SNA Perspective

Volume 12, Number 9
September, 1991
ISSN 0270-7284

The single source, objective monthly newsletter covering IBM's Systems Network Architecture

The SNA Subarea Under Siege

This article explores IBM's approach of evolving today's hierarchical SNA into a peer-oriented architecture based on advanced peer-to-peer networking (APPN). It discusses features of the SNA subarea network and details how those features have or have not been accommodated in APPN.

The Players

While it is commonly said that only two things in life—death and taxes—are certain, we could add another item to the list: the metamorphosis of subarea SNA. Within two years, SNA will look very different from the way it looks today. Three parties are working diligently to make sure that SNA indeed does change from its current rigid, host-centric hierarchical structure to a more peer-oriented network architecture: IBM, multiprotocol router vendors, and SNA customers.

(continued on page 2)

Integrating Two Technologies: Protocol Analysis of SNA on Token Ring

IBM has anointed the token ring as its LAN solution of choice in its SNA strategy. The differences between traditional SNA data links and LANs increase the complexity of day-to-day network operations. Users who integrate token rings into SNA networks need to understand the tools used to perform systems management and to increase administrator productivity. To do this, they need a real-time view of SNA traffic on their token ring networks.

This article looks at how IBM integrated the IEEE 802.5 token ring LAN architecture into its SNA product line and describes how a combination of IBM SNA and token ring hardware and software is used to capture and analyze trace data. It gives insights into the network management methodologies used to troubleshoot problems on a token ring network in an SNA environment.

(continued on page 9)

In This Issue:

(continued from page 1)

IBM

While a completely statically defined network seemed like a good idea in the 1970s, its time has long since passed and IBM is well aware of this. In the earlier days of corporate computing, not only did a company's MIS staff maintain this static network definition, lovingly known as the *sysgen*, but they also maintained the physical network itself. Enhancing the physical network and keeping its logical definition synchronized was manageable since a single group usually oversaw the entire operation and scheduled network downtime accordingly.

The advent of smaller systems into the business computing environment created a problem. Often, these systems were purchased independently of corporate MIS and they commonly had interconnection requirements separate from the corporate network. Small systems may enter and leave the network quite frequently, making the management and control of the network by traditional means too cumbersome. They also had greater intelligence than dumb terminals and thus a need for a higher level of communications independence than subarea SNA provided to any device without an SSCP. The first result of addressing this need was SNA low entry networking, which later evolved into APPN.

> "APPN's technology will be...ending the rule of hierarchical SNA."

IBM found itself with two variants of the SNA architecture: subarea SNA and APPN. Each architecture provided analogous yet incompatible networking services such as network routing, network control, and directory services. IBM has seen that APPN is what customers want and has propagated this new architecture onto more IBM systems. More significantly, IBM has also stated that APPN's technology will be introduced into the SNA subarea within the next few years, presumably ending the rule of hierarchical SNA.

Multiprotocol Router Vendors

The second group attacking the SNA subarea is a gang of multiprotocol router vendors such as Cisco Systems, Wellfleet, Vitalink, CrossComm, and Proteon. These companies are gradually whittling away at the SNA subarea by offering router products with higher performance than IBM's 3745 communications controllers at a significantly lower price. To these vendors, SNA represents the last major frontier. However, none of these vendors yet has products that operate as a replacement for an SNA subarea node. It is possible, however, to replace part of the SNA subarea network with a network of multiprotocol routers and thus reduce dependence on IBM's proprietary static routing protocols.

Customers

The third group wanting to significantly alter the SNA landscape is IBM's customers. They want to preserve their investment in SNA-based applications while being able to take advantage of advances in both SNA and non-SNA network technology. While they might not be enamored with today's method of defining the SNA network, they generally do like the underlying properties of the SNA subarea network. Any architectures or products that are presented to IBM customers and meant to either interoperate with an SNA subarea or replace the subarea entirely must accommodate certain SNA subarea features.

Current State of APPN

Before any comparison between SNA subarea and APPN can proceed, a description of the current status of APPN is in order. APPN first surfaced on the System/36 midrange computers in 1985. When the AS/400 replaced the System/3x, APPN was naturally one of the services that was brought along to the new machine. In terms of product availability, that is where APPN remained until the past few months.

IBM now makes APPN networking services available on two more platforms: the 3174 establishment controller and the Personal System/2 running Operating System/2 (OS/2). The 3174 can only execute in an APPN network as a network node while the OS/2 product, called Networking Services/2 (NS/2), can operate as either an APPN network node or end node. IBM has yet to make APPN networking services available on any of the subarea nodes, such as S/3x0 host systems or 37x5 communications controllers.

IBM agreed to make APPN end node specifications available to the public so that IBM-compatible vendors could interface devices to APPN as end nodes. At the time of the announcement, four companies were committed to development of APPN end node capabilities in their products—Novell, Siemens, Apple Computer, and Systems Strategies—and several others have added themselves to the list since then. However, due in part to the adverse publicity about not revealing the APPN network node specifications, IBM has announced that it is reconsidering the feasibility of preparing the network node specification for public availability.

Since only the APPN end node specifications are available for this issue of *SNA Perspective* (see sidebar "APPN End Node Specifications"), this analysis between SNA subarea and APPN is from the perspective of APPN networking services available via an end node. While it is the APPN network node that will eventually replace the SNA subarea's node type 4 (previously called PU 4) implementation, the APPN end node specification still yields a lot of information about the operation of APPN as a whole.

APPN End Node Specifications

Most people are not aware that the APPN end node specifications are now publicly available from IBM. While the four companies listed in this article as having committed to APPN end node at the time of the announcement got early copies of the end node specifications from IBM, the same material can now be ordered as a regular IBM publication through the usual channels.

The packaging of the APPN end node specification is somewhat curious. The specification is contained in a manual entitled *Systems Network Architecture: Type 2.1 Node Reference,* IBM publication number SC30-3422-2. The title of this manual should look familiar—it is the same title as the little 95-page node type 2.1 specification from IBM back in 1988. The IBM publication number for this first accurate description of the operation of a type 2.1 node was SC30-3422-1. The "update" to this manual in the form of the APPN end node specification is almost 800 pages!

If you're considering ordering this new manual, don't throw the "-1" version out. It still contains the clearest description of a type 2.1 node operating as an APPN LEN node that can be found anywhere. The relatively simple operation of an LEN Node was completely lost in the complexity of the end node specification of the "-2" version of the manual.

If this trend continues, the APPN Network Node specification can be expected to called *SNA: Type 2.1 Node Reference,* IBM document number SC30-3422-3, and would be 6400 pages long! For APPN's sake, *SNA Perspective* hopes this won't be so.

The contents of SC30-3422-2 are:

Part 1. Introduction and Overview
  Chapter 1. APPN Overview
  Chapter 2. Overview of the Node Structure
  Chapter 3. Node Operator Facility
Part 2. Control Point
  Chapter 4. Overview of the Control Point
  Chapter 5. Session Services
  Chapter 6. Address Space Manager
  Chapter 7. Topology and Routing Services
  Chapter 8. Directory Services
  Chapter 9. Configuration Services
Part 3. Routing
  Chapter 10. Path Control

Subarea SNA versus APPN

Differences between traditional subarea SNA and the newer peer-oriented APPN are considerable. We will consider and compare them in several areas:

- compatibility
- configurability
- static versus "dynamic" routing
- scalability
- throughput and reliability
- data transport efficiency

Compatibility

IBM has taken a lot of heat for abandoning older logical unit (LU) and physical unit types in APPN. While the node type 2.1 architecture that was published in 1986 did abandon node type 2.0 (previously called physical unit 2) addressing, IBM rectified the problem in the next revision of the node type 2.1 architecture that was published in 1988 by bringing "SSCP-dependent" devices back into the fold. The address assignments that were added in 1988 for dependent LUs still hold in today's APPN end node Specification. (See *SNA Perspective,* January 1990, *Breaking the Chains of Hierarchical Networking: Integrating Node Type 2.1,* for a complete discussion of dependent LUs.)

However, *end-user* services (provided by LUs in SNA) available in APPN are still misaligned with the majority of existing applications. This doesn't present an obstacle for businesses creating brand new applications for a distributed network—if they are using SNA, they are most assuredly using LU 6.2. It does present a significant problem, however, for businesses whose existing applications are based upon either LU 2 or LU 0 since APPN permits only LU 6.2 session traffic to flow within an APPN network.

Given that LU 6.2 is the strategic, converged LU for SNA, *SNA Perspective* expects LU 0, 1, 2, and 3 applications to be somehow accommodated within LU 6.2. What form this accommodation will take remains to be seen. Without this non-LU 6.2 to LU 6.2 "glue," APPN will find limited success in an SNA subarea. Today's SNA subarea network supports applications that use the end-user services provided by LU 0, 1, 2, and 3 as well as both variants of LU 6.2—dependent and the new independent LUs. Anything less than full SNA application support will meet with much resistance from IBM's customers.

Configurability

Definition of network resources is undoubtedly the one area of APPN that differs the most from traditional subarea SNA. Maintaining the sysgen for a large host-centered SNA network is a massive task that requires highly skilled, network-savvy systems programmers. There are host applications that assist in the process of defining the network configuration. One such tool is NETDA which takes a definition of the routes between subarea nodes and constructs VTAM and NCP PATH definition statements.

One of the primary initial design goals of APPN was ease of configuration. Each node in an APPN network contains a database of configuration information that defines what resources are available on the local node. Each APPN node contains a Node Operator Facility that permits the definition of local resources such as local LUs, control point names, etc. The latest manifestation of APPN—Networking Services/2 for OS/2—makes the job of local resource configuration even easier. There are several predefined local resources that will suffice for all but the most complex LU 6.2 applications.

The definition of remote resources that are resident on other APPN nodes in the network is left to the runtime components of the APPN architecture, specifically the Directory Services subcomponent of the node type 2.1 control point. This topic is discussed in more detail under "Scalability," below.

Static versus "Dynamic" Routing

Fortunately, the SNA subarea network is no longer the static, unchangeable monolith it was prior to VTAM 3.2/NCP 5.2. It is still static—but it can be changed without taking down the entire network. The Dynamic Path Update (DPU) facility permits the user (typically a systems programmer) to alter the contents of a PATH statement in the sysgen and dynamically download the new routing information to the affected subarea node. The download to a target subarea node does not disrupt any other subarea nodes nor does it affect any other paths through the affected subarea.

The ability to dynamically update the routing information in a subarea node is quite an advance over pre-VTAM 3.2 systems. However, note that the inclusion or exclusion of an SNA node that is visible from the subarea network still requires human intervention. Once the new routing tables are established following the DPU, SNA subarea routing is still purely static.

APPN has been touted as providing dynamic SNA routing. This is partially true—it is dynamic but not to the extent that many people believe. Dynamic routing in the Internet community, which is based on TCP/IP, is generally used to describe routing decisions that are made at the packet level. Routing decisions in an APPN network are made at the data flow control layer (roughly equivalent to the session layer in the OSI model) at the time the session is created (i.e., at BIND time) and remain in effect for the life of the session. While the Internet applications are built on a foundation of connectionless datagram protocols, SNA applications use connection-oriented network services.

The routing information that is placed into an APPN BIND image employs a technique similar to that found in OSI layer 2 bridging—a route selection vector is built containing consecutive hop information by the APPN network node control point serving the originator through communication with other network nodes. The completed route selection vector details which APPN network nodes must be traversed in order to complete the path from the source node to the destination node. Once this routing information is established at BIND time, all request/response units that flow on a session follow the same route.

Scalability

The one major problem with increasing the size of traditional SNA subarea networks is the increased complexity of the static definition. The addition of new nodes into a subarea network can increase the complexity of the sysgen exponentially depending on the location of the new nodes in the subarea. One benefit of subarea SNA, however, is that once the sysgen is in place, the overhead traffic flow through the subarea nodes increases linearly at most.

One of the most common fears cited when APPN is mentioned as a subarea architecture replacement is the scalability of APPN. As new nodes are inserted and removed from the APPN network independently of one another, the resulting alteration in the layout of an APPN network (either through link activations or link failures) results in the automatic propagation of topology database updates (TDUs) throughout the entire APPN network. Adding a subarea node in today's hierarchical SNA is a relatively infrequent event. But one of APPN's essential principles is peer networking, thus placing small systems on equal parity with 3745s. Carried to the extreme, this could imply that users powering on and off their PS/2s running Networking Services/2 software and operating as APPN network nodes would cause TDUs to flow across an entire APPN network. Of course, nodes likely to be frequently powered on and off would probably be configured as end nodes rather than network nodes.

There are ways around TDU "storms" caused by frequent changes in the state of network. The Internet community encountered the problem of scaling routing information updates quite some time ago. The Routing Information Protocol (RIP) used in XNS and in early TCP/IP implementations fell apart when the size of the network increased. The solution to this problem of scalability was today's Open Shortest Path First (OSPF) algorithm. Whether the APPN network node will contain optimizations to support large-scale APPN networks is only a matter of conjecture until the APPN network node specification is made available.

Even though there is potential for performance degradation due to changes in the state of the network, other APPN administrative exchanges occur with much greater regularity than TDUs. Since the amount of static configuration information has been greatly reduced in APPN, the Directory Services subcomponent must determine the exact location of remote resources on the network whenever those resources are needed by local applications.

There are optimizations in the design of APPN that reduce the amount of network traffic required to locate a remote resource. Two such optimizations are:

- support of a local directory containing the static definition of local resources and certain remote resources
- support of a local cache of recently located remote resources

Resources found in the local directory obviously do not result in any APPN administrative network traffic. However, even though remote resource information might be found in a local cache, the APPN node still needs to send a "directed" search to verify that the information found in the local cache is still accurate. The directed search goes directly to the destination node by traversing the APPN network according the known topology.

Worse yet could be "undirected" searches. The ease of configuration of APPN networks doesn't come without a price—and undirected searches are part of that price. If the remote resource information cannot be found in either the local directory or the local cache, the information must be located in the database that is distributed throughout the APPN network. These undirected searches are analogous to the broadcasts of the Internet world.

Typically, the transient loading and unloading of applications in a large APPN network has no impact on the network overhead. In the best case, the application's LU already has unused sessions to the application's targets, bound the last time the application ran or when some other application needed a session to the same target. In most of the remaining cases, the target location has been cached and the network needs only a directed search to verify that the target has not moved; the undirected search is avoided.

Undirected searches would only create significant overhead on the network if the commonly used target LUs were moved frequently. The current trend toward client-server computing may eliminate this danger.

In client-server computing, many client programs request service from a few server programs. In most cases, clients initiate the dialog to servers; servers rarely initiate the dialog. The LU supporting client programs is much more likely to move than an LU supporting server programs because a client workstation is typically on the fringe of the network where change is the norm and server nodes are typically in stable areas where many clients can be serviced effectively.

When a client moves, an undirected search is rarely caused because the server very rarely initiates the dialog to a client. On the rare occasion when a server moves, an undirected search will be generated when a network node discovers that its cached data is invalid, but once the cache is corrected, directed searches are used again.

Because today's SNA subarea network is statically defined, there is no need to dynamically locate remote resource information. However, since it is is possible for remote resources to reside in different host domains, there can be significant cross-domain traffic in order to establish SNA sessions. With APPN in the subarea, this cross-domain traffic will be replaced with APPN's Directory Service flows.

Throughput and Reliability

IBM implemented transmission groups in subarea SNA in the late 1970s in order to support multiple links between adjacent subarea nodes. Transmission group control (TGC) was a sublayer within the path control layer for type 4 and 5 nodes. Transmission groups provided three major benefits to subarea SNA:

- greater throughput between adjacent subarea nodes
- greater tolerance of transmission errors
- greater tolerance of link outages

The multiple physical connections that comprised a transmission group were treated by TGC as a single logical connection. A single path information unit (PIU) presented to the path control layer could result in several SNA basic link units (BLUs) being sent simultaneously over multiple parallel paths. TGC assumed responsibility for maintaining the proper sequence of BLUs that made up the whole PIU. This feature was very important when the speed of data transmission lines was significantly less than today. With the megabit and multiple-megabit data transfer rates commonly available now, however, using transmission groups as a technique to increase overall subarea throughput is much less important.

Another feature inherent in the operation of transmission groups is tolerance for adverse link conditions. If a transmission group consists of four actual data links and one of those links becomes disconnected, TGC will then use the remaining three links to transfer data between adjacent subareas without session disruption. In addition, if one of the links in a transmission group begins experiencing errors in transmission, the sending TGC attempts to minimize the effect of data retransmission by retransmitting the BLUs over another link.

Transmission groups also exist in APPN (or, more accurately, in a type 2.1 node) but in a greatly simplified form. A transmission group in node type 2.1 is equivalent to a link between two APPN nodes. While multiple transmission groups are permitted in APPN network node-to-network node and end node-to-network node connections, multiple links within a transmission group are not supported. As a result, the subarea feature of using transmission groups to increase effective throughput and to be more resilient to adverse link conditions does not exist with APPN.

Although the APPN transmission group details described here are a part of the node type 2.1 portion of the APPN end node specification, presumably the APPN network node also supports only single link transmission groups. This will not be verified until the APPN network node specification becomes available.

As mentioned, the effective throughput benefit of subarea transmission groups may have outlived its utility. However, if network managers take advantage of the link outage handling within today's subarea transmission groups to achieve high network availability to end users, then they should look closely at APPN once it becomes available in the subarea to see if the same level of reliability can be attained with APPN.

Data Transport Efficiency

All of the preceding topics covering the differences between subarea SNA and APPN SNA are involved with network administration. This last section discusses what happens as data flows through the two different types of networks and where the efficiencies lie.

Data flow in subarea SNA that is destined for a peripheral node typically occurs in two stages:

- from the host node to the subarea node containing the boundary function (usually called the boundary node)
- from the boundary node to the peripheral node

These two distinct stages give the SNA subarea data flow control technique its name—two-stage pacing. This second stage of the two-stage pacing is also the primary reason for more SNA layers being implemented in the boundary node NCP. The layers of SNA that exist in subarea SNA networks are shown in Figure 1. Note that the intermediate subarea node contains only path control while the boundary subarea node contains path control and transmission control.

The APPN pacing mechanism is quite different. Since several intermediate APPN nodes participate in

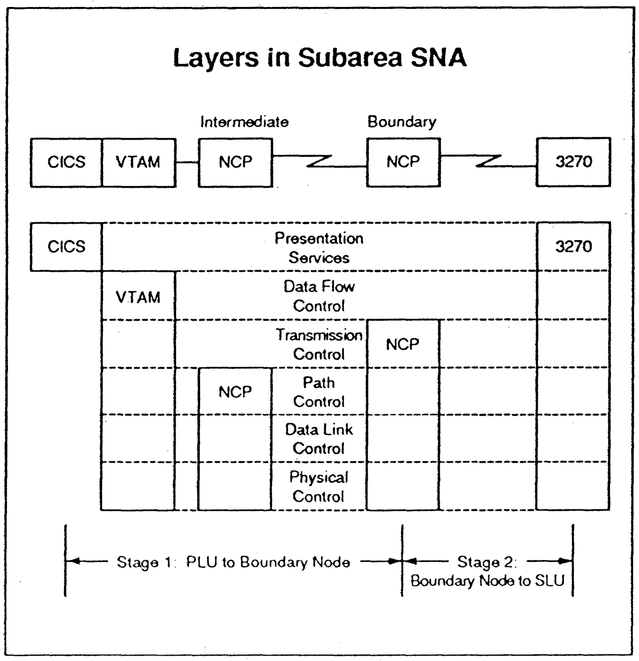


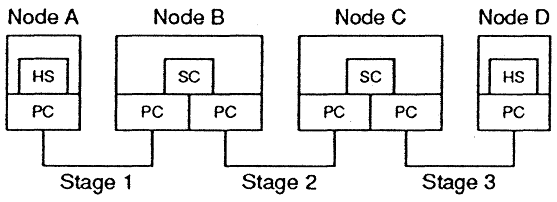
Figure 1

the end-to-end transfer of data, each hop has its own window size. Furthermore, unlike subarea SNA, the pacing window sizes can vary in APPN. This relatively new mechanism of variable window sizes in SNA is termed *adaptive session-level pacing*.

The window sizes used in the two-stage pacing of subarea SNA are fixed at session activation time (i.e., in the BIND image) and do not vary for the life of the session. APPN, on the contrary, uses variable window sizes between adjacent pairs of APPN nodes with the size of the window increasing or decreasing depending on the availability of buffers in those intermediate nodes. The agent within the APPN nodes that determines the window sizes differs according to the node. As seen in Figure 2, the nodes at the terminus of the end-to-end exchange each contain an LU 6.2 half session that governs the pacing window size between the APPN end nodes and the adjacent intermediate node. Conversely, the intermediate nodes contain a session connector that controls the window size between adjacent intermediate nodes.

Even though this multistage pacing seems much more complicated than the rather simple two-stage pacing above and more layers of SNA participate in the intermediate nodes of APPN, the overall effect will usually be greater throughput. Today's manually configured, fixed window sizes are usually assigned conservative values based on the systems programmer's best guess of buffer availability.

Adaptive Session-Level Pacing in APPN



- Pacing window of each stage is independent
- Pacing window of each stage varies depending upon buffer availability of within each node

HS LU 6.2 Half Session (consisting of TC and DFC)
PC Path Control
SC Session Connector

Figure 2

Adaptive session-level pacing, on the other hand, frees network managers from having to worry about pacing window sizes. The APPN nodes will accommodate the largest window size possible under continually changing load conditions and will dynamically adjust those windows as conditions change.

Summary

We have discussed several network characteristics that exist in the subarea implementation of SNA and described how those characteristics map onto APPN services. In some cases, APPN is superior while in others SNA subarea networking is better.

SNA subarea excels in:

- Network availability—SNA subarea's transmission groups permit outages on individual links without sacrificing subarea node connectivity.
- End-user protocol support—All LU types used by today's SNA applications can be routed through an SNA subarea network.
- Runtime impact of a large network—While creating the sysgen for a large SNA subarea network might be a nightmare, once the network is in place and running adding new nodes has minimal impact on its performance.

APPN excels in:

- Resilience to change—APPN nodes can be added and removed and the network will adjust automatically to the changes in topology. Resources on APPN nodes can be altered and the change can be discovered dynamically.
- Data transfer efficiency—The adaptive session-level pacing used between APPN nodes will result in greater throughput if load conditions permit.
- Routing—Even though APPN doesn't implement truly dynamic routing, it is much more dynamic than the purely static routing found in SNA subareas. ■

(continued from page 1)

IBM's LAN Strategy Circa 1984

It's been nearly seven years since IBM first announced the LAN interconnection of its personal computer product line using the broadband-based PC Network in September 1984. PC Network and its companion PC Network Program provided basic file, print, and message capabilities. Later that same year, IBM introduced SNA 3270 Emulation Services which provided PCs with 3270 SDLC communication to SNA host applications. In addition, IBM provided the NetBIOS communications interface for personal computer application programs.

In October 1985, at the unveiling of the IBM Token-Ring, IBM gave a statement of direction of its intent to support interconnection between the IBM Token-Ring Network and the PC Network (using the IBM Cabling System), and the IBM Token-Ring Network and the IBM Industrial Network.

Many advances have been made in token ring LAN technology over the years since IBM's initial announcement. Even before its announcement, the token ring was a strategic IBM architecture. IBM has developed a set of products that address the need for flexible communication among mainframes, minicomputers, and workstations in a business information system.

Token Ring Concepts

In February 1980, the IEEE began to draft standards for LANs and developed a model corresponding to the lower two layers of the ISO's open systems interconnection (OSI) model. However, the IEEE model divided the OSI data link layer into two sublayers. Figure 3 shows the relationship between the two lower layers of SNA and the OSI model and the IEEE 802.5 (token ring) and 802.2 standards.

As shown, the token ring architecture spans both the data link and physical layers. The data link layer is divided into two sublayers—logical link control (LLC) and medium access control (MAC). The LLC layer defines procedures for establishing, maintaining, and terminating logical links between devices on the LAN. Equally important, the LLC control procedures provide for reliable exchange of data between nodes on the LAN. The MAC sublayer provides procedures to control access to a shared transmission medium, and controls the routing of information between the physical layer and the LLC sublayer. When an SNA- or OSI-based device is connected to a token ring LAN, the two lower layers of the SNA architecture and OSI model are implemented by IEEE 802.2 and 802.5 token ring protocol standards.

VTAM's View of the Token Ring

In order to reduce the impact of new technologies on the maintenance and support of existing SNA networks, IBM tends to use existing product functionality to incorporate new technologies. When a token ring is connected to an IBM SNA network, the network's physical configuration uses the hardware and software components shown in Figure 4.

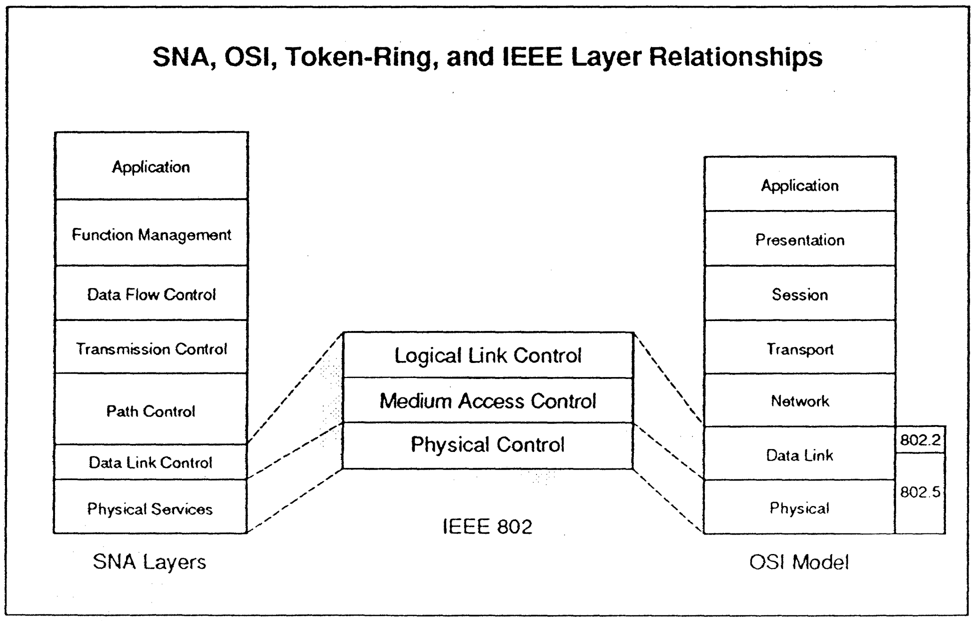


Figure 3

The key component providing token ring connectivity into the SNA environment is IBM's NCP Token-Ring Interface (NTRI) software. IBM had three objectives in developing NTRI:

- to map token ring protocols to SNA protocols
- to provide single connection of a communications controller to a token ring network
- to not impact applications

The purpose of NTRI is to map token ring protocols to SNA protocols and provide for data exchange between SNA subarea networks and token ring networks. A primary design objective for NTRI was to give users a way to attach a communications controller running NCP to a token ring network with a single physical medium, a single attachment scheme, and a single communication protocol. In addition, IBM developed NTRI to support the token ring without impacting the user's data processing application environment. In fact, IBM used the same implementation methods for NTRI as it did to support NCP X.25 connectivity with the NCP packet switching interface (NPSI) software. For instance, X.25 devices are represented in NPSI as node type 1 (previously called PU 1) terminal nodes. IBM uses the same methods for the token ring in NTRI.

Notice in Figure 5 that the 37x5's TIC is *represented* to VTAM as a traditional physical unit type 1 terminal node connected to a full-duplex (FDX) point-to-point link. To further clarify the illustration in Figure 5, it is important to note that the type 1 node is represented as a physical unit software control structure generated in NTRI. Also, the node type 2 devices connected to the ring are represented to VTAM as switched resources connected to a half-duplex point-to-point SDLC link. Simply stated, Figure 5 is VTAM's logical view of the physical connections in Figure 4. Therefore, IBM's implementations using traditional NCP features make the NTRI transparent to VTAM and the user's host-based subsystems.

Logical Link Control (LLC) and SNA Higher Layers

IBM's Token-Ring Network implements the single-byte service access points (SAPs), which are architected in the token ring specification, as code points in which application programs can be defined to the LLC software implementation. Simply stated, SAPs provide the interface between SNA's higher communication layers and the LLC. See Figure 6.

It is beyond the scope of this article to decode all control information in the IEEE's LLC frame format. However, we will look at the LLC information fields containing destination and source SAP entries. Control information and data exchanged between LLC sublayers in individual nodes conform to a common frame format called the LLC protocol data unit (LPDU) as shown in Figure 7 on page 12.

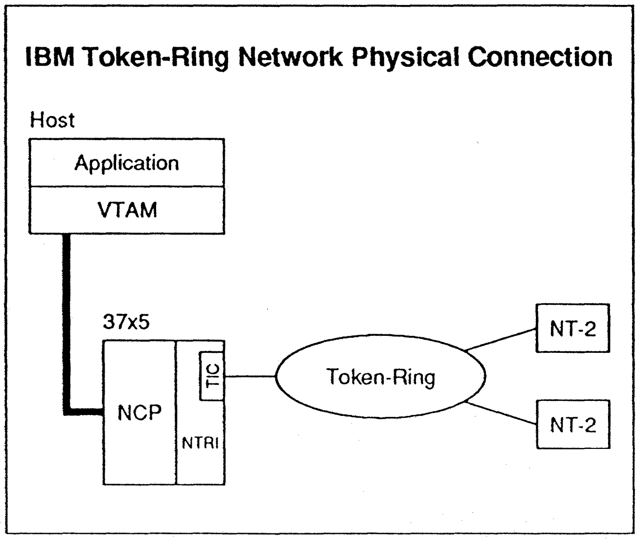


Figure 4

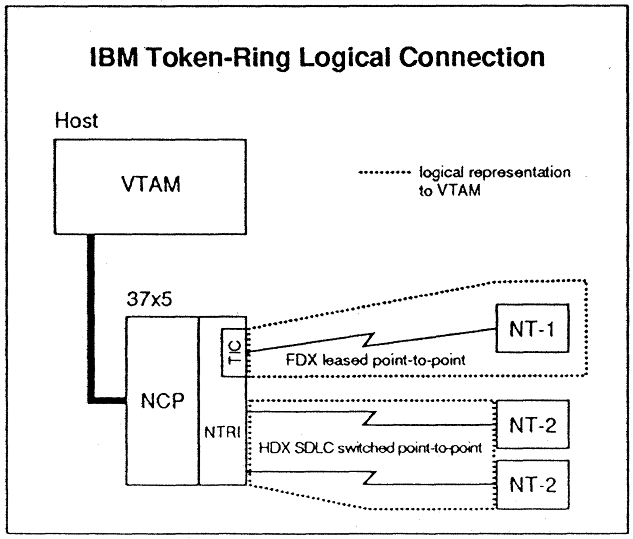

IBM Token-Ring Logical Connection
Host
VTAM
logical representation to VTAM
37x5
NCP
TIC
NTRI
FDX leased point-to-point
NT-1
HDX SDLC switched point-to-point
NT-2
NT-2

Figure 5

The source SAP (SSAP) identifies the SAP that originated the LPDU. The destination SAP (DSAP) identifies the SAP code point—the intended destination for the LPDU. The data units exchanged between SNA's path control layer and the LLC are the standard SNA path information units (PIUs). Several commonly used SAPs are listed below:

- SAP X'00' This is the null SAP that gives a node the ability to respond to another node before its SAP has been activated. The null SAP supports connectionless service and is only used for LLC TEXT and XID command LPDUs.
- SAP X'04' This is SNA's path control SAP. It is the SAP used for SNA nodes.
- SAP X'08' This SAP is used by the IBM 3270 Workstation Program.
- SAP X'F0' This SAP is used by NetBIOS.
- SAP X'F4' This SAP is used by the IBM network management functions of the IBM LAN Manager.

Token Ring Addressing

As shown in Figure 7, the token ring destination address (DA) and source address (SA) identify the stations connected to the ring. In fact, every station on the ring has a token ring adapter and every adapter has six-byte address that must be unique on the ring. In addition, the address must be unique among all bridged LANs accessible from that LAN.

Universal Adapter Address

In fact, it is the responsibility of the adapter manufacturer to assign a universal adapter address (UAA) as part of the adapter hardware. The UAA is also called the burned-in address and is administered by the IEEE. The IEEE guarantees that all addresses will be unique. Furthermore, each token ring manufacturer is assigned a range of addresses by the IEEE. For example, the UAA of IBM's token ring adapters would be X'10005Axxxxxx'. An example of a specific IBM universal adapter address would be X'10005A002007'.

Locally Administered Address

On the other hand, a locally administered address (LAA) is assigned by a non-IEEE authority such as a LAN administrator within a user's organization. LAAs are recommended to be in the format

X'4000*abbbbbbb*'

In this format, *a* is in the range of 0 through 7, and *b* is in the range of 0 through F. The meaning of byte zero (the leftmost byte, "40" in the example, is byte zero) has a different meaning in a destination address (DA) and a source address (SA). In both cases, bit 1 of byte 0, called the U/L bit, indicates whether the address is a UAA (B'0') or an LAA (B'1'). In a DA, bit 0 of byte 0 indicates whether the DA is an individual address (B'0') or a group address (B'1'). In an SA, bit 0 of byte 0 is used for the routing information indicator (RII) which is set to binary one (B'1') when the frame contains a routing information field and to binary zero (B'0') when no routing information field is present. As shown in the sidebar example on pages 14 and 15, the DA is an LAA and is an individual address, so byte zero is 40 (0100 0000), while the SA is an LAA and the frame contains a routing information field, so byte zero is C0 (1100 0000).

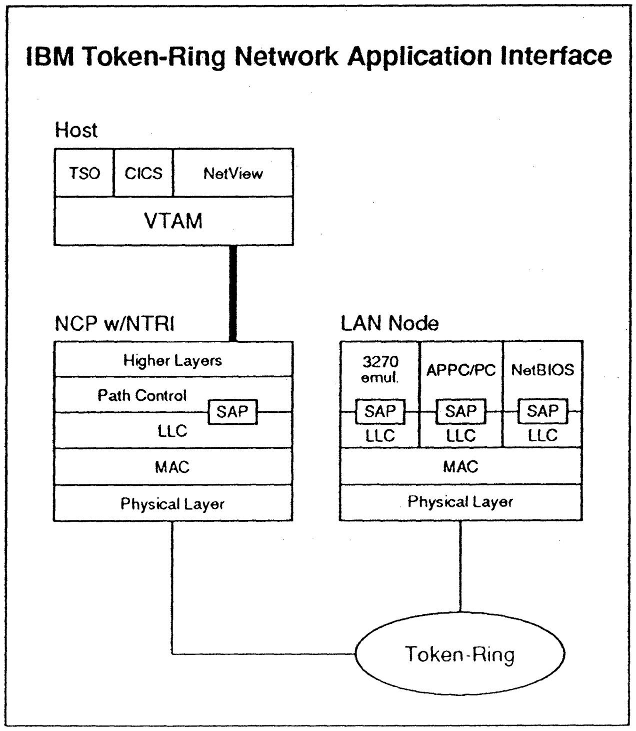


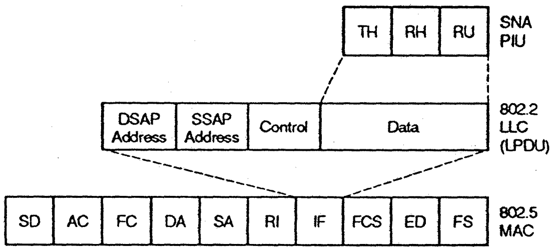
Figure 6

Source Routing Method

Before we introduce our sample trace, a brief discussion of token ring source routing will help clarify several of the initial frame flows. (Note that this article does not intend to debate the virtues or shortcomings of source routing.) The source routing method describes how frames are routed over a token ring network that consists of multiple interconnected rings. A route is the path from the originating station to the destination station and, since each frame contains information about its route, a centralized routing table is not required. The routing information field in the token ring header has information that determines the route the frame will travel. There are two ways in which source routing can be implemented:

- on-ring determination
- off-ring determination

LLC Protocol Data Unit



SD	Starting Delimiter	1 byte
AC	Access Control Field	1 byte
FC	Frame Control Field	1 byte
DA	Destination Address	6 bytes
SA	Source Address	6 bytes
RI	Routing Information	0-18 bytes
IF	Information Field	Variable length
FCS	Frame Check Sequence	4 bytes
ED	Ending Delimiter	1 byte
FS	Frame Status	1 byte
DSAP	Destination Service Access Point	1 byte
SSAP	Source Service Access Point	1 byte
Control	Control Field	1 or 2 bytes
TH	Transmission Header	26 or 6 bytes
RH	Request/Response Header	3 bytes
RU	Request/Response Unit	Variable length

Figure 7

On-ring determination

When an originating station initiates a logical link connection, it issues a TEST or XID (exchange identification) command on its local ring. If an acknowledgment is not received, the destination station is not on the ring. The on-ring method is accomplished when an originating station issues a non-broadcast frame without route designator fields present in the token header.

Off-ring determination

In the off-ring method, similar to the on-ring method, the originating station issues a logical link TEST or XID command to all rings with the all-routes broadcast. The difference in the off-ring method is that the frame fans out; that is, multiple copies are created through all active bridges, searching for the token ring adapter DA and accumulating routing information as the frame copies pass through the bridges.

If multiple routes exist to the destination, multiple TEXT or XID frames will reach the destination station. Next, the destination station processes each logical link TEST or XID command and returns the acquired routing information to the originating station in TEST or XID acknowledgment frame. The acknowledgment frames are sent as non-broadcast frames.

When the returning acknowledgments indicate multiple routes to the destination, the originating station uses the route of the first non-broadcast frame to arrive as the preferred route for all transmissions to that destination station. This ensures that the fastest route at the time of session establishment is used. However, although this is the fastest route, it may not be shortest.

Let the Traces Begin!

We are now ready to look at a sample trace for an understanding of the flows used by IBM's Token-Ring products. We must note that the flows depicted in the sidebar only show the adapter addresses and SAP. Equally important, the trace flows can be captured by a variety of IBM and third-party vendor trace and performance monitors.

The IBM Trace Facility (TRACE) is provided by the IBM Token-Ring Network Trace and Performance PC Adapter II for standard BIOS PCs and PS/2s, or for the IBM Token-Ring Network Trace and Performance Adapter A for Micro Channel Architecture PS/2s. In addition, TRACE has the ability to capture all or a part of frames on the ring. The trace analysis facility, called RTAP, is used to analyze data collected by the trace facility (TRACE), and provides levels of analysis ranging from an overview of frames to a bit-by-bit detail of each frame captured on the ring. For example, individual data perspectives include, LLC, MAC, NetBIOS, and SNA. In our example, we examine it from both the LLC and SNA perspective.

Trace Analysis Methodology: Where to Begin

Given the sample trace flows in the sidebar on pages 14 and 15, we can easily imagine the impact of a multiprotocol networking environment on network administrators. In fact, network support personnel are often required to troubleshoot problems that include all layers from the physical through the application layer. Although a complete examination of the proven network management methodologies are beyond the scope of this article, a few basic and useful guidelines can be mentioned.

People responsible for all or part of the network must understand the tools available for trace and performance analysis. One primary recommendation made for using any network trace tool is to capture a portfolio of "normal" trace flows that can be compared against any "failing" traces. Nevertheless, we can normally expect problems to occur where no "normal" traces exist. IBM has addressed this situation by providing sample trace data flows for most of its hardware and software product lines.

The first step in problem analysis is to isolate the failing components in the network. Although this step seems obvious, it actually requires an understanding of the applications and their program interfaces. Let us use an example of an LLC connection that could not be made between a PS/2 running OS/2 EE and a 3174 01L Token-Ring Gateway. The first step would be to isolate the 3270 emulation component of the OS/2 Communications Manager and not consider any other application traffic flows. Equally important, this isolation process reduces the amount of trace data to be interpreted.

Another recommendation is to identify the token ring network addresses of the nodes involved. This means that, instead of tracing all traffic on a particular ring, you would only trace the frames between either locally or universally administered addresses.

Finally, the IBM Token-Ring Network Trace and Performance Program can be used to analyze the end-to-end application data flows contained in the SNA PIUs. This means that any SNA LU session type can be traced and analyzed. However, the format and content of the SNA commands and application data (i.e., 3270 data stream compatibility or APPC LU 6.2 structured fields) must be interpreted by the user or application programmer

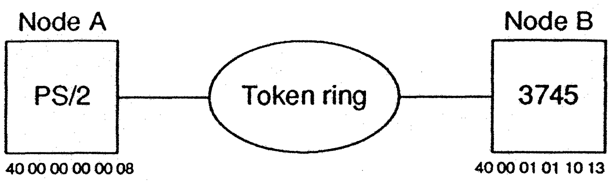
Summary

This article examined how IBM integrated the IEEE 802.5 token ring LAN architecture into its SNA product line and reviewed the key components of the token ring architecture to provide an understanding of LAN and SNA protocol analysis using trace and performance tools from IBM or third parties. Today, network managers and administrators within corporate MIS departments contend with an increasingly complex set of user applications and data networking technologies. In fact, IBM's major SNA users will continue to demand network management and performance tools that meet tough price/performance standards for the remainder of the 1990s.

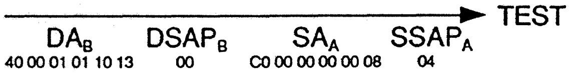
(continued on page 16)

Sample Trace Flows

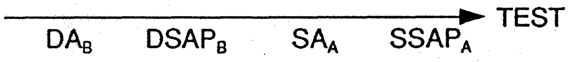
In our sample trace, we will show the typical token ring data flows between an IBM 37x5 Token-Ring Interface Coupler (TIC) Gateway and a PS/2 running IBM's OS/2 EE. The address of the 37x5 TIC is X'400001011013'. The address of the PS/2 token ring adapter is X'400000000008'. Communication is across the token ring between the 37x5 TIC and the PS/2. Finally, for simplicity and reduced clutter in our illustrations, the sample trace only shows the adapter destination address (DA) and destination service access point (DSAP), plus the adapter source address (SA) and source service access point (SSAP) values.



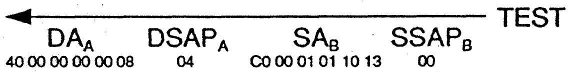
1. The first series of flows show LLC TEST frames sent by the PS/2 (using SAP "04" for SNA path control) to SAP "00" (a null SAP) of the destination address of the 37x5 TIC. This LLC TEST frame is sent as a non-Broadcast frame to locate the 37x5 TIC on the ring. (See page 11 for an explanation of these locally administered addresses.)



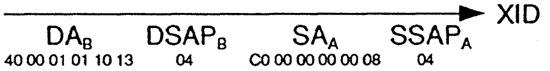
2. This LLC TEST frame is sent by the PS/2 to the 37x5. This frame is sent as an all-routes broadcast frame to locate the 37x5 TIC. The difference between this LLC TEST frame and the one shown in flow 1, is that this frame will cross a bridge on the LAN.



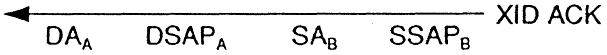
3. This is an echo back frame sent by the 37x5 TIC to the PS/2. This frame is sent as a non-broadcast frame that contains a specific route created by the token ring architecture's source routing algorithm. This route was the one that was used in the all-routes broadcast frame from flow 2 to locate the 37x5's TIC. In addition, this route will be the one used for all subsequent traffic between the PS/2 and the 37x5 TIC.



4. This is an LLC exchange identification (XID) frame from the PS/2 to the 37x5 TIC. This frame is used to signal the NCP to allocate a LINE/PU pair that was previously generated by the NCP ECLTYPE=LOGICAL group macro. (The 37x5 TIC's SAP has changed from '00' to '04' indicating SNA's path control SAP, as described on page 11.)



5. This frame indicates that the NCP/NTRI has successfully allocated the LINE/PU pair requested in flow 4.



6. This frame is an SDLC Format 3 XID from OS/2 EE to the destination address of the 37x5 TIC. This frame would contain the standard IDBLK (05D) and the IDNUM (17269) that is forwarded to VTAM for activation of the PU/LU pair associated with the PS/2. Also, receipt of this frame instructs the NCP/NTRI to initiate a link station connection with the PS/2.

——————→ XID
DA_B $DSAP_B$ SA_A $SSAP_A$

7. This frame begins the initiation of data transfer in the extended asynchronous balanced mode of operation with a remote link station. The 37x5 TIC is requesting a link connection with the PS/2.

←—————— SABME
DA_A $DSAP_A$ SA_B $SSAP_B$

8. This frame is the acknowledgment to the SABME. A link station connection now exists between the 37x5 TIC and the PS/2.

——————→ UA
DA_B $DSAP_B$ SA_A $SSAP_A$

9. This frame contains the activate physical unit (ACTPU) request PIU sent by VTAM. The entire PIU can be examined in detail by requesting a detailed TRACE view. It is important to note here that application level data flows can be analyzed in addition to token ring and SNA session flows.

←—————— ACTPU REQ
DA_A $DSAP_A$ SA_B $SSAP_B$

10. This frame shows the LLC acknowledgment (receiver ready or RR) of the ACTPU request from the PS/2 to the 37x5 TIC.

——————→ RR
DA_B $DSAP_B$ SA_A $SSAP_A$

11. This is the ACTPU response from the PS/2. If this request had been rejected, sense data would be included in the response unit (RU) of the PIU.

——————→ ACTPU RSP
DA_B $DSAP_B$ SA_A $SSAP_A$

12. This frame shows the LLC acknowledgment from the 37x5 TIC to the PS/2 of the frame in flow 11.

←—————— RR
DA_A $DSAP_A$ SA_B $SSAP_B$

13. This frame contains the activate logical unit (ACTLU) request command for the PS/2.

←—————— ACTLU REQ
DA_A $DSAP_A$ SA_B $SSAP_B$

14. This frame shows the acknowledgment from the PS/2 to the 37x5 TIC confirming receipt of the frame in flow 13.

——————→ RR
DA_B $DSAP_B$ SA_A $SSAP_A$

15. This frame shows the ACTLU response PIU from the PS/2 to the 37x5 TIC. If this request was rejected, sense data would be included in the response unit (RU).

——————→ ACTLU RSP
DA_B $DSAP_B$ SA_A $SSAP_A$

16. This frame shows the acknowledgment from the 37x5 TIC to the PS/2 confirming receipt of the frame in flow 15.

←—————— RR
DA_A $DSAP_A$ SA_B $SSAP_B$

(continued from page 13)

In this article, our sample trace implied an IBM Token-Ring network spanning multiple cities, states, and countries that may integrate a wide range of complex systems including mainframes, bridges, routers, gateways, terminals, workstations, personal computers, modems and multiplexers. Therefore network managers and administrators need to be skilled in a large number of data communication and application protocols.

Major data networking vendors must continue to provide the necessary analysis and performance tools allowing their customers to manage multivendor, multiprotocol global data networks that support the many of SNA and non-SNA business and engineering applications.

References

SNA Perspective June 1990 "The Integration of SNA and Token Ring"

SNA Perspective September 1990 "Improved SNA Support for Token Ring"

"Token-Ring Network Bridges and Management," IBM document GG24-3062

"Token-Ring Network Architecture Reference," IBM document SC30-3374

"IBM Token-Ring Network Problem Determination Guide," IBM document SY27-0280

"IBM Token-Ring Network Introduction and Planning Guide," IBM document GA27-3677

"IBM Token-Ring Network Installation Guide," IBM document GA27-3678

"37x5 NCP Token-Ring Interface Planning and Implementation," IBM document GG24-3110 ■

Architect's Corner

CMIP Is Not Esperanto

by Dr. John R. Pickens

> Esperanto - An artificial language with a vocabulary based on word roots common to many European languages and a regularized system of inflection. Designed to be a universal language.

I've talked previously about IBM's move to manage SNA networks with CMIP (see *Architect's Corner* regarding CMIP/APPC [June 1990] and CMIP/LLC1 [January 1991]). Since then, I've been asked, only CMIP? What about SNMP? Is one management model enough? Is a CMIP-only manager the answer? What about SNA/MS?

IBM is heavily promoting CMIP. Nevertheless, because of the success of SNMP, mixed CMIP/SNMP environments are going to exist for a long time. It is time to clear out some deadwood and misperceptions regarding the respective roles of CMIP and SNMP.

A widespread perception exists that CMIP (whether CMIP/OSI, CMIP/SNA, CMIP/TCP, or CMIP/LLC1) can transparently front-end SNMP agents through proxies or universal manager interfaces. This has caused many to view CMIP as a kind of manager-level "Esperanto" for network management, especially in conjunction with SNMP. I believe this perception is false, both for pragmatic and technical feasibility considerations.

First, a disclaimer: The discussion which follows neither praises nor criticizes CMIP and SNMP. If anything, it highlights the folly of not being able to achieve multivendor agreement on a single management model. Neither does it criticize IBM—this is an industry-wide problem which requires a pragmatic coexistence strategy.

To set the stage for this analysis, and to demonstrate one source of the "CMIP is Esperanto" misperception, I go back to an entertaining session held last year at the 1990 Interop conference. Jeff Case of the IETF SNMP Directorate and Dave Mahler of the OSI Network Management Forum painted an "all-is-well-and-beautiful" picture of a converged CMIP-SNMP management model. The short version goes something like this:

"CMIP is the manager-to-manager protocoL SNMP is the manager-to-agent protocol. The two models have differentiated and well integrated roles."

I disagree.

Continuing in the light of statements like the above, two corollaries can be drawn, both wrong (see Figure 8):

- A universal interface can be created within managers which provides full transparency to both CMIP and SNMP.
- An SNMP system managed through a CMIP-to-SNMP translation function (proxy) is fully and transparently manageable by CMIP managers.

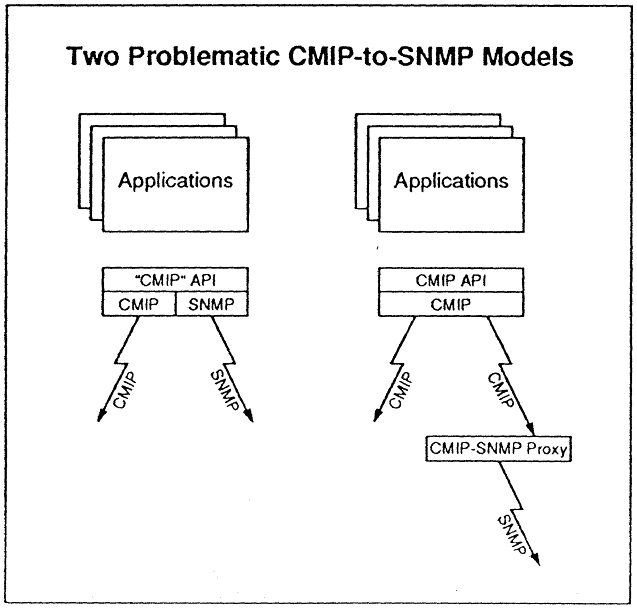


Figure 8

To counter the above, I would make a few technical observations:

- CMIP and SNMP have incompatible primitive operation sets
- CMIP and SNMP have incompatible object naming, inheritance, and containment models
- CMIP and SNMP MIBs for similar types of managed systems are being defined by different committees

The problem of accessing an SNMP MIB via a CMIP proxy is like trying to convert data between two similar database applications developed with different database schema and different database paradigms (e.g., network versus relational)—a great PhD thesis opportunity.

Here is a recent example highlighting the difficulty of mapping CMIP to SNMP even when a committee starts with good intentions to maintain compatibility. The example concerns management of an Ethernet hub, although the same discussion would apply to a token ring hub.

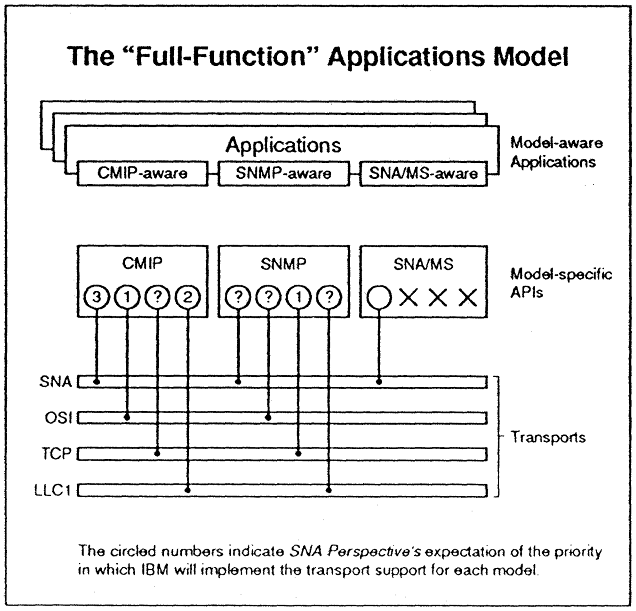
An IETF committee was formed in July 1991 with the charter to adapt the IEEE 802.3 Hub Repeater MIB (developed within the CMIP model) to SNMP. The original IEEE 802.3 version modeled a repeater system as having groups of repeater modules—in CMIP this is easily done by just adding a hub identifier attribute and creating a name binding. When considering this issue, the IETF committee rejected the repeater group concept—in SNMP this requires introduction of an extra level of indirection via an extra table object. Thus, a CMIP application cannot be developed which equally handles both SNMP-managed (via proxy) and CMIP-managed IEEE 802.3 repeaters.

Actually, by using other clever mechanisms (like SNMP community strings), such a proxy could offer a pretty close simulation of the IEEE 802.3 CMIP model. But this raises another even larger pragmatic issue. Consider the prospect of having to craft a clever SNMP-to-CMIP translation for each type of SNMP managed object (name mapping, function simulation, incompatible object models), and, even worse, having to keep the translation up-to-date, debugged, and distributed to the places in the network where proxies are executed. It just is not going to happen.

The same issues would occur if one attempted to write an application which transparently managed an 802.3 repeater hub from either CMIP or SNMP, even without a proxy.

This may be a bit of an understatement, but the industry is paying a terrible price for not being able to standardize on a single management model. No vendor, not even IBM, can afford to fully implement both models. And, since the functional and MIB models do not map, no application can be written that fully covers systems which contain both models.

From its announcements and statements of direction, it is clear that IBM has cast its lot in the OSI standards camp—CMIP. CMIP for SNA and OSI networks. CMIP to the bridge and hub. Alliances with leading LAN vendors for CMIP to the desktop. SystemView managed object libraries based upon CMIP. Strong political support of OSI Network Management Forum OmniPoint process to create managed object libraries based upon CMIP. As CMIP is implemented by increasing numbers of vendors and products, this strategy will be seen, looking back, as having been a smart forward-looking move.


The "Full-Function" Applications Model

The circled numbers indicate *SNA Perspective's* expectation of the priority in which IBM will implement the transport support for each model.

Figure 9

But, for the moment, IBM must provide parallel support for SNMP, both within its manager and within (some of) its agents. IBM has acknowledged this solution by choosing OpenView for its AIX based manager—OpenView exposes both an SNMP interface and a CMIP interface—and by providing SNMP agent implementations for desktop (OS/2 TCP), workstation (AIX TCP), and mainframe (MVS/VM TCP) platforms.

One final question needs to be addressed. What about support of today's SNA/MS flows, including NMVT, Alerts, Commands? I would just point out that all of the problems above concerning support of SNMP also apply to support of SNA/MS. The same solutions apply also. Proxies can be used, but suffer the same loss-of-function characteristics when passed through the translation process. Therefore, to preserve full function, applications must manage SNA/MS systems through an SNA/MS interface, just as in the SNMP discussion above.

The conclusion I draw from this analysis is that, in the long run, full-function management in IBM network environments will only be achieved by migrating everything to CMIP. But, in the interim, migration support is required, as shown in Figure 2—three interfaces—CMIP, SNMP, SNA/MS—and converged transport layer mappings—CMIP/SNA, SNMP/SNA, and SNA/MS/SNA). (Oops, did I say SNMP/SNA? Where did that come from?)

Such is the price of coexistence with de facto standards.

Footnote: It is noteworthy in the history of this column that CMIP has become an SNA discussion topic. This says much about IBM's progress toward converging SNA and OSI. ■

IBM Announcements

IBM Multiprotocol Router Plans Unveiled; No PU 4!

In August, IBM held a briefing to discuss its direction regarding its forthcoming multiprotocol router. No pricing information nor announcement/shipping dates were provided, though it was indicated that a limited version could be available in the first half of 1992. The company said its multiprotocol router would be based on the RS/6000, as we expected (see *SNA Perspective* July 1991).

Though IBM intends to do most of the work on the router itself, as was stated in June by Ellen Hancock, IBM VP and General Manager, Networking Systems (formerly Communications Systems), *SNA Perspective* believes that IBM is working with other companies for certain elements of the product. Although not specifically addressed at this briefing, *SNA Perspective* understands that IBM and Wellfleet are no longer working together on this product. (That relationship, though strongly rumored, was never officially made public.)

The router will support APPN protocols as well as OSI, TCP/IP, IPX, NetBIOS, DECnet, XNS, and AppleTalk. However, coming as a surprise to many, IBM stated that it would not provide native-mode node type 4 (previously called PU 4) routing support. Instead, traditional subarea SNA communication would probably be supported via encapsulation and synchronous pass-through, as is being done by most multiprotocol router vendors with SNA support.

The choice not to include native-mode node type 4 seems a reasonable strategy to *SNA Perspective* for several reasons.

First, full type 4 node support is a complex undertaking, much beyond the scope of other protocols now supported on multiprotocol routers. It would involve significant investment even for IBM to port the NCP code to the RS/6000, and might price the resulting product out of the router market.

Second, IBM does not want the router to impact sales of 37x5 communications controllers, which are at a higher price point than multiprotocol routers. Also, *SNA Perspective* believes that IBM's strategy includes support of separate SNA and non-SNA networks—operating in parallel over converged lower layers: locally on LANs and in WANs with newer technologies including frame relay and SMDS. For connection between SNA and non-SNA systems, its strategy is OSI, not open SNA.

Third, users we've heard from, far from clamoring for it, are concerned about adding an emulated type 4 node implementation into the subarea backbone, especially in a box that supports and is connected to non-SNA networks. They are concerned about how it might impact the rest of their SNA network.

APPC Developer's Resource

Have you been frustrated in your search to find classes and publications related to APPC? Two new catalogs from IBM can help save you time and effort. The "Education Catalog for APPC" lists every book and manual available on APPC, APPN, and CPI-C, along with every known educational opportunity and class on these topics, both inside and outside of IBM. The "APPC Development Tools Catalog" describes dozens of APPC software development tools designed to speed the deployment of distributed applications across platforms. To get copies of these free catalogs, write to:

APPC Market Enablement
IBM Corporation
Department E42, Building 673
P.O. Box 12195
Research Triangle Park, NC 27709

or send a note on the Internet to:

appcmrkt@ralvm6.vnet.ibm.com

Fourth, probably because of the difficulty and user concern noted above, no other multiprotocol router company is planning full type 4 node support. Cisco Systems created a lot of press earlier in the year with its five-phase IBM communications support strategy (see *SNA Perspective* April 1991), which included SNA routing. However, the company has been backpedaling since then and *SNA Perspective* does not expect it to complete the strategy as originally stated. However, we expect several multiprotocol router vendors to provide enhancements to their SNA synchronous pass-through products, as their SNA expertise develops, with some level of node type 4 "spoofing" (see *SNA Perspective* July 1991, "How Much Type 4 Do You Need?"). Since these other companies are unlikely to support it, and IBM is providing this product to compete with them, it need not support node type 4 either.

Fifth, momentum for APPN seems to be building in 1991. In large SNA accounts with significant multivendor protocol presence, which would be those most likely interested in SNA support on multiprotocol routers, are also the users most likely to be planning transitions from subarea SNA to APPN. Adding APPN rather than subarea SNA support to multiprotocol routers makes more sense.

Although IBM probably intends this revelation to stem sales of multiprotocol routers from competing vendors, it will likely instead serve to legitimize these products to those users reluctant to spring for multiprotocol routers with SNA support in the absence of an understanding of IBM's direction. However, *SNA Perspective* appreciates this as one of many recent statements from IBM regarding its future plans which address users' need for long-range direction to assist in today's decisions. ■

Copyright © 1991 CSI - Communication Solutions, Incorporated, all rights reserved. Reproduction is prohibited. • Subscription rates: U.S. - one year $350, two years $520. International - one year $385, two years $590 • *SNA Perspective* is published monthly by CSI • 2071 Hamilton Avenue • San Jose, CA 95125 • Telephone (408) 371-5790 • Fax (408) 371-5779 • Managing Editor: Louise Herndon Wells • Associate Editors: Vincent Busam, Alan Kucharski, Basil Treppa • Circulation Coordinator: Cheryl Roberts • Contributors: Wayne Clark, Howard Bernstein, Dr. John R. Pickens • Typesetting and illustration: Aaron Lyon at dSIGN • The information and opinions within are based on the best information available, but completeness and accuracy cannot be guaranteed.