syracuse University and a B.S. in computer systems engineering from Rensselaer Polytechnic Institute.

Ralph Case *IBM Networking Hardware Division, 800 Park Offices Drive, Research Triangle Park, North Carolina 27709 (electronic mail: rcase@vnet.ibm.com).* Mr. Case is the manager of APPN and APPC architecture at IBM. He has contributed to several recent enhancements to APPN and chairs the APPN Reference Special Interest Group at the AIW. Previously, he helped develop ES/9000™ mainframes, specializing in channel subsystems, ESCON, and total systems test. He has experience building automated tools and complex cooperative processing systems using communication protocols, knowledge-based systems, and object-oriented programming. He holds a B.S. in electrical engineering from Rensselaer Polytechnic Institute.

Gary Dudley *IBM Networking Hardware Division, 800 Park Offices Drive, Research Triangle Park, North Carolina 27709 (electronic mail: gdudley@vnet.ibm.com).* Mr. Dudley joined IBM in 1973. He worked in Store Systems until transferring to Networking Architecture in 1985. He is currently working on enhancements to allow APPN to exploit high-speed switched networks. Previously, he was a key contributor to the Networking BroadBand Services architecture, specializing in transport protocols and transfer mechanisms. He holds two patents and chairs the ATM Birds of a Feather at the AIW. He holds a B.S.E. degree from Duke University and an M.S. degree from North Carolina State University, both in electrical engineering.

Robert E. Moore *IBM Networking Hardware Division, 800 Park Offices Drive, Research Triangle Park, North Carolina 27709 (electronic mail: remoore@ralvm6.vnet.ibm.com).* Dr. Moore has worked in IBM Networking Architecture since he joined IBM in 1983. He was involved with the definition of several functions in SNA Management Services, most notably as the architect for the generic alert. He has also participated in the development of a number of ensembles for X.700 management, both within IBM and in the ISO/IEC standards. His current responsibilities include network management for APPN and its extensions. He holds a B.A. in mathematics and philosophy from Rice University, an M.A. and Ph.D. in philosophy from Duke University, and an M.S. in computer science from the University of Houston.

Marcia Peters *IBM Networking Hardware Division, 800 Park Offices Drive, Research Triangle Park, North Carolina 27709 (electronic mail: Marcia_Peters@vnet.ibm.com).* Ms. Peters is a senior architect at IBM Networking Architecture in Research Triangle Park, North Carolina. She leads a group within the Advanced Peer-to-Peer Networking architecture team that is developing APPN extensions for ATM and other switched subnets. She chairs the AIW High Performance Routing Special Interest Group. She was a key contributor to Networking BroadBand Services, an IBM architecture for ATM-based networks. Before joining IBM in 1988, she developed plug-compatible SNA networking products. She received a B.A. in music from Swarthmore College in 1975. A senior member of the IEEE, she has authored numerous patents and technical articles.

Reprint Order No. G321-5576.