OSI-SNA interconnections

by K. K. Sy
M. O. Shiobara
M. Yamaguchi
Y. Kobayashi
S. Shukuya
T. Tomatsu

As Open Systems Interconnection (OSI) becomes an international standard, it is gaining support in both industry and government agencies. One of the major applications of OSI is to act as an intermediary between heterogeneous networks. This paper discusses a scheme for interconnecting a Systems Network Architecture (SNA) network with OSI. This scheme is based on a joint study between IBM Japan and Nippon Telegraph and Telephone Corporation conducted during 1984. Fundamental relationships between OSI session and transport layers and SNA Logical Unit type 6.2 are explored. An OSI-SNA gateway structure is examined, and data units, address translation, and exception handling are discussed.

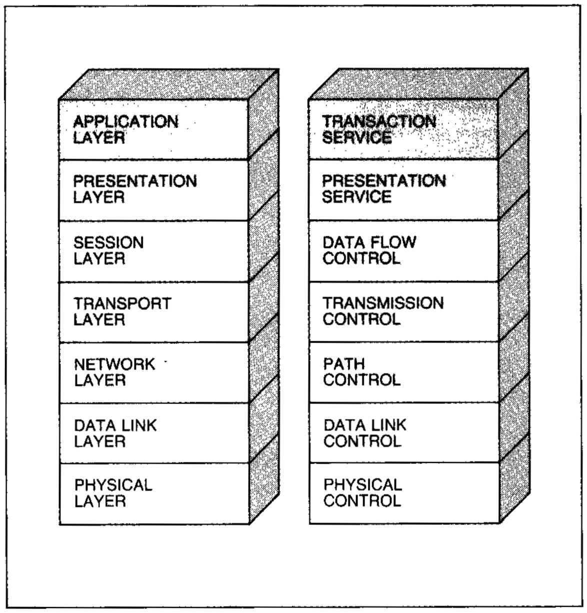
In the last decade, computer manufacturers have experienced a tremendous growth in the development of communications facilities. With advances in technology and greater customer demands, various computer manufacturers have developed their own communications networks, each with its own architecture and protocols. The result is that networks and systems from different manufacturers cannot easily communicate with one another. This difficulty in communicating is a major stumbling block to growth and flexibility for many users. It is apparent that a standard network architecture is needed that will allow various computer networks and systems to communicate with one another. In 1977 the International Organization for Standardization (ISO) Technical Committee 97 (TC97) established a new subcommittee (SC16, currently SC21) on Open Systems Interconnection (OSI). In the spring of 1983, the OSI Basic Reference Model[1] became an international standard. However, while computer manufacturers worked with ISO in developing these standards, they continued to develop and enhance their own communications architectures and networks. For example, IBM had over 10 000 Systems Network Architecture (SNA) networks installed by the time the OSI Basic Reference Model became an international standard.

One application of OSI is to use it as an intermediate protocol to interconnect various heterogeneous networks. Several research and development efforts have focused on the interconnection of dissimilar networks. The SNATCH Gateway between SNA and TRANSDATA,[2] the studies of François and Potocki on connecting and interfacing SNA with OSI standards,[3] and Green's study on general protocol conversion[4] are some examples of the growing work done in interconnecting heterogeneous networks.

In 1984, IBM Japan and Nippon Telegraph and Telephone Corporation (NTT) conducted a feasibility study to interconnect an SNA network and NTT's Data Communication Network Architecture (DCNA) network through OSI. The study focused on protocol conversion through a gateway between OSI session[5,6] and transport[7,8] layers and the related SNA/DCNA protocols. The IBM/NTT work was a theoretical study and is not related to any IBM development plan.

This paper, which is based on the results of the IBM/NTT study, addresses the OSI-SNA interconnection. Functions of OSI and SNA are compared, mapping techniques and a gateway structure are introduced, address translation is explained, and exception handling is discussed.[9]

Figure 1 Layer structure of OSI Basic Reference Model and SNA



Overall framework

Francois and Potocki[3] identified two approaches to interconnect OSI and SNA:

- Convert the protocols of one system to the semantically equivalent protocols of another system.
- Implement one protocol in the products of another protocol. An example would be the implementation of OSI protocol in SNA products, or vice versa.

The first approach was the focus of the joint study and is the subject of this paper. Alternatives other than protocol conversion are not discussed in this paper. The following steps were taken to study OSI-SNA interconnection based on protocol conversion:

- Comparison of OSI and SNA functions and identification of the common subset of OSI and SNA functions
- Definition of the protocol conversion procedure between OSI and SNA
- Establishment of the address translation method between OSI and SNA
- Study of exception handling

OSI-SNA relations

OSI and SNA have similar layer structures, as shown in Figure 1. Although the layer structures are similar, they do not allow OSI application programs and SNA application programs to communicate without protocol conversion. This section identifies the functions which are common to OSI and SNA and then defines mapping between these architectures.

Functional comparisons. In SNA, in order for an end user to communicate with another end user, their respective logical units (LUs) must be connected in a mutual relationship called an LU-LU session. There are several LU types, differing in the specific SNA protocols and options supported. LU type 1 is for a session between an application program and a terminal. LU type 2 is for a session between an application program and a display using the IBM 3270 data stream. LU type 3 is for a session between an application program and a printer using the 3270 data stream. LU type 4 is for a session between an application program and a terminal or a session between two terminals. LU type 6 is for a session between two application programs.

In this paper, LU type 6.2 (LU 6.2), a version of LU type 6, was chosen for the following reasons:

- LU 6.2 offers peer-to-peer communication which is conceptually the same as that offered by OSI.
- LU 6.2 has a well-defined interface which makes mapping comparatively easier than with the other LU types.

Hereafter "SNA LU-LU session" means an "LU-LU session of SNA LU 6.2" unless otherwise specified.

OSI session connection and SNA LU 6.2 conversation. Examining the OSI session connection and SNA LU 6.2 conversation,[10] one will find that many functions are similar, although the terminologies are different. OSI session connection provides the means necessary for associated service users to organize and synchronize their dialogue and to manage their data exchange. In contrast, SNA LU 6.2 conversation provides a logical interface through which the transaction program can access the SNA network and its resources. It provides the structure for programs to communicate with one another in order to process a transaction.

Table 1 lists some of the functions common to OSI session connection and SNA LU 6.2 conversation.

OSI session protocol specifies a kernel functional unit and either half-duplex or full-duplex functional units as basic mandatory functions. The kernel includes a session connection establishment and release as well as data transfer, whereas the half-duplex functional unit includes the token management. The OSI functions listed in Table 1 satisfy the basic OSI session protocol mandatory requirements.

SNA LU 6.2 also specifies a series of conversation verbs of mandatory requirements. These mandatory verbs are ALLOCATE, DEALLOCATE, CONFIRM, CONFIRMED, GET_ATTRIBUTE, RECEIVE_AND_WAIT, REQUEST_TO_SEND, SEND_DATA, and SEND_ERROR.[10] The SNA LU 6.2 functions described in Table 1 satisfy all of the mandatory verbs.

Figure 2 schematizes the relationship between OSI session protocol and SNA LU 6.2 conversation. The shaded area represents the common functions, which consist of the following categories:

1. Common functions that are mandatory in both OSI session connection and SNA LU 6.2 conversation:

OSI Session Connection	SNA LU 6.2 Conversation
Kernel	ALLOCATE
Half-Duplex	DEALLOCATE
	SEND_DATA
	RECEIVE_AND_WAIT
	REQUEST_TO_SEND

   Although both duplex and half-duplex are allowed in the OSI session protocol, only half-duplex is chosen in this study since SNA LU 6.2 operates half-duplex only.

2. Common functions that are mandatory in OSI session connection but optional in SNA LU 6.2 conversation:

OSI Session Connection	SNA LU 6.2 Conversation
None	None

3. Common functions that are mandatory in SNA LU 6.2 conversation but optional in OSI session connection:

OSI Session Connection	SNA LU 6.2 Conversation
Major Synchronize	CONFIRM
Exception Report	CONFIRMED
	SEND_ERROR

4. Common functions that are optional in both the OSI session connection and SNA LU 6.2 conversation:

OSI Session Connection	SNA LU 6.2 Conversation
None	None

Figure 2 OSI Session-SNA LU 6.2 Conversation relationship

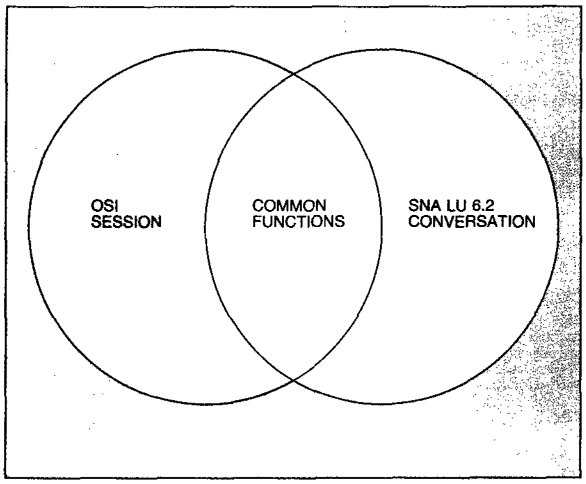


Table 1 Functions common to OSI Session Layer and SNA LU 6.2 Conversation

OSI Session	SNA LU 6.2 Conversation
Session connection establishment	Allocate a conversation
Data exchange	Send data to a transaction program Receive data from a transaction program
Token management (Half-duplex)	Request control of the flow between two programs Yield control of the flow between two programs
Exception report	Send error information to a transaction program
Synchronization	Synchronize the processing activities of the two programs
Session connection release	Deallocate a conversation

We also considered the mandatory functions that are unique to OSI or to SNA LU 6.2. Such functions are covered in the nonshaded area in Figure 2.

One category in this nonshaded area comprises the functions that are mandatory in OSI session connection but have no corresponding function in SNA LU

Table 2 Functions common to OSI Transport Layer and SNA LU-LU Session

OSI Transport Layer	SNA LU-LU Session
Transport connection establishment	LU-LU session establishment
Data exchange	Data transfer
Transport connection release	LU-LU session release

Table 3 OSI Session Layer—SNA LU 6.2 Conversation functional mapping

OSI Functions	SNA Functions
S-Connection Control	Conversation Control
• Establishment (from T-Connect initiator side)	• ALLOCATION (from both sides)
• Non-Negotiated Release (Normal/Abort)	• DEALLOCATION (Normal/ABEND)
• Functional Unit Negotiation	• Conversation Attribute Negotiation
• Relation to T-Connection – 1:1 Mapping – Reuse of T-Connection	• Relation to SNA Session – 1:1 Mapping – Reuse of SNA Session
Data Transfer on S-Connection	Data Transfer on Conversation
• Segmentation/Assembly – Data Unit Length	• Logical Record – Maximum RU Size
• Data Transfer – HDX Data Transfer	• Data Transfer – HDX Data Transfer
• Send/Receive Control – Tokens (Data and Major-Activity) – Token Assignment	• Send/Receive Control – Change Direction (CD) bit – Right-to-Send Control
• Session Synchronization – Major Sync	• Sync. Point Control – CONFIRM
• Exception Reporting (in Receive State only)	• Error Reporting (in Send/Receive State)

6.2 conversation. There are no items in this category, because all the mandatory functions that are required in OSI have corresponding functions in SNA LU 6.2.

Another category in this nonshaded area includes the functions that are mandatory in SNA LU 6.2 conversation but have no corresponding function in OSI session connection. There are two items in this category: (1) the SEND_ERROR verb in send state, and (2) conversation initiated from either partner.

For the first item, SNA LU 6.2 conversation allows a SEND_ERROR verb to be issued in both the send and receive states, whereas OSI session protocol only allows an EXCEPTION DATA function to be sent in the receive state.

For the second item, SNA LU 6.2 allows a conversation to be initiated from either partner, whereas OSI allows a session connection to be initiated only from the side that initiated the underlying transport connection.

In this study, the above items are restricted at the SNA side to achieve the interconnection to OSI, because these items are derived from the fundamental architectural difference between OSI and SNA LU 6.2. In other words, the SEND_ERROR verb in the send state and conversation initiation from the receiver of an LU-LU session are restricted to the SNA side.

OSI transport connection and SNA LU-LU session. Comparable functions can also be observed between the OSI transport connection and an SNA LU-LU session.

In OSI, the transport layer is concerned with creating a uniform transport service that is defined on an end-system-to-end-system basis with respect to characteristics such as error detection and recovery, multiplexing, addressing, and quality of service.

In SNA LU 6.2, an LU-LU session is a logical connection between two LUs that can be activated, tailored to provide various connection protocols, and deactivated, as requested. The session activation request and response determine connection options relating to such functions as the rate and concurrency of data exchange, the control of contention and error recovery, and the characteristics of the data stream.

Table 2 gives the functions common to the OSI transport protocol and SNA LU-LU session protocol.

Unlike the higher layer, the OSI transport connection and an SNA LU-LU session cover each other's mandatory functions. Therefore, no protocol restrictions are required for either OSI or SNA LU 6.2. These mandatory functions are connection establishment, data transfer, and connection release.

OSI-SNA functional mapping. We obtained the functional mapping of OSI and SNA LU 6.2 based on the functional comparison.[11]

Table 3 summarizes the functional mapping between OSI session connection and SNA LU 6.2 conversation,[12] and Table 4 summarizes the functional mapping between the OSI transport layer and the LU-LU session of SNA LU 6.2.

Data unit comparison. Aside from the functional relationship between OSI and SNA, an important area to be considered is the data unit relationship between the two.

In OSI, data units are expressed in terms of protocol data units (PDUs) and service data units (SDUs). Each layer has its own PDUs and SDUs.

In SNA, data units are expressed in terms of logical records, request/response units (RUs), basic information units (BIUs), and path information units (PIUs). The mapping between OSI and SNA data units can be shown as follows:

OSI	SNA LU 6.2
Session SDU (SSDU)	Logical record(s)
Session PDU (SPDU)	Request/Response Unit
Transport PDU (TPDU)	Basic Information Unit (BIU = RH + RU)
Network PDU (NPDU)	Path Information Unit (PIU = TH + BIU)

Table 4 OSI Transport Layer—SNA LU-LU Session functional mapping

OSI Functions	SNA Functions
T-Connection Control	SNA Session Control
• Establishment/Release	• Establishment/Release
• Parameter Negotiation	• Parameter Negotiation
• Multiplexing/Splitting to N-Connection	• Multiplexing onto VR
Data Transfer on T-Connection	Data Transfer on SNA Session
• Data Unit Concatenation	• (No Function)
• Data Transfer Control	• Data Flow Control
– Normal Data Transfer	– Normal Data Flow
– Expedited Data Transfer	– Expedited Data Flow
– Sequential Numbering	– Sequential Numbering
• Flow Control	• Pacing
• Error Detection	• (DLC Function)
• Error Correction	• (DLC Function)
• (No Function)	• Session Encryption

OSI-SNA gateway

A gateway that converts OSI and SNA LU 6.2 protocols can be designed on the basis of the functional mapping and data unit relationship between these network architectures. There are various alternatives to implement OSI-SNA gateways. This section introduces a chosen one, and the characteristics of its protocol conversion specifications are discussed.

Gateway structure. Conceptually, the chosen OSI-SNA gateway structure consists of three components:

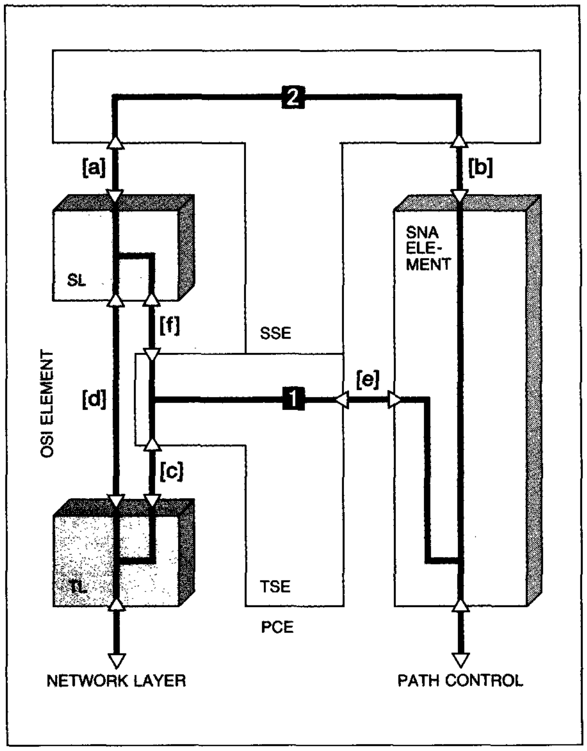
1. *OSI Element*—the portion of the OSI-SNA gateway that communicates with the partner OSI node. It only handles OSI protocol, and is divided into two subcomponents, the Session Layer Sub-Element (OSI-SL) and the Transport Layer Sub-Element (OSI-TL).
2. *SNA Element*—the portion of the OSI-SNA gateway that communicates with the partner SNA node. It only handles SNA LU 6.2 protocol.
3. *Protocol Conversion Element (PCE)*—the portion of the OSI-SNA gateway that converts OSI protocol to SNA LU 6.2 protocol and vice versa. It is divided into two subcomponents, the PCE Session Sub-Element (PCE-SSE) and the PCE Transport Sub-Element (PCE-TSE). PCE-SSE converts OSI session protocol to SNA LU 6.2 protocol and vice versa. PCE-TSE converts OSI transport protocol to SNA LU-LU session activation/deactivation protocol and vice versa.

A pictorial representation of the OSI-SNA gateway is shown in Figure 3. There are two control paths in the OSI-SNA gateway. The first path accomplishes the mapping of the OSI transport connection and the SNA LU-LU session with respect to connection establishment and release, using PCE-TSE as the focal point. (In Figure 3, note that the representation of the TSE between the transport element and the session element does not correspond to a function insertion between the OSI transport and session layers; rather, it corresponds to an interlock mechanism meant to ensure correct gateway handling of the protocols.) The second control path is used to accomplish the mapping of OSI session protocol and LU 6.2 conversation verbs using PCE-SSE as the focal point and as the data exchange between the two interconnected end nodes.

The path establishment and data exchange flows do not operate concurrently; therefore, the multiplicity of flows shown in Figure 3 is not meant to imply that at any given instant a flow is split and then recombined.

There are six boundaries defined in this OSI-SNA gateway. They are used for communication among

Figure 3 OSI-SNA gateway structure



various elements within the gateway. These boundaries are as follows:

a. PCE-SSE–OSI-SL. Used for communication between PCE-SSE and OSI-SL and is based on OSI session service primitives.[5]
b. PCE-SSE–SNA element. Used for communication between PCE-SSE and the SNA element and is based on the SNA LU 6.2 conversation verbs.[10]
c. PCE-TSE–OSI-TL. Used for communication between PCE-TSE and OSI-TL and is a subset of the OSI transport service primitives.[7]
d. OSI-SL–OSI-TL. Used for communication between OSI-SL and OSI-TL and is the remaining subset of OSI transport service primitives.[7]
e. PCE-TSE–SNA element. Used for communication between PCE-TSE and the SNA element and deals with the establishment and the release of an SNA LU-LU session.
f. PCE-TSE–OSI-SL. Used for communication between PCE-TSE and OSI-SL and is a subset of the OSI transport service primitives.[7] This subset deals with the establishment and release of an OSI transport connection.

With the exception of Case e, all of the boundaries are based on the existing definition of OSI session and transport services and SNA LU 6.2 architecture.

When an incoming command related to the establishment of OSI transport connection is received by the gateway from the partner OSI node, it is changed to the appropriate service primitive, which is then forwarded to PCE-TSE. If the transport connection has been established and the command is related to the session or higher layers, the command is changed to the appropriate service primitive, which is forwarded to OSI-SL. OSI-SL then generates the appropriate service primitive, which is forwarded to PCE-SSE.

When the incoming command is from the partner SNA node and is related to the establishment of an SNA LU-LU session, an appropriate service primitive will be forwarded to PCE-TSE. If the SNA LU-LU session has been established and the incoming command is related to a conversation, the appropriate parameters that were returned to the conversation verbs will be forwarded to PCE-SSE.

Both PCE-SSE and PCE-TSE perform the protocol conversion on the basis of the service primitives or the returned parameters to the conversation verbs that they received.

Characteristics of the gateway structure. The major characteristics of the gateway structure are summarized below.

- Protocol conversion is performed layer by layer. For example, if a Connection Request (CR) transport protocol data unit (TPDU) is initiated from an OSI node, that CR TPDU is converted by the gateway to a BIND request unit (RU), which is then forwarded to the partner SNA node. Nothing is returned to the OSI node until the response is received from the partner SNA node. A transport connection initiated by an OSI node to the gateway will cause an SNA LU-LU session to be established between the gateway and the partner SNA node.
- Session and transport layer service primitives are preserved. As explained earlier, the boundaries between the PCE and the OSI elements are based on the service primitives defined in the OSI protocols,[5,7] and one of the boundaries between the PCE and the SNA element is based on the conversation verbs that are defined in the LU 6.2 architecture.[10]

Table 5 Command conversion from OSI Session to SNA LU 6.2 Conversation

OSI SPDU	API Mapped Conversation Verbs	RH/RU
Connect (CN)	MC_ALLOCATE	Begin Bracket (BB), FM Header (FMH) 5
Accept (AC)	—	
Refuse (RF)	MC_DEALLOCATE	Conditional End Bracket (CEB), FMH7
Data Transfer (DT)	MC_SEND_DATA	Data
Finish (FN)	MC_DEALLOCATE	CEB
Disconnect (DN)	—	
Abort (AB)	MC_DEALLOCATE (ABEND)	CEB, FMH7
Give Tokens (GT)	MC_RECEIVE_AND_WAIT (in send state) MC_PREPARE_TO_RECEIVE	Change Direction (CD)
Please Tokens (PT)	MC_REQUEST_TO_SEND	SIGNAL (0001)
Major Sync Point (MAP)	MC_CONFIRM	Definite Request (RQD) 2
Major Sync Ack (MAA)	MC_CONFIRMED	Definite Response (DR) 2
Exception Data (ED)	MC_SEND_ERROR	FMH7
Exception Report (ER)	MC_DEALLOCATE (ABEND)	CEB, FMH7

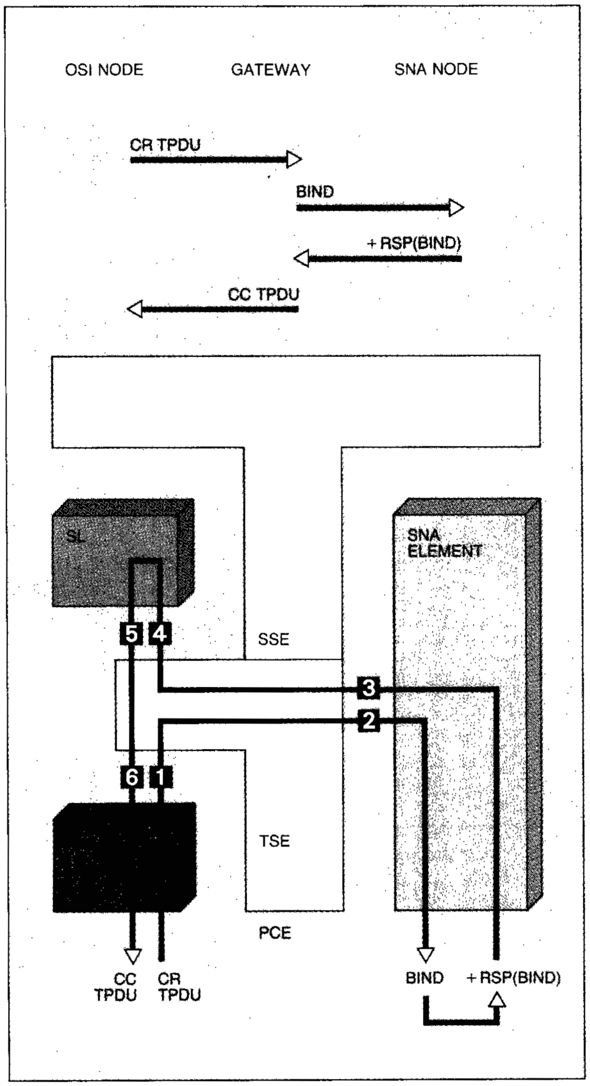
- Data transfer between end nodes is optimized in the gateway. The data transfer through the gateway has been optimized so that the additional delay caused by the gateway will be minimized. This is accomplished in the following ways:
  - Dual control paths are defined in the gateway. Data transfer between the end nodes will bypass PCE-TSE, thus increasing the efficiency of the data movement.
  - No segmentation or assembly is performed in the PCE. In both the OSI and SNA architectures, the service provider may choose to group the user data into smaller data segments. In the gateway, these segments are passed from one network to another on an "as is" basis to prevent delay of the data transfer. In order to accomplish this, boundary information is provided between the PCE and OSI-SNA elements to indicate the end of the user data.

Table 6 Command conversion from SNA LU 6.2 Conversation to OSI Session

SNA RH/RU	API and the Associated Mapped Conversation Verbs	OSI SPDU
BB, FMH5	—	CN
	RC = OK from MC_ALLOCATE	AC
	RC = ALLOCATION_ERROR from MC_ALLOCATE	RF
Data	WHAT_RECEIVED = DATA from MC_RECEIVE_AND_WAIT	DT
CEB	RC = DEALLOCATE_NORMAL from MC_RECEIVE_AND_WAIT or MC_SEND_ERROR	FN
CEB, FMH7	RC = 'derived from FMH7' MC_RECEIVE_AND_WAIT or MC_SEND_ERROR or MC_PREPARE_TO_RECEIVE or MC_CONFIRM or MC_DEALLOCATE or MC_SEND_DATA	AB (U)
FMH7	RC = PROG_ERROR_PURGING from MC_RECEIVE_AND_WAIT or MC_SEND_ERROR or MC_PREPARE_TO_RECEIVE or MC_CONFIRM or MC_DEALLOCATE or MC_SEND_DATA	ED
CD	WHAT_RECEIVED = SEND from MC_RECEIVE_AND_WAIT	GT
SIGNAL	REQUEST_TO_SEND_RECEIVED from MC_RECEIVE_AND_WAIT or MC_CONFIRM or MC_SEND_DATA	PT
RQD2	WHAT_RECEIVED = CONFIRM from MC_RECEIVE_AND_WAIT	MAP
DR2	RC = OK from MC_CONFIRM	MAA

Protocol conversion specifications. Once the gateway structure is defined, the next step is to define the protocol conversion specifications. Protocol conversion consists of command conversion and parameter

Figure 4 Transport connection initiated by OSI node



conversion. Command conversions for PCE-SSE are summarized in Tables 5 and 6. Command conversions for PCE-TSE are summarized in Tables 7 and 8.

A set of parameter conversions is associated with each command conversion. Given a command conversion, parameters of both the input and output commands can be grouped as follows:

- Parameters that need to be converted
- Parameters that are not forwarded but are managed in the gateway
- Parameters that are generated by the gateway

Table 7 Command conversion from OSI Transport to SNA LU-LU Session

OSI TPDU	SNA RU	Remarks
Connection Request (CR)	BIND	
Connection Confirm (CC)	Positive Response for BIND (+RSP(Bind))	
Disconnect Request (DR)	Negative Response for BIND (−RSP(BIND))	If Transport Connection is pending
DR	UNBIND	If Transport Connection has been established
Disconnect Confirm (DC)	Positive Response for UNBIND (+RSP(UNBIND))	If DR is converted from UNBIND
DC		If DR is initiated from Gateway
TPDU Error (ER)	−RSP(BIND)	If Transport Connection is pending
ER	UNBIND	If Transport Connection has been established
Data (DT)	Data	

Table 9 is a list of the parameters that need to be converted by the gateway and their associated commands.

Examples of the OSI-SNA gateway operation. Figure 4 shows a sample flow in which a transport connection request is initiated by the OSI node. This flow goes as follows:

1. An incoming Connection Request (CR) TPDU from the partner OSI node is changed to "T-CONNECT request" and is forwarded to PCE-TSE.
2. PCE-TSE converts the "T-CONNECT request" to a "BIND request" and sends it to the SNA element, which then sends a BIND RU to the partner SNA node.
3. A positive response for BIND (+RSP(BIND)) from the partner SNA node activates "BIND confirm" between the SNA element and PCE-TSE.
4. PCE-TSE converts the "BIND confirm" to a "T-CONNECT indication" and sends it to OSI-SL.
5. OSI-SL returns a "T-CONNECT response" to PCE-TSE.

6. PCE-TSE forwards the "T-CONNECT response" to OSI-TL, which causes a Connection Confirm (CC) TPDU to be sent to the partner OSI node.

Figure 5 shows a sample flow in which data transfer is requested by the OSI node. This flow is as follows:

1. An incoming Data Transfer (DT) TPDU from the partner OSI node is changed to a "T-DATA indication" in OSI-TL, which is then sent to OSI-SL.
2. OSI-SL processes the "T-DATA indication" and generates appropriate service primitives to PCE-SSE.
3. PCE-SSE converts the appropriate session service primitives to the corresponding conversation verbs, which causes an appropriate RU to be sent to the partner SNA node.

Address translation

One of the basic problems in OSI-SNA interconnection is address translation. Addressing is one of the fundamental differences between OSI and SNA LU 6.2. This section presents the addressing concept in each architecture, discusses the mapping technique chosen in this study, and gives examples illustrating how the scheme works.

Naming and addressing in OSI. Two naming authorities are proposed in SC21 in order to allocate names independently.[13]

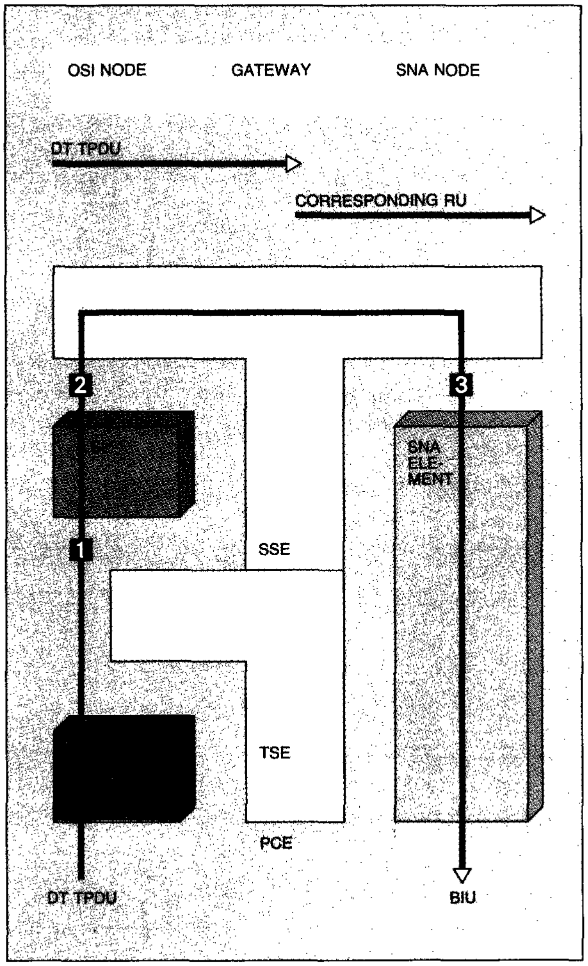
Table 8 Command conversion from SNA LU-LU Session to OSI Transport

SNA RU	OSI TPDU	Remarks
BIND	CR	
+RSP(BIND)	CC	
−RSP(BIND)	DR	
UNBIND	DR	
Data	DT	
+RSP(UNBIND)	DC	If UNBIND is converted from DR TPDU
+RSP(UNBIND)	—	If UNBIND is initiated from Gateway

Table 9 Parameter conversion between OSI and SNA LU 6.2

OSI Commands	OSI Parameters	SNA Commands	SNA Parameters
CN SPDU	Token Setting Items User requirement Called SSAP Calling SSAP	FMH5	End Chain Indicator (ECI) Change Direction Indicator (CDI) Sync. Level information TP Name (TPN) Calling TPN (to be in the PIP)
RF SPDU	Reason Code	FMH7, CEB	Sense Data
DT SPDU	Enclosure Item	Data RU	LL
GT SPDU	Token Item	RH	CDI
PT SPDU	Token Item	SIGNAL	SIGNAL Code
ED SPDU	Reason Code	FMH7	Sense Data
ER SPDU	(None)	FMH7	Sense Data
AB SPDU	Type	FMH7	Sense Data
CR TPDU	Request Code Calling TSAP Called SSAP ID	BIND	Request Code PLU Name SLU Name
CC TPDU	Request Code Calling TSAP ID Called SSAP ID	+RSP(BIND)	Request Code Primary LU (PLU) Name Secondary LU (SLU) Name
DR TPDU	Request Code	−RPS(BIND)	Request Code
ER TPDU	Request Code Reject Cause	−RSP(BIND)	Request Code Sense Code
DR TPDU	Request Code Reason	UNBIND	Request Code UNBIND Type

Figure 5 Data transfer initiated by OSI node



The first naming authority provides addresses for network service access points (NSAPs) in such a way that there is a one-to-one correspondence between NSAP addresses and NSAPs throughout the entire OSI. Associated with this naming authority is a directory that provides the mapping between an NSAP address and the routing information used below the network service boundary to access the NSAP.

The second naming authority provides application titles in such a way that there is a many-to-one correspondence between application titles and application entities. Associated with this naming authority is another directory that provides the mapping between the application title and the addressing information needed to access it.

Within the scope of the transport entity identified by a single NSAP address, a transport selector is allocated in such a way that there is a one-to-one correspondence between values of the selector and transport service access points (TSAPs), which can be accessed locally by the transport entity attached to that NSAP. The tuple (NSAP address, transport selector) is a TSAP address that is unique throughout the OSI.

Within the scope of the session entity identified by a single TSAP address, a session selector is allocated in such a way that there is a one-to-one correspondence between values of the selector and session service access points (SSAPs), which can be accessed locally by the session entity attached to that TSAP. The tuple (TSAP address, session selector) is an SSAP address that is unique throughout the OSI.

There is a one-to-one correspondence between presentation service access points (PSAPs) and SSAPs, and the SSAP address is equivalent to the PSAP address.[14]

In order to ensure that all routing decisions are taken below the network service boundary, all layer entities above the network layer have a single point of attachment to their lower service access point. Thus, there is a many-to-one correspondence between application titles and NSAP addresses.

This is summarized as shown in Figure 6.

Names and addresses in SNA. A network addressable unit in an SNA network could be a logical unit (LU), physical unit (PU), or system services control point (SSCP). Each of them has a unique network address in a given SNA network. The network address consists of two parts: a subarea address and an element address. The subarea address is the same for all network addressable units in the same subarea. The element address is unique to each network addressable unit within that subarea. Network addresses are used only within the SNA network. End users refer to network addressable units by their network names. Each network addressable unit within an SNA network must have a unique name. A network directory service is used to map the network names to their corresponding network addresses.

A transaction program (TP) uses the LU to communicate with another TP. An LU may run many TPs successively or concurrently, or both. Within a network, TP names need not be unique; however, within the same LU, TP names must be unique. The relationship between TP and LU is shown in Figure 7.

Address translation for OSI-SNA interconnection. As discussed earlier, an SNA LU 6.2 conversation is mapped into an OSI session connection and an SNA LU-LU session is mapped into an OSI transport connection. Therefore, the TP names and LU names in the SNA network are mapped into SSAP and TSAP in the OSI network and vice versa.

The mechanism for mapping is based on the SNA Network Interconnection (SNI).[15] A TP name and an LU name are represented respectively by SSAP and TSAP addresses in the OSI network. Similarly, an SSAP address and a TSAP address are represented respectively by a TP name and an LU name in the SNA network. The mapping is performed in the gateway, as shown in Figure 8. In this figure, TP-A and LU-A are represented respectively by SSAP-X and TSAP-X for the OSI network in the gateway, and SSAP-A and TSAP-A are represented respectively by TP-X and LU-X for the SNA network in the gateway. When TP-A wishes to communicate with SSAP-A, it only sees the existence of TP-X. Likewise, when SSAP-A wishes to communicate with TP-A, it only sees the existence of SSAP-X.

Figure 6 Proposed OSI addressing structure

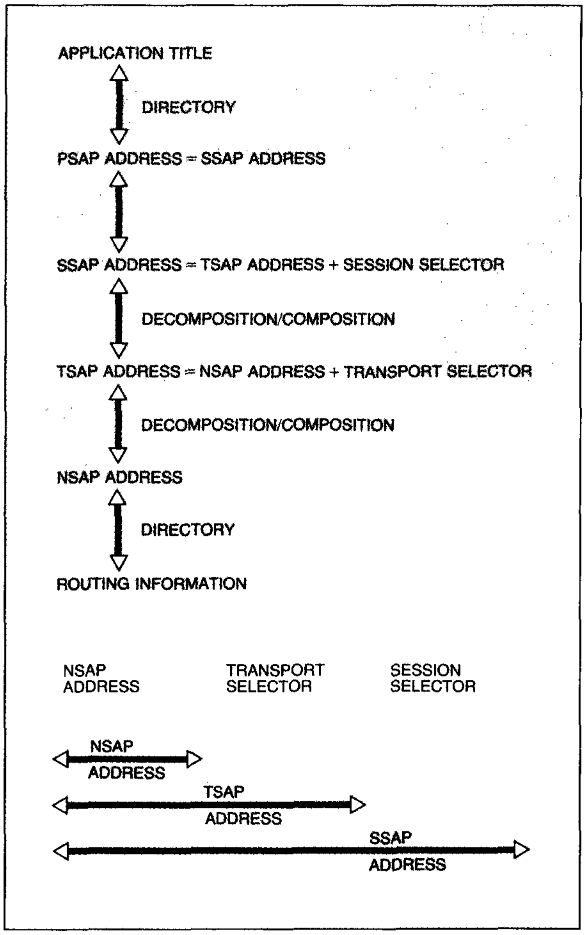
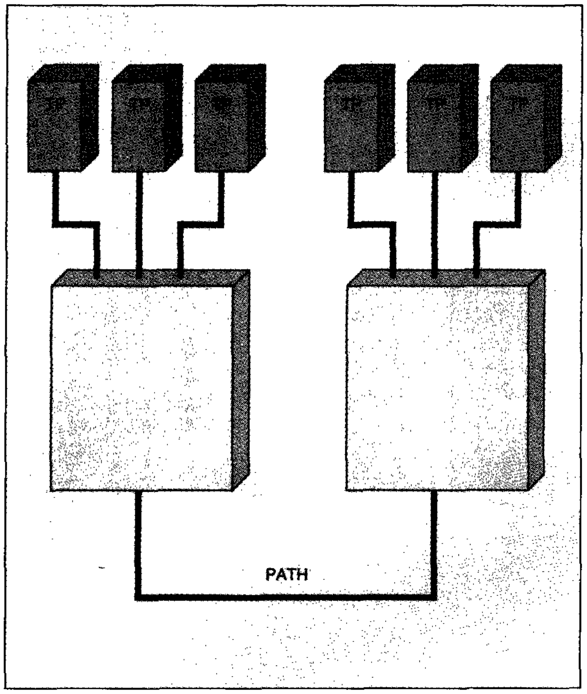


Figure 7 Relationship between LU and TP



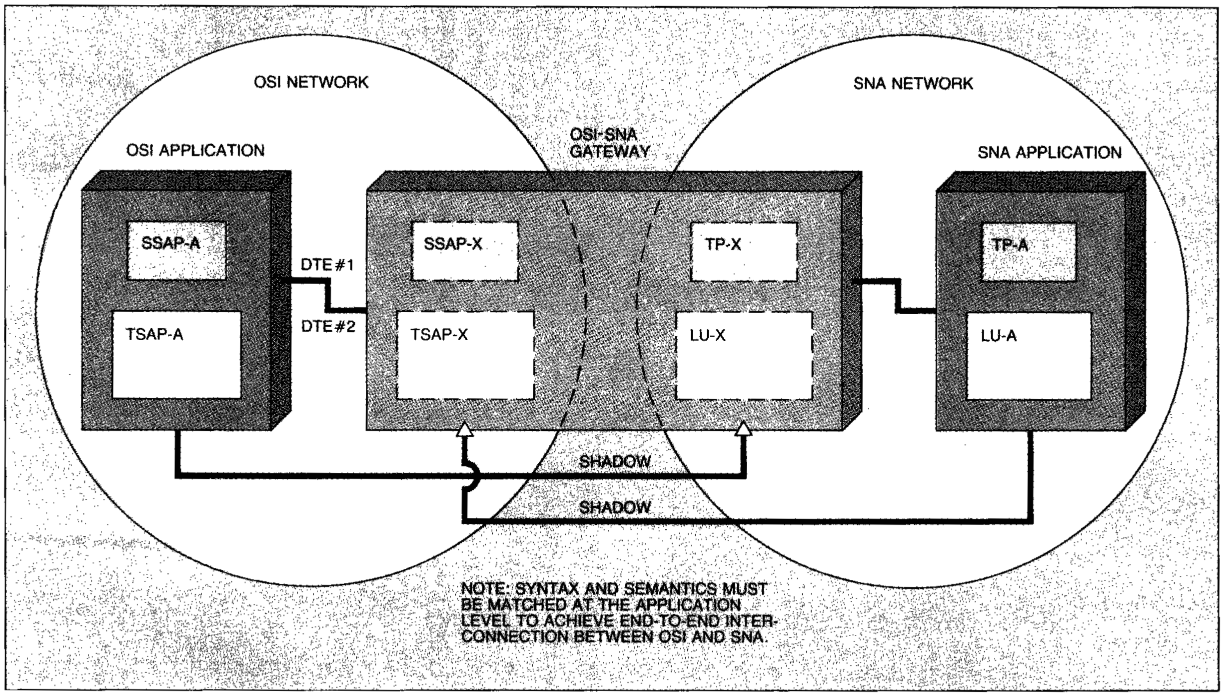
To accomplish the mapping, assume that all nodes in the OSI network are connected via an X.25 packet-switched network. The NSAP address is then equal to the Data Terminal Equipment (DTE) address that is assigned by the X.25 packet-switched network. In this paper, we assume that NSAP addresses DTE#1 and DTE#2 are assigned to an OSI node and to the OSI-SNA gateway, respectively (Figure 8).

According to Figure 6, the TSAP address and SSAP address are defined as follows:

TSAP address = NSAP address + transport selector
SSAP address = TSAP address + session selector
= NSAP address + transport selector
+ session selector

Since the NSAP address is nothing but the DTE address of the gateway, the uniqueness of the DTE address guarantees the uniqueness of the TSAP address and the SSAP address in the particular OSI network.

Figure 8 Address translation for OSI-SNA interconnection



TSAP-X, which represents the LU-A to the OSI network, must be assigned a unique TSAP address for addressing in the OSI network. Since the LU name, LU-A, is unique in the SNA network, the TSAP-X can be uniquely represented in the OSI network by using the LU name as a transport selector.

SSAP-X, which represents the TP-A to the OSI network, must be assigned a unique SSAP address for addressing in the OSI network. Since the TP name, TP-A, is unique in a particular LU, the TP name can be used as a session selector.[16]

With these rules, the TSAP address (TSAP-X) and SSAP address (SSAP-X) have the following values:

TSAP-X = DTE#2 + LU-A
SSAP-X = DTE#2 + LU-A + TP-A

Similarly, the OSI entities, TSAP-A and SSAP-A, must be assigned an LU name and a TP name for the SNA network. As shown in Figure 8, the transport entity and session entity are mapped to LU and TP. With use of this mapping, an LU name, LU-X, and a TP name, TP-X, are assigned to the TSAP address and SSAP address, respectively. The values of LU-X and TP-X are given as follows:

LU-X = DTE#1 + TSAP-A
TP-X = DTE#1 + TSAP-A + SSAP-A

As a result of the above mapping, translation tables A and B can be established, as shown in Figure 9. Table A is the translation table between the LU name and the TSAP address. Table B is the translation table between the TP name and the SSAP address. All the tables are predefined and established in the gateway prior to the connection.

Given the network configuration in Figure 8, Figure 10 shows the sequence including address translation when connection is initiated from the SNA network. Tables A and B in Figure 9 are used to translate LU names and TP names to their corresponding TSAPs and SSAPs, and vice versa. Figure 11 shows the sequence including address translation when connection is initiated from the OSI network. Tables A and B are also used to translate TSAPs and SSAPs to their corresponding LU names and TP names.

Exception handling

Another topic to be considered in designing a protocol conversion is exception handling. Both OSI and SNA LU 6.2 define two major categories of exceptions:

(1) user error, which means an error detected by service users, and (2) provider error, which means an error detected by service providers. Since the detailed categorization of exceptional cases and related recovery actions are quite different in the two architectures, it is difficult to map error information between OSI and SNA networks.

In OSI, all exceptions that occur on top of the OSI session layer must be handled as session service user (ss-user) exceptions by definition. In SNA LU 6.2, all exceptions that occur on top of the conversation are defined as user errors. As a result, it is not possible for the remote user to distinguish between real user exceptions issued by applications and provider exceptions detected by protocol converters inside the gateway.

On the basis of the above discussion, we designed the OSI-SNA gateway to handle exceptions as follows:

- A user error is propagated to the partner node as a user error.
- A provider error is not converted. The gateway requests abnormal connection release to both networks.

Usually, a provider error causes an abnormal connection release in the network that detected the error. Therefore, the above gateway design is a reasonable choice to handle a provider error in an interconnected network environment.

Figure 9 Address translation table in the gateway

TABLE A ADDRESS TRANSLATION TABLE BETWEEN LU NAMES AND TSAP ADDRESSES

LU NAME (SNA ENTITY)	TSAP ADDRESS		OSI ENTITY
	NSAP ADDRESS	TRANSPORT SELECTOR	
LU-A	DTE#2	LU-A	TSAP-X
LU-X	DTE#1	TSAP-A	TSAP-A

TABLE B ADDRESS TRANSLATION TABLE BETWEEN TP NAMES AND SSAP ADDRESSES

TP NAME (SNA ENTITY)	SSAP ADDRESS			OSI ENTITY
	TSAP ADDRESS		SESSION SELECTOR	
	NSAP ADDRESS	TRANSPORT SELECTOR		
TP-A	DTE#2	LU-A	TP-A	SSAP-X
TP-X	DTE#1	TSAP-A	SSAP-A	SSAP-A

Concluding remarks

The OSI-SNA gateway described in this paper provides a theoretical mechanism to interconnect an SNA network and non-SNA networks using OSI as an intermediate network. With the technique described in this paper, a gateway between OSI and another network architecture can also be designed. If the OSI-SNA gateway and another gateway between OSI and the third network have a common subset of functions, then, subject to common syntax and semantics, application programs in an SNA network and application programs in the third network can communicate with each other.

Acknowledgment

As indicated previously, this paper is based on the results of the joint study with NTT. The authors wish to acknowledge the NTT team for their cooperative and professional efforts, which were essential to the success of the study. We would also like to thank P. O. Lindfors and Y. Watanabe for organizing the IBM team, J. L. Cox and H. Akiyoshi for their management support, and W. Siddall for his help on architecture and standards questions and his review of the initial paper.

Cited references and notes

1. ISO 7498, *Open Systems Interconnection—Basic Reference Model*, International Organization for Standardization, Geneva, Switzerland (1983).
2. D. Einert and G. Glas, "The SNATCH gateway: Translation of high level protocols," *Journal of Telecommunication Networks*, 0276-0037/83 (1983), pp. 83–102. The SNATCH gateway interconnects SNA and TRANSDATA through PIX, a network architecture developed by several German universities and research institutes.
3. P. François and A. Potocki, "Some methods for providing OSI transport in SNA," *IBM Journal of Research and Development* **27**, No. 5, 452–463 (1983).
4. P. E. Green, "Protocol conversion," *IEEE Transactions on Communications* **COM-34**, No. 3, 257–268 (March 1986).
5. ISO/DIS 8326, *Open Systems Interconnection—Basic Connection Oriented Session Service Definition*, International Or-

Figure 10 Connection initiated by SNA network

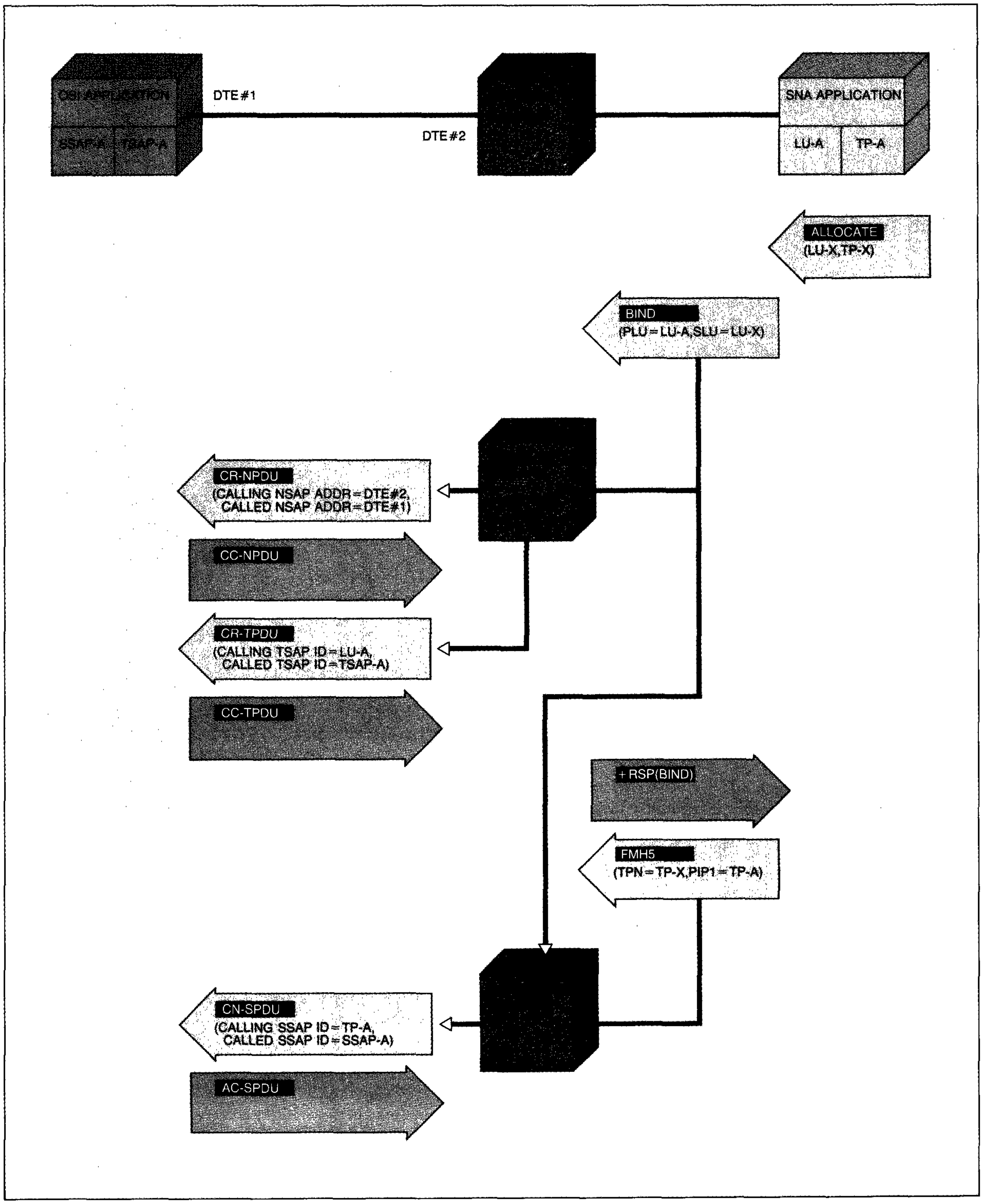
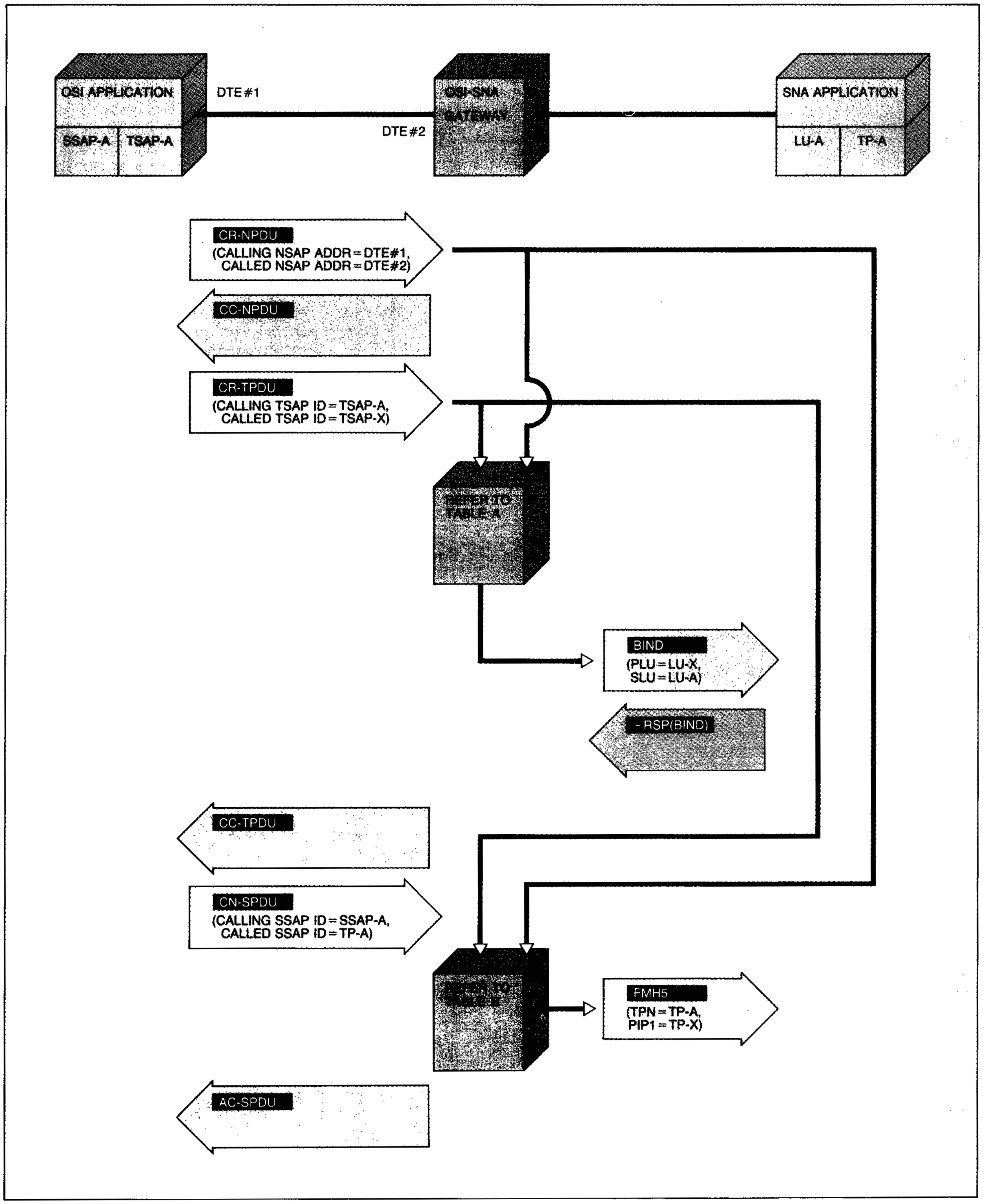


Figure 11 **Connection initiated by OSI network**



ganization for Standardization, Geneva, Switzerland (1983). This document is not the latest version. The time element of this study applied to the November 1983 version.

6. ISO/DIS 8327, *Open Systems Interconnection—Basic Connection Oriented Session Protocol Definition*, International Organization for Standardization, Geneva, Switzerland (1983). This document is not the latest version. The time element of this study applied to the November 1983 version.
7. ISO/DIS 8072, *Open Systems Interconnection—Transport Service Definition*, International Organization for Standardization, Geneva, Switzerland (1983). This document is not the latest version. The time element of this study applied to the November 1983 version.
8. ISO/DIS 8073, *Open Systems Interconnection—Basic Connection Oriented Transport Protocol Definition*, International Organization for Standardization, Geneva, Switzerland (1983). This document is not the latest version. The time element of this study applied to the November 1983 version.
9. Since 1984, significant additional standards work has been done in this area, and current standards proposals may involve different mappings due to the advance in OSI standards.
10. *SNA Transaction Programmer's Reference Manual for LU Type 6.2*, GC30-3084, IBM Corporation; available through IBM branch offices.
11. S. Shukuya, M. Yamaguchi, and Y. Kobayashi, "OSI Subsets from SNA LU 6.2 functional viewpoints," *Proceedings of IPSJ Symposium, Spring 1985*, Information Processing Society of Japan (March 1985), pp. 1053–1054. This paper is only available in the Japanese language.
12. In Table 3, the major synchronization of OSI is mapped into CONFIRM of SNA LU 6.2. This is one of the possible mappings. A possible alternative is to map the Minor Synchronization into CONFIRM.
13. ISO 7498/DAD, *Naming and Addressing*, International Organization for Standardization, Geneva, Switzerland (1984). This document is not the latest version. The time element of this study applied to the April 1984 version.
14. DAD7498 has been updated since the time frame of this study. Currently PSAP and SSAP do not have one-to-one correspondence.
15. 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).
16. The Program Initiation Parameter 1 (PIP1) of the FMH5 is used to carry the information of the session selector in the SNA environment. It is one of the constraints to SNA TPs.

General references

M. Yamaguchi and K. K. Sy, "Gateway to make the SNA network an open system through OSI," *Proceedings of IPSJ Symposium, Spring 1985*, Information Processing Society of Japan (March 1985), pp. 1055–1056. This paper is only available in the Japanese language.

M. O. Shiobara, K. K. Sy, and M. Yamaguchi, "Protocol conversion between SNA LU 6.2 and OSI session/transport layers," *Proceedings of IPSJ Symposium, Spring 1985*, Information Processing Society of Japan (March 1985), pp. 1057–1058. This paper is only available in the Japanese language.

Y. Kobayashi, M. O. Shiobara, H. Wakayama, and Y. Ohara, "OSI protocols as intermediary for DCNA-SNA network interconnection," *Proceedings of the Third International Conference on the Introduction of Open Systems Standards*, Department of Trade and Industry, United Kingdom (September 1985), pp. 312–344.

K. K. Sy and M. O. Shiobara, "SNA to OSI: Layer correspondence and gateway design," *Proceedings of Computer Communications '85 Asia* (December 1985), pp. 341–352.

Kian-bon K. Sy *IBM Communication Products Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Sy is an advisory engineer with the Architecture and Telecommunications Group, where he has worked on the architecture of local-area networks. In 1984, he was on assignment to Japan to work on the Heterogeneous Computer Network Interconnection and information network services. His current responsibility is in the area of network management dealing with ISDN. Mr. Sy has published several papers on local-area networks and OSI-SNA interconnection. He has received a second-level Invention Achievement Award and an Outstanding Technical Achievement Award. He received his B.S. and M.S. degrees in electrical engineering from the University of Pennsylvania.

Michael O. Shiobara *IBM Japan, Yamato Laboratory, 1623-14, Shimotsuruma, Yamato-shi, Kanagawa-ken 242, Japan.* Mr. Shiobara is currently a senior associate development engineer. He joined IBM Japan in 1981, participating at first in the system design of intelligent workstations such as the IBM 5550 Multistation. Since 1983, he has been working in the area of communication architectures. In 1984, he was named a working group member of the joint study with NTT of the Heterogeneous Computer Network Interconnection. His major contribution was the definition of the Intermediate OSI Protocols for OSI-SNA interconnection and the protocol conversion between OSI and SNA LU 6.2. His current responsibility is in the area of communication architectures such as ISDN and SNA. Mr. Shiobara has published several papers on OSI-SNA interconnection. He received his B.E. degree in applied physics from the University of Tokyo.

Masato Yamaguchi *IBM Japan, Systems Engineering, Kowa Tsukiji Building, 18-24, Tsukiji 7-chome, Chuo-ku, Tokyo 104, Japan.* Mr. Yamaguchi is currently an advisory systems engineer with the System Design Support Center. Since he joined IBM Japan in 1974, he has participated in marketing and installation support of communication systems such as ACF/VTAM, ACF/NCP, and SNA-related products. In 1984, he took part in the joint study with NTT of the Heterogeneous Computer Network Interconnection. He received his B.E. degree in mechanical engineering from the Kyushu Institute of Technology.

Yoshikazu Kobayashi *IBM Japan Headquarters, Technical Relations, 2-12, Roppongi 3-chome, Minato-ku, Tokyo 106, Japan.* Mr. Kobayashi is currently a senior standards planner. After joining IBM Japan, he worked as a systems engineer from 1970 to 1972, and then transferred to Standards. In 1984, he participated in the joint study with NTT of the Heterogeneous Computer Network Interconnection. He is currently involved in OSI standardization in ISO as a convener of ISO/TC97/SC21/WG4, which is responsible for OSI management. Mr. Kobayashi received his B.S. degree in physics from the University of Hiroshima in 1970.

Shohji Shukuya *IBM Japan, Yamato Laboratory, 1623-14, Shimotsuruma, Yamato-shi, Kanagawa-ken 242, Japan.* Mr. Shukuya is currently the manager of Systems Support and Architecture at the Yamato Laboratory. He joined IBM Japan in 1977 and contributed to the development of the IBM 3276 Control Unit Display Station and the IBM 3101 Display Station. From 1981 through 1983, he participated in SNA development while on assignment at the IBM facility in Research Triangle Park, North Carolina. Mr. Shukuya received his B.E. degree in electrical and electronic engineering from Sophia University, Tokyo, in 1975 and his M.S. degree in computer science from the University of California at Los Angeles in 1977.

Takahiro Tomatsu *IBM Japan, Tokyo Programming Center, 1-14, Nisshin-cho, Kawasaki-ku, Kawasaki-shi, Kanagawa-ken 210, Japan.* Mr. Tomatsu is currently the manager of Cross Industry Software Development. He joined IBM in 1971 as an engineer and participated in the system design and microprogramming of several products such as the IBM 3613 Japanese Banking Terminal. From 1977 to 1979, he was on assignment at the IBM laboratory in Research Triangle Park, North Carolina, where he worked on the development of SNA. After returning to IBM Japan, he participated in the implementation of SNA LU 6.2 and DIA/DCA on the IBM 8815 Scanmaster I. In 1984, he was the manager of Architecture and participated in the joint study with NTT of the Heterogeneous Computer Network Interconnection. Mr. Tomatsu received his B.S. and M.S. degrees in computer science from the University of California at Los Angeles.

Reprint Order No. G321-5292.