P. François
A. Potocki

Some Methods for Providing OSI Transport in SNA

This paper is made up of three parts: first, a review of open systems interconnection (OSI) objectives, followed by a comparison/contrasting of OSI and systems network architecture (SNA) objectives; second, a description of techniques for providing communication between programs residing in systems based on different architectures; and third, the possible application of these techniques to the four lower OSI layers, demonstrating some ways which could be used to provide connectivity between SNA end users and other heterogeneous system end users through OSI protocols.

Part 1: SNA and OSI objectives reviewed

The design of systems network architecture (SNA) started in the early 1970s and SNA was first announced publicly in September 1974 [1]. Its objective was to provide a system definition as a foundation for a unified approach to data and information communications. It defined a uniform set of functions to be supported and implemented by communication products and thus provided flexibility for the development of new products which could take advantage of advanced technologies. At the same time, SNA created a firm and stable basis for the development and evolution of user applications to address new opportunities in the rapidly evolving field of distributed information processing.

The definition of the open systems interconnection (OSI) scope and standards started in the mid-1970s, and a set of standards covering different layers of the OSI model should become available by the mid-1980s. The environment and the objective to be addressed by OSI have been defined as follows [2]:

> "In the concept of OSI, a system is a set of one or more computers, associated software, peripherals, terminals, human operators, physical processes, information transfer means, etc., that forms an autonomous whole capable of information processing and/or information transfer."
>
> "OSI is concerned with the exchange of information between open systems (and not with the internal functioning of each individual open system)."
>
> "OSI is concerned not only with the transfer of information between systems, i.e., transmission, but also with their capability to work together to achieve a common (distributed) task. In other words, OSI is concerned with cooperation between systems, which is implied by the expression 'systems interconnection.' "
>
> "The objective of OSI is to define a set of standards to enable open systems cooperation. A system which obeys applicable OSI standards in its cooperation with other systems is termed an open system."

Thus the objectives of SNA and OSI can be summarized as follows:

- SNA defines the internal structure of a system. It is concerned with the cooperation of products to form a coherent communication system. It also defines the functional responsibilities of each network component within a system, the way information is exchanged between products, and additionally, the internal control structure that provides the management of system resources and system services.
- OSI defines the external communication capability of a system to make it an open system, i.e., capable of cooperating with other open systems according to OSI standards. The normal OSI applicability is in cases where the cooperating open systems each have a different internal architecture.

Note from the above objectives that the two architectures tend to complement each other. In this paper we have retained the OSI definition of *system*, i.e., a complete homogeneous set of computers, nodes, terminals, etc. [3]. We next analyze the functional similarities and differences between the two architectures in more detail. (See also [4].)

Functional similarities

What SNA and OSI have in common is that they both provide functions and protocols to support data communication between remote application entities. These functions and protocols are the same in nature, whether the remote entities are SNA end users and the protocols are internal to an SNA system, or whether these entities are OSI applications and the corresponding ends extend between different systems.

The OSI functions and protocols are structured to conform to the OSI model, which includes seven functional layers. SNA defines equivalent functions and has a similar layered structure. The overall correspondence between the OSI and SNA functional layers is represented in Table 1.

Table 1 OSI and SNA layers contrasted.

OSI layer	*SNA layer*
Application	End user
Presentation	LU services manager Function management and presentation
Session	Data flow control
Transport	Transmission control
Network	Path control
Data link	Data link control
Physical	Physical control

The table does not imply an exact one-to-one mapping between OSI and SNA layers. In fact, the one-to-one comparison is fully applicable only to the two bottom layers and maybe to the two top layers. As to the three middle OSI layers (network, transport, and session), their composite functions are addressed by the composite functions of the three middle SNA layers (path control, transmission control, and data flow control). Note that, as we shall see later, and as can be expected from OSI objectives, only the total set of functions provided by a system (not how they are grouped internally into components) has an impact on their working together.

Functional differences

The different SNA and OSI objectives and scope we have just discussed result in different SNA and OSI functional contents. These differences mainly affect two areas, the internal network protocols and the network services and communication network management.

- *Internal network protocols*

As part of a total system solution, SNA must include the definition of mechanisms and their corresponding protocols, which provide functions such as routing within the SNA network or global flow control (which prevents network congestion), etc. All these functions and protocols are part of the SNA path control layer.

Equivalent functions and protocols do not have to be part of OSI, which is concerned with systems interconnection and not with internal systems operation. For example, for a packet-switched network in an OSI environment, the X.25 packet-level protocol defined at the packet network interface corresponds to the OSI network layer subset. This protocol permits the interconnection of packet network users regardless of the internal mechanisms and protocols specific to the packet network.

- *Communication system services and management versus OSI management*

In addition to the internal protocols directly used by messages flowing between end users through a network, controlled operation of a system (within the scope of SNA) requires a whole set of support functions, such as directory services (to provide the current location of the other party), configuration management (to organize the activation sequence of the various boxes, links, and programs which make up the system), network management (to monitor errors and performance data gathering, statistics computation, correlation and presentation of reports), and maintenance (to identify a failing element in the system).

Although some of these functions (e.g., directory) will likely be defined in OSI, the OSI management is different in scope from SNA, as it is concerned with "virtual resources" and "abstract objects" without considering how these "resources" are actually implemented within a real system. This is, as stated earlier, in line with the general OSI scope.

SNA systems in the OSI environment

The current environment of information communications and processing is characterized by a large, and increasing, number and diversity of systems and products built according to different architectures and specifications. Many manufacturers have unified their product lines into distributed system solutions by adopting and implementing their own system architectures (as IBM did with SNA). Currently, there are more than ten thousand SNA systems installed and operational. These support a great diversity of distributed data and information processing user applications, and they are also growing in size and in functional scope. Moreover, at

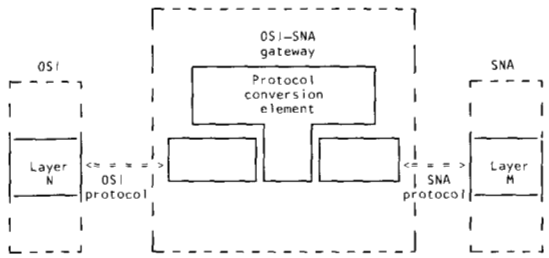

Figure 1 Protocol conversion between an OSI and an SNA layer, where each entity is operating as if it were communicating with its normal partner.

the present installation rate, many more SNA systems will be operational before the OSI standards become an implementable reality.

The advent of OSI standards will bring a new dimension to the current environment by providing universally agreed upon means of permitting communication and cooperation between (or among) *heterogeneous* systems and products. The existing systems will progressively implement OSI capability in response to user application needs. But the existence of OSI standards should not, and will not, slow down the increasing number and diversity of heterogeneous systems and products. In fact, in response to users' requirements, the systems built on heterogeneous architectures will grow in number and in size and they will even provide new functions that are not supported by OSI standards.

The resulting OSI environment will therefore be characterized by a large, and possibly increasing, diversity of heterogeneous open systems; *heterogeneous* because they are built on different architectures; and *open* because they are capable of cooperating with other systems by implementing the OSI protocols.

This trend will tend towards the situation presently existing in the worldwide telephone network in four respects. First, the overall network structure consists of private networks, i.e., private branch exchanges (PBX) and networks, built on different architectures and technologies, and interconnected through public networks which provide communication support between any two devices so connected. Second, a partitioned network management structure exists in which each private network has its own network services (e.g., its own directory) and its own management structure, independent of the equivalent services and structure provided by public networks and other private networks. Third, there are a wide range of private network sizes, ranging from small PBXs serving a few extensions to large private "distributed" networks composed of many interconnected PBXs. And fourth, a wide range of functions and features is provided internally by private networks in response to special user requirements.

Many SNA users have, and will have, a strong requirement to "open" their SNA systems and to make them capable of cooperating with non-SNA systems and products using OSI protocols. This comes about when some application programs and/or terminal operators which reside in and are part of SNA systems are required to communicate and cooperate with applications or operators residing in non-SNA systems or operating in non-SNA products. These users will need a solution which will allow them to become active members of the OSI environment at a minimal extra expense and disruption to their operations, and which will also protect their current SNA investments and preserve their freedom to grow their SNA systems further in size and in functional scope.

Possible approaches to open SNA systems

With these requirements in mind, let us investigate the possible approaches to *opening* SNA systems to the OSI environment. The problem we face is that SNA systems (or some elements within the SNA systems) as *seen from the outside* must be capable of "speaking" and "understanding" the OSI protocols. It is important to realize that the definition of "open" is relative to the external behavior of the system. The OSI reference model [2] states that "any real system which behaves externally as an abstract open system can be considered as an open system, i.e., in conformance with OSI standards."

With this definition there are two possible approaches to "opening" an SNA system:

- Provide protocol conversion between the SNA and OSI protocols (both ways) at the boundary between the two domains.
- Implement the OSI protocols in particular SNA products (in addition to the SNA protocols).

Protocol conversion is particularly attractive in the case of SNA systems in which the protocol conversion program can be implemented only once in a gateway shared by many products. It has the advantage of leaving unchanged the internals of SNA systems; i.e., it protects the existing investments and leaves freedom to provide new and optimized functions. But, as we see in the following sections, protocol conversion is applicable only when the OSI and SNA protocols have the same semantics, i.e., when the services are equivalent.

The implementation of OSI protocols within SNA systems obviously requires extra development work and additional

code in all the products which have to communicate with the OSI domain. As a consequence protocol implementation should be used only when protocol conversion is not applicable or when the protocol conversion code cannot be shared by many users.

In the following sections we first review protocol conversion techniques and their limitations, and then discuss in some detail how these techniques could be used to "open" SNA systems to the OSI environment.

Part 2: Protocol conversion principles

Let us first consider the simple case of protocol conversion between equivalent layers of two different architectures: OSI and SNA, as shown in Fig. 1.

Protocol conversion consists of expressing the semantics carried by one protocol in the message formats and protocol syntax of the other. In other words, it consists of converting the necessary message formats and protocol syntax to preserve the protocol semantics.

As in the case of language translation between individuals speaking different languages, protocol conversion can provide communication between entities belonging to different architectures, with each entity "speaking" and "understanding" only its native protocol. In the case illustrated in Fig. 1, the OSI and SNA entities can communicate while each entity is operating as if it were communicating with its normal partner (i.e., OSI with another OSI, and SNA with another SNA). The net result is that the SNA element in question behaves externally as an OSI element and is therefore in conformance with OSI standards.

Thus, protocol conversion appears to be the ideal approach for opening an existing architecture to the OSI world. It does the job at the expense of a conversion program placed at the boundary of the two architectures without requiring modifications to the existing architectural elements. Unfortunately, protocol conversion has an inherent limitation: It is applicable only when the same semantics exist in the protocols of both architectures. We take up this limitation in more detail later.

Protocol conversion, as shown in Fig. 1, can take place in an OSI/SNA gateway composed of three main functional parts:

- The OSI element which handles and checks the OSI protocols.
- The SNA element which handles and checks the SNA protocols.
- The protocol conversion element which has access to both the OSI and SNA protocols and which does the actual protocol conversion.

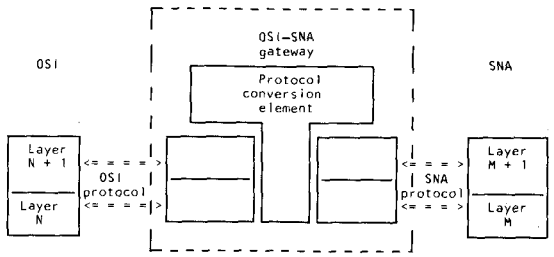

Figure 2 Protocol conversion extended to two or more layers, where the protocols of each layer in OSI are converted from one or more layers in SNA.

We mentioned earlier that the function distribution between layers is not necessarily the same in OSI and in SNA. In particular, in the middle layers, the composite function of two or three layers must be considered when comparing the two architectures. This fact does not prevent the possibility of protocol conversion. As shown in Fig. 2, protocol conversion can be extended to protocols pertaining to two (or more) layers, with the possibility of converting the protocol of level N of one architecture into the protocol of level $M + 1$ of the other architecture.

Another important point is that where the protocols can be converted between elements of two different architectures, the service interfaces of these elements are equivalent. In other words, the service interface on top of the SNA layer $M + 1$ in Fig. 2 is functionally equivalent to the corresponding OSI interface; i.e., it provides the same services. Let us also remember that OSI defines two standards for each layer: the service definition standard and the protocol specification standard. The *service definition* standard is conceptual. It defines the primitive actions and events (and the corresponding parameters) at the service interface on top of a layer. This definition is abstract in the sense that it describes the services offered by a layer (the functions it provides) and not how they are realized. This abstract definition of OSI service interfaces is sufficient to specify the higher-level protocols without constraining unnecessarily its design and implementation. With this definition, an SNA service interface which provides an OSI-defined service can be considered as an OSI interface. The immediate consequence is that, when necessary, the higher-level OSI protocols can be directly implemented on top of the SNA interface, as illustrated in Fig. 3.

Inherent limitation of protocol conversion

As mentioned in the previous section, the protocol conversion technique can be used effectively to support communication between two entities belonging to different architectures if, and only if, the protocols pertaining to these entities carry the same semantics. In other words, the protocol conversion

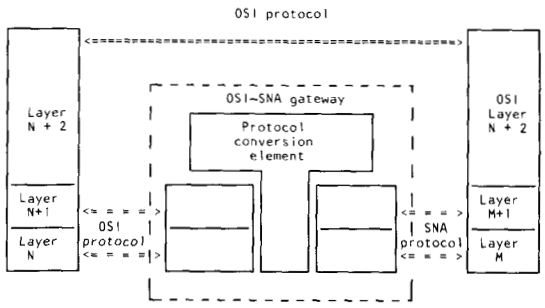

Figure 3 Here an SNA service interface provides an OSI-defined service and therefore can be considered an OSI interface on which other OSI layers can be built.

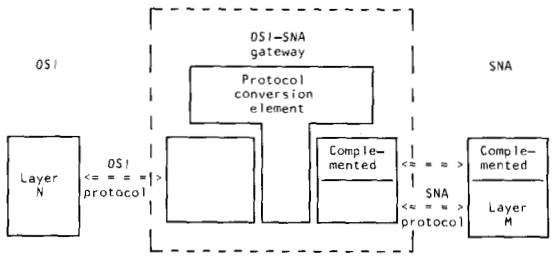

Figure 4 Mixed protocol conversion, where protocol complementation is used to provide services not equivalently available in SNA. Note that since OSI is the standard, services available in SNA but not in OSI cannot be provided.

technique is applicable only between entities which are functionally compatible; i.e., each can provide all the functions invoked and expected by the other. Thus, when two protocols belonging to different architectures support different sets of functions, protocol conversion can be applied effectively only to the functional subset common to both protocols.

When two protocols are not "convertible," the only possible solution for communicating with the other entity is actually to implement the other entity's protocol.

When two protocol sets differ only in some respects, a mixed solution can be used: Protocol conversion can be applied to the common subset and this can be complemented by implementing the missing functions on the corresponding side. This type of solution is illustrated in Fig. 4. In the case of SNA and OSI, this complementing can be done only on the SNA side by adding missing OSI functions, since the problem is to conform to the standard OSI protocols. In many cases, the missing functions can be added on top of the SNA elements, as shown in Fig. 4. In other cases, this could require modifications to the current SNA implementation. The protocols corresponding to the implemented functions can be designed to facilitate the conversion. In most cases these will be OSI protocols "enveloped" by SNA headers.

Performance considerations

An argument which is sometimes used against the protocol conversion solution is the potential performance degradation due to the extra processing required in the gateway (which increases the total execution path length) compared to a direct implementation of the same "foreign" protocol.

There is no doubt that the performance factor has to be considered when deciding between the two types of solution. Performance must also be taken into account in the design of the gateway. In particular, protocol conversion must be done at the most effective level of data message granularity: high enough to get the proper understanding of headers; low enough to avoid the extra delay which could be introduced by reassembling segmented data messages in the gateway.

However, with the current state of processor technology, the performance hit of a well-designed protocol conversion can be made acceptable and at minimal cost. In some cases, this approach is a more than reasonable price to pay for avoiding the development cost of direct implementation of "foreign" protocols in each product.

Part 3: A protocol conversion technique applied to the SNA-OSI environment

As stated earlier, the opening of an SNA system consists of providing the capability for some selected application programs and/or terminal operators which are part of the SNA system to communicate and cooperate, using OSI protocols, with applications and/or operators residing in non-SNA systems or operating in non-SNA products. To illustrate our discussion, we use the practical example of an application program operating on a VTAM (virtual telecommunication access method) [5] which has to communicate with an OSI application residing outside the SNA system. This example is shown in Fig. 5 and is subsequently described in more detail.

Note that the specific implementation examples given are provided only to illustrate the described techniques and to show their feasibility; the reader should not infer from the examples that this will be implemented in any product.

• *Typical configuration*

The SNA system represented in Fig. 5 as a simple host and a single 37X5/NCP (network control program) can be, in fact, composed of any number of hosts, communication controllers, and peripheral nodes; and the VTAM host under consideration can reside anywhere within the SNA system.

The OSI system represented in Fig. 5 as a single box can also be either a standalone product or part of a complex system. The only point important to our discussion is that the OSI system appears as a set of OSI protocols (as seen at the boundary of the SNA system) corresponding to the four upper layers of the OSI model. The three lower layers of the OSI model are represented in our example by the X.25 interface protocols [6]. These are not formally recognized as OSI standard protocols, but they are certainly representative of the future standard. The application program which has to communicate with an OSI application is written on top of VTAM.

As shown by these comments, the example places practically no restrictions on the nature (or on the configuration) of the OSI system, and in that respect it can be used to illustrate the general case of the OSI/SNA interconnection. However, to limit our discussion, we have to make some specific assumptions about the configuration of the SNA system. The configuration has been chosen as representative of the most typical user requirement and SNA implementation in the OSI environment. But these specific assumptions do not restrict the generality of our discussion. We show how the conclusions can be extended to other products and configurations.

- *Main aspects of OSI transport services*

When considering their function, we can split the seven layers of the OSI reference model into two general functional groups:

- The four lower layers (physical, data link, network, and transport), which together support the end-to-end data transport service (i.e., provide the end-to-end communication connectivity).
- The three upper layers, which include the application itself, the presentation layer, and the session layer. The latter two provide the necessary means and functions for the communicating entities to organize their cooperation from the points of view of dialog scheduling and defining a common syntax for the information exchanged.

Those two groups of layers differ significantly. The lower layers, as a whole, are functionally independent from the upper layers that they serve. On the other hand, the session and presentation layers perform functions which are directly used and driven according to the application requirements. As a consequence, while the transport service interface is always the same and is independent of the various transport classes [7], the session service interface varies according to the session options. The more functions provided, the more complex the service interface gets. In other words, the lower layers provide *connectivity* between processes located in remote boxes; they define a common solution to problems

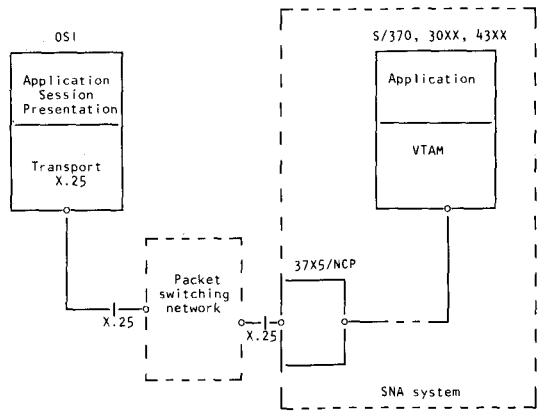

Figure 5 For purposes of discussion, the SNA system is lumped together into a single host with an IBM 37X5 network control program communicating through an X.25 interface to the OSI. VTAM is the virtual teleprocessing access method program product.

resulting from remoteness (line errors, transmission delays, bandwidth limitations, etc.). The upper layers define ways to achieve *cooperation* between processes, independent of where they reside.

Within this paper, only the four lower layers are taken into account; session services and presentation services would require a complementary study.

The OSI transport service functional definition is limited to the establishment and release of the transport connection, and to the sending and receiving of normal and expedited data over the transport connection. It is fully represented by the following conceptual service primitives: T-CONNECT, T-DISCONNECT, T-DATA, and T-EXPEDITED DATA (optional). T-CONNECT is a confirmed service, while the others are nonconfirmed services.

Transport service is further defined in terms of quality of service (QOS), which includes parameters relating to performance (e.g., throughput, transmission delay, etc.), and accuracy/reliability (e.g., failure probability, residual error rate, etc.). For the time being, neither values nor classes of values are specified as QOS parameters.

To summarize, transport service provides the end-to-end connectivity, while isolating the service user from all the complexities due to system configuration, transmission error detection, and recovery, etc. This transport service is sufficient and adequate to support all types of current application requirements.

In the most general case, the composite functions of the four lower OSI layers are necessary to provide the transport

service. The OSI reference model defines a multiplexing structure in which each of the four layers participates with its own protocols, granularity, and end-to-end characteristics, thereby supporting the OSI transport service over any combination of system configurations and transmission media. The specific role and the characteristics of each layer which are of importance to our discussion can be summarized as follows:

- The *physical layer* provides the mechanical, electrical, and procedural characteristics of the physical connection and of the encoding of the data bit stream. The physical protocol can be defined either locally (at the data terminal equipment/data circuit-terminating equipment interface), or as a modulation specification (on the line).
- The *data link* protocol provides the necessary functions to transfer data between adjacent (separated by a single connection) nodes. Its specific role is the detection and recovery of errors occurring in the physical transmission facility. The data link protocol represents the lowest multiplexing granularity and the shortest span (its "ends" are located in adjacent nodes) in the OSI reference model. Typically, an intermediate network node drives several data link protocols asynchronously. The node also drives the internal mechanisms which route data units between data link entities as defined by a higher-level protocol (e.g., network).
- The *network* protocol provides the necessary functions to transfer data across a network. Typically, a network connection spans (and relies on) the services of more than one data link and network node. A part of the network protocol function is to detect and possibly to correct errors occurring in network nodes. The network protocol corresponds to the intermediate multiplexing granularity (several network connections can be multiplexed on a single data link connection) and to the intermediate span in the structure defined by the OSI reference model (its "ends" are located in nodes interfacing the network directly).
- The *transport* protocol provides the necessary functions to transfer data all the way between the communicating data entities (the application and the corresponding session and presentation entities). Typically, a transport connection spans and relies on the services of one (or more) network connection(s) and several underlying data link connections. Some of the transport protocol classes are defined to correct errors not recovered by the network layer. The transport protocol corresponds to the finest granularity (several transport connections can be multiplexed on a network connection, but on top of the transport connection and at a given time, there is a one-to-one correspondence between the transport, session, and presentation active connections) and to the maximum end-to-end span (its "ends" are the entities which provide the transport service to the communicating applications).

Providing data transport

The very first (and mandatory) condition for "opening" an SNA system is to provide the data transfer capability (i.e., connectivity) between selected products residing in the SNA system and the OSI domain. In OSI terms, this condition is expressed by saying that SNA products have to provide the OSI transport service interface supported by protocols of the four bottom layers of the OSI reference model (as seen from outside the SNA system).

Let us reiterate that this condition can be fulfilled by implementing and converting the OSI protocols at the boundary between the OSI and SNA domains without impacting the corresponding SNA protocols. In other words, the OSI transport service provided in SNA products can be supported by SNA protocols within the SNA system. This is important since it resolves the first basic part of the SNA/OSI problem at a minimal cost and without impacting existing and future SNA investments.

The following is a brief description of the principle and an overview of three solutions.

The first solution would perform OSI transport to SNA transmission control (TC) session protocol conversion at the border of the SNA network. The two others are based on OSI network to SNA TC session protocol conversion already provided by NPSI/PCNE, the NCP packet-switching interface program product [5] which includes the protocol converter for non-SNA equipment; the OSI transport is then either implemented directly on top of the OSI network interface made available in SNA hosts, or the OSI protocol is converted into SNA session TC protocols.

It should be mentioned, of course, that several other approaches could be considered (both how to and where to provide the function) and that this paper reviews only the feasibility of some of them.

- *Method 1—based on conversion of the OSI transport protocol*

Let us consider the typical configuration shown in Fig. 5. In this case, providing the OSI data transport consists of implementing an effective OSI/SNA gateway at the boundary of the SNA system in such a way as to use the existing VTAM application program interface (API) primitives to support the OSI transport service (i.e., providing the OSI transport service interface on top of VTAM). This can be done without impact to the SNA resources up to and including VTAM. Moreover, the same OSI/SNA gateway can be shared to provide the OSI transport service to any SNA product capable of establishing a simple SNA session with a logical unit (LU) representing the gateway within the SNA system.

The layout of the corresponding gateway is shown in Fig. 6. The functional elements within the figure are very much simplified and are limited to the elements required for supporting a single transport connection. Not shown here are the multiplexing structures which exist both on the OSI and SNA sides. In reality, on the OSI side, a single link access protocol (LAP) element supports multiple packet level protocol (PLP) elements (each corresponding to a virtual circuit) and each of these PLP elements can support a multiplicity of transport elements. On the SNA side, the single synchronous data link control (SDLC) and path control (PC) elements are used to represent the SNA structure, including virtual routes (VRs), explicit routes (ERs), and transmission groups (TGs).

Another dimension not represented in this figure is the fact that a single NCP can support and route data on a number of separate X.25 interfaces (i.e., a multiplicity of X.21 and LAP elements) and a multiplicity of separate VR, ER, and TG elements on the SNA side. As a result, two OSI transport connections which use the same virtual circuit can be routed, after protocol conversion, on different VRs; and similarly, SNA TC protocols using the same VR can be routed to different X.25 interfaces.

Note also that most of the elements shown in Fig. 6 can be shared by SNA/SNA traffic. This is typically the case with SNA PC and SDLC elements and could also be the case with X.25 Level 1 and Level 2 elements (i.e., X.21 and LAP). Only the elements which are part of the OSI/SNA gateway and the PLP elements supporting the transport connection are completely devoted to the OSI/SNA traffic.

Another point which we want to clarify is that all protocols, up to and including PLP and PC, operate asynchronously from one another, and that there is no protocol conversion at these levels. In other words, all these protocols have their "ends" in the same NCP. Note also that only the elements which participate directly in protocol conversion, i.e., the OSI transport and the SNA TC and protocol converter elements, are shown as part of the OSI/SNA gateway. These also happen to be the only elements which are not implemented in NCP.

The OSI entities with which transport connections are to be established must be known to the SNA system and must be described as SNA resources. This can be done as follows. OSI entities supported by permanent virtual circuits (PVCs) can be described as secondary logical units (SLUs) connected to the NCP by a leased link (the PVC can be described as the leased link). They will be assigned SNA network names corresponding to their transport service access point identifiers (TSAP-IDs) with SNA network addresses. The OSI entities using switched virtual circuits

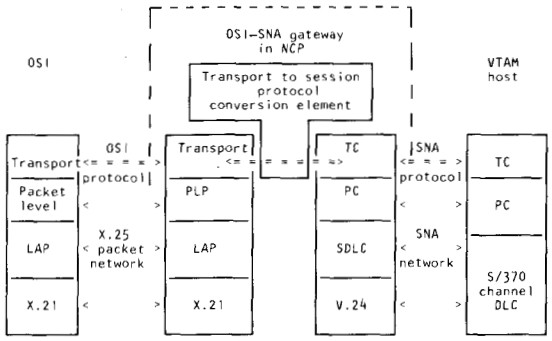

Figure 6 Method 1 proposes an OSI-SNA gateway in the network control program (NCP) whereby existing VTAM (virtual telecommunications access method) application program primitives might be used to provide OSI transport services. Note: LAP = link access protocol, PC = path control, PLP = packet-level protocol, SDLC = synchronous data link control, TC = transmission control.

(SVCs) can be defined as switched resources (SLUs); the SVCs can be described as switched links and they can be supported by normal SNA dial-up procedures. They can be assigned SNA network names corresponding to their TSAP-IDs, and SNA network address assignment can be done during the dial in/out procedure.

To the NCP, the gateway will appear as pools of SLUs, with each pool corresponding to a link, and each link representing a virtual circuit. The assignment of SLUs to virtual circuits can be done during system generation for the permanent virtual circuits (possibly modifiable by dynamic reconfiguration) and at dial in/out time [i.e., request network address assignment (RNAA) procedure] for switched virtual circuits.

With this in mind, let us briefly review the OSI/SNA gateway operation during the transport connection establishment, data transfer, and transport connection release phases.

Note that for simplicity a single-domain SNA network is assumed in all cases following, but it should be clear that the gateway operation remains unchanged for multi-domain networks.

Transport connection establishment phase When transport connection is initiated from the OSI side, the gateway operates as illustrated by the sequence diagram in Fig. 7(a). The following comments apply to Fig. 7(a); they assume that the virtual circuit is already established:

1. Upon reception of the request connection transport protocol data unit (TPDU) initiated in the OSI system, the gateway selects an SLU from the appropriate pool and

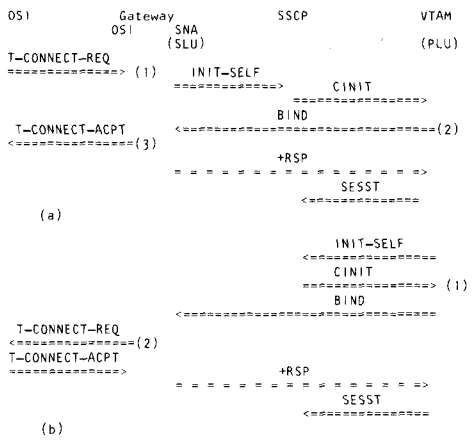

Figure 7 Sequence diagrams for establishing transport connection: (a) from the OSI side; (b) from the SNA side. See discussion in text.

sues an INIT-SELF request unit (RU) to its system services control program (SSCP). The called TSAP-ID included in the connection request (CR) TPDU is used as the destination logical unit (DLU) name in INIT-SELF, pointing to the destination address in the SNA system. If the calling TSAP-ID is included in the CR TPDU, it is put into the user data field in INIT-SELF and transmitted (in CINIT) to the destination partner.

2. This is followed by the normal SNA CINIT-BIND sequence. If necessary, a VR will be activated between VTAM and NCP prior to sending BIND.
3. The gateway receives BIND and issues the connection confirm TPDU to the OSI system and the +RSP to BIND. Both the OSI transport connection and the SNA session are now established and the gateway ties them together by establishing and maintaining the relationship between the OSI source/destination references and the SNA origin/destination networks addresses.

In case the transport connection is initiated on the SNA side, the operation is as shown by the sequence diagram in Fig. 7(b) and is explained as follows:

1. The DLU name in the INIT (which points to the OSI entity with which the transport connection has to be established) is resolved by the SSCP to the address of one of the SLUs which represents the remote destination to the SSCP.
2. Upon receipt of BIND, the gateway sends the CR TPDU on the virtual circuit corresponding to the SLU (which is supposed to be established already). The called TSAP-ID to be used in the CR TPDU is transmitted to the gateway in the SLU name field of BIND. When the connection accept (CA) TPDU is received, the gateway issues +RSP to BIND and ties together the OSI transport connection and the SNA session by establishing and maintaining the relationship between the OSI source/destination references and the SNA origin/destination address pair.

Note that these sequences strictly describe the conversion of OSI transport into SNA session (TC) protocols. Should the OSI session also be taken into account, one might easily foresee a complementary conversion: OSI transport and session would probably correspond to the SNA session protocols (transmission control plus data flow control). In that case one would consider T-CONNECT + S-CONNECT conversion into the SNA BIND.

Data transfer phase We next consider the OSI transport protocols, Class 2 and Class 3, as defined by OSI [8], which are used as examples here. Classes 0 and 1 have no multiplexing capability, and Class 4 is slightly more complex but could be handled in the same way as Class 3. Class 2 allows multiplexing of several transport connections in a single network connection. It provides transport connections with or without individual flow control but does not include any provisions for error detection nor for recovery.

In the data transfer phase, the gateway performs two main functions. First, it converts OSI data TPDUs into SNA path information units (PIUs) (and vice versa) by swapping the destination reference with the origin and destination fields (OAF/DAF) and by converting the EOT (end of TPDU) parameter into the mapping field parameter in the SNA header. No relationship is maintained between the OSI and SNA sequence numbers. The gateway might also have to readjust the TPDU or PIU sizes (if different maximum sizes were specified during connection establishment). Second, when individual flow control is specified for the connection, the gateway performs a functional bridging between the OSI mechanism and the SNA session pacing. There is no direct conversion of protocol elements, but the gateway monitors both flow-control protocols and tries to keep both receiving-end windows permanently open, while at the same time providing enough buffer space to prevent losing data when one of the receiving ends keeps its window closed. When establishing the flow-control parameters at connection establishment time, the fact that the OSI and SNA mechanisms are different and that they control different objects (TPDUs in OSI and RUs in SNA) must be taken into account.

The Class 3 ISO transport protocol can be used when the quality of service of the underlying OSI network layer is not adequate for the transport service user. It provides the same functions as Class 2 (with explicit flow control) plus the ability to recover after a failure signaled by the network layer without involving the user of the transport service. This error recovery function is based on the sending transport entity keeping a copy of all transmitted data TPDUs until it

receives positive acknowledgment, which allows copies to be released. It may also receive a reject (RJ) command, inviting it to retransmit all data TPDUs from the point in the sequence indicated in the RJ command.

The reliability of SNA products provides an adequate quality of transport service, and as a result SNA does not currently include a recovery mechanism equivalent to the Class 3 transport protocol.

In the SNA/OSI case, the Class 3 transport protocol recovery capability can be used when necessary to improve the overall quality of transport service. The Class 3 protocol can be implemented in the OSI/SNA gateway shown in Fig. 6 with its recovery functions being effective only in the OSI part of the connection (i.e., the packet network and the OSI system in the typical configuration shown in Fig. 5). For the remaining functions, the gateway performs protocol conversion as described above for Class 2, and it provides the end-to-end OSI transport service. The resulting quality of service is comparable to that provided by the SNA system.

Transport connection release phase The principle of the OSI/SNA gateway operation for transport connection release initiated by one of the transport service users is illustrated by the sequence diagrams in Fig. 8.

The transport connection might also have to be released in case of a failure of the underlying services. In case of a failure of the X.25 virtual circuit, the gateway has to send an UNBIND on all the SNA sessions corresponding to the transport connections carried by a virtual circuit. An SNA virtual route failure is identified via UNBIND (session outage notification) to all impacted sessions and the gateway converts each UNBIND into disconnect request TPDUs on the corresponding transport connections.

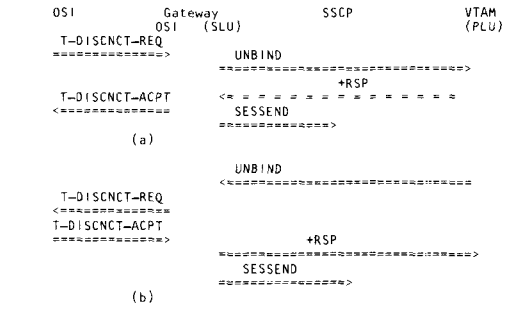

Figure 8 Sequence diagrams for releasing transport connection: (a) from the OSI side; (b) from the SNA side.

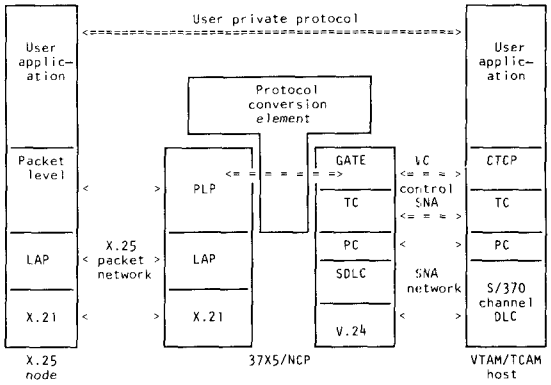

Figure 9 Program product NPSI/PCNE for protocol conversion through GATE (general access to X.25 transport extension) provides an OSI-SNA gateway in the network control program. Note: CTCP = communication and transmission control program, LAP = link access protocol, PC = path control, PLP = packet-level protocol, SDLC = synchronous data link control, TC = transmission control, VC = virtual circuit.

• *Methods 2 and 3—based on X.25 NPSI/PCNE*
The first release of the X.25 NPSI (X.25 NCP packet-switching interface) program product [5] includes the PCNE feature (protocol converter for non-SNA equipment), which provides a data transfer function, i.e., connectivity, between an SNA (VTAM or TCAM) host and a non-SNA node implementing the X.25 virtual circuit protocols. The PCNE performs protocol conversion between the X.25 packet level and a limited subset of the SNA session protocols.

With PCNE, most of the X.25 virtual circuit functions are accessible to users of SNA VTAM or TCAM access methods. PCNE is further complemented (in Release 2 of the X.25 NPSI) by the general access to X.25 transport extension (GATE) which provides the capability for a user application program residing in an SNA host to assume the complete task of monitoring an X.25 virtual circuit. The corresponding part of the user application program is called CTCP (communication and transmission control program). The principle of operation of PCNE and GATE is represented in Fig. 9 in such a way as to show the analogies and differences with the OSI/SNA gateway of Fig. 6

PCNE has been developed in response to SNA systems users' requirements to provide communication between SNA hosts and non-SNA equipment. As shown in Fig. 9, PCNE provides data transfer (i.e., connectivity) and is used to support user applications communicating via user-defined private protocols. In that sense PCNE can be considered as an OSI predecessor and a solution to users' OSI requirements. It is based on the X.25 virtual circuit protocol, which for the time being is the only international standard pervasive and stable enough to be used for this purpose.

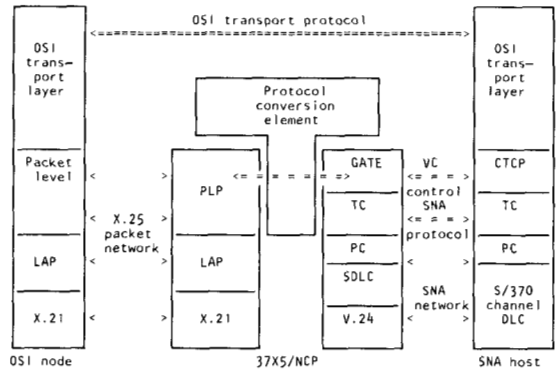

Figure 10 Method 2 proposes the implementation of the OSI transport layer on top of the network services interface in an SNA host. Note: CTCP = communication and transmission control program, GATE = general access to X.25 transport extension, LAP = link access protocol, PC = path control, PLP = packet-level protocol, SDLC = synchronous data link control, TC = transmission control, VC = virtual circuit.

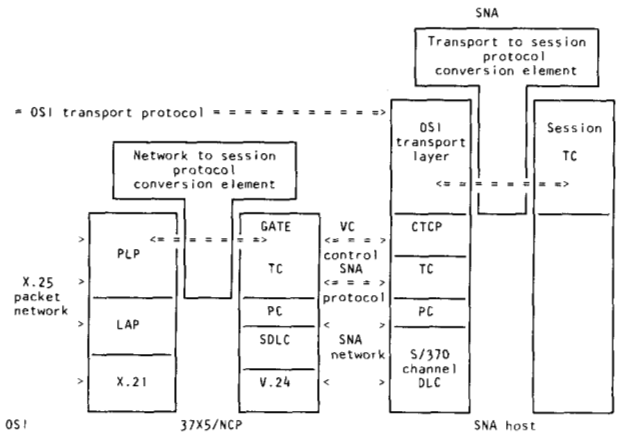

Figure 11 Method 3 is similar to the one shown in Fig. 6, except that the transport protocol conversion element now resides in the VTAM host instead of in the gateway at the border of the SNA network. Note: CTCP = communication and transmission control program, GATE = general access to X.25 transport extension, LAP = link access protocol, PC = path control, PLP = packet-level protocol, SDLC = synchronous data link control, TC = transmission control, VC = virtual circuit.

The OSI network service interface created by PCNE in SNA hosts could be used further in two ways to provide at least the transport layer. Method 2 consists of implementing directly the OSI transport layer on top of the network service interface. This is illustrated in Fig. 10. If implemented, it would provide an OSI transport service interface in SNA hosts which could either be used, as is, by an application above it, or it could be used as a base for building the upper OSI layers.

Method 3 also requires OSI transport implementation on top of the network service interface but from a different perspective. Instead of having the transport service interface used directly by an application or by an OSI session placed above it, the transport protocols can be converted into SNA session TC protocols. This is illustrated in Fig. 11 and is very similar to the case shown in Fig. 6; but the protocol conversion element now resides in the VTAM host instead of being located in a gateway right at the border of the SNA network. All flow sequences described for Method 1 (transport establishment, release, and data transfer) apply unchanged to this third approach. Note that once it is converted into an SNA session (TC protocols), this connection can be extended anywhere in the SNA network. Contrasted with Method 2, this transport gateway provides a solution potentially common to several products and to many users.

Note that these two methods force all OSI transport connections multiplexed in a virtual circuit to be carried in a single SNA session to the same host. The flexibility of a PCNE-based solution could be enhanced by providing in the NCP the capability of dispatching OSI transport connections multiplexed in a virtual circuit (or SNA session) to different SNA sessions (or virtual circuits).

Conclusion

Communication between programs residing in SNA and non-SNA systems through an OSI interface is feasible. An OSI interface is already provided in SNA hosts at the network level by the NPSI program product. By using the protocol conversion technique, an OSI transport service interface could be provided as well. Once a given OSI interface is available in a system, any higher OSI layer can be easily implemented on top of it. As far as the session and presentation layers are concerned, further study will be required once the current standardization efforts are completed.

References and note

1. *Systems Network Architecture—General Information*, Order No. GA27-3102; *Systems Network Architecture—Format and Protocol Reference Manual: Architectural Logic*, Order No. SC30-3112, available through IBM branch offices.
2. *ISO DIS 7498 Data Processing—Open Systems Interconnection, Basic Reference Model*, ISO/TC97/SC16, available through the American National Standards Institute, 1430 Broadway, New York, NY 10018 (Rev., January 1983).
3. A number of interesting studies are reported in the *Proceedings of COMPCON Fall 1982*, IEEE Catalog No. 82CH1796-2, IEEE, 445 Hoes Lane, Piscataway, NJ 08854.
4. J. H. Rutledge, *OSI and SNA: A Perspective*, Order No. GG22-9225, available through IBM branch offices.
5. *SNA Concepts and Products*, Order No. GC30-3072, available through IBM branch offices.
6. *IBM X.25 NCP Packet Switching Interface, General Information*, Order No. GC30-3080, Program No. 5668-981, available through IBM branch offices.

7. *ISO Transport Service Definition*, DP 8072, ISO/TC97/SC16/WG6, available through the American National Standards Institute, 1430 Broadway, New York, NY 10018 (January 1983).
8. *ISO Transport Protocol Definition*, DP 8073, ISO/TC97/SC16/WG6, available through the American National Standards Institute, 1430 Broadway, New York, NY 10018 (January 1983).

Received January 7, 1983; revised April 28, 1983

Philippe François *IBM France, Centre d'Etudes et Recherches, 06610 La Gaude, France.* Mr. François is manager of Communication Systems Architecture at the Communication Products Division laboratory in La Gaude. He joined IBM in 1963 at La Gaude, where he initially worked on simulators. From 1965 to 1974 he was involved in the IBM 2750 and 3750 line switching projects. Since 1976, he has worked on communication controllers and on SNA. Mr. François is an applied mathematics graduate of the University of Paris.

Arthur Potocki *IBM France, Centre d'Etudes et Recherches, 06610 La Gaude, France.* Mr. Potocki is a Senior Architect at the Communication Products Division laboratory in La Gaude. He joined IBM in 1956 at the IBM France laboratory located in Paris, where he worked on the early design of circuits and modems. He transferred to La Gaude as technology manager in 1960. From 1962 to 1964 he was assigned to IBM Europe headquarters in White Plains, New York. When he returned to La Gaude in 1964, he became deeply involved in new line switching developments. Since 1973, his main interest has switched from voice to data networks and he has become involved in the early definition of SNA, SNA multi-hosts, and the X.25/SNA interconnection. Mr. Potocki is an electrical engineering graduate of the Conservatoire National des Arts et Métiers in Paris.