इंटरनेट मानक

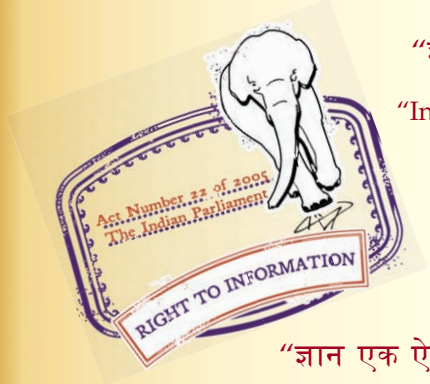
Disclosure to Promote the Right To Information

Whereas the Parliament of India has set out to provide a practical regime of right to information for citizens to secure access to information under the control of public authorities, in order to promote transparency and accountability in the working of every public authority, and whereas the attached publication of the Bureau of Indian Standards is of particular interest to the public, particularly disadvantaged communities and those engaged in the pursuit of education and knowledge, the attached public safety standard is made available to promote the timely dissemination of this information in an accurate manner to the public.

"जानने का अधिकार, जीने का अधिकार"
Mazdoor Kisan Shakti Sangathan
"The Right to Information, The Right to Live"

"पुराने को छोड नये के तरफ"
Jawaharlal Nehru
"Step Out From the Old to the New"

IS 12373-1 (1987): Basic Reference Model of open Systems Interconnection for Information Processing Systems [LITD 14: Software and System Engineering]



"ज्ञान से एक नये भारत का निर्माण"
Satyanarayan Gangaram Pitroda
"Invent a New India Using Knowledge"

"ज्ञान एक ऐसा खजाना है जो कभी चुराया नहीं जा सकता है"
Bhartṛhari—Nītiśatakam
"Knowledge is such a treasure which cannot be stolen"

BLANK PAGE

UDC 681.3.01 According to Amendment no. 1, it is 12373_1

IS : 12373 - 1987
ISO 7498 - 1984

Indian Standard

(Reaffirmed 2003)

BASIC REFERENCE MODEL OF OPEN SYSTEMS INTERCONNECTION FOR INFORMATION PROCESSING SYSTEMS

(ISO Title : Information Processing Systems — Open Systems Interconnection — Basic Reference Model)

National Foreword

This Indian Standard, which is identical with ISO 7498-1984 'Information processing systems — Open systems interconnection — Basic Reference model', issued by International Organization for Standardization (ISO), was adopted by the Bureau of Indian Standards on the recommendation of the Computers, Business Machines and Calculators Sectional Committee and approved by the Electronics and Telecommunication Division Council.

In the adopted standard certain terminology and conventions are not identical with those used in Indian Standards; attention is specially drawn to the following:

Wherever the words 'International Standard' appear referring to this standard, they should be read as 'Indian Standard'.

Adopted 16 October 1987

Gr 13

BUREAU OF INDIAN STANDARDS
MANAK BHAVAN, 9 BAHADUR SHAH ZAFAR MARG
NEW DELHI 110002

As in the Original Standard, this Page is Intentionally Left Blank

Contents

0 Introduction

0.1 About this standard

The purpose of this International Standard Reference Model of Open Systems Interconnection is to provide a common basis for the coordination of standards development for the purpose of systems interconnection, while allowing existing standards to be placed into perspective within the overall Reference Model.

The term Open Systems Interconnection (OSI) qualifies standards for the exchange of information among systems that are "open" to one another for this purpose by virtue of their mutual use of the applicable standards.

The fact that a system is open does not imply any particular systems implementation, technology or means of interconnection, but refers to the mutual recognition and support of the applicable standards.

It is also the purpose of this International Standard to identify areas for developing or improving standards, and to provide a common reference for maintaining consistency of all related standards. It is not the intent of this International Standard either to serve as an implementation specification, or to be a basis for appraising the conformance of actual implementations, or to provide a sufficient level of detail to define precisely the services and protocols of the interconnection architecture. Rather, this International Standard provides a conceptual and functional framework which allows international teams of experts to work productively and independently on the development of standards for each layer of the Reference Model of OSI.

The Reference Model has sufficient flexibility to accommodate advances in technology and expansion in user demands. This flexibility is also intended to allow the phased transition from existing implementations to OSI standards.

NOTE — The Reference Model is expected to be subject to future expansion. Some anticipated directions of expansion are indicated by notes or footnotes in this International Standard.

While the scope of the general architectural principles required for OSI is very broad, this International Standard is primarily concerned with systems comprising terminals, computers and associated devices and the means for transferring information between such systems. Other aspects of OSI requiring attention are described briefly (see 4.2).

The justification for development of standards shall follow normal administrative procedures even though such standards are identified in the Reference Model.

As standards emerge to meet the OSI requirements, a small number of practical subsets should be defined by the standards developers from optional functions, to facilitate implementation and compatibility.

The description of the Reference Model of OSI given in this International Standard is developed in stages :

Clause 4 establishes the reasons for Open Systems Interconnection, defines what is being connected, the scope of the interconnection and, describes the modelling principles used in OSI;

Clause 5 describes the general nature of the architecture of the Reference Model: namely that it is layered, what layering means, and the principles used to describe layers;

Clause 6 names, and introduces the specific layers of the architecture; and

Clause 7 provides the description of the specific layers.

An indication of how the layers were chosen is given in annex A to this International Standard.

The Reference Model serves as a framework for the definition of services and protocols which fit within the boundaries established by the Reference Model.

In those few cases where a feature is explicitly marked (optional) in the Reference Model it should remain optional in the corresponding service or protocol (even if at a given instant the two cases of the option are not yet documented).

0.2 Related OSI standards

Concurrently with the preparation of this International Standard, work is in progress within ISO on the development of OSI standards in the following areas :

a) virtual terminal protocols;

b) file transfer, access and management protocols;

c) job transfer and manipulation protocols;

d) common application services and protocols;

e) presentation layer services and protocols;

f) Session Layer services and protocols;

g) Transport Layer services and protocols;

h) Network Layer services and protocols;

j) Data Link Layer services and protocols;

k) Physical Layer services and protocols; and

m) OSI management protocols.

The first five items in this list relate to the Application and Presentation Layers of the Reference Model.

In addition, liaison is maintained with CCITT in the development of OSI standards.

1 Scope and field of application

This International Standard describes the Reference Model of Open Systems Interconnection. It establishes a framework for coordinating the development of existing and future standards for the interconnection of systems and is provided for reference by those standards.

This International Standard does not specify services and protocols for OSI. It is neither an implementation specification for systems, nor a basis for appraising the conformance of implementations.

2 Definitions

Definitions of terms are included at the beginning of individual clauses and sub-clauses. An index of these terms is provided in an annex B for easy reference.

3 Notation

Layers are introduced in clause 5. An (N)-, (N + 1)- and (N − 1)- notation is used to identify and relate adjacent layers :

(N)-layer : any specific layer;

(N + 1)-layer : the next higher layer;

(N − 1)-layer : the next lower layer.

This notation is also used for other concepts in the model which are related to these layers, for example (N)-protocol, (N + 1)-service.

Clause 6 introduces names for individual layers. When referring to these layers by name, the (N)-, (N + 1)- and (N − 1)- prefixes are replaced by the names of the layers, for example transport-protocol, session entity, and network-service.

4 Introduction to Open Systems Interconnection (OSI)

NOTE — The general principles described in clauses 4 and 5 hold for all layers of the Reference Model, unless layer specific statements to the contrary are made in clauses 6 and 7.

4.1 Definitions

4.1.1 real system : A set of one or more computers, the associated software, peripherals, terminals, human operators, physical processes, information transfer means, etc., that forms an autonomous whole capable of performing information processing and/or information transfer.

4.1.2 real open system : A real system which complies with the requirements of OSI standards in its communication with other real systems.

4.1.3 open system : The representation within the Reference Model of those aspects of a real open system that are pertinent to OSI.

4.1.4 application-process : An element within a real open system which performs the information processing for a particular application.

4.2 Open Systems Interconnection environment

In the concept of OSI, a real 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 performing information processing and/or information transfer.

An application-process is an element within an open system which performs the information processing for a particular application.

Application-processes can represent manual processes, computerized processes or physical processes.

Some examples of application-processes that are applicable to this open system definition are the following :

a) a person operating a banking terminal is a manual application-process;

b) a FORTRAN program executing in a computer centre and accessing a remote database is a computerised application-process; the remote database management systems server is also an application-process; and

c) a process control program executing in a dedicated computer attached to some industrial equipment and linked into a plant control system is a physical application-process.

OSI is concerned with the exchange of information between open systems (and not the internal functioning of each individual real open system).

As shown in figure 1, the physical media for Open Systems Interconnection provides the means for the transfer of information between open systems.

NOTE — At this point, only telecommunications media have been considered. The use of other interconnection media is for further study.

OSI is concerned only with interconnection of systems. All other aspects of systems which are not related to interconnection are outside the scope of OSI.

OSI is concerned not only with the transfer of information between systems, i.e. transmission, but also with their capability to interwork to achieve a common (distributed) task. In other words, OSI is concerned with the interconnection aspects of cooperation[1] between systems, which is implied by the expression "systems interconnection".

The objective of OSI is to define a set of standards to enable real open systems to cooperate. A system which complies with the requirements of applicable OSI standards in its cooperation with other systems is termed a real open system.

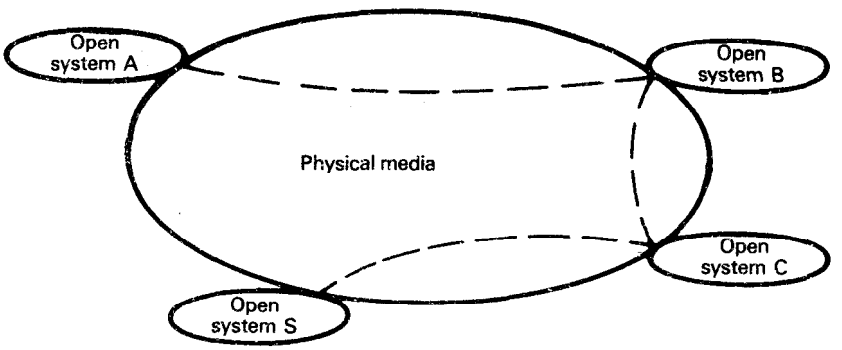


Figure 1 — Open systems connected by physical media

1) Cooperation among open systems involves a broad range of activities of which the following have been identified :

a) interprocess communication, which concerns the exchange of information and the synchronization of activity between OSI application-processes;

b) data representation, which concerns all aspects of the creation and maintenance of data descriptions and data transformations for reformatting data exchanged between open systems;

c) data storage, which concerns storage media, and file and database systems for managing and providing access to data stored on the media;

d) process and resource management, which concerns the means by which OSI application-processes are declared, initiated and controlled, and the means by which they acquire OSI resources;

e) integrity and security, which concern information processing constraints that have to be be preserved or assured during the operation of the open systems; and

f) program support, which concerns the definition, compilation, linking, testing, storage, transfer, and access to the programs executed by OSI application-processes.

Some of these activities may imply exchange of information between the interconnected open systems and their interconnection aspects may, therefore, be of concern to OSI.

This International Standard covers the elements of OSI aspects of these activities which are essential for early development of OSI standards.

4.3 Modelling the OSI environment

The development of OSI standards, i.e. standards for the interconnection of real open systems, is assisted by the use of abstract models. To specify the external behaviour of interconnected real open systems, each real open system is replaced by a functionally equivalent abstract model of a real open system called an open system. Only the interconnection aspects of these open systems would strictly need to be described. However to accomplish this, it is necessary to describe both the internal and external behaviour of these open systems. Only the external behaviour of open systems is retained as the standard of behaviour of real open systems. The description of the internal behaviour of open systems is provided in the Reference Model only to support the definition of the interconnection aspects. Any real system which behaves externally as an open system can be considered to be a real open system.

This abstract modelling is used in two steps.

First, basic elements of open systems and some key decisions concerning their organization and functioning, are developed. This constitutes the Reference Model of Open Systems Interconnection described in this International Standard.

Then, the detailed and precise description of the functioning of the open system is developed in the framework formed by the Reference Model. This constitutes the services and protocols for OSI which are the subject of other International Standards.

It should be emphasized that the Reference Model does not, by itself, specify the detailed and precise functioning of the open system and, therefore, it does not specify the external behaviour of real open systems and does not imply the structure of the implementation of a real open system.

The reader not familiar with the technique of abstract modelling is cautioned that those concepts introduced in the description of open systems constitute an abstraction despite a similar appearance to concepts commonly found in real systems. Therefore real open systems need not be implemented as described by the Model.

Throughout the remainder of this International Standard, only the aspects of real systems and application-processes which lie within the OSI environment are considered. Their interconnection is illustrated throughout this International Standard as depicted in figure 2.

5 Concepts of a layered architecture

5.1 Introduction

Clause 5 sets forth the architectural concepts that are applied in the development of the Reference Model of Open Systems Interconnection. Firstly, the concept of a layered architecture (with layers, entities, service-access-points, protocols, connections, etc.) is described. Secondly, identifiers are introduced for entities, service-access-points, and connections. Thirdly, service-access-points and data-units are described. Fourthly, elements of layer operation are described including connections, transmission of data, and error functions. Then, routing aspects are introduced and finally, management aspects are discussed.

The concepts described in clause 5 are those required to describe the Reference Model of Open Systems Interconnection. However, not all of the concepts described are employed in each layer of the Reference Model.

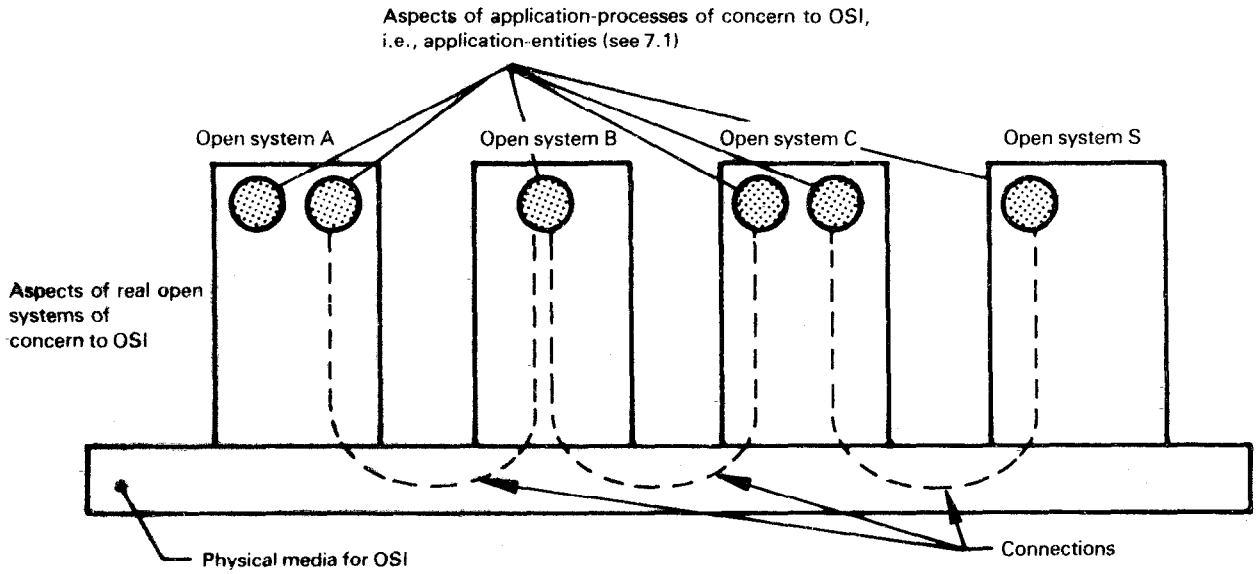


Figure 2 — Basic elements of OSI

Four elements are basic to the Reference Model (see figure 2) :

a) open systems;

b) the application-entities which exist within the OSI environment;

c) the connections (see 5.3) which join the application-entities and permit them to exchange information (see note 1); and

d) the physical media for OSI.

NOTES

1 This Basic Reference Model of Open Systems Interconnection is based on the assumption that a connection is required for the transfer of data. An addendum to this International Standard is currently being developed to extend the description to cover the connectionless forms of data transmission which may be found in a wide variety of data communications techniques (for example local area networks, digital radio, etc.) and applications (for example remote sensing and banking).

2 Security aspects which are also general architectural elements of protocols are not discussed in this International Standard.

5.2 Principles of layering

5.2.1 Definitions

5.2.1.1 (N)-subsystem : An element in a hierarchical division of an open system which interacts directly only with elements in the next higher division or the next lower division of that open system.

5.2.1.2 (N)-layer : A subdivision of the OSI architecture, constituted by subsystems of the same rank (N).

5.2.1.3 (N)-entity : An active element within an (N)-subsystem.

5.2.1.4 peer-entities : Entities within the same layer.

5.2.1.5 sublayer : A subdivision of a layer.

5.2.1.6 (N)-service : A capability of the (N)-layer and the layers beneath it, which is provided to (N + 1)-entities at the boundary between the (N)-layer and the (N + 1)-layer.

5.2.1.7 (N)-facility : A part of an (N)-service.

5.2.1.8 (N)-function : A part of the activity of (N)-entities.

5.2.1.9 (N)-service-access-point : The point at which (N)-services are provided by an (N)-entity to an (N + 1)-entity.

5.2.1.10 (N)-protocol : A set of rules and formats (semantic and syntatic) which determines the communication behaviour of (N)-entities in the performance of (N)-functions.

5.2.2 Description

The basic structuring technique in the Reference Model of Open Systems Interconnection is layering. According to this technique, each open system is viewed as being logically composed of an ordered set of subsystems, represented for convenience in the vertical sequence shown in figure 3. Adjacent subsystems communicate through their common boundary. Subsystems of the same rank (N) collectively form the (N)-layer of the Reference Model of Open Systems Interconnection. An (N)-subsystem consists of one or several (N)-entities. Entities exist in each layer. Entities in the same layer are termed peer-entities. Note that the highest layer does not have an (N + 1)-layer above it and the lowest layer does not have an (N – 1)-layer below it.

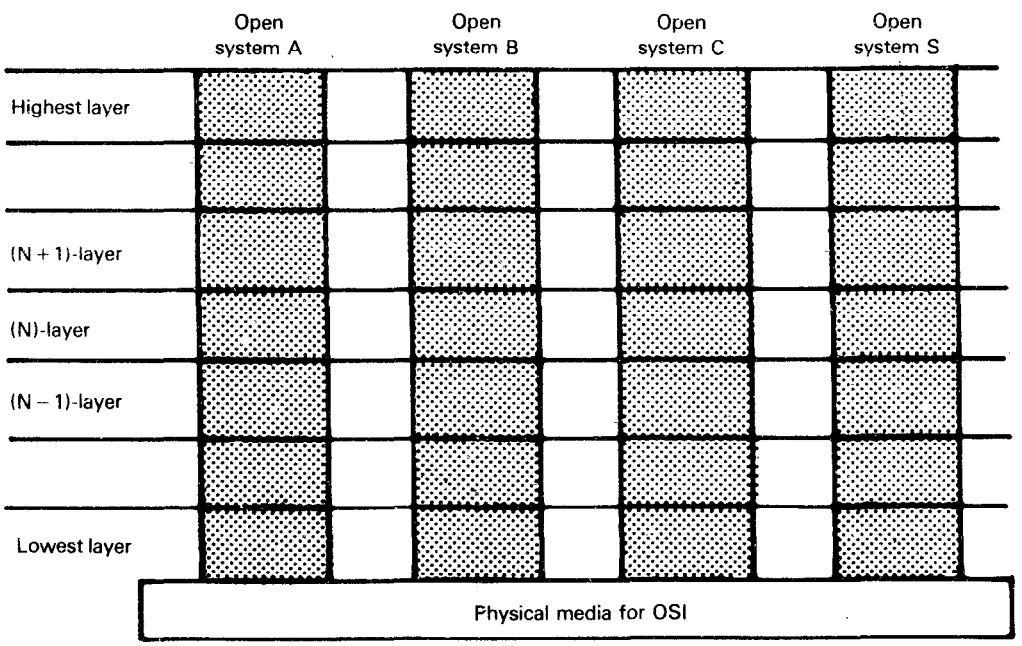


Figure 3 — Layering in co-operating open systems

Not all peer (N)-entities need or even can communicate. There may be conditions which prevent this communication (for example: they are not in interconnected open systems, or they do not support the same protocol subsets).

NOTES

1 The distinction between the type of some object and an instance of that object is a distinction of significance for OSI. A type is a description of a class of objects. An instance of this type is any object that conforms to this description. The instances of the same type constitute a class. A type, and any instances of this type can be referred to by an individual name. Each nameable instance and the type to which this instance belongs should carry distinguishable names.

For example, given that a programmer has written a computer program, that programmer has generated a type of something where instances of that are created every time that particular program is invoked into execution by a computer. Thus, a FORTRAN compiler is a type and each occasion where a copy of that program is invoked in a data processing machine one displays an instance.

Consider now an (N)-entity in the OSI context. It too, has two aspects, a type and a collection of instances. The type of an (N)-entity is defined by the specific set of (N)-layer functions it is able to perform. An instance of that type of (N)-entity is a specific invocation of whatever it is within the relevant open system that provides the (N)-layer functions called for by its type for a particular occasion of communication. It follows from these observations that (N)-entity types refer only to the properties of an association between peer (N)-entities, while an (N)-entity instance refers to the specific, dynamic occasions of actual information exchange.

It is important to note that actual communication occurs only between (N)-entity instances at all layers. It is only at connection establishment time (or its logical equivalent during a recovery process) that (N)-entity types are explicitly relevant. Actual connections are always made to specific (N)-entity instances, although a request for connection may well be made for arbitrary (N)-entity instances of a specified type. Nothing in this International Standard, however, precludes the request for a connection with a specific (named) instance of a peer (N)-entity. If an (N)-entity instance is aware of the name of its peer (N)-entity instance, it should be able to request another connection to that (N)-entity instance.

2 It may be necessary to further divide a layer into small substructures called sublayers and to extend the technique of layering to cover other dimensions of OSI. A sublayer is defined as a grouping of functions in a layer which may be bypassed. The bypassing of all the sublayers of a layer is not allowed. A sublayer uses the entities and connections of its layer. The detailed definition or additional characteristics of a sublayer are for further study.

Except for the highest layer, each (N)-layer provides (N + 1)-entities in the (N + 1)-layer with (N)-services. The highest layer is assumed to represent all possible uses of the services which are provided by the lower layers.

NOTES

1 Not all open systems provide the initial source or final destination of data. Such open systems need not contain the higher layers of the architecture (see figures 6 and 13).

2 Classes of service may be defined within the (N)-services. The precise definition of the term "classes of service" is for further study.

Each service provided by an (N)-layer may be tailored by the selection of one or more (N)-facilities which determine the attributes of that service. When a single (N)-entity cannot by itself fully support a service requested by an (N + 1)-entity it calls upon the co-operation of other (N)-entities to help complete the service request. In order to co-operate, (N)-entities in any layer, other than those in the lowest layer, communicate by means of the set of services provided by the (N – 1)-layer (see figure 4). The entities in the lowest layer are assumed to communicate directly via the physical media which connect them.

The services of an (N)-layer are provided to the (N + 1)-layer, using the (N)-functions performed within the (N)-layer and as necessary the services available from the (N – 1)-layer.

An (N)-entity may provide services to one or more (N + 1)-entities and use the services of one or more (N – 1)-entities. An (N)-service-access-point is the point at which a pair of entities in adjacent layers use or provide services (see figure 7).

Co-operation between (N)-entities is governed by one or more (N)-protocols. The entities and protocols within a layer are illustrated in figure 5.

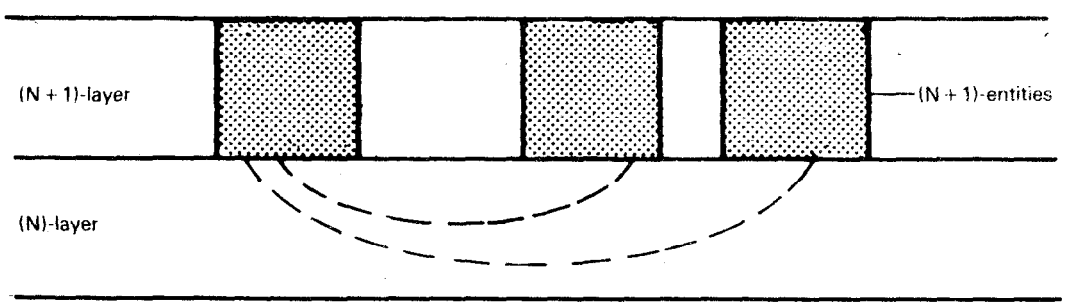


Figure 4 — (N + 1)-entities in the (N + 1)-layer communicate through the (N)-layer

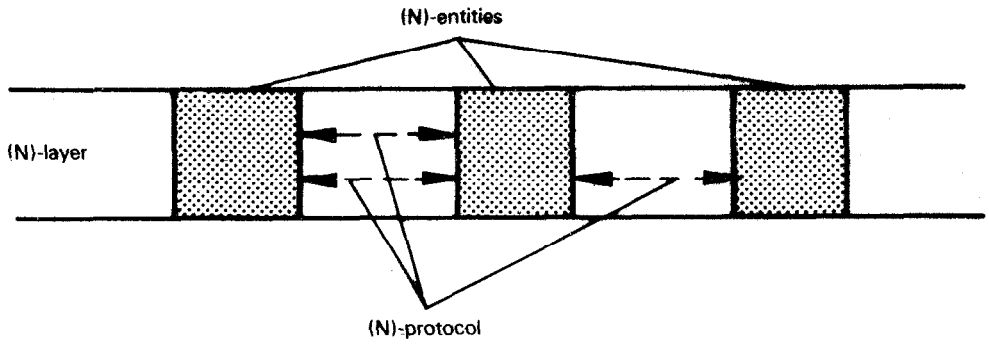


Figure 5 — (N)-protocols between (N)-entities

5.3 Communication between peer-entities

5.3.1 Definitions

5.3.1.1 (N)-connection : An association established by the (N)-layer between two or more (N + 1)-entities for the transfer of data.

5.3.1.2 (N)-connection-endpoint : A terminator at one end of an (N)-connection within an (N)-service-access-point.

5.3.1.3 multi-endpoint-connection : A connection with more than two connection-endpoints.

5.3.1.4 correspondent (N)-entities : (N)-entities with an (N − 1)-connection between them.

5.3.1.5 (N)-relay : An (N)-function by means of which an (N)-entity forwards data received from one correspondent (N)-entity to another correspondent (N)-entity.

5.3.1.6 (N)-data-source : An (N)-entity that sends (N − 1)-service-data-units (see 5.6.1.7) on an (N − 1)-connection.[1)]

5.3.1.7 (N)-data-sink : An (N)-entity that receives (N − 1)-service-data-units on an (N − 1)-connection.[1)]

5.3.1.8 (N)-data-transmission : An (N)-facility which conveys (N)-service-data-units from one (N + 1)-entity to one or more (N + 1)-entities.[1)]

5.3.1.9 (N)-duplex-transmission : (N)-data transmission in both directions at the same time.[1)]

5.3.1.10 (N)-half-duplex-transmission : (N)-data transmission in either direction, one direction at a time; the choice of direction is controlled by an (N + 1)-entity.[1)]

5.3.1.11 (N)-simplex-transmission : (N)-data-transmission in one pre-assigned direction.[1)]

5.3.1.12 (N)-data-communication : An (N)-function which transfers (N)-protocol-data-units (see 5.6.1.3) according to an (N)-protocol, over one or more (N − 1)-connections.[1)]

5.3.1.13 (N)-two-way-simultaneous communication : (N)-data-communication in both directions at the same time.

5.3.1.14 (N)-two-way-alternate communication : (N)-data-communication in both directions, one direction at a time.

5.3.1.15 (N)-one-way-communication : (N)-data-communication in one pre-assigned direction.

5.3.2 Description

For information to be exchanged between two or more (N + 1)-entities, an association shall be established between them in the (N)-layer using an (N)-protocol.

NOTE — Classes of protocols may be defined within the (N)-protocols. The precise definition of the term "classes of protocols" is for further study.

This association is called an (N)-connection. (N)-connections are provided by the (N)-layer between two or more (N)-service-access-points. The terminator of an (N)-connection at an (N)-service-access-point is called an (N)-connection-endpoint. A connection with more than two connection-endpoints is termed a multi-endpoint-connection. (N)-entities with a connection between them are termed correspondent (N)-entities.

(N + 1)-entities can communicate only by using the services of the (N)-layer. There are instances where services provided by the (N)-layer do not permit direct access between all of the (N + 1)-entities which have to communicate. If this is the case, communication can still occur if some other (N + 1)-entity can act as a relay between them (see figure 6).

The fact that communication is relayed by a chain of (N + 1)-entities is known neither by the (N)-layer nor by the (N + 2)-layer.

1) These definitions are not for use in this International Standard but are for use in future OSI standards.

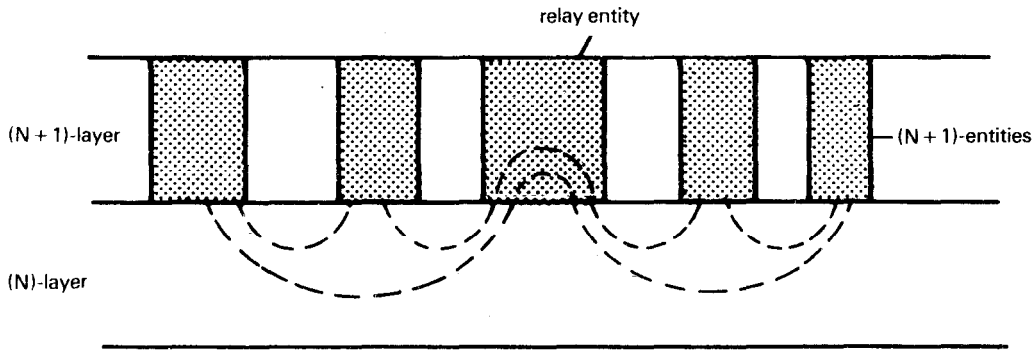


Figure 6 — Communication through a relay

5.4 Identifiers

5.4.1 Definitions

5.4.1.1 title : A permanent identifier for an entity.

5.4.1.2 title-domain : A subset of the title space of the OSI environment.

5.4.1.3 title-domain-name : An identifier which uniquely identifies a title-domain within the OSI environment.

NOTE — Title-domains of primary importance are the layers. In this specific case, the title-domain-name identifies the (N)-layer.

5.4.1.4 local-title : A title which is unique within a title-domain.

5.4.1.5 global-title : A title which is unique within the OSI environment and comprises two parts, a title-domain-name and a local-title.

5.4.1.6 (N)-address; (N)-service-access-point-address : An identifier which tells where an (N)-service-access-point may be found.

5.4.1.7 (N)-directory : An (N)-function by which the global title of an (N)-entity is translated into the (N – 1)-address of an (N – 1)-service-access-point to which the (N)-entity is attached.

5.4.1.8 (N)-address-mapping : An (N)-function which provides the mapping between the (N)-addresses and the (N – 1)-addresses associated with an (N)-entity.

5.4.1.9 routing : A function within a layer which translates the title of an entity or the service-access-point-address to which the entity is attached into a path by which the entity can be reached.

5.4.1.10 (N)-connection-endpoint-identifier : An identifier of an (N)-connection-endpoint which can be used to identify the corresponding (N)-connection at an (N)-service-access-point.

5.4.1.11 (N)-connection-endpoint-suffix : A part of an (N)-connection-endpoint-identifier which is unique within the scope of an (N)-service-access-point.

5.4.1.12 multi-connection-endpoint-identifier : An identifier which specifies the connection-endpoint of a multi-endpoint-connection which should accept the data that is being transferred.

5.4.1.13 (N)-service-connection-identifier : An identifier which uniquely specifies an (N)-connection within the environment of the correspondent (N + 1)-entities.

5.4.1.14 (N)-protocol-connection-identifier : An identifier which uniquely specifies an individual (N)-connection within the environment of the multiplexed (N – 1)-connection.

5.4.1.15 (N)-suffix : A part of an (N)-address which is unique within the (N)-service-access-point.

5.4.2 Description

An (N)-service-access-point-address, or (N)-address for short, identifies a particular (N)-service-access-point to which an (N + 1)-entity is attached (see figure 7). When the (N + 1)-entity is detached from the (N)-service-access-point, the (N)-address no longer provides access to the (N + 1)-entity. If the (N)-service-access-point is reattached to a different (N + 1)-entity, then the (N)-address identifies the new (N + 1)-entity and not the old one.

The use of an (N)-address to identify an (N + 1)-entity is the most efficient mechanism if the permanence of attachment between the (N + 1)-entity and the (N)-service-access-point can be assured. If there is a requirement to identify an (N + 1)-entity regardless of its current location, then the global-title assures correct identification.

An (N)-directory is an (N)-function which translates global-titles of peer (N)-entities into the (N – 1)-addresses through which they cooperate.

Interpretation of the correspondence between the (N)-addresses served by an (N)-entity and the (N – 1)-addresses used for accessing (N – 1)-services is performed by an (N)-address-mapping function.

Two particular kinds of (N)-address-mapping functions may exist within a layer :

a) hierarchical (N)-address-mapping; and

b) (N)-address-mapping by tables.

If an (N)-address is always mapped into only one (N – 1)-address then hierarchical construction of addresses can be used (see figure 8). The (N)-address-mapping function need only recognize the hierarchical structure of an (N)-address and extract the (N – 1)-address it contains.

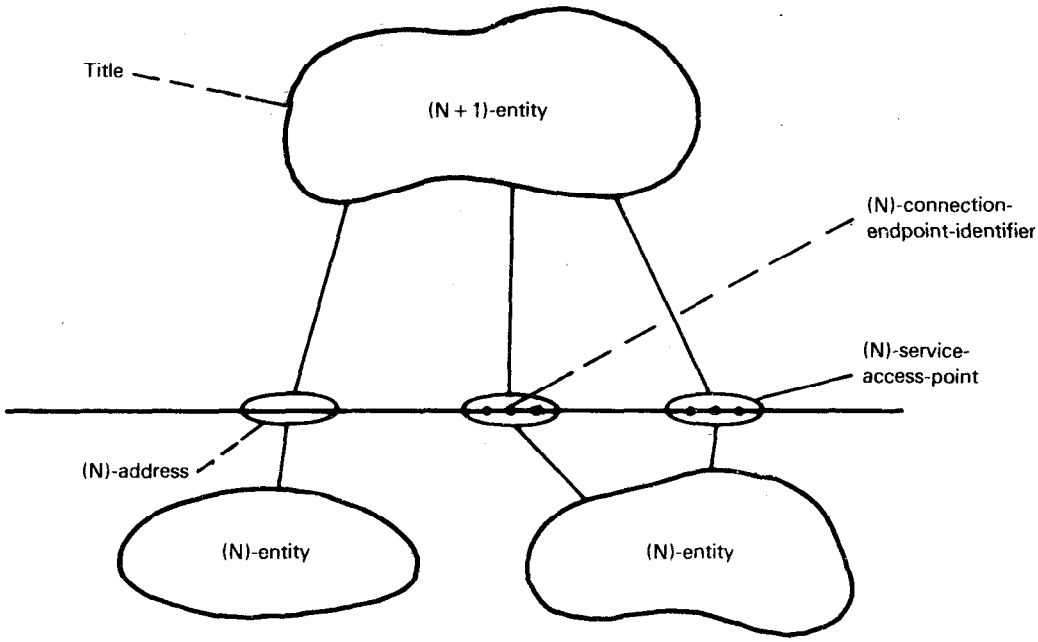
In this case, an (N)-address consists of two parts :

a) an (N – 1)-address of the (N)-entity which is supporting the current (N)-service-access-point of the (N + 1)-entity;

b) an (N)-suffix which makes the (N)-service-access-point uniquely identifiable within the scope of the (N – 1)-address.

Within a given layer, a hierarchical structure of addresses simplifies (N)-address-mapping functions because of the permanent nature of the mapping it presupposes. It is not imposed by the model in all layers in order to allow more flexibility in (N)-address-mappings and to cover the case where an (N)-entity attached to more than one (N – 1)-service-access-point supports only one (N)-service-access-point.

If the previous condition is not true, i.e. either an (N)-address can be mapped into several (N – 1)-addresses, or an (N)-address is not permanently mapped into the same (N – 1)-address, then hierarchical construction of an address is not possible and the (N)-address-mapping function may use tables to translate (N)-addresses into (N – 1)-addresses.



NOTE – Dashed arrows refer to identifiers.

Figure 7 — Entities, service-access-points, and identifiers

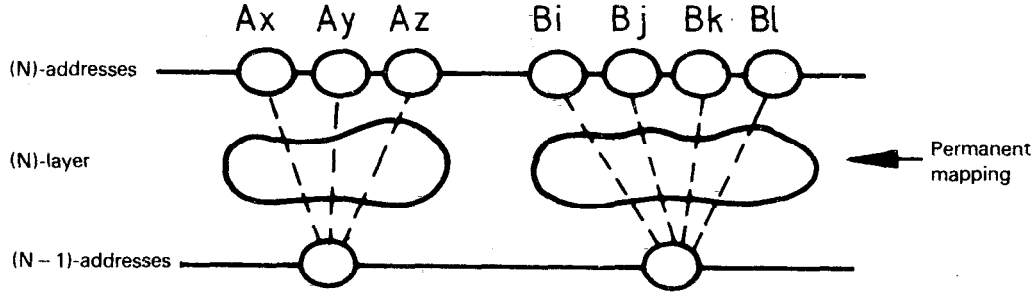


Figure 8 — Hierarchical (N)-address-mapping

The structure of an (N)-address is known by the (N)-entity which is attached to the identified (N)-service-access-point. However, the (N + 1)-entity does not know this structure.

If an (N + 1)-entity has two or more (N)-service-access-points with either the same (N)-entity or different (N)-entities, the (N)-entities have no knowledge of this fact. Each (N)-service-access-point is considered to identify a different (N + 1)-entity from the perspective of the (N)-entities.

A routing function translates the (N)-address of an (N + 1)-entity into a path or route by which the (N + 1)-entity may be reached.

An (N + 1)-entity may establish an (N)-connection with another (N + 1)-entity by using an (N)-service. When an (N + 1)-entity establishes an (N)-connection with another (N + 1)-entity, each (N + 1)-entity is given an (N)-connection-endpoint-identifier by its supporting (N)-entity. The (N + 1)-entity can then distinguish the new connection from all other (N)-connections accessible at the (N)-service-access-point it is using. This (N)-connection-endpoint-identifier shall be unique within the scope of the (N + 1)-entity which will use the (N)-connection.

The (N)-connection-endpoint-identifier consists of two parts :

a) the (N)-address of the (N)-service-access-point which will be used in conjunction with the (N)-connection; and

b) an (N)-connection-endpoint-suffix which is unique within the scope of the (N)-service-access-point.

A multi-endpoint-connection requires multi-connection-endpoint-identifiers. Each such identifier is used to specify which connection-endpoint should accept the data which is being transferred. A multi-connection-endpoint-identifier shall be unique within the scope of the connection within which it is used.

The (N)-layer may provide to the (N + 1)-entities an (N)-service-connection-identifier which uniquely specifies the (N)-connection within the environment of the correspondent (N + 1)-entities.

5.5 Properties of service-access-points

An (N + 1)-entity requests (N)-services via an (N)-service-access-point which permits the (N + 1)-entity to interact with an (N)-entity.

Both the (N)- and (N + 1)-entities attached to an (N)-service-access-point are in the same system.

An (N + 1)-entity may concurrently be attached to one or more (N)-service-access-points attached to the same or different (N)-entities.

An (N)-entity may concurrently be attached to one or more (N + 1)-entities through (N)-service-access-points.

An (N)-service-access-point is attached to only one (N)-entity and to only one (N + 1)-entity at a time.

An (N)-service-access-point may be detached from an (N + 1)-entity and reattached to the same or another (N + 1)-entity.

An (N)-service-access-point may be detached from an (N)-entity and reattached to the same or another (N)-entity.

An (N)-service-access-point is located by means of its (N)-address. An (N)-address is used by an (N + 1)-entity to request an (N)-connection.

5.6 Data-units

5.6.1 Definitions

5.6.1.1 (N)-protocol-control-information : Information exchanged between (N)-entities, using an (N − 1)-connection, to co-ordinate their joint operation.

5.6.1.2 (N)-user-data : The data transferred between (N)-entities on behalf of the (N + 1)-entities for whom the (N)-entities are providing services.

5.6.1.3 (N)-protocol-data-unit : A unit of data specified in an (N)-protocol and consisting of (N)-protocol-information and possibly (N)-user-data.

5.6.1.4 (N)-interface-control-information : Information transferred between an (N + 1)-entity and an (N)-entity to co-ordinate their joint operation.

5.6.1.5 (N)-interface-data : Information transferred from an (N + 1)-entity to an (N)-entity for transmission to a correspondent (N + 1)-entity over an (N)-connection, or conversely, information transferred from an (N)-entity to an (N + 1)-entity after being received over an (N)-connection from a correspondent (N + 1)-entity.

5.6.1.6 (N)-interface-data-unit : The unit of information transferred across the (N)-service-access-point between an (N + 1)-entity and an (N)-entity in a single interaction. Each (N)-interface-data-unit contains (N)-interface-control-information and may also contain the whole or part of an (N)-service-data-unit.

5.6.1.7 (N)-service-data-unit : An amount of (N)-interface-data whose identity is preserved from one end of an (N)-connection to the other.

5.6.1.8 expedited (N)-service-data-unit, (N)-expedited-data-unit : A small (N)-service-data-unit whose transfer is expedited. The (N)-layer ensures that an expedited-data-unit will not be delivered after any subsequent service-data-unit or expedited unit sent on that connection.

5.6.2 Description

Information is transferred in various types of data-units between peer-entities and between entities attached to a specific service-access-point. The data-units are defined in 5.6.1 and the relationships among them are illustrated in figures 9 and 10.

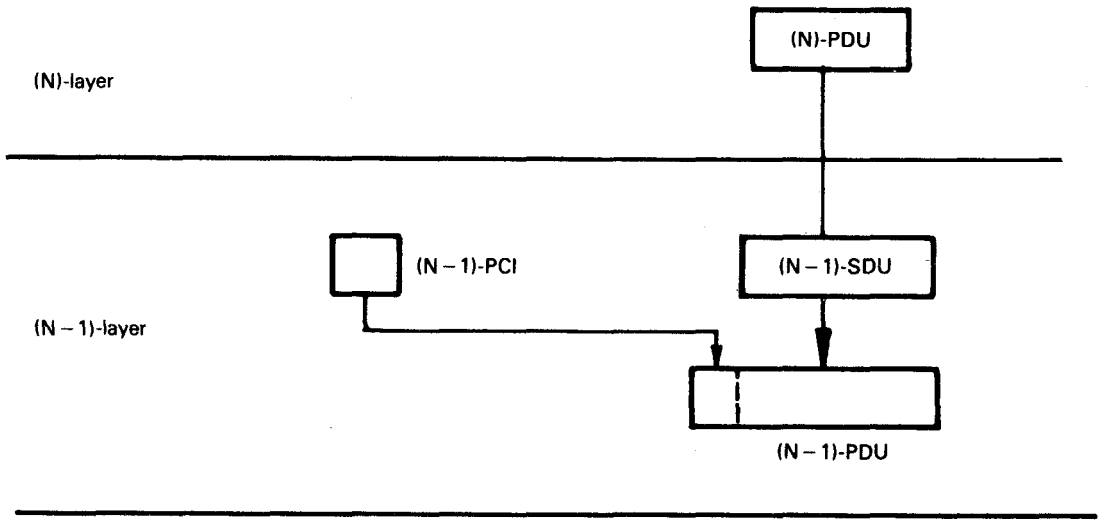
Except for the relationships defined in figures 9 and 10, there is no overall architectural limit to the size of data-units. There may be other size limitations at specific layers.

The size of (N)-interface-data-units is not necessarily the same at each end of the connection.

Data may be held within a connection until a complete service-data-unit is put into the connection.

	Control	Data	Combined
(N) – (N) peer-entities	(N)-protocol-control-information	(N)-user-data	(N)-protocol-data-units
(N + 1) – (N) adjacent layers	(N)-interface-control-information	(N)-interface-data	(N)-interface-data-unit

Figure 9 — Relationships among data-units



PCI = protocol-control-information
PDU = protocol-data-unit
SDU = service-data-unit

NOTES

1 This figure assumes that neither segmenting nor blocking of (N)-service-data-units is performed (see 5.7.6.5).

2 This figure does not imply any positional relationship between protocol-control-information and user-data in protocol-data-units.

3 An (N)-protocol-data-unit may be mapped one-to-one into an (N – 1)-service-data-unit, but other relationships are possible (see figure 11).

Figure 10 — An illustration of mapping between data-units in adjacent layers

5.7 Elements of layer operation

5.7.1 Definitions

5.7.1.1 (N)-protocol-identifier : An identifier used between correspondent (N)-entities to select a specific (N)-protocol to be used on a particular (N – 1)-connection.

5.7.1.2 centralized multi-endpoint-connection : A multi-endpoint-connection where data sent by the entity associated with the central connection-endpoint is received by all other entities, while data sent by one of the other entities is only received by the central entity.

5.7.1.3 decentralized multi-endpoint-connection : A multi-endpoint-connection such that data sent by an entity associated with a connection-endpoint is received by all other entities.

5.7.1.4 multiplexing : A function within the (N)-layer by which one (N – 1)-connection is used to support more than one (N)-connection.

NOTE – The term multiplexing is also used in a more restricted sense to refer to the function performed by the sending (N)-entity while the term demultiplexing is used to refer to the function performed by the receiving (N)-entity.

5.7.1.5 demultiplexing : The function performed by an (N)-entity which identifies (N)-protocol-data-units for more than one (N)-connection within (N – 1)-service-data-units received on a single (N – 1)-connection. It is the reverse function of the multiplexing function performed by the (N)-entity sending the (N – 1)-service-data-units.

5.7.1.6 splitting : A function within the (N)-layer by which more than one (N – 1)-connection is used to support one (N)-connection.

NOTE – The term splitting is also used in a more restrictive sense to refer to the function performed by the sending (N)-entity while the term recombining is used to refer to the function performed by the receiving (N)-entity.

5.7.1.7 recombining : The function performed by an (N)-entity which identifies (N)-protocol-data-units for a single (N)-connection in (N – 1)-service-data-units received on more than one (N – 1)-connection. It is the reverse function of the splitting function performed by the (N)-entity sending the (N – 1)-service-data-units.

5.7.1.8 flow control : A function which controls the flow of data within a layer or between adjacent layers.

5.7.1.9 segmenting : A function performed by an (N)-entity to map one (N)-service-data-unit into multiple (N)-protocol-data-units.

5.7.1.10 reassembling : A function performed by an (N)-entity to map multiple (N)-protocol-data-units into one (N)-service-data-unit. It is the reverse function of segmenting.

5.7.1.11 blocking : A function performed by an (N)-entity to map multiple (N)-service-data-units into one (N)-protocol-data-unit.

5.7.1.12 deblocking : A function performed by an (N)-entity to identify multiple (N)-service-data-units which are contained in one (N)-protocol-data-unit. It is the reverse function of blocking.

5.7.1.13 concatenation : A function performed by an (N)-entity to map multiple (N)-protocol-data-units into one (N – 1)-service-data-unit.

5.7.1.14 separation : A function performed by an (N)-entity to identify multiple (N)-protocol-data-units which are contained in one (N – 1)-service-data-unit. It is the reverse function of concatenation.

5.7.1.15 sequencing : A function performed by the (N)-layer to preserve the order of (N)-service-data-units that were submitted to the (N)-layer.

5.7.1.16 acknowledgement : A function of the (N)-layer which allows a receiving (N)-entity to inform a sending (N)-entity of the receipt of an (N)-protocol-data-unit.

5.7.1.17 reset : A function which sets the correspondent (N)-entities to a predefined state with a possible loss or duplication of data.

NOTE – Blocking and concatenation, though close to each other (they both permit grouping of data-units) are different (see 5.7.1.11 and 5.7.1.13). These two functions may serve different purposes. For instance, concatenation permits the (N)-layer to group one or several acknowledgment (N)-PDUs with one (or several) (N)-PDUs containing user-data, and this would not be possible with the blocking function only. Also the two functions may be combined so that the (N)-layer performs blocking and concatenation.

5.7.2 Protocol selection

One or more (N)-protocols may be defined for the (N)-layer. An (N)-entity may employ one or more (N)-protocols.

Meaningful communication between (N)-entities over an (N – 1)-connection requires the agreed selection of one (N)-protocol.

(N)-protocol-identifiers name the specific protocols defined.

5.7.3 Properties of connections

An (N)-connection is an association established for communication between two or more (N + 1)-entities, identified by their (N)-addresses. An (N)-connection is offered as a service by the (N)-layer, so that information may be exchanged between the (N + 1)-entities.

An (N + 1)-entity may have, simultaneously, one or more (N)-connections with other (N + 1)-entities, with any given (N + 1)-entity, and with itself.

An (N)-connection is established by referencing, either explicitly or implicitly, an (N)-address for the source (N + 1)-entity and an (N)-address for each of one or more destination (N + 1)-entities.

The source (N)-address and one or more of the destination (N)-addresses may be the same. One or more of the destination (N)-addresses may be the same while the source (N)-address is different. All may be different.

One (N)-connection-endpoint is constructed for each (N)-address referenced explicitly or implicitly when an (N)-connection is established.

An (N + 1)-entity accesses an (N)-connection via an (N)-service-access-point.

An (N)-connection has two or more (N)-connection-endpoints.

An (N)-connection-endpoint is not shared by (N + 1)-entities or (N)-connections.

An (N)-connection-endpoint relates three elements :

a) an (N + 1)-entity;

b) an (N)-entity; and

c) an (N)-connection.

The (N)-entity and the (N + 1)-entity related by an (N)-connection-endpoint are those implied by the (N)-address referenced when the (N)-connection is established.

An (N)-connection-endpoint has an identifier, called an (N)-connection-endpoint-identifier, which is unique within the scope of the (N + 1)-entity which is bound to the (N)-connection-endpoint.

An (N)-connection-endpoint-identifier is not the same as an (N)-address.

An (N + 1)-entity references an (N)-connection using its (N)-connection-endpoint-identifier.

Multi-endpoint-connections are connections which have three or more connection-endpoints. Two types of multi-endpoint-connection are defined[1] :

a) centralized; and

b) decentralized.

A centralized multi-endpoint-connection has a central connection-endpoint. Data sent by the entity associated with the central connection-endpoint is received by the entities associated with all other connection-endpoints. The data sent by an entity associated with any other connection-endpoint is received only by the entity associated with the central connection-endpoint.

On a decentralized multi-endpoint-connection, data sent by an entity associated with any connection-endpoint is received by the entities associated with all of the other connection-endpoints.

5.7.4 Connection establishment and release

The establishment of an (N)-connection by peer-entities of an (N)-layer requires the following :

a) the availability of an (N − 1)-connection between the supporting (N)-entities; and

b) both (N)-entities be in a state in which they can execute the connection establishment protocol exchange.

If it is not already available, an (N − 1)-connection has to be established by peer-entities of the (N − 1)-layer. This requires, for the (N − 1)-layer, the same conditions as described above for the (N)-layer.

The same consideration applies downwards until either an available connection or the physical medium for OSI is encountered.

Depending upon the characteristics of the (N − 1)-service and of the establishment protocol exchange, the establishment of an (N)-connection may or may not be done in conjunction with the establishment of the (N − 1)-connection.

The characteristics of the (N)-service with regard to the establishment of the (N)-connection vary depending upon whether or not (N)-user-data can be transferred by the connection establishment protocol exchange for each direction of the (N)-connection.

Where (N)-user-data is transferred by the (N)-connection establishment protocol exchange, the (N + 1)-protocol may take advantage of this to allow an (N + 1)-connection to be established in conjunction with the establishment of the (N)-connection.

The release of an (N)-connection is normally initiated by one of the (N + 1)-entities associated in it.

The release of an (N)-connection may also be initiated by one of the (N)-entities supporting it as a result of an exception condition occuring in the (N)-layer or the layers below.

Depending upon the conditions, release of an (N)-connection may result in the discarding of (N)-user-data.

The orderly release of an (N)-connection requires either the availability of an (N − 1)-connection, or a common reference to time [for example time of failure of the (N − 1)-connection and common time-out]. In addition, both (N)-entities shall be in a state in which they can execute the connection release protocol exchange. It is important to note, however, that the release of an (N − 1)-connection does not necessarily cause the release of the (N)-connection(s) which were using it; the (N − 1)-connection can be reestablished, or another (N − 1)-connection substituted.

1) Other types of multi-endpoint-connections are for further study.

The characteristics of the (N)-service with regard to the release of an (N)-connection can be of two kinds :

a) (N)-connections are either released immediately when the release protocol exchange is initiated [(N)-user-data not yet delivered may be discarded]; or

b) release is delayed until all (N)-user-data sent previous to the initiation of the release protocol exchange has been delivered (i.e. delivery confirmation has been received).

(N)-user-data may be transferred by the connection release protocol exchange.

Some (N)-protocols may provide for the combining of connection establishment and connection release protocol exchanges.

5.7.5 Multiplexing and splitting

Within the (N)-layer, (N)-connections are mapped onto (N – 1)-connections. The mapping may be one of three kinds :

a) one-to-one;

b) many (N)-connections to one (N – 1)-connection (multiplexing); and

c) one (N)-connection to many (N – 1)-connections (splitting).

Multiplexing may be needed in order to:

a) make more efficient or more economic use of the (N – 1)-service; and

b) provide several (N)-connections in an environment where only a single (N – 1)-connection exists.

Splitting may be needed in order to:

a) improve reliability since more than one (N – 1)-connection is available;

b) provide the required grade of performance, through the utilization of multiple (N – 1)-connections; and

c) obtain cost benefits by the utilization of multiple low cost (N – 1)-connections each with less than the required grade of performance.

Multiplexing and splitting each involve a number of associated functions which may not be needed for one-to-one connection mapping.

The functions associated with multiplexing are:

a) identification of the (N)-connection for each (N)-protocol-data-unit transferred over the (N – 1)-connection, in order to ensure that (N)-user-data from the various multiplexed (N)-connections are not mixed. This identification is distinct from that of the (N)-connection-endpoint-identifiers and is called an (N)-protocol-connection-identifier;

b) flow control on each (N)-connection in order to share the capacity of the (N – 1)-connection (see 5.7.6.4); and

c) scheduling the next (N)-connection to be serviced over the (N – 1)-connection when more than one (N)-connection is prepared to send data.

The functions associated with splitting are:

a) scheduling the utilization of multiple (N – 1)-connections used in splitting a single (N)-connection; and

b) resequencing of (N)-protocol-data-units associated with an (N)-connection since they may arrive out of sequence even when each (N – 1)-connection guarantees sequence of delivery (see 5.7.6.6).

5.7.6 Transfer of data

5.7.6.1 Normal data transfer

Control information and user data are transferred between (N)-entities in (N)-protocol-data-units. An (N)-protocol-data-unit is a unit of data specified in an (N)-protocol and contains (N)-protocol-control-information and possibly (N)-user-data.

(N)-protocol-control-information is transferred between (N)-entities using the (N – 1)-connection. (N)-protocol-control-information is any information that supports the joint operation of (N)-entities. (N)-user-data is passed transparently between (N)-entities over an (N – 1)-connection.

An (N)-protocol-data-unit has an arbitrary, but finite, size. (N)-protocol-data-units are mapped into (N – 1)-service-data-units. The interpretation of an (N)-protocol-data-unit is defined by the (N)-protocol in effect for the (N – 1)-connection.

An (N)-service-data-unit is transferred between an (N + 1)-entity and an (N)-entity, through an (N)-service-access-point, in the form of one or more (N)-interface-data-units. The (N)-service-data-unit is transferred as (N)-user-data in one or more (N)-protocol-data-units.

The exchange of data under the rules of an (N)-protocol can only occur if an (N – 1)-connection exists. If an (N – 1)-connection does not exist, it shall be established before an exchange of data can occur (see 5.7.4).

5.7.6.2 Data transfer during connection establishment and release

(N)-user-data may be transferred in the (N)-connection establishment protocol exchange and in the (N)-connection release protocol exchange.

The connection release protocol exchange may be combined with the connection establishment protocol exchange (see 5.7.4) to provide means for the delivery of a single unit of (N)-user-data between correspondent (N + 1)-entities with a confirmation of receipt.

5.7.6.3 Expedited transfer of data

An expedited-data-unit is a service-data-unit which is transferred and/or processed with priority over normal service-data-units. An expedited data transfer service may be used for signalling and interrupt purposes.

Expedited data flow is independent of the states and operation of the normal data flow, although the data sent on the two flows may be logically related. Conceptually, a connection that supports expedited flow can be viewed as having two subchannels, one for normal data, the other for expedited data. Data sent on the expedited channel is assumed to be given priority over normal data.

The transfer guarantees that an expedited-data-unit will not be delivered after any subsequent normal service-data-unit or expedited-data-unit sent on the connection.

Because the expedited flow is assumed to be used to transfer small amounts of data infrequently, simplified flow control mechanisms may be used on this data flow.

An expedited (N)-service-data-unit is intended to be processed by the receiving (N + 1)-entity with priority over normal (N)-service-data-units.

5.7.6.4 Flow control

If flow control functions are provided, they can operate only on protocol-data-units and interface-data-units.

Two types of flow control are identified :

a) peer flow control which regulates the rate at which (N)-protocol-data-units are sent to the peer (N)-entity on the (N)-connection. Peer flow control requires protocol definitions and is based on protocol-data-unit size; and

b) (N)-interface flow control which regulates the rate at which (N)-interface-data are passed between an (N + 1)-entity and an (N)-entity. (N)-interface flow control is based on the (N)-interface-data-unit size.

Multiplexing in a layer may require a peer flow control function for individual flows (see 5.7.5).

Peer flow control functions require flow control information to be included in the (N)-protocol-control-information of an (N)-protocol-data-unit.

If the size of service-data-units exceeds the maximum size of the (N)-user-data portion of an (N)-protocol-data-unit, then first segmentation shall be performed on the (N)-service-data-unit to make it fit within the (N)-protocol-data-units. Peer flow control can then be applied on the (N)-protocol-data-units.

5.7.6.5 Segmenting, blocking and concatenation

Data-units in the various layers are not necessarily of compatible size. It may be necessary to perform segmenting, i.e. to map an (N)-service-data-unit into more than one (N)-protocol-data-units. Similarly, segmenting may occur when (N)-protocol-data-units are mapped into (N – 1)-interface-data-units. Since it is necessary to preserve the identity of (N)-service-data-units on an (N)-connection, functions shall be available to identify the segments of an (N)-service-data-unit, and to allow the correspondent (N)-entities to reassemble the (N)-service-data-unit.

Segmenting may require that information be included in the (N)-protocol-control-information of an (N)-protocol-data-unit. Within a layer, (N)-protocol-control-information is added to an (N)-service-data-unit to form an (N)-protocol-data-unit when no segmenting or blocking is performed [see figure 11a)]. If segmenting is performed, an (N)-service-data-unit is mapped into several (N)-protocol-data-units with added (N)-protocol-control-information [see figure 11b)].

Conversely, it may be necessary to perform blocking whereby several (N)-service-data-units with added (N)-protocol-control-information form an (N)-protocol-data-unit [see figure 11c)].

The Reference Model also permits concatenation whereby several (N)-protocol-data-units are concatenated into a single (N – 1)-service-data-unit [see figure 11d)].

5.7.6.6 Sequencing

The (N – 1)-services provided by the (N – 1)-layer of the OSI architecture may not guarantee delivery of data in the same order as it was submitted by the (N)-layer. If the (N)-layer needs to preserve the order of data transferred through the (N – 1)-layer, sequencing mechanisms shall be present in the (N)-layer. Sequencing may require additional (N)-protocol-control-information.

5.7.7 Error functions

5.7.7.1 Acknowledgement

An acknowledgement function may be used by peer (N)-entities using an (N)-protocol to obtain a higher probability of detecting protocol-data-unit loss than is provided by the (N – 1)-layer. Each (N)-protocol-data-unit transferred between correspondent (N)-entities is made uniquely identifiable, so that the receiver can inform the sender of the receipt of the (N)-protocol-data-unit. An acknowledgement function is also able to infer the non-receipt of (N)-protocol-data-units and take appropriate remedial action.

An acknowledgement function may require that information be included in the (N)-protocol-control-information of (N)-protocol-data-units.

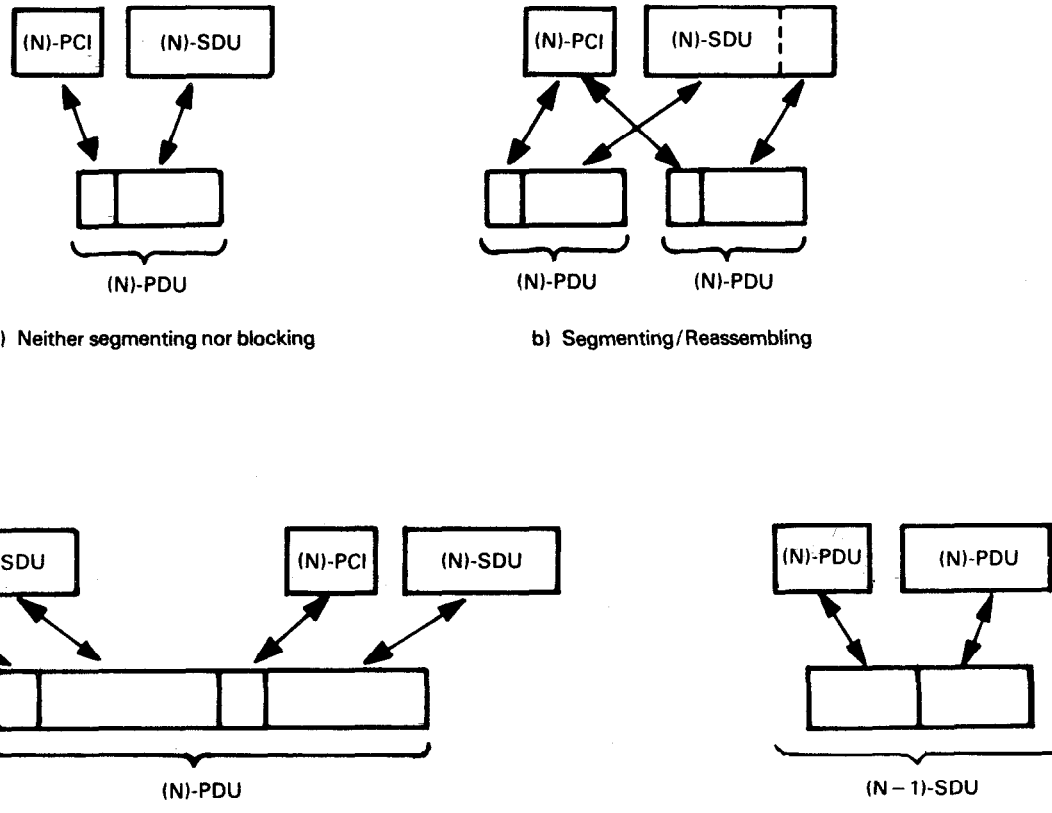
The scheme for uniquely identifying (N)-protocol-data-units may also be used to support other functions such as detection of duplicate data-units, segmenting and sequencing.

NOTE – Other forms of acknowledgement such as confirmation of delivery and confirmation of performance of an action are for further study.

5.7.7.2 Error detection and notification

Error detection and notification functions may be used by an (N)-protocol to provide a higher probability of both protocol-data-unit error detection and data corruption detection than is provided by the (N – 1)-service.

Error detection and notification may require that additional information be included in the (N)-protocol-control-information of the (N)-protocol-data-unit.



a) Neither segmenting nor blocking

b) Segmenting/Reassembling

c) Blocking/Deblocking

d) Concatenation/Separation

SDU = service-data-unit
PCI = protocol-control-information
PDU = protocol-data-unit

NOTES

1 This figure does not imply any positional relationship between protocol-control-information and user-data in protocol-data-units.

2 In the case of concatenation, (N)-protocol-data-unit does not necessarily include an (N)-service-data-unit.

Figure 11 — Relationship between (N)-service-data-unit, (N)-protocol-data-unit and (N − 1)-service-data-unit within a layer

5.7.7.3 Reset

Some services require a reset function to recover from a loss of synchronization between correspondent (N)-entities. A reset function sets the correspondent (N)-entities to a predefined state with a possible loss or duplication of data.

NOTE — Additional functions may be required to determine at what point reliable data transfer was interrupted.

A quantity of (N)-user-data may be conveyed in association with the (N)-reset function.

The reset function may require that information be included in the (N)-protocol-control-information of the (N)-protocol-data-unit.

5.8 Routing

A routing function within the (N)-layer enables communication to be relayed by a chain of (N)-entities. The fact that communication is being routed by intermediate (N)-entities is known by neither the lower layers nor by the higher layers. An (N)-entity which participates in a routing function may have a routing table.

5.9 Management aspects of OSI

5.9.1 Definitions

5.9.1.1 application-management : Functions in the application layer (see 6.1) related to the management of OSI application-processes.

5.9.1.2 application-management-application-entity : An application-entity which executes application-management functions.

5.9.1.3 OSI resources : Data processing and data communication resources which are of concern to OSI.

5.9.1.4 systems-management : Functions in the Application Layer related to the management of various OSI resources and their status across all layers of the OSI architecture.

5.9.1.5 systems-management-application-entity : An application-entity which executes systems-management functions.

5.9.1.6 layer-management : Functions related to the management of the (N)-layer partly performed in the (N)-layer itself according to the (N)-protocol of the layer (activities such as activation and error control) and partly performed as a subset of systems-management.

5.9.2 Introduction

Within the OSI architecture there is a need to recognize the special problems of initiating, terminating, and monitoring activities and assisting in their harmonious operations, as well as handling abnormal conditions. These have been collectively considered as the management aspects of the OSI architecture. These concepts are essential to the operation of the interconnected open systems and therefore are included in the comprehensive description of the Reference Model described in subsequent clauses of this International Standard.

The management activities which are of concern are those which imply actual exchanges of information between open systems. Only the protocols needed to conduct such exchanges are candidates for standardization within the OSI architecture.

This clause describes key concepts relevant to the management aspects, including the different categories of management activities and the positioning of such activities within the OSI architecture.

5.9.3 Categories of management activities

Only those management activities which imply actual exchanges of information between remote management entities are pertinent to the OSI architecture. Other management activities local to particular open systems are outside its scope.

Similarly, not all resources are pertinent to OSI. This International Standard considers only OSI resources, i.e. those data processing and data communication resources which are of concern to OSI.

The following categories of management activities are identified :

a) application-management;

b) systems-management; and

c) layer-management.

5.9.3.1 Application-management

Application-management relates to the management of OSI application-processes. The following list is typical of activities which fall into this category but it is not exhaustive :

a) initialization of parameters representing application-processes;

b) initiation, maintenance and termination of application-processes;

c) allocation and de-allocation of OSI resources to application-processes;

d) detection and prevention of OSI resource interference and deadlock;

e) integrity and commitment control;

f) security control; and

g) checkpointing and recovery control.

The protocols for application-management reside within the Application Layer, and are handled by application-management-application-entities.

5.9.3.2 Systems-management

Systems-management relates to the management of OSI resources and their status across all layers of the OSI architecture. The following list is typical of activities which fall into this category but it is not exhaustive :

a) activation/deactivation management which includes :

1) activation, maintenance and termination of OSI resources distributed in open systems, including physical media for OSI;

2) some program loading functions;

3) establishment/maintenance/release of connections between management entities; and

4) open systems parameter initialization/modification;

b) monitoring which includes :

1) reporting status or status changes; and

2) reporting statistics; and

c) error control which includes :

1) error detection and some of the diagnostic functions; and

2) reconfiguration and restart.

The protocols for systems-management reside in the application layer, and are handled by systems-management-application-entities.

5.9.3.3 Layer-management

There are two aspects of layer-management. One of these is concerned with layer activities such as activation and error control. This aspect is implemented by the layer protocol to which it applies.

The other aspect of layer-management is a subset of systems-management. The protocols for these activities reside within the Application Layer and are handled by systems-management-application-entities.

5.9.4 Principles for positioning management functions

Several principles are important in positioning management functions in the Reference Model of Open Systems Interconnection. They include the following[1] :

a) both centralization and decentralization of management functions are allowed. Thus, the OSI architecutre does not dictate any particular fashion or degree of centralization of such functions. This principle calls for a structure in which each open system is allowed to include any (subset of) systems-management functions and each subsystem is allowed to include any (subset of) layer-management functions;

b) if it is necessary, connections between management entities are established when an open system which has been operating in isolation from other open systems, becomes part of the OSI environment.

6 Introduction to the specific OSI layers

6.1 Specific layers

The general structure of the OSI architecture described in clause 5 provides architectural concepts from which the Reference Model of Open Systems Interconnection has been derived, making specific choices for the layers and their contents.

The Reference Model contains seven layers :

a) the Application Layer (layer 7);

b) the Presentation Layer (layer 6);

c) the Session Layer (layer 5);

d) the Transport Layer (layer 4);

e) the Network Layer (layer 3);

f) the Data Link Layer (layer 2); and

g) the Physical Layer (layer 1).

These layers are illustrated in figure 12. The highest is the Application Layer and it consists of the application-entities that co-operate in the OSI environment. The lower layers provide the services through which the application-entities co-operate.

Layers 1 to 6, together with the physical media for OSI provide a step-by-step enhancement of communication services. The boundary between two layers identifies a stage in this enhancement of services at which an OSI service standard is defined, while the functioning of the layers is governed by OSI protocol standards.

Not all open systems provide the initial source or final destination of data. When the physical media for OSI do not link all open systems directly, some open systems act only as relay open systems, passing data to other open systems. The functions and protocols which support the forwarding of data are then provided in the lower layers. This is illustrated in figure 13.

1) Other principles are for further study.

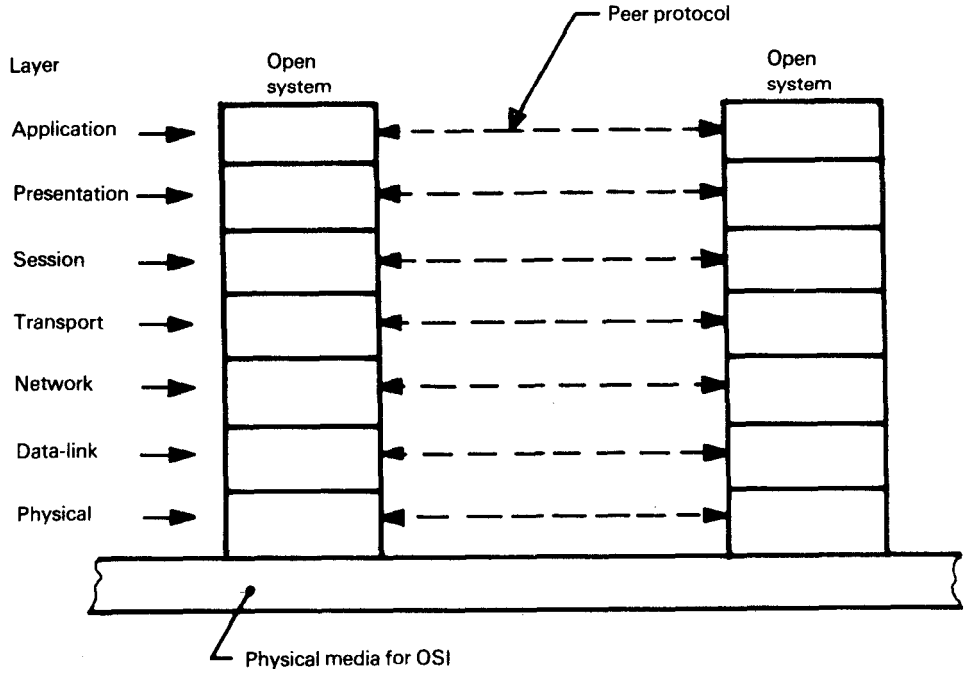


Figure 12 — Seven layer reference model and peer protocols

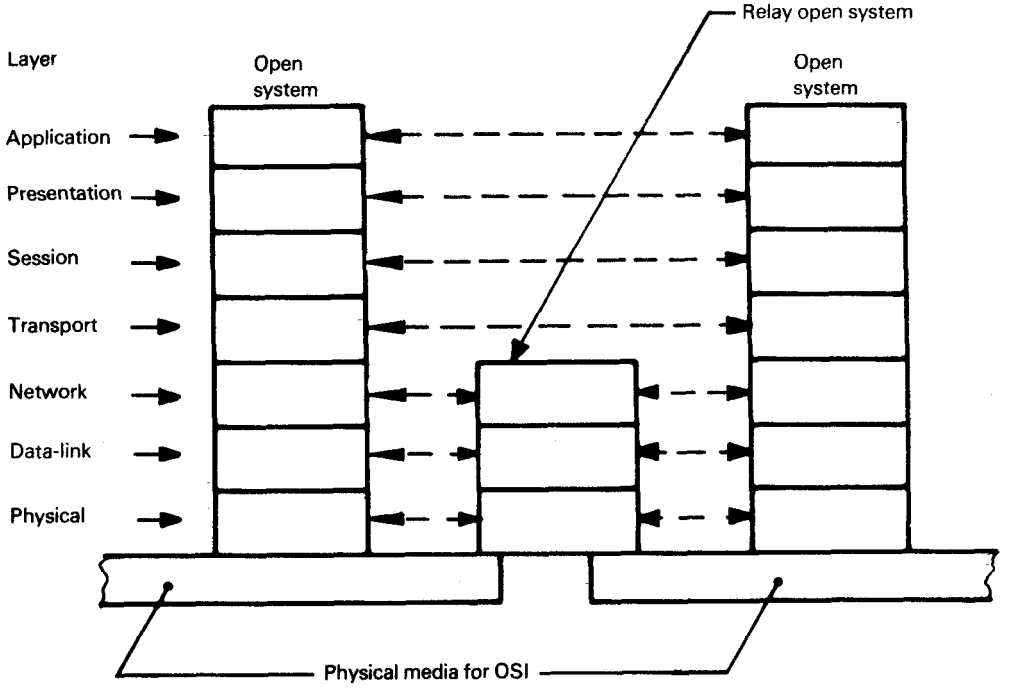


Figure 13 — Communication involving relay open systems

6.2 The principles used to determine the seven layers in the Reference Model

The following principles have been used to determine the seven layers in the Reference Model and are felt to be useful for guiding further decisions in the development of OSI standards,

NOTE – It may be difficult to prove that any particular layering selected is the best possible solution. However, there are general principles which can be applied to the question of where a boundary should be placed and how many boundaries should be placed.

a) Do not create so many layers as to make the system engineering task of describing and integrating the layers more difficult than necessary;

b) Create a boundary at a point where the description of services can be small and the number of interactions across the boundary are minimized;

c) Create separate layers to handle functions that are manifestly different in the process performed or the technology involved;

d) Collect similar functions into the same layer;

e) Select boundaries at a point which past experience has demonstrated to be successful;

f) Create a layer of easily localized functions so that the layer could be totally redesigned and its protocols changed in a major way to take advantage of new advances in architectural, hardware or software technology without changing the services expected from and provided to the adjacent layers;

g) Create a boundary where it may be useful at some point in time to have the corresponding interface standardized;

NOTES

1 Advantages and drawbacks of standardizing internal interfaces within open systems are not considered in this International Standard. In particular, mention of, or reference to principle g), should not be taken to imply usefulness of standards for such internal interfaces.

2 It is important to note that OSI *per se* does not require interfaces within open systems to be standardized. Moreover, whenever standards for such interfaces are defined, adherence to such internal interface standards can in no way be considered as a condition of openness.

h) Create a layer where there is a need for a different level of abstraction in the handling of data, for example morphology, syntax, semantics;

j) Allow changes of functions or protocols to be made within a layer without affecting other layers; and

k) Create for each layer boundaries with its upper and lower layer only.

Similar principles have been applied to sublayering :

m) Create further subgrouping and organization of functions to form sublayers within a layer in cases where distinct communication services need it;

n) Create, where needed, two or more sublayers with a common, and therefore minimal functionality to allow interface operation with adjacent layers; and

p) Allow by-passing of sublayers.

6.3 Layer descriptions

For each of the seven layers identified above, clause 7 provides :

a) an outline of the purpose of the layer;

b) a description of the services offered by the layer to the layer above; and

c) a description of the functions provided in the layer and the use made of the services provided by the layer below.

The descriptions, by themselves, do not provide a complete definition of the services and protocols for each layer. These are the subject of separate standards.

7 Detailed description of the resulting OSI architecture

7.1 Application Layer

7.1.1 Definition

7.1.1.1 application-entity : The aspects of an application-process pertinent to OSI

7.1.1.2 application-service-element: That part of an application-entity which provides an OSI environment capability, using underlying services when appropriate.

7.1.1.3 user-element: The representation of that part of the application-process which uses those application-service-elements needed to accomplish the communications objectives of that application-process.

7.1.2 Purpose

As the highest layer in the Reference Model of Open Systems Interconnection, the Application Layer provides a means for the application-processes to access the OSI environment. Hence the Application Layer does not interface with a higher layer. The Application Layer is the sole means for the application-process to access the OSI environment.

The purpose of the Application Layer is to serve as the window between correspondent application-processes which are using the OSI to exchange meaningful information.

Each application-process is represented to its peer by the application-entity.

All specifiable application-process parameters of each OSI environment communications instance are made known to the OSI environment (and, thus, to the mechanisms implementing the OSI environment) via the Application Layer.

7.1.3 Services provided to application-processes

Application-processes exchange information by means of application-entities, application-protocols, and presentation-services.

As the only layer in the Reference Model that directly provides services to the application-processes, the Application Layer necessarily provides all OSI services directly usable by application-processes.

The application-entity contains one user-element and a set of application-service-elements. The user-element represents that part of the application-process which uses those application-service-elements needed to accomplish the communications objectives of that application-process. Application-service-elements may call upon each other and/or upon presentation services to perform their function.

The only means by which user-elements in different systems may communicate is through the exchange of application-protocol-data-units. These application-protocol-data-units are generated by application-service-elements.

NOTE — The application services differ from services provided by other layers in neither being provided to an upper layer nor being associated with a service-access-point.

In addition to information transfer, such services may include, but are not limited to the following :

NOTE — Some of the services listed below are provided by OSI management.

a) identification of intended communications partners (for example by name, by address, by definite description, by generic description);

b) determination of the current availability of the intended communication partners;

c) establishment of authority to communicate;

d) agreement on privacy mechanisms;

e) authentication of intended communication partners;

f) determination of cost allocation methodology;

g) determination of the adequacy of resources;

h) determination of the acceptable quality of service (for example response time, tolerable error rate, cost vis-a-vis the previous considerations);

j) synchronization of cooperating applications;

k) selection of the dialogue discipline including the initiation and release procedures;

m) agreement on responsibility for error recovery;

n) agreement on the procedures for control of data integrity; and

p) identification of constraints on data syntax (character sets, data structure).

7.1.4 Functions within the Application Layer

The Application Layer contains all functions which imply communication between open systems and are not already performed by the lower layers. These include functions performed by programs as well as functions performed by human beings.

When a specific instance of an application-process wishes to communicate with an instance of an application-process in some other open system, it must invoke an instance of an application-entity in the Application Layer of its own open system. It then becomes the responsibility of this instance of the application-entity to establish an association with an instance of an appropriate application-entity in the destination open system. This process occurs by invocation of instances of entities in the lower layers. When the association between the two application-entities has been established, the application-process can communicate.

7.1.4.1 Grouping of functions in the Application Layer

An application-entity can be structured internally into groups of functions. The technique used to express this structure is not constrained by this International Standard. Use of one grouping of functions may depend on use of some other functions, and the active functions may vary during the lifetime of a connection.

The structuring of application-entities into application-service-elements and the user-element provides an organization of functions in application-entities. Furthermore, any given subset of application-service-elements, along with the user-element, constitutes an application-entity type. Each application-entity type, and each instance thereof, are unambiguously identifiable.

An application-process may determine the grouping of functions comprising the application-entity.

Two categories of application-service-elements are recognized: common-application-service-elements and specific-application-service-elements. Common-application-service-elements provide capabilities that are generally useful to a variety of applications. Specific-application-service-elements provide capabilities required to satisfy the particular needs of specific applications (for example, file transfer, data base access, job transfer, banking, order entry). Application-entities may contain application-service-elements from both categories, as illustrated in figure 14.

The partioning of application-service-elements into these two categories does not imply the existence of two independent protocols.

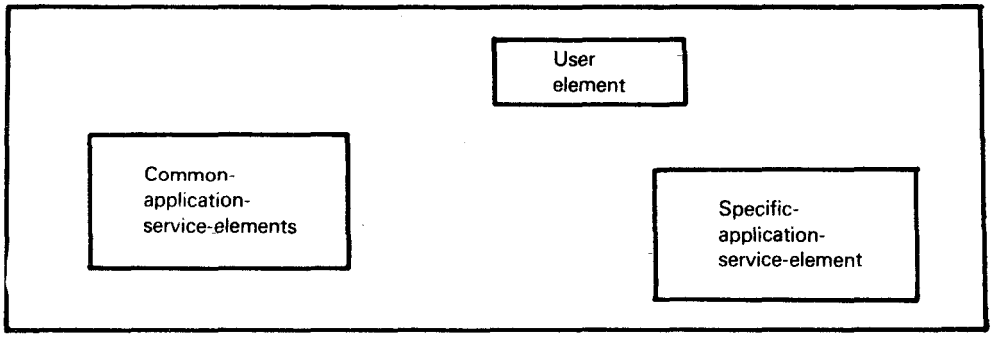


Figure 14 — Application-entity

7.1.4.2 Systems-management and application-management

Systems-management functions and application-management functions are located in the Application Layer. For details, see 5.9.

7.1.4.3 Application Layer management

In addition to systems and application-management, there are other activities specifically related to Application Layer management (such as activation and error control). See 5.9 for the relationship with other management aspects.

7.2 Presentation Layer

7.2.1 Definitions

7.2.1.1 concrete syntax: Those aspects of the rules used in the formal specification of data which embody a specific representation of that data.

7.2.1.2 transfer syntax: That concrete syntax used in the transfer of data between open systems.

7.2.2 Purpose

The Presentation Layer provides for the representation of information that application-entities either communicate or refer to in their communication.

The Presentation Layer covers two complementary aspects of this representation of information :

a) the representation of data to be transferred between application-entities; and

b) the representation of the data structure which application-entities refer to in their communication, along with the representations of the set of actions which may be performed on this data structure.

The complementary aspects of the representation of information outlined above refer to the general concept of transfer syntax.

The Presentation Layer is concerned only with the syntax i.e. the representation of the data and not with its semantics, i.e. their meaning to the Application Layer, which is known only by the application-entities.

The Presentation Layer provides for a common representation to be used between application-entities. This relieves application-entities of any concern with the problem of "common" representation of information, i.e. it provides them with syntax independence. This syntax independence can be described in two ways :

a) the Presentation Layer provides common syntactical elements which are used by application-entities; and

b) the application-entities can use any syntax and the Presentation Layer provides the transformation between these syntaxes and the common syntax needed for communication between application-entities. This transformation is performed inside the open systems. It is not seen by other open systems and therefore has no impact on the standardization of presentation-protocols.

In this International Standard the approach outlined in b) is used.

7.2.3 Services provided to the Application Layer

The Presentation Layer provides session-services (see 7.3) and the following facilities :

a) transformation of syntax; and

b) selection of syntax.

Transformation of syntax is concerned with code and character set conversions, with the modification of the layout of the data and the adaptation of actions on the data structures. Selection of syntax provides the means of initially selecting a syntax and subsequently modifying the selection.

Session services are provided to application-entities in the form of presentation services.

7.2.4 Functions within the Presentation Layer

The Presentation Layer performs the following functions to help accomplish the presentation-services :

a) session establishment request;

b) data transfer;

c) negotiation and renegotiation of syntax;

d) transformation of syntax including data transformation, formatting and special purpose transformations (for example compression); and

e) session termination request.

7.2.4.1 Transformation of syntax

The fact that there is or is not actual transformation of syntax has no impact on the presentation-protocol.

There are three syntactic versions of the data : the syntax used by the originating application-entity, the syntax used by the receiving application-entity, and the syntax used between presentation-entities (the transfer syntax). It is clearly possible that any two or all three of these syntaxes may be identical. The Presentation Layer contains the functions necessary to transform between the transfer syntax and each of the other two syntaxes as required.

There is not a single predetermined transfer syntax for all OSI. The transfer syntax to be used on a presentation-connection is negotiated between the correspondent presentation-entities. Thus, a presentation-entity must know the syntax of its application-entity and the agreed transfer syntax. Only the transfer syntax needs to be referred to in the Presentation Layer protocols.

To meet the service requirement specified by the application-entities during the initiation phase, the Presentation Layer may utilise any transfer syntax available to it. To accomplish other service objectives (for example data volume reduction to reduce data transfer cost), syntax transformation may be performed either as a specific syntax-matching service provided to the application-entities, or as a function internal to the Presentation Layer.

7.2.4.2 Negotiation of syntax

Negotiation of syntax is carried out by communication between the presentation-entities on behalf of the application-entities to determine the form that data will have while in the OSI environment. The negotiations will determine what transformations are needed (if any) and where they will be performed. Negotiations may be limited to the initiation phase or they may occur any time during a session.

In OSI, the syntaxes used by application-entities that wish to communicate may be very similar or quite dissimilar. When they are similar, the transformation functions may not be needed at all; however, when they are dissimilar, the Presentation Layer services provide the means to converse and decide where needed transformations will take place.

7.2.4.3 Addressing and multiplexing

There is a one-to-one correspondence between presentation-address and session-address. There is no multiplexing or splitting in the Presentation Layer.

7.2.4.4 Presentation Layer management

The Presentation Layer protocols deal with some management activities of the layer (such as activation and error control). See 5.9 for the relationship with other management aspects.

7.3 Session Layer

7.3.1 Definitions

7.3.1.1 quarantine service : A facility of the session-service by which an integral number of session-service-data-units sent on a session-connection are not made available to the receiving presentation-entity until explicitly released by the sending presentation-entity.

7.3.1.2 interaction management : A facility of the session-service which allows correspondent presentation-entities to control explicitly whose turn it is to exercise certain control functions.

7.3.1.3 two-way-simultaneous interaction : A mode of interaction where both presentation-entities may concurrently send and receive.

7.3.1.4 two-way-alternate interaction : A mode of interaction where the presentation-entity with the turn may send and its correspondent is permitted only to receive.

7.3.1.5 one-way interaction : A form of operation of two-way-alternate interaction in which the turn can never be exchanged.

7.3.1.6 session-connection synchronization : A facility of the session-service which allows presentation-entities to define and identify synchronization points and to reset a session-connection to a predefined state and to agree on a resynchronization point.

7.3.2 Purpose

The purpose of the Session Layer is to provide the means necessary for co-operating presentation-entities to organize and synchronize their dialogue and to manage their data exchange. To do this, the Session Layer provides services to establish a session-connection between two presentation-entities, and to support orderly data exchange interactions.

To implement the transfer of data between the presentation-entities, the session-connection is mapped onto and uses a transport-connection (see 7.3.4.1).

A session-connection is created when requested by a presentation-entity at a session-service-access-point. During the lifetime of the session-connection, session services are

used by the presentation-entities to regulate their dialogue, and to ensure an orderly message exchange on the session-connection. The session-connection exists until it is released by either the presentation-entities or the session-entities. While the session-connection exists, session-services maintain the state of the dialogue even over data loss by the Transport Layer.

A presentation-entity can access another presentation-entity only by initiating or accepting a session-connection. A presentation-entity may be associated with several session-connections simultaneously. Both concurrent and consecutive session-connections are possible between two presentation-entities.

The initiating presentation-entity designates the destination presentation-entity by a session-address. In many systems, a transport-address may be used as the session-address, i.e., there is a one-to-one correspondence between the session-address and the transport-address. In general, however, there is a many-to-one correspondence between session-addresses and transport-addresses. This does not imply multiplexing of session-connections onto transport-connections, but does imply that at session-connection establishment time, more than one presentation-entity is a potential target of a session-connection establishment request arriving on a given transport-connection.

7.3.3 Services provided to the Presentation Layer

The following services provided by the Session Layer are described below :

a) session-connection establishment;

b) session-connection release;

c) normal data exchange;

d) quarantine service;

e) expedited data exchange;

f) interaction management;

g) session-connection synchronization; and

h) exception reporting.

7.3.3.1 Session-connection establishment

The session-connection establishment service enables two presentation-entities to establish a session-connection between themselves. The presentation-entities are identified by session-addresses used to request the establishment of the session-connection.

The session-connection establishment service allows the presentation-entities co-operatively to determine the unique values of session-connection parameters at the time the session-connection is established.

NOTE The provision for change of session parameters after session-connection establishment is a candidate for further extension.

Simultaneous session-connection establishment requests typically result in a corresponding number of session-connections, but a session-entity can always reject an incoming request.

The session-connection establishment service provides to the presentation-entities a session-service-connection-identifier which uniquely specifies the session-connection within the environment of the correspondent presentation-entities, with a lifetime which may be greater than the lifetime of the session-connection. This identifier may be used by the presentation-entities to refer to the session-connection during the lifetime of the session-connection, and may also be used by management-entities for administrative purposes such as accounting.

7.3.3.2 Session-connection release

The session-connection release service allows presentation-entities to release a session-connection in an orderly way without loss of data. It also allows either presentation-entity to request at any time that a session-connection be aborted; in this case, data may be lost.

The release of a session-connection may also be initiated by one of the session-entities supporting it.

7.3.3.3 Normal data exchange

The normal data exchange service allows a sending presentation-entity to transfer a session-service-data-unit to a receiving presentation-entity. This service allows the receiving presentation-entity to ensure that it is not overloaded with data.

7.3.3.4 Quarantine service

The quarantine service allows the sending presentation-entity to request that an integral number of session-service-data-units (one or more) sent on a session-connection should not be made available to the receiving presentation-entity until explicitly released by the sending presentation-entity. The sending presentation-entity may request that all data currently quarantined be discarded. The receiving presentation-entity receives no information that data being received has been quarantined or that some data was discarded.

7.3.3.5 Expedited data exchange

The expedited data exchange service provides expedited handling for the transfer of expedited session-service-data-units. A specific size restriction is placed on expedited session-service-data-units. This service may be used by a presentation-entity any time a session-connection exists.

7.3.3.6 Interaction management

The interaction management service allows the presentation-entities to control explicitly whose turn it is to exercise certain control functions.

The service provides for voluntary exchange of the turn where the presentation-entity which has the turn relinquishes it voluntarily. This service also provides for forced exchange of the turn

where, upon request from the presentation-entity which does not have the turn, the session-service may force the presentation-entity with the turn to relinquish it. In the case of forced exchange of the turn, data may be lost.

The following types of session-service-data-unit exchange interaction are defined :

a) two-way-simultaneous (TWS);

b) two-way-alternate (TWA); and

c) one-way interaction.

7.3.3.7 Session-connection synchronization

The session-connection synchronization service allows presentation-entities to

a) define and identify synchronization points; and

b) reset the session-connection to a defined state and agree on a resynchronization point.

The Session Layer is not responsible for any associated checkpointing or commitment action associated with synchronization.

7.3.3.8 Exception reporting

The exception reporting service permits the presentation-entities to be notified of exceptional situations not covered by other services, such as unrecoverable session malfunctions.

NOTE – The following services are candidates for future extensions :

a) session-service-data-unit sequence numbering;

b) brackets;

c) stop-go; and

d) security.

7.3.4 Functions within the Session Layer

The functions within the Session Layer are those which shall be performed by session-entities in order to provide the session-services.

Most of the functions required are readily implied by the services provided. Additional description is given below for the following functions :

a) session-connection to transport-connection mapping;

b) session-connection flow control;

c) expedited data transfer;

d) session-connection recovery;

e) session-connection release; and

f) Session Layer management.

7.3.4.1 Session-connection to transport-connection mapping

There is a one-to-one mapping between a session-connection and a transport-connection at any given instant. However, the lifetime of a transport-connection and that of a related session-connection can be distinguished so that the following cases are defined :

a) a transport-connection supports several consecutive session-connections (see figure 15); and

b) several consecutive transport-connections support a session-connection (see figure 16).

NOTES

1 It is also possible to consider cases in which one transport-connection is used to support several session-connections (i.e., n-to-1 mapping). In this case peer flow control would be required in the Session Layer. This case is for future development if needed.

2 To implement the mapping of a session-connection onto a transport-connection, the Session Layer has to map session-service-data-units into session-protocol-data-units, and session-protocol-data-units into transport-service-data-units. These mappings may require the session-entities to perform functions such as segmenting. These functions are visible only in the session-protocols, therefore they are transparent to the Presentation and Transport Layers.

7.3.4.2 Session-connection flow control

There is no peer flow control in the Session Layer. To prevent the receiving presentation-entity from being overloaded with data, the receiving session-entity applies back pressure across the transport-connection using the transport flow control.

7.3.4.3 Expedited data transfer

The transfer of expedited session-service-data-units is generally accomplished by use of the expedited transport service.

7.3.4.4 Session-connection recovery

In the event of reported failure of an underlying transport-connection, the Session Layer may contain the functions necessary to re-establish a transport-connection to support the session-connection, which continues to exist. The session-entities involved notify the presentation-entities via the exception reporting service that service is interrupted and restore the service only as directed by the presentation-entities. This permits the presentation-entities to resynchronize and continue from an agreed state.

7.3.4.5 Session-connection release

The Session Layer contains the functions necessary to release the session-connection in an orderly way, without loss of data, upon request by the presentation-entities. The Session Layer also contains the necessary functions to abort the session-connection with the possible loss of data.

7.3.4.6 Session Layer management

The Session Layer protocols deal with some management activities of the layer (such as activation and error control). See 5.9 for the relationship with other management aspects.

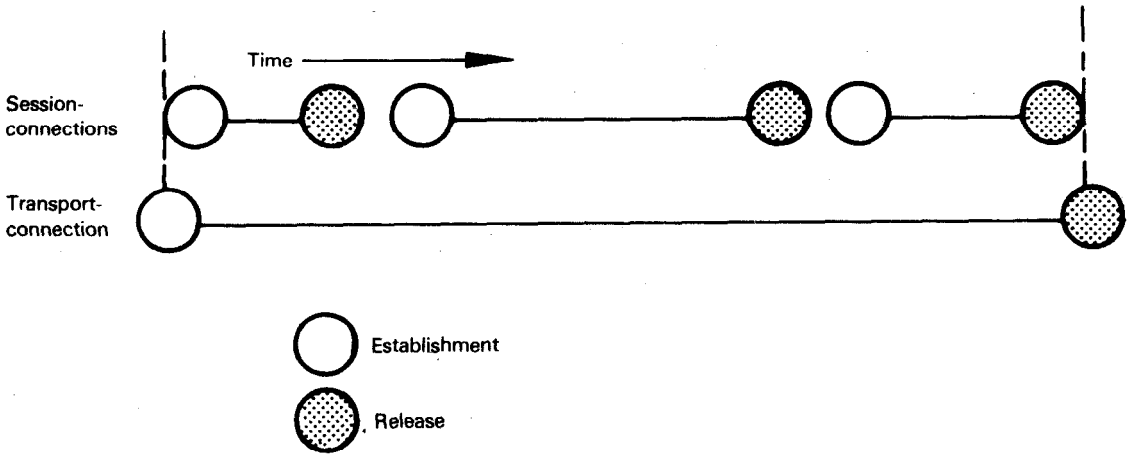


Figure 15 — Several consecutive session-connections

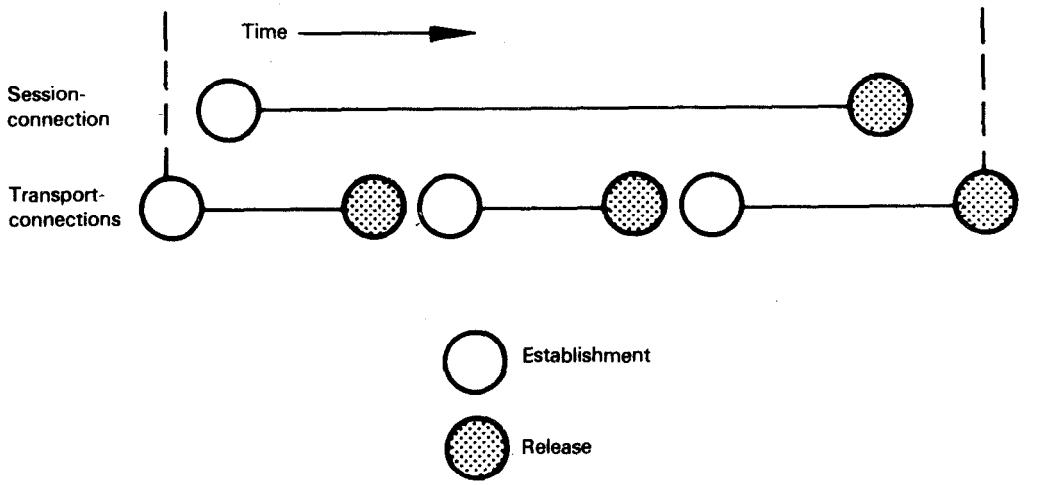


Figure 16 — Several consecutive transport-connections

7.4 Transport Layer

7.4.1 Definitions

No Transport Layer specific terms are identified.

7.4.2 Purpose

The transport-service provides transparent transfer of data between session-entities and relieves them from any concern with the detailed way in which reliable and cost effective transfer of data is achieved.

The Transport Layer optimizes the use of the available network-service to provide the performance required by each session-entity at minimum cost. This optimization is achieved within the constraints imposed by the overall demands of all concurrent session-entities and the overall quality and capacity of the network-service available to the Transport Layer.

All protocols defined in the Transport Layer have end-to-end significance, where the ends are defined as correspondent transport-entities. Therefore the Transport Layer is OSI end open system oriented and transport-protocols operate only between OSI end open systems.

The Transport Layer is relieved of any concern with routing and relaying since the network-service provides network-connections from any transport-entity to any other, including the case of tandem subnetworks (see 7.5.1).

The transport functions invoked in the Transport Layer to provide a requested service quality depend on the quality of the network-service. The quality of the network-service depends on the way the network-service is achieved (see 7.5.3).

7.4.3 Services provided to the Session Layer

The Transport Layer uniquely identifies each session-entity by its transport-address. The transport-service provides the means

establish, maintain and release transport-connections. Transport-connections provide duplex transmission between a pair of transport-addresses.

More than one transport-connection can be established between the same pair of transport-addresses. A session-entity uses transport-connection-endpoint-identifiers provided by the Transport Layer to distinguish between transport-connection-endpoints.

The operation of one transport-connection is independent of the operation of all others except for the limitations imposed by the finite resources available to the Transport Layer.

The quality of service provided on a transport-connection depends on the service class requested by the session-entities when establishing the transport-connection. The selected quality of service is maintained throughout the lifetime of the transport-connection. The session-entity is notified of any failure to maintain the selected quality of service on a given transport-connection.

The following services provided by the Transport Layer are described below :

a) transport-connection establishment;

b) data transfer; and

c) transport-connection release.

7.4.3.1 Transport-connection establishment

Transport-connections are established between session-entities identified by transport-addresses. The quality of service of the transport-connection is negotiated between the session-entities and the transport-service.

At the time of establishment of a transport-connection the class of transport service to be provided can be selected from a defined set of available classes of service.

These service classes are characterized by combinations of selected values of parameters such as throughput, transit delay, and connection set-up delay and by guaranteed values of parameters such as residual error rate and service availability.

These classes of service represent globally predefined combinations of parameters controlling quality of service. These classes of service are intended to cover the transport-service requirements of the various types of traffic generated by the session-entities.

7.4.3.2 Data transfer

This service provides data transfer in accordance with the agreed quality of service. When the quality of service cannot be maintained and all possible recovery attempts have failed, the transport-connection is terminated and the session-entities are notified.

a) The transport-service-data-unit transfer service provides the means by which transport-service-data-units of arbitrary length are delimited and transparently transferred in sequence from one sending transport-service-access-point to the receiving transport-service-access-point over a transport-connection. This service is subject to flow control.

b) The expedited transport-service-data-unit transfer service provides an additional means of information exchange on a transport-connection. The expedited transport-data-units are subject to their own set of transport-service and flow control characteristics. The maximum size of expedited transport-service-data-units is limited.

7.4.3.3 Transport-connection release

This service provides the means by which either session-entity can release a transport-connection and have the correspondent session-entity informed of the release.

7.4.4 Functions within the Transport Layer

The Transport Layer functions may include :

a) mapping transport-address onto network-address;

b) multiplexing (end-to-end) transport-connections onto network-connections;

c) establishment and release of transport-connections;

d) end-to-end sequence control on individual connections;

e) end-to-end error detection and any necessary monitoring of the quality of service;

f) end-to-end error recovery;

g) end-to-end segmenting, blocking and concatenation;

h) end-to-end flow control on individual connections;

j) supervisory functions; and

k) expedited transport-service-data-unit transfer.

7.4.4.1 Addressing

When a session-entity requests the Transport Layer to establish a transport-connection with another session-entity identified by its transport-address, the Transport Layer determines the network-address identifying the transport-entity which serves the correspondent session-entity.

Because transport-entities support services on an end-to-end basis no intermediate transport-entity is involved as a relay between the end transport-entities. Therefore the Transport Layer maps transport-addresses to the network-addresses which identify the end transport-entities (see figure 17).

One transport-entity may serve more than one session-entity. Several transport-addresses may be associated with one network-address within the scope of the same transport-entity. Corresponding mapping functions are performed within the transport-entities to provide these facilities (see figure 18).

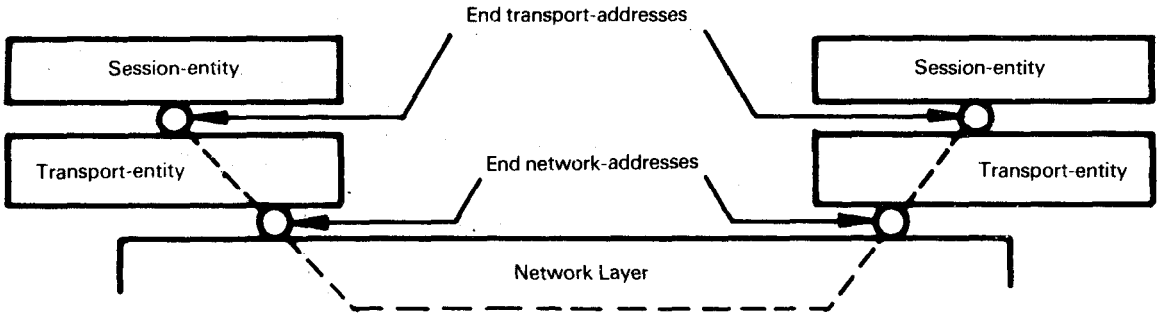


Figure 17 — Association of transport-addresses and network-addresses

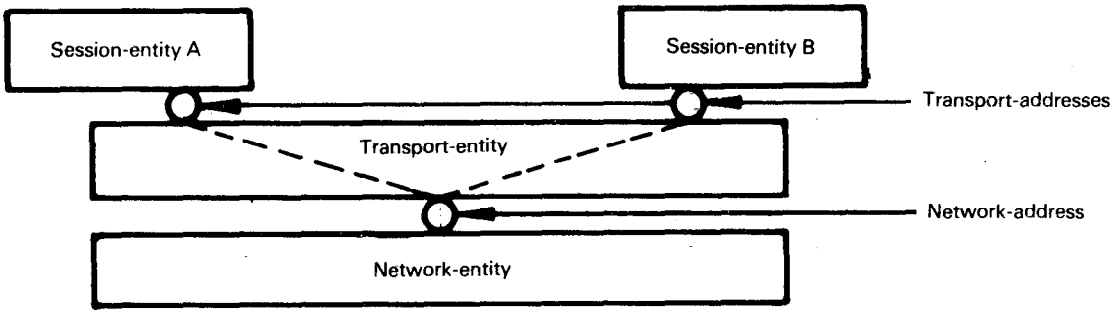


Figure 18 — Association of one network-address with several transport-addresses

7.4.4.2 Connection multiplexing and splitting

In order to optimize the use of network-connections, the mapping of transport-connections onto network-connections need not be on a one-to-one basis. Both splitting and multiplexing may be performed, namely for optimizing cost of usage of the network-service.

7.4.4.3 Phases of operation

The phases of operation within the Transport Layer are :

a) establishment phase;

b) data transfer phase; and

c) release phase.

The transfer from one phase of operation to an other will be specified in detail within the protocol for the Transport Layer.

7.4.4.4 Establishment phase

During the establishment phase, the Transport Layer establishes a transport-connection between two session-entities. The functions of the Transport Layer during this phase shall match the requested class of service with the services provided by the Network Layer. The following functions may be performed during this phase :

a) obtain a network-connection which best matches the requirements of the session-entity, taking into account cost and quality of service;

b) decide whether multiplexing or splitting is needed to optimize the use of network-connections;

c) establish the optimum transport-protocol-data-unit size;

d) select the functions that will be operational upon entering the data transfer phase;

e) map transport-addresses onto network-addresses;

f) provide identification of different transport-connections between the same pair of transport-service-access-points (connection identification function); and

g) transfer of data.

7.4.4.5 Data transfer phase

The purpose of the data transfer phase is to transfer transport-service-data-units between the two session-entities connected by the transport-connection. This is achieved by the transmis-

sion of transport-protocol-data-units and by the following functions, each of which is used or not used according to the class of service selected in the establishment phase :

a) sequencing;

b) blocking;

c) concatenation;

d) segmenting;

e) multiplexing or splitting;

f) flow control;

g) error detection;

h) error recovery;

j) expedited data transfer;

k) transport-service-data-unit delimiting; and

m) transport-connection identification.

7.4.4.6 Release phase

The purpose of the release phase is to release the transport-connection. It may include the following functions :

a) notification of reason for release;

b) identification of the transport-connection released; and

c) transfer of data.

7.4.4.7 Transport Layer management

The Transport Layer protocols deal with some management activities of the layer (such as activation and error control). See 5.9 for the relationship with other management aspects.

7.5 Network Layer

7.5.1 Definitions

7.5.1.1 subnetwork : A set of one or more intermediate open systems which provide relaying and through which end open systems may establish network-connections.

NOTE — A subnetwork is a representation within the OSI Reference Model of a real network such as a carrier network, a private network or a local area network.

7.5.1.2 subnetwork-connection : A communication path through a subnetwork which is used by entities in the Network Layer in providing a network-connection.

7.5.2 Purpose

The Network Layer provides the means to establish, maintain and terminate network-connections between open systems containing communicating application-entities and the functional and procedural means to exchange network-service-data-units between transport-entities over network-connections.

It provides to the transport-entities independence from routing and relay considerations associated with the establishment and operation of a given network-connection. This includes the case where several subnetworks are used in tandem (see 7.5.4.2) or in parallel. It makes invisible to transport-entities how underlying resources such as data-link-connections are used to provide network-connections.

Any relay functions and hop-by-hop service enhancement protocols used to support the network-service between the OSI end open systems are operating below the Transport Layer, i.e. within the Network Layer or below.

7.5.3 Services provided to the Transport Layer

The basic service of the Network Layer is to provide the transparent transfer of data between transport-entities. This service allows the structure and detailed content of submitted data to be determined exclusively by layers above the Network Layer.

All services are provided to the Transport Layer at a known cost.

The Network Layer contains functions necessary to provide the Transport Layer with a firm Network/Transport Layer boundary which is independent of the underlying communications media in all things other than quality of service. Thus the Network Layer contains functions necessary to mask the differences in the characteristics of different transmission and subnetwork technologies into a consistent network service.

The service provided at each end of a network-connection is the same even when a network-connection spans several subnetworks, each offering dissimilar services (see 7.5.4.2).

NOTE — It is important to distinguish the specialized use of the term "service" within the OSI Reference Model from its common use by suppliers of private networks and carriers.

The quality of service is negotiated between the transport-entities and the network-service at the time of establishment of a network-connection. While this quality of service may vary from one network-connection to another it will be agreed for a given network-connection and be the same at both network-connection-endpoints.

The following services or elements of services provided by the Network Layer are described below :

a) network-addresses;

b) network-connections;

c) network-connection-endpoint-identifiers;

d) network-service-data-unit transfer;

e) quality of service parameters;

f) error notification;

g) sequencing;

h) flow control;

j) expedited network-service-data-unit transfer;

k) reset; and

m) release.

Some of the services described below are optional. This means that :

1) the user has to request the service; and

2) the network-service provider may honour the request or indicate that the service is not available.

7.5.3.1 Network-addresses

Transport-entities are known to the Network Layer by means of network-addresses. Network-addresses are provided by the Network Layer and can be used by transport-entities uniquely to identify other transport-entities, i.e. network-addresses are necessary for transport-entities to communicate using the network-service. The Network Layer uniquely identifies each of the end open systems (represented by transport-entities) by their network-addresses. This may be independent of the addressing needed by the underlying layers.

7.5.3.2 Network-connections

A network-connection provides the means of transferring data between transport-entities identified by network-addresses. The Network Layer provides the means to establish, maintain and release network-connections.

A network-connection is point-to-point.

More than one network-connection may exist between the same pair of network-addresses.

7.5.3.3 Network-connection-endpoint-identifiers

The Network Layer provides to the transport-entity a network-connection-endpoint-identifier which uniquely identifies the network-connection-end-point with the associated network-address.

7.5.3.4 Network-service-data-unit transfer

On a network-connection, the Network Layer provides for the transmission of network-service-data-units. These units have a distinct beginning and end and the integrity of the unit's content is maintained by the Network Layer.

No limit is imposed on the maximum size of network-service-data-units.

The network-service-data-units are transferred transparently between transport-entities.

7.5.3.5 Quality of service parameters

The Network Layer establishes and maintains a selected quality of service for the duration of the network-connection.

The quality of service parameters include residual error rate, service availability, reliability, throughput, transit delay (including variations), and delay for network-connection establishment.

7.5.3.6 Error notification

Unrecoverable errors detected by the Network Layer are reported to the transport-entities.

Error notification may or may not lead to the release of the network-connection, according to the specification of a particular network-service.

7.5.3.7 Sequencing

The Network Layer may provide sequenced delivery of network-service-data-units over a given network-connection when requested by the transport-entities.

7.5.3.8 Flow control

A transport-entity which is receiving at one end of a network-connection can cause the network-service to stop transferring network-service-data-units across the service-access-point. This flow control condition may or may not be propagated to the other end of the network-connection and thus be reflected to the transmitting transport-entity, according to the specification of a particular network-service.

7.5.3.9 Expedited network-service-data-unit transfer (optional)

The expedited network-service-data-unit transfer is optional and provides an additional means of information exchange on a network-connection. The transfer of expedited network-service-data-units is subject to a different set of network-service characteristics and to separate flow control.

The maximum size of expedited network-service-data-units is limited.

7.5.3.10 Reset (optional)

The reset service is optional and when invoked causes the Network Layer to discard all network-service-data-units in transit on the network-connection and to notify the transport-entity at the other end of the network-connection that a reset has occurred.

7.5.3.11 Release

A transport-entity may request release of a network-connection. The network-service does not guarantee the delivery of data preceding the release request and still in transit. The network-connection is released regardless of the action taken by the correspondent transport-entity.

7.5.3.12 Receipt of confirmation

A transport-entity may confirm receipt of data over a network-connection. The use of receipt confirmation service shall be agreed by the two users of the network-connection during connection establishment.

The service is an optional service that may not always be available.[1)]

7.5.4 Functions within the Network Layer

Network Layer functions provide for the wide variety of configurations supporting network-connections ranging from network-connections supported by point-to-point configurations, to network-connections supported by complex combinations of subnetworks with different characteristics.

NOTE – In order to cope with this wide variety of cases, network functions should be structured into sublayers. The subdivision of the Network Layer into sublayers need only be done when this is useful. In particular, sublayering need not be used when the access protocol to the subnetwork supports the complete functionality of the OSI Network Service.

The following are functions performed by the Network Layer :

a) routing and relaying;

b) network-connections;

c) network-connection multiplexing;

d) segmenting and blocking;

e) error detection;

f) error recovery;

g) sequencing;

h) flow control;

j) expedited data transfer;

k) reset;

m) service selection; and

n) network layer management.

7.5.4.1 Routing and relaying

Network-connections are provided by network-entities in end open systems but may involve intermediate open systems which provide relaying. These intermediate open systems may interconnect subnetwork-connections, data-link-connections, and data-circuits (see 7.7). Routing functions determine an appropriate route between network-addresses. In order to set up the resulting communication, it may be necessary for the Network Layer to use the services of the Data Link Layer to control the interconnection of data-circuits (see 7.6.4.10 and 7.7.3.1).

The control of interconnection of data-circuits (which are in the Physical Layer) from the Network Layer requires interaction between a network-entity and a physical-entity in the same open system. Since the Reference Model permits direct interaction only between adjacent layers, the network-entity cannot interact directly with the physical-entity. This interaction is thus described through the Data Link Layer which intervenes transparently to convey the interaction between the Network Layer and the Physical Layer.

This representation is an abstract representation of something happening inside an open system and which does not model the functioning of real open systems and as such has no impact on the standardization of OSI protocols.

NOTE – When Network Layer functions are performed by combinations of several individual subnetworks, the specification of routing and relaying functions could be facilitated by using sublayers, isolating individual subnetworks routing and relaying functions from internetwork routing and relaying functions.

However, when subnetworks have access protocols supporting the complete functionality of the OSI network service, there need be no sublayering in the Network Layer.

7.5.4.2 Network-connections

This function provides network-connections between transport-entities, making use of data-link-connections provided by the Data Link Layer.

A network-connection may also be provided as subnetwork-connections in tandem, i.e. using several individual subnetworks in series. The interconnected individual subnetworks may have the same or different service capabilities. Each end of a subnetwork-connection may operate with a different subnetwork protocol.

The interconnection of a pair of subnetworks of differing qualities may be achieved in two ways. To illustrate these, consider a pair of subnetworks, one of high quality and the other of low quality :

a) The two subnetworks are interconnected as they stand. The quality of the resulting network-connection is not higher than that of the lower quality subnetwork (see figure 19).

b) The lower quality subnetwork is enhanced equal to the higher quality subnetwork and the subnetworks are then interconnected. The quality of the resulting network-connection is approximately that of the higher quality subnetwork.

The choice between these two alternatives depends on the degree of difference in quality, the cost of enhancement, and other economic factors.

1) This service is included in the network service only to support existing features of CCITT Recommendation X.25.

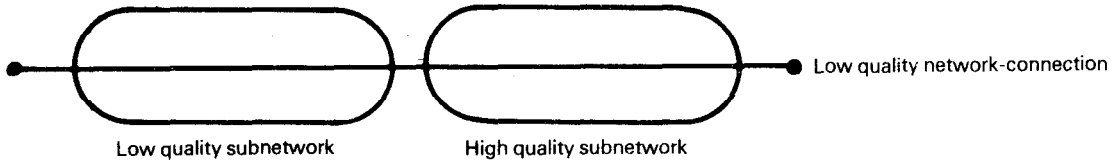


Figure 19 — Interconnection of a low quality subnetwork and a high quality subnetwork

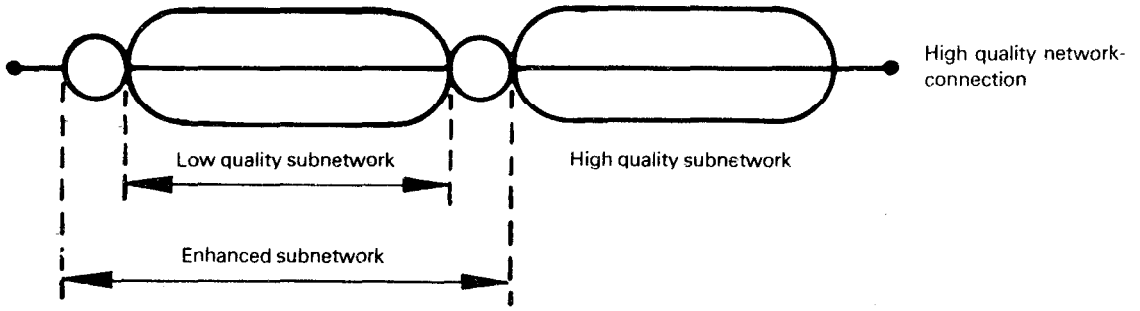


Figure 20 — Interconnection of an enhanced low quality subnetwork and a high quality subnetwork

7.5.4.3 Network-connection multiplexing

This function may be used to multiplex network-connections onto data-link-connections in order to optimize their use.

In the case of subnetwork-connections in tandem, multiplexing onto individual subnetwork-connections may also be performed in order to optimize their use.

7.5.4.4 Segmenting and blocking

The Network Layer may segment and/or block network-service-data-units for the purpose of facilitating the transfer. However the network-service-data-unit delimiters are preserved over the network-connection.

7.5.4.5 Error detection

Error detection functions are used to check that the quality of service provided over a network-connection is maintained. Error detection in the Network Layer uses error notification from the Data Link Layer. Additional error detection capabilities may be necessary to provide the required quality of service.

7.5.4.6 Error recovery

This function provides for the recovery from detected errors. This function may vary depending on the quality of the network service provided.

7.5.4.7 Sequencing

This function provides for the sequenced delivery of network-service-data-units over a given network-connection when requested by transport-entities.

7.5.4.8 Flow control

If flow control service is required (see 7.5.3.8), this function may need to be performed.

7.5.4.9 Expedited data transfer

This function provides for the expedited data transfer service.

7.5.4.10 Reset

This function provides for the reset service.

7.5.4.11 Service selection

This function allows service selection to be carried out to ensure that the service provided at each end of a network-connection is the same when a network-connection spans several subnetworks of dissimilar quality.

7.5.4.12 Network Layer management

The Network Layer protocols deal with some management activities of the layer (such as activation and error control). See 5.9 for the relationship with other management aspects.

7.6 Data Link Layer

7.6.1 Definitions

No Data Link Layer specific terms are identified.

7.6.2 Purpose

The Data Link Layer provides functional and procedural means to establish, maintain and release data-link-connections among network-entities and to transfer data-link-service-data-units. A data-link-connection is built upon one or several physical-connections.

The Data Link Layer detects and possibly corrects errors which may occur in the Physical Layer.

In addition, the Data Link Layer enables the Network Layer to control the interconnection of data-circuits within the Physical Layer.

7.6.3 Services provided to the Network Layer

The following services or elements of services provided by the Data Link Layer are described below :

a) data-link-connection;

b) data-link-service-data-units;

c) data-link-connection-endpoint-identifiers;

d) sequencing;

e) error notification;

f) flow control; and

g) quality of service parameters.

7.6.3.1 Data-link-connection

The Data Link Layer provides one or more data-link-connections between two network-entities. A data-link-connection is always established and released dynamically.

7.6.3.2 Data-link-service-data-units

The Data Link Layer allows exchange of data-link-service-data-units over a data-link-connection.

The size of the data-link-service-data-units may be limited by the relationship between the physical-connection error rate and the Data Link Layer error detection capability.

7.6.3.3 Data-link-connection-endpoint-identifiers

If needed, the Data Link Layer provides data-link-connection-endpoint-identifiers that can be used by a network-entity to identify a correspondent network-entity.

7.6.3.4 Sequencing

When required, the sequence integrity of data-link-service-data-units is maintained.

7.6.3.5 Error notification

Notification is provided to the network-entity when any unrecoverable error is detected by the Data Link Layer.

7.6.3.6 Flow control

Each network-entity can dynamically control (up to the agreed maximum) the rate at which it receives data-link-service-data-units from a data-link-connection. This control may be reflected in the rate at which the Data Link Layer accepts data-link-service-data-units at the correspondent data-link-connection-endpoint.

7.6.3.7 Quality of service parameters

Quality of service parameters may be optionally selectable. The Data Link Layer establishes and maintains a selected quality of service for the duration of the data-link-connection. The quality of service parameters include mean time between detected but unrecoverable errors, residual error rate (where errors may arise form alteration, loss, duplication, disordering, misdelivery of data-link-service-data-unit, and other causes), service availability, transit delay and throughput.

7.6.4 Functions within the Data Link Layer

The following functions performed by the Data Link Layer are described below :

a) data-link-connection establishment and release;

b) data-link-service-data-unit mapping;

c) data-link-connection splitting;

d) delimiting and synchronization;

e) sequence control;

f) error detection;

g) error recovery;

h) flow control;

j) identification and parameter exchange;

k) control of data-circuit interconnection; and

m) Data Link Layer management.

7.6.4.1 Data-link-connection establishment and release

This function establishes and releases data-link-connections on activated physical-connections. When a physical-connection has multiple endpoints (for example multipoint connection), a specific function is needed within the Data Link Layer to identify the data-link-connections using such a physical-connection.

7.6.4.2 Data-link-service-data-unit mapping

This function maps data-link-service-data-units into data-link-protocol-data-units on a one-to-one basis.

NOTE — More general mappings are for further study.

7.6.4.3 Data-link-connection splitting

This function performs splitting of one data-link-connection onto several physical-connections.

7.6.4.4 Delimiting and synchronization

These functions provide recognition of a sequence of physical-service-data-units (i.e. bits, see 7.7.3.2) transmitted over the physical-connection, as a data-link-protocol-data-unit.

NOTE — These functions are sometimes referred to as framing.

7.6.4.5 Sequence control

This function maintains the sequential order of data-link-service-data-units across a data-link-connection.

7.6.4.6 Error detection

This function detects transmission, format and operational errors occurring either on the physical-connection, or as a result of a malfunction of the correspondent data-link-entity.

7.6.4.7 Error recovery

This function attempts to recover from detected transmission, format and operational errors and notifies the network-entities of errors which are unrecoverable.

7.6.4.8 Flow control

This function provides the flow control service as indicated in 7.6.3.6.

7.6.4.9 Identification and parameter exchange

This function performs data-link-entity identification and parameter exchange.

7.6.4.10 Control of data-circuit interconnection

This function conveys to network-entities the capability of controlling the interconnection of data-circuits within the Physical Layer.

7.6.4.11 Data Link Layer management

The Data Link Layer protocols deal with some management activities of the layer (such as activation and error control). See 5.9 for the relationship with other management aspects.

7.7 Physical Layer

7.7.1 Definition

7.7.1.1 data-circuit : A communication path in the physical media for OSI between two physical-entities, together with the facilities necessary in the Physical Layer for the transmission of bits on it.

7.7.2 Purpose

The Physical Layer provides mechanical, electrical, functional and procedural means to activate, maintain and de-activate physical connections for bit transmission between data-link-entities. A physical-connection may involve intermediate open systems, each relaying bit transmission within the Physical Layer. Physical Layer entities are interconnected by means of a physical medium.

7.7.3 Services provided to the Data Link Layer

The following services or elements of services provided by the Physical Layer are described below :

a) physical-connections;

b) physical-service-data-units;

c) physical-connection-endpoints;

d) data-circuit identification;

e) sequencing,

f) fault condition notification; and

g) quality of service parameters.

7.7.3.1 Physical-connections

The Physical Layer provides for the transparent transmission of bit streams between data-link-entities across physical-connections.

A data-circuit is a communication path in the physical media for OSI between two physical-entities, together with the facilities necessary in the Physical Layer for the transmission of bits on it.

A physical-connection may be provided by the interconnection of data-circuits using relaying functions in the Physical Layer. The provision of a physical-connection by such an assembly of data-circuits is illustrated in figure 21.

The control of the interconnection of data-circuits is offered as a service to data-link-entities.

7.7.3.2 Physical-service-data-units

A physical-service-data-unit consists of one bit in serial transmission and of "n" bits in parallel transmission.

A physical-connection may allow duplex or half-duplex transmission of bit streams.

7.7.3.3 Physical-connection-endpoints

The Physical Layer provides physical-connection-endpoint-identifiers which may be used by a data-link-entity to identify physical-connection-endpoints.

A physical-connection will have two (point-to-point) or more (multi-endpoint) physical-connection-endpoints (see figure 22).

7.7.3.4 Data-circuit identification

The Physical Layer provides identifiers which uniquely specify the data-circuits between two adjacent open systems.

NOTE — This identifier is used by network-entities in adjacent open systems to refer to data-circuits in their dialogue.

7.7.3.5 Sequencing

The Physical Layer delivers bits in the same order in which they were submitted.

7.7.3.6 Fault condition notification

Data-link-entities are notified of fault conditions detected within the Physical Layer.

7.7.3.7 Quality of service parameters

The quality of service of a physical-connection is derived from the data-circuits forming it. The quality of service can be characterized by

a) error rate, where errors may arise from alteration, loss, creation, and other causes;

b) service availability;

c) transmission rate; and

d) transit delay.

7.7.4 Functions within the Physical Layer

The following functions performed by the Physical Layer are described below :

a) physical-connection activation and deactivation;

b) physical-service-data-unit transmission; and

c) Physical Layer management.

7.7.4.1 Physical-connection activation and deactivation

These functions provide for the activation and deactivation of physical-connections between two data-link-entities upon request from the Data Link Layer. These include a relay function which provides for interconnection of data-circuits.

7.7.4.2 Physical-service-data-unit transmission

The transmission of physical-service-data-units (i.e. bits) may be synchronous or asynchronous.

7.7.4.3 Physical Layer management

The Physical Layer protocols deal with some management activities of the layer (such as activation and error control). See 5.9 for the relationship with other management aspects.

NOTE — The above text deals with interconnection between open systems as illustrated in figure 12. For open systems to communicate in the real environment, real physical connections should be made as, for example, in figure 23. Their logical representation is as shown in figure 23 and is called the physical media connection.

The mechanical, electromagnetic and other media dependent characteristics of physical media connections are defined at the boundary beween the Physical Layer and the physical media. These characteristics are specified in other standards.

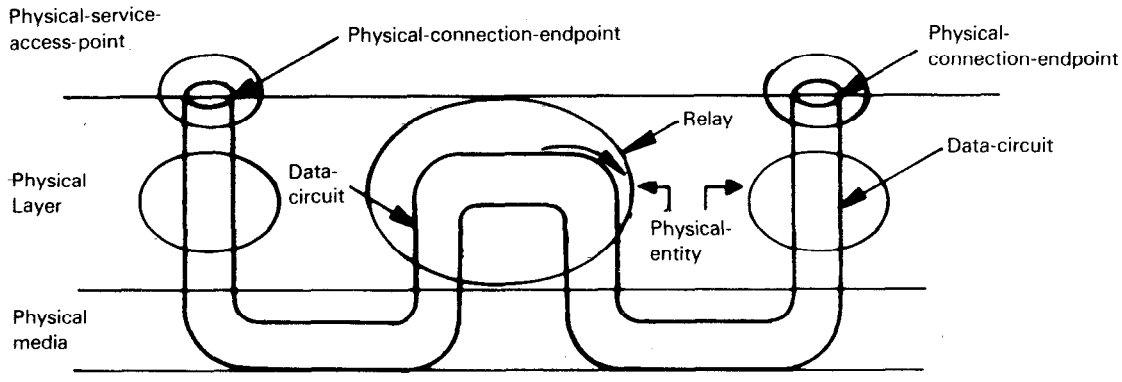
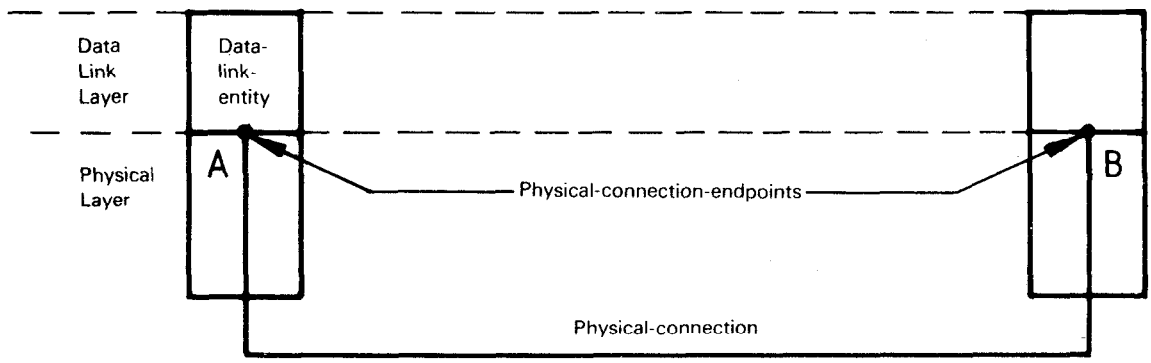
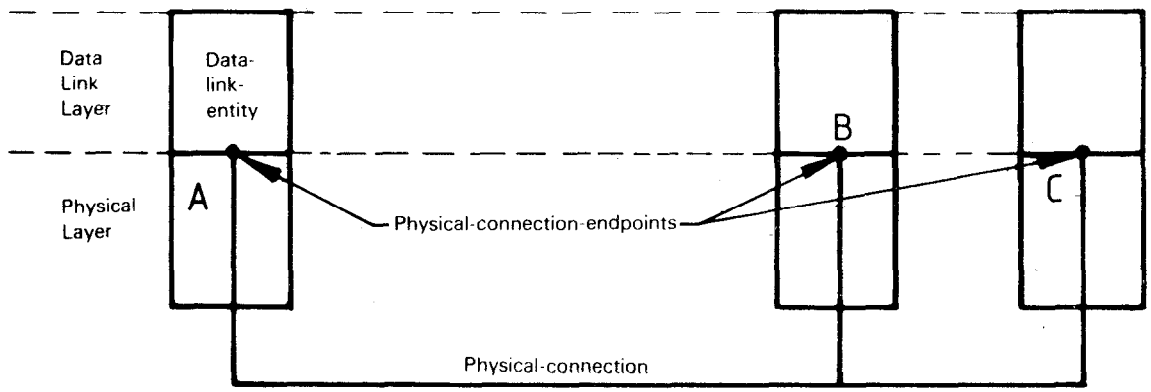


Figure 21 — Interconnection of data-circuits within the Physical Layer

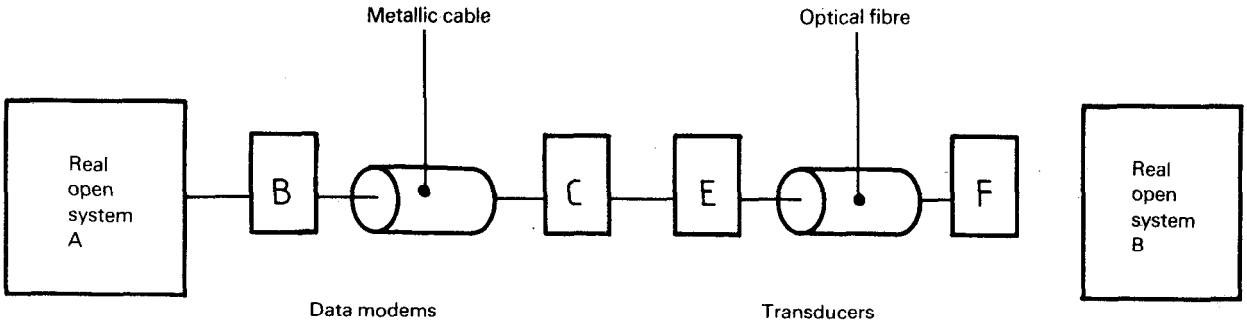


Example of a two-endpoint physical connection
(connection exists between A and B)

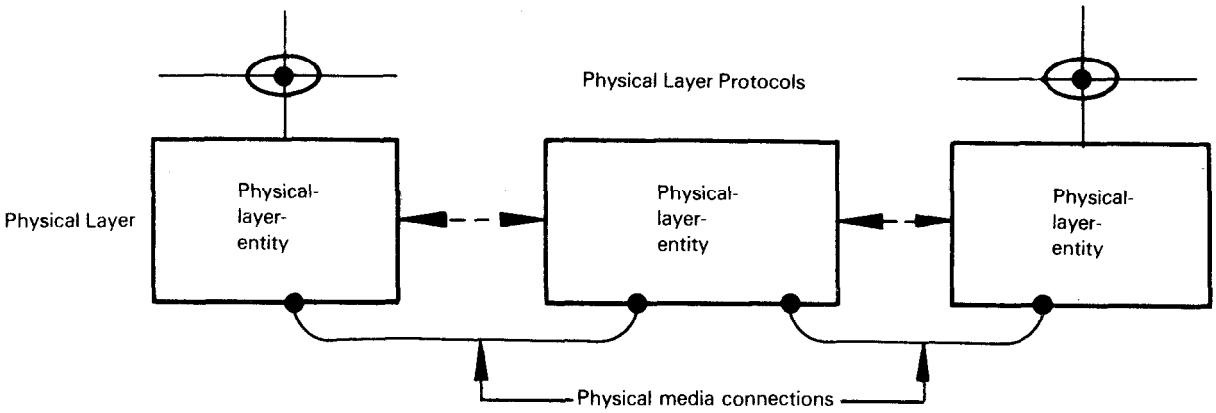


Example of a multi-endpoint-physical-connection
(connection exists between A, B and C)

Figure 22 — Examples of physical-connections



a) Real environment



b) Logical environment

Figure 23 — Examples of interconnection

NOTE — The area of physical media connections in OSI requires further study.

Annex A

Brief explanation of how the layers were chosen

(This annex does not form part of the standard.)

A.1 This annex provides elements giving additional information to this International Standard.

A.2 The following is a brief explanation of how the layers were chosen :

A.2.1 It is essential that the architecture permits usage of a realistic variety of physical media for interconnection with different control procedures (for example V.24, V.25, etc...). Application of principles in 6.2 c), e) and h) leads to identification of a **Physical Layer** as the lowest layer in the architecture.

A.2.2 Some physical communication media (for example telephone line) require specific techniques to be used in order to transmit data between systems despite a relatively high error rate (i.e. an error rate not acceptable for the great majority of applications). These specific techniques are used in data-link control procedures which have been studied and standardized for a number of years. It must also be recognized that new physical communication media (for example fibre optics) will require different data-link control procedures. Application of principles in 6.2 c), e) and h) leads to identification of a **Data Link Layer** on top of the Physical Layer in the architecture.

A.2.3 In the open systems architecture, some open systems will act as the final destination of data, see clause 4. Some open systems may act only as intermediate nodes (forwarding data to other systems), (see figure 13). Application of principles in 6.2 c), e) and g) leads to identification of a **Network Layer** on top of the Data Link Layer. Network oriented protocols such as routing, for example, will be grouped in this layer. Thus, the Network Layer will provide a connection path (network-connection) between a pair of transport-entities; including the case where intermediate nodes are involved, see figure 13 (see also 7.5.4.1).

A.2.4 Control of data transportation from source end open system to destination end open system (which is not performed in intermediate nodes) is the last function to be performed in order to provide the totality of the transport-service. Thus, the upper layer in the transport-service part of the architecture is the **Transport Layer**, on top of the Network Layer. This Transport Layer relieves higher layer entities from any concern with the transportation of data between them.

A.2.5 There is a need to organize and synchronize dialogue, and to manage the exchange of data. Application of principles in 6.2 c) and d) leads to the identification of a **Session Layer** on top of the Transport Layer.

A.2.6 The remaining set of general interest functions are those related to representation and manipulation of structured data for the benefit of application programs. Application of principles in 6.2 c) and d) leads to identification of a **Presentation Layer** on top of the Session Layer.

A.2.7 Finally, there are applications consisting of application-processes which perform information processing. An aspect of these application-processes and the protocols by which they communicate comprise the **Application Layer** as the highest layer of the architecture.

A.3 The resulting architecture with seven layers, illustrated in figure 12 obeys the principles in 6.2 a) and b).

A more detailed definition of each of the seven layers identified above is given in clause 7 of this International Standard, starting from the top with the Application Layer described in 7.1 down to the Physical Layer described in 7.7.

Annex B

Alphabetical index to definitions

(This annex forms part of the standard.)

Printed at PRINTOGRARH, New Delhi (India)

AMENDMENT NO. 1 JANUARY 1992
TO
IS 12373 : 1987
ISO 7498 : 1984

BASIC REFERENCE MODEL OF OPEN SYSTEMS INTERCONNECTION FOR INFORMATION PROCESSING SYSTEMS

(*First cover, title*)— Substitute the following for the existing title:

'*Indian Standard*

BASIC REFERENCE MODEL OF OPEN SYSTEMS INTERCONNECTION FOR INFORMATION PROCESSING SYSTEMS, PART 1'

(*First, third* and *fourth cover, pages* 1 *to* 40, *IS No.*) — Substitute the following for the existing IS No. :

' IS 12373 (Part 1) : 1987
ISO 7498 : 1984'

(LTD 36)

Reprography Unit, BIS, New Delhi, India

AMENDMENT NO. 2 NOVEMBER 1992
TO
IS 12373 (Part 1) : 1987/ ISO 7498 : 1984 BASIC REFERENCE MODEL OF OPEN SYSTEMS INTERCONNECTION FOR INFORMATION PROCESSING SYSTEMS : PART 1

[This Amendment No. 2 is based on Technical Corrigendum 1 to ISO 7498 : 1984. The page number mentioned in this amendment refer to the ISO page number given in the standard.]

Page 2

Subclause 4.2

Paragraph 4, item b): change "computerised" to "computerized".

Page 5

Subclause 5.2.2

line 3: change "as being logically" to "as logically".

Page 6

Subclause 5.2.2

NOTE 1, paragraph 1, line 7: delete "should".

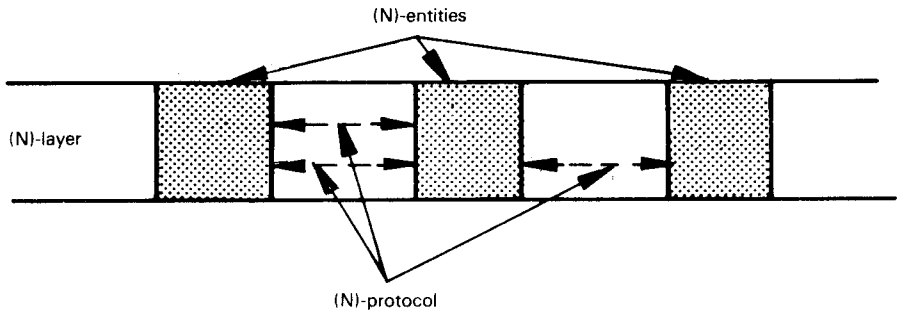
paragraph 2, line 3: change "that" to "that type".

line 6: change "instance." to "instance of that program."

paragraph 4, line 10: change "should be" to "is".

Page 7

Replace figure 5 by the following:

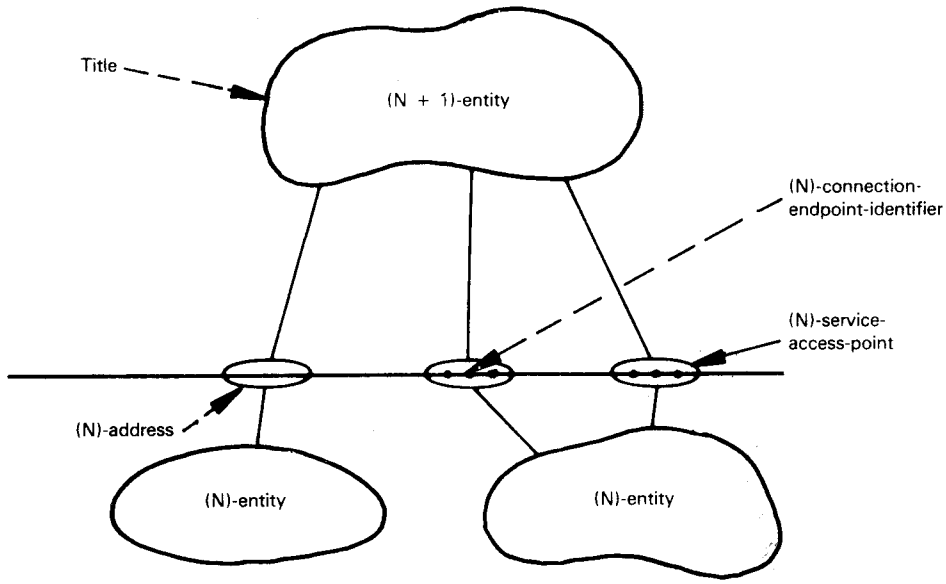


Subclause 5.3.2

Paragraph 1, line 2: change "shall be" to "is".

Page 9

Replace figure 7 by the following:



Page 10

Subclause 5.4.2

Paragraph 13, line 8: change "shall" to "must".

Subclause 5.6.1.3

Line 2: change "protocol-information" to "protocol-control-information".

Page 12

Subclause 5.7.1.17

NOTE, line 7: change "Also" to "Note also that".

Page 13

Subclause 5.7.4

Paragraph 10, line 4: change "both (N)-entities shall be" to "it requires that both (N)-entities are".

line 9: change "reestablished" to "re-established".

Page 14

Subclause 5.7.5

Paragraph 3, item a): change "since" to "where".

Page 15

Subclause 5.7.6.5

Paragraph 1, line 2: change "i.e." to "i.e.,".

line 4: change "-units" to "-unit.".

Subclause 5.7.6.6

lines 2 and 4: change "data" to "(N-1)-service-data-units".

Page 20

Subclause 6.2

item k): change "layer boundaries" to "layer, boundaries".

Subclause 7.1.1

Title: change "**Definition**" to "**Definitions**".

Page 22

Figure 14, lower right box: change "Specific-application-service-element" to "Specific-application-service-elements".

Subclause 7.2.2

Paragraph 4, line 1: change "i.e." to "i.e.,".

line 2: change "i.e." to "i.e.,".

Page 24

Subclause 7.3.3.5

line 4: change "by a presentation entity" to "by either presentation entity".

Page 25

Subclause 7.3.4

Paragraph 1, line 1: change "shall be" to "are".

Subclause 7.3.4.1

NOTE 2, line 2: change "has to map" to "maps".

Page 28

Subclause 7.4.4.3

Paragraph 2, line 1: change "an other" to "another".

Subclause 7.4.4.4

Paragraph 1, line 4: delete "shall".

Page 30

Subclause 7.5.3

Paragraph 7, item k: delete "and"; item m: change ";" to "; and"; and add item n). Add to the list "n) **receipt of confirmation**".

Subclause 7.5.3.1

lines 3 and 4: change "uniquely to identify" to "to identify uniquely".

line 4: change "i.e." to "i.e.,".

Page 31

Subclause 7.5.3.12

Paragraph 1, line 2: change "shall be" to "is".

Subclause 7.5.4.2

Paragraph 3, item b), last line: add "(see figure 20)".

Page 33

Subclause 7.6.3.7

lines 5 and 6: change "arise form" to "arise from".

Page 35

Subclause 7.7.4.3

Note, paragraph 1, line 4: change "figure 23" to "figure 23a".

line 5: change "figure 23" to "figure 23b".

Paragraph 2, line 3: change "These" to "Definitions of such".

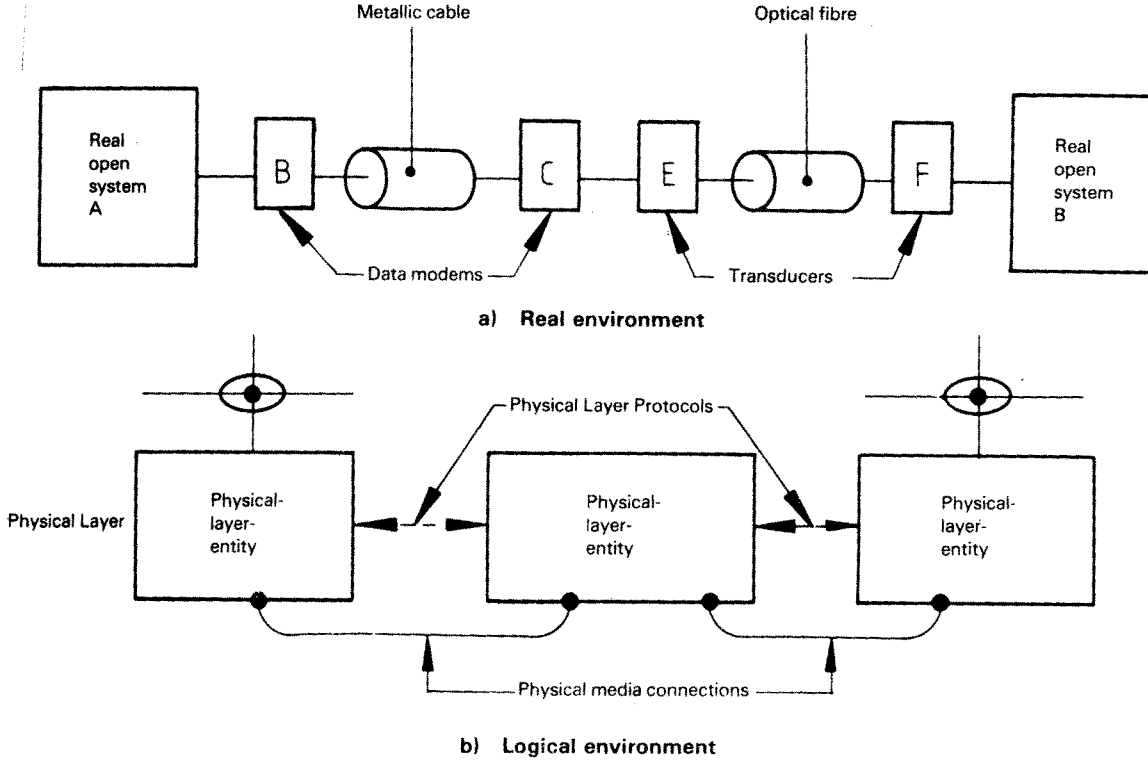
Page 36

Figure 22 – top figure, legend: change "physical connection" to "physical-connection".

– bottom figure, legend: change "endpoint-physical" to "endpoint physical".

Page 37

Replace figure 23 by the following:



Page 39

Annex B: change the term "application service element" to "application-service-element"

Page 40

Annex B: place the term "Transfer syntax" after "title-domain-name".

Printed at Simco Printing Press, Delhi, India

AMENDMENT NO. 3 MARCH 1993
TO
IS 12373 (Part 1) : 1987/ ISO 7498 : 1984 BASIC REFERENCE MODEL OF OPEN SYSTEMS INTERCONNECTION FOR INFORMATION PROCESSING SYSTEMS , PART 1

[This Amendment No. 3 is based on Addendum 1 : Connectionless-mode transmission to ISO 7498 : 1984. The page number mentioned in this amendment refer to the ISO page number given in the standard.]

Contents

0 Introduction

0.1 About this addendum

0.1.1 ISO 7498 describes the Reference Model of Open Systems Interconnection. It is the intention of ISO 7498 that the Reference Model should establish a framework for coordinating the development of existing and future standards for the interconnection of systems. The assumption that a connection is a fundamental prerequisite for communication in the OSI environment permeates the Reference Model and is one of the most useful and important unifying concepts of the architecture which it describes. However, since ISO 7498 was produced it has been realized that this deeply-rooted connection orientation unnecessarily limits the power and scope of the Reference Model, since it excludes important classes of applications and important classes of communication network technology which have a fundamentally connectionless nature.

0.1.2 The architectural objectives of the Reference Model do not depend on the exclusive use of connections for all OSI communications. It is the intention of this addendum to introduce terms and define their use within the Reference Model so that the two alternatives (connection-mode transmission and connectionless-mode transmission) can be treated as complementary concepts which can be applied appropriately in the different circumstances for which each is suited.

0.2 What is connectionless-mode transmission in the Reference Model ?

0.2.1 General

0.2.1.1 The concept of connectionless-mode transmission in one form or another has always played an important role in the specification of services and protocols for data communication.

The terms "message mode", "datagram", "transaction mode" and "connection-free" have been used in the literature to describe variations on the same basic theme : the transmission of a unit of data in a single self-contained operation without establishing, maintaining, and releasing a connection.

0.2.1.2 Since connectionless-mode transmission and connection-mode transmission are complementary concepts, they are best understood in juxtaposition, particularly since connectionless-mode transmission is defined most easily in relationship to the concept of a connection.

0.2.2 Connection-mode transmission in the Reference Model

0.2.2.1 In the formal terminology of the Reference Model, a connection is an association established for the transfer of data between two or more peer-entities. This association is established between the peer-entities themselves and between each entity and the next lower layer. The ability to establish and release a connection and to transfer data over it is provided to the entities in a given layer by the next lower layer as a connection-mode service. The use of a connection-mode service by peer-entities proceeds through three distinct phases :

a) connection establishment;

b) data transfer; and

c) connection release.

0.2.2.2 In addition to the clearly distinguishable lifetime exhibited by these phases a connection has the following fundamental characteristics :

a) it involves establishing and maintaining a three or more party agreement concerning the transmission of data between the peer-entities concerned and the layer providing the service;

b) it allows the negotiation between all the parties concerned of the parameters and options that will govern the transmission of data;

c) it provides connection identification by means of which the overheads involved in address resolution and transmission can be avoided on data transfers;

d) it provides a context within which successive units of data transmitted between the peer-entities are logically related, and makes it possible to maintain sequence and provide flow control for those transmissions.

0.2.2.3 The characteristics of connection-mode transmission are particularly attractive in applications which call for relatively long-lived, stream-oriented interactions between entities in stable configurations. Examples are provided by direct terminal use of a remote computer, file transfer, and long-term attachment of remote job entry stations. In these cases the entities involved initially discuss their requirements and agree to the terms of their interaction, reserving whatever resources they may need, transfer a series of related units of data to accomplish their mutual objective, and explicitly end their interaction, releasing the previously reserved resources.

The properties of connection-mode transmission are also relevant in a wide range of other applications.

0.2.3 Connectionless-mode transmission in the Reference Model

0.2.3.1 Again in the formal terminology of the Reference Model, connectionless-mode transmission is the transmission of a single unit of data from a source service-access-point to one or more destination service-access-points without establishing a connection. A connectionless-mode service allows an entity to initiate such a transmission by the performance of a single service access.

0.2.3.2 In contrast to a connection, an instance of the use of a connectionless-mode service does not have a clearly distinguishable lifetime. In addition it has the following fundamental characteristics :

a) it requires only a pre-arranged association between the peer-entities involved which determines the characteristics of the data to be transmitted, and no dynamic agreement is involved in an instance of the use of the service;

b) all the information required to deliver a unit of data, destination address, quality of service selection, options, etc., is presented to the layer providing the connectionless-mode service, together with the unit of data to be transmitted, in a single service access. The layer providing the connectionless-mode service is not required to relate this access to any other access.

0.2.3.3 As a result of these fundamental characteristics it may also be true that

a) each unit of data transmitted is routed independently by the layer providing the connectionless-mode service;

b) copies of a unit of data can be transmitted to a number of destination addresses.

0.2.3.4 These characteristics of connectionless-mode transmission do not preclude making available to the service user information on the nature and quality of service which may apply for a single invocation of the service or which may be observed over successive invocations of the service between pairs of service-access-points or among a set of service access points.

0.3 Structure of this addendum

This addendum has a structure which is similar to that of ISO 7498 in order to facilitate the eventual integration of the text of this addendum into a revision of ISO 7498. There are two major clauses :

Clause 5 : this clause deals with general architectural principles and the sub-clauses correspond to sub-clauses in clause 5 of ISO 7498 under which different aspects of connectionless-mode transmission need definition.

Clause 7 : this clause deals with details which are specific to the layers of the Reference Model of Open Systems Interconnection.

1 Scope and field of application

This addendum

a) provides a general description of a connectionless-mode service and of functions related to it, which may be provided by layers of the Reference Model, and

b) defines the positions within the Reference Model where the service and functions may be provided.

This addendum adds to the concepts and principles defined in ISO 7498; it does not modify them. This addendum

— does not specify services and protocols for OSI;

— is not an implementation specification;

— is not a basis for appraising the conformance of actual implementations.

2 Definitions

2.1 This addendum makes use of the following terms :

a) (N)-address;

b) acknowledgement;

c) blocking;

d) concatenation;

e) (N)-data-transmission;

f) (N)-entity;

g) (N)-facility;

h) flow control;

i) (N)-function;

j) (N)-interface-data-unit;

k) (N)-layer;

l) open system;

m) peer entities;

n) (N)-protocol;

o) (N)-protocol-control-information;

p) (N)-protocol-data-unit;

q) (N)-relay;

r) routing;

s) segmenting;

t) sequencing;

u) (N)-service;

v) (N)-service-access-point;

w) (N)-service-data-unit;

x) (N)-user-data.

2.2 For the purpose of this addendum, the following definitions also apply :

2.2.1 (N)-connection : An association established by the (N)-layer between two or more (N + 1)-entities for the transfer of data, which provides explicit identification of a set of (N)-data-transmissions and agreement concerning the (N)-data-transmission services to be provided for the set.

NOTE — This definition of (N)-connection is a refinement of the definition given in ISO 7498; it does not change it.

2.2.2 (N)-connection-mode transmission : (N)-data-transmission in the context of an (N)-connection.

2.2.3 (N)-connectionless-mode transmission : (N)-data-transmission not in the context of an (N)-connection and not required to maintain any logical relationship between (N)-service-data-units.

3 Notation

The layer notation is the same as that defined in ISO 7498.

4 Introduction to Open Systems Interconnection (OSI)

This clause makes no additions to clause 4 of ISO 7498.

5 Concepts of a layered architecture

5.1 Introduction

This addendum makes no additions to 5.1 of ISO 7498.

5.1 Principles of layering

5.2.1 This sub-clause complements 5.2 of ISO 7498.

5.2.2 In order for (N + 1)-entities to be able to communicate using an (N)-connection-mode service or an (N)-connectionless-mode service it is essential that a pre-arranged association exists between them, constituted by the pre-knowledge which it is essential that each (N + 1)-entity has of the others in order at least to initiate the use of the service. This association is established in ways which are not detailed in this addendum or in ISO 7498 and comprises four elements :

a) knowledge of the addresses of the peer entities involved;

b) knowledge of a protocol agreed by the peer entities for use at least to initiate communication;

c) knowledge of the availability for communication of the peer-entities;

d) knowledge of the quality of service available from the (N)-service.

NOTE – The pre-knowledge constituting a pre-arranged association can be acquired in many ways; some examples are listed below :

a) from information acquired manually when contracts are exchanged with a service provider;

b) from information which a network administration may provide in a directory or enquiry database;

c) from information that may be learned from previous instances of communication;

d) from information that may be provided dynamically through the operation of management protocols.

The total pre-knowledge constituting a pre-arranged association is likely to be acquired by a combination of the above.

5.2.3 An (N + 1)-entity provides no information to an (N)-connectionless-mode service about the logical relationships between (N)-service-data-units, apart from the source and destination (N)-service-access-point-addresses.

5.2.4 From the point of view of the (N + 1)-entity this means that it is not able to require the (N)-service to apply a particular function to a sequence of (N)-service-data-units sent by it. However, from the point of view of the (N)-layer, this does not imply any constraint on the functions which support the service.

5.2.5 An (N)-layer may offer a connection-mode service, a connectionless-mode service, or both, to the (N + 1)-layer, using the service or services provided by the (N – 1)-layer.

5.2.6 Both the (N)-connection-mode service and the (N)-connectionless-mode service are characterized by the facilities which they offer to, and the quality of service seen by, the (N + 1)-entities. For both the (N)-connection-mode service and the (N)-connectionless-mode service, functions may be provided by the (N)-layer to enhance the facilities offered to, and the quality of service seen by, the (N + 1)-entities over those which are offered to the (N)-layer by the (N – 1)-layer and, if necessary, to convert between one mode of service and another.

5.3 Communication between peer-entities

5.3.1 This sub-clause complements 5.3 of ISO 7498.

5.3.2 (N + 1)-entities can communicate using an (N)-connectionless-mode service provided that there is a pre-arranged association between them providing knowledge about each other which allows them to do so. This knowledge should allow the locations of the (N + 1)-entities to be determined, it should determine the correct interpretation of (N)-service-data-units by a receiving (N + 1)-entity, and it may define the rates of transfer, rates of response, and the protocol in use between the entities. The knowledge may result from prior agreement between the (N + 1)-entities concerning the parameters, formats and options to be used.

NOTE – Data transfer using an (N)-connection-mode service involves the establishment of an (N)-connection prior to the data transfer, setting up, dynamically, an association between the (N + 1)-entities and the (N)-connection-mode service in addition to the association identified in 5.1. This association involves elements which are not part of the pre-arranged association described in 5.1, in particular :

a) knowledge of the willingness of the peer entity or entities to undertake a specific communication, and of the willingness of the underlying service to support it; and

b) the ability for the peer entities to negotiate and renegotiate the characteristics of the communication.

5.3.3 (N + 1)-entities may require prior knowledge of the facilities offered by the service and the quality of service which they can expect to receive from it in order to choose an (N + 1)-protocol to be used for communication over an (N)-connectionless-mode service.

5.3.4 There are instances where the connectionless-mode service provided by the (N)-layer does not provide direct access between all of the (N)-service-access-points supported by the layer. Connectionless-mode transmission can still occur between these service-access-points if one or more (N + 1)-entities provide a relay (see figure 1 which complements figure 6 in ISO 7498). The fact that an (N)-connectionless-mode transmission is relayed by one or more (N + 1)-entities is known neither by the (N)-layer nor by the (N + 2)-layer.

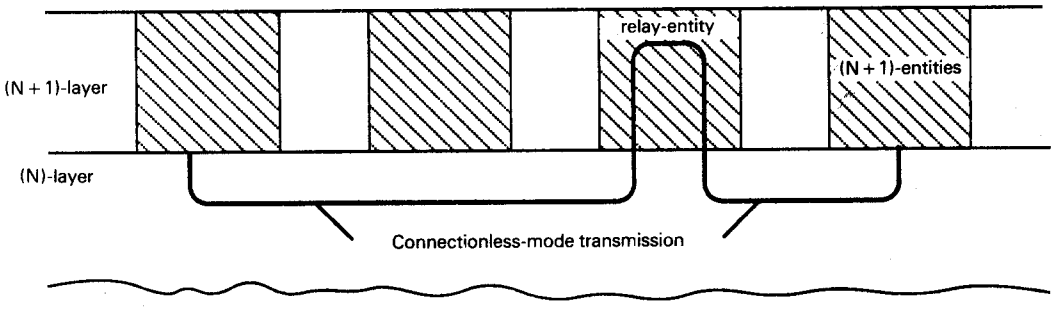


Figure 1 – Connectionless-mode transmission through a relay

5.4 Identifiers

This addendum makes no additions to 5.4 of ISO 7498.

5.5 Properties of service-access-points

5.5.1 This sub-clause complements 5.5 of ISO 7498.

5.5.2 An (N)-service-access-point may support :

a) (N)-connection-mode services only;

b) (N)-connectionless-mode services only; or

c) (N)-connection-mode services and (N)-connectionless-mode services concurrently.

5.5.3 A single (N+1)-entity may be using concurrently several (N)-connections and an (N)-connectionless-mode service through one or more (N)-service-access-points to which it is attached.

5.5.4 (N+1)-entities distinguish between instances of the (N)-connectionless-mode services and the (N)-connection-mode services offered concurrently through the same (N)-service-access-point by the uniqueness of the interactions prescribed for these services.

5.6 Data-units

This addendum makes no additions to 5.6 of ISO 7498.

5.7 Elements of layer operation

5.7.1 Introduction

This sub-clause complements 5.7 and 5.8 of ISO 7498, to which it relates.

5.7.2 Control of connectionless-mode transmission

5.7.2.1 This sub-clause is specific to this addendum and has no equivalent in ISO 7498.

5.7.2.2 All the information required by an (N)-connectionless-mode service to deliver an (N)-service-data-unit (destination address, quality of service required, options, etc.) is presented to it with the (N)-service-data-unit in a single logical service access by the sending (N+1)-entity.

5.7.2.3 All information related to an (N)-service-data-unit, together with the (N)-service-data-unit itself, is received from the (N)-service in a single logical service access by the receiving (N+1)-entity.

5.7.2.4 To provide the (N)-connectionless-mode service, the (N)-layer performs functions as described in 5.1. These functions are supported by (N)-protocols.

5.7.2.5 If an (N)-service-data-unit cannot be accepted by an (N+1)-entity at the time of its arrival at an (N)-service-access-point, the (N+1)-entity may apply interface flow control (see 5.7.3.2). This may result in the discarding of the (N)-service-data-unit by the (N)-service provider or, where flow control is provided, in the exercise of interface flow control at the sending (N)-service-access-point by the (N)-service provider.

5.7.2.6 An (N)-connectionless-mode service may allow the transmission of copies of an (N)-service-data-unit to a number of destination (N)-service-access-points. (N)-service-data-units transmitted from a number of source (N)-service-access-points can be received at one destination (N)-service-access-point. The (N)-layer does not assume any logical relationship between these (N)-service-data-units.

5.7.2.7 No (N)-protocol-control-information is exchanged between (N)-entities concerning the mutual willingness of the (N+1)-entities to exchange data using an (N)-connectionless-mode service.

NOTES

1 The specific interface mechanism employed by a particular implementation of a connectionless-mode service may involve more than one interface exchange to accomplish the single logical service access necessary to initiate a connectionless-mode transmission. However, this is a local implementation detail.

2 The transmission of each (N)-service-data-unit by an (N)-connectionless-mode service should be entirely self-contained. All the addressing and other information required by the (N)-layer to deliver the (N)-service-data-unit to its destination should be included in the service access for each transmission.

3 It is a basic characteristic of a connectionless-mode service that no negotiation of the parameters for a transmission takes place at the time the service is accessed and no dynamic association is set up between the parties involved. However, considerable freedom of choice can be preserved by allowing most parameter values and options (such as transfer rate, acceptable error rate, etc.) to be specified at the time the service is accessed. In a given implementation, if the local (N)-subsystem determines immediately (from information available to it locally) that the requested transmission cannot be performed under the conditions specified, it may abort the transmission, returning an implementation specific error message. If the same determination is made later, after the service access has been completed, the transmission is abandoned, since the (N)-layer is assumed not to have the information necessary to take any other action.

5.7.3 Transfer of data

5.7.3.1 General principles

5.7.3.1.1 This sub-clause complements 5.7.6.1, 5.7.6.2 and 5.7.6.3 of ISO 7498.

5.7.3.1.2 Control information and user data are transferred between (N)-entities in (N)-protocol-data-units. An (N)-protocol-data-unit is a unit of data specified in an (N)-protocol and contains (N)-protocol-control-information and possibly (N)-user-data.

5.7.3.1.3 (N)-protocol-control-information is transferred between (N)-entities using an (N−1)-service. (N)-protocol-control-information is any information that supports the joint operation of (N)-entities. (N)-user-data is passed transparently between (N)-entities using an (N−1)-service.

5.7.3.1.4 An (N)-protocol-data-unit has a finite size, which may be limited by the (N – 1)-service-data-unit size and by the capabilities of the (N)-protocol. (N)-protocol-data-units are mapped into (N – 1)-service-data-units. The interpretation of an (N)-protocol-data-unit is defined by the (N)-protocol in effect for the (N)-connectionless-mode service.

5.7.3.1.5 An (N)-service-data-unit is transferred between an (N + 1)-entity and an (N)-entity, through an (N)-service-access-point, in the form of one or more (N)-interface-data-units. Each (N)-service-data-unit is transferred as (N)-user-data in one or more (N)-protocol-data-units.

5.7.3.2 Flow control

5.7.3.2.1 This sub-clause complements 5.7.6.4 of ISO 7498.

5.7.3.2.2 If flow control functions are provided, they can operate only on protocol-data-units and interface-data-units.

5.7.3.2.3 Two types of flow control are identified :

a) Peer flow control which regulates the rate at which (N)-protocol-data-units are sent between (N)-entities supporting (N)-connectionless-mode transmission between pairs of (N)-service-access-points or among a set of (N)-service-access-points. Peer flow control requires protocol definitions and is based on protocol-data-unit size; and

b) (N)-interface flow control, which regulates the rate at which (N)-interface-data-units are passed between an (N + 1)-entity and an (N)-entity that supports an (N)-connectionless-mode service. (N)-interface flow control is based on (N)-interface-data-unit size.

5.7.3.3 Segmenting, blocking and concatenation

5.7.3.3.1 This sub-clause complements 5.7.6.5 of ISO 7498.

5.7.3.3.2 Data units in different layers of the Reference Model are not necessarily compatible in size. Segmenting, blocking or concatenation may be necessary.

5.7.3.3.3 Segmenting in an (N)-layer requires the implementation of a segmenting-reassembly protocol between (N)-entities that support (N)-connectionless-mode transmission between pairs of (N)-service-access-points.

5.7.4 Quality of service

5.7.4.1 Introduction

This sub-clause complements 5.7.7 of ISO 7498.

5.7.4.2 Quality of service parameters

5.7.4.2.1 This sub-clause is specific to this addendum and has no equivalent in ISO 7498.

5.7.4.2.2 An (N)-connectionless-mode service is characterized by two groups of quality of service parameters. The parameters in the first group are defined entirely by the behaviour of a single (N)-data-transmission and are the same as those defined for the (N)-connection-mode service. These parameters are

a) expected transmission delay;

b) probability of corruption;

c) probability of loss or duplication;

d) probability of wrong delivery;

e) cost; and

f) protection from unauthorized access.

5.7.4.2.3 The parameters in the second group apply for multiple (N)-data-transmissions between pairs of (N)-service-access-points. These parameters are

a) expected throughput; and

b) probability of out-of-sequence delivery.

5.7.4.2.4 The values of the parameters in both groups at a given source (N)-service-access-point may vary with the destination (N)-service-access-point.

5.7.4.2.5 Sequencing, acknowledgement and error detection and notification functions may be used in the (N)-layer in order to enhance the quality of service offered by the (N)-connectionless-mode service over that offered by the (N – 1)-connectionless-mode service.

5.7.4.3 Sequencing

5.7.4.3.1 This sub-clause complements 5.7.6.6 of ISO 7498.

5.7.4.3.2 The (N – 1)-services provided by the (N – 1)-layer of the OSI architecture may not guarantee delivery of data between pairs of (N – 1)-service-access-points in the same order as it was submitted. If the (N)-layer provides a higher probability of in-sequence delivery than is offered by the (N – 1)-services, sequencing mechanisms must be present in the (N)-layer.

5.7.4.4 Acknowledgement

5.7.4.4.1 This sub-clause complements 5.7.7.1 of ISO 7498.

5.7.4.4.2 An acknowledgement function may be used by peer (N)-entities supporting (N)-connectionless-mode transmission between pairs of (N)-service-access-points to obtain a higher probability of detecting protocol-data-unit loss than is provided by the (N – 1)-layer.

5.7.4.5 Error detection and notification

5.7.4.5.1 This sub-clause complements 5.7.7.2 of ISO 7498.

5.7.4.5.2 Error detection and notification functions may be supported by protocols used between (N)-entities supporting (N)-connectionless-mode transmission between pairs of (N)-service-access-points to provide a higher probability of both protocol-data-unit error detection and data corruption detection than is provided by the (N – 1)-service.

5.7.4.5.3 The error detection or notification function is only as reliable as the (N)-connectionless-mode service itself. While the (N)-service provider may attempt to provide a notification upon detection of data corruption or protocol-data-unit loss, misdelivery, etc., it cannot be relied upon to be capable of doing so for every instance of error detection.

5.8 Routing

5.8.1 This sub-clause complements 5.8 of ISO 7498.

5.8.2 An (N)-routing function enables an (N – 1)-connectionless-mode transmission to be relayed by a chain of (N)-entities. The fact that (N)-routing is being used is known by neither the (N + 1)-layer nor the (N – 1)-layer.

5.9 Management

5.9.1 This sub-clause complements 5.9 of ISO 7498.

5.9.2 Systems and layer management provide for initialization action to establish support for connectionless-mode services between systems.

5.9.3 Management facilities may be provided to allow characteristics of the nature, quality and type of connectionless-mode service provided by a layer to be conveyed to the next higher layer prior to the invocation of that service. These facilities may provide this information either prior to any invocation of the service or at any time during a period when it is available.

5.10 The relationship between services provided at adjacent layer boundaries

5.10.1 This sub-clause is specific to this addendum and has no equivalent in ISO 7498.

5.10.2 There are no architectural constraints on any vertical combination of an (N)-layer providing one type of (N)-service (connection-mode or connectionless-mode) using the other type of (N – 1)-service. In principle the services at the two layer boundaries can be

a) both connection-mode services;

b) both connectionless-mode services;

c) the (N)-service a connection-mode service and the (N – 1)-service a connectionless-mode service;

d) the (N)-service a connectionless-mode service and the (N – 1)-service a connection-mode service.

5.10.3 In order to allow combinations c) and d) two architectural elements are required :

a) a function to provide an (N)-connection-mode service using an (N – 1)-connectionless-mode service; and

b) a function to provide an (N)-connectionless-mode service using an (N – 1)-connection-mode service.

NOTE – Of these functions, function a) requires significant protocol-control-information. For example, there is a need to identify the connection which is constructed, control its state and provide sequencing of service-data-units. Function b) requires little or no additional protocol-control-information, rather, it places constraints on the way in which the connection-mode service is used.

5.10.4 An (N)-relay can either

a) join two connection-mode services to yield a connection-mode service; or

b) join two connectionless-mode services to yield a connectionless-mode service.

5.10.5 Provision of a service of a given type using services of different types requires that one of the services (either the connection-mode or the connectionless-mode one, depending upon economic and technical factors) be first converted to the other using the functions which have been identified, an (N)-relay can then operate.

5.10.6 The basic cases of relaying are illustrated in figure 2. The use of conversions between services within a layer is not explicitly constrained by the Reference Model but, where several services are connected in tandem, the use of conversions would be ordered to minimize the number of conversions necessary to arrive at a given composite service.

5.10.7 Where an (N – 1)-connectionless-mode service is enhanced to provide an (N)-connection-mode service, a number of (N)-connections may be supported by (N – 1)-connectionless-mode transmission between the same (N – 1)-service-access-points.

5.10.8 Where an (N – 1)-connection-mode service is used to provide an (N)-connectionless-mode service, (N)-connectionless-mode transmission between a number of different (N)-service-access-points may be supported by the same (N – 1)-connection.

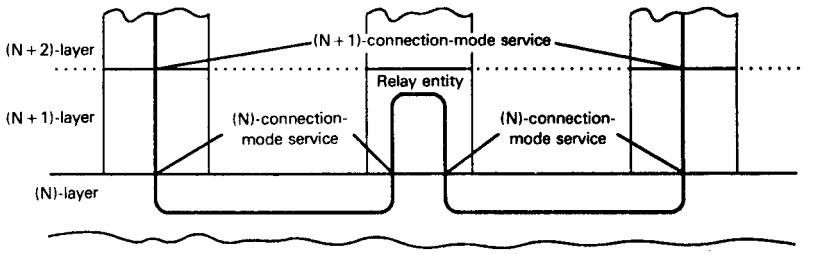
6 Introduction to the specific layers of the Reference Model

6.1 Introduction

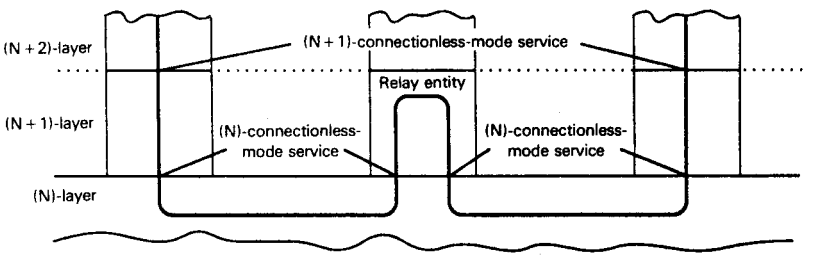
This clause complements clause 6 of ISO 7498.

6.2 General principles

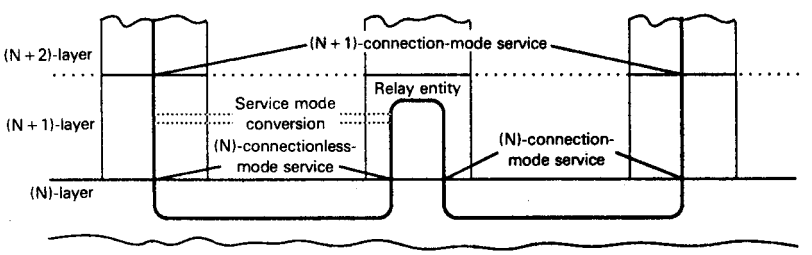
6.2.1 A connectionless-mode service for a particular layer may include none, some, or all of the detailed services and it may be characterized by none, some, or all of the quality of service parameters defined for the layer in ISO 7498 and in 5.7.4 of this addendum (for example error notification).



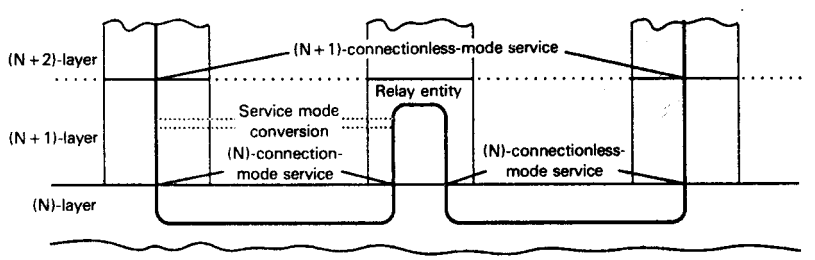
a) Relaying between two connection-mode services



b) Relaying between two connectionless-mode service



c) Connection- and connectionless-mode services yielding a connection-mode service



d) Connection- and connectionless-mode services yielding a connectionless-mode service

Figure 2 — The basic cases of relaying involving connection-mode and connectionless-mode services

6.2.2 The provision of connectionless-mode and connection-mode services in specific layers of the Reference Model and the characteristics of these services, together with the provision of functions providing for conversion within a layer between one mode of service and another, should be such as to ensure that it is possible to determine whether or not interworking between open systems is possible. In order to maximize the possibility of interworking and to limit protocol complexity, there is a restriction on the number of layers within which conversion between one mode of service and the other may take place.

6.2.3 For each layer, the sub-clauses of clause 7 identify those items which have relevance to the connectionless-mode service provided by that layer. Detailed descriptions are given only of those items which are specific to the connectionless-mode service; detailed descriptions of other items can be found in ISO 7498.

6.2.4 The basic (N)-connectionless-mode service is a service which meets the following conditions :

a) it is not required to exhibit any minimum values of the quality of service measures, in particular the sequence of (N)-service-data-units need not be maintained; and

b) it is not required to exhibit peer flow control.

Any (N)-connectionless-mode service definition should allow this basic service.

6.2.5 Since the basic service is not required to maintain the sequence of (N)-service-data-units, there is no requirement for any (N)-layer to provide sequencing functions. However, in real implementations the characteristics of the underlying medium or of real subnetworks may offer a high probability of in-sequence delivery and this may be reflected in the characteristics of the connectionless-mode services offered by higher layers.

6.2.6 Although no architectural limit on the size of (N)-service-data-units is defined in clause 5, in order to limit protocol complexity, segmentation and reassembly functions are not provided in layers above the Network Layer. Consequently, the size of service-data-units in layers above the Network Layer is limited by the size of the service-data-unit provided by the service of the layer below and by the size of the protocol-control-information for the layer itself.

6.3 Combinations of connection-mode and connectionless-mode service

6.3.1 As indicated in 6.2.2, there is a restriction on the number or layers within which conversion from a connection-mode service to a connectionless-mode service (or vice-versa) may take place. This restriction applies to the layers as follows :

a) Special considerations apply to the Physical and Data Link layers. Connection-mode and connectionless-mode services are not differentiated for the Physical Layer. The services of the Physical Layer are determined by the characteristics of the underlying medium and are too diverse to allow categorization into connection-mode and connectionless-mode operation. Functions in the Data Link Layer must convert between the services offered by the Physical Layer and the type of Data Link service needed.

b) Conversion may be provided in the Network Layer to support a Network service of a given mode over a Data Link or subnetwork service of the other mode. This, in conjunction with relaying, provides an end-to-end Network service of a given mode over concatenated subnetworks and/or Data Link services of either mode (see 5.10). Support of such conversions, where they are necessary to provide a given mode of Network service, is a requirement of OSI standards.

c) Conversion may be provided in the Transport Layer on condition that this makes use of only limited additional protocol functions over those required to support a given mode of Transport service over the same mode of Network service. Since relaying is not permitted in the Transport Layer, such conversions can only be applied between end-systems. Support for such conversions is not a requirement of OSI standards.

d) Conversion at the Session Layer and above is not permitted.

NOTE — It is not possible (since a Transport Protocol operates between end-systems) for a Transport Protocol to provide the Transport Service in an instance of communication between two end-systems utilising (in that instance of communication) different modes of Network Service.

6.3.2 It follows from these restrictions that

a) A real system which is fully open as defined in 4.1.2 of ISO 7498 shall support a given mode of Transport service over a Network service of the same mode (utilising conversion within the Network Layer if necessary); such a system may, in addition, provide conversion in the Transport Layer.

b) A real system which only supports a given mode of Transport service by providing conversion in the Transport Layer from a Network service of the other mode is not fully open as defined in 4.1.2 of ISO 7498, since such a system would be incapable of communicating with a system which only supports the given mode of Transport service over a Network service of the same mode.

NOTE — The restriction that a given mode of Transport Service has to be supported by the same mode of Network Service is applied so that systems may communicate without requiring prior agreement on the mode of Network Service to be used. Where prior agreement exists this restriction need not apply, although the requirements for systems to be fully open are as stated in 6.3.2a).

7 Detailed description of the resulting OSI architecture

7.1 Application Layer

7.1.1 Introduction

This sub-clause complements 7.1 of ISO 7498.

7.1.2 Purpose

The purpose of the Application Layer is the same as that defined in 7.1.2 of ISO 7498.

7.1.3 Services provided to application-processes

7.1.3.1 Where they are appropriate for connectionless-mode operation, equivalent services are provided by the Application Layer to application-processes to those provided for connection-mode operation.

7.1.3.2 In addition to information transfer, such services may include, but are not limited to, the following :

a) identification of intended communication partners;

b) establishment of authority to communicate;

c) authorization of intended communication partners;

d) determination of the acceptance quality of service; and

e) identification of constraints on data syntax.

7.1.4 Functions within the Application Layer

7.1.4.1 The Application Layer contains all functions which are necessary to support the connectionless-mode services offered to application-processes and which are not already performed by lower layers.

7.1.4.2 In particular, application-entities maintain, as part of the pre-knowledge necessary in order to communicate, information on the use of connection-mode and/or connectionless-mode transmission by peer entities with which they may need to communicate.

7.2 Presentation Layer

7.2.1 Introduction

This sub-clause complements 7.2 of ISO 7498.

7.2.2 Purpose

The purpose of the Presentation Layer is the same as that defined in 7.2.2 of ISO 7498.

7.2.3 Services provided to the Application Layer

The Presentation Layer enhances the session-services with the following facilities, invoked for each instance of connectionless-mode transmission :

a) transformation of syntax; and

b) selection of syntax.

7.2.4 Functions within the Presentation Layer

The Presentation Layer performs the following functions to provide the above services :

a) data transfer;

b) identification of syntax; and

c) transformation of syntax including data transformation and formatting and special purpose transformations (for example compression).

7.3 Session Layer

7.3.1 Introduction

This sub-clause complements 7.3 of ISO 7498.

7.3.2 Purpose

The only purpose of the Session Layer for connectionless-mode communication is to provide a mapping of transport-addresses to session-addresses.

7.3.3 Services provided to the Presentation Layer

The Session Layer provides the following services :

a) connectionless-mode transmission using the transport-connectionless-mode service; and

b) exception reporting.

NOTE – The connectionless-mode Session Service and Protocol are under study. This study may result in the identification of requirements for additional functions in the Session Layer.

7.3.4 Functions within the Session Layer

The Session Layer provides a one-to-one mapping of session-connectionless-mode-transmissions onto transport-connectionless-mode-transmissions.

7.4 Transport Layer

7.4.1 Introduction

This sub-clause complements 7.4 of ISO 7498.

7.4.2 Purpose

The purpose of the Transport Layer is the same as that defined in 7.4.2 of ISO 7498.

7.4.3 Services provided to the Session Layer

The Transport Layer provides a connectionless-mode service which maps a request for transmission of a transport-service-data-unit onto a request to the connectionless-mode network service.

7.4.4 Functions within the Transport Layer

The Transport Layer provides the following functions to support connectionless-mode transmission :

a) mapping between transport-addresses and network-addresses;

b) mapping end-to-end transport-connectionless-mode transmissions onto network-connectionless-mode transmissions;

NOTE – There may be specific situations where performing conversion from connection-mode to connectionless-mode operation in the Transport Layer can be justified and may thus be permitted provided that this requires only limited extensions to existing protocols. In such cases it is accepted that communication using such conversions can only take place between end systems supporting them (see 6.3.1).

c) end-to-end error detection and monitoring of the quality of service;

d) transport-service-data-unit delimiting; and

e) supervisory functions.

7.5 Network Layer

7.5.1 Introduction

This sub-clause complements 7.5 of ISO 7498.

7.5.2 Purpose

In addition to the purpose defined in 7.5.2 of ISO 7498, the Network Layer provides the functional and procedural means for connectionless-mode transmission among transport-entities and, therefore, provides to the transport-entities independence of routing and relay considerations associated with connectionless-mode transmission.

7.5.3 Services provided to the Transport Layer

The following services and elements of services are provided by the Network Layer, operating between pairs of network service-access points :

a) transmission of network-service-data-units of a defined maximum size;

b) quality of service parameters; and

c) local error notification.

7.5.4 Functions within the Network Layer

7.5.4.1 The Network Layer provides the following functions to support network-connectionless-mode transmission :

a) mapping between network-addresses and data-link-addresses;

b) mapping network-connectionless-mode transmissions onto data-link-connectionless-mode transmissions;

c) converting from data-link-connection-mode service to network-connectionless-mode service;

d) routing and switching;

e) segmenting and blocking;

f) error recovery; and

g) service selection.

7.5.4.2 The Network Layer also provides the functions necessary to enhance a data-link-connectionless-mode service to provide a network-connection-mode service.

7.6 Data Link Layer

7.6.1 Introduction

This sub-clause complements 7.6 of ISO 7498.

7.6.2 Purpose

In addition to the purpose defined in 7.6.2 of ISO 7498, the Data Link Layer provides the functional and procedural means for connectionless-mode transmission among network-entities.

7.6.3 Services provided to the Network Layer

The following services and elements of services are provided by the Data Link Layer :

a) data-link-service-data-unit transmission;

b) local error notification; and

c) quality of service parameters.

7.6.4 Functions within the Data Link Layer

The functions in the Data Link Layer provide the services offered by the layer using the services offered by the Physical Layer.

7.7 Physical Layer

7.7.1 Introduction

This sub-clause complements 7.7 of ISO 7498.

7.7.2 Purpose

The purpose of the Physical Layer is the same as that given in 7.7.2 of ISO 7498.

7.7.3 Services provided to the Data Link Layer

The services provided by the Physical Layer are determined by the characteristics of the underlying medium and are too diverse to allow categorization into connection-mode and connectionless-mode operation.

7.7.4 Functions within the Physical Layer

The functions in the Physical Layer are determined by the characteristics of the underlying medium and are too diverse to allow categorization into connection-mode and connectionless-mode operation.

Printed at Simco Printing Press, Delhi, India