SNA Perspective

Volume 13, Number 5
May, 1992
ISSN 0270-7284

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

Integrating TCP/IP into SNA Part I: Network Connectivity

Many companies find themselves with networks using multiple protocols, especially SNA and Transmission Control Protocol/Internet Protocol (TCP/IP). As the end users of these networks increasingly need access to resources across these multiple networks, integration between the protocols becomes more important.

This article is the first in a series of articles intended to assist users considering this challenge from the point of view of integrating TCP/IP into traditional subarea SNA networks. We discuss the issues faced by the user in this process, not just in technology and functions, but also performance, reliability, and management. Several vendors offer products in this area but *SNA Perspective* has chosen to focus on IBM solutions. These solutions are addressed in three dimensions of integration: networks, systems, and applications. In this article, we focus on network and transport

(continued on page 2)

CPI-C Part II: IBM's Strategic API

Application developers, network developers, and end users are investing significant and increasing amounts of time and money creating and maintaining distributed applications. A major cause of schedule and budget overruns has been the inherent incompatibility of networked applications and their runtime environments. Coherent connections and portability of applications are therefore vital to distributed application resources. IBM and many others have for years been developing various application program interfaces (APIs) in an effort to move toward portability.

This article is the second of a two-part exploration of IBM's Common Programming Interface-Communications (CPI-C). Part one appeared in *SNA Perspective*, March 1992, focused on the benefits of common APIs, and compared CPI-C to APPC. In this article, we begin by noting five categories of transaction programs that are appropriate for CPI-C. We consider CPI-C's importance to IBM and third-party software developers. We analyze IBM's design for CPI-C as a common transaction processing interface for SNA and OSI, which IBM has recently said it will extend to TCP/IP. We then compare CPI-C to Remote Procedure Call (RPC) at a high level. We also consider CPI-C evolution, which will likely include additional SNA/OSI services and CPI-C support for TCP/IP applications and protocols.

(continued on page 10)

In This Issue:

(continued from page 1)

connectivity. We also provide a brief overview and background on TCP/IP and review IBM's evolving position on TCP/IP and its place in IBM's networking blueprint.

A future article will address system and application access and integration. In addition, it will grapple with the question of OSI, organizational responsibility for TCP/IP in the new IBM, and IBM's push to decouple applications from networks. Also in a future article in this series, we will explore how all of this has affected the experience of several end users, in both their decision process and implementation realities.

TCP/IP Series Focus

Because the topic of multiprotocol, multivendor integration is so multifaceted, we want to clarify for our readers the focus of this TCP/IP series.

Supporting SNA data flows across non-SNA networks such as TCP/IP is a very dynamic market today. *SNA Perspective* has addressed this topic in several articles recently (see *SNA Perspective,* January, April, May, June, July, and October 1991 and February 1992). The focus of this series, on the other hand, is TCP/IP integration into subarea SNA networks, either for access to systems in the SNA network or across it.

A significant percentage of IBM mainframes today have TCP/IP support and this percentage is rising quickly. *SNA Perspective* believes the majority of large U.S. companies with SNA are actively considering TCP/IP. In some of these installations, the mainframe with TCP/IP does not concurrently support SNA networking. Since the focus of this series is multiprotocol integration, these single-protocol environments are not addressed here.

Several vendors have offered SNA and TCP/IP support for years, including OpenConnect Systems (formerly Mitek), McDATA, NCR, and Network Systems Corporation. Again, for purposes of focus and based on reader input, in this series we examine IBM's position and products.

These other aspects will continue to be addressed in *SNA Perspective;* in fact, the other article in this issue looks at TCP/IP sockets support over SNA as well as interfacing CPI-C to TCP/IP transport.

The User's Challenge

Many users find themselves facing the challenges of integrating SNA and TCP/IP networks. Existing multiple networks may have resulted from a merger or from separate departmental decisions (e.g., finance and engineering departments or remote site LANs and a local SNA-based data center).

End users increasingly need access to resources across these multiple networks and the applications themselves need to share data, both of which create the need for multiprotocol internetworking. Many of the integration solutions can also address the related challenge of migration—the desire on the part of many users to migrate from subarea hierarchical SNA to a peer networking environment, for which some have selected TCP/IP.

The challenges faced by the user in this decision process go beyond the technical issues of interfacing or paralleling the two environments or even weighing the functionality of either stack's applications or network services. The decision must also consider total price over the life cycle of the purchase, including the cost of WAN bandwidth. A measure of the expected life-cycle cost of a purchase must take into account whether the solution chosen is a single-purpose solution or part of an architected approach. The former is often available sooner and at a lower cost, for example, but may have long-run drawbacks of incompatibility and limited upgradability.

Existing investments and network usage also affect choices: two alternatives may provide comparable functionality, but one may support the existing environment better or at least disrupt it less. Beyond the technology, "personality" differences between SNA users and TCP/IP users should not be ignored. Other important concerns include requirements for performance, reliability, manageability, support, and service. A future article in this series will look at the experiences of several users who have grappled with these issues.

IBM's Position

In the 1980s, IBM and many other vendors expected TCP/IP to fade away in the face of massive OSI migration. However, this migration failed to materialize in the face of slow OSI standards development and product availability along with user resistance to the high prices, low performance, and low functionality in the early OSI products. A large number of users, including many who had initially committed to an OSI strategy, found TCP/IP to be a viable, general purpose, multivendor connectivity and interoperability solution while waiting for OSI protocols and products to be fully functional and consistently available. Of these users, many have come to regard the TCP/IP application suite and underlying network services to be sufficiently rich in function as to exclude consideration of OSI as a replacement even in their long-term planning horizon.

IBM could not ignore the business opportunities for TCP/IP or the lower than expected income from OSI products either in the U.S. or in Europe. In June 1991, IBM Networking Systems Vice President and General Manager Ellen Hancock formally acknowledged the evolution that had been taking place internally at IBM for several years regarding the relative strategic or tactical importance of SNA, OSI, and TCP/IP: whereas TCP/IP was formerly considered a tactical step on the way to OSI, all three are now formally strategic architectures for IBM.

As a result, the company has significantly ramped up investment in TCP/IP and, *SNA Perspective* believes, is more quietly scaling back its OSI development efforts. A future article in this series will address the question of OSI and TCP/IP in greater depth, at both the application and network level. A generalized view of several IBM products that support IP traffic across a subarea SNA network, including XI, SNAlink, and NPSI are shown in Figure 1.

Expectations of IBM

SNA Perspective expects many TCP/IP announcements from IBM before the end of 1992, including a major multiprotocol blitz in May or June. We are also seeing some results from IBM's push to shorten every phase of the architecture-to-delivery cycle, especially in Networking Systems. So don't be surprised to find that a Statement of Direction, which used to mean "two years until announcement," can now be as short as three months.

We also expect fast movement from IBM to better integrate its TCP/IP products with SNA and with network management—continuing its coordination of NetView and SNMP. In addition, we expect new releases of existing IBM TCP/IP products will roll out more frequently. IBM knows it is playing catch-up with regard to being considered a serious contender in the TCP/IP market, but it is investing heavily to address this.

Networking Blueprint

As discussed in the April issue of *SNA Perspective,* IBM unveiled its networking blueprint in March. At the networking level, IBM has combined multiprotocol solutions into two groups:

- Multiprotocol backbone accessed via routers
- Single protocol backbone accessed via gateways

Currently, most multiprotocol solutions, including integrating TCP/IP into subarea SNA networks, are examples of the second type.

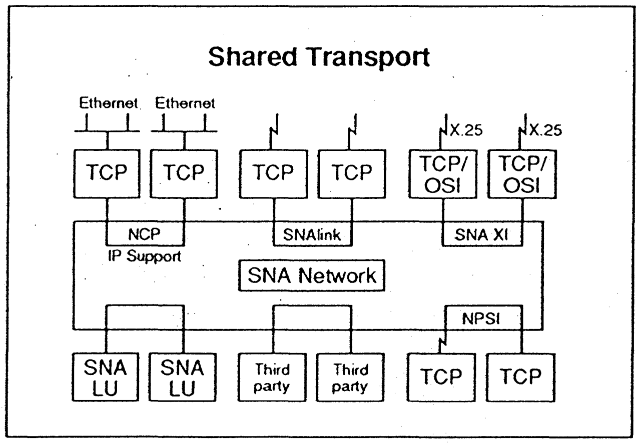

Figure 1

TCP/IP Overview

TCP/IP is a collection of layered network protocols defined during the late 1960s and 1970s by the U.S. Department of Defense (DoD) Advanced Research Projects Agency to support the ARPAnet. The original design goal of TCP/IP was to construct an interconnection of networks (an internetwork or internet) providing universal communication services among physical networks that each possess their own unique, technology-dependent interfaces. The ARPAnet has since been split into several government networks, including Milnet and the defense data network (DDN), as well as the large, public, unregulated group of subnetworks referred to as the Internet.

Early TCP/IP users were the DoD and its contractors, including many universities. Because of its presence in the university environment and because it was free (public domain) software, the University of California at Berkeley included TCP/IP networking as a standard component with its popular BSD version of UNIX. As a result, many UNIX users also became TCP/IP users by default, particularly the engineering/scientific community where UNIX was pervasive on workstations in the 1980s. More recently, TCP/IP popularity has also been moving into the business environment of corporations worldwide, especially in financial and process manufacturing companies.

TCP/IP "Standards" Process

The TCP/IP protocols were originally developed under contract as a government project more than twenty years ago. However, TCP/IP is continually upgraded today through an open process called Request for Comments (RFC), which is overseen by the Internet Activities Board (IAB) and the Internet Engineering Task Force. The IAB is not an official standards development body as ANSI or ISO are. However, the RFC process is becoming increasingly formalized and can certainly be considered a "standards" process. Further, several TCP/IP protocols and applications are under review as potential formal standards by the official standards bodies.

TCP/IP "Architecture"

Figure 2 gives a view of the TCP/IP layers. Strictly speaking, the TCP protocol provides transport functions and IP provides internetwork functions. However, all the elements in Figure 2 (and there are several more at the application level than those shown) are often collectively referred to as TCP/IP.

TCP/IP is a layered architecture. The internet services and protocols are shown in Figure 2 as modeled in four functional "layers" shown on the right of the figure—application services, transport, internetwork, and network interface and hardware. The seven layers shown on the left of the figure are a rough mapping of the various TCP/IP services and protocols to an OSI- or SNA-like model.

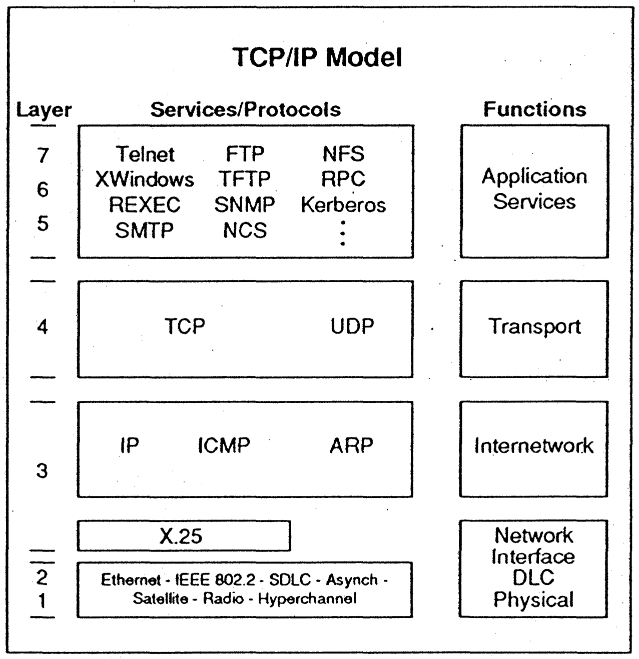

Figure 2

TCP/IP Protocols

The lower layer TCP/IP protocols are discussed here. The application services and protocols will be addressed in a future article in this series.

TCP and UDP

These application services and protocols can use either TCP or the user datagram protocol (UDP) as a transport mechanism. TCP provides a reliable, connection-oriented protocol, whereas UDP is unreliable and provides no flow control. TCP provides for reliable data transmission in order and supports error-free delivery with error-checking. TCP is used much more frequently than UDP for transport. It is a peer-to-peer, connection-oriented protocol. However, supporting applications usually select a client/server model of interaction.

Sockets—A TCP connection is defined and identified by a pair of sockets. Each connection is defined by four parameters—originating port and IP_address and destination port and IP_address. A TCP socket can be considered similar in function to an OSI transport service access point (TSAP) address.

IP

IP functions as the "layer" that makes the underlying physical network transparent by creating a virtual network view. IP is an unreliable, best-effort, connectionless packet delivery protocol. Any IP packets (datagrams) that are lost, received out of order, or accidentally duplicated will not be retransmitted by IP; all retransmission is a TCP issue. IP addresses are 32-bit, specify source and target hosts on the internet, and are in the logical form <network address><host address>.

Contrary to popular misconception, IP is a delivery protocol which acts as a routing protocol only to the next hop in the network. Interconnected (multihop) networks need IP gateways. IP datagram services are provided in an internet through IP gateways that exchange routing information using routing protocols such as Border Gateway Protocol, Exterior Gateway Protocol, HELLO protocol, Routing Information Protocol (RIP), the newer Open Shortest Path First (OSPF), and the emerging OSI intermediate system-to-intermediate system (IS-IS).

ICMP

IP and the Internet Control Message Protocol (ICMP) together are functionally equivalent to OSI layer 3c (independent convergence) because IP routes messages across multiple networks in connectionless (best-effort) fashion.

ICMP is used by IP gateways or destination hosts to communicate with the source host to report control information, for example, errors in datagram processing. While ICMP uses IP as if IP were a higher-level protocol, ICMP is integral to IP and is implemented by every IP module.

X.25

X.25 in Figure 2 corresponds to OSI layer 3a (subnetwork access protocol). It is important to note that X.25 is not the only subnetwork protocol that can be used at layer 3a beneath the IP. For example, an SNA path control subnetwork could be used if an appropriate connection were made.

ARP

The Address Resolution Protocol (ARP) is used on LANs to map IP addresses to physical hardware addresses. Reverse ARP (RARP) is also used over LANs but, in this case, the hardware address of the device on the LAN is known and the IP address is the queried parameter.

Dimensions of Integration

SNA Perspective has chosen to examine the integration options between SNA and TCP/IP from three dimensions—network, applications, and systems.

- Network connectivity: support TCP/IP access to or across the subarea SNA network—which this article addresses—and/or SNA in the TCP/IP backbone
- Application connectivity: support for multivendor applications, which itself has two major aspects:
  - Application to network interface: support interfaces for the applications from one environment to run over the other network, such as TCP/IP sockets support over LU 6.2
  - Application gateways: support gateways between applications such as an SMTP to PROFS gateway.
- System connectivity: support for multivendor protocols across various platforms

The remainder of this article examines IBM solutions for TCP/IP support in SNA networks from a network connectivity perspective. A future article in this series will consider the application and system dimensions.

Network Connectivity

Network connectivity deals with two issues. First, the user wants, to the greatest extent possible, to leverage existing investment—expanding the use of today's boxes and links to support increased capabilities or access. Second, the user has an eye toward a long-term goal of a common network infrastructure, even if made up of multivendor and multiprotocol components.

If IP support across SNA were the only function provided by the hardware and software, several of these solutions discussed below could be considered prohibitively expensive. However, these network connectivity solutions are intended to be leveraged solutions—taking advantage of existing products to increase connectivity.

The next sections look at new and existing IBM support for TCP/IP on today's traditional SNA network components, including the host, the communication controller, the newer 3172 LAN interconnect controller, the 3174, and the RS/6000. There are many other single solution gateway-type products from IBM users may consider, but they are beyond the scope of this article. In addition, as discussed above, there are many products from other vendors that operate on either these same IBM platforms or on separate platforms, ranging from full UNIX and TCP/IP support on IBM hosts to LAN gateways, but the focus of this article is limited to IBM solutions.

SNAlink: A Host-Based Solution

SNAlink is a function in TCP/IP for VM and TCP/IP for MVS that allows hosts to route TCP/IP messages over SNA LU 0 sessions. *SNA Perspective* believes SNAlink will be enhanced to support LU 6.2 sessions. Hosts can interconnect their TCP/IP applications over either IP internets or an SNA backbone network, as shown in Figure 3. In the case of SNA backbones, these host-based TCP/IP applications send IP datagrams to SNAlink to be routed over the SNA network.

In addition to host-to-host TCP/IP application interconnection, SNAlink allows the hosts to route traffic from devices on a TCP/IP network attached to one host across an SNA backbone to devices on a

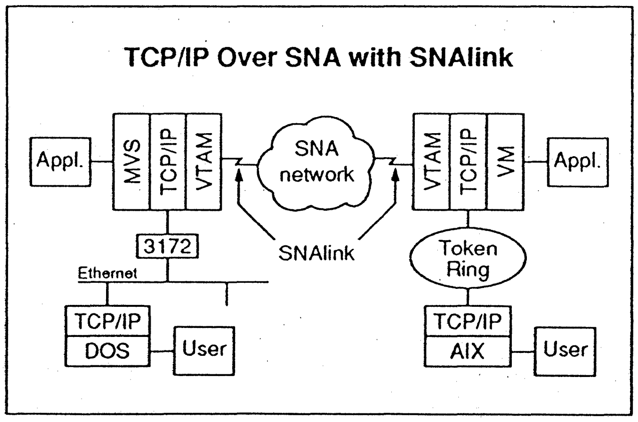

Figure 3

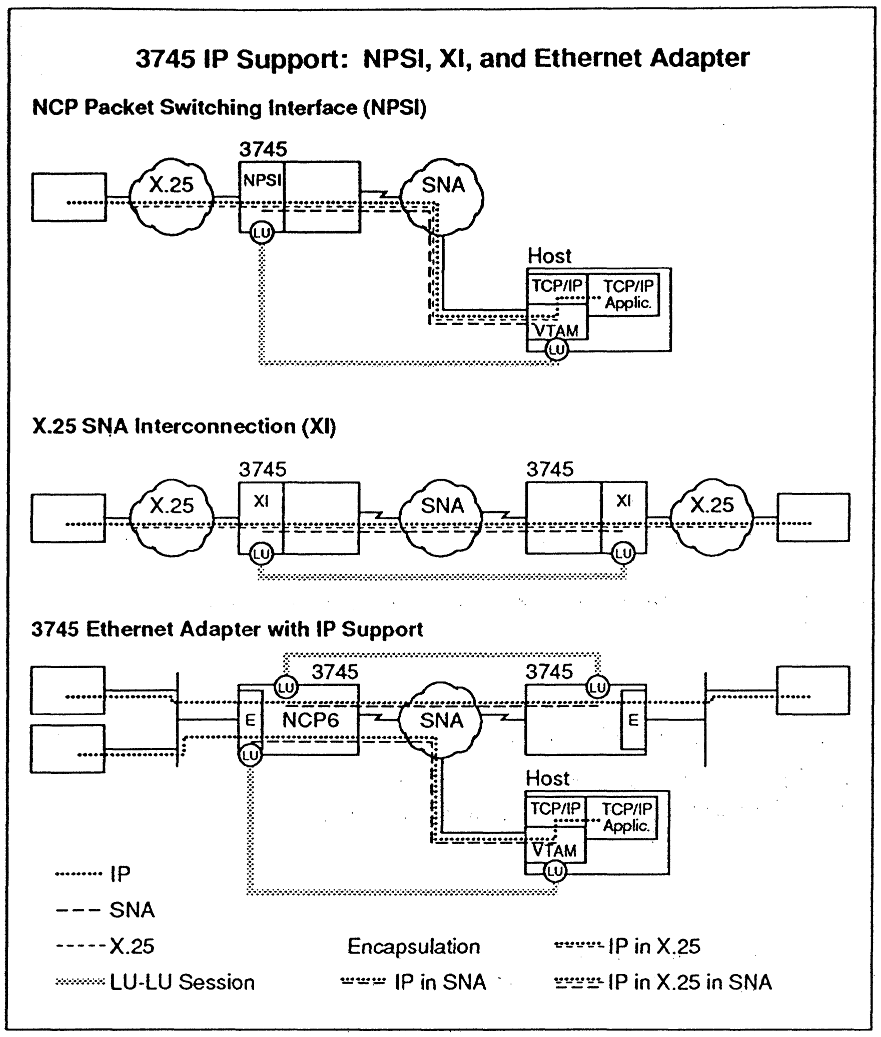

Figure 4

TCP/IP network attached to the other host. These possible SNAlink configurations are shown in Figure 3.

Supported application services in this scenario include:

- Remote login with Telnet from any user to any TCP/IP platform over SNA
- File transfer with FTP to any TCP/IP platform (except DOS) over SNA
- Electronic mail with SMTP from any user to any user over SNA
- AIX user access to files on MVS or VM using NFS over SNA

3745 IP Support

The IBM communication controller has long had support for X.25 interfaces and much TCP/IP traffic accesses SNA networks through X.25 networks. The two primary IBM products in this area are NCP Packet-Switching Interface (NPSI) and X.25 SNA Interconnection (XI). In addition, IBM announced in June 1991 that it will provide an Ethernet adapter with IP support for the 3745.

Each of these options is depicted in Figure 4 and described below. Although the figure shows the three products configured across an SNA backbone, the reader should note that, in each case, both ends of the interface can exist inside the same 3745. Also, although the SNA host in a subarea network is involved in session startups, several of these 3745 solutions can, with an active session, support IP traffic without host intervention.

NPSI

NPSI runs with the Network Control Program (NCP) in a 37xx communication controller. NPSI has for several years primarily supported connection of SNA-to-host as well as non-SNA-to-host networks over X.25. NPSI can be used support the transport of TCP/IP data over SNA/X.25 connections, usually for access from a device on an X.25 network to resources on an SNA network host.

X.25 SNA Interconnection

XI, like NPSI, runs with NCP. XI provides connection between X.25 devices over an SNA transport network. TCP/IP applications quite often interconnect over WANs using X.25. XI provides these

TCP/IP-X.25 connections over SNA through encapsulation of the X.25 packets within SNA over subarea virtual routes.

3745 Ethernet Adapter

Ethernet LANs have long been popular in TCP/IP environments but, although the 3745 communication controller has supported a token ring for many years, it has not had an IBM-supplied connection to Ethernet. During 1991, IBM announced an Ethernet adapter with IP support for the 3745. To enable this functionality, the user must upgrade to NCP Version 6, which will ship in September 1992. *SNA Perspective* believes that the NCP V6 TCP/IP support will be based on SNAlink and will communicate with host SNAlink.

The 3745 Ethernet Adapter with IP Support will be able to either route IP traffic across an SNA network or route IP traffic to a host-based TCP/IP application, as shown in Figure 4. The user should note that the 3745 Ethernet adapter does *not* support SNA over LLC2 traffic from Ethernet as the TIC does from a token ring. *SNA Perspective* does not expect IBM to ever add this support.

3172 IP Support

The 3172 LAN interconnect controller is a networking product intended to allow fast transparent data flow from LANs (token ring, Ethernet, and FDDI, to date) across a channel connection to the host. It was announced in October 1989 with TCP/IP support; SNA support was added in September 1990. The 3172 is optimized for TCP/IP traffic and thus its performance with TCP/IP is higher than with SNA.

The 3172 is intended primarily for LAN to host connection. In Figure 5, the four nodes shown—A, B, X, and Y—could each access applications on the host. However, Figure 5 is intended to illustrate that the 3172 can participate in LAN to LAN communication as well, but host support is required.

The 3172 performs no protocol processing of the data, as shown in Figure 5. Processing for either SNA or IP is performed inside the host. For SNA traffic, the IEEE 802.2 MAC layer is terminated in the 3172. This termination slows the 3172's performance for SNA but also allows the SNA traffic to access multiple hosts through the 3172, which TCP/IP traffic cannot do.

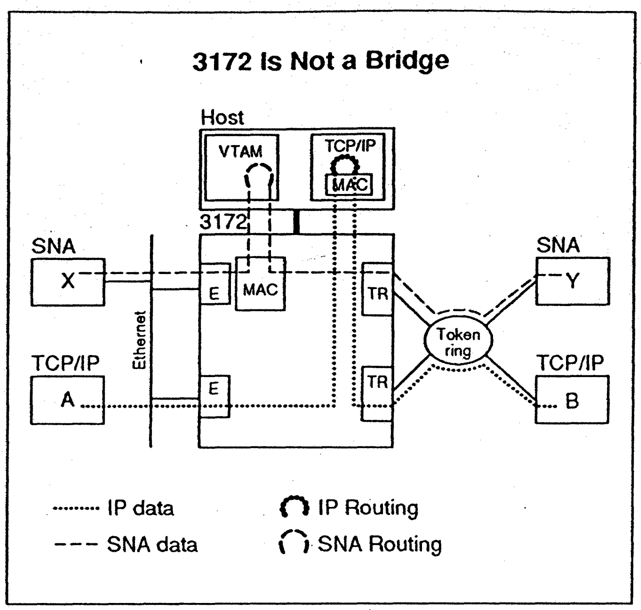

Figure 5

Note that, if the user wants to support both SNA and IP traffic simultaneously from the same LAN, two LAN adapters are required. In contrast to the 3172, the 3745 does not support SNA over its Ethernet adapter nor TCP/IP over its token-ring interface coupler (TIC).

3174

In March, IBM announced the availability of an optional software component, called TCP/IP for 3174 Function, which allows a 3174 to support Telnet sessions from either 3270 and ASCII terminals over token ring to TCP/IP servers anywhere on the LAN/WAN environment accessible from that token ring—but not across SNA. (The 3174 already supports Telnet for DOS and OS/2 workstations with TCP/IP.)

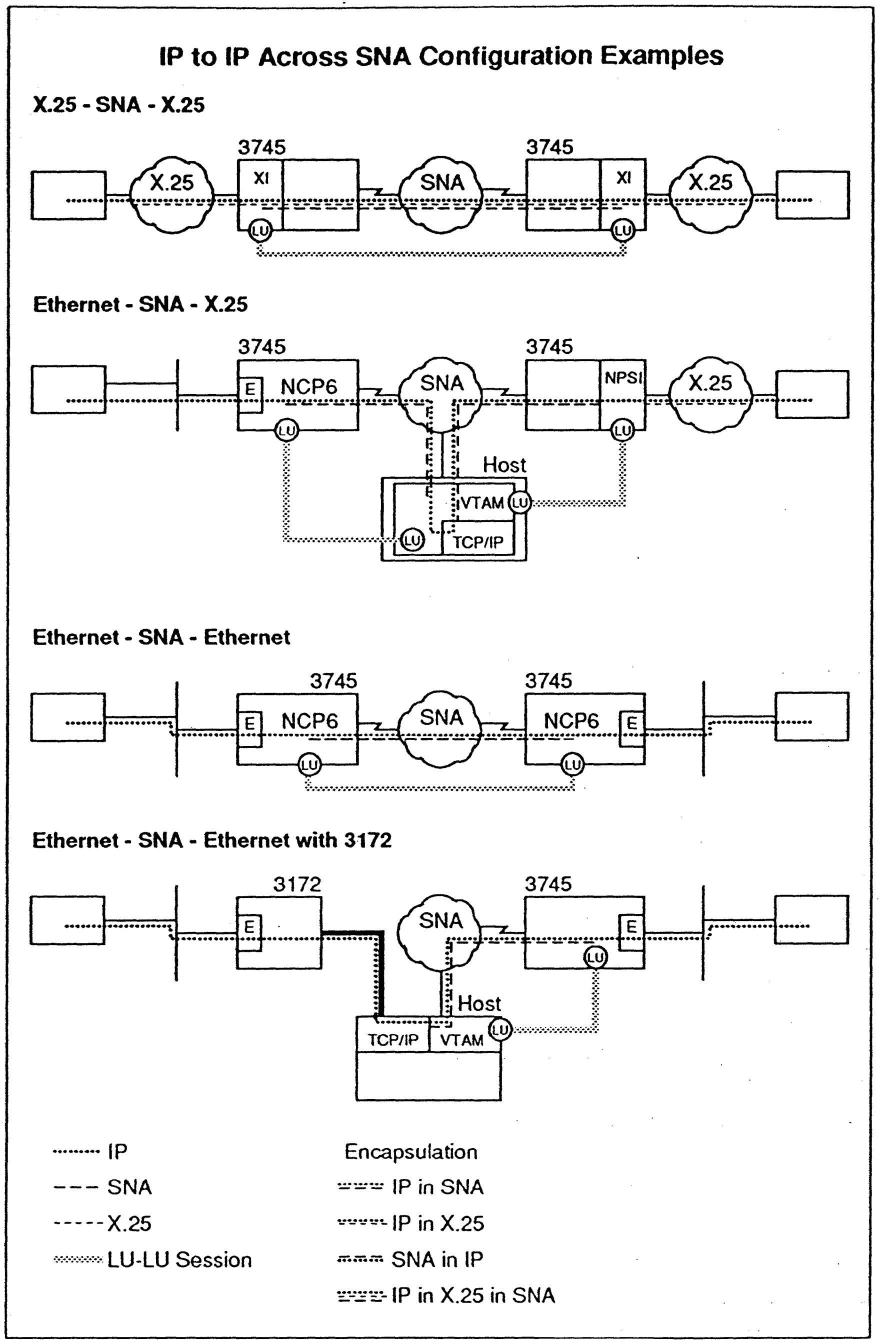

Figure 6

To use the Telnet support for terminals, the 3174 must also have 3174 Peer Communication Support Program. Peer Communication provides a bridge between the 3174 and a token ring and allows workstations coax-attached to the 3174 to communicate with each other as if through a LAN.

RS/6000

The RS/6000 is also emerging as an option for TCP/IP connectivity into SNA. In February, IBM announced channel support for the RS/6000. This support is limited to parallel channels and not the newer ESCON fiber channels. Since the RS/6000 has a full TCP/IP suite as well as available connections to Ethernet, token ring, and X.25, this provides the basic data link connectivity needed for TCP/IP support. At the same time, IBM announced a host software driver for VM that will support TCP/IP communication with the RS/6000 across the channel, and has indicated that it will extend this support to MVS.

Additional Configurations

The above products can be combined in several ways to support TCP/IP access to or across an SNA backbone, as shown in Figure 6. As with Figure 4, the examples here are shown with two 3745s and an SNA backbone but a single 3745 with access to both networks could provide the same functionality.

- TCP/IP from X.25 to X.25 connection across SNA is supported by XI.
- TCP/IP from Ethernet to X.25 connection across SNA is supported with a combination of a 3745 Ethernet adapter, SNAlink, and NPSI.

(continued on page 20)

(continued from page 1)

The Benefits of CPI-C

Part one of this series includes a list of thirteen benefits of common APIs in general and of CPI-C specifically. CPI-C is a valuable tool because it provides a consistent API for applications that require interprogram connections and resource sharing. In summary, the major benefits of CPI-C include the following:

- Underlying session and network resources and states are program-transparent
- Programs communicate by using standard calls
- These calls are platform-independent

IBM to Promote CPI-C

IBM needed an API like CPI-C that could run over both subarea SNA and APPN as well as interface to multivendor systems. IBM knows that it exists in a multivendor world and, to encourage its customers to maintain their investment in IBM systems, it must provide options that both support existing IBM environments and incorporate open systems.

IBM has realized in the past year or two that it also needs to support and promote APPC and CPI-C to both users and third-party developers with publicity, training, sample programs, technical and marketing support, and forums for interaction. Previously, the company seemed to be operating from an assumption that the market would automatically embrace new IBM solutions. In reality, however, although some adopted APPC, many users and developers were increasingly migrating to non-IBM alternatives. These migrations sometimes led to removing SNA networks that could not interoperate.

To date, IBM has not been promoting CPI-C as heavily as APPC but *SNA Perspective* expects that to change during 1992. Of the approximately forty third parties supporting APPC today, about a fourth have also committed to supporting CPI-C. We expect to see some of these products in 1992 and expect that other APPC developers will announce CPI-C plans. However, we also sense that third-party developers are tentative in their commitment to CPI-C. It is incumbent upon IBM to prove to them that a strong market exists for CPI-C in the face of competition from RPC, message queueing, and other middleware APIs.

Terms: LU 6.2, APPC, and CPI-C

It bears repeating from the first part of this series that IBM now uses the terms LU 6.2 and APPC interchangeably. LU 6.2 used to refer to the layer 4, 5, and 6 protocols while APPC referred to the interface or API services used to access LU 6.2. That API is now called, by IBM, the "native APPC API," which IBM is endeavoring to replace with CPI-C because of the problems and limitations described in part one of this series.

This is a marketing redefinition on IBM's part and not a technical change. *SNA Perspective* believes this redefinition serves two purposes for IBM. First, the non-SNA world, unfamiliar with logical units, finds the term APPC more comfortable. Second, IBM can say that CPI-C interfaces to APPC rather than replaces it.

CPI-C, then, operates as an API to the LU 6.2/ APPC protocol. LU 6.2, in turn, natively runs over APPN or can operate as a dependent LU in subarea SNA networks. It has also been adapted to run over OSI through OSI TP, as discussed below under CPI-C as SNA/OSI Integrator. IBM also stated in March that it is mapping CPI-C to run over TCP/IP as well as mapping TCP/IP sockets applications to run over APPC and APPN, a capability it informally calls SNAckets.

A CPI-C application on one side of a conversation can interact with an APPC API application on the other side. Therefore, it is *rarely* necessary for either the user or other vendors to change their existing APPC applications if CPI-C is added to the applications with which they interact. However, the user should be aware that not all CPI-C functionality is included in APPC and not all APPC functions are supported in CPI-C. Therefore, some existing APPC applications may need to be changed if CPI-C is added on one side. Also, CPI-C applications may not be able to use all the CPI-C functionality when interfacing to an APPC application.

Transaction Environments

As we illustrated in part one of this analysis, CPI-C is a simpler interface than the native APPC API. CPI-C is designed as a platform- and operating environment–independent interface through which transaction programs can interact. Five types of transactions that can use CPI-C, the APPC API, or other APIs designed for transaction environments are described below.

Inquiry Transaction

This transaction type is typically used to request information from a server program and is also referred to as "request and reply." It is called "request and reply" because the entire conversation consists of one request and one reply. Inquiry transactions are often used by banks and retailers, and support remote procedure calls, database applications, and status queries.

Credit Check Transaction

This transaction type uses confirmed delivery. In this approach, a client program requests permission to perform a specific function. The major distinction between a credit check transaction and an inquiry transaction is that, in a credit check transaction, no data is returned by the server if the credit check is successful and there are no flags; the server simply returns an acknowledgement that the client program data was successfully processed. There will be cases where a credit authorization request is rejected. In such cases, an additional data flow that conveys the rejection of confirmation is generated in reply.

Database Update Transaction

This transaction supports a client program that requests information from a database server, updates the information, and returns the updated data to the server to be written to the database. A relational database using structured query language (SQL) calls is the most frequently supported multivendor kind of database for this transaction type. The database update transaction is also called a "conversation reply" transaction because the reply invokes an additional reply from the operator.

File Transfer Transaction

The file transfer transaction is really an extended version of the credit check transaction and is called "batch send." File transfer as a transaction type over LU 6.2 is used to send relatively large volumes of data. SNA Distribution Services (SNADS) provides a general-purpose object delivery service for this transaction. Delivered objects can include binary data, text, graphical data, image data, mixed-object data, files, revisable-form documents, final-form documents, or software distributions.

Pipeline Transaction

The pipeline transaction is also called a "one-way bracket." It is supported when the client or server program is resource-constrained. Data is sent, the sender removes itself from the conversation, and there is no positive acknowledgement of delivery of the data. The expression "one-way bracket" is used for a pipeline transaction because SNA data flow control (SNA layer 5) creates a bracket strictly for a one-way flow of data. This is also called a "one-way" conversation. The corollary expression in TCP/IP networks is a datagram (see TCP/IP article in this issue).

A major use of the pipeline transaction is found in advanced peer-to-peer networking (APPN) environments when network nodes transmit topology updates and conduct network-wide nodal resource searches. In this case, the objective is to minimize extended use of session and conversation resources.

Another use of the pipeline transaction occurs when a relatively large number of independent units of work (transactions) need to be accomplished but it is not possible to concurrently allocate sufficient sessions and conversations to run these in parallel. The general remedy in this case is to write a pipeline transaction program that uses two conversations—one that sends data and one that receives data.

CPI-C Platforms

IBM has implemented CPI-C on all its strategic SAA and AIX platforms including:

- CICS/ESA
- IMS/ESA
- MVS/ESA
- VM/ESA CMS
- OS/400
- OS/2, with NS/2 or Extended Services
- DOS
- AIX, RS/6000 (statement of direction)

Figure 8 shows that CPI-C–interfacing applications running on these platforms are able to share data for transactions independent of platform. That is, transaction programs running in any of these environments can issue send and receive calls in a consistent way without regard to underlying operating system environments.

CPI-C States and Calls

Conversation States

Programs written to make use of CPI-C are designed from the perspective of the remote program—that is, a local program issues a CPI-C call for a given conversation on the assumption that the remote program will issue another CPI-C call for the same conversation. This design approach to CPI-C conversations (and, for that matter, APPC conversations) gives rise to the use of conversation states. The state of a conversation determines the subsequent set of actions within that conversation.

CPI-C conversation states are shown in Table 1. A CPI-C conversation between a local and a remote program can be in only one of the states listed in Table 1 at any given time.

CPI-C Calls

CPI-C programs communicate by using program calls that are functionally equivalent to APPC conversation verbs. CPI-C calls establish the characteristics of a conversation and subsequently enable the exchange of data and control information between programs. These program calls are categorized as starter set calls and advanced function calls.

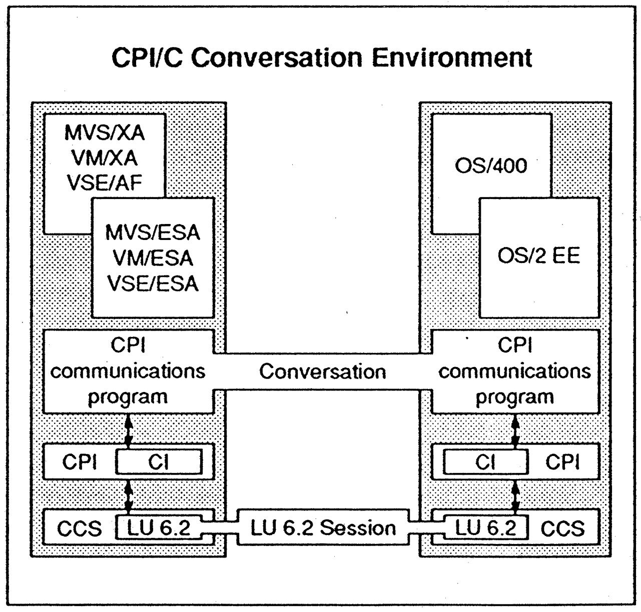

Figure 8

CPI-C Conversation States

State	Description
RESET	No conversation
INITIALIZE	Initialize_conversation successful completion. Conversation_ID has been assigned.
SEND	CPI communications program can send data on this conversation.
RECEIVE	CPI communications programs can receive data on this conversation.
SEND-PENDING	CPI communications program has received both data and a send indication on the same receive call.
CONFIRM	Remote program has issued a confirm call and is waiting for the local program to issue confirmed.
CONFIRM-SEND	Both a confirmation request and permission-to-send received. Remote program has issued prepare_to_receive. Local program has issued confirmed, and enters send state.
CONFIRM-DEALLOCATE	Both a confirmation request and deallocation notification received. Local program issue confirmed, conversation is deallocated.

Table 1

Program calls establish conversation characteristics and exchange data as well as control information between programs. The major program calls, both starter set and advanced function calls, are shown in Tables 2 and 3.

Starter set calls provide for simple communication of data between two programs and assume that the program uses the initial values for the CPI-C conversation characteristics. Default CPI-C conversation characteristics which are based on successful completion of the Initialize_Conversation call are given in Table 4 (see page 14).

All starter set and advanced function calls are in the general syntax form of:
CALL CMINIT*(conversation_ID, sym_dest_name, return_code),* where this is an example of the use of the Initialize_Conversation call.

CPI-C Program Calls

Starter Set

Calls	**Description**
INITIALIZE_ CONVERSATION	Initialize conversation characteristics
ACCEPT_ CONVERSATION	Accepts incoming conversation
ALLOCATE	Establishes conversation
SEND_DATA	Sends data
RECEIVE	Receives data
DEALLOCATE	Ends conversation

Advanced Function General

Calls	**Description**
CONFIRM	Sends confirmation request to partner program
CONFIRMED	Sends confirmation reply to partner program
FLUSH	Flushes LU send buffer
PREPARE_TO_ RECEIVE	Changes conversation from send to receive state to receive data
SEND_ERROR	Notifies partner program of error
TEST_ REQUEST_TO_ SEND_RECEIVED	Determines whether or not remote program is requesting to send data

Table 2

Advanced function calls generally provide more specialized processing than is provided for by the default set of conversation characteristic values

CPI-C Advanced Function Calls

Extract Calls

Calls	**Description**
EXTRACT_ CONVERSATION_ TYPE	Used to view current conversation characteristic (mapped, basic)
EXTRACT_ MODE_NAME	Used to view current mode name (network properties: COS, enciphered data)
EXTRACT_ PARTNER_ LU_NAME	Used to view same
EXTRACT_ SYNC_LEVEL	Used to view same (none, confirm)

Set Calls

Calls	**Description**
SET_ CONVERSATION_ TYPE	To mapped or basic
SET_ DEALLOCATE_ TYPE	To sync_level flush, confirm or abend
SET_ERROR_ DIRECTION	Specifies direction of data flow in which program detected error
SET_FILL	LL (logical records) or buffer
SET_LOG_DATA	For LU system error loss. Formatted by sending LU into error log GDS variable
SET_MODE_ NAME	Network properties
SET_PARTNER_ LU_NAME	Specifies the partner LU name
SET_PREPARE_ TO_RECEIVE_ TYPE	Sync_level, flush or confirm
SET_RECEIVE_ TYPE	Receive and wait or receive immediate
SET_RETURN_ CONTROL	When_session_allocated or immediate
SEND_SET_ TYPE	Buffer_data, send_and_flush, send_ and_confirm, send_and_prepare_to_ receive, send_and_allocate
SET_SYNC_ LEVEL	None or confirm
SET_TP_NAME	Specifies remote program name (1-64 bytes)

Table 3

shown in Table 4. Advanced function calls provide for a greater degree of synchronization and monitoring of data than starter set calls.

A CPI-C Conversation

Figure 9 (see page 15) gives an example of the interprogram flows between programs and CPI-C in a two-way conversation. The two-way conversation provides for an interactive data flow for send and receive between the program pair.

CPI-C As SNA/OSI Integrator

In August 1990, IBM published a paper entitled "Mapping IBM's CPI for Communications onto OSI Distributed Transaction Processing Services." This paper describes a mapping between CPI-C and the kernel, polarized control, and handshake functional units described in the *OSI Distributed Transaction Processing Part 2: Service Definition,* Sept. 1989.

Default CPI-C Conversation Characteristics

Characteristic Name	Initialize_Conversation Setting	Accept_Conversation Setting
Conversation_Type	Mapped_Conversation	Value received on startup request
Deallocate_Type	Deallocate_Sync_Level	Deallocate_Sync_Level
Error_Direction	Receive_Error	Receive_Error
Fill	Fill_LL	Fill_LL
Log_Data	Null	Null
Log_Data_Length	0	0
Mode_Name	From side information Sym_Dest_Name startup request arrived	For session where conversation
Mode_Name_Length	Length of Mode_Name	Length of Mode_Name
Partner_LU_Name	From side information Sym_Dest_Name	For session where conversation startup request arrived
Partner_LU_Name_Length	Length of Partner_LU_Name	Length of Partner_LU_Name
Prepare_To_Receive_Type	Prep_To_Receive_Sync_Level	Prep_To_Receive_Sync_Level
Receive_Type	Receive_And_Wait	Receive_And_Wait
Return_Control	When_Session_Allocated	Null
Send_Type	Buffer_Data	Buffer_Data
Sync_Level	None	Value received on startup request
TP_Name	From side information referenced by Sym_Dest_Name	Null
TP_Name_Length	Length of TP name	0

Table 4

A wide range of CPI-C calls can be mapped to OSI TP requests, indications, and parameters. This is not surprising because the OSI TP model, services, and protocol are based, to a great extent, on LU 6.2 logic. LU 6.2 could therefore support SNA layers 6, 5, and 4, as well as an OSI TP-ASE interface. A common LU 6.2/OSI TP application interface can be provided across IBM platforms and allow upper-layer SNA/OSI services to share common network and link services. This CPI-C call mapping between SNA/LU 6.2 and OSI TP was illustrated on page 17 of *SNA Perspective,* March 1992 in "IBM Makes Partners of SNA and OSI."

The remote transaction processing standard (ISO 10026) defines transaction processing and a framework for coordinating across multiple transaction processing resources in an OSI environment. The OSI distributed transaction processing model provides the basis for multivendor interactive applications. Transaction processing services include:

- Interaction partitioning between OSI application processes
- Dialogue establishment, control, and termination
- Complex commitment and rollback of multiple transaction resource types

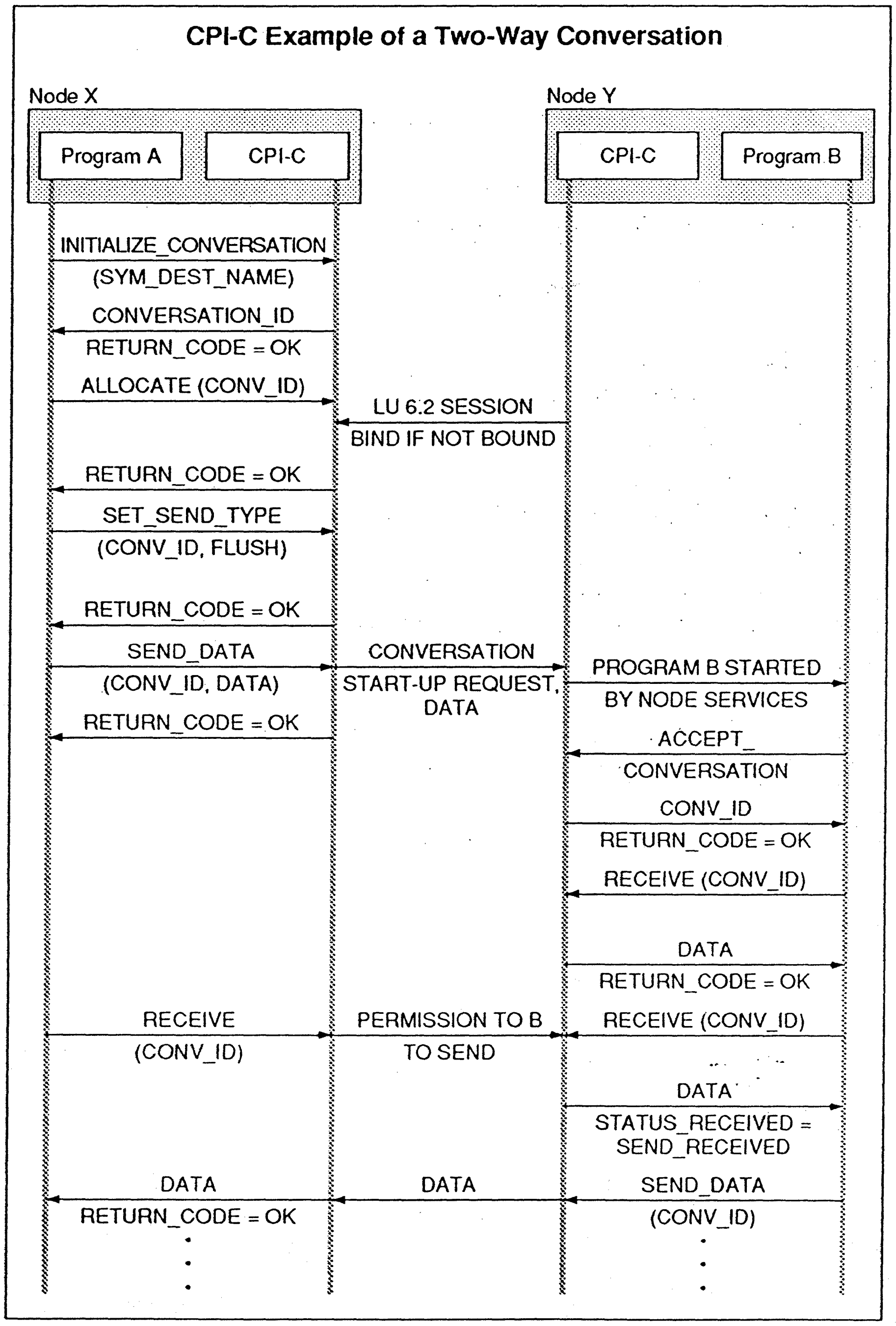

Figure 9

Beyond Transaction Processing

This SNA/OSI integration for transaction processing could be taken further, for example, to incorporate support through CPI-C for remote database access (ISO 9576), remote operation service element (ISO 9072, the basis of remote database access), ACSE (ISO 8649/8650), and commitment, concurrency, and recovery (ISO 9804/9805).

IBM has also said that it will enhance the CPI-C interface to enable protocol-independent application calls to and from a variety of underlying networking architectures including SNA, OSI, TCP/IP, and NetBIOS. A significant advantage of such a common, architecture-independent API approach would be that application developers could write applications that utilize networking protocols transparently—without the need to encode architecture- dependent syntax into communications modules.

X/Open Endorsement

The utility of CPI-C with regard to platform and operating system environment independence has not gone unnoticed in the industry. In 1990, X/Open adopted CPI-C as part of its overall mission to implement practical open systems. Specifically, CPI-C is now incorporated by X/Open as part of the *X/Open Portability Guide*—XPG3. This guide contains an evolving portfolio of APIs that significantly enhance portability at the source code level.

X/Open CPI-C is incorporated in the X/Open Common Applications Environment (CAE) which provides the API to allow X/Open-compliant systems to communicate with other processing environments that implement LU 6.2. This endorsement will prove particularly valuable to users needing interoperability between mainframe-based applications and other platforms and for any environment currently using APPC.

CPI-C and RPC

RPC, like CPI-C, also works well in distributed transaction processing and database environments. However, RPC does not have as rich a set of interface capabilities. Therefore, RPC is not as robust as CPI-C for complex transaction environments; on the other hand, it is also not as complex to use.

CPI-C: Conversational Model

CPI-C provides support for the conversational model which, in turn, supports distributed transaction processing. The conversational model is essentially based on send and receive functions through a logical connection called a conversation or dialog. APPN provides the underlying connection in a distributed network.

Database connectivity, read/write, commits, and rollbacks are often provided for in the conversational model supported by CPI-C and through SAA CPI-Resource Recovery. CPI-Resource Recovery provides transaction protection through establishment of synchronization points to protect database resources. These CPI-Resource Recovery functions, in turn, are enabled over LU 6.2 sessions which are then bound, maintained, and unbound over APPN connections.

RPC: Call/Return Model

RPC is based on a call/return model—the requester issues calls as if the server were local. RPC is used for process-to-process communication and basically supports a single request followed by a single response. The RPC model is based on an approach that allows individual procedures within an application to run elsewhere on the network. RPC presents the procedure call construct and generalizes this capability from a local application platform to a network of platforms. The resulting networked call procedure is implemented in a client/server architecture.

RPC in OSF, ANSI, and ECMA

RPC has been adopted by the Open Software Foundation (OSF) for its distributed computing environment (DCE). IBM had proposed CPI-C to OSF to be included in DCE, but it was not among the elements selected for inclusion. Both the American National Standards Institutes (ANSI) and the European Computer Manufacturers Association (ECMA) have generated RPC draft standards.

Expected CPI-C Enhancements

SNA Perspective expects that IBM will continue to enhance the functionality of CPI-C and LU 6.2. These enhancements will include additional SNA, OSI, and TCP/IP services over time. Some expected CPI-C directions include the following:

CPI-C over TCP/IP

IBM has stated that it is developing the capability to map CPI-C and LU 6.2 to run over TCP/IP. CPI-C over TCP/IP will probably be among IBM's 1992 multiprotocol announcement barrage, but we do not expect IBM to ship a product supporting it until mid-1993 at the earliest. This is certainly not a trivial problem—one of the architectural challenges, for example, is that LU 6.2 wants block or record service from the underlying network while TCP/IP expects streams. IBM representatives have said that the company has architected a solution for this mapping as part of it's networking blueprint through what IBM calls common transport semantics (see *SNA Perspective*, April 1992).

Improved server support

Among the user-requested features we expect IBM to add soon to CPI-C are non-blocking calls and support for multiple incoming conversations. The ability to support multiple incoming conversations is dependent on IBM following through on its plans to add full duplex support to LU 6.2 and that time frame is unclear. In addition to support for multiple incoming conversations, full duplex LU 6.2 will allow multiple local programs to send and receive at the same time. With full duplex, LU 6.2 will be in even closer alignment with the OSI TP standard.

X.500 directory services

We expect CPI-C to provide support for OSI X.500 directory services. X.500 is one of the major OSI application layer services and is a fundamental element of true distributed computing. Although X.500 itself and OSI applications as a whole are not major revenue sources today, *SNA Perspective* believes that X.400 messaging and X.500 directory services will be the first OSI layer seven services to find market acceptance. This is also significant because, to date, IBM has only supported transaction processing layer seven services over CPI-C. This would be a first step toward generalizing CPI-C as a common API across a range of application services, a trend that we discussed in depth in *SNA Perspective*, December 1991.

Automatic data conversion

From the early days of data processing, IBM used the EBCDIC character set while most other vendors standardized on ASCII data. (The IBM PC was a maverick in many ways, including being an ASCII device.) Today, CPI-C does not support automatic data conversion and native APPC APIs either don't support it or do so in a relatively primitive fashion. Support for automatic data conversion is a basic but very necessary element for CPI-C to provide the multivendor support it claims.

Summary and Conclusions

CPI-C is an improvement over the native APPC API since it is more platform-independent, as was detailed in the first part of this two-part series. *SNA Perspective* recommends that users and developers implementing LU 6.2 applications use the CPI-C interface.

SNA Perspective believes that CPI-C will be continually enhanced to support applications communicating over several intervening networking environments. Currently, it supports APPN, SNA, and OSI. IBM has stated that it will also run over TCP/IP as a part of its networking blueprint. *SNA Perspective* believes it will also interface CPI-C, through a similar structure, to NetBIOS.

It is IBM's hope that, eventually, CPI-C will increasingly find its way into the standards process, including ISO/CCITT. However, IBM will have to promote it heavily and create significant market momentum first. *SNA Perspective* expects to see significant IBM promotion of CPI-C as well as APPC during 1992, both to users and to developers.

SNA Perspective believes that IBM and others will increasingly adopt both CPI-C and RPC in addition to APIs based on a third model, the message queueing interface. As with other areas of communication, no single solution will emerge as the victor—the user is again faced with integrating several incompatible "solutions" that were each intended to solve another incompatibility problem.

The ultimate goal of all these common APIs, coming to be called middleware, is to present users and their applications with consistent interfaces that are simple and predictable and to decouple the underlying networked environment from the processor platform or operating system environment. Architected approaches to decoupling the network from the application in SNA environments will be addressed in a future issue of *SNA Perspective*. ■

Architect's Corner

Stupid LAN Tricks

by Dr. John R. Pickens

Have you ever noticed the myriad of little interesting, clever, and sometimes inane characteristics of some LAN technologies? Things that aren't as they seem? So-called improvements that turn out to be liabilities? Shifting foundations? Standards that aren't?

Well, with tongue-in-check and inspired by U.S. talk show host David Letterman's "Stupid Pet Tricks," here are several of my own personal favorites. Some reflect the foibles of changing product directions; some the unavoidable confusion caused by using common terms in unqualified ways; some the politics within standards communities.

Read and enjoy. (And if you have your own favorite "Stupid LAN Tricks," I would like to hear about them.)

Trick #1: Boca LAN: the LAN to Beat all LANs

In the early 1980s one IBM group (Boca Raton) decides to beat the rest of IBM (Raleigh) to the punch for establishing a standard for LANs. Remember the broadband LAN supplied by Sytek (now Hughes LAN Systems)? The LAN to "blow away" those token ring and Ethernet foot-draggers. Remember the media hype?

I remember visiting with one of my Sytek friends in the early 1980s, sensing his pride at "winning" the LAN wars for personal computer networking. Success but for one miscalculation—no LAN adapter for the 37xx communication controller. Like the tortoise against the hare, along came token ring complete with 37xx support. Where is PC Network today?

Trick #2: The NetBIOS "Protocol" Hat Trick

Next comes NetBIOS, the "protocol." This is the second half of the Boca Raton LAN offering. This proprietary protocol was designed by Sytek and licensed to IBM. Closed and never published.

I was surprised at how quickly that (Sytek) NetBIOS came and went. Then came NetBIOS Version 2, then NetBIOS Version 3. But what a metamorphosis—from closed full seven-layer proprietary stack to open short two-layer semi-proprietary stack. Remember the "conversion" stack? This trick bears further discussion.

Contrary to popular perception (and contrary to my usage of the term in the preceding paragraphs), NetBIOS is really *not* a protocol. It is an interface. NetBIOS specifies a set of procedural function calls (API) for the provision of a connection-oriented service to be used by applications and system functions for program-to-program communications.

In reality, NetBIOS can be realized by many different protocol implementations. Some of the more popular designs include IBM's own current design (which maps the NetBIOS primitives to Logical Link Control Type 2 (LLC2) protocol), NetBIOS for TCP, NetBIOS for XNS, NetBIOS for IPX, and NetBIOS for OSI. In each case, the NetBIOS service interface is mapped into elements of procedure (with augmentations) contained within the appropriate underlying protocol family.

Incidentally, I'm waiting for a mapping of NetBIOS to APPC. This, however, is undoubtedly dependent upon roll-out of full-duplex APPC, the subject of a future discussion.

So, NetBIOS is really two tricks rolled up into one—the switch from the Sytek protocol to the LLC2 protocol baseline in the mid-1980s, and the popular misuse of NetBIOS as a "protocol" concept rather than its real meaning as an "interface" concept.

Trick #3: "Unroutable" Source Routing

Token ring offers a source routing capability. So is it possible to "route" token ring traffic? How many times have I heard this question!

This is a case of using a common term in an unqualified way. Any lower protocol layer could offer a "routing" function—in the academic sense, "routing" is the function that calculates where the protocol data units should next be delivered. However, in the industrial vernacular, "routing" has come to be identified with protocol-specific network layers, e.g., TCP's IP, Novell's IPX, XNS PUP, SNA Path Control, etc. One of these architectures, SNA APPN, indeed uses "source routing" at the path control layer. No wonder confusion reigns.

To avoid the confusion, perhaps the source routing layer should be spelled out more precisely—e.g., source route bridging, source route path control... Then, a better question and answer would be: Can source route bridging be routed at the network layer? Of course not. Different layers.

Trick #4: Token Ring—Dysfunctional Addressing

When token ring was designed, IBM chose to design it "better" than other IEEE 802 LANs. In particular, two "enhancements" were offered:

1. An addressing form that reversed the bit ordering for addresses (to gain efficiencies in the chip implementation, I suppose).
2. Function addressing, in which multiple station functions can be identified by a bit pattern in a single address. (No support of group addressing, otherwise known as selective multicast.)

Clever design, but one problem. Users want to integrate token rings with other LAN types, Ethernet, FDDI, etc. Such integration requires translation function, often with loss of function. These other LAN types support the IEEE 802 standard for proper (canonical) ordering of the address bits and also standards for group addressing.

Here is an example of the problem. Some protocols, e.g., the IP Address Resolution Protocol (ARP), carry the MAC address as higher level protocol data. Since the token ring bit ordering is reversed from all the other LAN types, layer 2 bridges must "see" the IP-layer ARP frames and translate the token ring MAC addresses to/from IEEE 802 canonical form in the IP layer frame!

Note: To be fair, original Ethernet could also be impugned for incompatible frame formats that also require translation However, since original Ethernet predated the IEEE 802, I'll let it off the hook.

Looking Forward—More Stupid LAN Tricks?

Despite the fits and starts described above, the LAN environment is really beginning to stabilize and a few key themes are worth noting.

1. Token ring, Ethernet, and FDDI have emerged as the leading LAN types. All major protocol families (SNA, TCP/IP, OSI, IPX) and most product implementations now support these LANs.
2. A single standard has emerged for LAN bridging—source route transparent (SRT) bridging—which specifies source routing for those that need it, transparent bridging for the rest. Despite the lack of SRT support in IBM's initial bridge products (PC-based bridge and 6611 bridge/router), this support will be forthcoming. Source routing will continue as an option but will only be active between pairs of end stations on source routable media such as token ring and FDDI. All other traffic will be transparently bridged.
3. Two key themes are emerging for LAN management—SNMP and CMIP. The older incompatible approaches, .5 MAC management and FDDI SMT management, will continue to exist for some functions but will yield to media independent schemes. Despite the IBM 6611 with its (proprietary MIB) SNMP management, IBM will continue to push the OSI CMIP (and its derivative IEEE 802.1B) profile.
4. Part of the dysfunctional addressing problem inherent in token ring—function addressing—will be fixed by adding support for standard IEEE 802 group addressing. The second half—non-canonical addressing—will likely not be fixed.

The future? Will we see fewer "Stupid LAN Tricks"? Well, like David Letterman's Stupid Pet Tricks, I suspect an abundant ongoing supply of candidates. (In fact, while writing this column, I just learned of another (USA) standards organization designing a new LAN type, insisting that the bit-ordering for MAC addresses should be bit-reversed from that of IEEE 802...) ■

(continued from page 9)

- TCP/IP from Ethernet to Ethernet across SNA is supported with 3745 Ethernet adapters.
- TCP/IP from Ethernet to Ethernet across SNA can also be supported with the 3172, but since the 3172 provides no routing or protocol processing, a host with TCP/IP must be used to encapsulate the IP traffic for routing on the SNA network.

3745 vs. 3172

Figure 7 shows a 3745 and a 3172 each being used for a TCP/IP user on an Ethernet to access a TCP/IP application on a host. With the 3172 option, the protocol processing is done on the host, while the 3745 performs the protocol processing itself. Which is better? It depends on the user's environment. If only a limited amount of TCP/IP traffic from an Ethernet is expected and an existing 3745 has some available processing power, the 3745 can be the easiest and most cost-effective solution. However, if significant TCP/IP traffic is expected, the 3172 will probably provide higher performance. In either case, the host will be involved—either with MAC and IP processing for the 3172 or with stripping the SNA from the encapsulated IP packets from the 3745 and passing them to TCP/IP on the host.

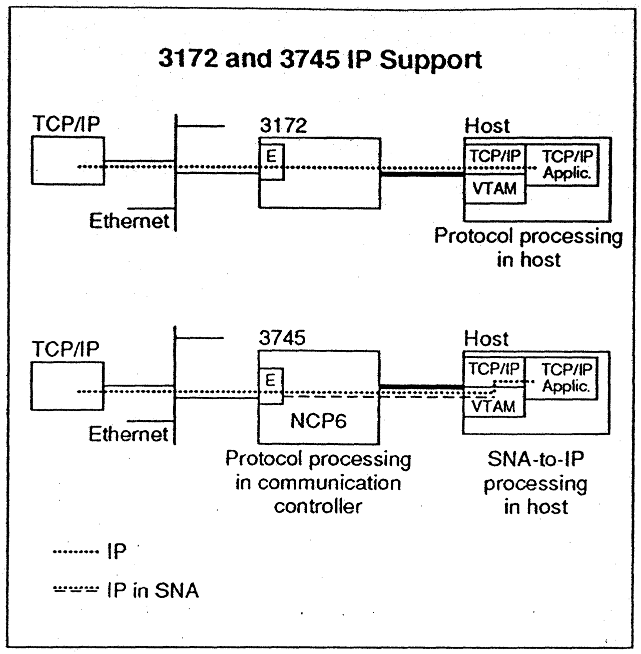

Figure 7

Summary and Conclusions

This *SNA Perspective* series focuses on users facing the challenge of TCP/IP integration into traditional SNA networks. TCP/IP is growing at an astounding rate in corporate sites with large SNA networks. User decisions for this integration extend beyond technology to include leverage of existing investments, life-cycle cost, performance, and reliability.

In response to strong user demand, IBM has been investing heavily in TCP/IP with a major acceleration since the beginning of 1991. Our expectations from IBM with regard to TCP/IP include: many more TCP/IP products in 1992, particularly in an expected May or June multiprotocol blitz; a shorter design-to-shipment cycle; better integration with SNA and network management; and continued enhancements and additional features on existing products.

A future article in this series will examine the protocols, application gateways, and application interfaces in TCP/IP products. We will also analyze the question of OSI and TCP/IP; examine the organizational structure for TCP/IP, OSI, and SNA responsibility in "the new IBM"; and review IBM's drive to decouple applications from networks. Also in this series, we will examine real-life end-user experiences in interconnecting TCP/IP and SNA to understand their goals, their planning and decision process, and the positive and negative elements of their actual implementation experience. ■

Copyright © 1992 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 • Marketing/Development: Alisson Walsh • Circulation Coordinator: Cheryl Roberts • Contributors: Thomas J. Routt, 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.