SNA Perspective

Volume 12, Number 5
May 1991
ISSN 0270-7284

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

SNA Blues in the 1990s

The industry dominance of SNA does not mean that SNA is without fault or limitations. Some of SNA's traditional shortcomings—such as its S/370 host-centered nature, hierarchical control structure, and penchant for static configurations—have long been bandied around vociferously, though are somewhat obsolete today.

SNA will continue to face difficulties over the next several years in cohesively dealing with the demands of contemporary, PC-centered networking while maintaining its customer's investments and its traditional strengths in reliability, manageability, and security. This article addresses several current concerns including SNA-LAN interconnection, dynamic routing, connectionless networking support, and standard APIs.

(continued on page 2)

How SNA Fits Different Models of Interoperability

The subject of interoperability has been a hot topic in the Internet and OSI community for several years now. Many networking professionals view interoperability as the key to most successful TCP/IP to OSI transition strategies. Until recently, the subject of SNA interoperability was largely ignored by everyone except IBM. Why all of this recent activity surrounding SNA interoperability? Probably the most obvious reason is that, for the first time, there are viable alternatives to SNA in the business world. No one vendor—not even IBM—can provide all of the best solutions for business. Today's multivendor environment opens the door to SNA interoperability.

Numerous side-by-side comparisons have been made between SNA and the other two contemporary networking architectures: OSI and TCP/IP. These comparisons often disparage SNA, making SNA out to be somehow inferior to these other architectures. This article shows that SNA holds its own when viewed from a perspective of interoperability, using six different interoperability models:

- Application gateways
- Dual protocol stacks
- Common APIs
- Hybrid stacks
- Network access tunnels
- Transport relays

(continued on page 7)

In This Issue:

(Continued from page 1)

This is Not Your Father's SNA

Through most of the 1980s, but with greater urgency over the last four years, IBM has made a creditable effort to revamp and update SNA to better cope with the expected rigors of the 1990s networks. Given that SNA came to being a full seven years prior to the birth of the IBM PC—which so revolutionized the fabric and feel of modern computing—and eleven years before the token-ring LANs that now criss-cross SNA networks, it is not surprising that SNA required some major reworking and functional torquing.

Eleven far-reaching extensions have been made to SNA over the last ten years. These include:

- LU Type 6.2
- Type 2.1 nodes
- Type 2.1 node integration into subarea networks
- Advanced Peer-to-Peer Networking (APPN)
- Systems Network Interconnect (SNI)
- SNA Distribution Services (SNADS)
- Alternate sessions (i.e., Extended Recovery Facility (XRF))
- SNA Management Services (SNA/MS)
- SNA File Services (SNA/FS)

Most of these architectural extensions have also been updated during this period, some more than once. In concert, there have also been six major, function-bolstering releases of VTAM over the last nine years, with the last two—Version 3 Release 3 and Version 3 Release 4—being announced ten months apart. This despite most mid- to large-size multi-host SNA installations realistically requiring twelve to eighteen months to evaluate and migrate to a new release of VTAM. IBM certainly has been active in moving SNA into the 1990s.

Because of these architectural and implementation enhancements, the salient characteristics of the emerging SNA are addressing much of the old "what is wrong with SNA" list. The S/370 host-bound, hierarchical, static SNA of the 1980s is being supplanted with a dynamic, peer-oriented, cooperative processing-based, robust, resilient, and manageable networking scheme, spearheaded by APPN.

A New Set of Woes

Today's SNA, though a vast improvement over the old mid-1980s SNA, is by no means perfect. It still has some beguiling, even contradictory, shortcomings. For example, automatic, dynamic alternate routing in the event of a path failure is still not available within the backbone subarea network, while adaptive alternate routing is a key feature of the APPN architecture.

Generalized LAN interconnection across SNA backbone networks is another well-publicized concern. The OS/2-based LAN-to-LAN Wide Area Network Program, announced in September 1990 and scheduled to be available in April 1991, as well as the recently announced OS/2 APPN routing with NetBIOS coexistence, may alleviate this somewhat. The first option permits NetBIOS flows to be transported between LANs using LU 6.2 across SNA. The second theoretically supports NetBIOS and APPN traffic across SNA WAN links. (See the Interoperability article in this issue for further discussion.) The source of these trans-SNA LAN interconnection woes could have been IBM's belief that LU 6.2, rather than NetBIOS, would be the predominant data flow on IBM-oriented LANs by the early to mid-1990s.

While some of the current SNA issues—such as trans-SNA LAN interconnection—are caused by technical and/or implementational limitations, others—like the reluctance to publish the APPN Network Node (NN) architecture—are more likely due to political and competitive reasons. The good news, however, is that most if not all of SNA current woes are surmountable, provided there is sufficient motivation (read customer pressure).

Key Shortcomings

The key shortcomings of today's SNA are:

- Poor LAN interconnection across SNA backbones
- Lack of automatic, dynamic routing in subarea networks
- Need for support of interactive mode interactions on LU 6.2
- Lower emphasis on application management
- Lack of connectionless interactions
- Need for standard APIs, à la CPI-C, for SNADS, DIA, SNA/FS.
- Actions indicating some movement away from the philosophy of open SNA

LAN Interconnection Across SNA Backbones

In many SNA networks, LANs can be found in locations that were previously served either by 3270 terminals or a departmental midrange system. Typically these LANs are connected to the SNA backbone network via a PC or PS/2 gateway, a 37x5 communications controller, or a 3174 establishment controller. Such LAN connections are primarily used to provide S/370 host access to PCs and midrange systems, such as AS/400s, attached to these LANs.

While the process for defining such LAN devices to VTAM and NCP is fairly gruesome, these connections, once defined, provide satisfactory access to host applications via 3270 or even LU 6.2 SNA protocols. Type 2.1 node-based, device-to-device interactions may also be realized across such SNA backbone interconnected LANs in conjunction with the new VTAM 3.2 (and greater) and NCP T2.1 Node Integration support. (See *SNA Perspective*, January 1990, *Breaking the Chains of Hierarchical Networking: Integrating Node Type 2.1.*)

SNA (or to be precise NCP with VTAM), however, does not presently permit any non-SNA traffic, such as NetBIOS or TCP/IP, to be exchanged between LANs across an SNA backbone network. The OS/2 front-ended IBM LAN to LAN Wide Area Network Program product, scheduled to be available in April 1991, will attempt to address this crucial shortcoming to an extent. *SNA Perspective* believes it unlikely to provide adequate capacity and throughput to satisfy the demands of most customers.

This LAN interconnection shortfall is exacerbated in many locations which already have SNA backbone connections between several locations and now require some non-SNA LAN-to-LAN interactions between them. The inability to use these existing SNA links forces the deployment of costly duplicate parallel networks between the same locations—one for SNA traffic and one for non-SNA traffic. Routers are a preferred option for realizing the non-SNA network. However, with routers becoming able to support SNA traffic, new heterogeneous networks are moving to supplant large tracks of what were originally 37x5-routed SNA backbone networks. Some believe that IBM will enter this fray with an RS/6000-based multiprotocol router.

In addition, over the next eighteen months, IBM is likely to respond to this obvious threat to its traditional SNA subarea networks with an NCP interconnection adjunct for non-SNA LAN traffic along the lines of its existing X.25 SNA Interconnection (XI) and Non-SNA Interconnection (NSI) products. This will most likely be announced in conjunction with a new, higher-capacity, more powerful 3745 replacement. A relatively costly solution such as this will not displace router-supported SNA backbones. Non-IBM multiprotocol routers will soon be a long-term, strategic fixture in most SNA networks.

Automatic Dynamic Alternate Routing

Fully interconnected, mesh-based, multiple, alternate routes between networking nodes have been available in SNA since 1978. However, inexplicably, SNA has yet to provide for automatic alternate routing within subarea environments, even in the event of a route failure, let alone as an optional means of easing traffic congestion. Alternate route usage within traditional SNA networks are today only possible with manual intervention. Ironically, APPN Network Nodes provide adaptive alternate routing along the lines of classic packet switching networks.

When a SNA subarea network route fails, as a result of either permanent link outages or the failure of an intermediate networking node, all the sessions that were using that route are automatically terminated, disrupting any active end-to-end interactions. Such failures are even possible in XRF configurations, particularly in the event of the failure of an intermediate NCP node (see *SNA Perspective* November, 1990: *XRF: High Availability à la SNA*). SNA does offer a degree of relief against such failures with its multiple parallel link, single logical pipe, transmission groups between subarea nodes. When transmission groups are in use, all the links that comprise a particular group have to be inactive at the same time before a route using that group is considered to be inoperable.

X.25 Interconnection

X.25 Interconnection (XI) permits X.25 packets to be transported between X.25 devices across an NCP-supported SNA backbone network, while Non-SNA Interconnection (NSI) permits bisynchronous (BSC) remote job entry (RJE) traffic (e.g., 3780) to be conveyed over an SNA network. NSI offers an insight into IBM's inherent hesitancy to offer non-SNA interconnection capabilities with SNA.
It was announced in 1983 when BSC RJE traffic was at last declining. NCR Comten had addressed this requirement on their 3600 (IBM 3705-alternatives) since 1974. For more on SNA X.25 support, see the Interoperability article in this issue. ■

When there is a route failure within a subarea network, all the affected end user pairs are notified that their sessions have been disrupted. The end users then have the option of immediately attempting to re-establish new sessions by invoking the appropriate session initiation processes—such as logging on again in the case of terminal users. If suitable alternate routes had been previously defined, SNA will establish the sessions using the first available alternate route. This manual process could obviously be replaced with a network-arbitrated scheme that automatically, transparently, and nondisruptively transfers sessions that were using a failed route to an alternate route, à la APPN.

Automatic, dynamic alternate routing to circumvent route failures and route congestion is a standard feature on most packet switching networks. Ironically, the basic underlying methodology for realizing automatic alternate routing, such as the separate management of logical (i.e., Virtual Routes (VRs)) and physical routes (i.e., Explicit Routes (ERs)) with a mapping mechanism to assign logical routes to the appropriate physical routes, has been in SNA since 1978.

Automatic alternate routing is not always desirable. For example, an alternate route may severely degrade the performance of interactive sessions as result of being very circuitous, using slower links, or having to cope with increased traffic. In such cases, it might still be more desirable to notify the users of the route failure and give them the option of logging on to other applications until the optimum route to their previous destinations is reestablished.

Interactive Mode LU Type 6.2 Interactions

LU 6.2 is targeted at application-to-application interactions. Such interactions, on the whole, have less demanding response-time requirements than those involving terminal users, given that the former can be done in background mode. This lack of emphasis on response time critical processing is reflected in the actual LU 6.2 architecture. LU Type 6.2, in addition to being built on top of a half-duplex (i.e., one way at a time) SNA communication protocols, is also a local, 'asynchronous' interface protocol whose default mode of operation does not concern itself with real-time, *end-to-end* data exchanges.

Generically, an LU 6.2 in essence acts as a local store-and-forward processor that accepts data from the local transaction processing applications it is serving in real-time, but reserves the right to forward that data to physically remote partner LU 6.2s for delivery to their eventual destinations, not in real-time, that is, asynchronously. An application could send a series of messages to a partner application by executing a series of SEND_DATA verbs. Provided that there were no errors at the local interface level, the local LU will acknowledge the receipt of these messages. The LU 6.2 is, however, not obligated to immediately forward each message to the remote LU 6.2. It will only forward the messages, usually in the form of a block, when the data exceed a certain buffer threshold or a maximum packet length. An application can force the forwarding of messages by using a special verb, referred to as FLUSH, or by requesting certain syncpoint related confirmations.

As an aside, programmers who are cognizant of this "asynchronous" nature of LU 6.2 transactions, but nonetheless still wish to use it for response-critical applications, compensate for this by either issuing FLUSH at regular intervals, or by selecting buffer and message sizes to prevent the buffering of multiple messages. Though tedious, this does work. A more elegant solution would be to provide an optional explicit interactive mode which, when in effect, would automatically forward data on receipt and, in the future, may even permit full-duplex interactions.

Lower Emphasis on Application Management

Even with SystemView, IBM's management philosophy, reflecting its 1970s RECFMS-based hardware statistics collecting origins, is heavily biased towards monitoring and controlling hardware entities rather than logical entities. There are few, if any, integrated facilities in SNA Management Services (SNA/MS) today enabling direct, management-oriented interactions with remote applications. An LU 6.2 session with application specific dialogues would be the closest true solution. In essence, SNA's current management facilities revolve around SSCP-PU interactions, rather than SSCP-LU or even LU-LU interactions.

Response-Time Monitor (RTM) and Asset Management are good examples of leading-edge management facilities that could benefit from a more balanced approach that favors the management of both logical and physical entities. For RTM to be meaningful, response times have to be measured on a per-terminal-user (i.e., LU) basis. Given that SNA management requests are not sent on SSCP-LU sessions, the extraction of response time statistics and the resetting of the counters are achieved through a convoluted process involving non-architected, product-specific data flows between LUs and their local PU.

While physical entity management using the current SNA/MS techniques is imperative to ensure that a reliable and resilient transport network can be maintained, SNA (or possibly an anticipated systems management architecture (SMA)) must also provide comparable direct facilities and interfaces whereby the operation of applications could also be explicitly monitored and manipulated by a host-based SystemView management scheme.

Connectionless Interactions

SNA has always been and still is largely a session (connection-oriented) architecture. Even inter-T2.1 node interactions are conducted within the context of sessions. This is unfortunate, since there are applications ideally suited to be addressed with T2.1 nodes that only require require the exchange of two messages. Credit card validation is an example of such very short duration, one-two, single message in each direction applications. However, SNA still insists on the overhead of establishing a session in order to conduct such a quick-fire transaction. (An undocumented, back-door technique exists by which a connectionless transaction may be realized in a

T2.1 node environment, along the lines of the X.25 Fast Select function, provided that all necessary data is contained in 65 characters.)

Many existing and future SNA applications could obtain performance and fiscal gains in switched link configurations by being able to perform connectionless interactions when dealing with short-duration transactions. IBM should seriously consider providing such connectionless support, even if such support is restricted to T2.1 nodes. SNA is coming close to meeting this need in some new APPC features, which are discussed in this issue's Architect's Corner.

Standard Interfaces to LU 6.2 Services

Thankfully, with CPI-C, there is now a standard, uniform interface to LU 6.2 across all the major SAA platforms—MVS, VM, OS/400, and OS/2 (and CICS and IMS). Unfortunately, CPI-C does not provide direct access to LU 6.2 services such as SNADS and SNA/FS. SNADS and DIA implementations on the various IBM platforms still sport their own idiosyncratic interfaces, as was the case with LU 6.2 prior to CPI-C. This lack of consistent exposed interfaces is now unacceptable, especially in the light of SAA. Given that SAA embraces all these services as being Application Services, it is now time for CPI-C–like interfaces to be defined for all of these services.

Movement Away from Open SNA

In the early years of its existence, SNA was an open, public domain architecture. In effect, it was the epitome of an open communications architecture. IBM worked to promote and ensure this. If timely access to detailed and accurate architectural specifications was the metric for openness, SNA passed with flying colors. All SNA architectural workings, including specifications for extremely esoteric functions such as those for SNI, were available in the form of published, general-access, orderable IBM manuals.

Whether this openness was promoted by a genuine egalitarian desire to promote SNA as a true, nonproprietary *de facto* international standard or just to allay the 1970s anti-trust pressure, it certainly paid handsome dividends for IBM. By the mid-1980s, all major computer manufacturers provided SNA coexistence products and SNA was often the preferred common denominator in the commercial sphere for multivendor integration. If not for this initial openness which encouraged coexistence solutions, SNA is unlikely to have maintained its dominance in commercial networking and there would have been a more strident demand for earlier OSI-based interoperability.

Ironically, as OSI becomes much more of a viable competitor, IBM, rather than ensuring that SNA remains a credible open architecture, seems to be drawing a opaque veil around SNA. This was painfully highlighted on March 5, 1991, when IBM announced APPN support on 3174s and OS/2. Though APPN is now a formal SNA architecture, IBM is currently only going to make public the architectural specification for APPN End Nodes (ENs). The specification for the APPN Network Nodes (NNs)—which provides intermediate node routing, alternate routes, and dynamical LU registration—is to remain IBM proprietary. One can appreciate the full implications of this regrettable decision by IBM if one thinks of NNs as being the equivalent of Type 4 and Type 5 nodes in traditional subarea network environments. Only making available the EN specification is comparable to only making the architecture for Type 2 nodes available, rather than that for all the node types, as was customary with SNA. (It should be noted, however, that the PU 4 and PU 5 specifications are not truly open either, as they were moved to be licensed documentation in the mid-1980s.)

Things are also not that rosy in terms of subarea network specifications. While SNA subarea network architectural manuals are still published at irregular intervals, there is a distinct and noticeable paucity of technical information on leading edge facilities such as T2.1 node integration, alternate sessions for XRF, and casual connections. Even the up-to-date, accurate formats of the SNA request, in the past a common currency, are now difficult to come by and only published in "licensed" documentation explicitly restricted to VTAM customers.

SNA, especially with IBM's recent APPN stance, appears to be becoming a black-box, with just the LU 6.2 protocol boundary (i.e., API), T2.1 Node physical interfaces, and APPN EN interfaces exposed. This serves as the only means by which this powerful networking resource can be used for end-to-end data interchange. The internal workings of this black-box—in terms of session control, presentation services, and the SNA(/APPN) path control network—may well end up being shielded from customers as well as other suppliers. Fortunately, this will not totally preclude SNA coexistence, provided it is through standard LU Type 6.2 interfaces. However, it could hinder problem determination and performance analysis. It will also discourage competitive and complementary SNA networking products from being developed.

A closed SNA architecture would not only be regressive, it would be counter to today's open system dominated thinking. After championing the cause of open architectures and thus reaping handsome profits, it would be unfortunate for SNA moving into its well-deserved prime as a crabby, un-emulated recluse. Given SNA's commercial installed base gives it the opportunity to be a major networking alternative well into the next millennium, *SNA Perspective* hopes that this current direction is but a temporary aberration. ■

(Continued from page 1)

Internetworking vs. Interoperability

First, let us draw a distinction between internetworking and interoperability. It might seem a minor point, but the two terms are often used interchangeably even though they have an entirely different scope.

- Internetworking can be thought of as the mechanism for communicating between heterogeneous networks.
- Interoperability addresses more than merely passing information between dissimilar networks.
- Interoperability is concerned with a broader range of issues of which internetworking is but one.

When a vendor provides an interoperable solution, it addresses several issues, including network interconnection, shared data and programs, common services, etc.

Why all of this consideration on distinguishing interoperability from internetworking? In its traditional and well-established way, IBM specifies its communication components—from its architectures to its formats and protocols—in a very extensible and comprehensive manner. This is typically intimated by IBM as a "total systems approach." Merely focusing on internetworking issues as they pertain to the SNA protocols, according to IBM, detracts from the overall interoperability solution.

When evaluating how SNA interoperates with other networking architectures, one must first understand the models of interoperability. This article discusses six different interoperability models:

- Application gateways
- Dual protocol stacks
- Common APIs
- Hybrid stacks
- Network access tunnels
- Transport relays

These models represent interoperability at several levels in the SNA architecture. As shown in this article, SNA is not nearly as closed as prevailing opinion may suggest.

Application Gateways

Application gateways are the most prevalent form of interoperability products in today's SNA world. Additionally, almost all of the SNA application gateways are for electronic mail between IBM's mainframe-based mail systems and other vendor's mail packages.

Disadvantages

Even though application gateways are the most common interoperability product in SNA, they suffer from a number of disadvantages:

- They are the least efficient of all interoperability models.
- The end-to-end transmission of packets is subject to a high latency.
- There are usually high resource requirements at the application gateway node.
- They are not general purpose (i.e., they serve a single application).

As the name implies, application gateways operate on top of the application layer of the OSI model. Figure 1 shows the relationship of an application gateway with its underlying protocol stacks—in this case SNA and OSI. When a protocol data unit is to be transferred between these heterogeneous networks, it has to traverse the entire protocol stack for both networks.

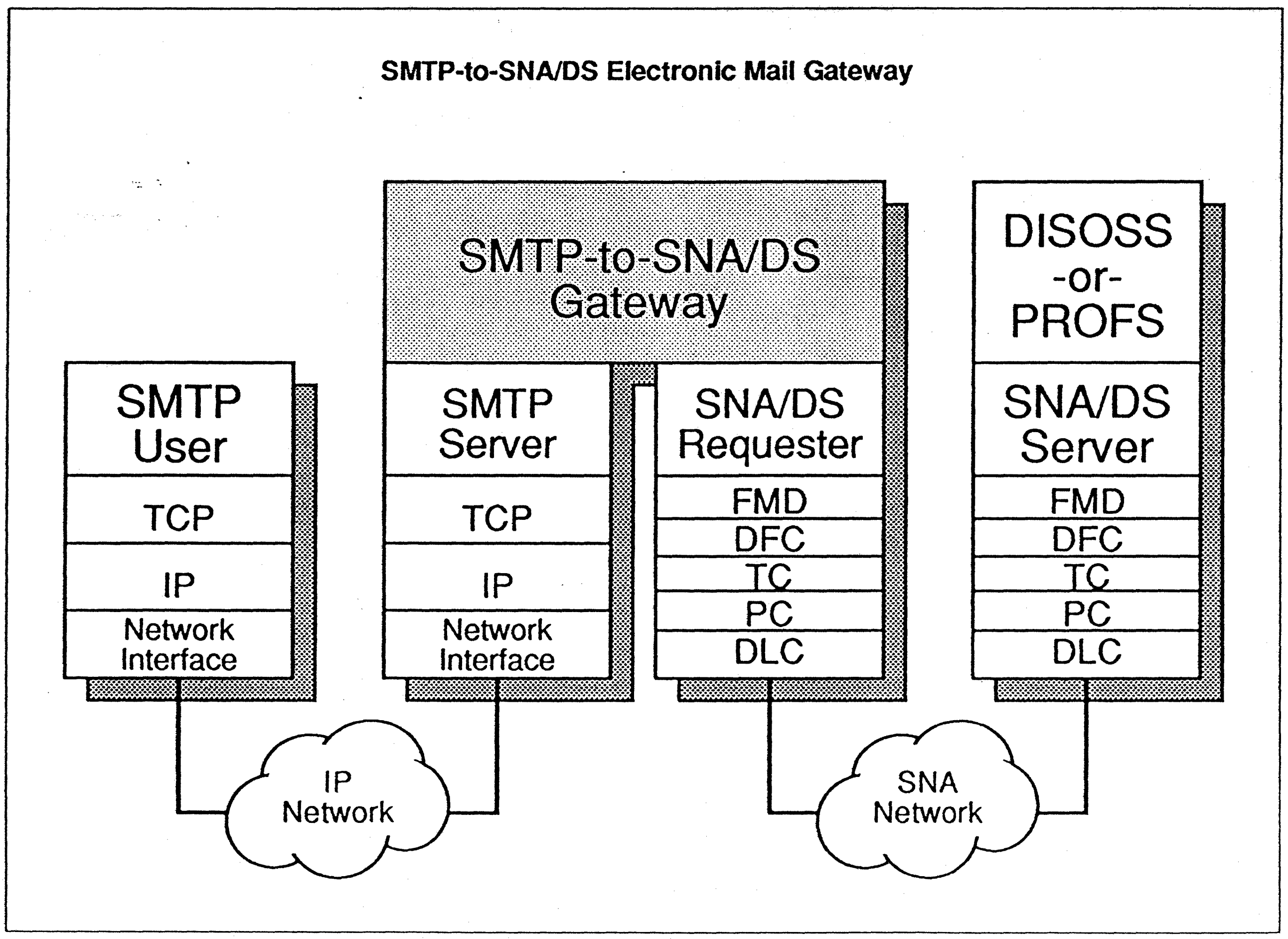


Figure 1

The second and third items above, latency and resource requirements, are related. The granularity of an object that is processed by the application gateway is quite large. In the case of an electronic mail application gateway, for example, it is the entire message with all of its attachments. If the application gateway enabled heterogeneous file transfer applications to transfer files between them, then the granularity of the object would most likely be the entire file.

The electronic mail application gateway must receive the entire message from one network before beginning to transform it and send it out on the other network. This inherent operation the application gateway results in a noticeable end-to-end latency between cooperating electronic mail applications. Also, since the entire electronic mail message is received before any transformation begins, the resource requirements of the intermediate node that implements the application gateway can be quite large.

Advantages

There are also distinct advantages to application gateways. Some of the more important advantages are:

- Preservation of existing services
- Isolation of differences at the edges of the local environment

With application gateways, the heterogeneous end systems do not require any modifications in order to communicate with one another. All differences are absorbed by the application gateway in the node that serves requests from the each of the end systems.

Figure 1 shows a typical electronic mail application gateway enabling communications between two dissimilar electronic mail environments: Simple Mail Transfer Protocol (SMTP) of the TCP/IP world and SNA/Distribution Services (SNA/DS) of the SNA world. SMTP users believe they are sending and receiving messages exclusively from the SMTP server while users of the SNA network believe they are dealing with only SNA-native messaging services. The transformation between the SMTP protocol and formats and the SNA/DS protocol and formats is handled transparently by the SMTP-to-SNA/DS application gateway.

Dual Protocol Stacks

The dual protocol stack model of interoperability looks very similar to the application gateway. Looking inside the interoperability enabling component itself, one sees that the operating style is quite different. While the application gateway transformation component interfaces to an application itself, the dual protocol stack component interfaces directly to a specific protocol layer in the network stacks resident on the intermediate node.

This direct interface to the protocol stack allows the bridging component in the dual stack approach to take advantage of the inherent properties of the underlying networks. The granularity of the objects that are handled is a protocol data unit within the network architecture as opposed to an entire file or message. End-to-end latency is reduced and throughput is increased. However, the task of interoperating between two protocol stacks in this manner is significantly more complicated than with application gateways.

There are many examples of dual protocol stack implementations within the IBM product line. Most of the implementations that fit this category have traditionally existed within the Network Control Program (NCP) of the communications controller (i.e., the 37x5). Products that fit this model are:

- Network Packet Switching Interface / Protocol Converter for Non-SNA Equipment (NPSI/PCNE)
- NPSI / X.25 Interconnect (NPSI/XI)

NPSI/PCNE running in the NCP permits X.25-based applications using X.25 network services access to SNA-based application in IBM hosts. NPSI/XI permits cooperating X.25-based applications running on X.25 networks to use SNA backbone networks for transport services.

The most recent example of a dual protocol stack implementation from IBM is shown in Figure 2. The *LAN-to-LAN Wide Area Network Program* allows peer NetBIOS-based applications to communicate with each other using SNA backbone networks. The program itself sits atop both the NetBIOS stack and the SNA stack within an OS/2 node. In simple terms, it concatenates a NetBIOS connection with an SNA session using LU 6.2 protocols. Since SNA is required on these intermediate OS/2 nodes, they must be running IBM's OS/2 Extended Edition Communications Manager.

The benefit of such a product is that, for a fairly modest price, IBM customers can use an SNA subarea backbone with all of its inherently reliable properties to connect remote NetBIOS-based applications. A typical application of this product would be to allow PC-based users access to file servers in remote locations.

To our knowledge, all of the dual protocol stack implementations in IBM products interface to SNA at the top layer of the SNA architecture, the Function Management Data (FMD) layer. One of the canons of interoperability is that it is much easier when the fringes of the network are addressed. This is especially true of SNA since the uppermost layer (i.e., FMD) is precisely the boundary where IBM's own applications come into contact with the SNA stack and where IBM provides application programming interfaces (APIs) for its customers to use.

Common APIs

Common application programming interfaces are extremely rare in the industry. SNA interoperability is unique in this regard since IBM's own suggestions for application portability in Systems Application Architecture (SAA) use a common API—the Common Programming Interface for Communications (CPI-C).

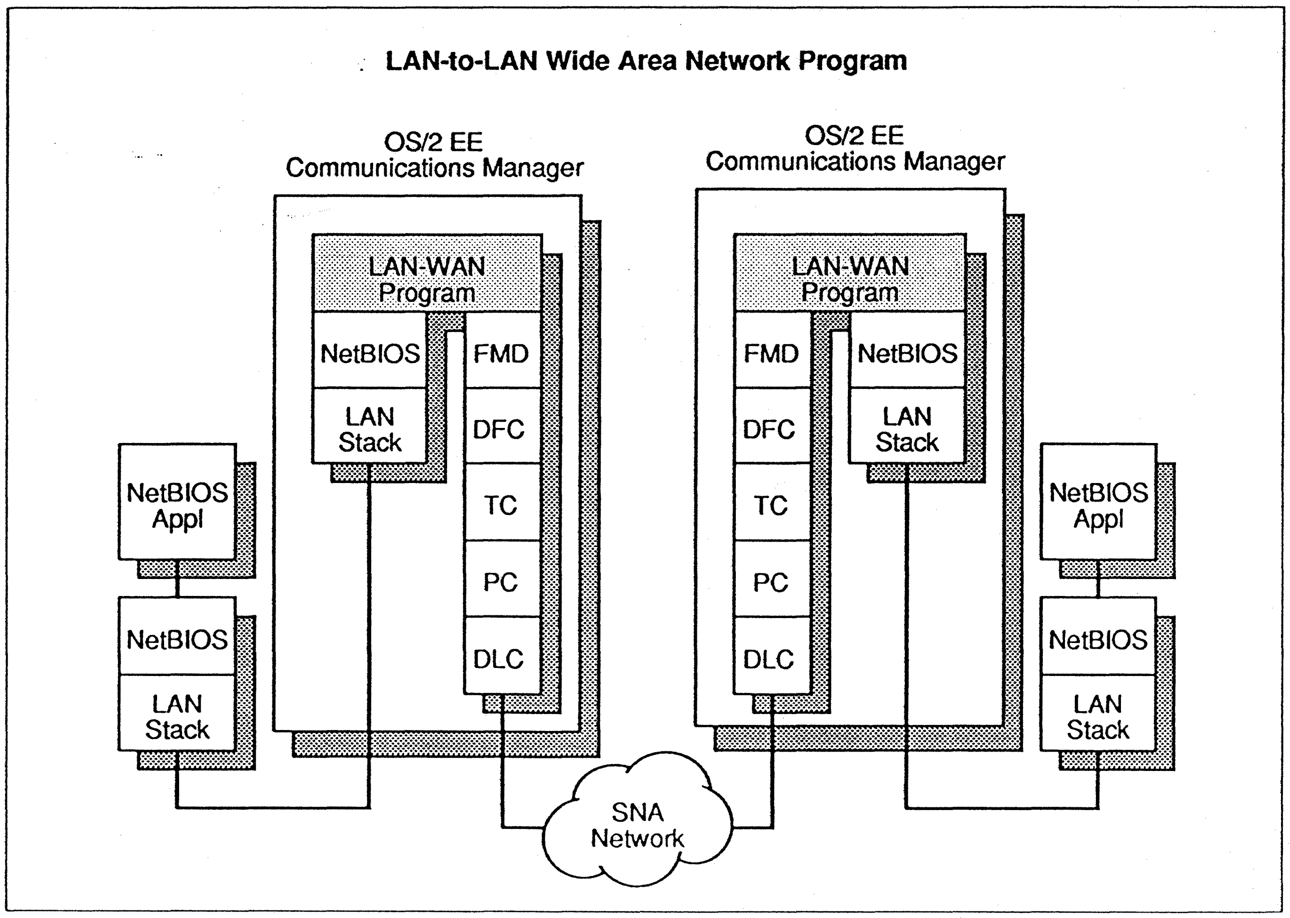


Figure 2

IBM advocates the use of CPI-C to develop network-independent applications so that these applications are insulated from the underlying network. SAA supports only the SNA and OSI networking architectures and, therefore, CPI-C can be used only for SNA / OSI interoperability. A customer can develop an application that uses CPI-C and then CPI-C will deal with interfacing to either the SNA or OSI protocol stacks. The details of this approach are shown in Figure 3.

Disadvantages

Common APIs have several disadvantages:

- The API definition is quite complex, resulting in increased application complexity.
- The API provides services according to the "lowest common denominator" of underlying protocol stacks.
- Some of the burden of interoperability is placed upon the application developer.

The second item is especially important when one examines CPI-C in relation to the underlying SNA and OSI networks. As is shown in Figure 3, CPI-C sits on top of either LU 6.2 for the SNA network or Transaction Program Application Service Element (TP-ASE) for the OSI network. LU 6.2 services and TP-ASE services are curiously similar but there are some differences. Also, there are a variety of underlying LU 6.2 services that are inaccessible to applications because they are not revealed at the CPI-C interface.

Advantages

With all of these disadvantages it might sound as if there is no good reason to use a common API. However, there are several distinct advantages:

- There is a one-time development cost for the application.
- The application appears as a network-native application to either network.
- There is insignificant performance degradation.

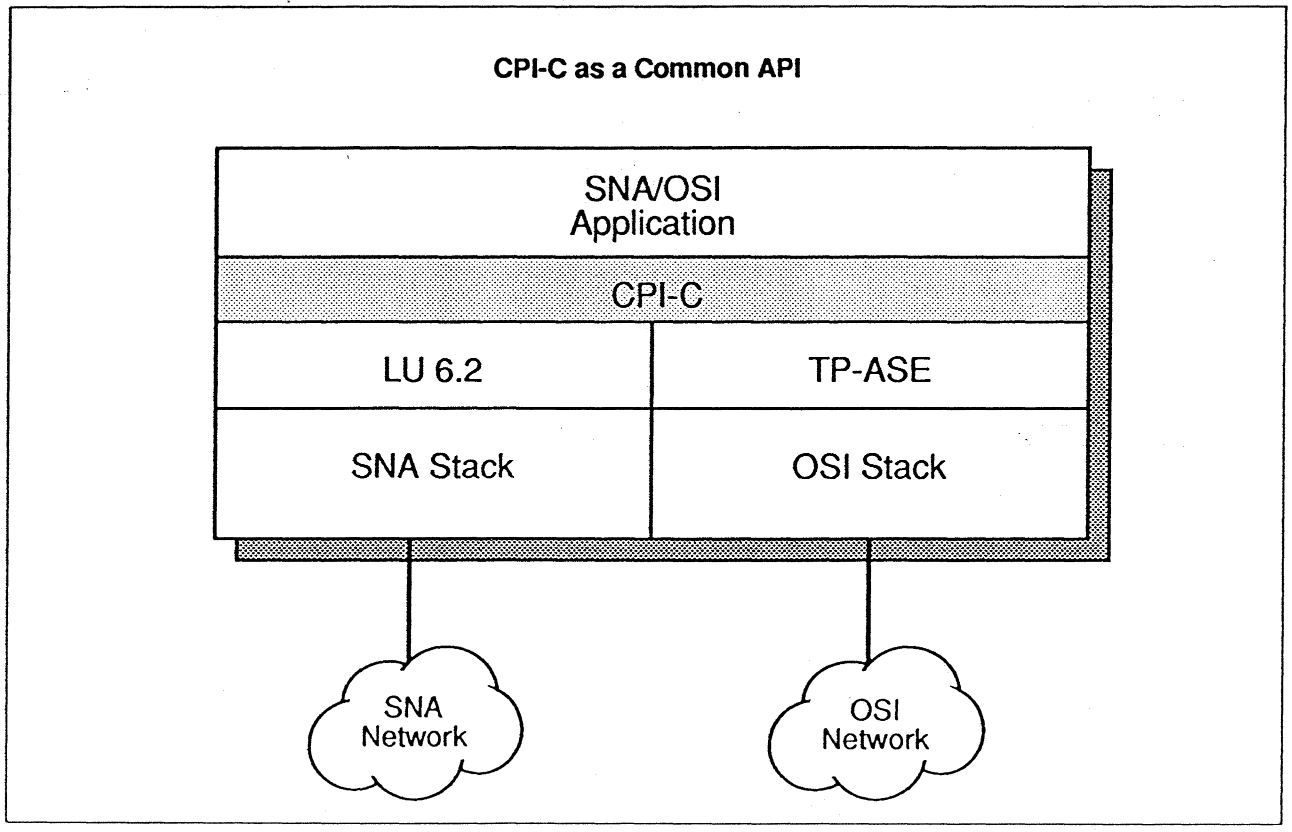


Figure 3

Hybrid Stacks

Perhaps the most rare of all interoperability models and, not coincidentally the most difficult to implement, is the hybrid stack. A hybrid stack implementation is one in which certain layers of one network architecture are grafted onto the appropriate layers of another different network architecture. This blending of the layers of disparate architectures is quite tricky and requires intimate knowledge of the operation of both architectures.

Perhaps the most famous of the hybrid stack implementations is contained in the ISO Development Environment (ISODE). ISODE is a platform for making the transition from TCP/IP-based services to OSI-based services and contains an implementation of the higher layers of OSI. The implementation of the OSI application, presentation, and session layers in ISODE permit TCP/IP's transport layer (i.e., TCP) to be used instead of the OSI transport layer. This fact should not be too surprising since OSI's reliable transport class (i.e., TP4) was patterned after TCP.

What might come as more of a surprise is a hybrid stack implementation incorporating both SNA and OSI layers. The more recent end user protocols in SNA appear to have been designed with interoperability in mind. The Logical Unit type 6.2 (LU 6.2) architecture embodies the Function Management Data (FMD) layer of SNA through Transmission Control (TC). There is a natural split in the middle of the LU 6.2 architecture that permits the Advanced Program-to-Program Communication (APPC) portion of the LU 6.2 architecture to be divorced from SNA.

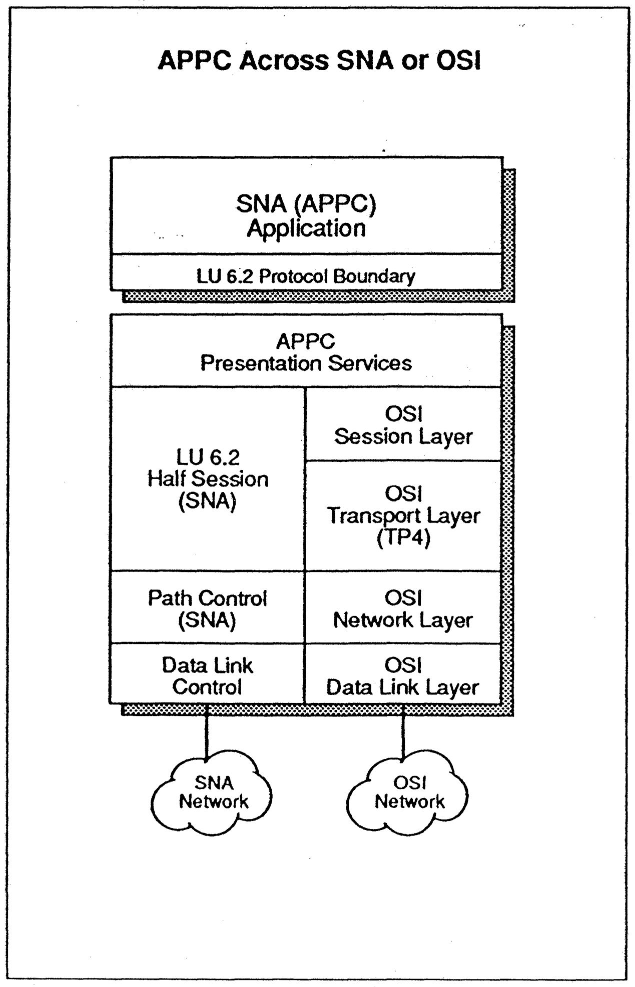


Figure 4

An SNA / OSI hybrid stack implementation of APPC is shown in Figure 4. In that diagram, one sees that the APPC Presentation Layer (also known as FMD in SNA parlance) sits atop either the LU 6.2 Half Session services for SNA or the OSI session layer. It is this interface that presents the biggest problem to hybrid stack implementations: mapping a layer of one network architecture to the non-native service boundary of a different network architecture is very difficult. However once the difficulty of grafting different protocol stacks together has been accomplished, there is little, if any, performance degradation when switching network architectures in the middle of the stack.

Figure 4 further emphasizes that there is indeed a difference between APPC and LU 6.2. APPC is an architecture and LU 6.2 is the manifestation of APPC within the SNA architecture. As this diagram exemplifies, the APPC architecture was designed to be flexible enough to use non-SNA networks for interconnection. through the SNA network is established at the time an SNA session is created and remains in effect for the duration of the session. Each packet within an SNA session takes the same path every time. This end-to-end static routing in SNA maps very well onto the virtual circuit architecture of X.25.

Network Access Tunnels

Network access tunnels are the oldest and best understood of all interoperability models. In this model, two end systems of a particular network type use an intermediate network of a different type to transport data packets. The intermediate network merely provides a path over which the two end systems communicate with one another. All of the complexity of the intermediate network is hidden from the end systems.

Advantage and Disadvantage

The primary effect on the end system networks can be viewed either as an advantage or a disadvantage: the end system networks are subject to the properties of the intermediate network. If the network properties of the tunnel are better than the native network services of the end systems, then the users of the end system network will see the tunnel as a benefit. The converse is true if the intermediate network offers a lower quality of service.

Figure 5 shows the most well-known example of network access tunneling in the world of IBM communications—SNA over X.25. SNA packets can be accommodated quite readily by X.25 since both SNA and X.25 share a very important network property: connection-oriented transfer of data. The path that a session takes through the SNA network is established at the time an SNA session is created and remains in effect for the duration of the session. Each packet within an SNA session takes the same path every time. This end-to-end static routing in SNA maps very well onto the virtual circuit architecture of X.25.

In SNA over X.25, the bottom of the SNA stack the Path Control layer—interfaces to Qualified Link Level Control (QLLC). QLLC provides all of the functionality for end-to-end control across the X.25 network. SNA data is transmitted in X.25 data packets while control packets are transmitted using the X.25-qualified data packets (thus the "qualified" in QLLC).

A variation on network access tunnels has arisen recently in router products from vendors such as Cisco Systems. Instead of using X.25 as the intermediate network over which SNA packets are transferred, an IP network is used.

Transport Relays

The final interoperability model is the transport relay. This model (also known as the transport service bridge) is concerned only with the transformation of layer 4 protocols between two dissimilar networks or between two different transport classes within the same network, as with OSI.

To our knowledge, there is no transport relay in existence between SNA's layer 4 (Transmission Control) and the transport layer of any other architecture. Level 4 relays work well only when the characteristics of the transport layer of the two networks are somewhat similar. SNA's Transmission Control is different enough from other transport layer protocols (e.g., TCP in TCP/IP and TP0 or TP4 in OSI) that no effective mapping exists. ■

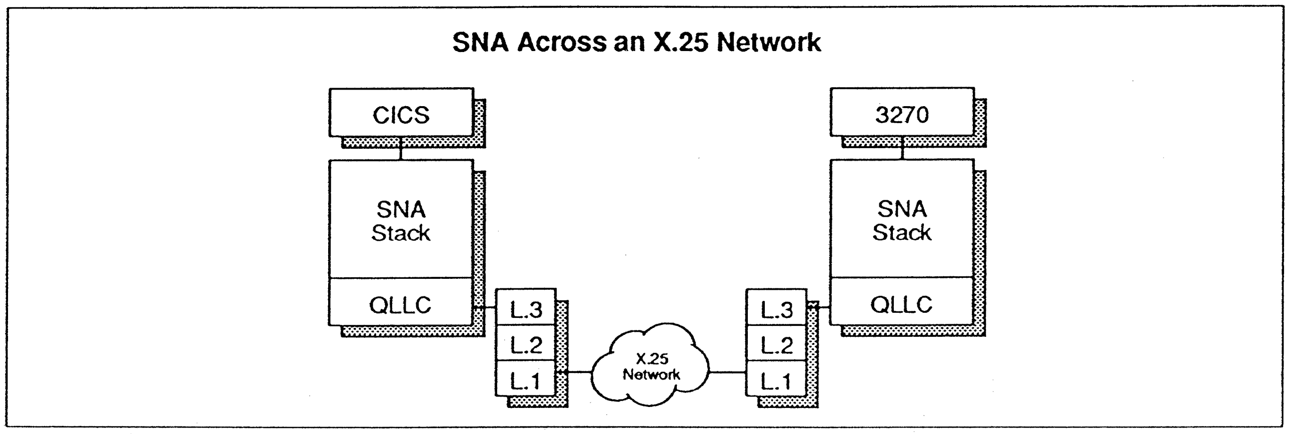


Figure 5

Architect's Corner

WHAT I LIKE ABOUT (NEW) APPC

by Dr. John R. Pickens

In late 1990, IBM upgraded its APPC peer protocols specification. With all the hoopla in 1991 over Ethernet, APPN, etc., this upgrade has been largely ignored. With the announcements dust of recent months now somewhat settled, I thought it would be useful and interesting to review what has changed in the world of APPC. I have always been a fan of APPC. With the new updates I like it even more.

The Specification Itself

The latest architectural specification, *IBM SC31-6808 - LU 6.2 Reference: Peer Protocols*, has doubled in size—from 6 to 12 chapters. With each rewrite of this book, the structure and functionality of APPC gets increasingly clearer. With the latest release, the frontend textual tutorial and overview section is much improved and the backend pseudo code is clarified. In particular, the current version of Peer Protocols makes consistent use of MUs as a modeling technique for intercomponent flows, especially compared to the old 6.2 protocols—*IBM SC30-3269 - Architecture Logic for LU Type 6.2.*

Also, in earlier versions of the specification there were areas of unreachable pseudo code or areas of functionality that seemed to completely missing. For example, the pseudo code for handling conversation mapping had a few obtuse logic flows and dead ends. In the current version, I have detected few such flaws (to date).

The Architectural Network Model

LU 6.2 developers will certainly appreciate the additional tutorial material which now appears at the head of each major component description. Much new architectural conceptual material is published for the first time in this specification.

For example, buried in the discussion of session-level pacing algorithms is a very useful new definition of SNA sessions, from a routing perspective (routing is my current hot button). An SNA session can be defined as a succession of adjacent pairs of session-level processing points (at the half session layer). A new term, session stage, is coined to describe the connection between pairs of these processing points. Session stage partners can support adaptive pacing or fall back to fixed-pacing protocols, and so forth. So now we have an architectural definition for the function of an SNA router—an SNA router performs the functions of a session stage partner.

The Functionality

Published support has been added for several features that had crept into IBM's proprietary APPN product for the AS/400. Three are especially notable:

- One way conversations
- Limited-resource sessions
- Architected modes

One Way Conversations

I once spent an evening with a colleague debating the merits of LU 6.2 vs. LU 0 on the basis of being able to efficiently handle high volume one-way transactions. (As you might have guessed, this individual was responsible for airline reservation systems.) The answer? One way conversations—also called reliable one-way brackets. This new function of the resource manager (RM) allows one to implement transactions which do not require a response. (This was previously done by my friend via LU 0 pipelining.) To handle the case where transaction order is required, another new feature, conversation grouping, is added. A conversation group is a set of one-way brackets that can be logically related (ordered) by transaction programs.

Limited Resource Sessions

Another problem concerns how to manage transactions across switched facilities. When session traffic ends on a switched link (i.e., no more conversations are active), a new optional feature exists which allows the Resource Manager to set a timer, and if the timer expires all contention-winner sessions are terminated. This ultimately allows the control point to disconnect from the (potentially expensive) switched facility. This feature is called limited-resource session handling.

Architected Modes

Finally, several mode names have been in de facto standard use within existing IBM products. These names—default (8 spaces), BATCH, INTER, BATCHSC, INTERSC, CPSVCMG, and SNASVCMG—are now finally published. A small detail but important nonetheless.

Still To Come

So much for the new stuff. Now the future stuff. What would I like to see in APPC?

Functionality

The just-announced APPN T2.1 extensions (which I'll cover in a future *Architect's Corner*) satisfied much of my wish list, including registration, directory server, and name query.

However, I would prefer a better LAN implementation including, for example, better use of broadcast based (or group addressed) name resolution services. But, in general, the functionality set is pretty complete. Well, actually, I would like to also see extensions bringing SNA APPC in closer alignment with OSI DTP, but this will come in due (OSI) time.

APPC FAPL

What I really would like IBM to provide is a formal specification for APPC—a true Formats and Protocol Logic specification (FAPL). "But we have a FAPL," you say? No, not really. We haven't seen a true FAPL for any SNA protocol since the 1980 publication of SNA—*SC30-3112 - SNA Format and Protocol Reference Manual: Architectural Logic.* In IBM's specification formalism, a FAPL contains:

- Real pseudo code (FAPL is really the name of the specification language itself.)
- Real programmatic flow control
- Real variable and structure element references

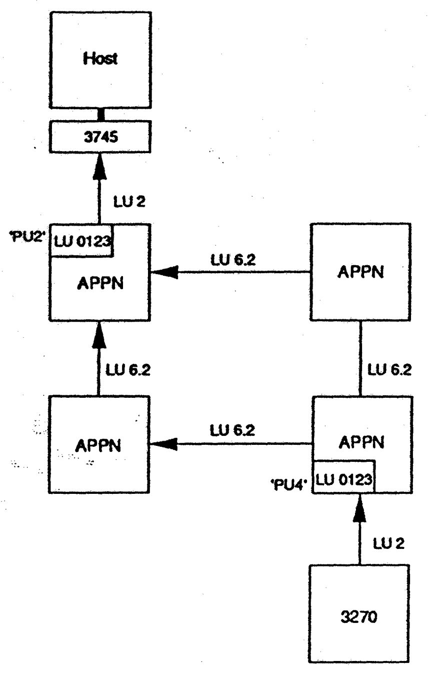
What we have instead are the extracted comments from the true LU 6.2 FAPL, which sits on the shelves of internal IBM developers.

Why would I like to see a real FAPL? Implementation integrity. There is always a risk that the FAPL comments (which are published externally) are out-of-synch with the real FAPL pseudo code. (When was the last time one of your programmers updated the code without changing the comments?) Also, no matter how carefully the comments are crafted, there are interpretations that simply cannot be translated to English—the real variables and structure references tell the real truth.

IBM positions the interoperability test service as one tool to help get around this problem—just test it and, if it works, it probably is alright. But, internally, IBM developers have the real FAPL to consult if questions about interpretation arise. If published, other developers would similarly benefit and the overall quality of APPC implementations would be correspondingly improved.

But with the level of detail that is now in the current specification, this is a nit. A wish but not a demand. Overall, APPC has become a very likeable architecture. ■

Correction:
In the April 1990 *SNA Perspective*'s Architect's Corner, the second figure contained some errors. This is the correct chart for Figure 2 on page 15. This figure shows LU 0123 tunneling, one of two ways discussed in that issue whereby 3720 (and other non-6.2 LU types) could be routed across APPN backbones. With tunneling, a PU 4 interface is presented downstream to cluster controllers and PU 2 end nodes, and a PU 2 or subarea interface upstream to communications controllers. The non-6.2 LUs are encapsulated in APPC sessions. It is a pragmatic transition solution. APPN routing services can be used to advantage for existing devices—cluster controllers, banking terminals, and retail store controllers—which make up the vast majority of devices on SNA networks today. ■



SNA Perspective Order Form

Yes! Please begin my subscription to *SNA Perspective* immediately. I understand that I am completely protected by CSI's 100% guarantee, and that if I am not fully satisfied I can cancel my subscription at any time and receive a full, pro-rated refund. **For immediate processing, call Cheryl Roberts at (408) 764-5136.**

Make Payable to CSI
☐ Check enclosed ☐ VISA
☐ Purchase order enclosed ☐ MasterCard
(P.O. # required) ☐ American Express

☐ Sign me up for 1 year of *SNA Perspective* at a cost of $350 (US$).
(International, please add $35 for airmail postage.)

☐ Sign me up for 2 years of *SNA Perspective* at a cost of $520 (US$).
(International, please add $70 for airmail postage.)

CSI - *SNA Perspective*,
ATTN: Cheryl Roberts
5201 Great America Parkway, Suite 3224
Santa Clara, CA 95054

Account Number ______
Expiration Date ______
Signature ______

I am authorized to place this order on behalf of my company. My company agrees to pay all invoices pertaining to this order within thirty (30) days of issuance.

Name & Title ______
Company ______
Address ______

______
City, State & Zip ______
Phone (______) ______

Copyright© 1991 CSI - Communications Solutions, Incorporated, all rights reserved. Reproduction is prohibited.• Subscription: U.S. - one year $350, two years $520. International - one year $385, two years $590 • SNA Perspective is published monthly by CSI • 5201 Great America Parkway • Santa Clara, CA 95054 • Telephone • (408) 562-6031 • Fax (408) 562-5745 • Managing Editor: Louise Herndon Wells • Associate Editors: Dr. John R. Pickens, Vincent Busam • Contributors: Anura Gurugé, Wayne Clark—Novell • The information and opinions within are based on the best information available, but completeness and accuracy cannot be guaranteed.