
British Standard

Reference model of Open Systems Interconnection

Part 1. Basic reference model (incorporating connectionless-mode transmission)

[ISO titles: Information processing systems — Open Systems Interconnection — Basic Reference Model and Addendum 1. Connectionless-mode transmission]

Modèle de référence pour l'interconnexion de systèmes ouverts
Partie 1. Modèle de référence de base (incorporant la transmission en mode sans connexion)

Basisreferenzmodell für die Kommunikation Offener Systeme
Teil 1. Basisreferenzmodell (einschließlich der verbindungslosen Übertragung)

Contents	Page		Page
National foreword	1	11 Relationship between (N)-service-data-unit, (N)-protocol-data-unit and (N – 1)-service-data-unit within a layer	17
Committees responsible	Back cover	12 Seven layer reference model and peer protocols	20
Reference Model		13 Communication involving relay open systems	20
0 Introduction	2	14 Application-entity	23
1 Scope and field of application	3	15 Several consecutive session-connections	27
2 Definitions	3	16 Several consecutive transport-connections	27
3 Notation	3	17 Association of transport-addresses and network-addresses	29
4 Introduction to Open Systems Interconnection (OSI)	3	18 Association of one network-address with several transport-addresses	29
5 Concepts of a layered architecture	5	19 Interconnection of a low quality subnetwork and a high quality subnetwork	33
6 Introduction to the specific OSI layers	19	20 Interconnection of an enhanced low quality subnetwork and a high quality subnetwork	33
7 Detailed description of the resulting OSI architecture	21	21 Interconnection of data-circuits within the Physical Layer	37
Annexes		22 Examples of physical-connections	37
A Brief explanation of how the layers were chosen	39	23 Examples of interconnection	38
B Alphabetical index to definitions	40	Addendum 1. Connectionless-mode transmission	
Figures		0 Introduction	42
1 Open systems connected by physical media	4	1 Scope and field of application	44
2 Basic elements of OSI	5	2 Definitions	44
3 Layering in co-operating open systems	6	3 Notation	44
4 (N+1)-entities in the (N+1)-layer communicate through the (N)-layer	7	4 Introduction to Open Systems Interconnection (OSI)	44
5 (N)-protocols between (N)-entities	8	5 Concepts of a layered architecture	44
6 Communication through a relay	9	6 Introduction to the specific layers of the Reference Model	48
7 Entities, service-access-points, and identifiers	10	7 Detailed description of the resulting OSI architecture	50
8 Hierarchical (N)-address-mapping	10		
9 Relationships among data-units	12		
10 An illustration of mapping between data-units in adjacent layers	12		

National foreword

This Part of BS 6568 has been prepared under the direction of the Information Systems Technology Standards Committee. It is identical with ISO 7498-1984 'Information processing systems — Open Systems Interconnection — Basic Reference Model', published by the International Organization for Standardization (ISO), and as updated by Addendum 1 to ISO 7498 'Connectionless-mode transmission' published in 1987.

The UK has actively participated through BSI Technical Committee IST/21 in the development of the international standard.

This Part of BS 6568 supersedes BS 6568 : 1984 and DD 125 : 1986 which are withdrawn. It has been renumbered as BS 6568 : Part 1 in anticipation of the international standard being issued as part of a standard on the Reference Model of Open Systems Interconnection with the following parts:

- Part 1. Basic Reference Model.
- Part 2. Security Architecture.
- Part 3. Naming, including Addressing.
- Part 4. Management Framework.

Parts 2, 3 and 4 have reached draft international standard status and are expected to be issued as Parts 2, 3 and 4 of BS 6568 when the international standards have been published. The texts of the draft international standards are available to members of the BITS (BSI Information Technology Services) scheme.

The ISO Subcommittee ISO/TC 97/SC 21, Information Processing Systems — Information Retrieval, Transfer and Management for Open Systems Interconnection, handles possible questions arising from the use of ISO 7498 and maintains a register of answers to questions of general interest. These answers are not interpretations of the standard (as understood by BS 0) but may be used in future revisions or extensions to the standard. This register is maintained as an ISO/TC 97/SC 21 committee document and copies are available in the UK to members of the BITS scheme. Any such questions arising out of BS 6568 : Part 1 should, in the first instance, be sent to the Secretary of IST/21, BSI, 2 Park Street, London W1A 2BS.

Up-to-date information on existing and projected British Standards publications for Open Systems Interconnection is included in the 'Basic List', published as part of the BSI Information Technology Services (BITS) initiative. For details contact: Marketing Department, BSI, Linford Wood, Milton Keynes MK14 6LE.

Terminology and conventions. The text of the international standard has been approved as suitable for publication as a British Standard without deviation. Some terminology and certain conventions are not identical with those used in British Standards; attention is drawn especially to the following.

Wherever the words 'International Standard' or 'ISO 7498' appear, referring to this standard, they should be read as 'British Standard' or 'BS 6568 : Part 1', respectively.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

British Standard

Reference model of Open Systems Interconnection

Part 1. Basic reference model (incorporating connectionless-mode transmission)

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 cooperation¹⁾ 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.

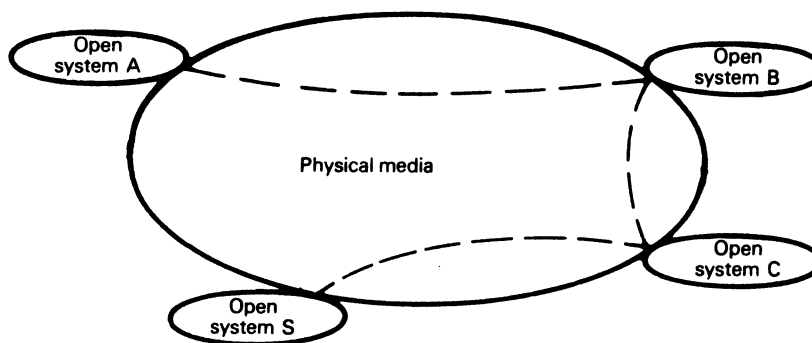


Figure 1 — Open systems connected by physical media

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 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.

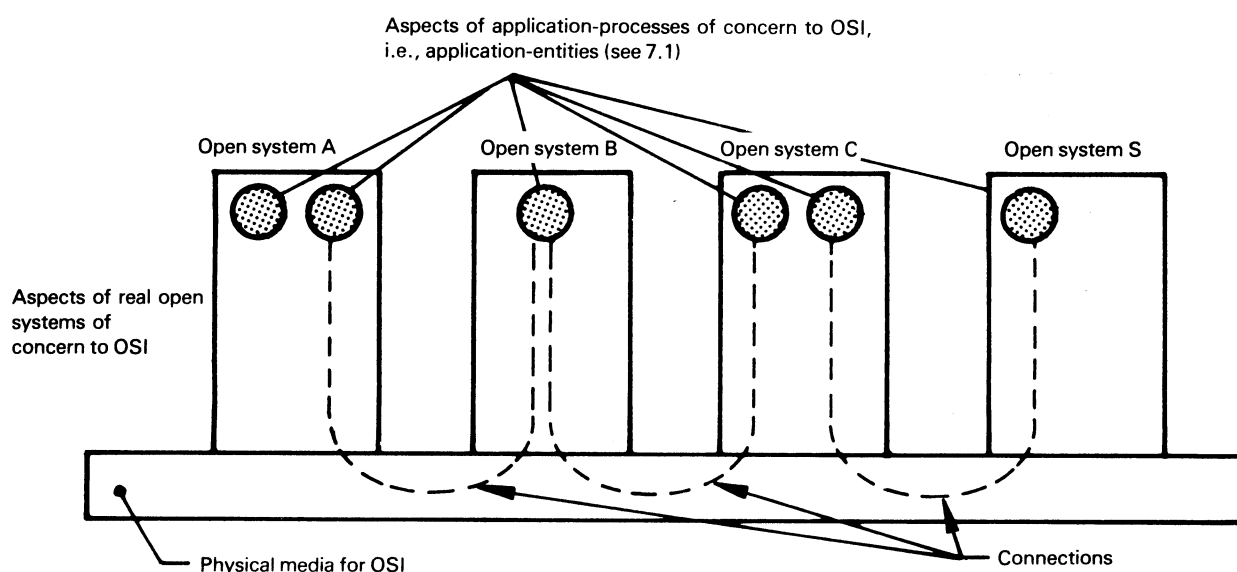


Figure 2 — Basic elements of OSI

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.

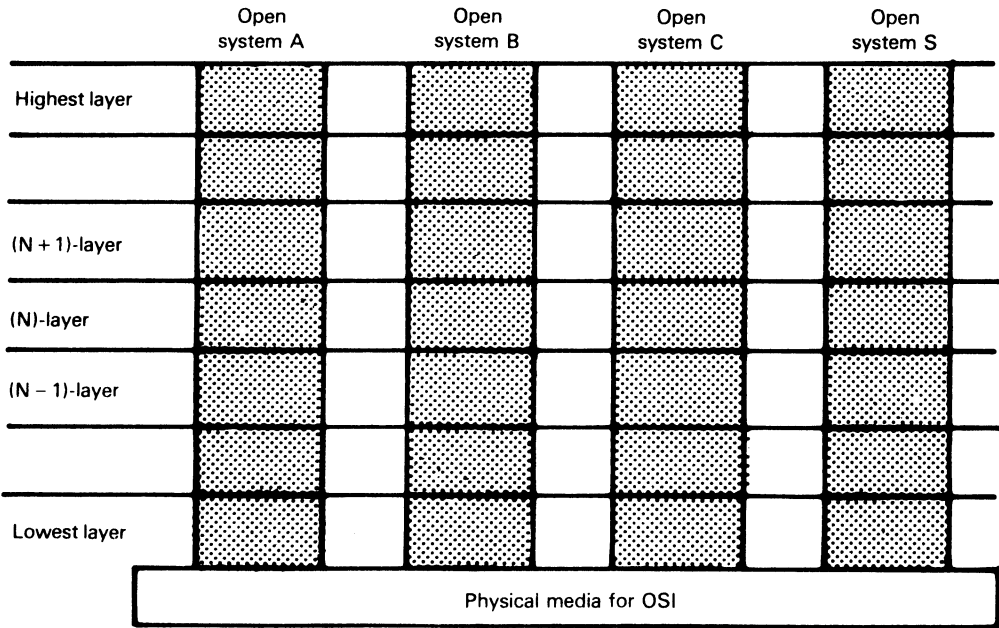


Figure 3 — Layering in co-operating open systems

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.

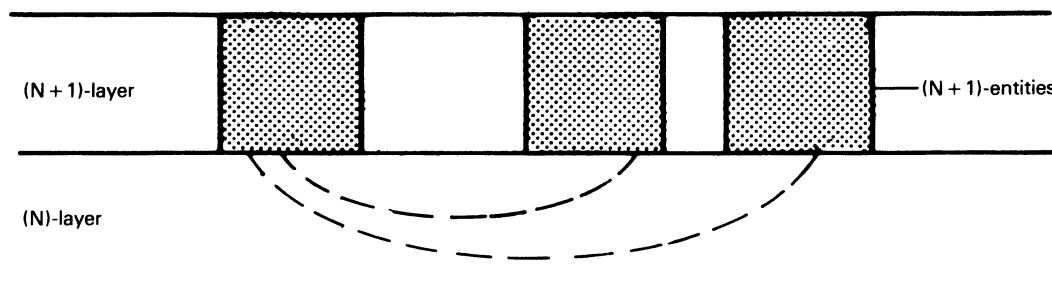


Figure 4 — (N+1)-entities in the (N+1)-layer communicate through the (N)-layer

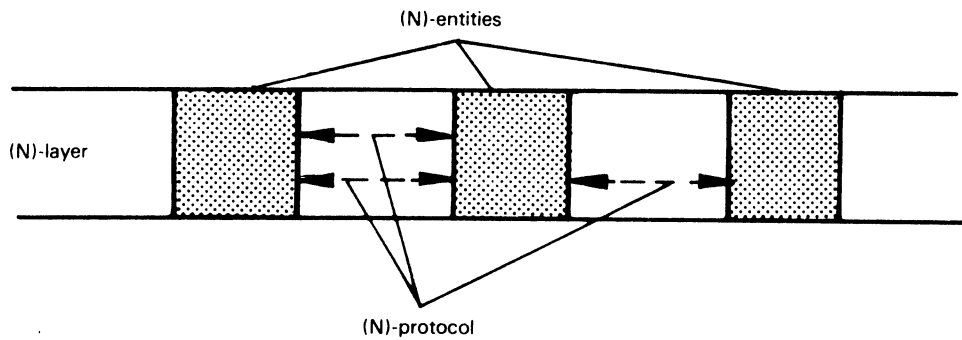


Figure 5 — (N)-protocols between (N)-entities

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.

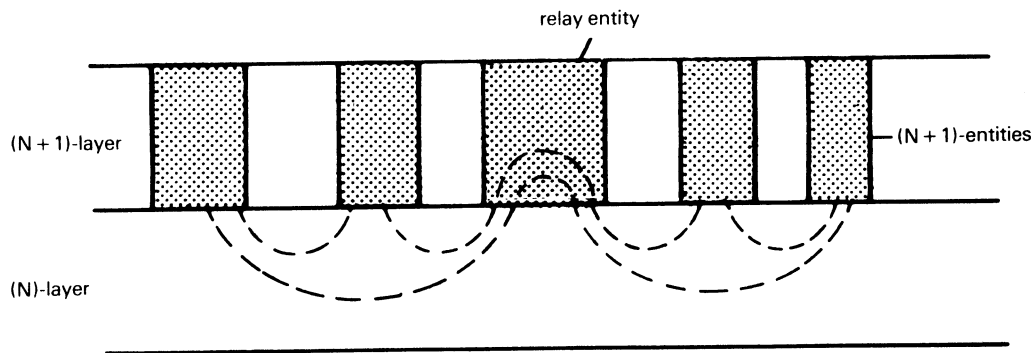


Figure 6 — Communication through a relay

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.

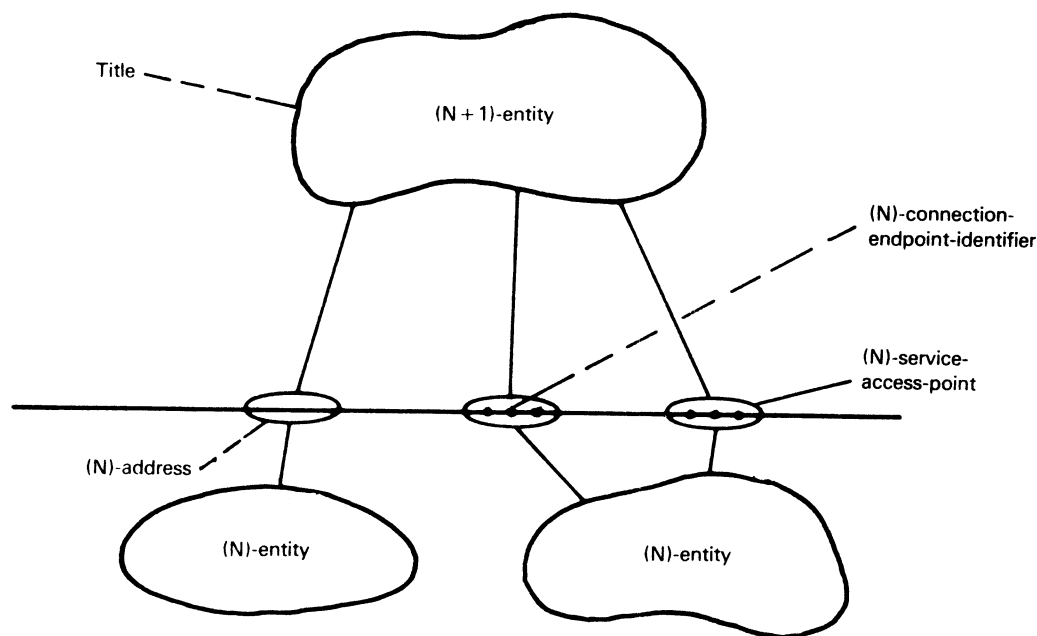
In this case, an (N)-address consists of two parts :

- 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

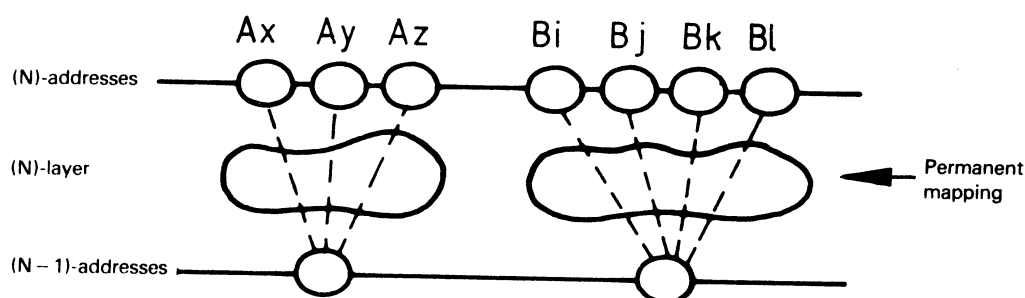


Figure 8 — Hierarchical (N)-address-mapping

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.

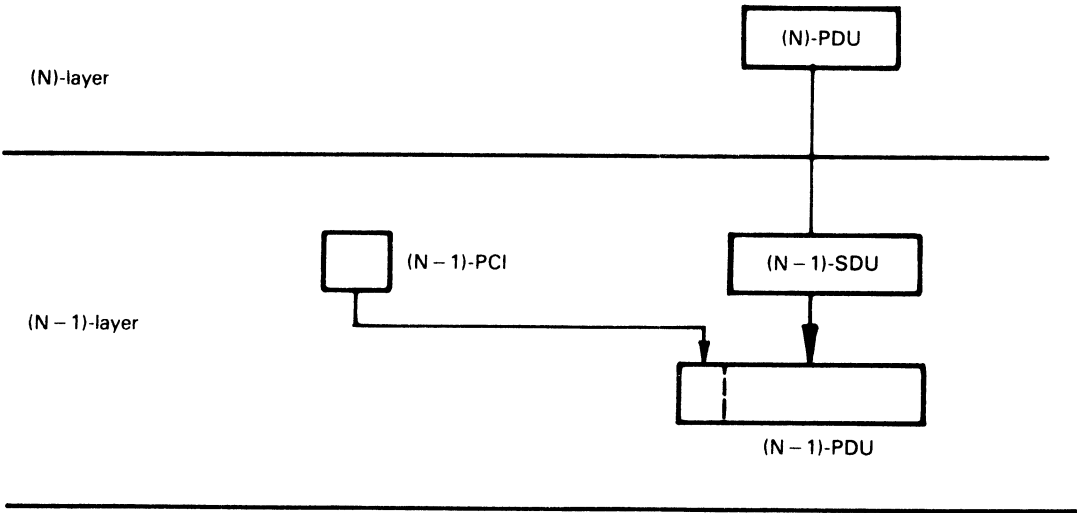
Except for the relationships defined in figures 9 and 10, there is no overall architectural limit to the size of data-units. There may be other size limitations at specific layers.

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 defined¹⁾ :

- 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 occurring 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.