Volume 14, Number 5
May, 1993
ISSN 0270-7284

SNA Perspective

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

Points of Contention: APPN vs. TCP/IP

Introduction

Peer-to-peer networks, primarily LANs, are merging with the older, hierarchical networks like subarea SNA. Legacy applications will have to live on the same networks as modern client-server applications. The two protocols competing for the backbone of these merged environments are TCP/IP and APPN. Recently, much has been written about corporations abandoning their SNA networks in favor of TCP/IP despite IBM's efforts to bring SNA into the peer-to-peer world with the development of APPN. In this article we will examine some of the issues and compare the features that may determine the types of environments for which each will be better suited.

(continued on page 2)

6611 Network Node Combines APPN and Data Link Switching

The availability of APPN network node support on IBM's 6611 multiprotocol router fills an important gap—native routing of SNA traffic. The 6611 network node is being deployed within a new release of the Multiprotocol Network Program (MPNP) for the 6611 multiprotocol router. Prior to this, the 6611 handled SNA in very much the same way that other router vendors deal with this "unroutable" protocol. The only available alternatives were to employ source route bridging or to encapsulate the SNA data within a routable protocol such as TCP/IP. The 6611 is the first multiprotocol router to hit the market with this capability, but a growing list of other vendors are promising to deliver their own APPN network nodes. 3Com actually demonstrated theirs at last year's Interop show.

(continued on page 16)

In This Issue:

(continued from page 1)

Historical Design Influences

The current designs of both APPN and TCP/IP reflect both the history of the protocol suites and the recent technologies which have been incorporated. The roots of APPN are well known to anyone who has worked with SNA. SNA subarea networking, which is based on the use of host processors and communications controllers, has been in existence since 1974. Over the years, IBM's customers have made huge investments in mainframe-based application programs and 3270-compatible communications equipment. This enormous installed base influences every networking decision that IBM and its customers make. Some of the key requirements of this installed base are:

- Delivery of carefully controlled levels of service to individual network users
- Centralized network management
- Security for mission-critical applications and data
- Reliable networking support for mission-critical applications

SNA subarea networking met these requirements with a highly centralized style of networking. The pre-defined, fixed configurations for which SNA has been criticized were the key to delivering networks that met these requirements. Even though this hierarchical network design has been frequently criticized, it was appropriate for supporting the corporate networking environments of the 1970s and 1980s. These were networks of dumb terminals, usually 3270s, whose only networking requirement was connection to the mainframes in the corporate data center. Since all of the data traffic converged at the mainframe, it also made sense to centralize network management and control in the corporate data center.

Obviously, enterprise networking requirements have changed drastically in the late 1980s and the 1990s. Networks now require a much more decentralized design. This includes decentralization of user data traffic patterns as well as a requirement for more distributed management and control facilities. APPN, of course, is the technology that IBM is using to adapt SNA to the current enterprise networking environment, but APPN must still take into consideration the requirements of the installed base of older SNA applications and equipment. These requirements account for some of the differences between the design of APPN and that of TCP/IP.

The Origins of TCP/IP

TCP/IP originally evolved in an environment that was just about the opposite of that in which SNA grew up. The original design objective of TCP/IP was to provide a network for sharing information among researchers who were working on defense-related projects. These researchers worked at universities and research centers which had their own installed networks. The objective was not to create a complete new network to serve the researchers, but to interconnect their existing networks. This network of networks is known as the Internet.

A completely decentralized style of networking was needed because there was no single organization which controlled the entire network. Each research organization would continue to manage its own network independently of the other organizations. This is in contrast to the original design of SNA where a single organization was assumed to manage the entire network.

The users of the Internet were also very different from the users of the original SNA networks. The purpose of the Internet was to allow all users to freely share information across the network. Since there was no single, central clearing house for information, users needed to communicate directly among themselves. TCP/IP was, therefore, designed from the ground up to support the any-to-any connectivity that many in the SNA world now call peer-to-peer communications.

Another important characteristic of these Internet users is that they make unpredictable demands on the network. Since users can freely connect to the Internet and there is no central authority which controls network utilization, the network itself must either provide virtually unlimited resources, which is not economically practical, or the available resources must be rationed by the network. TCP/IP uses the latter approach and uses its best efforts to deliver data when requested to do so by a user. Note that this best effort delivery is, again, in contrast to the original SNA design which provided guaranteed delivery. But, in order to guarantee delivery, user requirements must be reasonably predictable.

Since SNA and TCP/IP clearly developed in very different environments, it's not surprising that the resulting technologies are quite different from one another. APPN is the result of IBM's efforts to make SNA a viable solution, not only for its traditional SNA users, but also for the rapidly growing class of users whose requirements mirror those of the TCP/IP community.

A generic comparison of SNA with TCP/IP is not very meaningful due to the fact that there are, for all practical purposes, two completely different SNAs—the original subarea SNA networking and the new Advanced Peer-to-Peer Networking (APPN). SNA subarea networking was targeted specifically at relatively static mainframe-centric networks and has almost nothing in common with TCP/IP. APPN, which is the focus of our discussion, is the style of SNA networking which is designed to compete directly with TCP/IP to build decentralized and dynamically reconfigurable enterprise networks.

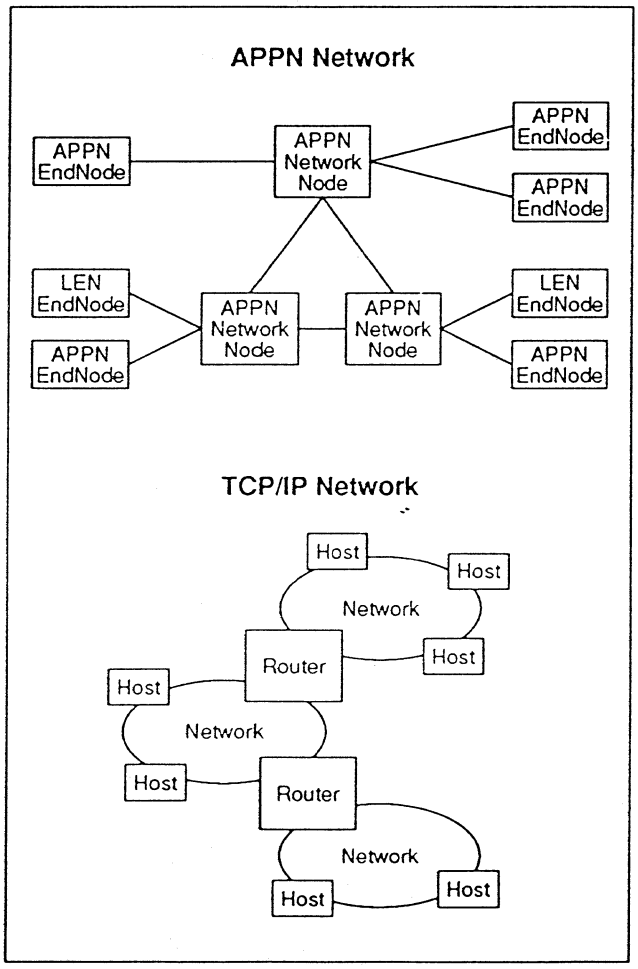

Figure 1

The Structure of APPN and TCP/IP Networks

Any comparison between APPN and TCP/IP has to begin with a look at the overall structure of each of the two types of networks. There are quite a few similarities between the structures of APPN and TCP/IP networks, but the terminology used in each networking environment is different. Figure 1 compares the elements of a simple APPN network and its TCP/IP counterpart.

Both APPN and TCP/IP networks are made up of nodes that can be grouped into two major categories. The first category of nodes are essentially packet switches which form the backbone of the network and are responsible for routing data across the network. In APPN networks they are known as Network Nodes, and in TCP/IP networks they are usually called routers, or sometimes gateways.

The second category of nodes are those which reside at the end points of the network and support the end users of the networks. These devices attach to the backbone network and depend on it to route data across the network. In APPN networks these

nodes are either LEN End Nodes or APPN End Nodes. These two types of end nodes differ in their ability to interact with the Network Nodes through which they attach to the network.

In TCP/IP networks the nodes which exist at the end points of the networks and support the end users are called hosts. These TCP/IP hosts communicate with one another through a backbone network which is made up of interconnected routers. Note that the term host is frequently confusing to SNA people who typically use the term host to refer to mainframes. This is a great example of the problems that differences in terminology can cause as networking professionals begin to deal with mixed protocol networks.

Comparison of Protocol Stacks

APPN and TCP/IP differ greatly from one another, not only in the protocols that they support, but also in the general structure of the protocol suites. These structures are a direct function of the original purposes of each protocol suite. SNA was intended to specify a complete set of networking protocols that would be implemented by every component of an enterprise network. As a result, SNA defines the protocols that are implemented at each of seven functional layers which loosely correspond to those of the OSI Reference Model. This SNA layering is shown on the left-hand side of Figure 2.

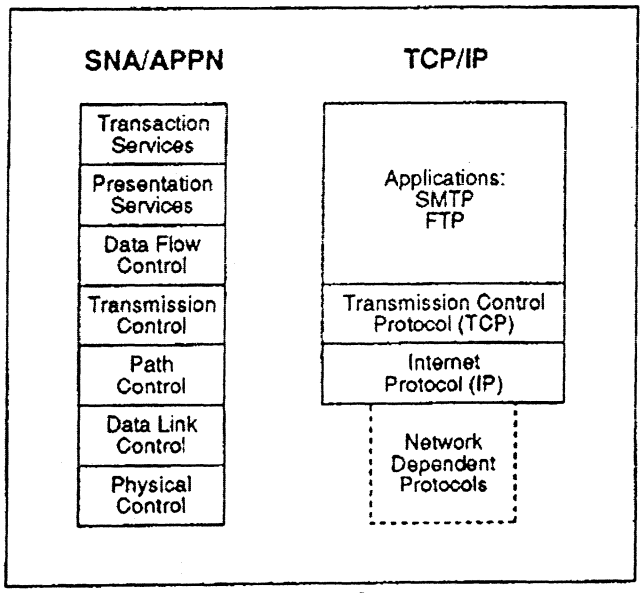

Figure 2

Where does APPN fit into this SNA layering model? The answer is a little complicated, but basically APPN deals with the end-to-end routing of data among the users of an APPN network. Therefore, APPN is usually said to operate at the Path Control layer which is the functional layer that handles routing in SNA networks. APPN is more complex than that, though, because the nodes in an APPN network exchange configuration and management information with one another and, therefore, they use the full protocol stack for that purpose. More specifically, the nodes use LU 6.2 sessions to communicate with one another which implies the use of protocols at the Transmission Control, Data Flow Control, and Presentation Services layers.

The functional layering model used by TCP/IP is quite different from that of SNA or the seven layer OSI Reference Model. Probably the most obvious difference is the apparent lack of data link and physical layer support within TCP/IP. These layers aren't really missing, of course, but our dotted lines reflect the fact that TCP/IP leaves the definition of the lower layer protocols to the individual networks which are being interconnected. Recall that TCP/IP is designed to build networks of networks. The assumption is that each of these networks will provide their own lower layer protocols. The most common example of this is the interconnection of LANs using physical and data link layer protocols defined by LANs such as Ethernet or Token-Ring.

The Internet Protocol (IP) layer is responsible for routing data across the network of networks which is called an internet. The routers in a TCP/IP network use the IP protocols to deliver data across the internet.

Resource Identification: Addressing and Naming Conventions

Every protocol defines a means of identifying resources within the network and even how to identify the network itself. APPN and TCP/IP are no exeption. Network resources are identified by a combination of names and addresses. People like to use names to identify resources because names are easy for them to remember. The distinction between names and addresses is arbitrary, but most people intuitively refer to nmemonic alphanumeric identifiers as names and to less human-friendly labels, usually numeric sequences, as addresses. Generally, names are intended to be used by the users and operators of the network while addresses are used by network software and hardware. Naming and addressing differ quite a bit between the two protocols and need to be looked at carefully because these items are closely related to routing algorithms and directory services.

Overview of Naming and Addressing

The role of naming and addressing in any network is to provide a method for logically connecting users or application programs. The techniques used to create these connections differ considerably between TCP/IP and SNA networks. TCP/IP assigns addresses to each host. The types of resources that are named and addressed also differs considerably between APPN and TCP/IP. Hosts are generally assigned a name which can be mapped via a distributed directory service into a host address. Host addresses, if known, can also be used directly, thus bypassing the directory service. Applications running on TCP/IP hosts are accessed via ports which are assigned numeric addresses.

In APPN networks the resources which can be identified by names and addresses are the Logical Units (LUs) and Control Points (CPs) within the nodes of the APPN network. Like TCP/IP, the process of connecting with a resource in the networks starts with a directory search operation. The input into the directory search is the name of the LU that represents the resource to be accessed. Unlike TCP/IP, though, the directory search does not result in the address of the target LU; the search results in the name of the CP which owns the target LU. A route to the target LU is then computed based on information in the network topology databases which reside in network nodes. The addresses which will be used to route data to the target LU are created dynamically when the LU-to-LU session is started. Resources in APPN networks have no fixed addresses.

APPN Network Qualified Names

In APPN networks, a name is given to each LU and each CP in the network. The LUs represent the users who are communicating with each other across the network while the CPs are the entities which exchange network management information.

APPN uses the same naming scheme as is used in SNA subarea networks. SNA employs a relatively flat name space using a two-level naming hierarchy. Each SNA name consists of a network name and the name of a specific Network Accessible Unit (NAU). In APPN networks, NAUs are either LUs or CPs while in SNA subarea networks they are either LUs, Physical Units (PUs), or System Services Control Points (SSCPs). These names are generally written in the following format:

Netid.NAUname

The network ID acts as a qualifier for the NAU portion of the name. Since each SNA network is given a unique network ID, NAU names can be assigned independently within each network without concern that they might be duplicated within other SNA networks. The network ID and NAU names are each made up of from one to eight EBCDIC characters.

TCP/IP Host Names

TCP/IP has its own naming scheme which assigns a unique name to each host in the Internet. TCP/IP uses a much more hierarchical naming system than SNA. The hierarchical naming convention that is used reflects the fact that TCP/IP is designed to support communications across a large number of autonomous organizations. TCP/IP names consist of a variable number of character strings which are separated from one another by periods. The following is an example of a TCP/IP name:

ipac.caltech.edu

The left-most character string is the highest level of the naming structure. While TCP/IP specifies only the very general dotted hierarchical naming string, most TCP/IP users generally conform to Internet naming conventions. This approach ensures that users will be able to communicate on the Internet without requiring any name changes. The Internet authority assigns the highest-level portions of the name, in this case caltech.edu, to ensure that the resulting names are unique across the Internet. Local administrators can then assign the lower level qualifiers within their own organizations.

Directory Services

The naming schemes employed by TCP/IP and APPN relate directly to their directory services functions. TCP/IP, in particular, takes advantage of its naming structure to implement a distributed directory search strategy.

Both APPN and TCP/IP provide users with directory services which can be used to discover the locations of remote users and applications within a network. From a user's point-of-view the directory services of both APPN and TCP/IP provide a similar function, but their underlying operations are fundamentally different.

Both protocol suites employ a distributed directory system but they are quite different from one another in the way that they operate and in the type of information provided to users of the directory services.

APPN Directory Services

In APPN networks each node contains a directory which contains information about the location of some of the LUs and CPs in the network. The APPN directory service ties these directories together to form a distributed directory service.

Each node contains directory information about its own resources and, in some cases, resources that exist in other nodes in the network. In most cases resources are automatically registered to the network upon a node's initial connection to it. From an administrative point of view, this means that resources are defined in just a single node. If the node is moved, or given another network node server, no changes need to be made. Directory services will automatically reflect the changes. When initiating sessions, the End or LEN node specifies the destination LU name to APPN's Directory Services and a search is done for that LU. The local directory database of the originator is searched first and if the destination LU is not found a search request is propagated to the network node nerver for that node. Note that in the case of LEN nodes, if an LU is not found in the node's directory database, the search ends there in failure.

Directory searches always begin with the local directory. Each Network Node is also capable of acting as a directory server to its own LUs and to other network nodes. The LU which represents the user that is requesting the directory service forwards the name of the target LU to its Network Node server.

In order to service the request, the Network Node first checks to see if the target LU resides locally in either the Network Node itself or in one of the adjacent end nodes, or if it has previously learned the location of the LU via a network search. If both of these searches fail, the Network Node broadcasts a

request for the target LU across the network and the Network Node which owns the target LU will respond to the request. An alternative to the broadcast search is to use a centralized directory such as that implemented in VTAM Version 4. The use of a centralized server can greatly reduce the amount of network overhead which is created by broadcasts from individual Network Nodes. The response to a request for APPN directory services is the name of the CP which owns the target resource.

TCP/IP Directory Services

TCP/IP also implements a distributed directory service and this service is also initiated by naming the resource which is to be located. The TCP/IP directory service is called the Domain Name Service (DNS). The services are provided by name servers which cooperate with one another to provide directory services.

These servers are logically connected in a tree structure that corresponds to the TCP/IP naming scheme that we discussed previously. Each client of the DNS system must know the address of at least one name server. Each name server must also know the address of at least one root server which is at the highest level of the tree structure. This ensures that each server can communicate with the other servers because the servers each know the addresses of each of the name servers at that next lower level. DNS exploits the hierarchical naming structure of TCP/IP which was discussed previously.When names are not found in the local directory the search can proceed by searching the directories along the defined hierarchy. Let's look at the search procedure for finding the address that corresponds to the following name:

nic.ddn.mil

The host originating the directory search will first search its local DNS server. If the name is not found on the local server, a distributed search is initiated. The local DNS server in this case will know the address of the DNS server at the top of the hierarchy and direct the request to that server. This server corresponds to the highest level name qualifer which in this case is "mil." If the name is not found in this server, the search will continue down the hierarchy which is reflected in the name.

Like APPN, DNS also employs a caching technique to reduce search overhead for names whose addresses have been previously found. Like APPN, addresses that are found in the cache can be verified before they are used. For example, when a name server responds to a directory services request it can indicate that the response was obtained from a cache and, therefore, may be unreliable since it is based on information that may have been obtained some time ago. The response also includes the address of the server from which the cached information was originally obtained. This allows the client system to either decide to use the potentially unreliable cached information, or it can choose to incur additional network overhead and query the original server.

The information that is ultimately returned to the DNS requestor is the IP address of the named host. Note that this differs from APPN where the information returned is the name of the node which owns the target LU.

APPN vs. TCP/IP Directory Services

Note that the DNS distributed directory search is quite different from APPN's directory services. Within an APPN network the distributed search would either involve the broadcast procedure or, if available, a centralized directory could be accessed. DNS's hierarchical search strategy is ideally suited to the Internet environment which is made up of many autonomous networks. Each of these networks can administer their own directories which are then logically linked together via the DNS naming structure. This type of directory srtucture would also be very useful in many commercial networks where individual department and workgroups can administer their own local directories.

The APPN approach, on the other hand, relies on broadcast searches which can result in a considerable amount of network overhead. A central APPN directory can be used to reduce the number of broadcasts because only the central server initiates broadcasts in this environment. Caching of names in local directories should prove to be very effective in reducing broadcast traffic in APPN networks because SNA users tend to connect with a relatively small number of remote applications and users on a regular basis.

APPN's directory service has the edge when it comes to administering directory databases because DNS directories must be manually updated while APPN's benefit from the automatic registration of users as end nodes connect to the network.

Addressing

The address techniques used by TCP/IP and APPN networks could hardly be more different from one another. TCP/IP assigns fixed addresses to resources while in APPN networks, addresses are not used to locate resources at all, instead, APPN dynamically creates addresses that are used to define a route to the target resource.

APPN Addresses

In APPN, addresses are not used to locate a node, but are used to identify a session between two adjacent nodes. The network qualified name (discussed previously) is used to discover where a node is located and determine a route to it. An address space exists for each transmission group (TG) or link that a node can send and receive data on. The addresses used in the packets transmitted between nodes are called local form session identifiers (LFSIDs) and they are created as each session stage is created between two endpoints. These addresses don't refer to any specific resource on the network, but are used by the session to connect resources across the network. They are temporary in nature, as are the sessions. At each hop of the session, this address is swapped to use the LFSID assigned between the two nodes which make up the hop. Figure 3 shows how session connectors, which are setup at session initiation, provide the logical link between the LFSIDs used on each transmission group.

This process is often referred to as label swapping or address swapping. As a session between two resources is created, so are the LFSIDs. APPN combines the ODAI, OAF, and DAF fields of a FID2 Transmission Header into a field which contains the 17-bit LFSID.

The important thing to note as an administrator of a network is that one does not have to generate and manage these APPN addresses. They are generated dynamically within the node on a session by session basis. On the other hand, when performing problem diagnosis this can become a problem. A single session through the network may really be a series of chained segments between adjacent nodes and each link in that chain has an individual and temporary address or LFSID to track. Identifying the entire session across the network may be difficult without adequate reporting software linking the individual connections to the two resources attempting to connect end-to-end. Also, as each segment can have different LFSIDs between them, these labels must be swapped at each hop. This incurs processing overhead and storage overhead in the network nodes. High Performance Routing (HPR), discussed

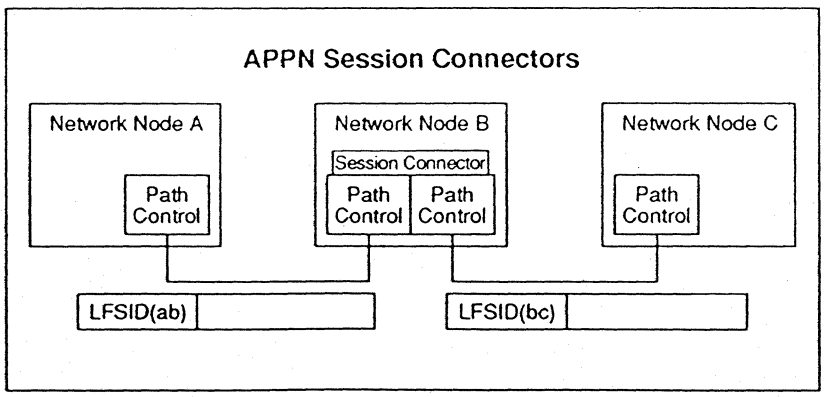

Figure 3

in detail in a previous *SNA Perspective*, attempts to deal with this by carrying all the label information in the packet to make the swapping more efficient and eliminate the storage overhead.

In addition to addressing logical units, APPN network users can also identify specific application programs within the target LU. Technically, this level of addressing is provided by LU 6.2 rather than APPN. LU 6.2 uses a variable length (1–64 bytes) alphanumeric field called the transaction program name (TPN). IBM has created a set of standard TPNs which identify the architected applications which support IBM architectures such as SNA/Distribution Services. TPNs can also be assigned to user-written applications.

TCP/IP Addresses

In TCP/IP, adresses are significantly different. First of all, they are static, 32 bit quantities. They identify a host or node on a network, all hosts on the network (a broadcast address), or the network itself. An internet address is made up of two components—the netid, which identifies the network, and the hostid, which identifies a particular host on that network. As shown in Figure 4, address space is divided into classes. The first three bits define the three primary classes of address. Class A, whose first bit is always zero, has a seven bit netid and 24 bits to define the host address. This allows for 127 very large networks. A Class B address has 14 bits for the netid and 16 bits for host addresses. Class C addresses have 21 bits to identify the network and 8 bits for identifying a host, which allows for many networks with 256 nodes each.

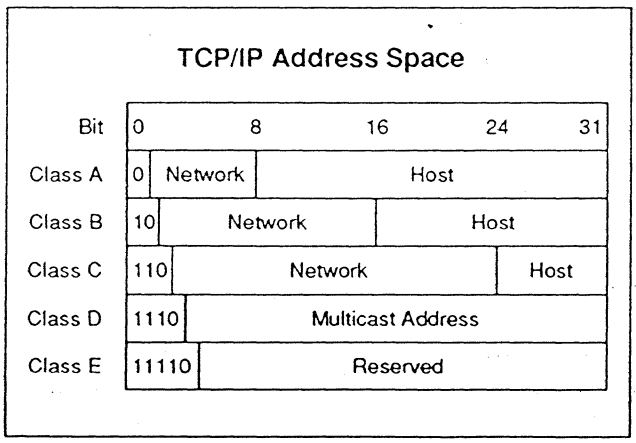

Figure 4

The static nature of IP addresses is nice for following data through the network, as a node's IP address is used in all communication with that host. There are several weaknesses in this type of addressing though. If a host moves from one network to another, its IP address must change. If a network outgrows its class and a new netid is assigned, each node on the network must be reconfigured with a new address—a tedious and time consuming process. A routing issue also occurs with this addressing scheme—the path chosen to a node determines the route taken to get there because the netid is carried in the address. Thus, a node with more than one connection has a route to it associated with each connection. Depending on the address used, the most efficient route to that host may not be chosen. In APPN, routes are address independent.

Network addresses are assigned by the Network Information Center (NIC). Class B addresses are the most commonly requested because of the number of hosts allowed. A Class A address is too small for most organizations and the number of Class A addresses is very limited. As a result, a problem has arisen, because the number of Class B addresses is in short supply. While some say the issue of running out of Class B addresses isn't really all that pressing (an organization could use multiple Class A networks and connect them using routers), the Internet Engineering Task Force (IETF) has acknowledged for quite some time that this is a major issue. There are currently about five proposals out to address the addressing problem, but no definitive solution has been arrived at yet. SNA has faced this crisis in its history as well. The addresses used in SNA subarea networks have been expanded several times.

TCP/IP's network addresses are used to identify a particular host within a network. The applications which run within these hosts must also be addressed. This is accomplished through the use of

logical connectors called ports. Figure 5 shows the relationship between IP addresses and ports. Each port is assigned a port number which is used to address a specific application within a host. By convention, certain port numbers are preassigned to applications which provide generic network services such as file transfer and electronic mail. These preassigned ports are commonly known as well-known ports. Other ports are available for use by user-written applications.

Which Addressing Scheme is Better?

A comparison between the addressing schemes used by APPN and TCP/IP is very much an apples and oranges comparison. APPN's dynamic label swapping technique only makes sense in support of a connection-oriented network protocol such as APPN while TCP/IP's connectionless network layer requires the use of fixed addresses.

There are some important addressing characteristics, though, which should be emphasized. TCP/IP's addresses must be changed whenever a host moves from one network to another. This can result in additional administrative overhead. Another shortcoming of the TCP/IP addressing which has recently come to light is that its address space is quickly reaching its maximum capacity. This has occured because the objective within the Internet community is to assign a unique address to every TCP/IP host in the world which is connected to the Internet. Changes to the addressing scheme are now under discussion.

Connectionless vs. Connection-Oriented Delivery

Before comparing the specifics of how either APPN or TCP/IP perform their routing functions, a brief discussion of connection-oriented vs. connectionless delivery systems will be given. Traditionally, SNA and the original implementation of APPN has been connection-oriented. This means that a connection is created between each node traversed to comprise an end-to-end session. All of the packets for a session must traverse the same path and delivery is guaranteed between each node. Sequencing, flow control, and error control occur at each segment traversed. TCP/IP, on the other hand, uses a connectionless delivery system. No connection is set up between segments traversed, each packet is treated independently, and no guarantee is made that a packet successfully traverses an individual segment. It is up to the ends of the session to resequence packets and to perform the error control.

There is great debate as to which method is "better." The proponents of connection-oriented services typically cite advantages such as:

- Processing is off-loaded from the communicating end systems because the network handles packet sequencing and ensures that packets of data are not lost.
- The network can reserve resources at call set-up time to ensure that end users receive the levels of service that they require.
- Routers can operate faster because the optimal route is computed only during call set-up.

The connectionless camp offers its own arguments including:

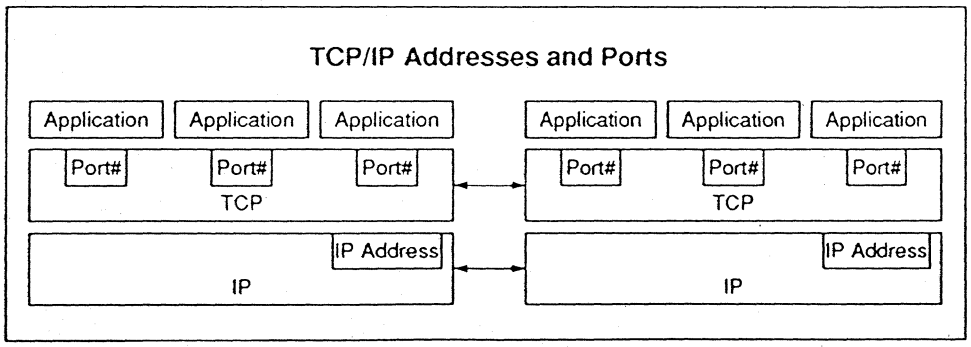

Figure 5

- Since routing decisions are made on a packet-by-packet basis, traffic can be transparently rerouted to avoid failed resources such as data links or intermediate routing nodes.
- Connectionless networks are better suited to handling bursty traffic because no network resources are reserved and thus wasted during periods when little or no data is being transmitted.
- It is better to provide degraded services to all network users during periods of congestion rather than reserve capacity for certain users.

Obviously there are some inherent tradeoffs between these two styles of networking. From the perspective of the traditional SNA user community, the greater control of end user service levels (such as 3270 response times) provided by APPN's connection-oriented network layer is probably the key advantage. The downside for these traditional SNA users is the lack of dynamic rerouting, but this is consistent with the capabilities that they have had in the past.

Routing Protocols

There are many router products available from numerous vendors based on the TCP/IP protocol suite. The protocols used in those products have changed as the products were put into more and more environments and their limits were tested. APPN has been available for a number of years in AS/400 environments and is about to become available in more traditional router products as more and more router vendors decide to implement it in their products. APPN, too, is evolving to address the needs of users and vendors. We will describe briefly how each routing protocol works and then look at several key issues in comparing their implementations.

APPN, Today: Intermediate Session Routing (ISR)

APPN employs a link state routing protocol. Characteristics of link state routers are that they each contain a topology database which contains information about all of the links in the network and their state. Each node is responsible for computing the best routes from it to any other node in the network. In APPN, the Network Nodes are responsible for maintaining the topology database. Topology updates occur on CP-CP sessions between adjacent nodes and occur only when there are changes in the network. This is different from other routing protocols where information is periodically broadcast whether it is needed or not, or whether the topology of the network has changed or not. The messages exchanged by the Network Nodes to maintain consistent databases across the network are called Topology Database Updates (TDUs).

When a session is requested between LUs residing in two end nodes, a request is made to locate the destination LU using a requested mode or class of service describing the transmission requirements for that session. Directory services will find the target LU. A route will be returned containing a list of nodes that will be traversed for the entire duration of the session. A BIND will establish this session, using session stages between each of the intermediate nodes named as part of the route. Between each node a pacing window exists to control the level of traffic between nodes and avoid congestion.

The major weakness of this scheme is that sessions can be fairly long lived and the conditions which determine optimal routes at session startup may not remain the same for the life of the session. Better routes may become available and not be used. Or, if any link in the series fails, the entire session fails. There is no rerouting around failures and the session must be restarted from scratch, potentially disrupting the application.

APPN, Coming Soon: High Performance Routing (HPR)

To address these concerns, an enhancement to the APPN routing protocol has been specified and will be available in twelve to eighteen months. This is High Performance Routing. This is a connectionless service that allows for dropping packets when congestion occurs and allows for rerouting around failures without disruption of the session. Rapid Transport Protocol (RTP) is used at endpoints of HPR sessions to perform packet resequencing and guarantee reliable delivery of packets. An adaptive rate-based flow control algorithm is used at ends of HPR sessions to prevent congestion in the network. HPR can be easily integrated into networks currently using ISR, mixing and matching segments, depending on the capabilities of the nodes and links between them. A more detailed description of HPR is provided in the March, 1993 issue of *SNA Perspective*.

Commonly used Routing Protocols in the TCP/IP Environment: RIP and OSPF

Routing Information Protocol (RIP) was an early routing protocol implemented and used widely. This was not necessarily due to the technical merits of the protocol, but because it was distributed with many UNIX systems. It uses a distance vector algorithm to determine the shortest path to a node. This is quite simplistic, as it only considers the number of hops to arrive at a particular destination, while that may not actually be the best path. For example, more hops on higher speed links may be faster than fewer hops on slower links. RIP gateways use frequent broadcasts to advertise their current routing database. Not only can these messages clog the network, but they can also propogate slowly creating inconsistencies in routing tables. One gateway can be advertising a route that's no longer available, but it hasn't discovered that yet. This can create routing loops. This convergence problem and the amount of broadcast traffic generated in large networks has led to the development of other routing protocols.

The IETF proposed another routing protocol which has grown in popularity and is replacing RIP. The Open Shortest Path First (OSPF) protocol is a link-state protocol for which there is a published specification. (It should be noted that RIP was widely implemented before a standard was published.) Like APPN network nodes, OSPF routers maintain topology information about the entire network. Costs for each link are computed and the classic shortest path, or Dijkstra's algorithm, is used to compute the optimal route. Link state advertisements (LSAs) are broadcast when the status of a link changes so routers have current information and can recompute its shortest path tree. Each node receives an LSA, it must recompute the shortest path tree. While this can be expensive for large networks, in practice it is done fairly infrequently. Routing metrics can be assigned by the system administrator to be any combination of network characteristics, such as delay, bandwidth, etc. OSPF includes type of service routing, although it is not implemented (see detail section below). The quantity of routing information propogated through the network can be reduced by splitting the network up into areas where each router keeps a topology database representing the area. Border nodes communicate between areas and advertise costs to those areas in the same fashion a route to any other node is advertised.

Vendor Proprietary Protocols

IBM is often accused of perpetuating a proprietary protocol with its push of APPN. It should be noted that other router vendors have also proposed their own protocols to provide alternatives to weaker, standard protocols. One notable example is IGRP used in cisco routers. Like RIP, it uses a distance vector algorithm, but attempts to provide convergence rates similar to those of link state protocols while keeping processing and bandwidth requirements low. Providing sophisticated type of service routing was not a goal of IGRP. While limitations of the standard routing protocols may push vendors to quickly provide solutions to their users, developing

protocols such as IGRP without having them accepted as standard in the community increases the probability that different vendors' products will not communicate with one another and often leaves the user implementing the least common denominator in their networks. On the other hand, useful new protocols can sometimes be delivered to customers more quickly by using vendor-specific solutions.

Class of Service (COS) and Transmission Priority

Characteristics of the underlying network technology as well as costs and the current state of a link or node can be considered in addition to what is the shortest path to a destination when making routing decisions. Different types of sessions or connections have different requirements for data transmission. Batch data may have a lower priority and require lower speed transmission than interactive data, where users expect quick and consistent response times. To support this, a notion of transmission priority and class or type of service must exist within the protocol suite.

All APPN network nodes support a COS database and manager. End nodes with the COS/TPF function, which is the ability to translate a mode name to a COS name and a COS name to a transmission priority, also support these. The database is managed independently at each node by a system administrator. Defined in the database are transmission groups, or links, and node characteristics required for a given COS. These characteristics include capacity, propogation delay, congestion, cost per byte, and route addition resistance, amongst others. There is also the capability for the adminstrator to define characteristics used in COS selection. APPN supports several predefined mode names and corresponding COS tables, but allows an installation to choose its own if it wishes. When a session is requested a mode name is specified. This mode name is mapped to a given COS name and an appropriate route out of a node is selected for a session's data.

TCP/IP has an architected field in the IP header to indicate type of service and precedence. In APPN, using ISR, a route is calculated when a session is created and used throughout the session. Because of the connectionless nature of TCP/IP, the type of service field must be carried on each packet and looked at by each router. This would allow for rerouting different packets as conditions in the network change, but incurs additional decision overhead at each hop in the path the packet is taking. Three bits are allotted for precedence, or transmission priority. Three other bits are used as flags, indicating low delay, high throughput, or high reliability. This scheme is obviously much less flexible than the COS database provided in APPN. More important, however, is that this field, though architected, is not generally looked at in any of the router implementations which exist today. Thus, although the architecture allows for type of service/precedence based routing, it is rarely, if ever, used. It has been acknowledged by the Internet community that support will be required to "guarantee" performance for different classes of applications such as real-time voice and video.

Not all applications require the same link characteristics to service them. Not every network can provide high-speed, high-bandwidth service to everyone on it. APPN allows for this with a sophisticated and flexible COS database and routing capabilities. The OPSF standard requires TOS support, but most router vendors don't implement it. They have decided the databases reqired place too great a memory requirement in their products. Instead they may implement transmission priorities, but if every vendor does not do it the same way, and they don't, it may not be truly available in networks containing routers from more than one vendor.

Congestion Control

While both APPN and TCP/IP provide for congestion control, they again differ in their methodologies. APPN provides a preventative approach, while TCP/IP, because of its connectionless nature, deals with congestion after it detects it. All other things being equal, it's better to prevent the problem than to simply recover after the fact.

As mentioned above, in APPN route selection, congestion on a link or in the node itself can be used as one of the criterion in deciding whether or not that link provides an acceptable route. Node congestion status is determined to direct sessions away from a node where 90% of its maximum intermediate sessions are currently in use. This only prevents sessions from being routed through that node if the COS requested finds this congestion level unacceptable. Another mechanism used to control congestion is adaptive session-level pacing, where the receiver can adjust the window size based on the node's congestion level by informing the sender of the new window size in a pacing response. Adaptive BIND pacing allows the setting of a BIND window size, which limits the number of BINDs sent across a particular link. This is necessary because BINDs do not flow on a session, yet can generate bursty traffic at node or network startup, thus creating congestion themselves. APPN does not reroute around congestion.

TCP/IP, because of its connectionless nature, can not adjust for congestion on a session level basis. Routing around congestion would have to be performed for each packet, although this is not commonly implemented either. Generally, all flow control is performed by the ends of a TCP connection, and intermediate gateways and routers drop packets when they become congested. There is a source quench message a router or gateway can send back to the source of congestion causing packets to request it reduce transmission, but this is rarely, if ever, implemented in the real world. When TCP detects packet loss, two mechanisms are used to recover, slow start and multiplicative decrease. TCP knows the receiver's window size, which is advertised in acknowledgements, like adaptive session-level pacing. TCP also maintains a congestion window limit. The actual window size it will use is the minimum of the two. When a packet is lost, it decreases the congestion window limit by half, with the smallest window size being one. It also increases the size of its retransmission timer exponentially as to not flood the network with retransmissions, which would cause greater congestion. After congestion clears, a slow-start algorithm is used to start transmission slowly and increase the window size with each acknowledgement.

HPR is more like TCP/IP in that it allows for dropping packets when a node becomes congested, as it is designed for high speed links where retransmission may not be too costly. It uses an adaptive rate-based (ARB) congestion control algorithm instead of the adaptive session-level pacing used in ISR. While HPR does not explicitly reroute around congestion, it can reroute around failure. If the delays caused by congestion appear as a link failure, HPR will reroute around the congestion.

It seems to be more of a philosophical question whether the preventive or reactive approach to congestion control is the more appopriate solution. In a multiprotocol network things get a bit more complicated. For example, TCP traffic could be being routed in an APPN network. While APPN can use adaptive pacing to slow congestion, it cannot tell a TCP node pumping too much data into the network to slow down, because it does not generate messages that node would understand.

Implementation and Interoperability Concerns

SNA bashing is a popular sport in the TCP/IP community, especially its proprietary nature. TCP/IP is perceived to be an "open" environment. Part of that is attributable to its close connection to the UNIX

world, but other reasons for the perception include its availability on so many platforms and the fact that it is controlled by the Internet Activities Board (IAB). This organization coordinates the research and development of TCP/IP and seeks input from just about anyone who is interested via working groups and RFCs (Requests for Comments). Actual enhancement projects are overseen by the Internet Engineering Task Force (IETF), which is an organization underneath the IAB. In a perfect world, the organization would address everyone's concerns through the standards it creates for use in industry. Unfortunately, such openness is often cumbersome. Drafting new standards can be a painstakingly slow and political process. Once drafted, vendors produce implementations of these standards, but it is hard to find a reference point for a correct implementation. Often different vendors' implementations of protocols don't work together, or as in the case of type of service and precedence, the architecture is vague enough that it is not clear how to go about implementing it correctly to begin with. Interoperability showcases, such as those at Interop, have helped a great deal in ensuring products will work together by creating a public forum to demonstrate interoperability, but certainly haven't solved the overall problem. In addition, when a vendor feels the current standard is deficient in some way, it may implement a proprietary solution, thus guaranteeing that its product will only communicate with others of its own type. The IGRP routing protocol is an example.

APPN, on the other hand, is thought of as controlled strictly by IBM and, therefore, proprietary. IBM is making a strong attempt to dispel that image by opening up conferences to vendors and holding implementor workshops on its newer technologies. Recently IBM has announced the formation of the APPN Implementors Workshop which will allow vendors and users to have a say in the future development of the architecture. Through a relatively inexpensive intellectual property package, complete APPN specifications, patents, copyrights, and trademarks are being made available and the source code is available for a more expensive licensing fee.

Despite all this, IBM still does make the final decisions on what does, and doesn't, go into APPN and to a certain extent IBM deserves the proprietary reputation. The question which must be asked, though, is this all bad? There can be some good results: decisions and implementations can be turned around faster by a single organization and by making the source code available to vendors, there is an implementation to benchmark against for interoperability.

Conclusions

The APPN architecture today has some technical advantages over TCP/IP, but both protocols continue to evolve and incrementally improve. *SNA Perspective* believes they both offer strengths as backbone protocols depending on the user environment. This is evidenced by the fact IBM has invested heavily in both APPN and TCP/IP technologies and other vendors are now signing up with APPN. IBM's recent moves to make the APPN evolution more open have certainly made it more attractive to those vendors.

Because TCP/IP solutions are available today and have been for some time, TCP/IP may seem to be winning in corporate environments. This may prove more important than the technical advantages APPN offers over TCP/IP. However, if the "weaknesses" of TCP/IP prove more hindering in corporate environments than they have in the primarily research environments the protocol has thrived in, then APPN may become more attractive. This will be especially true if the political structure of the TCP/IP world cannot resolve these problems quickly and vendors don't promptly bring solutions to market as new standards are developed. *SNA Perspective* believes that APPN will provide a more seamless migration path for exisiting SNA users through its Dependent LU Server/Requester support. This should allow the older applications and devices supported in subarea SNA to be integrated smoothly with the client-server and high speed networking world of the future.

Two Different Views of Networking

From our discussion of APPN and TCP/IP it should be clear that these two technologies reflect very different views of the world. APPN continues SNA's tradition of creating a controlled networking environment, while acknowledging the need for more decentralized control and dynamic reconfigurability. APPN retains the connection-oriented delivery techniques that are consistent with SNA subarea networking while employing link-state routing protocols, albeit proprietary, that are coming into use in networks based on industry standards.

TCP/IP continues its own tradition of creating a networking environment whose purpose is allowing free connectivity among users and the equitable sharing of network resources among those users. These values are reflected in TCP/IP's connectionless delivery systems and emphasis on promoting open industry standard protocols.

Where are these two views of networking moving in the future? We feel that they will inevitably converge. The technologies won't converge into a single set of protocols, but the two cultures are converging. SNA must support the needs of decentralized computing and rapid change. As TCP/IP moves into enterprise networks it will have to provide a more controlled and secure networking environment. Each group can learn a lot from the other—and they'll have to in order to compete in the enterprise networking market. ■

(continued from page 1)

IBM's 6611 brings together much more than just the basic network node technology. The 6611 network node exploits the services provided by IBM's Data Link Switching (DLSw). DLSw is the technique which IBM uses to route its unroutable SNA and NetBIOS protocols. LAN-based SNA and NetBIOS users employ IEEE 802.2 Logical Link Control Type 2 (LLC 2) to provide the reliable, connection-oriented data link service that they require. The main purpose of DLSw is to encapsulate LLC 2 packets within TCP/IP so that they can travel across router-based internets, but it also includes the following features:

- A technique for eliminating LLC 2 timeouts across complex LAN/WAN networks
- Flow and congestion control for LLC 2 traffic
- Reduction of overhead due to LAN broadcast packets
- Elimination of the source route bridging hop count limitation

DLSw is likely to become an important industry standard because IBM's goal was to make it a true open standard. In accordance with the Internet's standards process, IBM submitted the specification document to the Internet Engineering Task Force (IETF). The document is now designated as Request For Comments (RFC) 1434. The publishing of RFCs is the first phase of the Internet's standards process.

DLSw is capable of making a combination of IEEE 802.2 LANs, SDLC data links, and TCP/IP networks into a network which is logically a single IEEE 802.2 network. The difference between the physical and logical DLSw networks is shown in Figure 6 (see page 17). The bottom line is that with DLSw all of the users attached to the various physical networks become part of a single logical LAN.

There's nothing new about interconnecting LANs to create a single logical LAN. Many enterprise networks already use bridges for exactly this purpose. What does DLSw bring to the party? In order for remote bridges to effectively handle LLC 2 traffic they must employ WAN connections that introduce very little delay in transporting packets of data. This is required because LLC 2 was not designed to run over wide area connections and it sets timers which require quick end-to-end response times. WAN connections must also be able to provide enough bandwidth to prevent congestion because bridges have no flow control mechanism and simply discard packets whenever congestion occurs. DLSw eliminates the LLC 2 time-out problems and adds flow control procedures to avoid congestion problems. This allows LLC 2 LANs to be interconnected over networks which provide varying levels of throughput and variable transit delays—both characteristics of TCP/IP networks.

Benefits of the APPN/DLSw Marriage

The 6611 network node exploits the seamless LLC 2 LAN/WAN networks which are provided by DLSw by using APPN's connection network feature. Connection networks support direct any-to-any connectivity among APPN end nodes attached to a single shared access transport facility (SATF). An SATF is any network which supports direct connections among all of its attached users. Commonly available SATFs would include LANs, X.25, and frame relay networks. The logical LLC 2 networks created by DLSw are also SATFs.

In APPN networks without connection networks the data flows between communicating APPN end nodes as shown at the top of Figure 7 (see page 18). This approach is necessary when the end nodes don't have a direct data link level connection between them, but when they are both connected to a common SATF the use of the network node as an intermediate router makes no sense.

Groups of APPN end nodes attached to a common SATF can be logically defined as members of an APPN connection network. In this case, members of the connection network use a network node to initiate connections between the end nodes, but not for intermediate routing of data traffic. This is shown on the bottom of Figure 7 (see page 18). The connection network eliminates the unnecessary overhead of passing data traffic through the intermediate network node.

The logical LLC 2 LAN created by DLSw is a SATF and is, therefore, capable of supporting APPN connection networks. This creates an interesting environment for APPN users. It means that in a large enterprise internet made up of many geographically dispersed LANs connected by TCP/IP networks, direct LLC 2 connections could be established between any pair of end node users as long as their nodes were defined as members of a single connection network.

This "mega connection network" could theoretically use a single APPN network node to set-up connections between

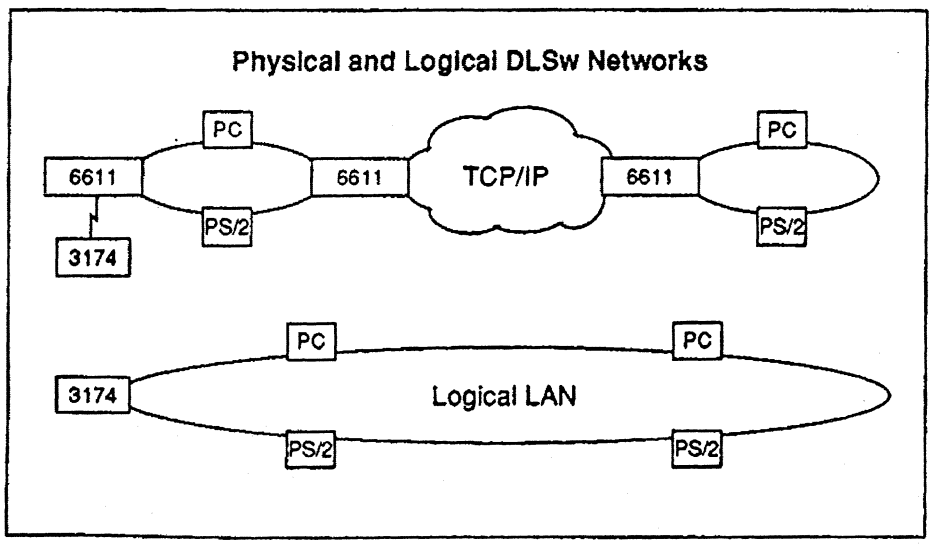

Figure 6

users, but rely on very cost-effective, high peformance IP routers (including 6611s and other vendor's products) to handle the end-to-end data traffic.

Using DLSw to Connect Network Nodes

DLSw and TCP/IP can also be used to interconnect APPN network nodes. In fact, for NN-to-NN connections over WANs, only TCP/IP connections will be initially supported. This means that direct WAN connections from 6611 network nodes to IBM's other network node products such as the AS/400 or the OS/2-based Extended Services are not initially supported. Network nodes can communicate with one another either via DLSw or by using a direct socket connection across a TCP/IP network.

How The 6611 Solution Compares With APPI

The 6611's support for connecting APPN network nodes across TCP/IP networks is very similar to the network configuration proposed by the Advanced Peer-to-Peer Internetworking (APPI) Forum. Few networking initiatives have received more publicity and been the subject of so much controversy over the last few years as the APPI Forum. The APPI Forum was initially launched in response to industry concerns about the proprietary nature of APPN and the licensing and patent issues associated with APPN. The objective of the APPI Forum is to define an industry standard technique for connecting APPN End Nodes via an industry standard transport network, specifically, a TCP/IP transport network.

While industry analysts and the trade press have focused on the work of the APPI Forum, IBM was developing its own approach for connecting APPN users over TCP/IP internets. Even though the 6611 network node and APPI sound similar on the surface, there are some very important differences between the two approaches.

The 6611 uses an encapsulation approach to TCP/IP connectivity. This has several consequences. First, there is some additional overhead involved in any type of encapsulation, or tunneling, because the headers used by the TCP/IP transport network are added to those used by APPN.

More significantly, the use of a tunneling strategy means that the network nodes continue to interact with one another using their native APPN protocols. TCP/IP is simply used to transport the protocols. This is where the 6611 approach differs most fundamentally from that of the APPI Forum.

At the core of the APPI movement is the desire to use open industry standard protocols to connect network nodes rather than APPN protocols which are controlled by IBM. This is where a comparison between the two approaches enters into the realm of politics and religion. The bottom line, from a customer's point-of-view, is that if a technology is available from a wide range of vendors and it meets the requirements of a wide range of customers it becomes an industry standard. People with real networks to run don't have time to argue the political and religous issues of "openness," they're looking for solutions. ■

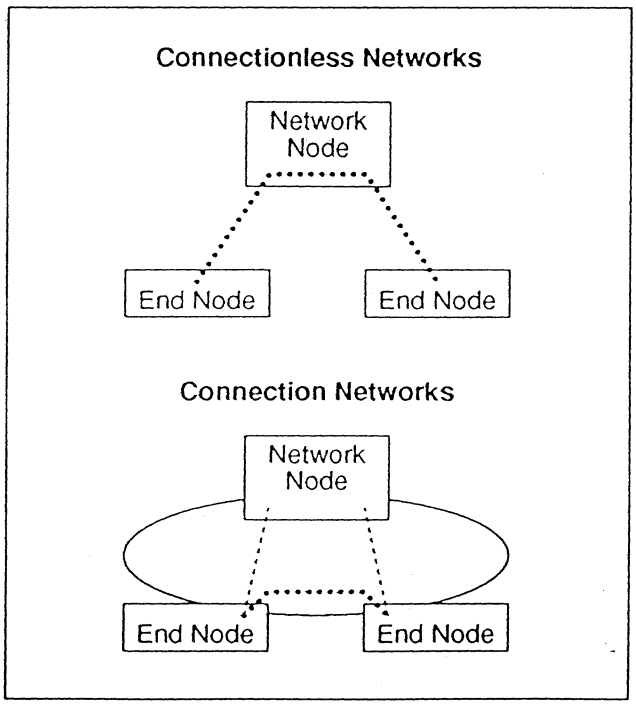

Figure 7

Architect's Corner

From ARB to AIW—New Architecture Process

by Dr. John R. Pickens

As had happened so many times in the past, American Airlines routed me from Raleigh, North Carolina through Dallas. And, as usual, I sought out my favorite airport yogurt stand. With yogurt in hand (nuts topping dripping) I would dash down the ramp to catch the flight to San Jose. Dropping my missed-connection statistic by a few percentage points, I made the flight, just barely. But as I settled into my cramped seat, my thoughts were elsewhere.

Running over and over in my mind was a conversation a few nights earlier with John Walker of the IBM Market Enablement group, the VTAM band playing in the background, hors d'oeuvres in hand. John was remarking to me that a number of attendees of the APPN Implementers Workshop (AIW) had expressed amazement at the openness of the IBM developers and architects—during speeches, on panels, in the breaks between sessions. IBMers would candidly admit problems, disagree over strategy/tactics in public, and somehow find any and every way to completely avoid the verbal two-steps of the past (like, for the question "are you going to do x?" the shuffle-double-response, "we are not saying we are not going to do x.").

This observation of John's, however, was not the focus of my mental rehashing. Rather, it was an assessment of my own reactions. Unlike the first-time attendees John was describing, I had not noticed the behavior change in this conference. I was already used to it!

And then my memory reached back to an earlier time, the mid-80s, when IBM was still holding information-dissemination conferences for analysts and consultants. Shortly after arrival, I would be surrounded by listeners/facilitators—pilot fish I called them. Raise an objection about IBM-the-company, and you would be routed to the appropriate pinstriped executive who would set things straight.

My, how things have changed in just a few years.

During earlier years IBM would often claim openness. This claim had an element of truism—for example, publication of specifications (however selective). However, such a claim had a hollow ring against the background of such public behavior. But such behavior is now gone. Rather a candor and a real willingness to open up is evident.

Perhaps the change is best reflected by two events at the first AIW in Raleigh, April 14-15, 1993:

AIW Event #1 — For the first time ever (that I can remember) IBM commits to making a specification available before—long before—the corresponding IBM product ships. The Dependent LU Requestor specification is to be made available in June 1993, perhaps 12–18 months (or more) ahead of the (IBM) product. And with requests for review comments. And with an explicit commitment to make changes to the specification based upon the comments!

AIW Event #2 — The formation of a working group to create architecture extensions to DLS (sometimes called DLSw). DLS provides a SNA and NetBios bridge-across-TCP-tunneling service, with options for local termination of LLC 2 connections. At least a dozen vendors sign up to working on the specification. Several vendors contribute suggestions for improvements to the specification, including IBM. And IBM commits to supporting the output of the group.

These are not the normal, historical methods of working with IBM. Indeed, they seem to trend toward openness.

Actually, the procedures for openness are not yet streamlined, nor are they completely articulated. It has not yet been determined how consensus will be achieved, nor what process will be used to resolve disputations. Perhaps an IETF-style mediaeval guttural "mmmmm." Perhaps an IEEE-style strait-laced balloting procedure. Certainly freewheeling use of Internet mail and Internet anonymous ftp servers to cut down on required meeting attendance. (Now when is that AIW meeting going to be scheduled in California?)

Whatever the processes, the climate is now clear for open dialogue between IBM and the multi-vendor community. Even some of IBM's harshest detractors, such as (no, I won't say the name) are now assenting to the new way of defining SNA-oriented defacto standards.

Historically the Architecture Review Board (ARB) was the method used by IBM to achieve consensus among IBM divisions on new architectures. In 1985 the then chairman of the Architecture Review Board (ARB) told us consultants that the method for achieving ARB consensus was by vote—one vote for each IBM division. However, he assured us he had extra votes in his back pocket, just in case.

Today, the extra votes are coming from the vendors. And so the baton passes from the ARB to the AIW.

Interesting, but promising times. ■