Architectural directions for opening IBM networks: The case of OSI

by P. Janson
R. Molva
S. Zatti

This paper discusses the results of a research project that developed an architectural framework for integrating non-IBM network architectures to the reference model and node structures of IBM's Systems Network Architecture (SNA). The unique features of the selected integration approach allow multiple protocol stacks to coexist and interoperate within the same computer, to share use of common physical network ports, links, and switching nodes, and to be accessed and managed through homogeneous interfaces. The architectural framework was developed for the specific purpose of integrating the Open Systems Interconnection (OSI) Reference Model to that of SNA, but its basic philosophy and key aspects turn out to be generally applicable to the integration of other network technologies as well, such as TCP/IP or NetBIOS.

Back in the mid-1970s, IBM introduced its host-based Systems Network Architecture (SNA), which was later enhanced by cross-domain SNA Network Interconnection (SNI) and decentralized Advanced Peer-to-Peer Networking (APPN) functions. In the early 1980s, IBM also introduced NetBIOS (Network Basic Input/Output System), a communication technology originally targeted at local area networks (LANs), and started offering support for the Transmission Control Protocol/Internet Protocol (TCP/IP) family and the Open Systems Interconnection (OSI) architecture to address multivendor network environments. This broad palette of network offerings raises sometimes difficult selection and compatibility issues for users of IBM products. To address user requirements and support IBM's commitment to open networking, a research project discussed in this paper sets forth a model pulling the above protocols together into one integrated architectural framework that provides a rich, but flexible, modular and uniform set of functions across all system[1] platforms and all network types.

This paper presents a general model for the integrated architecture and then focuses on the technical details of the SNA-OSI case. The paper is organized into six sections and a summary: The first section presents the motivations and objectives of the project; the second section lists the technical requirements that must be met; the third section positions the effort with respect to prior attempts at relating the SNA and OSI architectures; the fourth section stresses the basic approach of the integration project and underlines its key features; the fifth section then outlines the overall design for the resulting integrated system structure; the final section discusses in more detail one selected and particularly important aspect of the design, the integration of SNA and OSI naming, addressing, and internetwork routing mechanisms.

Motivations and objectives

The early releases of SNA enabled a host computer to communicate first with a hierarchical network of communication controllers and device control units, then with terminals and printers attached to that network. This basic architecture was later enhanced to allow two or more hosts to communicate with one another and with terminals and printers via a meshed network of communication and device controllers.[2] Then SNI allowed several such meshed SNA networks to be interconnected into essentially unlimited internetworks.[3] Most recently, the APPN architecture[4, 5] was defined to support networks of minicomputers and personal workstations, without requiring the centralized control usually provided by hosts in traditional SNA networks (often called subarea SNA networks). Since its original introduction, and through continuous enhancements, SNA is installed on thousands of Systems Application Architecture* (SAA*)[6] and Advanced Interactive Executive* (AIX*) system platforms worldwide, and is widely supported by industry for equipment designed to communicate with IBM hosts.

IBM also supports NetBIOS, which enables communication among personal computers and similar workstations attached to a local area network (LAN).[7] NetBIOS architecture is successful and is used in IBM and non-IBM LAN workstations and servers.

While SNA and NetBIOS have been copied and are supported by many manufacturers, they remain proprietary designs. Several versions of NetBIOS are in use to support the same or similar application programming interface (API).

Among nonproprietary architectures, one endorsed by many vendors, including IBM, is the TCP/IP protocol suite originally designed for the Defense Advanced Research Projects Agency (DARPA) Internet.[8] However, while TCP/IP offers a popular solution to multivendor networking because it is not a proprietary architecture, it is also not really an international standard architecture in the sense of the OSI architecture. OSI includes a much richer and evolving set of protocols, and is viewed as a key to open networking in multivendor environments of the future. Many government agencies around the world, such as the U.S. Department of Defense, require OSI on all systems they purchase. Thus, manufacturers are increasing their support of OSI. IBM is committed to open networking in multivendor environments and supports OSI on both its AIX and SAA systems.

While OSI and TCP/IP address multivendor networking issues, their existence in parallel to the proprietary networks, SNA and NetBIOS, results in two specific requirements from users of IBM products:

1. IBM must continue to offer and maintain software supporting multiple network architectures. It is not clear whether or when OSI will displace and supersede TCP/IP. If that occurs, OSI will not likely displace proprietary designs such as SNA or NetBIOS for two reasons:

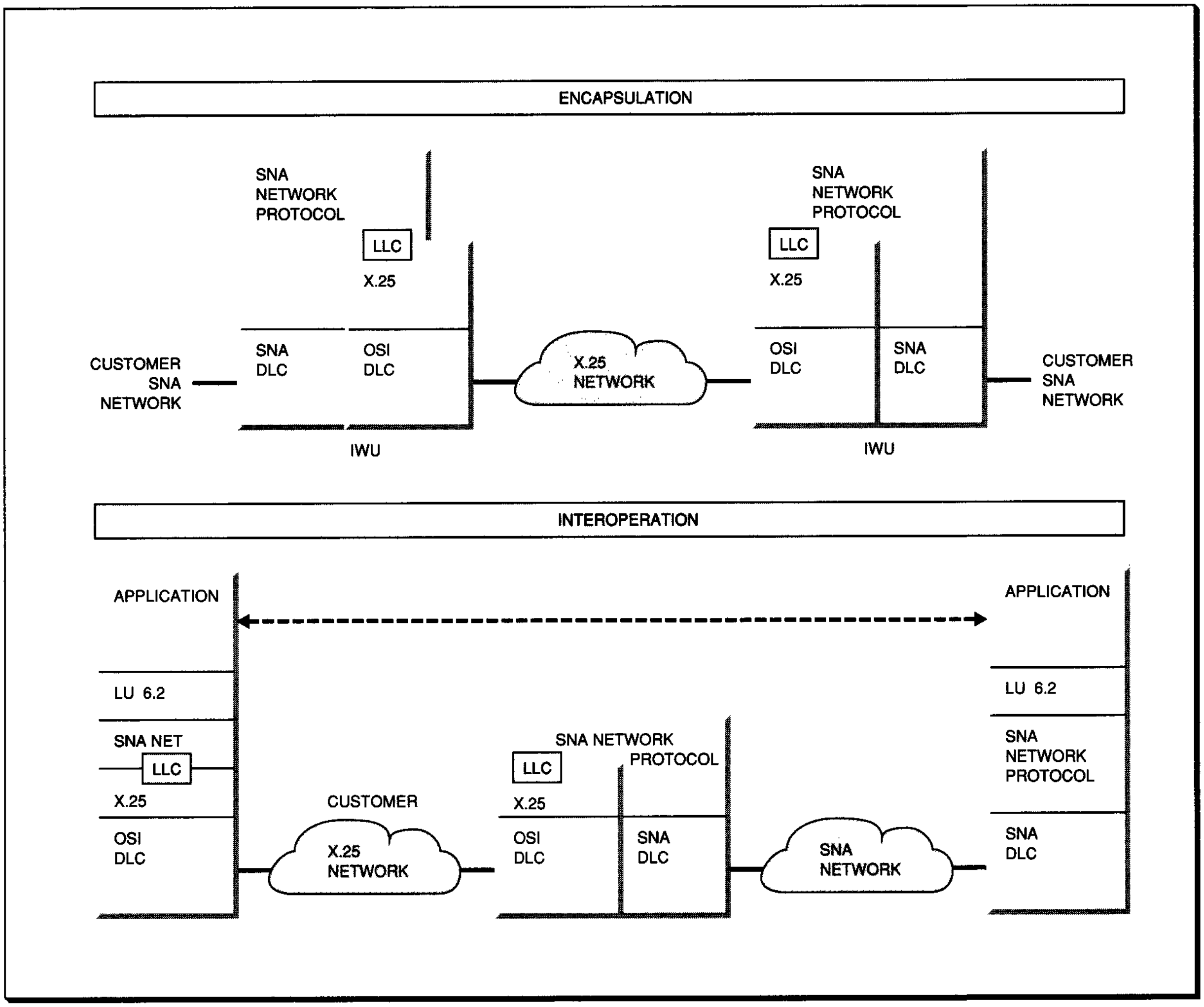
   a. These architectures represent huge investments in hardware and software that neither IBM nor users of its products can afford to discard.

   b. Users may wish to migrate selected applications from proprietary networks like SNA to nonproprietary ones like OSI. However, SNA continues to evolve rapidly to meet new requirements with new network technology, while OSI cannot be expected to evolve as fast because of the inherent delays in international standards approval procedures. SNA will continue to offer added value to users with advanced networking requirements.

2. Given that each network architecture addresses somewhat different requirements with its proper functional capabilities, users must be given the flexibility to mix and match two or more of these architectures on the same physical network or even in the same computers.

The two user requirements, open networking through support of multiple network architectures, and uniform and flexible coexistence between these on the same network or in the same computers, are the motivations for the research project discussed in this paper.

Figure 1 Transporting SNA traffic through non-SNA (OSI) networks



Technical requirements

In order to derive the detailed technical requirements prompted by the above objectives, it is useful to review three scenarios of IBM equipment use, which the integrated architecture must address.

Transporting SNA traffic through non-SNA networks. The first scenario is depicted in Figure 1, which represents the transport of SNA traffic through non-SNA networks, such as an X.25 (OSI) network in the present example. Two subcases to this scenario follow.

Encapsulation. A user owning and operating an SNA network that is physically divided into several clusters of computers wants to use (carrier) networks of another technology (X.25 in the present example) to interconnect the clusters of the user's own network and carry traffic between them. This scenario requires internetworking support that allows the user to build the SNA network using (carrier) subnetworks of other architectures. In OSI terms, this amounts to providing the user with a capability to relay SNA traffic through non-SNA networks by encapsulating the SNA protocols inside non-SNA protocols to traverse the non-SNA network. This capability is provided by internetwork gateways that are the SNA equivalent of OSI's Interworking Units (IWU).

Interoperation. A user owning and operating IBM systems attached to a non-SNA network (X.25 in this example) wants the user's applications to communicate with applications residing on remote SNA networks. This scenario requires internetworking support that allows SNA applications to be attached directly to non-SNA networks and to communicate with peer applications on remote SNA networks. In OSI terms, this requires essentially the same relay capability as in the encapsulation scenario to relay SNA traffic between SNA and non-SNA networks. Only one IWU is required since, in this case, the SNA traffic ends up in but does not traverse the non-SNA network.

Internetworking capabilities similar to the above, though not as general, have long been available from IBM to carry subarea SNA traffic into or through X.25 networks, using the Network Control Program (NCP) Packet Switching Interface (NPSI) product.[9] The IBM Application System/400* (AS/400*) system also allows the transport of APPN traffic through X.25 networks. Similar capabilities are desirable on multiple system platforms to carry SNA subarea and APPN traffic across X.25 as well as other non-SNA networks (e.g., TCP/IP and networks implementing the OSI Connectionless Network Protocol [CLNP]).

Transporting non-SNA traffic through SNA networks. The second scenario, depicted in Figure 2, represents a situation requiring the transport of non-SNA traffic—in the present example OSI, through SNA networks. Again, there are two subcases to this scenario. The levels representing layers of OSI architecture (i.e., 3–7, 3a, 3b, 3c) are shown in this and several subsequent figures.

Encapsulation. A user owning and operating a non-SNA network (OSI in the present example) physically divided into several clusters of computers wants to use a network based on SNA technology to link the clusters of the user's non-SNA network and carry traffic between them. This scenario requires internetworking support that allows the user to build the non-SNA network using SNA technology for subnetworks. Seen from the OSI point of view, this is a classic scenario requiring a capability to relay OSI traffic through whatever subnetwork is used. In this scenario where the subnetwork technology is SNA, the relay capability must encapsulate the OSI protocols inside the SNA protocols to traverse the SNA subnetwork. In OSI terminology, this relay capability is provided by interconnection systems called IWUs. They relay what OSI calls Subnetwork Independent Convergence Protocols (SNICP), at layer 3c in the figure, over Subnetwork Dependent Access Protocols (SNAcP), at layer 3a in the figure. OSI includes two basic protocols at the SNICP layer: X.25[10] for connection-oriented networking, and CLNP[11] for connectionless networking. In principle, it accepts anything (e.g., X.25 or SNA, as in this scenario) at the SNAcP layer.

Interoperation. A user owning and operating an SNA network wants the user's applications to communicate with applications on remote non-SNA (OSI in this example) networks (e.g., networks operated by business partners, subsidiaries, suppliers, or customers). This scenario requires internetworking support that allows non-SNA applications to be attached directly to SNA networks and to communicate with peer applications on remote non-SNA networks. In OSI terms, this requires essentially the same relay capability as in the encapsulation scenario to relay non-SNA traffic into an SNA network. Only one IWU is required, since the non-SNA traffic ends up in but does not traverse the SNA network in this case.

Products offering internetworking capabilities similar to the above are also available from IBM, to carry X.25 (OSI) traffic into (NPSI[9]) or through (X.25-SNA Interconnection [XI][12]) SNA networks. In the future, a flexible set of similar products, running on a number of system platforms, is desirable to carry non-SNA traffic (e.g., OSI, TCP/IP, and NetBIOS) across subarea SNA as well as APPN networks.

Mixing SNA and non-SNA (OSI) traffic on the same network. A third scenario for integration is depicted in Figure 3. Here a user with diverse application requirements (e.g., business and engineering) owns and operates equipment from multiple vendors so that the user needs to mix several architectures on the same physical network (links and nodes). In such a situation, the IBM equipment will support multiple protocol stacks simultaneously, including SNA (and/or NetBIOS) and OSI (and/or TCP/IP), so the user can capitalize on existing investment in and advantages of SNA (NetBIOS), while allowing interoperation with equipment from other vendors.

Figure 2 Transporting non-SNA (OSI) traffic through SNA networks

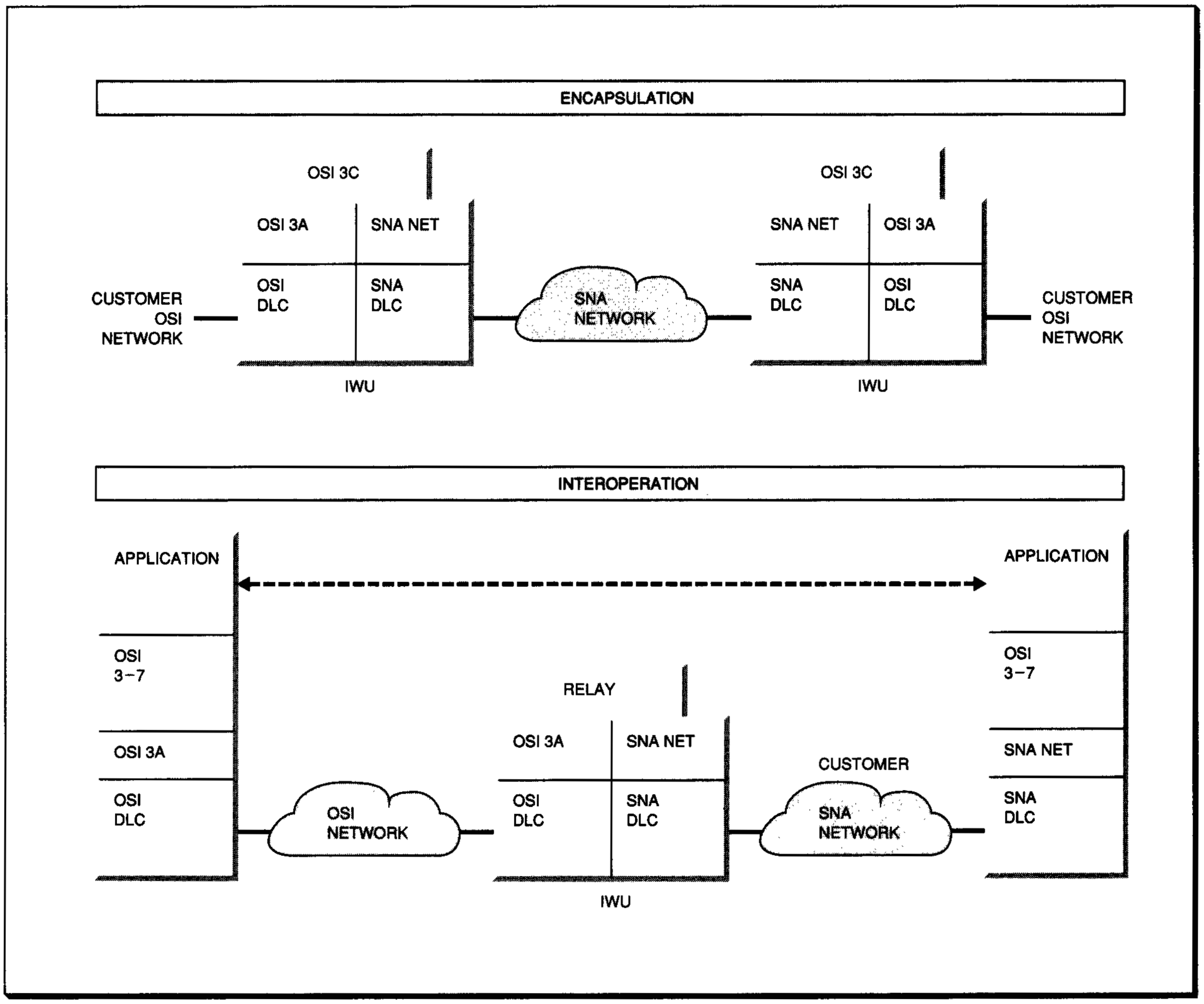
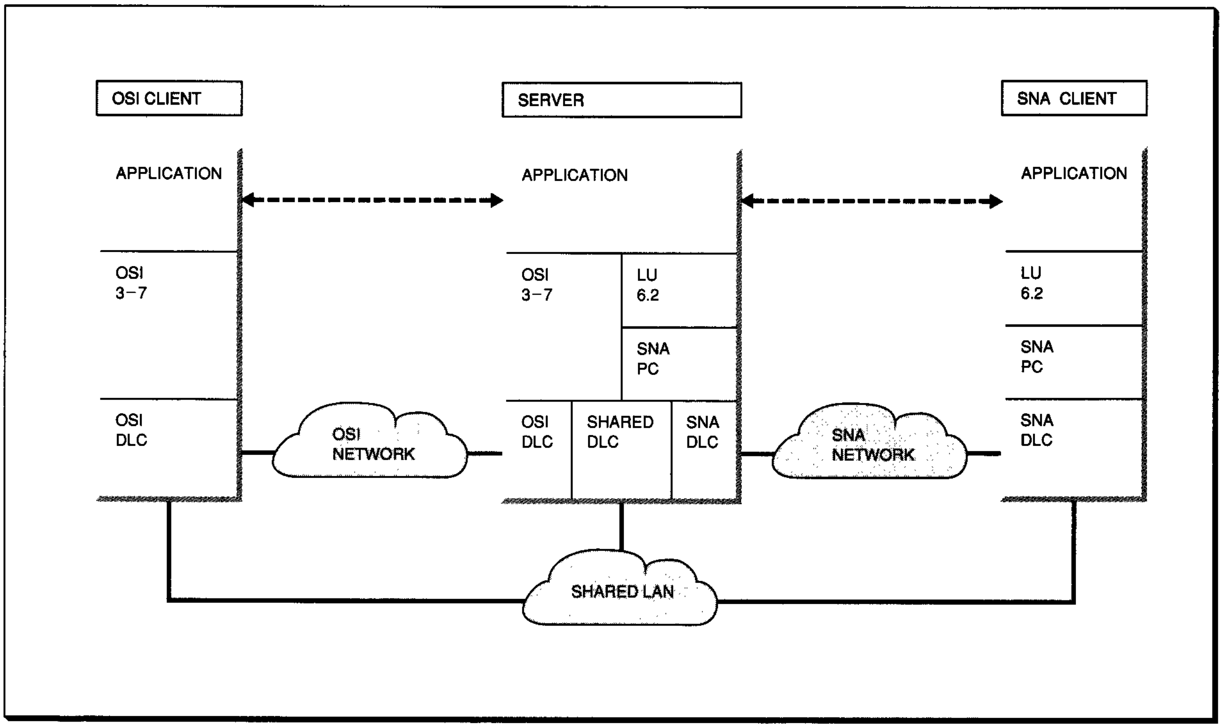


Figure 3 represents a server application on an IBM system (in the middle) communicating simultaneously over the same LAN (or possibly over separate physical networks), with client applications located respectively in an IBM workstation supporting only SNA (right) and an IBM or other vendor station (left) not supporting SNA. Contrary to previous scenarios, the issue here is not internetworking. Instead, it is sharing physical network components, offering Common Programming Interfaces (CPIs) for communication across networks of different architectures, and providing operators and users with homogeneous installation and management views of the heterogeneous network environment. Uniform CPIs for communication across different protocol stacks provide the same communication semantics, allowing programmers on the integrated server to write applications that can use SNA or non-SNA communication protocols transparently. By allowing multiple protocol stacks in the same system to share access to physical links, the integrated server is enabled to use two or more stacks side-by-side to communicate simultaneously with heterogeneous clients without forcing all clients to implement the same network architecture.

Homogeneous installation and management procedures across the different network architectures on the integrated server make the config-

Figure 3 Mixing SNA and non-SNA (OSI) traffic in the same system



uration, operation, maintenance, and use of the heterogeneous network by that server as uniform and convenient as possible for the user.

Summary of technical requirements. The previous scenarios provide useful insight with which to derive technical requirements for achieving uniform and open networking in IBM systems. These requirements are summarized as follows:

- Rich connectivity. It is necessary to be able to connect IBM systems to different SNA and non-SNA networks, possibly to several such networks simultaneously (see the third scenario).
- Full interoperation. IBM systems on any of the important networks must be able to communicate with other IBM or non-IBM systems on different networks through flexible internetworking capabilities, allowing the transport of non-SNA protocols through SNA networks and vice versa (see the first and second scenarios).
- Sharing. The integrated architecture must allow full sharing of resources such as physical links, intermediate systems, management mechanisms, operator consoles, etc. among heterogeneous networks in a common administrative domain (see the third scenario).
- Homogeneity. Network users, operators, and programmers must be able to use, manage, and access heterogeneous networks in a homogeneous way (see the third scenario).

Based on this set of requirements, an architectural framework was developed for integrating the OSI Reference Model to that of APPN. The basic philosophy and key aspects of the integrated model are applicable to the integration of non-SNA designs such as TCP/IP or NetBIOS to SNA in general. The rest of the paper focuses on the specific case of integration of OSI to APPN.

Related work

The relation between SNA and OSI has been a subject of several studies in the past.[13] However, these studies have focused generally on interconnection between SNA and OSI networks.[9, 12, 14–16] In fact, the focus has been on interconnection above the network layer (layer 3) of OSI, whereas OSI normally places internetworking at the network layer. Other studies investigated issues dealing with the integration of directories, and naming and addressing mechanisms between OSI and architectures such as SNA.[17, 18] However, none of these studies looked at total integration of OSI and SNA to the extent discussed in this paper, i.e., coexistence of the two stacks side-by-side in the same computer, with full internetworking at the network layer, link sharing, common CPIs, and homogeneous network appearance to users and operators.

Approach and key features

The first feature that differentiates the discussed design is its approach to internetworking. The design rigorously follows the OSI Reference Model, placing the internetworking functions at the network layer, so that the upper portion of one protocol stack (down to and including the equivalent of the OSI SNICP layer) can use the lower portion of another one (up to and including the equivalent of the OSI SNAcP layer), effectively enabling the lower-layer networking mechanisms of the latter to transport the higher-layer end-to-end protocols of the former. This design introduces a rich set of uniform internetworking capabilities, and it does not preclude the development of specialized higher-layer gateways to allow other applications to interact using completely different protocol stacks.

A second distinguishing feature is its mixing of different communication architectures on the same physical networks through the definition of a communication system structure that allows flexible selection and smooth coexistence among multiple protocol stacks residing side-by-side in the same computer.

A third salient feature of the design, which results directly from the above two, is the ability to share physical links and subnetworks among co-located protocol stacks, which is achieved by giving each protocol stack a different Link (or Network) Service Access Point (LSAP/NSAP) address, in OSI terms. This design is followed for all links and subnetworks that can potentially be shared between SNA and OSI, e.g., X.25, LANs (including FDDI), ISDN, ATM connections, etc.

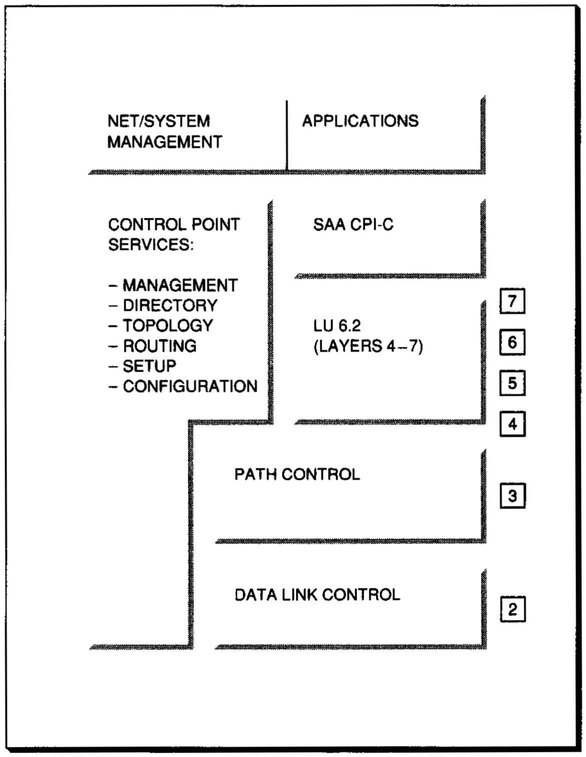
A last but essential difference between this architecture project and previous ones is the objective of homogeneity. Where different protocol stacks offer similar communication semantics, uniform CPIs for communication are provided over the different stacks so applications can use the stacks transparently. Similarly, homogeneous network control, management, and operation mechanisms are shared across the different networks: a common Network Management Focal Point using OSI Common Management Information Protocols (CMIP)[19] can manage OSI as well as SNA networks; uniform application-layer naming conventions and directory structures are defined; network-layer addressing syntaxes are unified (e.g., registered SNA network identifiers and addresses map directly into OSI network addresses); SNA networks support OSI interdomain routing so they can participate fully in global OSI networks; uniform network security mechanisms are supported across the different architectures; etc. Not only does the design aim at homogeneity between different network architectures, but it also aims at homogeneity of the implementations of these architectures on different (AIX and SAA) system platforms.

The approach taken to meet the technical requirements and support the key features consists of integrating the reference model of OSI to the architectural model of an APPN system and arranging for both protocol stacks to share the same (enhanced) control and management mechanisms in the resulting integrated system model.

The architectural model of an APPN system serves as a reference for APPN product implementations. A high-level structural view of this model is given in Figure 4. The figure shows how all network control and management mechanisms are collected in one component, called the Control Point (CP), which at the same time provides management services to and uses the logical unit (LU) 6.2 protocol stack, the APPN Path Control (PC) network protocol and the Synchronous Data Link Control (SDLC) and LAN links available to it.

The OSI Reference Model[20] is the blueprint for implementing OSI and was used in IBM's OSI/Communications Subsystem implementation, which is at the heart of all IBM products supporting OSI. The OSI/Communications Subsystem code already implements or will eventually implement all OSI layers and important protocols, including the CMIP Network Management protocols,[19] the X.500 directory protocols,[21] MAP 3.0, FTAM, MHS, TP, etc. The code is designed to be portable across multiple system platforms. It is or will soon be

Figure 4 APPN system structure



offered on all SAA and AIX systems, thus presenting uniform application interfaces and configuration mechanisms across all these platforms. A high-level representation of the OSI/Communications Subsystem code structure is depicted in Figure 5. (An operating system services component is omitted from the figure.) The figure highlights immediate similarities with the APPN system model described earlier, as it shows an OSI Subsystem Management component, which both provides management services to and uses the seven layers of OSI protocols. The line stepping through the network layer indicates that the OSI/Communications Subsystem code includes its own support for the SNICP part of the layer, while it relies on the system platform on which it is installed to provide the necessary support for the SNAcP (e.g., LAN and X.25 attachment).

Given the similarities (at a very high level) between the APPN and OSI system structures, the integrated system structure adopted is depicted in Figure 6. It consists of two main components represented by the two planes of the figure: the background plane includes a common set of mechanisms providing system and network control and management functions (including directory, topology, routing, configuration, etc.) to all protocol stacks present in the system; the foreground plane includes the individual protocol stacks themselves (APPN on the left, OSI, etc. toward the right) sandwiched between internetworking and shared subnetwork/link mechanisms below, and uniform CPIs supporting user applications above.

Since it is not possible to cover in detail all the issues that arise in the design of such an integrated system model, we focus on selected key aspects. Specifically, the next section is devoted to an examination of the integrated system structure, which will provide an opportunity to discuss link sharing, CPIs, and other homogeneity issues in more detail. The last section discusses the directory, naming, addressing, and routing issues raised by the internetworking and homogeneity requirements.

Integrated system structure

Figure 7 shows a detailed picture of the integrated system structure discussed in the next section. This discussion also alludes to work done as part of four related development projects involving three different IBM laboratories: a link architecture project undertaken by the La Gaude and Raleigh laboratories, an SNA restructuring project conducted in Raleigh, a CPI integration project carried out by Palo Alto, and a management architecture project in Raleigh.

Co-located protocol stacks. Figure 7 is composed of the shared control point and network management functions (on the left); the LU 6.2 protocol stack interfacing with the APPN Path Control, the data link control protocols interfacing with the physical ports and links (toward the middle); and the OSI protocol stack as implemented by the OSI/Communications Subsystem interfacing with other CPIs (on the right). Within the context of this model and for the rest of this paper, the OSI/Communications Subsystem is referred to as the OSI component, or OSI LU, for reasons to be explained soon.

One of the questions raised by the juxtaposition of the two protocol stacks within the same system is the nature of their mutual relationship and of the appearance of the OSI/Communications Subsystem to other system components.

In the APPN system model, LU 6.2 instances, i.e., instances of the LU 6.2 protocol stack, define at the same time addressable ports on the APPN network and the applications using the network through these ports. Each LU 6.2 presents three different interfaces, shown in Figure 7: a lower one to the Path Control networking mechanism, an upper one to applications (through the SAA CPI for Communications [CPI-C][22]), and a lateral one to the CP, via the Shared Control Point. The lateral interface provides the LU 6.2 with access to CP services (directory, topology, routing, configuration, etc.). The upper LU interface provides an application with access to the LU 6.2 services and protocols, much as an OSI Presentation Service Access Point (PSAP) offers an application access to the OSI services and protocols. The lower LU interface provides the LU with access to the Path Control network service, much as an OSI Network Service Access Point offers OSI Transport Entities (TEs) access to the OSI network service.

An essential difference, though, is that in OSI many PSAPs may be tied to (located at) one common NSAP, whereas in SNA each PSAP is tied one-to-one to its own NSAP, as represented in Figure 7. In OSI, PSAPs are associated with applications, while NSAPs are associated more with systems; and the PSAP addresses of applications located at the same NSAP are hierarchically structured and include the address of that common NSAP. In SNA, by contrast, both PSAPs and NSAPs are associated with individual applications, and they are tied together by a common address, called the SNA LU name; systems (called nodes in SNA) are associated with their CP, which is denoted by its CP name.

SNA LU names serve both as presentation and network addresses, and individual applications located on the same system never share the same network address (LU name), whereas OSI PSAP addresses are separate and different from NSAP addresses, and individual applications located on the same system share a common NSAP address, which is then hierarchically included in their PSAP addresses.

The question thus arises as to how the traditional components of an APPN system may regard a new component such as OSI, which supports network addresses that are distinct from presentation addresses, a concept not present in SNA. The answer to this question depends on the viewpoint.

Figure 5 OSI communications subsystem structure

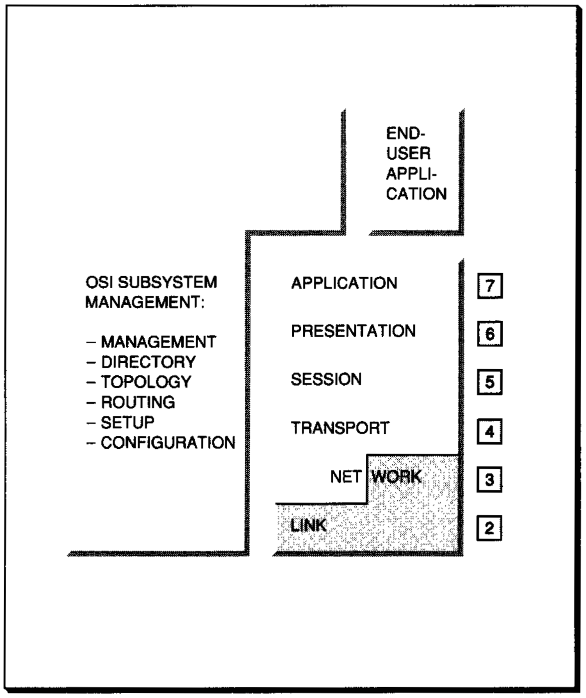


Figure 6 Integrated system structure

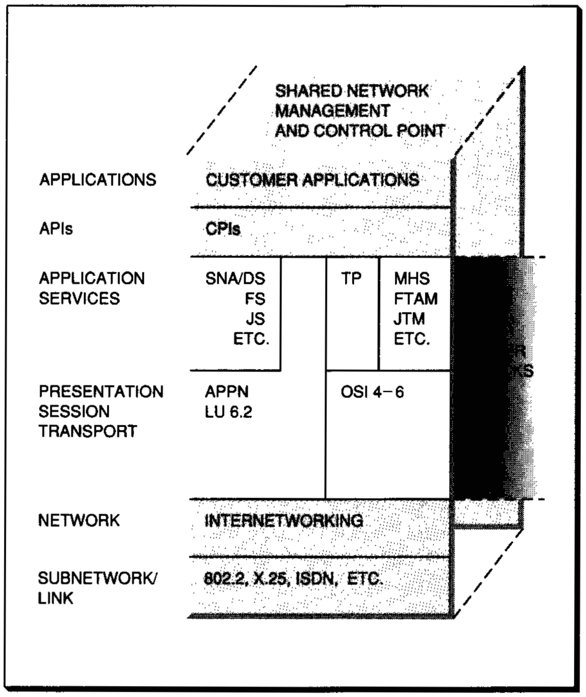
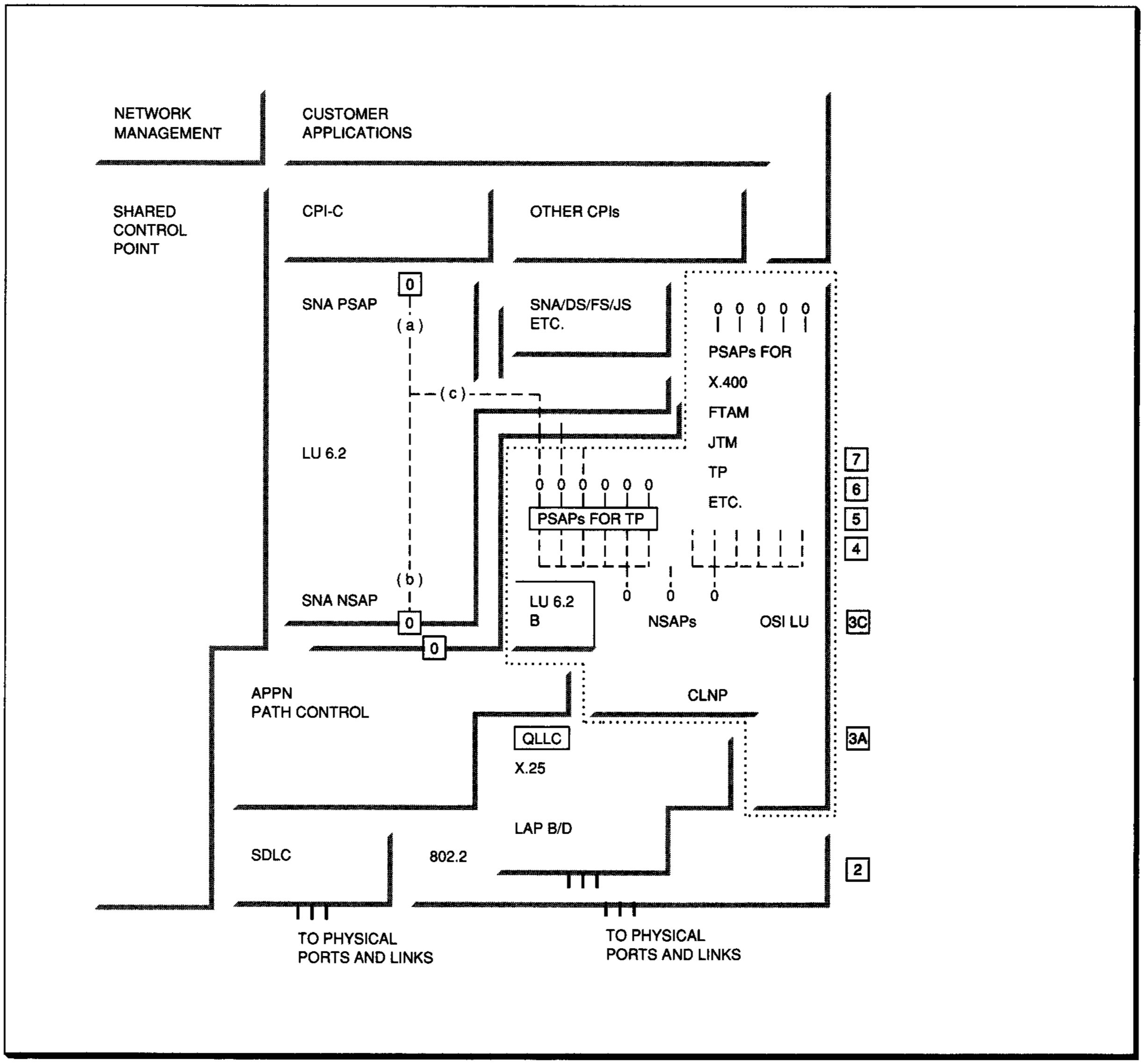


Figure 7 Detailed system structure



The CP, the LUs, the Path Control network, and the link/subnetwork layer of the APPN system each relates to the OSI component in a different way, because each has a different interface to it. The interesting aspect of this design is precisely that it is possible to confer different appearances to the new OSI component without perturbing the cohesion of the original APPN system model.

To the link/subnetwork services layer, the OSI component appears as an element similar to Path Control in the sense that it is able to receive link/subnetwork traffic directly like Path Control, as explained in the next section on link sharing.

To the CP and the Path Control network, the OSI component appears as one LU with its own LU name. The OSI component is therefore often referred to as the OSI LU on its system. As such it may engage in network connections (called sessions in SNA) with other OSI LUs attached to the same SNA network.

To applications, the OSI component appears as a collection of LUs offering access to OSI services, one LU for each supported PSAP, each with its own LU name. The role of the OSI LU name as seen from Path Control and the CP, and the relationship between the multiple LU names associated with the PSAPs it supports will become clear in the later section on naming and addressing issues.

Shared links and subnetworks. As a result of the existence of two (or more) protocol stacks side-by-side in the same system and the desire that they be able to share access to physical network facilities such as links and network adapters, the Lower Layer Architecture of SNA has been revised in a parallel project involving the IBM laboratories in Raleigh, North Carolina, and La Gaude, France.

The essence of this architectural revision consists of assigning different Link Service Access Points (LSAPs in OSI terminology) to the OSI layers 3–7 portions of different protocol stacks, so that incoming layer-2 traffic can be fanned out to the layer 3 of the proper stack based on its target LSAP address. Thus, as represented in Figure 7, incoming X.25 or LAN traffic may be destined to different LSAP addresses to allow the different protocol stacks in the system to entertain communication with their respective peer instances in other systems over the same physical ports and links.

Certain link protocols (layer 2) may of course never be used by certain layers 3–7 stacks simply because they were not defined as part of the same architecture. For instance, SDLC is part of SNA but not of other stacks, so it will be used only by SNA Path Control and never shared with other stacks. On the other hand, the IEEE 802.2 LAN Logical Link Control (LLC)[23] protocol or the X.25 subnetwork protocol are defined as part of both SNA and OSI, and can thus be shared, simply using different LAN and X.25 addresses for each stack.

The SNA and OSI stacks use X.25 subnetworks in slightly different ways. First, the reliability of virtual circuits in certain commercial X.25 networks is not up to the level expected by the SNA stack and provided by SDLC, so that when an SNA network spans an X.25 subnetwork, an additional LLC procedure (QLLC) is needed around the X.25 network to raise its perceived reliability to that expected by SNA. By contrast, some OSI Transport protocols (e.g., Class 4) include recovery mechanisms that allow them to use X.25 subnetworks directly. Second, SNA uses the X.25 Packet Level Protocol (PLP) protocols only across wide area network (WAN) links implementing the HDLC LAP B (High-Level Data Link Control, Link Access Procedure B), whereas OSI allows the use of X.25 protocols also across 802 LANs.

Internetworking. This section discusses internetworking issues in some more detail, namely the transport of SNA protocols through OSI networks, and vice versa.

An architecture implemented by the NPSI product[9] has been defined and used to transport SNA traffic across X.25 networks. Except for the fact that this architecture clearly takes an SNA perspective, it actually follows fairly closely the OSI model of internetworking in that it defines essentially an SNICP layer over an SNAcP layer. The SNICP is SNA Path Control (as opposed to CLNP or X.25 PLP in OSI), while the SNAcP is X.25. The QLLC layer between SNA Path Control and X.25 constitutes an example of what OSI calls a subnetwork Dependent Convergence Protocol (SNDCP), namely an optional layer inserted to match the service offered by an SNAcP to that expected by an SNICP. This architecture is preserved with only one significant change in the integrated system: In the present NPSI design, the mapping from SNA internetwork addresses to X.25 subnetwork addresses is ad hoc; in fully integrated systems, a much more comprehensive and general solution to that address mapping question is proposed in the later section on integrated naming and addressing. (A similar architecture can easily be envisioned for the transport of SNA protocols through connectionless OSI networks. For lack of a clear and wide user requirement at the time this research was performed, the issue was not pursued in further detail. It would present no really fundamental challenge.)

The dual internetworking function, namely the transport of OSI traffic through SNA networks, is a requirement and presents additional challenges. The XI product[12] allows the transport of X.25 traffic through SNA subarea networks of communication controllers. A generic solution is required to transport any type of OSI network protocols, CLNP as well as X.25, through APPN as well as subarea networks. This is discussed below.

In this scenario, SNA offers the SNAcP function, while OSI dictates that the SNICP be either X.25 PLP or CLNP. The SNA Path Control layer does not provide a complete and self-contained networking mechanism. It implements only the routing part of a connection-oriented network, where the connections, called sessions, must be anchored inside LU instances whose lowest layer maintains the session state information. Thus a non-SNA protocol layer such as the SNICP layer of the OSI component cannot just drop packets into the SNA Path Control mechanism and expect that these will be routed properly. It must first pass these packets through an SNA session anchor for proper processing and only then down to Path Control. However, the session anchor mechanism of a SNA LU is only a small piece of the complete LU component, so that a solution is needed to allow the OSI component to access just the session anchor mechanism without having to use the entire LU protocol stack.

To this end, a structure reflected in Figure 7 was proposed that embodies two design decisions:

1. The decision to use LU 6.2 sessions as the foundation for carrying non-SNA traffic across SNA networks, because they are supported by both APPN and subarea SNA networks
2. The decision to split the existing LU 6.2 protocol stack in such a way that the basic session anchor mechanism is separated from the higher-level protocols so that it becomes accessible to other higher-level non-SNA protocols as well

Splitting the LU 6.2 stack required defining an internal interface between two new subcomponents, respectively called LU 6.2A (above the interface) and LU 6.2B (below it), in such a way that the 6.2A/6.2B combination conforms in all respects to the original LU 6.2 architecture, while the 6.2B stack alone provides just the necessary anchoring mechanism for full-duplex sessions required by non-SNA protocols to cross SNA Path Control networks. The interface through LU 6.2 and the 6.2A and 6.2B components are in the process of being defined.

As a result of the above work, access to the SNA network is provided to the OSI component by tying it to an LU 6.2B instance. As such, the OSI component acquires the appearance of an LU with respect to the Path Control and CP components, as suggested earlier, and is referred to as the OSI LU. Two OSI LUs in different integrated systems attached to the same SNA network can thus communicate with one another by exchanging SNICP (X.25 or CLNP) Protocol Data Units (PDUs) across SNA sessions they can establish, just as any other LU 6.2 in the network. For instance, an OSI ES (end system) and an OSI IS (intermediate system) or two OSI ISs attached to the same SNA network can communicate with one another in this way to relay OSI traffic through the SNA network, as suggested in the second scenario discussed earlier.

When the IS function in an OSI LU receives an SNICP PDU (CLNP Unit_Data or X.25 Call_Request) that it must forward to some remote destination NSAP, it must be able to map that destination NSAP onto an SNAcP address on the next subnetwork, i.e., the OSI LU name of the target ES or the next IS on the route to the target. These address mapping and routing issues are discussed later in the section on naming, addressing, and routing.

Common CPIs. Moving from the lower interfaces of LUs to their upper one, the present section discusses the unification of CPIs across the SNA and OSI protocol stacks.

OSI includes a growing set of application services such as Job Transfer and Manipulation (JTM), remote Virtual Terminal (VT) access, File Transfer, Access, and Management (FTAM), electronic mail and Message Handling Service (MHS), Transaction Processing (TP), Commitment, Concurrency, and Recovery (CCR), Remote Database Access (RDA), Remote Operation Service (ROS), etc. Some of these services use one another but they are otherwise generally independent and use OSI layers 1–6 directly. SNA has comparable services, e.g., Job Services (SNA/JS), File Services (SNA/FS) Distribution Services (SNA/DS), etc. In contrast with OSI application services, many SNA services do not use SNA layers 1–6 directly. Instead, they are built on top of the SNA Transaction Processing service provided by the LU 6.2 stack through the SAA CPI-C interface. The Transaction Processing service thus plays a preponderant role in SNA.

Without ruling out, but also without exploring the possibility of common CPIs for other application services, the present research project has thus focused on integration of SNA and OSI Transaction Processing services under the SAA CPI-C interface. The SNA and OSI Transaction Processing services are semantically very close to one another. Essentially the same set of CPI-C primitives can be used to access the services of either protocol stack. Four adjustments and enhancements to CPI-C and the associated mechanisms for transaction concurrency and recovery are, however, necessary. These are discussed in the following paragraphs.

Once a conversation or dialog is set up, all subsequent verbs referring to it use a conversation (or dialog) identifier.

AE Title support. In its original LU 6.2-based definition, the CPI-C interface primitives (verbs) allow transaction programs (TPs) residing in different LU 6.2 instances to communicate with one another over connections called conversations.[24] For one TP to cause the establishment of a conversation with some target TP, the calling TP must provide its LU 6.2 instance with the name of the target TP and the name of the LU 6.2 instance at which that target TP is located. A TP normally does not manipulate target TP and LU names directly. It uses local aliases called symbolic_destination_names (SDN). These are mapped to target TP and LU name pairs through CPI-C side information that can be either initialized from a file or defined dynamically through appropriate CPI-C verb calls.

Transposing that model in the OSI Application context, the CPI-C verbs must allow what OSI calls Transaction Processing Service Users (TPSUs)[25] residing at different OSI Application Entities (AEs) to communicate with one another over connections called dialogs. In this context, a TPSU is also referred to by a name (TPSU name) and an AE by a Title (AE Title). Extending CPI-C support to cover the LU 6.2 and OSI TP protocol stacks transparently implies that SDNs must be able to refer indifferently to a TP denoted by a TP name located at some named LU on an SNA network or to a TPSU bearing a TPSU name and located within some AE bearing an AE Title in an OSI network. CPI-C side information tables must thus be enhanced to accept TPSU names and AE Titles, respectively, wherever they use TP names and LU names today.

CPI-stack binding. When a CPI-C verb is issued to start (allocate) a conversation or dialog, to send or receive some information over it, or otherwise manipulate it, that verb is always passed to the underlying LU 6.2 instance (see (a) in Figure 7). However, it may have to be switched over to the OSI component in case its target is an OSI TPSU. For the opening (allocation) verb of a conversation or dialog, this switching decision is determined by the nature of the target names found in the CPI-C side information. If TP and LU names are found, the verb stays within the LU 6.2 stack (b), otherwise it is passed to the OSI TP stack (c). All LU 6.2 instances have addressability to the OSI component so they can pass it such verbs. Once a conversation or dialog is set up, all subsequent verbs referring to it use a conversation (or dialog) identifier, which is by then bound to either a conversation instance within the LU 6.2 or to a dialog associated to some PSAP within the OSI component, thus determining the switching decision implicitly.

Transfer syntax. A third required enhancement to CPI-C is the ability to negotiate a transfer syntax in the way OSI does. This is not an issue in an SNA network because only a very small number of different syntaxes occur in SNA networks, so that the mapping from any of these to any other is well defined. Data conversion is always performed at the sender side so that the transfer syntax is always the syntax of the receiver. By contrast, OSI allows negotiating the transfer syntax to be either that of the receiver, that of the sender, or some other commonly agreed upon syntax (typically the one defined by the OSI Basic Encoding Rules [BER]).

Shared transactions. Finally, once an application TP may access indifferently the LU 6.2 and OSI TP stacks across the CPI-C interface, the possibility immediately arises that it could access both simultaneously within the same transaction, meaning that a transaction may involve a set of TPs and TPSUs on interconnected SNA and OSI networks. This in turn implies that all TPs and TPSUs within one system must be under the control of a shared transaction scheduler and that the transaction synchronization mechanisms provided by the system to its local TPs and TPSUs through the SAA CPI for Resource Recovery (CPI-RR) must also be integrated, centrally controlled, and commonly accessible to the LU 6.2 and OSI TP components to allow the coordination of transaction operations involving both protocol stacks.

Common control and management mechanisms. The focus of the discussion now moves to the left side of Figure 7, namely the CP and the common network management services it provides to both protocol stacks. Given that multiple architectures may coexist on the same physical network, sharing common links and nodes, they must coordinate themselves in managing these physical resources. Thus, network management must be capable of providing common services to all protocol stacks and to use any of them for transport of its own PDUs.

To this end, the key features of the recommended design are:

- To continue supporting the SNA Network Management formats and protocols (e.g., alerts, etc.) for migration reasons, but to offer parallel support for and meet future requirements with the OSI Network Management services and protocols, specifically allowing the transport of management PDUs over any network, encoding these PDUs in a negotiable OSI transfer syntax, and using the OSI Common Management Information Protocol[19] provided by the OSI/Communication Subsystem at the application layer.
- To register all managed systems in a CCITT (Consultative Committee on International Telegraph and Telephone) X.500[21] or functionally equivalent directory, using X.500 naming conventions, so that all managed objects within these systems can be organized into structures grafted onto the X.500 Directory Information Base (DIB), and identified through outgrowths of the X.500 naming tree.

In addition to providing integrated network management services, the CP also provides directory, topology, and route control services to all stacks. Concerning routing, for instance, the OSI ES-IS Intra-domain Routing Information Exchange Protocol[26] is supported by the OSI/Communications Subsystem, but the topology and routing information it conveys is passed to/from the CP, which maintains the actual topology databases for OSI as well as SNA, and computes shortest-path-first routing tables for both stacks. Interdomain routing and application directory issues are the subject of the next section.

Integrated naming, addressing, and routing

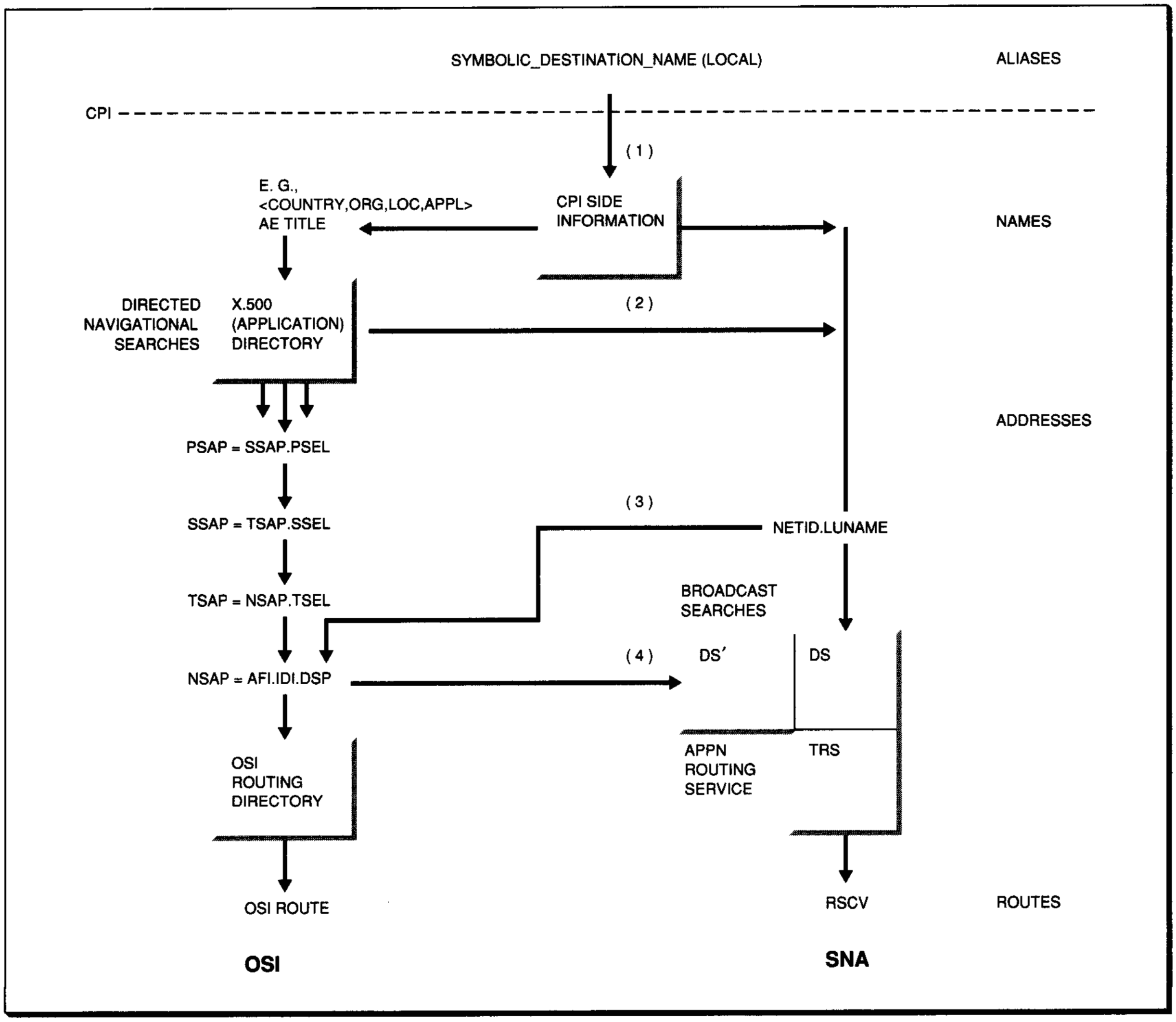
An overall picture of the integrated directory, naming, addressing, and routing issues to be discussed in the remainder of this paper is given in Figure 8. To understand what the figure represents, it is necessary to explain the operation of its SNA and OSI components separately, then to discuss their integration.

SNA naming and addressing overview. The discussion starts from the top of Figure 8 with the SNA mechanisms on the right side. When an application wants to communicate with a partner application, it issues a CPI-C conversation allocation primitive (verb) indicating the desired partner by means of a local alias called symbolic_destination_name (SDN), as explained earlier.

Through the CPI-C side information, the SDN is mapped into the LU name of the desired target application, among other things. This target LU name must be known in advance and saved in the CPI-C side information for the SNA communication to take place. To guarantee uniqueness of LU names, they are qualified by SNA network identifiers (Netid s), so that the format of LU names is actually Netid.LUname, where the Netid and LU name are each eight-character strings. With the proper Netid.LUname, the LU 6.2 protocol stack underlying the CPI-C can establish a connection across the SNA network to reach the target application LU 6.2; this connection is called an LU 6.2 session. To reach the target LU, the connection request, called a BIND PDU, must carry a source Routing Service Control Vector (RSCV) indicating the path to be taken by the future session across the network. This RSCV is built through a two-step process.

In the first step, a network Directory Service (DS) is invoked in the local CP—or, if the calling LU is located at an end node, in the CP of the nearest network node—to search the DS tables in all network nodes until a target LU with the given LU name is located in one of them. This is achieved by propagating a search request across all links to all network nodes, in a sense simulating a broadcast process. Then in the second step, a Topology and Routing Service (TRS) is invoked in the node

Figure 8 Integrated directories, naming, and addressing



that initiated the search to compute the best RSCV leading to the location discovered in the first step, based on network topology information available in all network nodes.

OSI naming and addressing overview. Moving to the left side of the Figure 8, one can observe that OSI naming and addressing, while fairly different in its details, bears similarities to the SNA mechanisms.

In OSI, as explained earlier, applications bear AE Titles. Unlike LU names, which are really addresses meant for systems to use, AE Titles are human readable strings of typed name tokens, indicating typically the country, organization, and administrative unit of the corresponding application within a hierarchical X.500 Directory Information Tree (DIT). For one application to establish a connection, called an association in OSI, with another application known by its AE Title involves a two-stage process.

In a first stage, a Directory User Agent (DUA) in the calling system must contact a Directory Service Agent (DSA) in the same or some other nearby system to locate the desired target application, i.e., to obtain an address corresponding to the AE Title. The DSA obtains this address information through a directed navigational search. The DSA parses the AE Title of the target application and, depending on information in its own directory and knowledge it has about DSAs in other locations, organizations, and countries, it answers the query right away if it can or passes it on (navigates) to one or more presumably better qualified DSAs somewhere else in the network until information about the desired target application is located. The set of protocols used between DUA and DSA or among DSAs is specified by the CCITT X.500 directory standard.[21]

The search then yields the desired addressing information, as well as possibly many other attributes that the calling application may have requested about the target application. Indeed the X.500 directory allows storing much more than just addressing information about applications. The addressing information consists of the address(es) of the PSAP(s) at which the application can be found. Such an address is hierarchically structured as the concatenation of presentation-, session-, and transport-selectors, together with an NSAP address for the system on which the target application resides. Within certain length limits, the allowable syntax and semantics of selectors are unconstrained. However, the syntax and semantics of an NSAP address are subject to well-defined standards. An NSAP address contains first an Authority and Format Identifier (AFI) field that determines how the rest of the address is to be interpreted, then an Initial Domain Identifier (IDI) denoting the registration domain out of which the address is assigned, and finally a Domain Specific Part (DSP) defining the address within that registration domain's address space.

In a second stage, the association connection request from the calling application to the target application is imbedded within a network PDU bearing the target NSAP address. The OSI network-layer routing mechanisms determine how to route that NPDU based on that target NSAP address. Further details about OSI naming and addressing mechanisms may be found in References 27 and 28.

Given the above descriptions of naming and addressing mechanisms, it is now possible to discuss the four key features of the integrated design, noted (1) to (4) in Figure 8.

Unification of CPI aliases. This paragraph refers to CPI-level integration, noted (1) in the figure. As suggested earlier, the definition of common CPIs suggests that local aliases, such as symbolic_destination_names (SDNs) in the context of CPI-C, must be allowed to refer to OSI applications as well as SNA applications. Thus CPI-C side information tables are enhanced to allow use of OSI TPSU names as well as SNA TP names, and X.500-style hierarchical AE Titles as well as LU names.

Unification of application names. Given integration at the CPI alias level, one can envision integration at the next level down, the application-layer directory, noted (2) in the figure. As is apparent from the earlier descriptions of SNA and OSI naming mechanisms, the APPN DS and X.500 directory mechanisms are very different. First, the intrinsic natures of AE Titles and LU names are different. The former are names meant for humans to use, while the latter are more like addresses meant for systems to use. Then, the X.500 directory can be used to retrieve many more attributes than just network-level address information (location). Finally, each directory mechanism achieves its purpose through markedly different techniques (broadcast search in APPN vs navigational search in X.500).

The concept of navigational searches to retrieve multiple object attributes, based on human-oriented hierarchical object names, is becoming widely accepted. To support X.500, the natural direction for directory integration consists of providing similar application-layer naming support under SAA and extending it to include SNA objects as well as OSI objects.

A first implication of this statement is that all systems providing application access to the OSI and/or SNA protocol stacks must include at least a DUA function to be able to access the SAA application directory. If an X.500 Directory Information Base is available locally, the DUA can access it directly, otherwise it must use either the SAA directory protocols provided by the SNA stack or the OSI directory protocols provided by the OSI component to access the nearest DSA. Similarly a DSA must be able to interact with peer DSAs using either the SNA or OSI protocol stacks, as represented in Figure 9.

A second implication is that X.500-style names and directory support must be provided not only to

OSI but also to SNA entities such as CPs LUs, and many others. The difference, which determines the switching of a CPI primitive to the SNA or the OSI stack, as discussed earlier, becomes apparent only at the address level. If looking up an AE Title in the application directory yields a PSAP address, the corresponding application is an OSI application (i.e., uses at least the upper part of the OSI stack), whereas if an AE Title yields an LU name, the application is an SNA application (uses at least the upper part of the SNA stack). This observation confirms that, from an OSI perspective, LU names really are addresses, in spite of what they are called. Two successive mechanisms, DS and TRS, are needed to derive an RSCV from an LU name, rather than one to determine an OSI route from an OSI NSAP address simply because OSI addresses include enough location information to compute routes, while SNA LU names are still location-independent and require a network search to obtain location information prior to computing the route.

In some cases, the directory search for an AE Title may yield both an OSI PSAP address and an SNA LU name at the same time. This would specifically indicate that the corresponding application is capable of using either protocol stack, as in the third scenario described at the beginning of this paper. In this case, the LU name and the PSAP address that the application bears simultaneously are in fact related. To simplify system management and internal packet routing, the P-selector of the PSAP address is defined by design to be equal to the LU name itself. This underlines the fact that the two addresses really denote the same application entity. It is in fact this addressing convention that ties together the LU 6.2 instance providing SNA services with the proper PSAP of the OSI component providing OSI services to the application. It also allows the OSI component in the integrated system to route an incoming PDU for that application to the proper PSAP, and on to the LU 6.2 instance of the same name, toward the right application.

Unification of network addresses. Moving down from the application directory level to the network routing level, the next issue to be addressed is the unification of network-level addresses. In the first two scenarios described earlier, SNA internetworking using OSI subnetworks, and OSI internetworking using SNA subnetworks, SNA applications may be attached to OSI networks and OSI applications to SNA networks, which raises the questions of mapping SNA internetwork addresses to OSI subnetwork addresses, and OSI internetwork addresses to SNA subnetwork addresses. The latter question, mapping OSI addresses to SNA subnetwork addresses, is discussed in the next section. The former question, mapping SNA addresses to OSI subnetwork addresses, is solved in existing products such as NPSI[9] for the specific case of X.25 networks. However, a more generic solution was developed as part of the present project and is outlined in this section.

Figure 9 Integrated SAA application directory

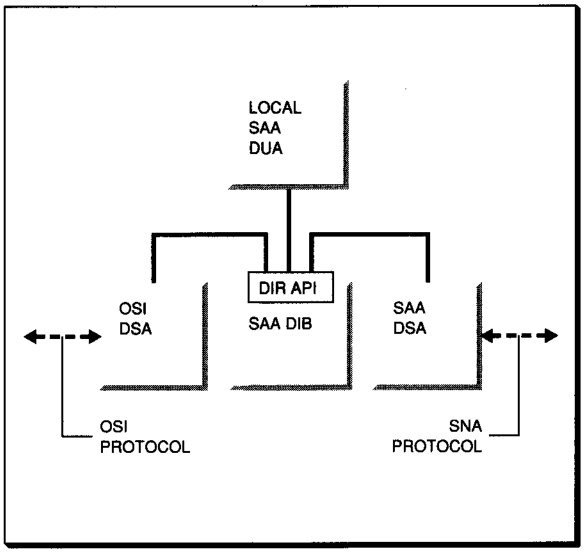
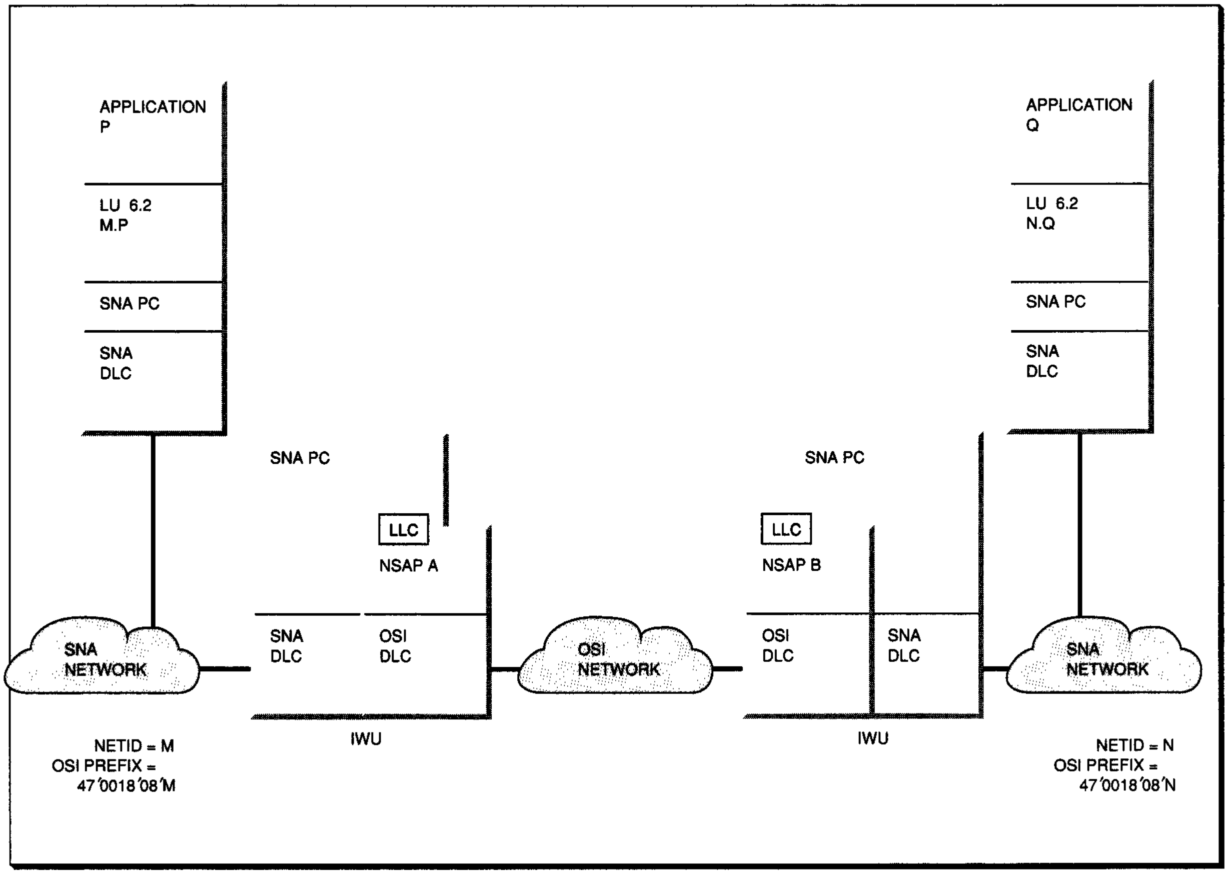


Figure 10 is used as a basis for this discussion. In this simple example based on the first scenario seen earlier, an application located at an LU named M.P in an SNA network on the left side of the figure wants to communicate with a peer application located at an LU named N.Q in another SNA network on the right, where the two SNA networks are interconnected through some intermediate OSI network(s). For this purpose, application P may look up the AE Title of Q in the integrated application directory to obtain the address (LU name) N.Q of the target application. Then, the calling LU M.P will send to the called LU N.Q an SNA BIND PDU bearing the destination name N.Q. For its transit through the intermediate OSI subnetwork, the BIND (and all subsequent session traffic) must, however, be encapsulated

Figure 10 SNA internetworking over OSI subnetwork



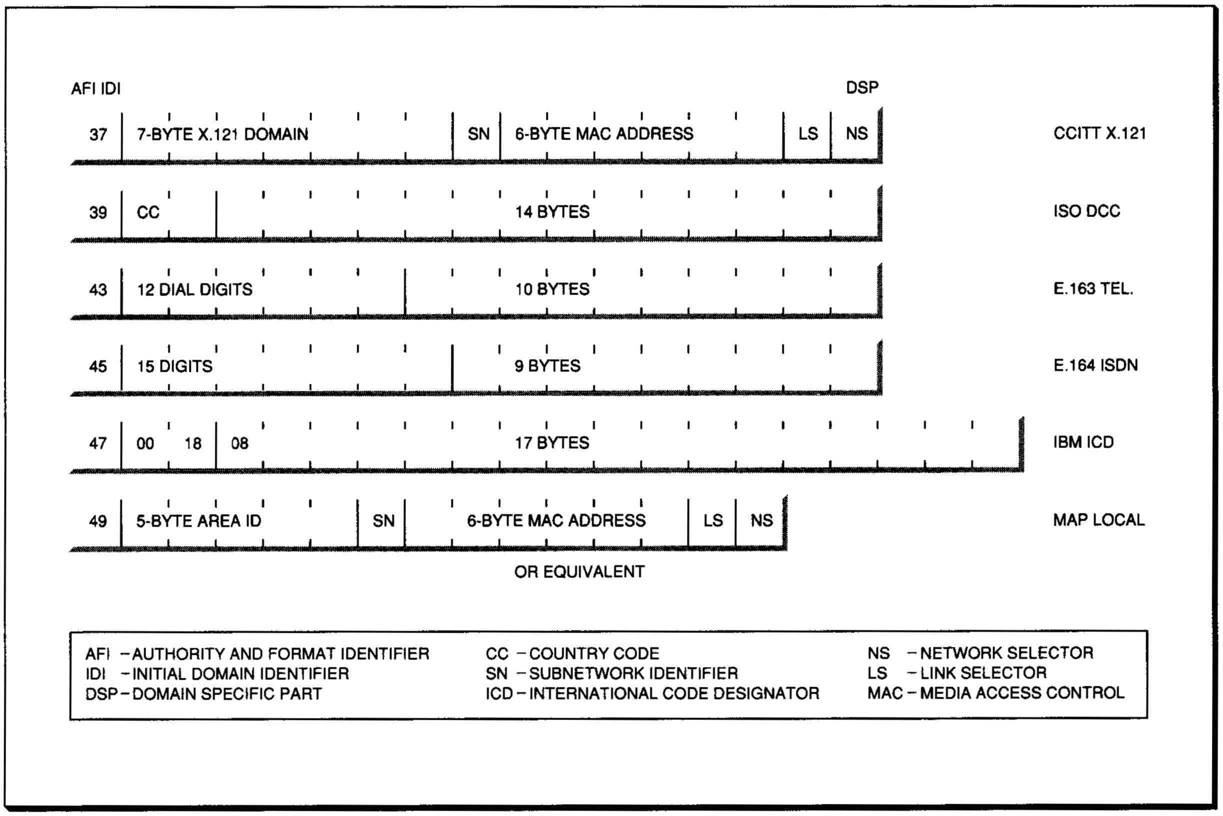
within the OSI subnetwork protocols, and PDUs at this level must be routed between the two IWUs with OSI NSAP addresses A and B. Clearly, neither an SNA RSCV nor the LU name N.Q as it is, is of any use for the routing mechanisms within the OSI network.

A solution was designed to make SNA internetwork addresses such as N.Q meaningful within OSI networks. This was done in two steps: in a first step, the target SNA address is converted into an OSI internetwork address; in the second step, the resulting OSI internetwork address is used to derive the OSI subnetwork addresses of suitable IWUs on a path to the target, using applicable OSI internetwork routing mechanisms.

1. The first step is achieved by integrating the SNA address space into the OSI internetwork address space. Through the direct syntactic mapping suggested by (3) in Figure 8, any SNA LU name corresponds to a valid OSI NSAP address. Figure 11 lists some examples of standard NSAP address formats.[28, 29] In any of these formats, NSAP addresses include an AFI, IDI, and DSP. In order to fold the SNA address space into the OSI NSAP address space, IBM has taken two initiatives:

   a. It has obtained from ISO (the International Organization for Standardization) a designated code point under one of the standardized NSAP formats to identify the IBM address space (actually more than just the SNA address space) within the OSI address space. This is code 0018 within the ISO ICD (International Code Designator) format 47. IBM has further allocated the first of the DSP bytes available within that format to a type field for separating the different address

Figure 11 Examples of standard NSAP address formats

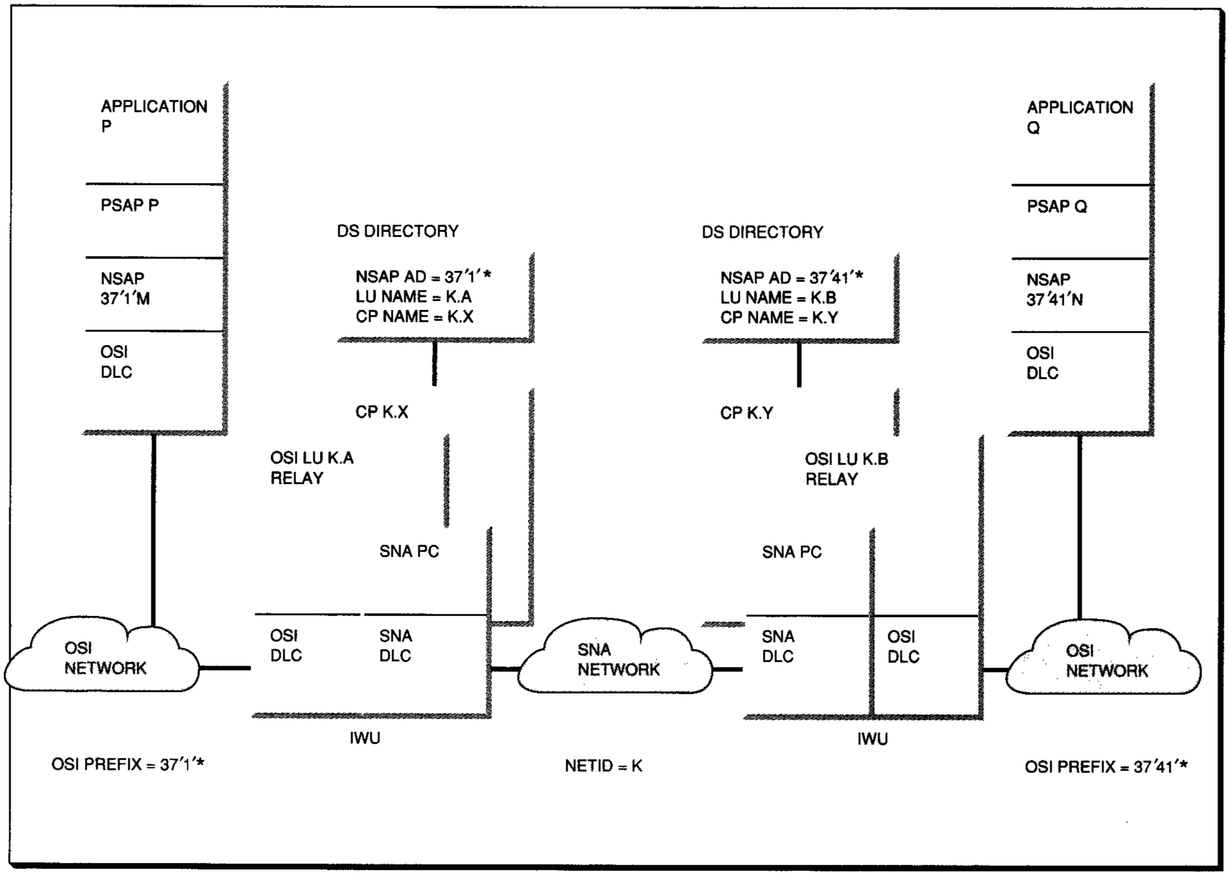


spaces under its control: type code 08 denotes the SNA address space. This leaves the rest of the DSP field available to store the Netid and LU name parts of the SNA address. (At the time of this writing, the maximum DSP length is being extended to 17 bytes so it can contain an SNA address.)

b. IBM has put in place a Registration Authority for SNA Netids from which users can get their own Netid values that are guaranteed to be unique on a worldwide basis. Users further administer LU names within their own Netid space. With these two mechanisms, any SNA LU name can be mapped syntactically into a valid OSI internetwork address.

2. For the second step, deriving addresses for IWUs on the path to the target, OSI internetworking mechanisms are used. The OSI Interdomain Routing Protocol (IDRP) is still under development. The present description is thus a very superficial account of its operation, only for illustrative purposes. IWUs between different routing domains, such as SNA and OSI, use so-called NSAP address prefixes to advertise on each of their sides the routing domains reachable on the other side. Thus in the example of Figure 10, the SNA routing domains are advertised to the OSI routing domain(s) by NSAP address prefixes such as 47'0018'08'N'*, where * stands for any LU name. IWUs regularly exchange such NSAP prefixes among themselves so each IWU learns what the others can reach. Thus, the IWU with NSAP address A on the left learns that the SNA NSAPs 47'0018'08'N'* are reachable over the IWU with NSAP address B on the right, which is all it needs to route the BIND through that IWU. As explained in the next section, the IDRP prefixes are even stored in the SNA network directories so that SNA LUs

Figure 12 OSI internetworking over SNA subnetwork



can discover what remote SNA addresses are reachable through what IWUs.

Essentially the same mechanisms are invoked if one or both of the SNA networks are reduced to single nodes so that SNA LUs M.P and/or N.Q are attached to the OSI network directly instead of indirectly through an IWU. In this case the source and destination SNA nodes on the OSI network recognize the source and destination LU names as being local instead of being in a distant node across some SNA network.

Integration of internetwork routing. This section now focuses on the OSI-over-SNA internetworking scenario depicted in Figure 12, where OSI traffic must cross an SNA subnetwork. In this example, an application P located in the OSI network on the left of the figure wants to communicate with a peer application Q located in the OSI network on the right, where the two OSI networks are interconnected through some intermediate SNA network. For this purpose, application P may look up the AE Title of Q in the integrated application directory to obtain its PSAP address. That PSAP address includes some NSAP address at which the target PSAP resides, 37'41'N in this hypothetical example. Thus, the calling NSAP must send to the called one a PDU bearing the destination NSAP address 37'41'N. For its transit through the intermediate SNA subnetwork, this PDU (and any subsequent between the same two NSAPs) must, however, be encapsulated within the SNA subnetwork protocols, and PDUs at this level must be routed between the two OSI relay LUs bearing LU names K.A and K.B on the intermediate SNA network. Clearly, the source and destination NSAP addresses and any OSI routing mechanism are of no use here.

Even the syntactic address mapping described in the previous section is of no use here. It maps any SNA address into a valid OSI address, but this mapping does not work the other way around. An NSAP address such as 37'41'N plainly has no SNA equivalent. Thus another solution is required.

The solution adopted to meet this requirement consists of a minor extension of the APPN DS mechanism, which in effect incorporates OSI NSAP addresses into the LU name space understood by APPN DS. The original DS mechanism maps the names of an SNA resource (e.g., an LU name) into the CP name of the network node closest to that resource. The extension designed to support OSI internetworking consists of allowing IWUs to store in their DS additional entries binding the NSAP address prefixes that they learn about through IDRP to their own OSI LU name and CP name. In effect, the extension amounts to storing in the APPN DS the routing information derived from IDRP. The resulting additional bindings are represented by the DS table entries managed by the CPs of the two IWUs in Figure 12, and by the box called DS' (4) in Figure 8. They allow an OSI LU such as K.A having to send a PDU to a foreign OSI address such as 37'41'N, to discover that doing so requires BINDing an LU 6.2 session with the relay function of its peer OSI LU K.B to ensure proper routing of interdomain traffic destined to an NSAP address in the set described by 37'41*. It is interesting to observe that the very same bindings may be used for internetwork routing by the Network-Layer Relays discussed here, as well as by higher-layer gateways, if desired. For instance, Transport-Layer gateways that would concatenate OSI Transport connections to SNA sessions could use the very same IDRP-based DS extension for OSI internetwork routing.

Until the OSI IDRP is standardized, the above design works fine if foreign NSAP address prefixes are simply manually defined to the IWUs. However, once a suitable standard is in place, IWUs will learn about reachable NSAP address prefixes dynamically and thus will be able to update the corresponding DS entries dynamically. In fact, the presently proposed IDRP mechanism will distribute not just reachability information but also cost, delay, error rates, quality of service, and other metric information for comparing alternate interdomain routes. Using this information, it will be possible to advertise in the APPN DS and subsequently discover only selected routes with given characteristics.

As in the SNA-over-OSI scenario, the same mechanisms may be invoked if any of the applications resided in a system directly attached to the intermediate subnetwork as opposed to being remotely attached through an IWU. A directly attached system would recognize its own NSAP address and it would use the PSAP addresses to route traffic internally to the proper application.

This concludes the description of the design unifying the CPI aliases, application names, network-layer addresses, and internetwork routing mechanisms of SNA and OSI, which was a key aspect of the integration study.

Summary and outlook

Because of existing investments in proprietary technologies such as SNA and NetBIOS, and simultaneous requirements for open architectures such as TCP/IP and OSI, IBM's future networking products will indifferently attach to and interoperate with multiple heterogeneous networks. This requires that the reference model for IBM networking systems be enhanced and opened to incorporate other protocol stacks besides SNA.

A research project described in this paper has defined guidelines and laid groundwork for this enhancement, pursuing four specific technical objectives: rich connectivity, full interoperation, sharing, and homogeneity. These were defined in a previous section on summary of technical requirements.

The work has been carried out in the specific context of SNA and OSI networks. Its key results are the definition of an enhanced structure for network-attached systems that will allow a flexible coexistence of multiple protocol stacks within the same system, the sharing of link, management, and control facilities among them, the interoperation between their network-layer protocols, and the unification of their CPIs. An essential factor to such structural integration is the unification of directory, naming, addressing, and routing mechanisms described in the last section.

Detailed work is still needed to fully realize the proposed integration and achieve its intended objectives. First, the general concepts and underlying structures must be extended to encompass other architectures beyond SNA and OSI. Second, more work is required to allow the interoperation of these other architectures among them and with SNA and OSI. Third, the same form of unification that was applied to the transaction processing interface must be extended to other application services. Finally, work is still going on to define the details of the common network management architecture, and integrate SNA and other resources into the X.500 directory environment.

Acknowledgments

The research discussed in this paper is so vast in scope that it could not have been performed without substantial contributions from many people in different organizations.

We are particularly indebted to Ellis Miller and Jim Gray of the Architecture and Telecommunications (A&T) Department at the IBM laboratory in Raleigh, North Carolina, and to Jerry Mouton of the OSI/Communications Subsystem design team at the IBM laboratory in Palo Alto, California. All three provided considerable encouragement, advice, and support throughout the project.

We are also much indebted to several people who, through their own work, made specific technical contributions that helped define and achieve the overall objectives for the SNA-OSI integration. These are, in particular: Andy Schwartz of the OSI/Communication Subsystem design team in Palo Alto and Andy Citron of the A&T Department in Raleigh for their substantial contribution to the CPI-C unification; Mike Gering and his team of A&T, for their work on common network management; Diane Pozefsky and her colleagues of A&T, for their on-going work on defining the LU 6.2 A and B stacks and open internetworking mechanisms; George Deaton and his people in A&T, and Serge Martel and his team in CSA, La Gaude, for their design of the Lower Layer Architecture enabling link sharing.

We thank John Hunter, director of A&T, Jack Sanders and Matt Hess of A&T, Liba Svobodova of the Zurich laboratory, and more product representatives from around the world than we can remember and list here, for their enthusiastic support and valued criticisms throughout the effort. Finally, we thank Liba Svobodova, Ellis Miller, Rudy Cypser, and Jim Gray for their extensive and insightful comments on various drafts of this paper.

Cited references and notes

1. Within this paper, the generic term *system* is used without qualification in the OSI sense of an end system (ES) or an intermediate system (IS), meaning a computer attached to some network either as provider or user of the network service. In other network architectures, such as APPN, the term *node* is used instead, as in end node or network node. The system structures discussed in this paper refer to the hardware and/or software structures found inside such "systems."
2. R. J. Sundstrom, J. B. Staton III, G. D. Schultz, M. L. Hess, G. A. Deaton, Jr., L. J. Cole, and R. M. Amy, "SNA: Current Requirements and Direction," *IBM Systems Journal* **26**, No. 1, 13–36 (1987).
3. J. H. Benjamin, M. L. Hess, R. A. Weingarten, and W. R. Wheeler, "Interconnecting SNA Networks," *IBM Systems Journal* **22**, No. 4, 344–366 (1983).
4. P. E. Green, R. J. Chappuis, J. D. Fisher, P. S. Frosch, and C. E. Wood, "A Perspective on Advanced Peer-to-Peer Networking," *IBM Systems Journal* **26**, No. 4, 414–428 (1987).
5. R. A. Sultan, P. Kermani, G. A. Grover, T. P. Barzilai, and A. E. Baratz, "Implementing System/36 Advanced Peer-to-Peer Networking," *IBM Systems Journal* **26**, No. 4, 429–452 (1987).
6. *IBM Systems Journal* **27**, No. 3 (1988, whole issue).
7. *NetBIOS Application Development Guide,* SG8X-2270-00, IBM Corporation (July 1987); available through IBM branch offices.
8. *TCP/IP Tutorial and Technical Overview,* GG24-3376-01, IBM Corporation (July 1990); available through IBM branch offices.
9. *NPSI (X.25 NCP Packet Switching Interface) General Information,* Version 3, GC30-3469-02, IBM Corporation (December 1989); available through IBM branch offices.
10. *Information Processing Systems—Data Communications—X.25 Packet Level Protocol for Data Terminal Equipment,* ISO 8208, International Organization for Standardization, Geneva (1990).
11. *Information Processing Systems—Data Communications—Protocol for Providing the Connection-less-Mode Network Service,* ISO 8473, International Organization for Standardization, Geneva (1988).
12. *X.25-SNA Interconnection (XI),* G511-1087-00, IBM Corporation (January 1988); available through IBM branch offices.
13. J. H. Rutledge, "OSI and SNA: A Perspective," *Journal of Telecommunication Networks*, 13–27 (1982).
14. P. François and A. Potocki, "Some Methods for Providing OSI Transport in SNA," *IBM Journal of Research and Development* **27** No. 5, 452–463 (September 1983).
15. H. Wakahama, et al., "Application of OSI Protocols as Intermediary for DCNA-SNA Network Interconnection," *Proceedings, 3rd International Conference on In-*

troduction of OSI Standards, Cambridge, UK, 312–344 (September 1985).

16. K. K. Sy, M. O. Shiobara, M. Yamaguchi, Y. Kobayashi, S. Shukuya, and T. Tomatsu, "OSI-SNA Interconnections," *IBM Systems Journal* **26**, No. 2, 157–173 (1987).
17. S. Zatti and P. Janson, "Interconnecting Heterogeneous Networks to OSI with a Global Naming Scheme and Gateway Address Mapping," *Proceedings, International Zurich Seminar* (March 1988). Also IBM Research Report, RZ-1651 (August 1987).
18. S. Zatti and P. Janson, "Inter-Network Naming, Addressing, and Directory Systems: Towards a Global OSI Context," *Computer Networks and ISDN Systems* **15**, 269–283 (1988).
19. *Information Technology—Open Systems Interconnection—Common Management Information Protocol Specification,* Recommendation X.cmip|ISO/IEC 9596, International Organization for Standardization, Geneva (December 1989).
20. *Information Processing Systems—Open Systems Interconnection—Basic Reference Model,* ISO 7498, International Organization for Standardization, Geneva (October 1984).
21. *CCITT Blue Book Volume VIII, Fascicle VIII.8, Data Communications Networks—Directory,* Recommendations X.500-X.521|ISO/IEC DIS 9594, International Organization for Standardization, Geneva (1989).
22. *Systems Application Architecture, Common Programming Interface, Communications Reference,* SC26-4399-02, IBM Corporation (August 1990); available through IBM branch offices.
23. *Information Processing Systems—Local Area Networks—Part 2: Logical Link Control,* ISO 8802-2, International Organization for Standardization, Geneva (December 1989).
24. *SNA Transaction Programmer's Reference Manual for LU Type 6.2,* GC30-3084-03, IBM Corporation (September 1990); available through IBM branch offices.
25. *Information Processing Systems—Open Systems Interconnection—Distributed Transaction Processing—Parts 1 (Model), 2 (Service definition), 3 (Protocol specification),* ISO/IEC DIS 10026-1, 2, 3, International Organization for Standardization, Geneva (March 1990).
26. *End System to Intermediate System Routing Exchange Protocol for Use in Conjunction with ISO 8473,* ISO 9542, International Organization for Standardization, Geneva (1987).
27. *Information Processing Systems—Basic Reference Model—Part 3: Naming and Addressing,* ISO 7498/Add. 3, International Organization for Standardization, Geneva (1984).
28. *Encoding of NSAP Addresses in Network Layer Protocols,* ISO/TC97/SC6/WG2 N10, International Organization for Standardization, Geneva (June 1985).
29. *Information Processing Systems—Data Communications—Addendum to the Network Service Definition Covering Network Layer Addressing,* ISO 8348/Add. 2, International Organization for Standardization, Geneva (1988).

Accepted for publication January 23, 1992.

Philippe Janson *IBM Research Division, Zurich Research Laboratory, CH-8803 Rueschlikon, Switzerland.* Dr. Janson received the B.S. in electrical engineering from the University of Brussels, Belgium, in 1972, and the M.S., E.E., and Ph.D. in computer science from the Massachusetts Institute of Technology in 1974, 1975, and 1976, respectively. From 1974 to 1976 he was a research assistant at the M.I.T. Laboratory for Computer Science, working on the internal structure of Multics. Since 1977 he has been with the IBM Zurich Research Laboratory in Switzerland, where he has worked on high-speed packet switches, the IBM Token Ring, LAN servers, and gateways. Since 1987 he has managed communication system architecture projects dealing with the integration of heterogeneous network architectures, and more recently network security. During 1986–87, he was on assignment with the IBM development laboratory in Austin, Texas, working on LAN gateways for OS/2*. He is a visiting professor with the Department of Engineering of the University of Brussels, where he has been teaching the operating system course since 1976. He has also taught various courses on operating systems, LANs, OSI, and network security at the IBM International Education Center in Brussels and at the Swiss chapter of the ACM-IEEE (SI). His research interests include operating systems, distributed systems, and computer communication. He is the author of many papers and a few patents in these fields, as well as a textbook on operating systems. He is a member of the IEEE Computer Society, the ACM, SIGOPS and SIGCOMM, and Sigma Xi. He received a Harkness Fellowship in 1972 and invention and technical awards from IBM since then.

Refik Molva *IBM Research Division, Zurich Research Laboratory, CH-8803 Rueschlikon, Switzerland.* Dr. Molva is a research staff member at the IBM Zurich Research Laboratory in Switzerland. He joined IBM in 1987 after receiving a Ph.D. in computer science from the Paul Sabatier University at Toulouse, France. He worked in the areas of heterogeneous network interconnection, protocol specification, and validation using formal definition techniques. He is currently working on a network security project where he has been actively involved with the design of highly secure authentication and key distribution protocols.

Stefano Zatti *IBM Research Division, Zurich Research Laboratory, CH-8803 Rueschlikon, Switzerland.* Mr. Zatti joined IBM as a research staff member in 1985 and has since been operating in the area of application services for communications, with particular interest in naming, addressing, and directory services. In 1989–1990, he was on international assignment with the IBM West Coast programming laboratory in Palo Alto, California, where he was responsible for the architecture of IBM directory services for the OSI product line. Mr. Zatti received the *Laurea* in mathematics (cum laude) from the University of Pavia, Italy, in 1980, and the M.S degree in computer science from the University of California, Berkeley, in 1985. His research interests include operating systems, distributed systems, network security, and computer communication. He is a member of the IEEE Computer Society and a Senior Member of IEEE.

Reprint Order No. G321-5476.