SNA: Current requirements and direction

by R. J. Sundstrom
J. B. Staton III
G. D. Schultz
M. L. Hess
G. A. Deaton, Jr.
L. J. Cole
R. M. Amy

Since its announcement in 1974, Systems Network Architecture (SNA) has evolved in terms of its functional content, configurational flexibility, and network management services. This paper briefly traces this progress to the present and examines the more recent advances in greater detail. It then discusses known requirements for enhanced application and transaction services, for additional provisions for very large networks, for continuing adaptation of small-system and transmission media advances, for inclusion of additional management capabilities, and for further integration of network standards—all of which will shape future SNA developments.

A dozen years have passed since IBM introduced Systems Network Architecture (SNA) as the blueprint for the design of its communication products and for interconnecting them within networks. Introduced to meet an evident need for a long-term strategy for interconnecting disparate products, SNA has performed this role with notable success. This success can be measured by various indices: first, by the ever-growing number of SNA products of various manufacture and their enthusiastic acceptance in the marketplace; second, by the recognition in the industry of SNA as a pacesetter for functional richness and completeness; and third, by the facility of the architecture to meet new requirements and adapt to new technologies in an evolutionary fashion. An earlier paper[1] traced the early years in a comprehensive manner; here, after several years of additional developments, we again provide a checkpoint on the status of SNA and look at new requirements that will likely shape its future.

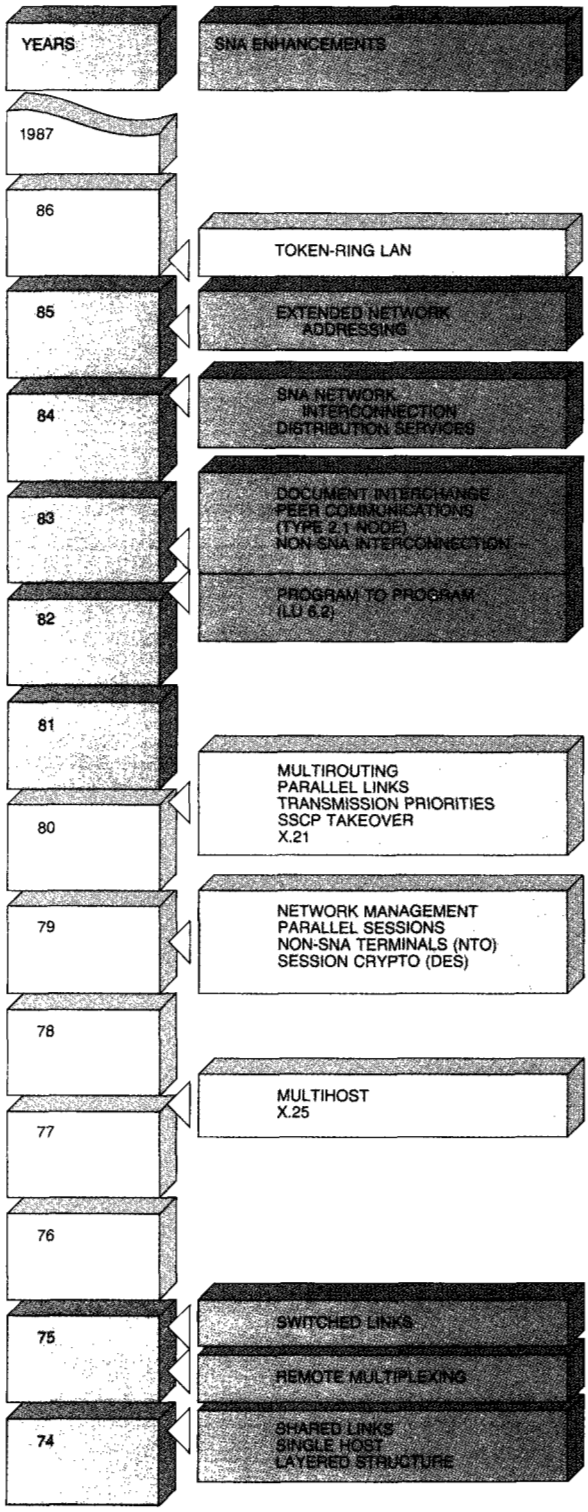
In the next section, we review briefly the highlights of the evolution of SNA since its introduction. Then, in the ensuing sections, we discuss the recent advances in more detail, according to function. In each case, we examine trends and speculate on the future on the basis of new requirements for SNA.

Evolution of SNA—Highlights and trends

In 1974, SNA began as a simple tree-oriented, single-host network. Today, it has evolved to support multiple, independent, mesh-configured networks separately administered but interconnected by gateways into composite networks. End users can communicate freely with application programs anywhere in these composite networks without being aware of the network configurations. Some of the major highlights in the progress of SNA are shown in Figure 1.

When SNA was introduced in 1974, it was supported by a single operating system (Disk Operating System/Virtual Storage, or DOS/VS) running on a single host connected to terminals through a front-end communication controller. At that time, the only SNA terminals available were on the IBM 3600 Finance Communication System used in banking. Host programs communicated with the 3600 controller.

Figure 1 SNA evolution



Within a year, SNA coverage was expanded to include remote communication controllers, the Multiple Virtual Storage (MVS) environment, and supermarket and retail point-of-sale controllers and their terminals. Still, SNA remained a nonswitched-line system only—a situation that changed in 1976 with the addition of switched-line capability.

In 1977, with the release of IBM's Advanced Communication Function (ACF) products, more general networking function became available. SNA allowed multiple host "trees" to be interconnected, using single links to connect front-end communication controllers. This capability opened the way for a terminal to access application programs in multiple-host processors. Also in 1977, IBM introduced its initial support for the X.25 standard for public packet-switched networks. We discuss this in greater detail in a later section.

With the basic networking in place, it was time for more emphasis to be placed on incorporation of network management services to monitor and control the network. This focus, which began in 1979, continues to the present. The need for such services increases according to the complexity of network configurations and to the criticality of network reliability, availability, and serviceability to the users of the network. With businesses increasingly relying on their network operations and with the scope of their applications widening, network management services have taken on greater importance as the architecture and implementing product set have been extended.

In 1979, IBM also enhanced session capabilities to allow parallel sessions between two application subsystems such as the Customer Information Control System (CICS) and the Information Management System (IMS). Another new session feature at about that time was the support of the National Bureau of Standards' Data Encryption Standard (DES) for session cryptography. In addition, SNA capability to handle non-SNA terminals was significantly advanced by the inclusion of the Network Terminal Option (NTO) software product on the Network Control Program (NCP) in the IBM 3705 Communication Controller.

In 1980, major extensions were made to SNA configuration flexibility and to its transport services. Improvements included multirouting, parallel links between nodes, priority transport, and global congestion control within the network; fully meshed con-

nectivity within the backbone transport network was introduced.

The 1980s have brought the following advances:

- Advanced Program-to-Program Communication (APPC)—or LU 6.2—has introduced new, peer-oriented capabilities for communication between application programs.
- SNA Low-Entry Networking (SNA/LEN)—or node type 2.1—provides direct (one-hop) peer communication for small systems.
- Document Interchange Architecture (DIA) and Document Content Architecture have extended SNA support for applications within the office environment.
- SNA Distribution Services (SNADS) provides a store-and-forward, or asynchronous, distribution service to SNA that complements the synchronous delivery support of sessions between two end users.
- SNA Network Interconnection (SNI) and extended network addressing (ENA) have enhanced SNA routing and configuration flexibility, particularly for large networks.
- The IBM Token-Ring Local-Area Network (LAN), in conformance with national and international standards, provides high-speed communication for interconnecting information processing equipment at a local site, such as a building or campus.

These and other recent advances will be discussed in more detail in the following sections.

In summary, some of the major evolutionary trends in SNA have been the following:

- Increasing configuration flexibility, particularly to exploit advances in transmission technology
- A burgeoning set of SNA products from IBM and other suppliers
- Greater attention to network management services
- Inclusion of network standards as these have become available
- Widening support for non-SNA devices
- Expansion of routing and transport services to keep pace with installation of ever-larger networks
- Increasing function available to end users

The steady development of SNA has been tempered all along by a concern for compatibility of past products with new SNA releases. The care taken is manifested by the continued operation of older SNA terminals under newer releases of SNA. This migration sensitivity is one of the hallmarks of SNA evolution.

In the following sections, we discuss how new requirements will likely bear on the above trends. Some emerging trends are also discussed; these concern an increased focus on continuous (24-hour) operation—with a reduction in system-definition downtime; a greater emphasis on peer communication, independent of the traditional backbone network; and adaptation of local-area networking and other state-of-the-art technology.

VTAM-NCP transport network

In our survey of the directions that will shape the SNA of tomorrow, we start with the Virtual Telecommunications Access Method (VTAM) and NCP transport network. VTAM and NCP are two of the first three SNA products (along with the 3600 Finance Communication System). They provide the SNA transport network and many of the control and service functions needed to operate SNA networks.

Recent additions to the VTAM-NCP transport network include SNA Network Interconnection (SNI) and extended network addressing (ENA). SNI was announced in November 1983; it provides for the interconnection of autonomous SNA networks through gateways consisting of specialized interconnection logic in VTAM and NCP. SNI is appropriate for intercompany communication, for companies experiencing mergers and acquisitions, and for situations where independence of company divisions is needed; it enables two or more networks to merge for the purpose of user communication but to be independently managed and controlled.

To maintain the integrity of the component networks, constraints are built into the gateway to prevent one network from disrupting an adjacent network or gathering information to which it should not have access. Flow control prevents one network from flooding its neighbors with more traffic than they are prepared to handle. Names (e.g., of logical units—LUs, discussed later) and routing addresses are also independently assigned. To resolve potential conflicts in name assignment, the NetView program product provides an optional name-aliasing facility. The SNI extensions to SNA are described in detail elsewhere.[2,3]

Because each of the independent networks can use its full SNA address space, SNI can also be used to

configure networks larger than would otherwise be possible. Each network can allocate a pool of addresses available as local aliases for destinations in other networks; dynamic assignment can result in using them as needed, thereby sharing the pool over a large number of destinations in other networks. Thus, SNI provides the network interconnection function and, at the same time, possible relief from the addressing constraints some users were experiencing. This is why SNI was provided prior to a more direct solution to the addressing problem (which is our next topic).

As SNA networks grew in size, a requirement arose to extend the original 16-bit address space. These 16-bit addresses were partitioned into two pieces: a subarea address that identified the destination subarea node (containing VTAM or NCP), and an element address that was used by the destination subarea node for routing—for example, to the correct VTAM application program or to the intended terminal. In a particular SNA network, the subarea address can be chosen to be any size from one to eight bits; the remainder of the 16 bits is then used for the element address. In theory, over 64000 destinations could be addressed; in practice, this number could not be achieved because the subarea/element split has to be uniform throughout the network, and the optimum split varies from node to node.

In September 1984, IBM announced ENA, which provides for 23-bit addresses, thereby allowing over eight million destinations in a single SNA network. The subarea portion is fixed at eight bits (the previous maximum for subarea addresses), and the element portion is fixed at 15 bits (the previous maximum for element addresses). These sizes were chosen because they were already accommodated by the existing routing tables.

We view this extension as an interim step and recognize the need to provide much larger subarea addresses. This next step would be relatively easy in the architecture from an addressing perspective because space exists for 48-bit addresses in SNA formats, but it would raise more serious problems with the routing schemes. Under the current SNA routing implementation, generating and storing routing tables becomes increasingly difficult as the network grows in size.

Today, if you have a large SNA network and want to add a new subarea node to it, you must first decide where to locate the new addition and what links should connect it to the existing network. Then, usually with the help of an IBM program such as Routing Table Generator (RTG) or Network Design and Analysis (NETDA), you design the routes of the enlarged network. Once this work is completed, you load the new routing tables into the subarea nodes of the network through a system definition process. This procedure lends itself to very efficient routing, since all the routes are predefined to the network, but, because of definition time and complexity, it limits what can be done to accommodate unanticipated changes in the network configuration, such as the addition or deletion of a link or node. Such changes are likely to occur more frequently as the size of the network increases.

In addition to supporting very large networks, SNA has a requirement to make all networks easier to install and change. One of our long-term SNA design directions is to reduce, and, wherever possible, to completely eliminate, system definition.

One potential solution is to have a route-activation message carry a description of the route that the session will use through the network. This routing information could be created in an off-line process, such as is done to create the coordinated routing tables used today. But since the routing information would be stored at each source node, rather than in each intermediate routing node, routing information could be updated independently at each node without having to stop the operation of the network.

The route-activation message could be routed using the list of nodes and links that was provided at the originating node. At each intermediate routing node, a temporary entry could be created in a routing table, for the life of the session, that related the link by which the route-activation message arrived and the local addressing information generated at the predecessor node for use on that link, with the link the route-activation message was to be routed across (based on the routing information received in the message), and addressing information locally generated for use on that successor link. Thus, subsequent session traffic could be routed quickly by using the link and local address information to index the routing table, and would not have to carry the list of nodes and links defining the route.

This routing could be made more dynamic by having the network update itself through an exchange of node and link characteristics whenever a change occurs in any of these parameters. The network

could then use the most current topology information to compute the best route between two points at the time the route was requested. This route computation could also consider a class of service requested by the user; this is how the user specifies to the network whether he needs, for example, the least-cost route, the route with the least delay, or the route with the greatest bandwidth.

This dynamic routing capability could address a number of current requirements for SNA networks. First, it would permit larger networks, since the difficulties with the current route-generation process could be eliminated and intermediate nodes could store routing information only for currently active routes, not for all potential routes. Second, it would allow networks to be installed and changed more easily by reducing the workload now experienced in doing coordinated system definition. Third, it would be an important step toward fostering continuous operation, since a network need not be taken down for the purpose of updating routing tables. Further discussions of dynamic routing for SNA networks can be found in Eisenbies and Smetanka[4] and in Jaffe et al.[5]

The need for continuous operation runs deep in SNA.

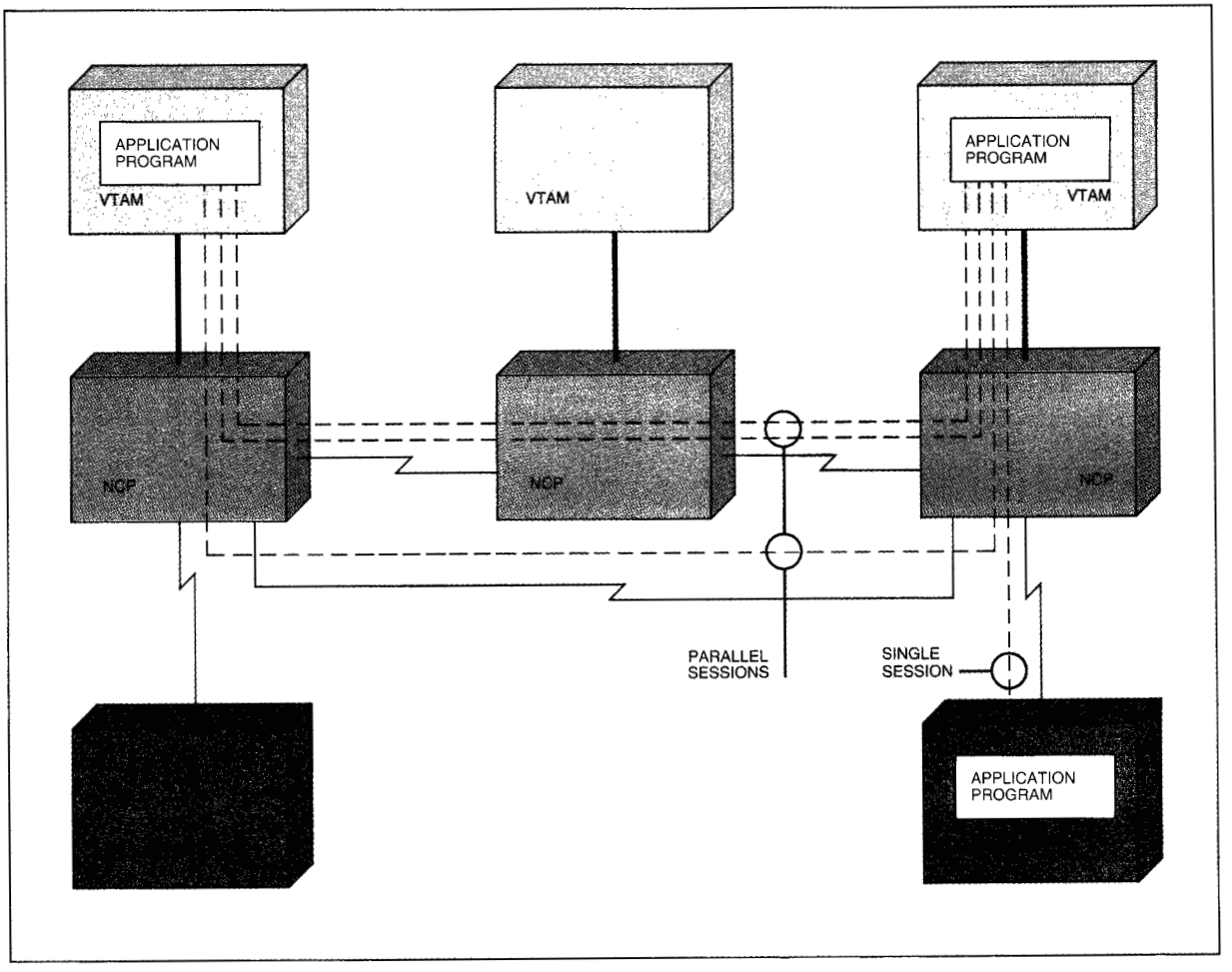
The need for continuous operation runs deep in SNA and is worthy of more discussion here. A number of features currently available in SNA can be used to increase the availability of networks and to insulate its users from outages. These include the following:

- Pause and retry logic in Synchronous Data Link Control (SDLC), which allows SDLC links to remain operational across periods of transient errors on the links.
- Multilink transmission groups, which allow bundling a number of SDLC links into a single logical link. The sender schedules data traffic for the first available link in a group, and the receiver reorders received messages, if necessary, to maintain the FIFO (first-in first-out) property of the logical link. Individual links can be dynamically added to or deleted from a transmission group without disrupting the ongoing flow of information; a transmission group fails only when the last operational link in the transmission group fails.
- Multiple routes. Networks can be configured with multiple predefined routes so that if the route serving a session fails, that session can be re-established over an alternate route.
- Parallel sessions. SNA currently allows multiple simultaneous (*parallel*) sessions to be established between host application subsystems such as IMS and CICS. These sessions can traverse different routes through the network and, where supported, comprise a resource pool; when a transaction program needs to communicate with a partner program at another host, the first available session with the desired class of service can be assigned from that pool. Should one session fail, other sessions in the pool will continue to provide session connectivity with the partner subsystem.
- Host control-point (system services control point, or SSCP) takeover, which provides protocols in the hosts and NCPs for detecting failure of controlling hosts and informing their backup hosts.
- Distributed processing. By moving application programs and data closer to the user, the user can often continue working uninterrupted by link or node failures in the communication network (and experience improved response times under normal operation).

Although SNA today provides numerous functions that can be used to configure highly available networks, further requirements exist in this area. For example, while SNA provides multiple routes, should a route fail, the sessions it carries are deactivated prior to possible reactivation over a backup route. A desirable extension is to have the network perform this route switch without disrupting the sessions.

Another requirement is to provide for backup application subsystems. However, just having a backup application subsystem ready to start taking over the moment the primary application subsystem fails will not always be sufficient. Some critical application programs can support thousands of simultaneous users; it could take a number of minutes for the backup application subsystem to re-establish and resynchronize all the user sessions. To reduce the recovery time for these critical applications, a need exists to pre-establish the backup sessions and have them available for immediate use should the primary

Figure 2 SNA session capability today



application subsystem fail. For IMS applications, this capability has been announced as the Extended Recovery Facility (XRF). At the time that IMS support of XRF was announced, IBM also stated that the general direction of its development effort is to provide XRF capability for CICS.

Small systems

Thus far, we have been focusing on the VTAM-NCP transport network. We now examine SNA requirements and directions from the perspective of the peripheral nodes of the SNA network. In the past, these peripheral nodes have been predominantly display terminals, printers, and remote job entry stations. With the steadily decreasing cost of mini- and micro-processors and storage, more and more of the peripheral nodes are small *systems* such as personal computers, distributed processors, intelligent workstations, and office systems. These small systems, because of their more general nature, have greater connectivity requirements than traditional terminal devices.

One requirement is more flexible session connectivity. Figure 2 shows the ways sessions can be connected in SNA networks today. Host application subsystems can have sessions with other host application subsystems and with outboard terminals and small systems. The host-to-host connections can employ parallel sessions. These parallel sessions can be used to increase transaction bandwidth (each session can serve one active transaction), to provide for a distinct

class of service selection, and to improve performance and availability by fanning out traffic across different routes between the hosts. Peripheral nodes, however, are currently limited to a single session per LU (a user port discussed later), and that session must be with an LU in a subarea node, e.g., either a host application subsystem such as CICS or a transaction-routing LU such as Network Routing Facility (NRF) in a communication controller. A requirement exists for small systems to enjoy the session connectivity that hosts enjoy today, namely the ability to use parallel sessions and to have direct session connectivity with any other destination in the network.

These small systems also have requirements for communication outside the VTAM-NCP environment. One simple but important form of communication is direct peer-to-peer: just two nodes and a link between them. Although this configuration is the simplest possible, it is increasingly important because of the current trend in the communications environment toward high-connectivity, multiaccess facilities such as local-area networks, X.25 networks, and Integrated Services Digital Networks (ISDNs), which are discussed later.

In 1983, a new peripheral node type, node type 2.1, was incorporated into SNA to provide this peer-to-peer form of communication. Initial IBM implementations of this new protocol, called *Low-Entry Networking*, are available on the systems listed in Table 1. Direct peer-to-peer communication had been available earlier on IBM SNA products such as 8100/DPPX and previous releases of the 5520 Administrative System. The newer node type 2.1 protocols provide the capability to carry LU 6.2 sessions (including parallel sessions) and demonstrate the SNA direction for peer-to-peer communication between compatible small systems.

In designing networking solutions for small systems, it is important to recognize the differences in operating environments between large and small systems. Procurement and operational decisions for small systems are generally decentralized and dynamic, resulting in frequent change. Yet, technical support from systems programmers and network operators is far more limited. Another difference affecting design decisions is that small systems typically need not support the high traffic volumes of large systems; moreover, small systems have more stringent entry-cost requirements.

To better meet the networking requirements of customers with small systems, the System/36 *Advanced Peer-to-Peer Networking* (APPN) feature was introduced last year. APPN has been designed for the small-system environment; ease of use, simplicity, low entry cost for small systems, configuration flexibility, peer-oriented protocols, and tolerance of node and link failures were some of the key design considerations.

Table 1 Initial IBM implementations of Low-Entry Networking

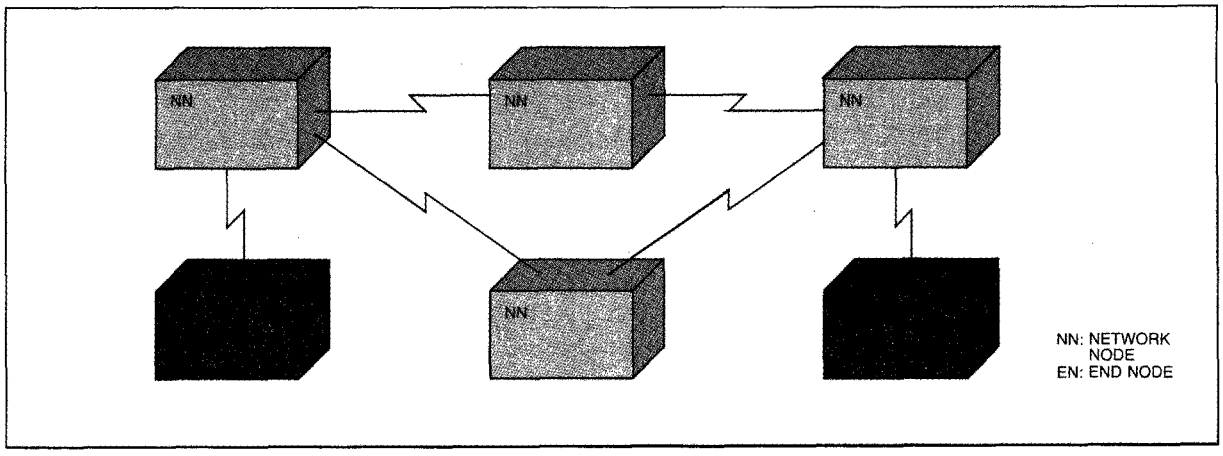
System/36 System Support Program
System/38
System/88 APPC
APPC/PC
Series/1 Realtime Programming System
IBM 3820 Page Printer
IBM LAN Print Manager Program

The System/36 APPN feature extends the direct peer-to-peer communication provided by low-entry networking and attaches type 2.1 nodes as end nodes to the APPN network. Figure 3 shows the two types of nodes in an APPN network: *network nodes*, which provide session-level intermediate routing and other functions such as directory services, and *end nodes*. (Either type of node can be attached directly as a traditional *peripheral node* through a *boundary function* to the SNA backbone transport network of subarea nodes.) End nodes in APPN networks have the same session connectivity as network nodes. This session connectivity includes the ability to use parallel sessions and to have session connectivity with any other node (network or end) in the network. Thus, APPN provides the same session connectivity that has been recognized earlier as a requirement for the VTAM-NCP transport network.

APPN is a nonhierarchical, peer-oriented scheme that uses dynamic topology update and route-determination protocols similar to those discussed earlier as requirements for the large-system environment. A dynamic, distributed directory allows destination LUs to be found without predefining the location of each LU; a later section discusses this further. These two features, dynamic route determination and a dynamic LU directory, provide for ease of installation and of frequent network changes by eliminating the need for coordinated network definition of this information.

Further discussion of the requirements for networking in a small-system environment and of the APPN solution appears elsewhere.[6]

Figure 3 A System/36 APPN network



The evolution of small-system networking can hardly end with stand-alone networks; we anticipate a requirement to connect these networks of small systems to the high-capacity transport networks as shown in Figure 4. These connections should allow sessions between *small-system A*, for example, and an application in one of the System/370 hosts; they should also allow small systems to communicate peer-to-peer across the SNA backbone networks (for example, from A to G) and share the high-capacity links that often are in place.

Provision must also be made for managing networks of small systems. These network management techniques should be consistent with the network management functions that are in place for SNA backbone networks, so that when networks of small systems are connected to networks of large systems the entire consolidated network can be centrally managed.

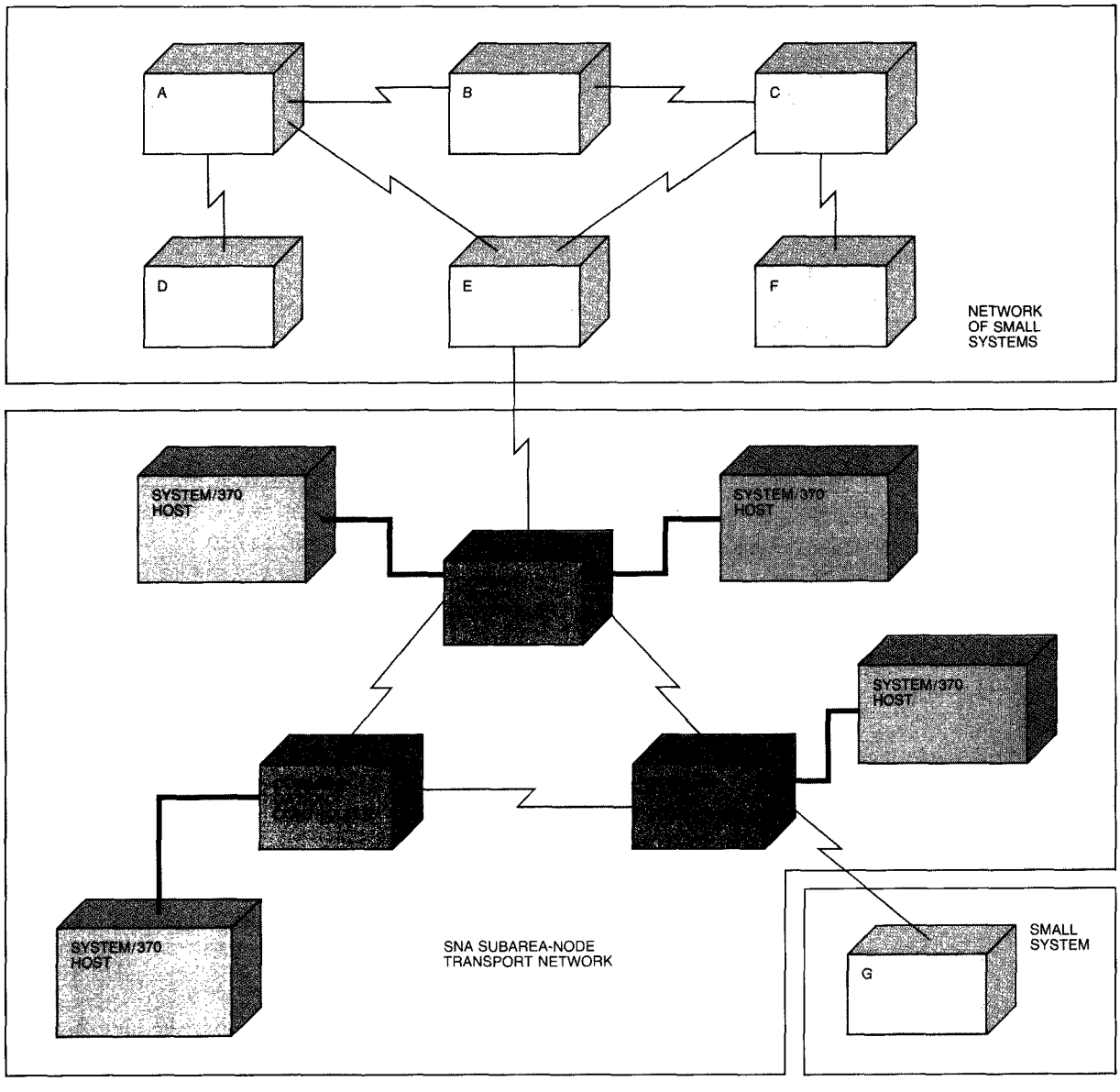
Logical units

The *logical unit* (LU) has a central role in SNA, namely as the intermediary between the transport network and the people, devices, and application programs using or attached to the network. Figure 5 shows the position of the LU in the layered structure of SNA.

Up to this point, we have discussed functions existing mainly in the transport network, which provides global protocols and services such as network-wide flow control and routing. By contrast, the LU is concerned primarily with session protocols for paired end users. LUs serve as the attachment points for the ultimate destinations of data (application programs, data bases, and devices) and provide the end-to-end session protocols in support of communication between these network resources. Whereas the transport network provides global flow control so that links and intermediate nodes in the network are not overloaded, the LU provides end-to-end flow control so that, for example, an application program does not send data to a printer faster than the printer can handle it. Other functions include session cryptography, name-to-address translation (using a distributed directory services capability of the network), and blocking and subdividing message units for efficiency in transmission and buffer usage by the LU.

A number of LU types are defined in SNA, with the earlier ones (LU types 1, 2, and 3) optimized for asymmetric host-application-to-device communication. With today's trend toward personal computers, office workstations, and other small systems, new communication requirements exist at the LU level as well as at the transport network level. The key requirement at the LU level is for general program-to-program communication. A single set of protocols is needed for all types of communication, including host to host, host to small system, and small system to small system. It should provide a range of functions suited to such products as the IBM Personal Computer and the IBM 3820 Page Printer at the low end, and CICS and the System/38 at the high end. The protocols should also provide a new base for device support, allowing for flexible function distribution between devices and host applications.

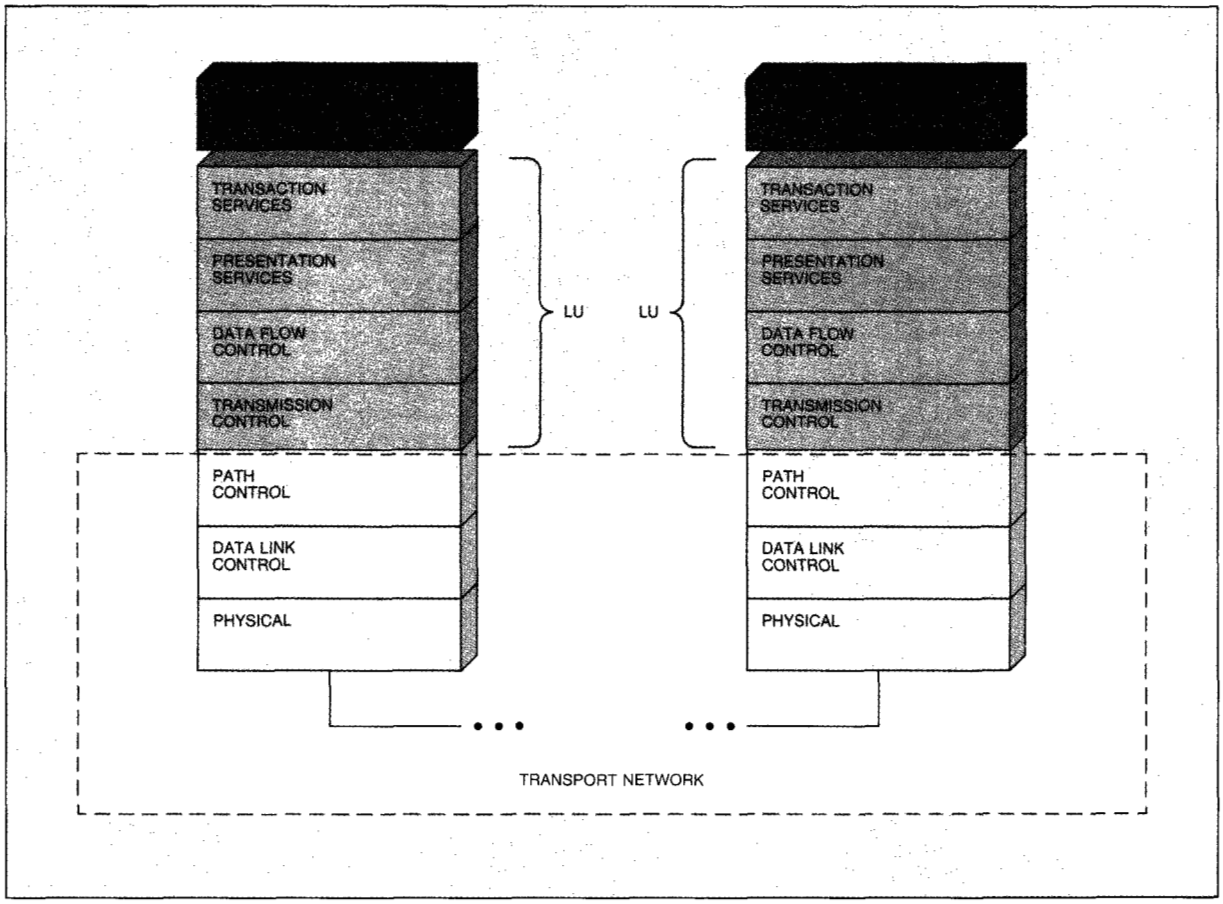
Figure 4 Combined network of small systems and large systems



To meet these requirements, a new LU type, 6.2, was introduced in 1982 as Advanced Program-to-Program Communication (APPC). Initial IBM implementations included CICS, Print Services Facility, 8100/DPPX, and the systems listed in Table 1 that provided node type 2.1 support. LU 6.2, as well as other SNA protocols, is widely available from other vendors. An important contributing factor in the development of APPC was the availability of powerful microprocessors in the outlying terminals, allowing these units to handle their end of a peer LU-LU session. Making use of this capability is a good example of exploiting advances in technology.

To support a broad range of distributed applications across any set of LU 6.2 products, LU 6.2 defines a set of generic commands (*verbs*), implemented by each LU 6.2 product in its own syntax but with

Figure 5 Role of the LU



common semantics, that application transaction programs can use to communicate, independent of the details of the underlying configurations and protocols. The services (e.g., session allocation, sending, receiving) provided by these verbs are defined elsewhere.[7]

LU 6.2 was designed to include SNA-defined transaction programs such as those for SNADS and DIA (discussed later), to serve device- or product-specific transaction programs such as those used to communicate with the IBM 3820 Page Printer, and to be used by user-written transaction programs. Users can provide their own transaction programs on LU 6.2 implementations such as CICS, System/36, System/38, and APPC/PC (a general LU 6.2 and node type 2.1 product for the IBM Personal Computer), which can be programmed by the customer. Such implementations are said to have an *open* application program interface (API); implementations such as Scanmaster I, in which the LU 6.2 implementation is limited to serving only prepackaged transaction programs, are called *closed*-API products.

To meet the low entry cost and the high function requirements, LU 6.2 has a base set of functions and a limited set of options. Every open-API implementation supports the base and may also support any of the option sets. One option, implemented by CICS, can synchronize updates to multiple data bases so that all updates either succeed or fail together as an atomic unit of work. Closed-API products need provide only the functions used by their coupled transaction programs.

LU 6.2 has recently been enhanced with the addition of option sets for transaction program security. The foundation of these protocols is a two-way verifica-

tion exchange used to check the identity of the session partner. This verification is accomplished by having each LU generate and transmit a random number, and having the partner LU return that value encrypted under a shared session password. Verifying the identity of the session partner by determining whether it can correctly encrypt a random value protects against a "playback attack" (using data obtained from a previous session through illicit monitoring), since the password is not sent over the network. Later, when a transaction program initiates a conversation with a remote program using the session, it can include a user identification and a user password. Because of the trust established in the earlier verification, these fields need not be encrypted; they are used by the receiver to verify that the conversation initiator has the authority to gain access to the requested resource, such as a file or transaction program. Transaction programs with very high security requirements can additionally transmit fully encrypted data (including the user identification and user password) throughout their sessions.

SNADS provides the asynchronous transport of data.

Further information on LU 6.2 can be found in References 8–10.

In the following sections, we explore the progress in the transaction services layer, the top layer of SNA.

SNA Distribution Services

SNA Distribution Services (SNADS), made available in 1984, consists of a set of IBM-supplied transaction programs that use a network of LU 6.2 sessions to provide a store-and-forward distribution capability in an SNA network. The LU 6.2 base provides a synchronous, connection-oriented, logically point-to-point service for application programs, analogous to a telephone service. SNADS builds on this basic service to provide a distribution capability analogous to a mail service, wherein distributions can be made, possibly to multiple destinations, without the need for active sessions between the endpoints.

SNADS provides, at a user's request, the asynchronous transport of data such as documents and files supplied by that user to one or more other users. The service is provided by a network of nodes called *distribution service units*. A distribution typically begins at one such node and may spread out to many, as copies are fanned out at the appropriate point in the network and forwarded toward the specified destination users. Each move from one node to the next complies with formats and protocols defined for SNADS and the lower layers of SNA. When failures or planned outages occur in the network, SNADS allows material to be stored at distribution service units for later forwarding when the network resources are available. SNADS frees the user of concerns about network resource availability. SNADS is particularly appropriate for batch-oriented, one-way flows found in such applications as document distribution, file transfer, and job networking.

A full-service distribution service unit can support multiple users and multiple applications and provide intermediate store-and-forward routing services on behalf of other distribution service units. The architecture describes how a distribution service unit can be optimized to a particular application, user, or role in a network. For example, a printing device could be defined as having a receive-only single-addressee role in the network. In this way, low-cost basic implementations can be provided in one release, and their functions and role expanded in subsequent releases, without inconveniencing session partners in the network.

SNADS provides its users with defined user names to identify the senders and recipients of distributions. The user names are location-independent, thereby allowing users to move from one node to another without changing their names. The current location of a user is found by reference to a SNADS-defined directory. Movement of a user from one node to another need be reflected only in the directories of the user's old and new nodes. Updating other directories in the network can proceed on a schedule determined by a network administrator.

The route taken by a distribution through a network is determined not only by the location of the recipient, but also by other properties of the distribution request. Depending on integrity requirements, the

amount of data, and the priority of the request, different routes may be selected through a network.

SNADS defines a full set of actions to be taken in various error situations. If a distribution cannot be delivered, feedback to the originator specifies exactly which recipients were omitted and why.

SNADS is currently implemented in a number of products that will use it primarily for document distribution; these are the Distributed Office Support System/370 (DISOSS/370), System/36, System/38, 8100/DPPX, Series/1, and 5520 Administrative System. Because document distribution is the most common application in these first implementations of SNADS, the initial SNADS formats were designed to be compatible with those in Document Interchange Architecture, one of IBM's architectures for the office.

Further information about SNADS exists in References 11 and 12.

Architectures for the office

To handle situations generic to the automated office, IBM has developed architectures specific to this important application: Document Interchange Architecture[13-15] and Document Content Architecture.[15-17]

Document Interchange Architecture. Like SNADS, Document Interchange Architecture (DIA), introduced in 1982, is also part of the transaction services layer of SNA. It provides a set of protocols that define how several common office functions are performed cooperatively by IBM products. These include the filing, searching, and retrieving of documents and memos as part of the document library services of DIA. The document distribution services of DIA provide for the sending and receiving of documents or memos via SNADS or an underlying LU 6.2 session, and include listing items pending receipt, canceling or sequencing their delivery at the recipient's request, and allowing access to software mail boxes and files by other authorized users. The formatting and processing of documents are defined in application processing services. The implementation status of DIA is similar to that of SNADS, but also includes IBM PC, Displaywriter, and Scanmaster implementations.

Document Content Architecture. A vital component of the office architectures is the Document Content Architecture, which provides the formats for describing the form and meaning of objects that are managed by DIA. Currently, two forms are implemented:

- Final-Form Text provides primitive format controls within a data stream in a generic fashion to allow device-independent document presentation. This provision allows the sender of a document to control the formatting and print integrity of a document at its final destination without knowing the print device characteristics of the destination.
- Revisable-Form Text allows interchange of documents in a form that is suitable for revision. Text processing indicators are included with the text, and may themselves be revised.

While the two current types of Document Content Architecture are a good beginning, others are needed to describe the mixing of information types within a document, such as text, graphics, image, and voice annotation data. An architecture serving mixed data would allow documents that integrate a variety of data types to be exchanged by future office workstations.

Directory services

One of the design goals of SNA has been to insulate the user and the user's application program from the characteristics of the communication network and to allow workstations and application programs to be moved among processors without impacting other workstations and application programs that communicate with them.

To achieve this design goal, SNA has distinguished between resource names and their addresses. A name is a relatively stable identifier that users can apply to remote workstations and application programs (typically those with which they want to interact via a session). By contrast, an address can vary according to operations decisions made in the network. Users and their application programs access resources by name. The system uses the name as a key to a directory that provides the current address of the requested resource.

In early SNA networks, the directory tables were defined in the system services control point in a single System/370 processor. The control point acts as a mediator in LU-LU session initiation, translating names to addresses and checking resource availability. The use of a central directory in the control point simplified the management of the directory for additions, deletions, and changes.

With the introduction of multiple-host networks in 1977, the control points cooperated in providing

directory services, with each control point being responsible for the detailed address information on a subset of the LUs in the network; each control point knew all LU names and the associated control point that could resolve a specific name to an address. Successive designs of the control point have reduced the amount of coordinated predefinition of resources by eliminating redundancy among the directories in different control points. Starting with the SNI release of SNA, a control point could perform a trial-and-error search of other control points for LUs not found in its own directory; this could further reduce the number of control points that need to be updated when new LUs are brought on line.

In SNADS, a user directory is referenced to determine the distribution service unit (address) of an intended recipient. SNADS products have implemented their directories, which are manually maintained, as part of the distribution service unit, rather than in the control point.

Two trends stress the current design point of manually maintained directories; as networks become larger, the frequency of directory updating increases and the number of directories that must be consistently maintained grows; furthermore, the trend toward peer connectivity among small systems requires that small systems also maintain directories, whereas they may formerly have depended upon a large host directory. Both trends result in many more directories and increased update activity in a network, and point to a growing need for the directories to be automatically and dynamically maintained.

The System/36 APPN feature has introduced the ability to register LUs only at their local (home) directories. Automatic searches of the various directories throughout the APPN network result in eventually finding the LU if an active path to the LU is available. Once found, the location of an LU can be stored in cache memory so that searches need not be repetitive.

Management services

The previous sections of this paper discussed the advances in the functional richness and configuration flexibility allowed by SNA. SNA users can connect multiple SNA networks, small systems to large systems, and various other devices, all within the same, composite network. The actual attachments may be over various transmission media (e.g., telecommunications links, System/370 channels, X.25 networks, local-area networks), using equipment from many different suppliers. This configurational flexibility exacts a price in making the network more complex to manage. It places strenuous requirements on the architecture to allow for such things as nondisruptive change, immediate and simple notification of problems, and high availability. The goal for SNA is to allow full end-to-end management of a network, including the telecommunications and non-SNA equipment.

The goal for SNA is to allow full end-to-end management of a network.

Management services, also known as *network management* or *communications network management*, include the monitoring and controlling of a network. The requirements for managing an SNA network fall into four major management services categories:

- Problem management—the function of managing a problem from its detection through its resolution. The steps of problem management are (1) problem determination, (2) problem diagnosis, (3) problem bypass and recovery, (4) problem resolution, and (5) problem tracking and control.
- Performance and accounting management—the process of quantifying, measuring, reporting, and controlling the usage, responsiveness, availability, and cost of a network.
- Change management—the planning, control, and application of changes (additions, deletions, and modifications) to the resources of a network.
- Configuration management—the control of information necessary to both logically and physically identify network resources and to indicate their relationships to one another.

Management services components in SNA. The management services functions are represented in the architecture[18] by a management services component in the control point, in the physical unit (PU),[19] and in the individual layers of SNA. This

structure provides a framework by which the aforementioned requirements can be satisfied.

Each layer of SNA (see Figure 5) is responsible for controlling the resources associated with that layer. This control is accomplished through a component called *local management services*. For example, routing information that is used for problem management is gathered by the local management services component in path control. Once gathered, this information is sent to the management services component in the PU.

Physical unit management services is responsible for gathering local management services information to its node or attached links, performing some services such as reformatting and time-stamping the data, and sending the information to the control point for processing. Control point management services is responsible for collecting management services data from the network, analyzing the data, and taking the appropriate action based on that analysis.

Management services trends. The release of multihost support in SNA focused more attention on problem determination. Network Problem Determination Application[20] was introduced in 1979 to help the network operator perform problem determination. Initially, statistics were kept at each node and collected by a host computer.[21,22] This procedure allowed a person with access to the host to look at the statistics and attempt to diagnose problems in the network. Today, each SNA node is responsible for its own error analysis to determine whether a problem exists and whether recovery action can be performed. If a problem exists that cannot be resolved locally, the node sends an Alert signal to the host to indicate that a component in the network is unavailable and that intervention is required. The Alert contains a general classification of the problem, a description of the probable cause of the problem, identification of the failing resource, and any additional details pertaining to the problem. The Alert mechanism allows immediate notification of problems in the network in order to foster quick and easy problem determination. Statistics are still kept by each node and collected at a host for problem trend analysis.

Recent enhancements to problem determination have focused on the communication links. In 1986, another Alert-like signal called *Link Event* was announced for management of the attachment to the IBM Token-Ring Network, and Link Problem Determination Aid 2 (LPDA-2) was announced for enhanced management of the IBM modems. Link Event is similar to Alert, but differs in that the node reporting the problem cannot provide a probable cause of the problem, possibly because of a lack of configuration knowledge at that node. In this case, configuration knowledge at the host is required to complete problem determination. LPDA-2 allows for more comprehensive diagnostics and increased configuration flexibility with the new IBM modems.

In accordance with the previously stated goal of managing the entire system, SNA management services have recently been extended to the private branch exchange (PBX) environment to include support of Alerts from the ROLM Computerized Branch Exchange (CBX). This is a step toward providing centralized management for both data and voice equipment.

In 1984, SNA management services were extended in the category of performance management. Network Logical Data Manager (NLDM)[23] introduced the support of response-time monitoring, which allows the network operator to validate certain predefined end-user service levels for specified LU-LU sessions. Response time is measured by the time elapsed between the instant an LU recognizes a request from its end user and the instant it receives the reply from the session-connected partner LU to which it sent the request. The predefined values can be changed dynamically by the network operator. The response-time values can be sent unsolicited upon threshold overflow or can be solicited by the network operator.

The network operator can request a summary of the response-time data for a specific LU over a user-defined period of time, detailed data for a specific session for a single collection period, and the long-term trend for a specific LU.

In addition, management services (and NLDM) have been extended to capture information about logical resources. Data captured by the control point on logical resources include session start information, normal session termination information, abnormal termination information (e.g., sense data), and virtual route information. This information is available for sessions that were started through the assistance of single or multiple control points.

With the introduction of SNA network interconnection, a session can be established wherein the endpoints of the session reside in different networks. In

this case, additional session information is required for problem management. Retrieval of session information includes the gathering of session control-block information from gateway nodes (NCPs) for cross-network sessions.

Future requirements. The requirements for SNA management services increase as SNA protocols and telecommunications complexity continue to advance; management services will need to be an integral part of each SNA enhancement. Common solutions are sought, whether for the management of small or large systems, interconnected via SDLC, X.25, or local-area networks, and whether transporting voice or data.

A trend in both IBM SNA products and SNA management services has been to provide more granular monitoring and control of network components. Hence, such products as the 586X series of modems and the 3710 Network Controller offer extensive problem management features integral to their design. In turn, the architecture allows telecommunication links, for example, to be monitored and controlled at the level of their most basic subsystems: component adapters, modems, line concentrators, and transmission media. This provision has vastly improved problem management in SNA networks.

In the future, similar attention to other management services categories will be vital. For example, the ability to nondisruptively effect changes to the network from a central location is a key requirement. SNA Distribution Services could be used to distribute microcode and software updates to many nodes at the same time. This distribution might include installation instructions, such as when to install the change, and back-out instructions in case of failure. Additionally, management services must keep pace with other SNA capabilities such as peer-to-peer communication and networks of small systems. Requirements such as centralized management must be maintained regardless of the network reporting structure (i.e., hierarchical or peer-to-peer). Of course, the location of the central facility may be affected by the time of day or system backup status.

As SNA strives to provide full end-to-end network management, inclusion of integrated voice and data, and management of a multivendor environment, more comprehensive management services become paramount. A more generic management services architecture, for example, could facilitate a greater integration of non-SNA products into SNA management services. An existing manual[18] describes the current formats and protocols involved in managing the components of an SNA network.

Local-area networks

Even as architectural solutions for improved connectivity continue at the higher layers of SNA, the announcement of the IBM Cabling System in 1984 and the Token-Ring Network in 1985 provided major improvements in dynamic connections at the physical and data link control layers. The first product supporting the Token-Ring Network was the IBM Personal Computer. Subsequently, support was added to the IBM 3174 Communications Control Unit, the 3725 Communication Controller, and the 3270 PC. The System/36 can attach to the Token-Ring Network via support in the IBM PC AT.

When first announced in 1984, the Cabling System allowed an SNA network to be physically reconfigured without running new coaxial cable to specific locations in a building. By introducing a structured wire approach and wiring closets to prewire a building with either twisted-pair copper conductors or optical fibers, today's workstations can be moved from office to office by simply plugging them into a wall and reconfiguring them at a conveniently located wiring closet.

Although ease of reconfiguration is a desirable goal, the ultimate objective is to eliminate entirely the need for manual intervention by a systems professional when moving a workstation from one office to another. This objective was accomplished in 1985 with IBM's announcement of the Token-Ring Network on the Cabling System.

The Token-Ring Network consists of the wiring system, a set of communication adapters (*stations*), and an access protocol that controls the sharing of the physical medium by the stations attached to the local-area network (LAN). The IBM Token-Ring Network is based on the IEEE standard for a token-ring LAN. The token-ring LAN[24] is one of several LAN standards developed by the IEEE 802 committee and submitted to the International Standards Organization (ISO). (Other IEEE standards include one for CSMA/CD on baseband cable and a token-bus standard on broadband cable. Each of these IEEE standards is also being processed by ISO.) A token-ring LAN is unique among these LANs, in that the nodes are physically connected serially by a transmission medium, such as twisted pairs or optical fiber. Access

to the transmission medium is controlled through the use of a unique bit sequence (*token*) that is passed from one station to the next. When a station has a message unit (*frame*) to transmit, it modifies the token to a frame by changing the bit pattern of the token to a start-of-frame sequence; the frame is then transmitted. When the station has completed frame transmission, and after appropriate checking for proper operation, it initiates a new token so that other stations have an opportunity to gain access to the ring.[25]

A bridge combines two token rings into one logical ring.

An important part of the token protocol is the ability of a station to reserve the token for use at a specified priority. It ensures that the next token issued will be at the highest priority requested, and allows a station to gain faster access to the ring for frame transmission than would otherwise have been possible.

To ensure that a token is always available on the ring, one station is elected as the *token monitor*. The function of the token monitor is to detect error conditions in token operation such as a continuously circulating frame or the absence of a token on the ring. The capability to be a token monitor resides in each station, and is determined by an election process when normal token operation is disrupted.[26-28]

Several advantages exist in choosing a token-ring configuration for a LAN, including ease of fault isolation, performance stability under load, the use of predominantly digital, rather than analog, engineering, and its potential to use optical fiber technology.[29]

To take full advantage of the peer-to-peer connection capabilities inherent in a shared physical medium, a station on the Token-Ring Network could use the data link control, called a *logical link control* (LLC), as defined by the IEEE 802.2 committee for LANs. This LLC employs the asynchronous balanced mode of operation (like that in high-level data link control, or HDLC) when a link connection is established, thereby allowing either station to send data link commands at any time, and to initiate responses independently of the other link station. This mode provides for a balanced type of data transfer between two link stations that operate as equals on a logical point-to-point link.[30] The number of logical links sharing the same ring equals the number of distinct pairs of communicating stations.

When a token ring reaches its capacity, either physically, in terms of the number of stations it is capable of supporting for the required distance, or when the bandwidth is exhausted and the performance is not acceptable, a *bridge* can be added to combine two token rings into one logical ring. A bridge is a device that copies a frame from one ring and transmits it on the other. Bridges can be used to combine a number of small rings to preserve the integrated connectivity in an establishment while providing better fault independence and performance.[27] Locating stations on a ring, or on multiple rings connected by bridges, can be performed dynamically by broadcasting requests for specific station addresses. Once the station is located, routing data through bridges can be done efficiently by including the routing information to the destination station with each frame; by this action bridges are allowed to copy frames from one ring to another based on routing information in the frame format, without building, referencing, and maintaining complex tables. Thus, expansion of the LAN to include additional rings need affect only the connecting bridge and can be transparent to all other stations.

In April 1986, IBM announced extensions of the token-ring support allowing multiple token rings to be interconnected via bridges to form logically composite LANs and to include connection to System/370 computers through the IBM 3725 Communication Controller and its related software; up to eight token rings can be connected to host processors through a single controller. Special gateways are not required to connect to the full SNA backbone network because the LAN requires only the functions of the physical and data link control layers of SNA, not the higher layers such as path control. These layers remain independent of the LAN, just as they are for alternative lower layers. Additional network management support was also announced whereby an IBM Personal Computer on a ring may be used to monitor errors, perform problem determination, and interact with an operator wanting to evaluate ring status. Later in 1986, support for exchanging such network management data (via Alert signals) to NPDA in a host processor was also announced.

The introduction of the Token-Ring Network demonstrates again that the layered structure of SNA allows the inclusion of new technology in a nondisruptive fashion; supporting the new physical and data link control layers for the token ring had little effect on the rest of the architecture.

The long-term effects of the IBM Token-Ring Network on SNA could be far-reaching. Improving dynamic connectivity and reducing system generation requirements in SNA products become even more important when physical connectivity in an establishment creates the possibility of a "hot-pluggable," fully meshed network. That is, once an SNA workstation is plugged into an office wall, it can have immediate physical access to all SNA workstations and other SNA nodes attached to the Token-Ring Network. To translate this physical access into intelligent communication requires a consistent application program interface and a set of protocols allowing peer attachment to workstations and mainframe computers. Thus, LU 6.2 and node type 2.1 protocols in SNA will become even more important.

Link subsystems

Initially, SNA supported terrestrial links, slow-speed satellite links, and channel attachments; soon it added public telephone network dial capability. Other link-level options that have since been added to SNA include packet-switched virtual circuits using X.25, high-speed satellite links, and the previously described local-area networks. Another technology, Integrated Services Digital Networks (ISDNs), is on the horizon and is being examined as a requirement for SNA.

Recommendation X.25 defines a packet-mode interface for attaching data terminal equipment (DTE)—such as host computers, communication controllers, and terminals—to packet-switched data networks (PSDNs). The International Telegraph and Telephone Consultative Committee (CCITT) introduced X.25 in 1976 and updated it in 1978, 1980, and 1984. IBM products that offer X.25 capability comply with the 1980 version of the interface; some products provide an option supporting the 1984 version of the interface. IBM recognizes the requirement for additional products to support the 1984 version. A PSDN provides connectivity to other DTEs using X.25 virtual circuits. Permanent virtual circuits provide fixed connectivity between DTEs, whereas switched virtual circuits provide dynamic connectivity using virtual call setup and clearing capabilities. The X.25 interface also defines user facilities such as interface parameter negotiation, reverse charging, and closed user groups.

Table 2 IBM products that support X.25

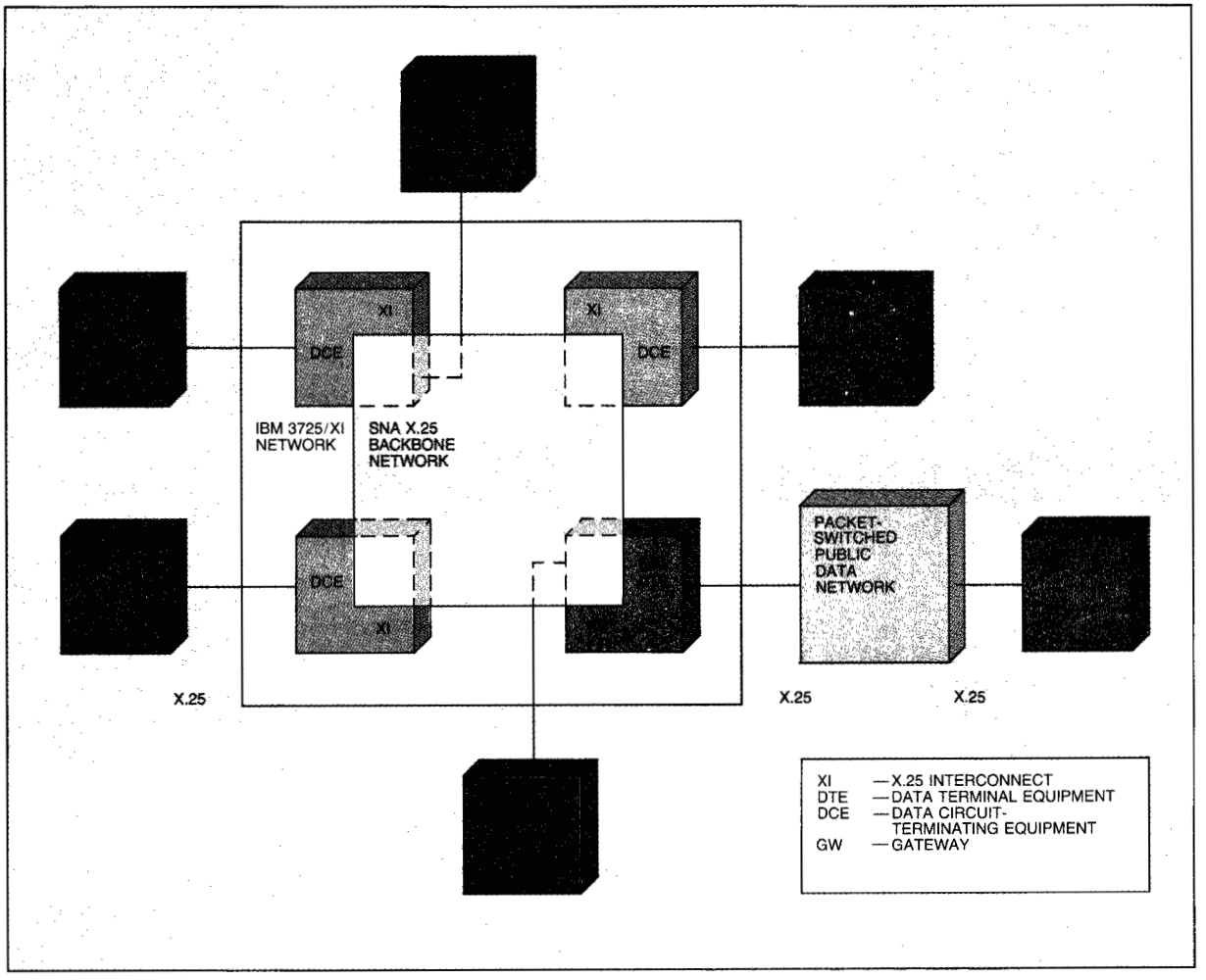
Product Type	IBM Product
Communication Controller	3705/3725/3720 (all with NPSI)
Processor	Series/1 (RPS, EDX), 4361, 9370, System/36, System/38, System/88, 8100
Personal Computer	5150, 5160, 5170
Licensed Program or PRPQ	XI (for DCE function on 3725/3720) TPNS, X.25 NPSI (for DTE function on 3705/3725/3720)
Network Interface Adaptor	5973-L02
Display Station	5251-12
Display Control Unit	3174, 3274, 5294
Finance Controller	4701
Network Controller	3710
Computerized Branch Exchange	ROLM CBX

Having participated in the development of X.25, IBM announced the capability in 1977 for attaching several DTE products to PSDNs in Canada and France. One of the early products was a network interface adapter, the IBM 5973, a stand-alone unit that is a converter between the link control protocol of SNA nodes—SDLC—and the X.25 protocols. This adapter allowed most IBM products that communicate with System/370 hosts to use X.25. Initially, the communication controller (IBM 3705) for SNA System/370 hosts used a special software adaptation for X.25. In 1980, an X.25 program product for the communication controller was introduced, allowing packet-switched communication with other SNA products and connections with non-SNA DTEs.

The IBM direction with respect to X.25[31] has been to integrate the interface into SNA products where required, so that the customer can choose the most economical communication medium. If network tariffs favor X.25, one can choose packet-switched services; otherwise, traditional switched or nonswitched services can be used. By the end of 1986, 24 IBM products had been announced supporting the X.25 interface in more than 30 countries. Table 2 lists a broad sample of the IBM products that have X.25 capability. Reference 32 contains additional information about IBM's X.25 SNA products.

All SNA products that offer an X.25 1980 interface conform to an IBM-defined specification[33] for attachment of SNA products to PSDNs. The most recent enhancement to this specification is called Enhanced Logical Link Control (ELLC). Sometimes virtual circuits are interrupted by the PSDN, causing inconven-

Figure 6 X.25 SNA interconnect



ience to the users of certain products. This inconvenience is reduced with the implementation of ELLC in low-end computers and terminals that provide a dynamic packet error detection and recovery procedure across one or more PSDNs between SNA nodes.

An IBM SNA X.25 interface specification[34] has been published to describe aspects of the CCITT 1984 recommendation to be supported in SNA. Some of the new functions in the 1984 recommendation that have been included in the SNA X.25 1984 interface specification are

- Multiple links between the DTE and the PSDN
- Redirecting a virtual call by the PSDN from the called DTE to an alternate DTE
- Hunt groups that allow the network to assign a call to one of several target DTEs
- An address extension capability that allows a DTE on a private packet-switched network to call a DTE on a public PSDN, and conversely

The SNA nodes discussed above are DTEs that attach to packet-switched networks. Some customers have the need for equipment from different vendors to communicate with one another over packet-switched networks. If the customer has a mix of SNA traffic and non-SNA traffic, the SNA backbone network can be used to carry the non-SNA traffic. The capability to add a PSDN appearance of the X.25 interface to SNA has been announced as a special SNA adaptation.[35] In such a configuration, the SNA network provides

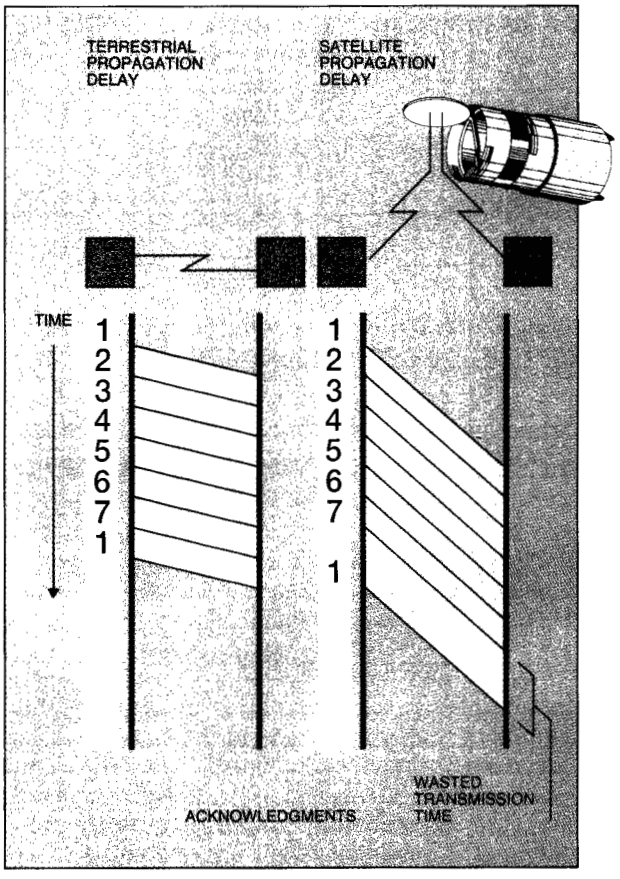
X.25 permanent virtual circuit, virtual call, and user facility services to using X.25 DTEs. The SNA traffic and the X.25 traffic share the common SNA backbone network. Another feature of the adaptation, called X.25 SNA Interconnect (XI), is a gateway capability to allow X.25 DTEs on the SNA backbone network to communicate with X.25 DTEs on a public PSDN or another private PSDN. See Figure 6.

A key aspect of X.25 is that it is an interface specification, not a network architecture. The internal operation below the X.25 interface, such as routing, flow control, and management services, is not specified by standards. Inclusion of an X.25 data circuit-terminating equipment (DCE) capability and related network services within SNA is a relatively straightforward and natural step.

Some users find it advantageous to send their SNA traffic over satellite circuits. Because most communication satellites are in geostationary orbits above the equator at an altitude of about 23000 miles, the delays in sending information from one earth station to another are long compared to those for a terrestrial circuit that connects the same two points on the earth's surface. Consequently, communication protocols must be designed to accommodate the long propagation delay of satellite circuits. Figure 7 shows the effects of delay on a link-level protocol that allows the transmitter to have up to seven unacknowledged message units in transit. In the terrestrial case, acknowledgments return to the transmitter in time for it to continue sending new messages, allowing efficient use of the link. In the satellite case, the transmitter stops to wait for an acknowledgment because of the longer propagation delay, and unused time is introduced on the link.

Detailed studies[36,37] show that interactive and batch applications can use satellite links satisfactorily at speeds up to 19200 bits per second when the satellite link is attached to an SNA peripheral node. The 3710 Network Controller provides satisfactory performance over satellite links at speeds up to 64000 bits per second when the satellite link connects the 3710 to a 3725 or 3720 Communication Controller (using SDLC with a modulus of 128). Batch and interactive traffic are carried satisfactorily at link speeds up to 256000 bits per second over satellite links connecting SNA 3725 and 3720 Communication Controllers. A special adaptation is available that allows two SNA 3725s to communicate over satellite links at speeds up to 1.344 million bits per second.

Figure 7 Effects of delay on a link-level protocol



At satellite speeds above 256000 bits per second, special consideration[38,39] must be given to the types of protocol and the value range of protocol parameters at several architectural layers of the system. Because high-speed satellite and terrestrial circuits are becoming more widely available, a requirement exists to enhance SNA to accommodate them. Capabilities such as support for larger link-level sequence numbers (an SDLC modulus of 128) have been added; selective retransmission of information and other protocols optimized to the high-speed or long-delay environment, or both, are being studied.

The rapidly approaching feasibility of high-speed digital communication (in units of 64000 bits per second) will have significant impact on both data communication and telephony, and will open possibilities for interactive video applications in the foreseeable future. The ISDN standardization of the

user-to-network interfaces for these applications has been going on for several years within CCITT. In Europe, many countries already offer digital network services on "pre-ISDN" networks. These networks use a CCITT X.21-like interface for signaling and data transfer at 64000 bits per second. Some of them use slower speeds for signaling. IBM's requirement is to connect SNA DTEs to many of these networks. As ISDN evolves around the world, IBM will adapt SNA to take advantage of these and other new digital network services to meet customer requirements.

From the start, IBM has been involved in the OSI work.

IBM has consistently represented the needs of data communication applications in the standardization effort. We continue to cooperate actively in the development of a single set of worldwide standards. The CCITT ISDN Recommendations of 1984 and the interim Recommendations approved in 1986 are significant steps in the advancement of these new digital transmission services.

SNA and OSI

IBM recognizes the widespread interest on the part of users in interconnecting networks using different communication architectures. IBM supports such interconnection and publishes extensive information about SNA, including formats and protocols, which facilitates the interconnection of other systems to IBM SNA networks.

The widespread interest also gave rise to the Open System Interconnection (OSI) standards project,[40] whose aim is to provide communication protocols for interconnecting systems of different communication architectures, such as SNA and a system of another architecture.

From the start, IBM has been involved in the OSI work. We have contributed what we have learned about layered communication architectures in the past several years and also increased our understanding of advances elsewhere in that area. Clearly, some capability for interconnecting heterogeneous systems is desirable. From a vendor's perspective, a single international protocol for interconnecting heterogeneous systems is preferable to a number of national protocols.

IBM has stated that for industrial communications, it supports the National Bureau of Standards specifications of OSI Transport layer class 4 used over IEEE 802.4 LAN. The capability was demonstrated at the National Computer Conference in July 1984. More recently, at the Autofact Conference and Exposition in November 1985, IBM introduced file transfer and directory server programs that support the OSI-based Manufacturing Automation Protocol (MAP).

IBM Europe has developed native OSI software that provides System/370 support for selected functions in the OSI 4 (Transport) and 5 (Session) layers. The Open System Transport and Session Support (OTSS) program makes OSI Session layer services available to IBM host application programs that need to communicate with programs in a system implementing another architecture. This represents a further step in IBM's commitment to provide products capable of system interconnection in conformance with OSI standards.

Additionally, OSI protocols could be supported in SNA for protocol conversion; that is, we could transform SNA protocols to and from OSI protocols through conversion code in a host to allow attachment to other networks that use existing network architectures. Indeed, as reported in References 41–43, IBM Japan, in cooperation with Nippon Telegraph and Telephone (NTT), has judged the use of OSI as intermediate protocols to be a viable way to interconnect SNA networks with networks that use DCNA protocols.

The OTSS and Open Systems Network Support (OSNS) products work in conjunction with the VTAM, NCP, and NPSI SNA products to provide OSI Session layer services over an X.25 network. While SNA remains IBM's strategic network architecture, IBM will continue to evaluate standards and develop timely, OSI-compatible products based on customer and business requirements.

Refer to Reference 44 for further information on IBM's activities related to OSI.

Conclusions

SNA has evolved continually and will do so as long as new technology, applications, and requirements unfold. The layered structure of the architecture and of the implementing products allows this process to be natural and nondisruptive.

Some of the historical trends have been cited in the preceding sections. For example, the network management services provided in SNA networks have been an ongoing concern. Moreover, recently announced extensions in this area not only supply greater function for SNA networks but facilitate managing non-SNA, multivendor, and voice network components.

These extensions provide users the ability to manage mixed SNA and non-SNA networks using a single consistent framework and a proven set of network management products. The framework involves a *focal point*, which consists of a set of products that provide centralized management application support for all network components; *entry points*, which are those products that can transmit formatted network management information (such as Alert data) to the focal point about themselves as well as attached communication devices; and *service points*, which provide comparable network management support for IBM Token-Ring Networks, ROLM CBXs, selected PBXs, and non-SNA components for which entry-point support may not exist. Examples of products providing support for a focal point are NetView, Distributed Systems Executive (DSX), Network Performance Monitor (NPM), and INFO/MANAGEMENT. Examples of entry points are the System/36, System/38, 3720, 3725, 3174, System/88, Series/1, and 3708 Network Conversion Unit. An example of a service point is NetView/PC, which also provides an application program interface (API/CS) for communicating with the NetView product in System/370; various application programs can work with NetView/PC to provide problem determination and Alert management for the IBM Token-Ring Network, the IBM PC Network, and the ROLM CBX.

Configurational flexibility and accommodation of larger networks have been another continuing concern. Besides the generalized topology and larger address space now available through SNA network interconnection (SNI) and extended network addressing, specific offerings such as the IBM 3720 Communication Controller and the 3710 Network Controller, which offer a remote link concentration capability, have resulted in cost-performance advantages. Of course, the Low-Entry Networking capability allows the flexibility of peer communication without host mediation, while the System/36 APPN feature builds on this further, allowing multihop System/36 configurations.

Additional offerings have resulted in the extension of SNA capabilities into areas that are important to many customers, such as for asynchronous distribution services (SNADS) and document services in an office environment (DIA and Document Content Architecture). Another example is the inclusion of VTAM as an integral component of the native virtual machine (VM) environment, thereby enhancing performance of SNA network operation from the VM viewpoint. A more recent example of increased transaction services support in the SNA environment is the introduction of Distributed Data Management (DDM). DDM is designed (using LU 6.2 sessions for transport) to fit within the local data management of a system so that access to remote files is transparent to the application program. Currently, CICS can provide file-server support for System/36 and System/38 in this way, while System/36 and System/38 can provide both file-access and file-server support for another System/36 or System/38.

Continuous network operation needs, as served by the extensive list of capabilities mentioned earlier—from link, route, and session parallelism and redundancy to well-defined backup and takeover support—will foster ever more features that promote high availability. Advances in route dynamics and distributed directories will also play a significant role in meeting requirements in this area. The whole matter of reducing the static nature of network definition is a consuming interest; a long-term goal is to eliminate the need for static definition entirely.

Another area of traditional concern to customers is non-SNA device support. One technique, using format envelopment, was incorporated into the Non-SNA Interconnection (NSI) program product on the NCP, which allows Binary Synchronous Communication (BSC) remote job entry terminals and BSC network job entry subsystems to communicate through an SNA network and share the SNA links. Another technique employs protocol conversion. Here, besides the long-time Network Terminal Option (NTO) support for pre-SNA terminals in the NCP, new capabilities such as that in the IBM 3174 Communications Control Unit, which allows SNA and

non-SNA terminals to communicate with SNA and non-SNA hosts, continue to enhance SNA coverage in this important area. In general, the trend here has been to perform protocol conversion to SNA as close to the non-SNA interface as possible, in order to gain quickly in the operation the SNA benefits of resource sharing and network management.

Of course, one of the most visible areas of non-SNA protocol support is national and international standards. IBM will continue to play a leading role both to cooperate in formulating such standards and to include support for such standards in SNA products subject to appropriate business decisions. This paper has cited several examples of such standards, such as the CCITT X.25 and the more recent IEEE LAN standards, that have been integrated into SNA.

Finally, SNA will continue to exploit advances in technology as they appear. Here, a known requirement is to extend the peer-to-peer operation in SNA. This need follows from developments both in processor design—especially in small systems—and in transmission technology, such as for local-area networks, satellites, PSDNs, and ISDN. Other developments, not yet evident, will affect future requirements. The process will continue and undoubtedly be the cause of much interesting evolution of SNA for a long time to come.

Cited references and notes

1. R. J. Sundstrom and G. D. Schultz, "SNA's first six years: 1974–1980," *Fifth International Conference on Computer Communication*, Atlanta, GA, North-Holland Publishing Co., Amsterdam (September 1980), pp. 578–585.
2. J. H. Benjamin, M. L. Hess, R. A. Weingarten, and W. R. Wheeler, "Interconnecting SNA Networks," *IBM Systems Journal* **22,** No. 4, 344–366 (1983).
3. *SNA Format and Protocol Reference Manual: SNA Network Interconnection*, SC30-3339, IBM Corporation (1985); available through IBM branch offices.
4. J. L. Eisenbies and T. D. Smetanka, "An automatic topology update scheme for SNA networks," *Proceedings of the Seventh ICCC*, North-Holland Publishing Co., Amsterdam (1984), pp. 725–730.
5. J. M. Jaffe, F. H. Moss, and R. A. Weingarten, "SNA routing: Past, present, and possible future," *IBM Systems Journal* **22,** No. 4, 417–434 (1983).
6. A. E. Baratz, J. P. Gray, P. E Green, J. M. Jaffe, and D. P. Pozefsky, "SNA networks of small systems," *IEEE Journal on Selected Areas in Communications* **SAC-3,** No. 3, 416–426 (May 1985).
7. *SNA Transaction Programmer's Reference Manual for LU Type 6.2*, GC30-3084, IBM Corporation (1985); available through IBM branch offices.
8. *SNA Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2*, SC30-3269, IBM Corporation (1985); available through IBM branch offices.
9. J. P. Gray, P. J. Hansen, P. Homan, M. A. Lerner, and M. Pozefsky, "Advanced program-to-program communication in SNA," *IBM Systems Journal* **22,** No. 4, 298–318 (1983).
10. R. J. Sundstrom, "Program-to-program communications—A growing trend," *Data Communications* **13,** No. 2, 87–92 (February 1984).
11. B. C. Housel and C. J. Scopinich, "SNA Distribution Services," *IBM Systems Journal* **22,** No. 4, 319–343 (1983).
12. *Systems Network Architecture Format and Protocol Reference Manual: Distribution Services*, SC30-3098, IBM Corporation (1984); available through IBM branch offices.
13. T. Schick and R. F. Brockish, "The Document Interchange Architecture: A member of a family of architectures in the SNA environment," *IBM Systems Journal* **21,** No. 2, 220–244 (1982).
14. *Document Interchange Architecture: Concepts and Structures*, SC23-0759, IBM Corporation (1983); available through IBM branch offices.
15. R. Cordell, "Rock-solid office architecture," *Computerworld on Communications*, 21–23 (October 3, 1984).
16. *Document Content Architecture: Revisable-Form-Text Reference*, SC23-0758, IBM Corporation (1983); available through IBM branch offices.
17. *Document Content Architecture: Final-Form-Text Reference*, SC23-0757, IBM Corporation (1983); available through IBM branch offices.
18. *SNA Format and Protocol Reference Manual: Management Services*, SC30-3346, IBM Corporation (1986); available through IBM branch offices.
19. A physical unit (PU) is an addressable entity, like the SSCP or LU; one PU exists in each node for local control and to represent the node to a control point (SSCP) in a host processor if one is currently controlling the PU. In a type 2.1 node, a control point (CP) exists to perform local control and to provide required PU services for a remote SSCP.
20. *NPDA General Information Manual*, GC34-2111, IBM Corporation (1985); available through IBM branch offices.
21. The Network Communication Control Facility (NCCF)[22] was also introduced in 1979 to allow a network operator to interact with each such host in a multiple-host network to control testing and data collection and to examine the data.
22. *NCCF General Information Manual*, GC27-0429, IBM Corporation (1985); available through IBM branch offices.
23. *NLDM General Information Manual*, GC27-0657, IBM Corporation (1985); available through IBM branch offices.
24. *Token-Ring Access Method and Physical Layer Specifications*, IEEE Standard 802.5 (ISO/DIS 8802/5), IEEE Computer Society, Los Angeles (1985).
25. D. W. Andrews and G. D. Schultz, "A token-ring architecture for local-area networks—An update," *Proceedings COMPCON Fall '82*, Washington, DC, IEEE Computer Society, Los Angeles (September 1982), pp. 615–624.
26. W. Bux and D. Grillo, "Flow control in local-area networks of interconnected token rings," *IEEE Transactions on Communications* **COM-33**, No. 10, 1058–1066 (October 1985).
27. W. Bux, F. H. Closs, K. Kuemmerle, H. J. Keller, and H. R. Mueller, "Architecture and design of a reliable token-ring network," *IEEE Journal on Selected Areas in Communications* **SAC-1,** No. 5, 756–765 (November 1983).
28. N. C. Strole, "A local communications network based on interconnected token-access rings: A tutorial," *IBM Journal of Research and Development* **27,** No. 5, 481–496 (September 1983).
29. J. H. Saltzer, K. T. Pogran, and D. D. Clark, "Why a ring?" *Computer Networks* **7,** 223–231 (1983).
30. D. E. Carlson, "Bit-oriented data link control," *Computer*

Network Architectures and Protocols, P. E. Green, Jr., Editor, Plenum Press, New York (1982), pp. 111–143.
31. G. A. Deaton, Jr., and R. O. Hippert, Jr., "X.25 and related recommendations in IBM products," *IBM Systems Journal* **22,** Nos. 1/2, 11–29 (1983).
32. *X.25 SNA Guide*, GG24-1568, IBM Corporation (1985); available through IBM branch offices.
33. *XI Original Equipment Manufacturer's Informatiion Manual*, GC11-6065, IBM Corporation (1986); available through IBM branch offices.
34. *The X.25 1984 Interface for Attaching IBM SNA Nodes to Packet-Switched Data Networks—General Information Manual*, GA27-3761, IBM Corporation (1986); available through IBM branch offices.
35. *X.25 SNA Interconnect*, TDCE-XI-685, IBM Corporation (1985); available through IBM branch offices.
36. G. A. Deaton and D. J. Franse, "Performance analysis of computer networks that access satellite links," *ICCC-80 Conference Proceedings*, Atlanta, North-Holland Publishing Co., Amsterdam (1980), pp. 297–302.
37. G. A. Deaton and D. J. Franse, "Analyzing IBM's 3270 performance over satellite links," *Data Communications* **9,** No. 10, 117–132 (October 1980).
38. D. W. Andrews, "Throughput efficiency of logical links over satellite channels," *Proceedings of the Satellite and Computer-Communications Symposium*, Versailles, France (April 27–29, 1983), North-Holland Publishing Co., Amsterdam, pp. 231–241.
39. W. D. Brodd and R. A. Donnan, "Data link control requirements for satellite transmission," *Proceedings of the Satellite and Computer-Communications Symposium*, Versailles, France (April 27–29, 1983), North-Holland Publishing Co., Amsterdam, pp. 201–213.
40. J. D. Day and H. Zimmerman, "The OSI Reference Model," *Proceedings of the IEEE* **71,** No. 12, 1334–1340 (December 1983).
41. S. Shukuya, M. Yamaguchi, and Y. Kobayashi, "A study of OSI subsets from LU 6.2 functional viewpoints," *Proceedings of the 30th Annual Convention, IPS Japan*, Information Processing Society of Japan, Tokyo (March 1985), pp. 1053–1054 (in Japanese).
42. M. Yamaguchi and K. K. Sy, "A comparison of various gateways for SNA-OSI interconnection," *Proceedings of the 30th Annual Convention, IPS Japan*, Information Processing Society of Japan, Tokyo (March 1985), pp. 1055–1056 (in Japanese).
43. M. Shiohara, K. K. Sy, and M. Yamaguchi, "A study of the protocol conversion between SNA LU 6.2 and OSI session/transport layers," *Proceedings of the 30th Annual Convention, IPS Japan*, Information Processing Society of Japan, Tokyo (March 1985), pp. 1057–1058 (in Japanese).
44. J. R. Aschenbrenner, "Open Systems Interconnection," *IBM Systems Journal* **25,** Nos. 3/4, 369–379 (1986).

Robert J. Sundstrom *IBM Communication Products Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Dr. Sundstrom is currently manager of Communication Systems Architecture and the chairman of the SNA Architecture Maintenance Board. He is responsible for the architecture departments that are developing the VTAM-NCP transport network, small-systems networking, and logical unit 6.2 extensions to SNA. He received the B.S. degree in applied mathematics, and the M.S.E. and Ph.D. degrees in computer, information, and control engineering from the University of Michigan, Ann Arbor, in 1970, 1971, and 1974, respectively. He joined IBM in 1974 and initially worked in the area of formal specification of communication architectures. Since that time, he has held development and management positions while contributing to the ongoing evolution of SNA. He is a member of the ACM and the IEEE Computer Society.

James B. Staton III *IBM Communication Products Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Staton joined IBM in 1978 at Research Triangle Park. He is currently on assignment to the European Networking Center in Heidelberg, West Germany, where he is developing prototype systems using OSI protocols. In 1981, Mr. Staton joined the Systems Network Architecture (SNA) group, where he worked on several enhancements to SNA. He initially participated primarily on network management issues involving modems and communication links. Before his current assignment, he managed the Communications Architecture Department, which was responsible for the development of the communication protocols for the IBM Token-Ring Network and its integration into SNA. Prior to that, he helped design and develop Version 2 of the Teleprocessing Network Simulator program product, which simulates networks and devices for performance and reliability testing. Mr. Staton received the A.B. degree in mathematics from Guilford College in 1972, the M.A. degree in mathematics from the University of North Carolina at Greensboro in 1974, and the M.S. degree in computer science from the Ohio State University in 1978.

Gary D. Schultz *IBM Communication Products Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Schultz joined IBM in 1965. His work assignments have included telecommunications access method systems programming; participation in the specification of IBM's binary synchronous communication, along with design and review of its initial software and hardware implementations; development of a time-sharing system used by the University of North Carolina computer science department for a decade; mathematical modeling of buffer storage schemes and ad-tech programming of modem algorithms while in the IBM Research Division; and, since 1973, development of SNA, primarily in terms of its formal definition and external publication. Currently, he is a global reviewer of the architecture and chief technical editor of the growing library of SNA specifications. He has a B.S. in statistics from the University of Chicago and an M.S. in computer science from the University of North Carolina at Chapel Hill.

Matthew L. Hess *IBM Communication Products Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Dr. Hess joined IBM Canada in 1968 as a systems engineer. He specialized in performance analysis and in the design of on-line systems. In 1976, he joined the telecommunications center in La Gaude, France, to study new public data network offerings, and continued this work when he came to Research Triangle Park. For the past seven years, he has participated in the development of Systems Network Architecture, particularly SNI and, more recently, SNADS, as project manager. Dr. Hess received a B.Eng. from McGill University, Montreal, Canada, in 1964. He received an M.Sc. in 1966 and a Ph.D. in 1968 from the University of Birmingham, England.

George A. Deaton, Jr. *IBM Communication Products Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* When Mr. Deaton first began working for IBM, he contributed to manned and unmanned satellite projects done under IBM contracts to NASA. He then became involved in the development of IBM commercial communication products and has since worked with the architecture and design of digital time division systems, in high-speed loop communication systems, and in the development of computer-communication architecture for local-area and wide-area networks. Currently, he is manager of the Network Development Project Office, which has architecture and product coordination responsibilities for attachment interfaces and standards supported by IBM products for public data networks. Mr. Deaton has a B.S. degree in physics from Virginia Polytechnic Institute and has done graduate studies in physics and astrodynamics. He has written numerous papers on computer communication and lectures frequently on that subject.

Leo J. Cole *IBM Communication Products Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Cole received a B.S. degree in 1979 from Syracuse University. He joined IBM immediately after graduation, working in Architecture and Telecommunications in Research Triangle Park. In 1980, he moved to Tampa, Florida, working in Network Design for the IBM Information Network. In 1982, he returned to Architecture and Telecommunications, where he is currently manager of Network Management Architecture.

Robert M. Amy *IBM Communication Products Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Amy joined IBM in 1969 in technical market support, specializing in data communications at the Kalamazoo, Michigan, branch office. In 1974, he accepted an assignment in LSI and microcode simulation for the IBM 4341 development project in Endicott, New York. He became a part of Systems Network Architecture development in Raleigh in 1979 and later managed an architecture development group within that function. Currently, he is manager of Network Standards Development, with responsibility for IBM participation in standards activities related to the lower two layers of OSI, and has been concentrating on ISDN for the last three years. He participates in ANSI T1 and T1D1, as well as CCITT SG XI and SVIII, on the subject of ISDN. Mr. Amy received a B.S. in physics from the University of Michigan in 1966.

Reprint Order No. G321-5285.