AD-A107 644 NATIONAL COMMUNICATIONS SYSTEM WASHINGTON DC F/G 9/2
OPEN SYSTEMS INTERCONNECTION (OSI) REFERENCE MODEL (AUGUST 1981--ETC(U)
SEP 81
UNCLASSIFIED NCS-TIB-81-1A NL

1 OF 2
AD A
107644

2

NCS TIB 81-1A

NATIONAL COMMUNICATIONS SYSTEM

TECHNICAL INFORMATION BULLETIN
81-1A
OPEN SYSTEMS INTERCONNECTION (OSI)
REFERENCE MODEL (AUG 1981)
(ISO SECOND DRAFT PROPOSAL
(DP) 7498

SEPTEMBER 1981

APPROVED FOR PUBLIC RELEASE
DISTRIBUTION UNLIMITED

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entere

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NCS-TIB 81-1A	2. GOVT ACCESSION NO. AD-A109	3. RECIPIENT'S CATALOG NUMBER [illegible]
4. TITLE (and Subtitle) Open Systems Interconnection (OSI) Reference Model (Aug 1981) (ISO Second Draft Proposal (DP) 7498) September 1981		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS National Communications System Technology and Standards Office Washington, D.C. 20305		12. REPORT DATE September 1981
		13. NUMBER OF PAGES 105
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution unlimited, approved for public release

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Supersedes NCS-TIB-81-1, Open Systems Interconnection (OSI) Reference Model (Nov 1980) (ISO Draft Proposal (DP) 7498), January, 1981.(AD-A096086)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Open System Interconnection, Basic Reference Model, Open System, Layered Architecture, Telecommunication Media

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Reference Model was developed to establish a framework for the coordination of existing and future standards developed to facilitate the interconnection of information systems. The Model has a hierarchical organization of 7 architectual layers, based on the transfer of data by telecommunication means.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TECHNICAL INFORMATION BULLETIN 81-1A

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO)

SECOND DRAFT PROPOSAL 7498
OPEN SYSTEMS INTERCONNECTION (OSI) REFERENCE MODEL
AUGUST 1981

PROJECT OFFICER:

GEORGE W. WHITE
Office of NCS Technology
and Standards

APPROVED FOR PUBLICATION:

Marshall L Cain
MARSHALL L. CAIN
Assistant Manager
(Technology and Standards)
National Communications System

FOREWORD

Among the responsibilities assigned to the Office of the Manager, National Communications System, is the Management of the Federal Telecommunication Standards Program, which is an element of the overall General Services Administration's (GSA) Federal Standardization Program. Under this program, the NCS, with the assistance of the Federal Telecommunication Standards Committee, identifies, develops, and coordinates proposed Federal Standards which either contribute to the interoperability of functionally similar Federal telecommunication systems, or to the achievement of a compatible and efficient interface between computer and telecommunication systems. In developing and coordinating these standards, a considerable amount of effort is expended in initiating and pursuing joint standards development efforts with appropriate technical committees of the Electronic Industries Association, the American National Standards Institute, the International Organization for Standardization, and the International Telecommunication Union. This Technical Information Bulletin presents the latest version of the Reference Model for Open Systems Interconnection, prepared by Subcommittee 16, Open Systems Interconnection, of Technical Committee 97, Computers and Information Processing of the International Organization for Standardization. It has been prepared to inform Federal agencies of the latest developments in the architectural structure of open (distributed) systems. Comments, inputs, or statements of requirements, which could assist in the advancement of this work, are welcome and should be addressed to:

Office of the Manager
National Communications System
ATTN: NCS-TS
Washington, DC 20305
Telephone: (202) 692-2124

This document supercedes NCS TIB 81-1, January 1981.

BACKGROUND

Since its formation in 1977, Subcommittee 16, Open Systems Interconnection, of the International Organization for Standardization's (ISO) Technical Committee 97, Computers and Information Processing, has been developing a Reference Model for Open Systems Interconnection (OSI). The latest version of this Reference Model has been approved as a second Draft Proposal, and is being voted on by the ISO subcommittees on Data Communication and Open Systems Interconnection. A Draft Proposal is the first step in the process of making the Reference Model an International Standard.

The first Draft Proposal (DP) 7498 (see NCS TIB 81-1, January 1981) was circulated for letter ballot on December 3, 1980. All comments were reviewed by a Rapporteur's Group meeting in Paris, June 9-12, 1981. All comments were classified into those addressing content and those addressing form. Concerning comments on content, the Rapporteurs either resolved tnem by making a series of improvements in DP 7498 or by providing reasons why the comments could not be incorporated (see N 718). Concerning comments on form, these were resolved by editing the text of the DP in accordance with a series of principles established by the Rapporteurs.

The recommendation of the Rapporteurs' Meeting is that the revised text of DP 7498 be circulated to Member Bodies along with the Report of the Rapporteurs' Meeting for a two month letter ballot. The ballot closes October 21, 1981.

The concept of "open systems" is based on the need for any application process anywhere in the world to be able to interact with any other existing application process. In the United States, this concept has often been referred to as "distributed systems." At the first meeting of Subcommittee 16, however, it was felt that the latter term was too restrictive, and thus the term "Open Systems" was adopted.

The Reference Model was developed to establish a framework for the coordination of existing and future standards developed to facilitate the interconnection of information systems. At this time, the interconnection by telecommunication means has been considered. The use of other means of transporting data (such as the physical transport of computer magnetic tapes) is left for further study. The Model does not specify services and protocols for interconnection. It is designed to be the common basis for the planning and coordination of consistent standard protocols and families of protocols in a hierarchical manner. Implementation of standards development in consonance with the Model should allow the exchange of information among terminal devices, computers, people, application processes, etc., that are "open" to each other by virtue of their mutual use of applicable standards.

The Reference Model has a hierarchical organization of 7 architectual layers, based on the transfer of data by telecommunication means. The layer concept has been accepted by the American National Standards Institute Committee X3, Computers and Information Processing, as a part of their Master Plan. Study Group VII of he International Telegraph and Telephone

Consultative Committee (CCITT) has also accepted the seven layer model, as have other national and international standards development organizations. A great deal of further study is required to complete the work of defining each of the layers, the services furnished by each layer, and the functions performed by each layer, including all of the peer-to-peer protocols necessary to complete the application of the Reference Model to all communicating Application Processes. There are also issues which must be resolved, such as the minimum functionality of layers within the architecture, and consequently, the minimum protocol requirements for each of the layers. However, the present version of the Model is sufficiently defined so that it can be used as a tool in the development of new data communication networks and facilities. In order to give this document widespread distribution, we have made it an Annex to this NCS Technical Information Bulletin. Comments are solicited prior to January 8, 1982. These comments will be forwarded to the American National Standards Technical Committees on Data Communication and Open Systems Interconnection, which are developing the US position on the Reference Model, as well as recommended changes to progress the model as an international (ISO) standard.

NCS TIB 81-1A

APPENDIX

ISO SECOND DRAFT PROPOSAL

OPEN SYSTEMS INTERCONNECTION (OSI) REFERENCE MODEL

AUGUST 1981

american national standards institute, inc.
1430 broadway, new york, n.y. 10018
(212) 354-3300

ISO/TC 97/SC 16 N 735
August 19, 1981

ISO
INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION

ISO/TC 97/SC 16
OPEN SYSTEM INTERCONNECTION
Secretariat: USA (ANSI)

ERRATA SHEET TO SECOND DP 7498 (97/16 N 719)

Since DP 7498 was sent to you, we have discovered several typographical errors in the document. These errors have been compiled in the errata listed below:

Section 4.3, page 10, last sentence on page should read: "Therefore real open systems need not be ..."

Section 5.2.1.2, page 13 should read: "A well defined subdivision of the OSI architecture, constituted by subsytems of the same rank (N)."

Section 5.2.1.9, page 14 : Delete "and formats." The sentence now reads : "The access means by which (N)-services are provided by an (N)-entity to an (N+1)-entity."

Section 5.3.1.6, page 17 : Last word should be "connections".

Section 5.3.1.7, page 17 : Change "from" to "into". The sentence should now read: "That (N)-entity that enters (N-1)-service-data-units into an (N-1)-connection."

Section 5.7.6.5, page 41, 2nd paragraph: Change "peer form" to "perform". The sentence should now read: "Conversely, it may be necessary to perform blocking ..."

Section 5.8, page 44, 2nd sentence: "The fact that communication is being routed by intermediate (N)-entities is known neither by the Lower Layers nor by the higher Layers."

Section 6.1, page 49: "... derived, by making specific choices for the Layers and their contents."

Section 7.2.1.3, page 58: "The syntactic view of data"

Section 7.3.1.1 and Section 7.3.1.2, page 63: Change "explicitely" to "explicitly".

Section 7.4.3.2, page 74: Change "uses" to "users". The sentence should now read: "... is terminated and the transport users are notified."

Section 7.5.1.2, page 80, line 2: Change "is" to "in". The sentence should now read: "... in the Network Layer in providing a network-connection."

Section 7.5.2, page 80, 2nd paragraph, last word: Change "connection" to "connections". The sentence should now read: "... data-link-connections to provide network-connections."

Section 7.6.3.7, page 93, line 6 : Change "data-unit" to "data-units". The sentence should now read: "... misdelivery of data-link-service-data-units, and ..."

SECOND

DRAFT PROPOSAL ISO/DP 7498	
date 1981- August 06	ISO/TC 97/SC 16 N 719
supersedes document ISO/TC 97/SC 16 N 537 Rev.	

THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. SHOULD NOT BE USED FOR REFERENCE PURPOSES

Secretariat

American National Standards Institute
1430 Broadway
New York, New York 10018

Circulated to P- and O-members, technical committees and international organizations in liaison for :
- discussion at
- comments by
- voting by October 21, 1981

Title

DATA PROCESSING - OPEN SYSTEMS INTERCONNECTION - BASIC REFERENCE MODEL

Introductory note

First Draft Proposal (DP) 7498 was circulated for letter ballot on December 3, 1980. A summary of voting was circualted in document N 627. All comments were reviewed by a Rapporteur's Group meeting in Paris, June 9-12, 1981. All comments were classified into those addressing content and those addressing form. Concerning comments on content, the Rapporteurs either resolved them by making a series of improvements in DP 7498 or by providing reasons why the comments could not be incorporated (see N 718). Concerning comments on form, these were resolved by editing the text of the DP in accordance with a series of principles established by the Rapporteurs (N 718/Part 4).

The recommendation of the Rapporteurs' Meeting is that the revised text of DP 7498 be circulated to Member Bodies along with the Report of the Rapporteurs' Meeting for a two month letter ballot.

Data processing - Open Systems Interconnection - Basic Reference Model

CONTENTS Page

Page

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 developments 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.

"Openness" does not imply any particular systems implementation, technology or interconnection means, but rather 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 to serve as an implementation specification, nor as a basis for appraising the conformance of actual implementations, nor 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 developing standards for each layer of the Open Systems Interconnection Reference Model.

The Reference Model allows sufficient flexibility so that, as technology and user demands expand, the Reference Model can accommodate such advancements. This flexibility is also intended to allow the phased transition from existing implementations to Open Systems Interconnection standards.

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

While the scope of the general architectural principles required for Open Systems Interconnection is very broad, it is the primary intent of this International Standard to consider systems comprising terminals, computers and associated devices and the means for transferring information between such systems. Other aspects of Open Systems Interconnection requiring attention are described briefly (see 4.2).

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

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

This International Standard provides a description of the Reference Model of Open Systems Interconnection. It is developed in stages:

Clause 4 establishes the reasons for Open Systems Interconnection, defines what is being connected, and the scope of the interconnection;

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;

Clause 7 provides the description of the specific layers.

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

0.2 Related OSI standards

Concurrently with the preparation of this standard, work is in progress within the International Standards Organization 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) Session Layer services and protocols;

e) Transport Layer services and protocols;

f) Network Layer services and protocols,

g) Data Link Layer services and protocols;

h) Physical Layer services and protocols:

j) OSI management protocols.

The first three items in this list relate to the Application and Presentation Layers of the Reference Model and it is expected that experience gained in the development of the corresponding standards will lead to general service and protocol standards for the Presentation Layer.

1 SCOPE AND FIELD OF APPLICATION

This International Standard describes the Reference Model of Open Systems Interconnection. It establishes a framework for the co-ordination 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 Open Systems Interconnection. It is not an implementation specification for systems, nor a basis for appraising the conformance of actual implementations.

2 REFERENCES

NOTE - It is anticipated that this International Standard will contain references.

3 DEFINITIONS AND NOTATIONS

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

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, e.g. (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, e.g. transport-protocol, session-entity,. 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 are made in clauses 6 and 7.

4.1 Definitions

4.1.1 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. In this International Standard, except in this clause, the term system is used to refer to those aspects of a system pertinent to OSI.

4.1.2 Open system: A system which obeys OSI standard in its communication with other systems. In this International Standard, except in this clause, the term open system is used to refer to those aspects of an open system pertinent to OSI.

4.1.3 Application - process: An element within a system which performs the information processing for a particular application. In this International Standard, except in this clause, the term application-process is used to refer to those aspects of a system pertinent to OSI.

4.2 The Open Systems Interconnection environment

In the concept of OSI, a system is a set of one or more computers, associated software, peripherals, terminals, human operators, physical processes, information transfer means, etc., that forms an autonomous whole capable of performing information processing and/or information transfer. An application-process is an element within a system which performs the information processing for a particular application.

Application-processes can be 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 an automated banking terminal is a manual application-process;

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

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 systems (and not the internal functioning of each individual system).

As shown in figure 1, the physical media for systems interconnection provides the means for the transfer of information between systems.

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

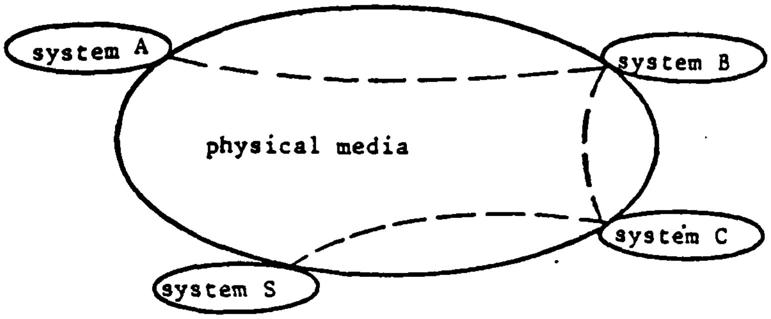


Figure 1 - Systems connected by physical media

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

The objective of OSI is to define a set of standards allowing systems to cooperate. A system which obeys applicable OSI standards in its communication with other systems is termed an open system.

[1] The scope of cooperation among open systems entails a broad range of major subjects of which the following have been identified:

a) interprocess communication, which concerns the exchange of information and the synchronization of activity between 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 systems;
c) data storage, which concerns storage media, and file and data base systems for managing and providing access to data stored on the media;
d) process and resource management, which concerns the means by which application-processes are declared, initiated and controlled, and the means by which they acquire resources;
e) integrity and security, which concern information processing constraints that must be preserved or assured during the operation of the systems;
f) program support, which concerns the definition, compilation, linking, testing, storage, transfer, and access to the programs executed by application-processes.

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

4.3 Modelling the OSI environment

The development of OSI standards, i.e. standards for the interconnection of open systems is assisted by the use of abstract models. To specify the external behaviour of interconnected open systems, each "real" open system is replaced by a functionally equivalent "abstract" open system. Only the interconnection aspects of these abstract systems would strictly need to be described. However to accomplish this, it is necessary to describe both the internal and external behaviour of these abstract systems. Only the external behaviour of abstract open systems is retained as the standard of behaviour of real open systems. The description of the internal behaviour of abstract 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 abstract open system can be considered an open system, i.e. in conformance with the OSI standards.

This abstract modelling is used in two steps.

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

Then, the detailed and precise description of the functioning of the abstract open system is developed in the framework formed by the Reference Model. This constitutes the services and protocols for Open Systems Interconnection.

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

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

Throughout the remainder of this International Standard, only the aspects of 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. In this International Standard, except where noted, the terms "system" and "open system" represent abstract open systems, not real open systems, i.e. only those aspects of a system which are of concern to OSI.

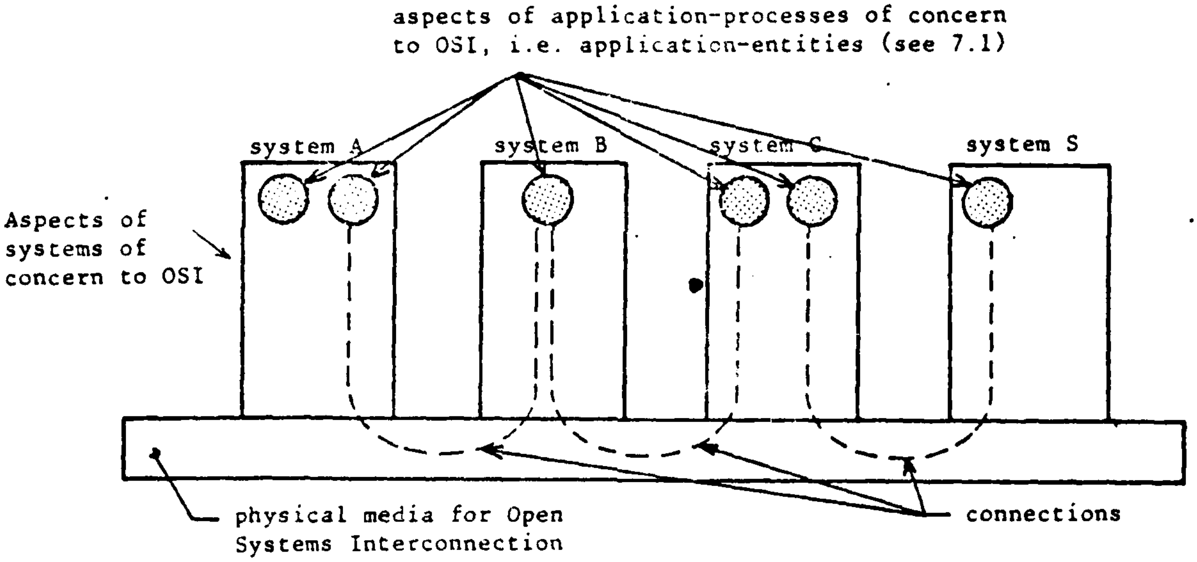

aspects of application-processes of concern to OSI, i.e. application-entities (see 7.1)
system A
system B
system C
system S
Aspects of systems of concern to OSI
physical media for Open Systems Interconnection
connections

Figure 2 - Basic elements of Open Systems Interconnection

5 CONCEPTS OF A LAYERED ARCHITECTURE

5.1 Introduction

As illustrated in figure 2, the Reference Model has been approached with the thought that there are four elements which are basic to the subject. These are:

a) the application-processes which exist within the Open Systems Interconnection environment ;

b) the connections (see 5.2) which join the application-processes and permit them to exchange information ;

c) systems;

d) the physical media for Open Systems Interconnection.

NOTE - This Basic Reference Model for Open Systems .Interconnection is based on the assumption that a connection is required for the transfer of data. An addition to this basic model is currently being developed to extend the description to cover the "connectionless" forms of data transmission. "Connectionless" data transmission may be found in a wide of data communications techniques, such as local area networks, digital radio, etc. and applications, such as remote sensing and banking.

These elements are illustrated in figure 2.

Clause 5 sets forth the architectural principles that are to be 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 discussed. Fourthly elements of layer operation are discussed including connections, transmission of data and error functions. Then, routing aspects are introduced and finally, management aspects are discussed.

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

The functions and concepts described in this clause are those known to be useful to achieve Open Systems Interconnection. However, not all functions and concepts described will necessarily be employed by each layer of the Reference Model.

5.2 Principles of layering

5.2.1 Definitions

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

5.2.1.2 (N)-layer: A well defined subdivision of the OSI architecture, constitutes 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 grouping of functions in a layer.

NOTE - The detailed definition of a sublayer is for further study.

5.2.1.6 (N)-service: A capability of the (N)-layer and the layer 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 activity of (N)-entities.

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

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 Open Systems Interconnection Reference Model is layering. According to this technique, each system is viewed as being logically composed of an ordered set of subsystems, represented for convenience in a vertical sequence as in figure 3. Adjacent subsystems communicate through their common boundary. Subsystems of the same rank (N) collectively form the (N)-layer of the OSI Reference Model. An (N)-subsystem is made of one or several (N)-entities. Entities exist at 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.

NOTE - 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 Open Systems Interconnection.
A sublayer is a grouping of functions in a layer which may be bypassed; a sublayer uses the entities and connections of its layer. The detailed definition or additional characteristics of a sublayer are for further study.

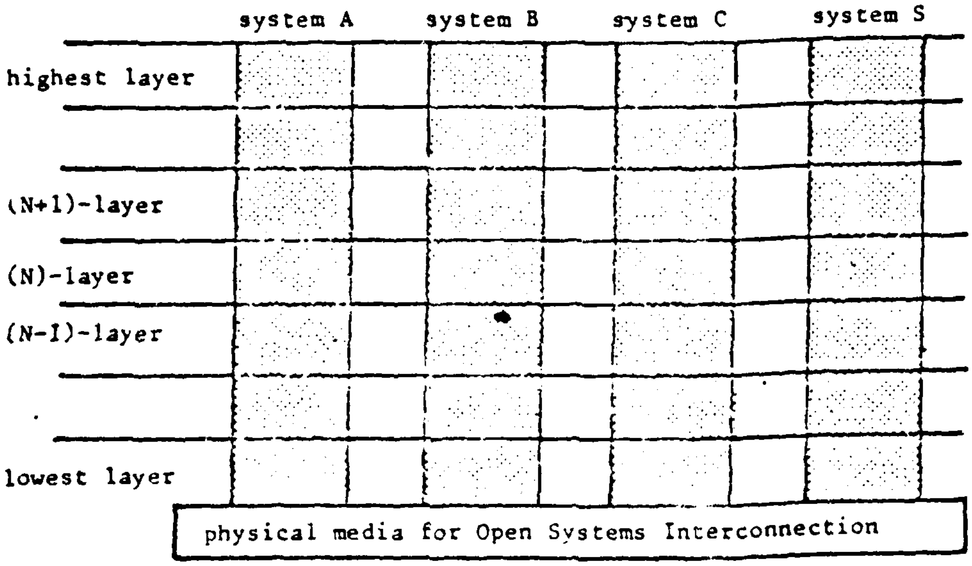


Figure 3 - Layering in co-operating open systems

Except for the highest layer, each (N)-layer provides 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 next lower layer.

NOTE 1 - A system which is not a source or sink of data need not contain the higher layers of the architecture.

NOTE 2 - Within the (N)-services, classes of service may be defined. 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 the service. When a single entity of a layer cannot by itself fully support a service requested by one of its next higher layer entities, it calls upon the co-operation of other entities in the same layer to help complete the service request. In order to co-operate, entities in any layer, other than the lowest layer, communicate by means of the set of services provided by the next lower 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 next higher layer, using the (N)-functions performed within the (N)-layer and the services available from the next lower layer.

An entity may provide services to one or more entities in the next higher layer and use the services of one or more entities in the next lower layer. A service-access-point is the access means by which a pair of entities in adjacent layers use or provide services (see figure 7).

The operation of a layer, i.e., the co-operation between entities in a layer, is governed by a set of protocols specific to the layer. The entities and protocols within a layer are illustrated in figure 5.

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 at the ends of an (N-1)-connection.

5.3.1.5 (N)-relay: An (N)-function through which an (N)-entity forwards data received from a correspondent (N)-entity to another correspondent (N)-entity.

5.3.1.6 (N)-data communication: The part of the (N)-function corresponding to transferring of (N)-protocol data-units according to an (N)-protocol over one or more (N-1)-connection.

5.3.1.7 (N)-data-source: That (N)-entity that enters (N-1)-service-data-units from an (N-1)-connection.

5.3.1.8 (N)-data sink: That (N)-entity that receives (N-1)-service-data-units from an (N-1)-connection.

5.3.1.9 (N)-data transmission: The part of the (N)-service corresponding to conveying (N)-service-data-units from one (N+1)-entity for reception at one or more (N+1)-entities via (N)-connections.

5.3.1.10 (N)-duplex transmission: (N)-data transmission of (N)-service-data-units in both directions at the same time.

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

5.3.1.12 (N)-one-way communication: (N)-data communication such that (N)-protocol-data-units are transferred in one pre-assigned direction.

5.3.1.13 (N)-simplex transmission: (N)-data transmission of (N)-service-data-units in one pre-assigned direction only.

5.3.1.14 (N)-two-way alternate communication: (N)-data-communication such that (N)-protocol-data-units are transferred in both directions, one direction at a time.

5.3.1.15 (N)-two-way simultaneous communication: (N)-data-communication such that (N)-protocol-data-units are transferred in both directions at the same time.

5.3.2 Description

In order for information to be exchanged between two or more (N+1)-entities, an association must be established in the (N)-layer using an (N)-protocol.

NOTE - Within the (N)-protocols, classes of protocols may be defined. The precise definition of the term classes of protocols is for futher 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 end 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. Entities with a connection between them are termed correspondent entities.

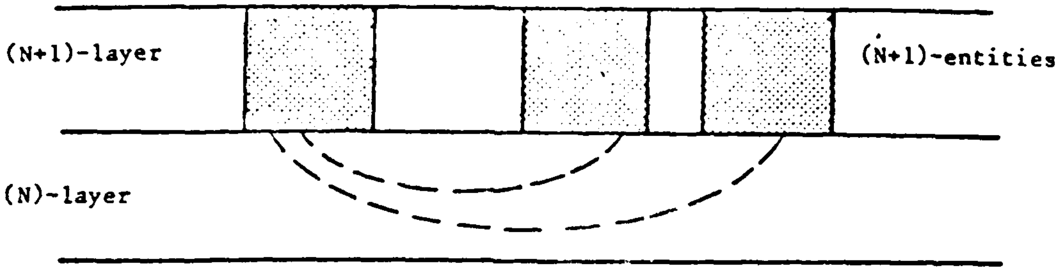


Figure 4 - Entities in a layer communicate through the next lower layer

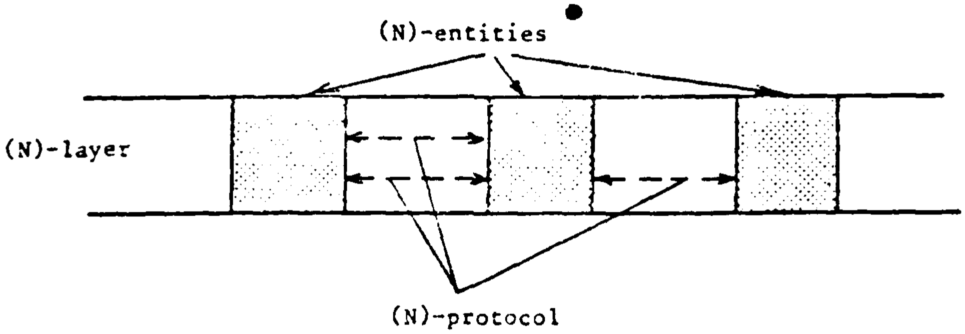


Figure 5 - Protocols between entities

Entities within a layer can communicate only by using the services of the next lower layer. There are instances where the set of services provided by the next lower layer does not permit direct access between all the entities in the layer. In such cases, communication between these entities of the layer can still be achieved if some other entity in the same layer can act as a relay between them (see figure 6).

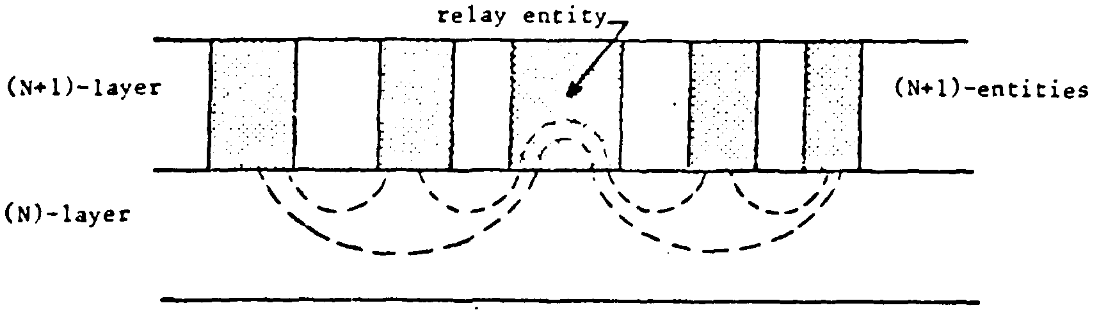


Figure 6 - Communication through a relay

The fact that communication is relayed by a chain of entities in a layer is known neither by the lower layer nor by the higher layer.

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 witin 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 it 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 to translate the title or address of an entity into a path by which the entity is to 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.

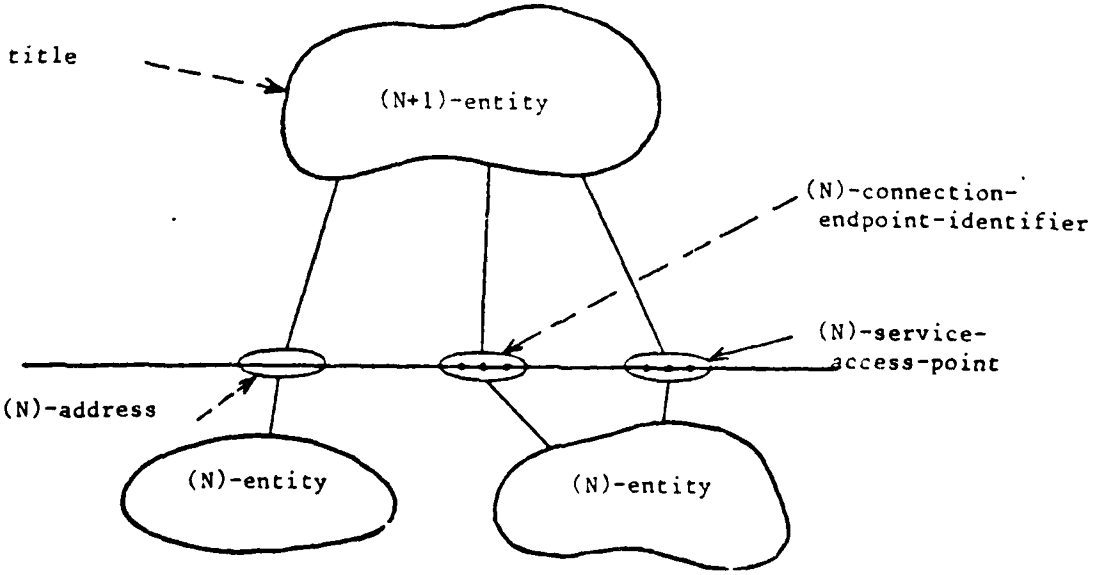
5.4.1.12 Multi-connection-endpoint-identifier: A identifier required to specify which connection-endpoint of a multi-connection-endpoint should accept the data that is being transferred.

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

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 entity is attached (see figure 7). Thus if the entity is no longer attached to the service-access-point, the address will no longer provide access to the entity. If the service-access-point is reattached to a different entity, then the service-access-point-address identifies the new entity and not the old one.

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



NOTE - dashed arrows pertain to identifiers.

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

An (N)-directory concept is defined within the architecture. It serves as an (N)-function within an (N)-layer to translate global-titles of co-operating (N)-entities (peer entities) into the (N-1)-addresses through which they interwork.

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

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

a) hierarchical (N)-address-mapping ;
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) such that the (N)-address is made of an (N-1)-address and an (N)-suffix.

In this case, the (N)-address-mapping function simply consists of recognizing the hierarchical structure of the (N)-address and extracting the (N-1)-address it contains.

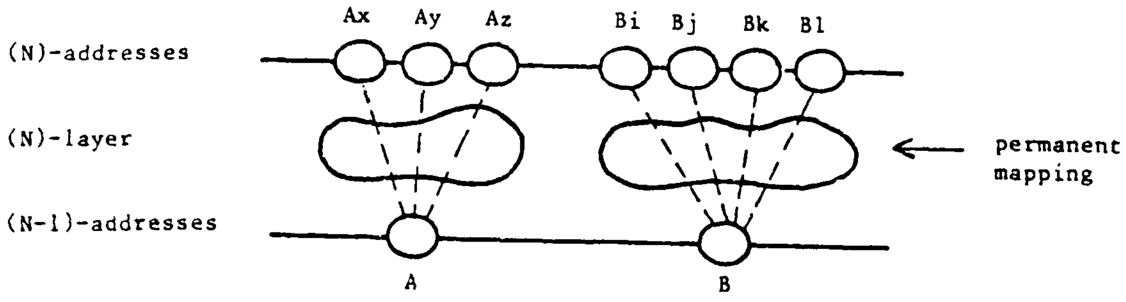


Figure 8 - Hierarchical (N)-address-mapping

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.

This hierarchical structure of addresses within a given layer simplifies (N)-address-mapping functions within that layer 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 such (N)-address-mappings and to cover the case where one (N)-entity attached to more than one (N-1)-service-access-point supports only one (N)-service-access-point identified by an (N)-address.

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 must use tables to translate (N)-addresses into (N-1)-addresses.

The structure of an (N)-address is known by the (N)-entity which supports the (N)-service-access-point so identified. 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)-layer.

A routing concept is defined within the architecture. It serves as a function to translate the address of an entity into a path or route by which the entity may be reached.

An (N)-entity may establish an (N-1)-connection with another (N)-entity by using an (N-1)-service. When an (N)-entity gets an (N-1)-connection with another (N)-entity, each (N)-entity is given an <u>(N-1)-connection-endpoint-identifier</u> by its supporting (N-1)-entity so that the (N)-entity can distinguish the new connection from all other (N-1)-connections accessible at the (N-1)-service-access-point it is using. This (N-1)-connection-endpoint-identifier is required to be unique in the scope of the (N)-entity which will use the (N-1)-connection.

The (N-1)-connection-endpoint-identifier consists of two parts:

a) an (N-1)-address of the (N-1)-service-access-point which will be used in conjunction with the (N-1)-connection ;

b) an <u>(N-1)-connection-endpoint-suffix</u> which is unique within the scope of the (N-1)-service-access-point.

A multi-endpoint-connection requires <u>multi-connection-endpoint-identifiers</u>. Each such identifier is used to specify which connection-endpoint should accept the data which is being transferred. The multi-connection-endpoint-identifier must be unique in 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 in the (N)-layer.

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

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

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

A service-access-point is located by means of its address. An (N)-address is also 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 specific in an (N)-protocol which consists of (N)-protocol-control-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 which has been 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 service-access-point between an (N+1)-entity and an (N)-entity in a single interaction and which contains (N)-interface-control-information and/or possibly the whole or part of an (N)-service-data-unit. The size of (N)-interface-data-units is not necessarily the same at each end of the connection.

5.6.1.7 (N-1)-service-data-unit: The amount of (N-1)-interface-data whose identity is preserved from one end of an (N-1)-connection to the other. Data may be held within a connection until a complete service-data-unit is put into the connection.

5.6.1.8 Expedited (N-1)-service-data-unit: A small (N-1)-service-data-unit whose transfer is expedited. The (N-1)-layer ensures that an expedited-data-unit will not be delivered after any subsequent service-data-unit or expedited unit sent on that connection. An expedited (N-1)-service-data-unit is intended to be processed by the receiving (N)-entity with priority over normal (N-1)-service-data-units. An expedited (N-1)-service-data-unit may also be referred to as an (N-1)-expedited-data-unit.

5.6.2 Description

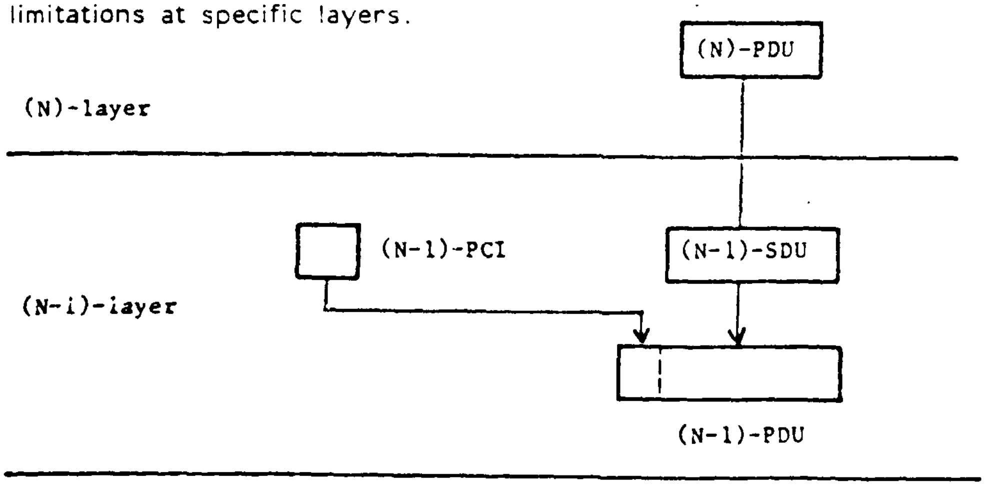
Information is transferred, in various types of data units, between peer entities and between entities attached to a specific service access point. The data units are defined above and the interrelationship among several of them is illustrated in figure 9.

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

Figure 9 - Interrelationship between data units

NOTE - An (N)-protocol-data-unit may be mapped one-to-one into an (N-1)-service-data-unit (see figure 10).

Except for the relative relationship defined above, there is no overall architectural limit to the size of data units. There may be other size limitations at specific layers.



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

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

NOTE 2- This figure illustrates one of the ways that data-units and protocol-control-information can be mapped.

NOTE 3- This figure does not imply any positional relationship between protocol-control-information and user-data in protocol-data-units

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 entit, 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 where data sent by any 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 of an (N)-entity to identify multiple (N)-connections within (N-1)-service-data-units received on one single (N-1)-connection supporting the multiple (N-1)-connections. 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 suppo.t 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 of an (N)-entity to identify one (N)-connection within (N-1)-service-data-units received on several (N-1)-connections supporting the (N)-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 for the control of the data flow within a layer or between adjacent layers.

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

5.7.1.10 Reassembling: A function of an (N)-entity to get one (N)-service-data-unit out of multiple (N)-protocol-data-units. It is the reverse function of segmenting.

5.7.1.11 Blocking: A function of an (N)-entity to map multiple (N)-service-data-units on one (N)-protocol-data-unit.

5.7.1.12 Deblocking: A function of 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 of an (N)-entity to map multiple (N)-protocol-data-units on one (N-1)-service-data-unit.

5.7.1.14 Sequencing: A function of the (N)-layer to provide the (N)-service of delivering data in the same order it was submitted.

5.7.1.15 Acknowledgement: A function used between peer (N)-entities to obtain a higher probability of detection of (N)-protocol-data-unit loss than provided by the (N-1)-layer.

5.7.1.16 Reset: A function which permits the correspondent (N)-entities to come back to a predefined state with a possible loss or duplication of data.

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, which is offered as a service by the (N)-layer, so that information may be exchanged between the (N+1)-entities.

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

There will be one (N)-connection-endpoint 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 will have two or more (N)-connection-endpoints.

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

(N)-connection-endpoints relate three elements:

a) an (N+1)-entity ;
b) an (N)-entity ;
c) an (N)-connection.

The (N)-entity and the (N+1)-entity associated by an (N)-connection-endpoint are those implied by the (N)-address referenced when the (N)-connection was 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 those with three or more connection-endpoints. Two types of multi-endpoint-connections are defined [1] :

a) centralized ;

b) decentralized.

With centralized multi-endpoint-connections, the data sent by the entity associated with the central connection-endpoint is received over the connection by the entities associated with all other connection-endpoints. The data sent by one of the entities associated with a non-central connection-endpoint is received over the connection only by the entity associated with the central connection-endpoint.

With decentralized multi-endpoint-connections, the data sent by the entity associated with any connection-endpoint is received over the connection 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 ;

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

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

If required and 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 Open Systems Interconnection 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, this release 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 (e.g. time of failure of the (N-1)-connection and common time-out). In addition, both (N)-entities must be in a state in which they can execute the connection release protocol exchange. It is important, however, to note 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.

The characteristics of the (N)-service with regard to the release of an (N)-connection can be of two kinds: (N)-connections are either released immediately when release protocol exchange is initiated ((N)-user-data not yet delivered may be discarded) or release is delayed until all (N)-user-data sent previous to the initiation of the protocol exchange has been delivered (i.e. delivery confirmation has been received).

Some (N)-protocols may be such that connection establishment and connection release protocol exchanges may be combined.

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

5.7.5 Multiplexing and splitting

Within the (N)-layer, (N)-connections are mapped onto (N-1)-connections. The mapping may be:

a) one-to-one ;

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

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

The (N)-layer may need to perform multiplexing, i.e., use one (N-1)-connection to support more than one (N)-connection.

It may also need to perform splitting, i.e., use more than one (N-1)-connection to support one (N)-connection.

Multiplexing may be needed in order to:

a) make more efficient or more economic use of the (N-1)-service ;
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 would be available ;

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

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

These connection mapping functions involve a number of associated functions which might not be needed when connection mapping is done on a one-to-one basis.

The functions associated with multiplexing are :

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

b) when the capacity of the (N-1)-connection is shared by the introduction of a multiplexing function, it is necessary that flow control functions be performed on each individual flow (see 5.7.6.4).

c) when more than one (N)-connection is prepared to send data, it is necessary that scheduling functions be performed to select the next (N)-connection to be serviced over the (N)-connection.

The functions associated with splitting are :

a) scheduling of the utilization of multiple (N-1)-connections for splitting of a single (N)-connection ;

b) resequencing of (N)-protocol-data-units associated with an (N)-connection that is split onto several (N-1)-connections since they may arrive out of sequence despite the fact that each (N-1)-connection may guarantee 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. The (N)-service-data-unit is transferred as (N)-user-data in one or more (N)-protocol-data-units.

An (N)-protocol-data-unit has an arbitrary, but finite, size. (N)-protocol-data-units are mapped into (N-1)-service-data-units for transportation purposes. 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 exchange of data under rules of an (N)-protocol may only occur if an (N-1)-connection exists. If an (N-1)-connection does not exist, it must be established before data transfer can be provided (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 protocol establishment exchange.

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

By combining connection release protocol commands with connection establishment protocol commands (see 5.7.4), these mechanisms provide means for the delivery of a single unit of (N)-user-data between peer (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 mechanisms.

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.

5.7.6.4 Flow control

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

If the maximum size of service-data-units is not known, then segmentation must be first performed on service-data-units to make them fit within data units of a known size. Flow control can then be applied on the fragmented data units of known size.

Two types of flow control are identified:

a) peer-to-peer flow control which requires protocol definitions and is based on protocol-data-unit size ;

b) (N)-interface flow control which permits an (N+1)-entity and an (N)-entity servicing it to regulate the rate at which (N)-interface-data is sent onto the (N)-connection or received from the (N)-connection at the (N)-service-access-point. (N)-interface flow control is based on the (N)-interface-data-unit size.

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

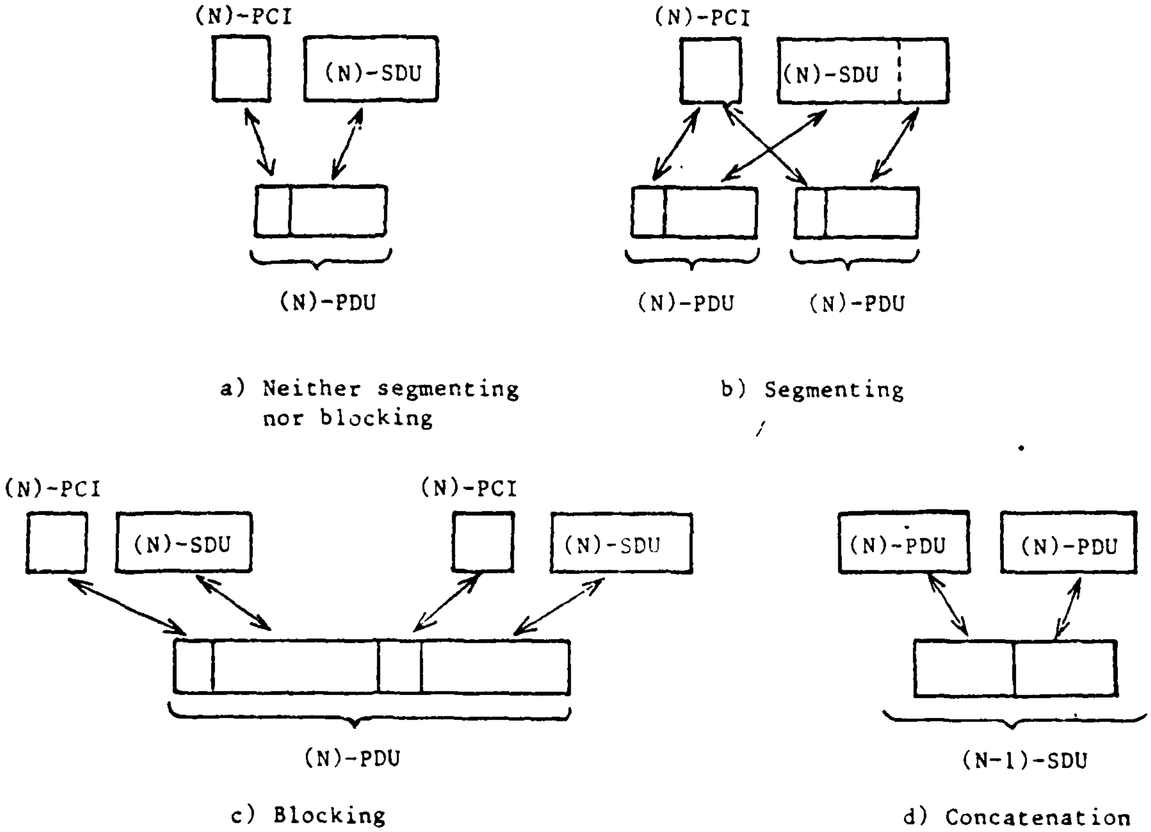
Peer-to-peer flow control mechanisms will require that flow control information be included with the (N)-protocol-control-information of an (N)-protocol-data-unit.

5.7.6.5 Segmenting, blocking and concatenation

Data units in the various layers will not necessarily be 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-unit. Simiiarly, segmenting may occur when (N)-protocol-data-units are mapped into (N-1)-interface-data-units. It is necessary to preserve the identity of (N)-service-data-units on an (N)-connection. Thus, mechanisms must be available for identifying the segments of an (N)-service-data-unit, such that the correspondent (N)-entity can determine when the (N)-service-data-unit is complete.

Segmenting mechanisms 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 segmented into several (N)-protocol-data-units with added (N)-protocol-control-information (see figure 11b).

Conversely, it may be necessary to peer form blocking whereby several (N)-service-data-units with added (N)-protocol-control-information form an (N)-protocol-data-unit (see figure 11c). The model also permits concatenation whereby several (N)-protocol-data-units are concatenated into a single (N-1)-service-data-unit (see figure 11d).



a) Neither segmenting nor blocking

b) Segmenting

c) Blocking

d) Concatenation

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

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

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

5.7.6.6 Sequencing

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

5.7.7 Error functions

5.7.7.1 Acknowledgement

An acknowledgement function may be used between peer (N)-entities using an (N)-protocol to obtain a higher probability of detection of protocol-data-unit loss than provided by the (N-1)-layer. Each (N)-protocol-data-unit transferred between corresponding (N)-entities must be uniquely identifiable, such that the receiver can inform the sender of the receipt of the (N)-protocol-data-unit. The acknowledgement functions must also be 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 duplicate detection, 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

Certain error control functions may be used by an (N)-protocol to provide a higher probability of protocol-data-unit error detection and data corruption detection than is provided by the (N-1)-layer.

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

5.7.7.3 Reset

Some services will require a reset function to recover from a loss of synchronization between correspondent (N)-entities. The reset function permits the correspondent (N)-entities to come back 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+1)-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

Routing is a function within the (N)-layer which enables communication to be relayed by a chain of (N)-entities. The fact that communication is being routed by intermediate (N)-entities is know either by the lower layers 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: A function in the Application Layer 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: A function 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: A function 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 Open Systems Interconnection architecture there is a need to recognize the special problems of initiating, terminating, monitoring on-going activities and assisting in their harmonious operations, as well as handling abnormal conditions. These have been collectively considered as the management aspects of the Open Systems Interconnection 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 sections of this document.

The objectives are to identify management activities which imply actual exchanges of information between systems. Only the protocols needed to conduct such exchanges are candidates for standardization within the Open Systems Interconnection architecture framework.

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

5.9.3 Categories of management activities

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

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

The following categories of management activities are identified:

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 in this category but it is not claimed to be 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 :
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 various OSI resources and their status across all layers of the Open Systems Interconnection architecture. The following list is typical of functions which fall into this category but it is not claimed to be exhaustive:

a) activation/deactivation management which includes :

   1) activation, maintenance and termination of OSI resources distributed in open systems, including physical media for Open Systems Interconnection,
   2) some program loading functions,
   3) establishment/maintenance/release of connections between management entities,
   4) open systems parameter initialization/modification ;

b) monitoring which includes :

   1) reporting status or status changes,
   2) reporting statistics ;

c) error control which includes :

   1) error detection and some of the diagnostic functions,
   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 OSI 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 system-management-application-entities.

5.9.4 Overall principles of positioning of management functions in the model

Several principles have been considered important in positioning management functions in the Open Systems Interconnection architecture model. They include the following [1]:

a) Both centralization and decentralization of management functions are allowed. In other words, the Open Systems Interconnection architecture does not dictate any particular fashion or degree of centralization of such functions. This principle calls for a structure in which each system is allowed to include any (subset of) system management functions and eacn layer segment of each system is allowed to include any (subset of) layer management functions.

b) If it is necessary, establishment of connections between management entities is performed when a system which has been operating in isolation from other systems, becomes part of the Open Systems Interconnection architecture.

[1] Other principles are for further study.

6 INTRODUCTION TO THE SPECIFIC OSI LAYERS

6.1 Specific layers

The general structure of the Open Systems Interconnection 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);

g) the Physical Layer (layer 1).

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

Layers 1-6, together with the physical media for Open Systems Interconnection 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.

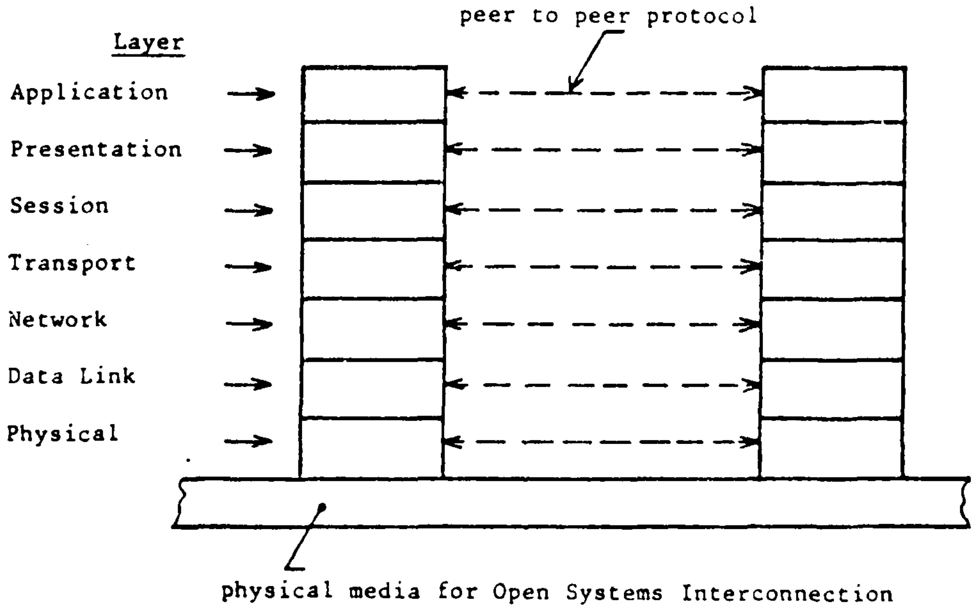


Figure 12 - Seven layer reference model and peer to peer protocol

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

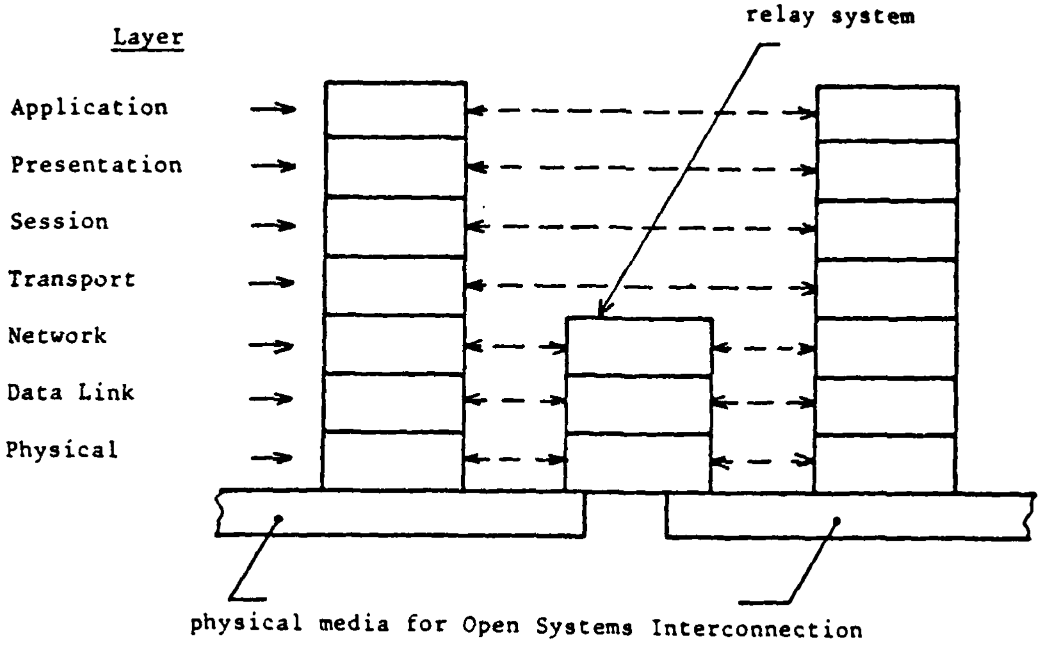


Figure 13 - Communication involving relay systems

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 fu*r*ther decisions in the development of OSI standards,

NOTE 1 - In its present definition, the Reference Model for Open Systems Interconnection architecture does not cover the case where systems are interconnected by means other than telecommunications. This case is be for further study.

NOTE 2 - 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.

P1 : do not create so many layers as to make the s/stem engineering task of describing and integrating the layers mor*e* difficult than necessary,

P2 : create a boundary at a point where the description of services can be small and the number of interactions across the boundary are minimized,

P3 : create separate layers to handle functions that are manifestly different in the process performed or the technology involved,

P4 : collect similar functions into the same layer,

P5 : select boundaries at a point which past experience has demonstrated to be successful,

P6 : 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 and interfaces with the adjacent layers, P7 : create a boundary where it may be useful at some point in time to have the corresponding interface standardized

NOTE 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 P7, should not be taken to imply usefulness of standards for such internal interfaces.

NOTE 2 - It is important to note that Open Systems Interconnection 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.

P8 : create a layer when there is a need for a different level of abstraction in the handling of data, e.g. morphology, syntax, semantics,

P9 : enable changes of functions or protocols within a layer without affecting the other layers,

P10 :create for each layer interfaces with its upper and lower layer only,

P11 :create further subgrouping and organization of functions to form sublayers within a layer in cases where distinct communication services need it,

P12 :create, where needed, two or more sublayers with a common, and therefore minimal functionality to allow interface operation with adjacent layers,

P13 :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) the services offered by the layer to the layer above;

c) 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, since these are the subject of separate standards.

7.1 The Application Layer

7.1.1 Definitions

7.1.1.1 Application-entity: Tne aspect of an application-process of concern to OSI.

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 purpose of the Application Layer is to serve as the window between communicating application-processes using the OSI environment to exchange meaningful information.

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

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

7.1.3 Services provided to the using application-process

The application-processes exchange data by means of application-protocols and presentation-services.

As the only layer in the OSI environment that directly provides services to the application-processes, the Application Layer necessarily provides all services directly usable by application-processes. In addition to information transfer, such services may include, but are not limited to the following:

NOTE - Some of the services listed after are provided by application management.

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

b) determination of the current availability of the intended communicants ;

c) establishment of authority to communicate ;

d) agreement on privacy mechanisms required ;

e) authentication of intended communicants ;

f) determination of cost allocation methodology ;

g) determination of adequacy of required resources ;

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

i) synchronization of cooperating applications ;

j) selection of dialog discipline including initiation and release procedures ;

k) agreement on responsibility for error recovery ;

l) agreement on procedure for control of data integrity ;

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

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.

7.1.4.1 Grouping of functions in the Application Layer

The 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 user application-process may determine the grouping of functions comprising the application-entity visible in the OSI environment.

7.1.4.2 Systems 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 The Presentation Layer

7.2.1 Definitions

7.2.1.1 Presentation-image: The OSI view of a data structure which application-entities wish to refer to in their dialogue, along with the set of actions performed on this data structure.

7.2.1.2 Presentation-image-definition: The syntactic view of a presentation-image, known by presentation-entities, thus excluding the semantic view, i.e. the meaning of the presentation-image to application-entities, which is not known by presentation-entities.

7.2.1.3 Data syntax: The syntactic view fo data transferred between application-entities, known by presentation-entities, thus excluding the semantic view, i.e. the meaning of the data to application-entities, which is not known by presentation-entities.

7.2.2 Purpose

The purpose of the Presentation Layer is to represent information to communicating application-entities in a way that preserves meaning while resolving syntax differences.

The Presentation Layer covers two complementary aspects of this representation of information to application-entities:

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

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

A presentation-image is the OSI view of such a data structure along with the set of actions possible on it.

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

In the remainder of this clause, the term "syntax" is used only to refer to the syntax of the data being transferred, while the specific term presentation-image-definition is introduced to refer to the syntactic view of a presentation image (including the definition of actions possible on it).

NOTE - The terms "internal attribute" and "external attributes" have sometimes been used to refer respectively to the syntax and to the semantics of a presentation image. The concept of "internal attribute" is similar to the concept of presentation-image-definition defined in this International Standard.

7.2.3 Services provided to the Application Layer

The Presentation Layer adds on session-services (see 7.3) the following facilities:

a) data syntax transformation ;
b) data formatting ;
c) transformation of presentation-image-definition ;
d) syntax selection ;
e) selection of presentation-image-definition.

Data syntax transformations deal mainly with code and character set conversions. Data formatting deals with the modification of the layout of the data. Syntax selection provides the means of initially selecting and subsequently modifying the selection of the transformation and formats to be used.

Selection of the presentation-image-definition provides the means for initially selecting and subsequently modifying the presentation-image-definition. Adaptation of actions on the presentation-image are those concerning only the syntactic view of this presentation-image.

7.2.4 Functions within the Presentation Layer

The layer performs the following functions to help accomplish the above services:

a) session establishment request;
b) data transfer;
c) negotiation and renegotiation of syntax and presentation-image-definition;
d) transformation of syntax and presentation-image-definition including data transformation and formatting and special purpose transformations (e.g. compression);
e) session termination request.

7.2.4.1 Transformation of syntax

There are three syntactic versions of the data being transferred : the syntax used by the application-entity of the originator of the data, the syntax used by the application-entity of the recipient of the data, and the syntax used to transfer the data between presentation-entities ("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 local system 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. Syntax transformation may be performed either as a specific syntax-matching service provided to the application-processes, or as a function internal to the Presentation Layer to accomplish other service objectives (e.g. data volume reduction to reduce data transfer cost).

7.2.4.2 Transformations of presentation-image-definition

There are three versions of the presentation-image-definition known by the Presentation Layer: the presentation-image-definition used by the application-entity of the originator of the actions on the data structure, the presentation-image-definition used by the application-entity of the recipient of the actions on the data structure, and the presentation-image-definition used to propagate the actions between presentation-entities ("common presentation-image-definition"). It is clearly possible that any two or all three of these presentations-images-definitions may be identical. The Presentation Layer contains the functions necessary to transform between the common presentation-image-definition and each of the other two presentations-image-definitions as required.

There is not a single predetermined common presentation-image-definition for all OSI. The common presentation-image-definition to be used on a presentation-connection is negotiated between the correspondent presentation-entities. Thus, a presentation-entity must know the presentation-image-definition of its local system and the agreed common presentation-image-definition. Only the common presentation-image-definition 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 presentation-image-definition available to it.

7.2.4.3 Negotiation of syntax and presentation-image-definition

Negotiation of syntax and presentation-image-definition is carried out by a *dialogue* 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 and presentation-image-definitions used by application-processes 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.

7.2.4.4 Addressing and multiplexing

There is a one-to-one correspondence between presentation-address and session-address. There is no multiplexing nor splitting in the Presentation Layer.

7.2.4.5 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 relationship with other management aspects.

7.3 The 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 being sent on a session-connection are not made available to the receiving presentation-entity until explicitely released by the sending presentation-entity.

7.3.1.2 Interaction management: A facility of the session-service which allows presentation-entities to control explicitely 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 defined 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 cooperating presentation-entities to organize and synchronize their dialogue and manage their data exchange. To do this, the Session Layer provides services to establish a session-connection between two presentation-entities, and to support their 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.

A session-connection between presentation-entities 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, ensuring orderly message exchange across the session-connection. The session-connection exists until release by the presentation-entities. While the session-connection exists, session services maintain the state of the dialogue even over data loss by the Transport Layer.

There are only two ways a presentation-entity can access another presentation-entity: by initiating a session-connection or by accepting a session-connection. A presentation-entity may be engaged in several session-connections simultaneously. Between two presentation-entities, concurrent or consecutive session-connections are possible.

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;

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 cooperatively 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 cooperating session-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 the presentation-entities to release the session-connection in an orderly way without loss of data. Also, either presentation-entity may at any time request that the 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 includes the assurance that the receiving presentation-entity 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) being sent on a session-connection 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 either presentation entity at any time that a session-connection exists.

7.3.3.6 Interaction management

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

This 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 has not 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) interaction where both presentation-entities may concurrently send and receive ;

b) two-way-alternate (TWA) interaction where the presentation-entity with the turn may send and its correspondent is permitted only to receive ;

c) one-way interaction which is a form of operation of TWA interaction in which the turn can never be exchanged.

7.3.3.7 Session-connection synchronization

The session-connection synchronization service allows presentation-entities to:

a) define and identify synchronization points ;

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.

7.3.3.8 Exception reporting

The exception reporting service permits the presentation-entities to be notified of unanticipated 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 ;
d) security.

7.3.4 Functions within the Session Layer

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

Most of the functions required are readily implied by the services provided. However, several of the functions require some additional description which is given in the following:

a) mapping of session-connection onto transport-connection;
b) session-connection flow control;
c) expedited data exchange;
d) session-connection recovery;
e) session-connection release;
f) Session Layer management.

7.3.4.1 Mapping of session-connection onto transport-connection

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 the related session-connection can be distinguished so that the following cases are defined :

a) A transport-connection supports several consecutive session-connections (figure 14):

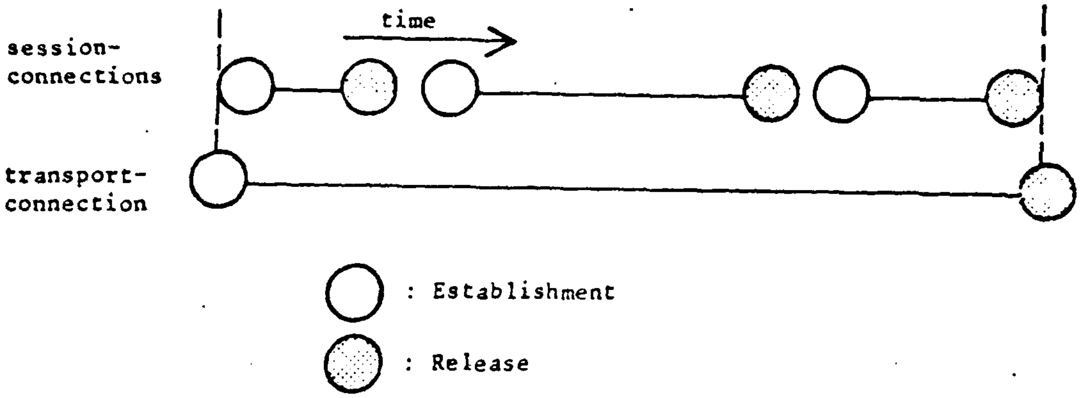


Figure 14 - Several consecutive session-connections

b) Several consecutive transport-connections support a session-connection (figure 15):

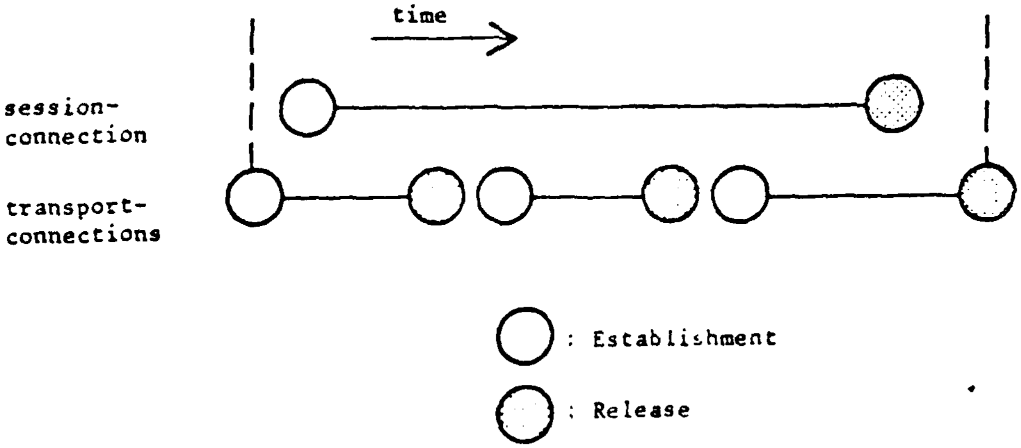


Figure 15 - Several consecutive transport-connections

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

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

7.3.4.2 Session-connection flow control

There is no peer-to-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 exchange

The exchange of expedited-session-service-data-units is generaly accomplished by use of the expedited transport service.

7.3.4.5 Session-connection recovery

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

7.3.4.6 Session-connection release

The Session Layer contains the necessary functions 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.

7.3.4.7 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.

7.4 The Transport Layer

7.4.1 Definitions

NOTE - It is expected that Transport Layer specific terms could be defined later.

7.4.2 Purpose

The Transport Layer exists to provide the transport-service in association with the underlying services provided by the supporting layers.

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

The Transport Layer is required to optimize the use of the available network-service to provide the performance required by each communicating transport user at minimum cost. This optimization will be achieved within the constraints imposed by considering the global demands of all concurrent transport users and the overall quality and capacity of the network-service available to the Transport Layer.

Since the network-service provides network-connections from any transport-entity to any other, including the case of using tandem subnetworks and, relieves the Transport Layer of any concern with switching, routing, and relaying, all protocols defined in the Transport Layer will have end-to-end significance, where the ends are defined as the correspondent transport-entities. In other words, the Transport Layer is OSI end-system oriented and transport protocols operate only between OSI end-systems.

The transport functions invoked in the Transport Layer to provide requested service quality will depend on the quality of the network-service. The quality of the network-service will depend 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 of its users by its transport-address. Users of the Transport Layer are provided with the means to establish, maintain and release transport-connections which represent a two-way simultaneous data path between a pair of transport-addresses.

More than one transport-connection can be established between the same pair of transport-addresses; the means by which the user can distinguish between the transport-connection-end-points will be provided by the Transport Layer, in terms of transport-connection-end-point-identifiers.

The operation of each transport-connection is independent of all other such connections, except for the limitations imposed by finite resources available to the Transport Layer.

The quality of service provided on a transport-connection will depend on the service class requested by the session-entities when initiating the establishment of the given transport-connection.

The selected quality of service will be maintained throughout the lifetime of the transport-connection. The transport service user will be notified of any failure to maintain the agreed quality of service on a given connection.

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

a) transport-connection establishment;
b) data transfer services;
c) transport-connection release.

7.4.3.1 Establishment services

a) Transport-connection establishment

Transport-connections are established to a peer transport address for users of the transport-service. The quality of service of the transport-connection is negotiated among the session-entities and the transport-service.

b) Class of service selection

At the time of establishment the class of transport service to be provided can be selected from a defined set of available classes of service. These classes of service represent globally predefined combinations of parameters and grades of service to be provided.

These service classes are intended to cover the transport-service requirements of the various types of traffic generated by the session-entities.

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

7.4.3.2 Data transfer services

This service provides data transfer in accordance with the agreed upon quality of service. When this quality of service cannot be maintained and all possible recovery attempts have failed, then the transport-connection [illegible] terminated and the transport uses are notified.

a) Transport-service-data-unit transfer provides the means by which transport-service-data-units of arbitrary length are delimited and transparently transfered 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) Expedited transport-service-data-unit transfer provides an additional means of information exchange on a transport-connection. These 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 the connection and have the correspondent session-entity informed of the release.

7.4.4 Functions within the Transport Layer

The Transport Layer will perform all those functions that are necessary to bridge the gap between the services provided by the Network Layer and the services needed by the Session Layer. Therefore the functions performed are dependent on two criteria: the services provided by the underlying Network Layer and the services required by the Session Layer.

In particular, 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 and blocking;

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

i) supervisory functions;

j) 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 transport-address associated with a correspondent session-entity, the Transport Layer needs to determine the network-address identifying the transport-entity which serves that correspondent session-entity, i.e. which maintains that correspondent transport-address.

Because transport-entities support services on an end-to-end basis by means of end-to-end functions, no intermediate transport-entity is involved as a relay between the end transport-entities. Therefore the network-address on which the Transport Layer maps transport-addresses are those identifying the end transport-entities (see figure 16).

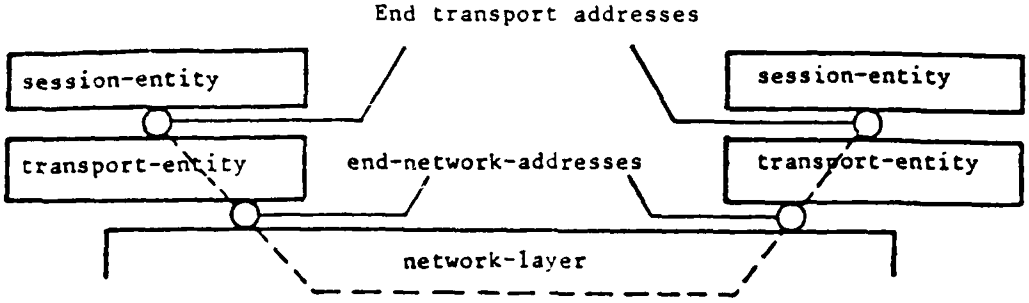


Figure 16 - Association of transport-entities

One transport-entity may serve more than one session-entity. Therefore, several transport-addresses may be associated with one network-address within the same transport-entity.

Corresponding mapping functions must then be performed within the transport-entities to provide these facilities (see figure 17).

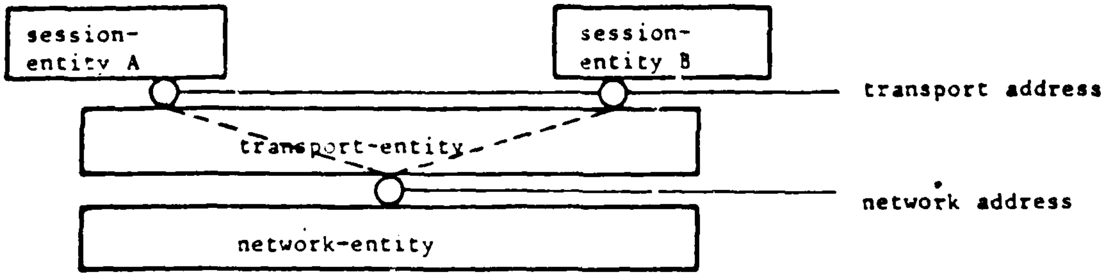


Figure 17 - One network-address associated with several transport-addresses

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 ;
c) release phase.

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

7.4.4.4 Establishment phase

The goal of the establishment phase is to establish a transport-connection between two transport users. The functions of the Transport Layer during this phase must match the requested class of services with the services provided by the Network Layer, as follows :

a) Select network-service which best matches the requirements, taking into account charges for various services;
b) decide whether to multiplex transport-connections onto a single network-connection;
c) establish optimum transport-protocol-data-unit size;
d) select the functions that will be operational upon entering the data phase,
e) map transport-addresses onto network-addresses;
f) provide a means to distinguish different-transport connections between the same pair of transport-service-access-points (connection identification function) ;
g) transportation of users 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 purpose is achieved by means of transmission of transport-protocol-data-units and by the following functions, each of these being used or not used in accordance with the result of the class of service selection performed in the connection phase.

a) sequencing ;

b) blocking ;

c) concatenation ;

d) segmenting ;

e) multiplexing or splitting ;

f) flow control between two transport-entities on one transport-connection;

g) error detection to detect the loss, corruption, duplication, misordering or misdelivery of transport-protocol-data-units;

h) error recovery to recover from detected and signalled errors;

i) expedited data transfer ;

j) transport-service-data-unit delimiting to determine the beginning and end of a transport-service-data-unit;

k) transport-connection identification to uniquely identify a transport-connection between the pair of transport entities supporting the connection.

7.4.4.6 Release phase

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

a) notification of reason for release ;

b) identification of the transport-connection released ;

c) possible additional information.

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 The Network Layer

7.5.1 Definitions

7.5.1.1 Subnetwork: A set of one or more systems which provide relaying and through which other 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 is providing a network-connection.

7.5.2 Purpose

The Network Layer provides the means to establish, maintain and terminate network-connections between 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 transmission resources are used in tandem (see 7.5.4.2) or in parallel. It makes invisible to transport-entities how the Network Layer uses underlying resources such as data-link-connections to provide network-connection.

Any relay functions and hop-by-hop service enhancement protocols used to support the network service between the OSI end-systems are operating below the OSI end-systems 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 all data submitted by the Transport Layer. 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 detailed services provided by the Network Layer on a given network-connection are described below.

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 is assumed to contain functions necessary to mask the differences in the characteristics of different transmission and network technologies into a consistent network service.

The service provided at each end of a network connection shall be the same even in the case of a network-connection spanning several subnetworks where each of the subnetworks offers 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 network service users and the network service provider at the time of establishment of a network-connection. While this quality of service may vary from one network-connection to anotner 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;
i) expedited network-service-data-unit transfer;
j) reset;
k) release services.

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

a) the user has to request the service ;
b) 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 to uniquely identify other transport-entities, i.e. network-addresses are the means by which transport-entities can communicate using the network service. The Network Layer uniquely identifies each of the end 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 which are identified by network-addresses. The Network Layer provides the means to establish, maintain and terminate 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 identifies the network-connection-endpoint uniquely with the associated network-address.

7.5.3.4 Network-service-data-unit transfer

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

No limit will be 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 errors, service availability, reliability, throughput, transit delay including variations on the transit delay, and delay for network-connection establishment.

7.5.3.6 Error notification

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

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

7.5.3.7 Sequencing

The Network Layer may provide the service of 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 over the service interface between the Transport and Network Layers. 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 (option)

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 different sets of network service characteristics and separate flow control.

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

7.5.3.10 Reset (option)

The reset service is provided as an option. Invocation of a reset service causes the Network Layer to discard all network-service-data-units associated with 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 services

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

NOTE - confirmation of receipt is a candidate for future extension.

7.5.4 Functions within the Network Layer

The functions to be provided within the Network Layer are outlined below. Network Layer functions must cover 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 sould be structured into sublayers.

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

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;
i) expedited;
j) reset;
k) service selection;
l) Network Layer management.

7.5.4.1 Routing and relaying

Network-connections are between network-entities in end systems but may involve intermediate systems which provide relaying.
These intermediate systems may interconnect subnetwork-connections, data-link-connections, and data-circuits (see 7.7). Routing functions determine an appropriate route between network adresses. 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).

NOTE - When Network Layer functions are performed by combinations of several individual communication services (or subnetworks), the specification of routing and switching functions could be facilitated by using sublayers, isolating individual subnetworks routing and switching functions from internetwork routing and switching functions.

7.5.4.2 Network-connections

This function includes mechanisms for providing 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 a tandem subnetwork-connection.

A tandem subnetwork-connection is a network-connection that uses several individual communication services (or subnetworks) in series. The interconnected individual subnetworks may have the same or different service capability. Each end of a subnetwork-connection, or connections between intermediate nodes, may operate with a different subnetwork protocol.

The concatenation 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 connected as they stand. The resultant quality is not higher than that of the lower quality subnet.

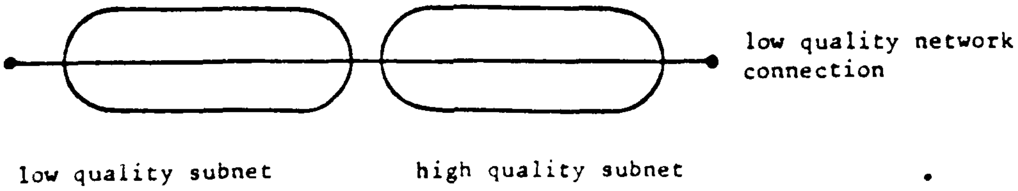


Figure 18 - Connection of low quality subnetwork and high quality subnetwork

AD-A107 644 NATIONAL COMMUNICATIONS SYSTEM WASHINGTON DC F/G 9/2
OPEN SYSTEMS INTERCONNECTION (OSI) REFERENCE MODEL (AUGUST 1981--ETC(U)
SEP 81
UNCLASSIFIED NCS-TIB-81-1A NL

2 OF 2
AD A
107644

END
DATE
FILMED
1 -82
DTIC

b) The lower quality network is enhanced to equal to the higher quality subnet and then the enhanced subnet concatenated with the high quality subnet. The resultant quality is approximately that of the higher quality subnetwork.

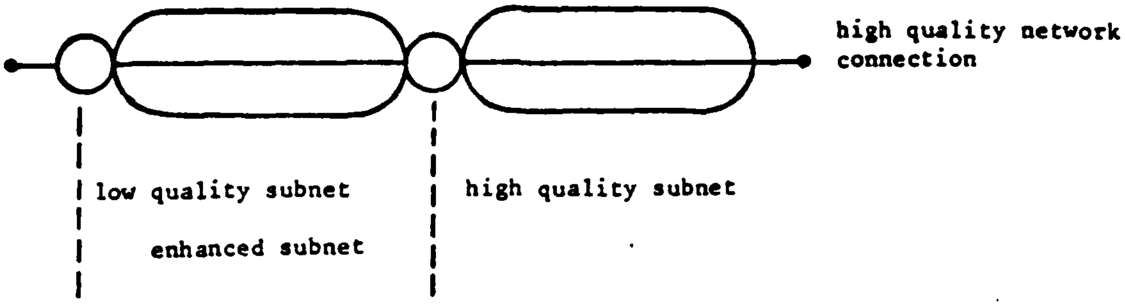


Figure 19 - Connection of enhanced low quality subnetwork and high quality subnetwork.

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

7.5.4.3 Network-connection multiplexing

In order to optimize the use of data-link-connections, the Network Layer may multiplex network-connections onto data-link-connections.

In the case of tandem subnetwork-connections, multiplexing network-connections onto individual subnetwork-connections may also be performed.

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 might be necessary to provide the required quality of service.

7.5.4.6 Error recovery

This function includes mechanisms for recovering from detected errors. Depending on the quality of the network service provided, these functions may vary.

7.5.4.7 Sequencing

This function includes mechanisms for providing the service of 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), flow control functions may need to be performed in the Network Layer..

7.5.4.9 Expedited

This function includes mechanisms for providing the expedited service.

7.5.4.10 Reset

This function includes mechanisms for providing the reset service.

7.5.4.11 Service selection

In the case where a network connection spans several subnetworks of dissimilar quality, service selection functions should be performed to ensure that the service provided at each end of the network-connection is the same.

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 The Data Link Layer

7.6.1 Definitions

NOTE - It is expected that Data Link Layer specific terms could be defined later.

7.6.2 Purpose

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

The objective of this layer is to detect and possibly correct errors which may occur in the physical layer.

In addition, the Data Link Layer enables the Network Layer to control 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;
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 another network-entity; for example, when data-link-connections are built upon multipoint-connections.

7.6.3.4 Sequencing

When required, the sequence integrity of data-link-service-data-units will be 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 will accept 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 from 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;
i) identification and parameter exchange;
j) control of data-circuit chaining;
k) Data Link Layer management.

7.6.4.1 Data-link-connection establishment and release

This function establishes and releases data-link-connections on existing activated physical-connections. When a physical-connection has multiple endpoints (e.g. 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.

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 allow the 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 - This function is sometimes referred to as framing.

7.6.4.5 Sequence control

This function maintains the sequential order of data-link-service-data-units across the 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 detected transmission, format and operational errors and notifies the network-entities of those which are unrecoverable.

7.6.4.8 Flow control

This function permits provision of 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 chaining

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 The Physical Layer

7.7.1 Definitions

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

7.7.2 Purpose

The Physical Layer provides mechanical, electrical, functional and procedural characteristics to activate, maintain and deactivate physical connections for bit transmission between data-link-entities possibly through intermediate systems, each relaying bit transmission within the Physical Layer.

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;
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 systems, together with the facilities necessary in the Physical Layer for the transmission of bits on to it.

A physical-connection may be provided by the interconnection of data-circuits by the use of relaying functions in the Physical Layer. The provision of a physical-connection by such an assembly of data-circuits is illustrated in figure 20.

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

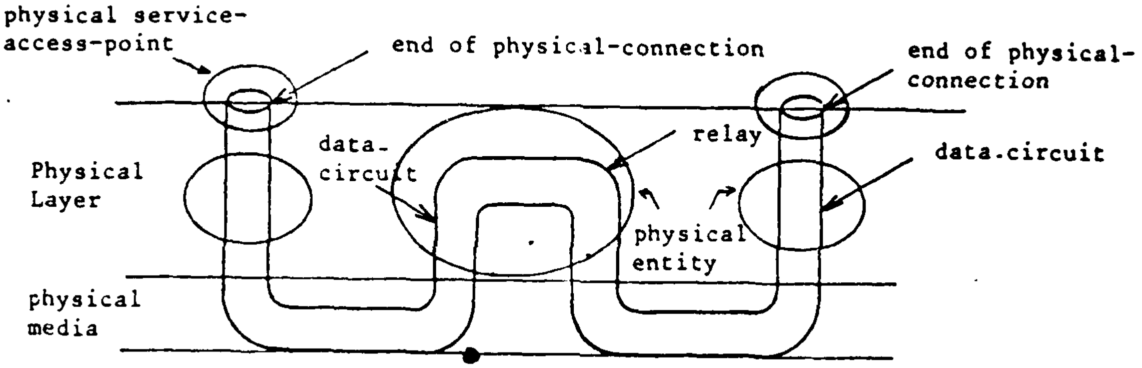

physical service-access-point
end of physical-connection
end of physical-connection
relay
data-circuit
data.circuit
Physical Layer
physical entity
physical media

Figure 20- Concatenation of data-circuits within the Physical Layer

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

(See figure 21)

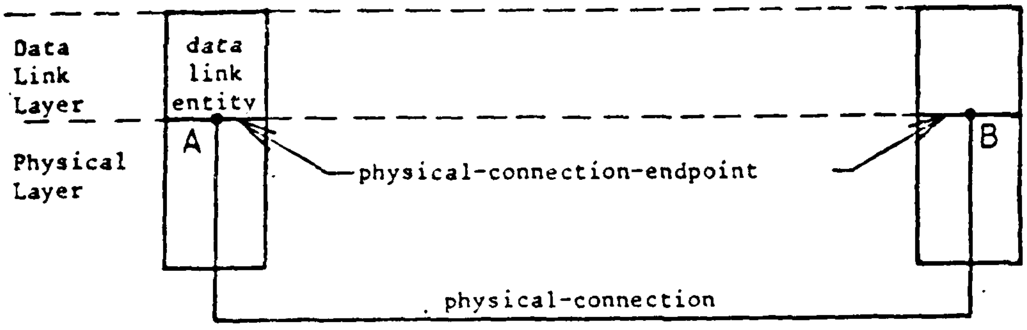
The Physical Layer will provide physical-connection-endpoint-identifiers which may be used by a data-link-entity to identify one or several physical-connection-endpoints.

A physical-connection will have two (point-to-point) or several (multiple endpoint) physical-connection-endpoints.

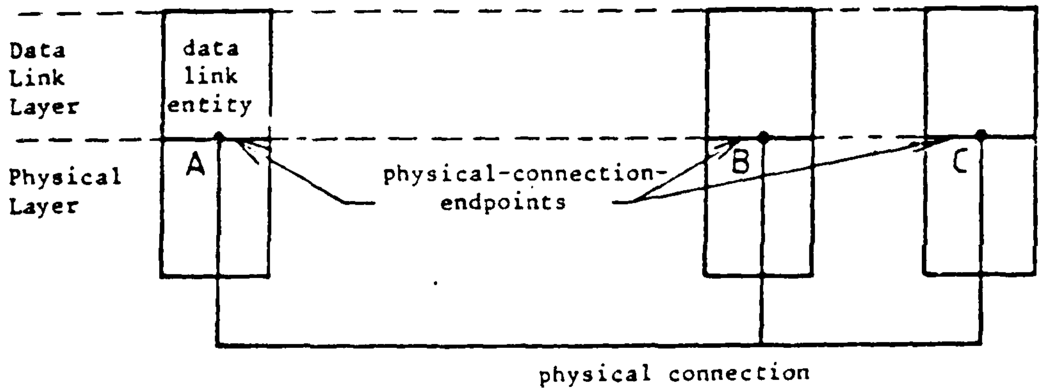
7.7.3.4 Data-circuit identification

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

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



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



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

Figure 22 - Examples of physical connections

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 detected fault conditions within the Physical Layer.

7.7.3.7 Quality of service parameters

Quality of service of a physical connection results from data circuits forming this physical-connection. It can be characterized by:

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

b) service availability ;

c) transmission rate ;

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;
c) Physical Layer management.

7.7.4.1 Physical-connection activation and deactivation

The Physical Layer must perform functions associated with the activation and deactivation of physical-connections between two data-link-entities including concatenation of data circuits upon request from the Data Link Layer.

7.7.4.2 Physical-service-data-unit transmission

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

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.

ANNEX A

Brief explanation of how the layers were chosen

This annex provides elements giving additional information to this International Standard, of which it does not form an integral part.

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

a) It is essential that the architecture permit usage of a realistic variety of physical media for interconnection with different control procedures (e.g. V.24, V.25, X.21, etc...). Application of principles P3, P5 and P8 leads to identification of a Physical Layer as the lowest layer in the architecture.

b) Some physical communication media (e.g. 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 (e.g. fibre optics) will require different data-link control procedures. Application of principles P3, P5, and P8 leads to identification of a Data Link Layer on top of the Physical Layer in the architecture.

c) In the open systems architecture, some systems will act as the final destination of data, see 4. Some systems may act only as intermediate nodes (forwarding data to other systems), see figure 13. Application of principles P3, P5 and P7 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).

d) Control of data transportation from source end-system to destination end-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.

e) There is a need to organize and synchronize dialogue, and to manage the exchange of data. Application of principles P3 and P4 leads to the identification of a Session Layer on top of the Transport Layer.

f) 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 P3 and P4 leads to identification of a Presentation Layer on top of the Session Layer.

g) 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.

The resulting architecture with seven layers, illustrated in figure 12 obeys principles P1 and P2.

A more detailed definition of each of the seven layers identified above is given in the remainder of this clause, starting from the top with the Application Layer described in 7.1.2 down to the Physical Layer described in 7.7.2.

ANNEXE B

Alphabetical index to definitions