

INTERNATIONAL STANDARD

ISO
7498-3

First edition
1989-03-01

Information processing systems — Open Systems Interconnection — Basic Reference Model —

Part 3 : Naming and addressing

*Systèmes de traitement de l'information — Interconnexion de systèmes ouverts —
Modèle de référence de base —*

Partie 3 : Dénomination et adressage



Reference number
ISO 7498-3 : 1989 (E)

Contents

Foreword	iv
Introduction	v
1 Scope	1
2 Normative references	1
3 Definitions	1
4 Abbreviations	3
5 Basic concepts of naming	3
6 OSI naming and addressing concepts and the correct use of addresses	4
6.1 The naming of real open systems	4
6.2 The naming and addressing of elements of an (N)-layer	4
6.3 The correct use of (N)-addresses	4
7 OSI addressing model	5
7.1 Associations between peer (N)-entities	5
7.2 Attachment of (N)-entities to (N)-SAPs	5
7.3 (N)-addresses and (N)-SAPs	5
7.4 (N)-directory-functions and directory facilities	7
8 Addressing information and (N)-services	7
8.1 Introduction	7
8.2 Address parameters	8
8.3 Called-(N)-address	8
8.4 Calling-(N)-address	8
8.5 Responding-(N)-address	9
9 Addressing information and (N)-protocols	9
9.1 Introduction	9
9.2 Addressing information in (N)-PAI	9
9.3 Assignment of values to elements of (N)-PAI	10
9.4 Network-addresses and network-PAI	10
9.5 (N)-addresses and (N)-PAI above the Network Layer	10
9.6 Obtaining (N)-PAI	11

© ISO 1989

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

International Organization for Standardization

Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

10	(N)-Directory functions	11
10.1	Introduction	11
10.2	The initiator (N)-directory-functions	11
10.3	The recipient (N)-directory-functions	12
11	Addressing in specific OSI layers	13
11.1	Application processes and the Application Layer	13
11.2	Presentation Layer	14
11.3	Session Layer	15
11.4	Transport Layer	16
11.5	Network layer	16
11.6	Data Link Layer	18
11.7	Physical Layer	18
12	Naming domains and authorities	19
13	Registration procedures for naming within OSI	19
14	Directory facility requirements	20
14.1	Introduction	20
14.2	The Application Title Directory Facility	20
14.3	The Network Address Directory Facility	20

STANDARDSISO.COM : Click to view the full PDF of ISO 7498-3:1989

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75 % approval by the member bodies voting.

International Standard ISO 7498-3 was prepared by Technical Committee ISO/TC 97, *Information processing systems*.

Introduction

This part of the Basic Reference Model for Open Systems Interconnection (ISO 7498) extends the basic architectural concepts of identifiers described in 5.4 of ISO 7498.

This part of ISO 7498 states the architectural principles which are followed in the production of any standard which involves the identification (naming) and location (addressing) of objects for the purpose of interconnection within the Open System Interconnection Environment (OSIE).

This part of ISO 7498 has sufficient flexibility to accommodate advances in technology and expansion in user demands. This flexibility is also intended to allow the phased transition from existing implementations to OSI standards.

NOTE - This part of ISO 7498 is expected to be subject to future expansion, in particular with regard to Multi-Peer Data Transmission (MPDT).

The architectural principles stated within this part of ISO 7498 ensure that any ISO standard that involves the identification and location of objects within the OSIE for the purpose of interconnection will:

- a) avoid any restrictions on:
 - 1) the functionality that may be made available through current or future International Standards,
 - 2) the functionality of any real open system,
 - 3) the internal design of any real open system;
- b) preserve the principle of layer independence in the OSIE. That is, the internal functioning of one layer is not constrained by any other layer;
- c) preserve the principle of implementation independence in the OSIE, as expressed in 4.2 of ISO 7498. That is, no real open system (or administrator thereof) is required to know anything about the implementation design of any other real open system (or administration thereof), nor does any real open system impose such knowledge as a condition for communication using OSI standards;
- d) allow economical support for interconnection within the OSIE; in particular individual standards produced within the

framework specified by this part of ISO 7498 should make it possible to provide facilities which give adequate levels of performance, reliability, and integrity and which ease the administration by humans with respect to identifying and locating objects within the OSIE for the purpose of interconnection.

The description of naming and addressing for the OSIE given in this part of ISO 7498 is developed in stages.

Clauses 1 - 4 provide basic introductory and reference information.

Clause 5 introduces concepts of naming.

Clause 6 prescribes, for the OSIE, the objects named, the operation of addressing, and the uses of addressing.

Clause 7 prescribes, for the OSIE, the objective of naming and addressing and the mechanisms to be employed to meet that objective.

Clause 8 prescribes the principles governing the nature and use of addressing information in (N)-services.

Clause 9 prescribes the principles governing the nature and use of addressing information in (N)-protocols.

Clause 10 provides a layer independent description of the layer directory-functions necessary to support the addressing structure established by clauses 7, 8, and 9, based on the general mechanisms and principles established in clauses 5 and 6.

Clause 11 prescribes the use of the directory-functions in each layer.

Clause 12 defines the nature of addressing domains and registration authorities.

Clause 13 prescribes the registration procedures required for naming in the OSIE.

Clause 14 prescribes the requirements for directory facilities in the OSIE.

NOTE - This part of ISO 7498 provides clarifications of the basic architecture defined in ISO 7498 where this is necessary for a full understanding of the naming and addressing requirements within the OSIE.

STANDARDSISO.COM : Click to view the full PDF of ISO 7498-3:1989

Information processing systems — Open Systems Interconnection — Basic Reference Model —

Part 3 : Naming and addressing

1 Scope

This part of ISO 7498 :

- a) defines general mechanisms for the use of names and addresses to identify and locate objects in the OSIE; and
- b) defines the use of these mechanisms within the layered structure of the Basic Reference Model.

This part of ISO 7498 extends the concepts and principles defined in ISO 7498. This part is not intended to be either an implementation specification or a basis for appraising the conformance of actual implementations.

The specific form of names and addresses is not within the scope of this part.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 7498. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO 7498 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 7498:1984, *Information processing systems - Open systems interconnection - Basic reference model*.

ISO 7498/Add. 1:1984, *Information processing systems - Open systems interconnection - Basic Reference Model - Addendum 1: Connectionless-mode transmission*.

ISO 7498-4:-¹⁾, *Information processing systems - Open systems interconnection - Basic Reference Model - Part 4: OSI management framework*.

ISO 8348/Add. 2:1988, *Information processing systems - Data communication - Network service definition - Addendum 2: Network layer addressing*.

ISO/TR 8509:1987, *Information processing systems - Open systems interconnection - Service conventions*.

ISO 9545:-¹⁾, *Information processing systems - Open systems interconnection - Application layer structure*.

3 Definitions

3.1 This part of ISO 7498 makes use of the following terms defined in ISO 9545:

- a) application-process-type;
- b) application-process-invocation.

3.2 This part of ISO 7498 makes use of the following terms defined in ISO/TR 8509:

- a) (N)-service-request-primitive,
- b) (N)-service-indication-primitive,
- c) (N)-service-response-primitive,
- d) (N)-service-confirm-primitive.

3.3 This part of ISO 7498 makes use of the following term defined in ISO 8348/Add. 2:

- a) subnetwork point of attachment.

3.4 For the purpose of this part of ISO 7498, the following definitions apply.

3.4.1 (N)-address: A name unambiguous within the OSIE which is used to identify a set of (N)-service-access-points which are all located at a boundary between an (N)-subsystem and an (N+1)-subsystem in the same open system.

NOTES

1 This definition of (N)-address is different from that in ISO 7498. This definition is the definitive one and will be moved to ISO 7498 to replace the existing definition when ISO 7498 is revised.

2 A name is unambiguous within a given scope when it identifies one and only one object within that scope. Unambiguity of a name does not preclude the existence of synonyms.

3.4.2 (N)-address-selector; (N)-selector: A n element of addressing information that identifies a set of (N)-SAPs which are all in the same (N)-subsystem; an (N)-selector value is assigned by the local administration.

¹⁾ To be published.

NOTE - The concept of (N)-address-selectors only applies above the Network Layer.

3.4.3 (N)-association: A cooperative relationship among (N)-entity-invocations.

NOTE - This may be formed by the exchange of (N)-protocol-control-information.

3.4.4 calling-(N)-address: A parameter which may appear in an (N)-service request or indication primitive and which identifies the (N)-address at the (N)-initiator.

NOTE - In the service definition of a particular layer, such a parameter may be referred to either as a "calling-(N)-address" or "source-address". Throughout this part of ISO 7498, however, only the term "calling-(N)-address" is used.

3.4.5 called-(N)-address: A parameter which may appear in an (N)-service request or indication primitive which identifies the (N)-address at the (N)-recipient.

NOTE - In the service definition of a particular layer, such a parameter may be referred to either as a "called-(N)-address" or "destination-address". Throughout this part of ISO 7498, however, only the term "called-(N)-address" is used.

3.4.6 descriptive name: A name that identifies a set of one or more objects by means of a set of assertions concerning the properties of the objects of the set.

3.4.7 (N)-directory-function: An (N)-function that processes (N)-addresses, (N-1)-addresses, (N)-entity-titles, and (N)-PAI to provide mappings among these categories of information.

3.4.8 (N)-entity: An active element within an (N)-subsystem embodying a set of capabilities defined for the (N)-layer that corresponds to a specific (N)-entity-type (without any extra capabilities being used).

NOTE - This definition of (N)-entity is different from that in ISO 7498. This definition is the definitive one and will be moved to ISO 7498 to replace the existing definition when ISO 7498 is revised.

3.2.9 (N)-entity-invocation: a specific utilization of part or all capabilities of a given (N)-entity (without any extra capabilities being used).

NOTE - This definition will be moved to ISO 7498 to replace the existing definition when ISO 7498 is revised.

3.4.10 (N)-entity-title: A name that is used to identify unambiguously an (N)-entity.

3.4.11 (N)-entity-type: a description of a class of (N)-entities in terms of a set of capabilities defined for the (N)-layer.

NOTE - This definition will be moved to ISO 7498 to replace the existing definition when ISO 7498 is revised.

3.4.12 generic name: A name of a set of objects.

NOTE - A generic-title is a specific form of generic name.

3.4.13 (N)-initiator: An (N)-entity-invocation which issues an (N-1)-service request primitive.

3.4.14 name: A linguistic construct which corresponds to an object in some universe of discourse.

3.4.15 naming-authority: A registration authority which allocates names according to specified rules. Where the naming-authority allocates titles, it is known as a title-authority. Where the naming-authority allocates addresses, it is known as an addressing-authority.

3.4.16 naming-domain: The set of names that are assignable to objects of a particular type. Where the names are titles, the set is known as a title-domain. Where the names are addresses, the set is known as an addressing-domain.

3.4.17 naming-subdomain: A subset of a naming-domain, which is disjoint from all other naming-subdomains of that naming-domain.

3.4.18 primitive name: A name that identifies an object and which is assigned by a designated naming-authority. The internal structure of the name is not required to be understood or to have significance to users of the name.

3.4.19 (N)-recipient: An (N)-entity-invocation which receives an (N-1)-service indication primitive.

3.4.20 (N)-protocol-addressing-information; (N)-PAI: Those elements of (N)-PCI which contain addressing information.

3.4.21 responding-(N)-address: A parameter which may appear in an (N)-service response or confirm primitive and which identifies the (N)-address at the (N)-recipient.

NOTE - In the service definition of a particular layer, such a parameter may be referred to either as a "called-address" or "responding address." Throughout this part of ISO 7498, however, only the term "responding-(N)-address" is used.

3.4.22 (N)-service-access-point-address; (N)-SAP-address: An (N)-address that is used to identify a single (N)-SAP.

NOTES

1 This definition of (N)-service-access-point-address is different from that in ISO 7498. This definition is the definitive one and will be moved to ISO 7498 to replace the existing definition when ISO 7498 is revised.

2 (N)-address is the general term which applies to any set of (N)-SAPs, including sets of one and only one (N)-SAP. (N)-SAP address is only used where it is necessary to specify precisely that the address identifies one and only one (N)-SAP. Whether an (N)-address is an (N)-SAP address or not is a matter

local to the (N)-subsystem and is not known to other open systems. Nevertheless, at some layers and because of their possible use in subsequent communications, calling-(N)-addresses and responding-(N)-addresses may be constrained to identify a single (N)-SAP (see 8.4.4 and 8.5.5). The decision whether or not to apply this constraint is made on a layer-by-layer and protocol-by-protocol basis.

3.4.23 subnetwork-address: An identifier assigned to a subnetwork point of attachment by the registration authority of the subnetwork.

3.4.24 synonymous name; synonym: A name that identifies an object that is also identified by another distinct name. Synonymous generic names are distinct generic names that name the identical set.

3.4.25 system-title: A name, unique within the OSIE, which is used to identify a single real open system.

4 Abbreviations

For the purposes of this part of ISO 7498, the following abbreviations apply:

(N)-CEPI	(N)-Connection-EndPoint-Identifier
DLSAP	Data-Link-Service-Access-Point
NSAP	Network-Service-Access-Point
OSI	Open Systems Interconnection
OSIE	OSI Environment
(N)-PAI	(N)-Protocol-Addressing-Information
(N)-PCI	(N)-Protocol-Control-Information
PhSAP	Physical-Service-Access-Point
PSAP	Presentation-Service-Access-Point
(N)-SAP	(N)-Service-Access-Point
SNPA	SubNetwork Point of Attachment
SSAP	Session-Service-Access-Point
TSAP	Transport-Service-Access-Point

5 Basic concepts of naming

5.1 Names are linguistic constructs expressed in some language. They correspond to objects in some universe of discourse. The correspondence between names (in the language) and objects (in the universe of discourse) is the

relation of identifying. A name identifies the object to which it is bound.

5.2 Within the context of OSI, names identify particular communications objects in the Open Systems Interconnection Environment (OSIE). There are two distinct kinds of names, primitive and descriptive.

5.3 Within any particular universe of discourse, a primitive name is a name assigned by a naming-authority to a specific object. A naming-authority is simply a source of names. The only architectural constraints imposed upon naming-authorities are that all of the names it provides:

- a) are expressed in a prescribed language; and
- b) are unambiguous (identify just one object).

5.4 A descriptive name consists of a set of assertions which are expressed in a formally defined language. The definition of the formal language determines those linguistic constructs which are well-formed descriptive names. A descriptive name may be incomplete, in that many objects satisfy all the assertions, or it may be complete, in that it serves to identify a single object. A complete descriptive name is equivalent to a primitive name in that it unambiguously identifies an object. Primitive names may be components of a descriptive name.

5.5 Although a primitive name is unambiguous, there may be more than one name that unambiguously identifies the same object.

5.6 A generic name is a primitive name or a descriptive name that identifies a set comprising more than one object with the intent that, when a generic name is used to denote an object, the result is that exactly one member of the set of objects will be selected. A generic name may be used to identify a set of objects of a particular type, which need not be located in the same open system.

5.7 A title is assigned to an object where the purpose of the name is to discriminate among different objects and to permit retrieval of information associated with an object from a Directory Facility. A title is assigned to an object type where the purpose of the name is to discriminate among different object types and to permit retrieval of information associated with an object type from a Directory Facility. The name may identify a system, application-process, application-process-type, (N)-entity, or (N)-entity-type.

NOTE - These objects and types are defined in either ISO 7498 or ISO 9545.

5.8 An identifier is assigned to an object where the purpose of the name is only to discriminate among occurrences of this object. The name may identify an (N)-

association, an application-process-invocation, or an (N)-entity-invocation.

NOTE - These objects are defined in either ISO 7498 or ISO 9545.

6 OSI naming and addressing concepts and the correct use of addresses

6.1 The naming of real open systems

6.1.1 A system-title is a layer independent primitive name, i.e., the same identifier is used within various layers to identify the same real open system. A single real open system is named by one and only one system-title.

6.1.2 A system-title is used to identify a real open system as a whole. It may also be used:

- a) in conjunction with other qualifiers to identify specific OSI resources in the relevant parts of the management information base within the real open system; or
- b) as an attribute of a Directory Facility entry pertaining to an OSI resource associated with a single real open system.

6.2 The naming and addressing of elements of an (N)-layer

6.2.1 Introduction

6.2.1.1 Since an (N)-entity-type describes a class of (N)-entities, it needs to be named, but not located. Since (N)-entities and (N)-entity-invocations are active elements within an (N)-layer, they need to be unambiguously identified and located.

6.2.1.2 Within an open system, (N+1)-entities and (N)-entities are bound together at (N)-service-access-points [(N)-SAPs]. (N)-entities provide services to (N+1)-entities via the exchange of service primitives at (N)-SAPs.

6.2.1.3 An (N)-entity is identified unambiguously by an (N)-entity-title. An (N)-entity-type is identified by an (N)-entity-type-title. An (N)-entity-invocation is identified by an (N)-entity-invocation-identifier which is unambiguous within the scope of an (N)-entity.

6.2.2 (N)-addresses

6.2.2.1 An (N)-address identifies a set of (N)-SAPs which are all located at the boundary between an (N)-subsystem and an (N+1)-subsystem. An (N)-SAP-ad-

dress is an (N)-address which identifies a set containing exactly one (N)-SAP.

6.2.2.2 While (N)-entities are the objects being addressed, the result of communication to an address is communication with an (N)-entity-invocation.

6.2.2.3 An (N+1)-entity is located by its binding to one or more (N)-SAPs. An (N)-SAP is identified by one or more (N)-addresses.

NOTE - A physical-address is used to access a data-link-entity; a data-link-address is used to access a network-entity; a network-address is used to access a transport-entity; a transport-address is used to access a session-entity; a session-address is used to access a presentation-entity; and a presentation-address is used to access an application-entity.

6.2.3 (N)-selectors

An (N)-selector is that part of the addressing information which is specific to the (N)-subsystem. (N)-selectors are used to identify (N)-SAPs or sets of (N)-SAPs within an end open system, once this end open system is unambiguously identified. Since the end open system is implicitly known at the Network Layer, (N)-selectors are used above the Network Layer, along with local information, to address the desired (N+1)-entity within the open system. (N)-selector values are exchanged between open systems as part of the (N)-PAI.

6.3 The correct use of (N)-addresses

6.3.1 (N)-addresses have a limited scope. They are used to distinguish among sets of (N)-SAPs, and only (N)-SAPs. Addressing rules are not used to make the structure of a real open system visible to the OSI environment.

6.3.2 (N)-addresses are used to identify sets of (N)-SAPs in order to locate (N+1)-entities. An (N+1)-subsystem is partitioned into (N+1)-entities:

- a) to support different (N+1)-protocols or sets of (N+1)-protocols;
- b) to accommodate security and/or management requirements; and
- c) in the case of the application-subsystem, to distinguish between different application-processes and different application-entities of the same application-process.

6.3.3 (N)-addresses are not used:

- a) to distinguish among aspects of protocols that are subject to negotiation (classes, subsets,

quality of service, protocol versions) or parameter values;

- b) to derive routing information above the Network Layer; or
- c) to distinguish among hardware components.

NOTE - In some configurations, the use of an (N)-address as defined in 6.3.2 can lead to an (N+1)-entity being wholly contained within a single hardware component. Nevertheless, within the OSIE, the (N)-address identifies the (N+1)-entity; it does not identify the hardware component.

7 OSI addressing model

7.1 Associations between peer (N)-entities

7.1.1 An (N)-association is a cooperative relationship between two (N)-entity-invocations. Cooperation between (N)-entity-invocations requires the establishment and maintenance of related state information in each (N)-entity-invocation. This state information supports an (N)-association between the (N)-entity-invocations.

7.1.2 An (N)-entity invocation may support one or more independent (N)-associations at any one time. The communications behavior of the (N)-entity-invocation with respect to a specific (N)-association is defined by the (N)-entity and by the state information which is maintained by the (N)-entity-invocation and is specific to that (N)-association.

7.1.3 An (N)-association-identifier is associated with each (N)-association. This identifier is unique within the scope of a pair of cooperating (N)-entity-invocations. It serves to identify the related state information associated with each (N)-entity-invocation. The identifier has two components, one being determined by each (N)-entity-invocation.

NOTE - Certain (N)-protocols may not need explicit (N)-association-identifiers.

7.1.4 Two (N)-entity-invocations can establish (N-1)-connection(s), or can make use of an (N-1)-connection-less service, to support an (N)-association. The lifetime of an (N)-association may exceed the lifetime of any supporting (N-1)-connection(s). The binding of an (N)-association with (N-1)-connection(s) may change with time.

NOTE - An (N)-association could be associated with a sequence of (N-1)-connections with a one-to-one binding at any point in time; alternatively in the case of splitting, there could be a one-to-many binding at any point in time.

7.1.5 When the operation of an (N)-association requires it, (N)-entity-titles are used to identify (N)-entities independent of their locations. When the operation of an (N)-association requires it, (N-1)-addresses are used in requests for (N-1)-services to identify the locations of the (N)-entities concerned.

7.2 Attachment of (N)-entities to (N)-SAPs

An (N)-entity may provide (N)-services through one or more (N)-SAPs and may use (N-1)-services through one or more (N-1)-SAPs. In consequence an (N)-entity may have the following relationships with (N)-SAPs and (N-1)-SAPs (see Figure 1):

- a) an (N)-entity may provide (N)-services through one (N)-SAP making use of (N-1)-services through one (N-1)-SAP;
- b) an (N)-entity may provide (N)-services through multiple (N)-SAPs making use of (N-1)-services through one (N-1)-SAP;
- c) an (N)-entity may provide (N)-services through one (N)-SAP making use of (N-1)-services through multiple (N-1)-SAPs;
- d) an (N)-entity may provide (N)-services through multiple (N)-SAPs making use of (N-1)-services through multiple (N-1)-SAPs;

NOTES

1 There is no relationship between the SAP/entity correspondences identified above and multiplexing. An (N)-multiplexing function provides for the mapping of several (N)-connections onto a single (N-1)-connection. The (N)-connections may all terminate in a single (N)-SAP or they may terminate in separate (N)-SAPs. Multiplexed (N)-connections are distinguished from each other by elements of (N)-PCI at the service boundary and by elements of the (N)-PAI, e.g. an association-identifier within the (N)-protocol.

2 Logical channel numbers in X.25 (ISO 8208) and connection references in the OSI Transport Protocol (ISO 8073) are examples of information exchanged in (N)-PCI to distinguish connections when multiplexing is used.

7.3 (N)-addresses and (N)-SAPs

7.3.1 The OSI addressing structure allows:

- a) (N)-addresses to identify the location of an (N+1)-entity without constraining the structure of lower layer subsystems in the open system concerned; and
- b) multiple (N)-entities to be defined within an (N)-subsystem.

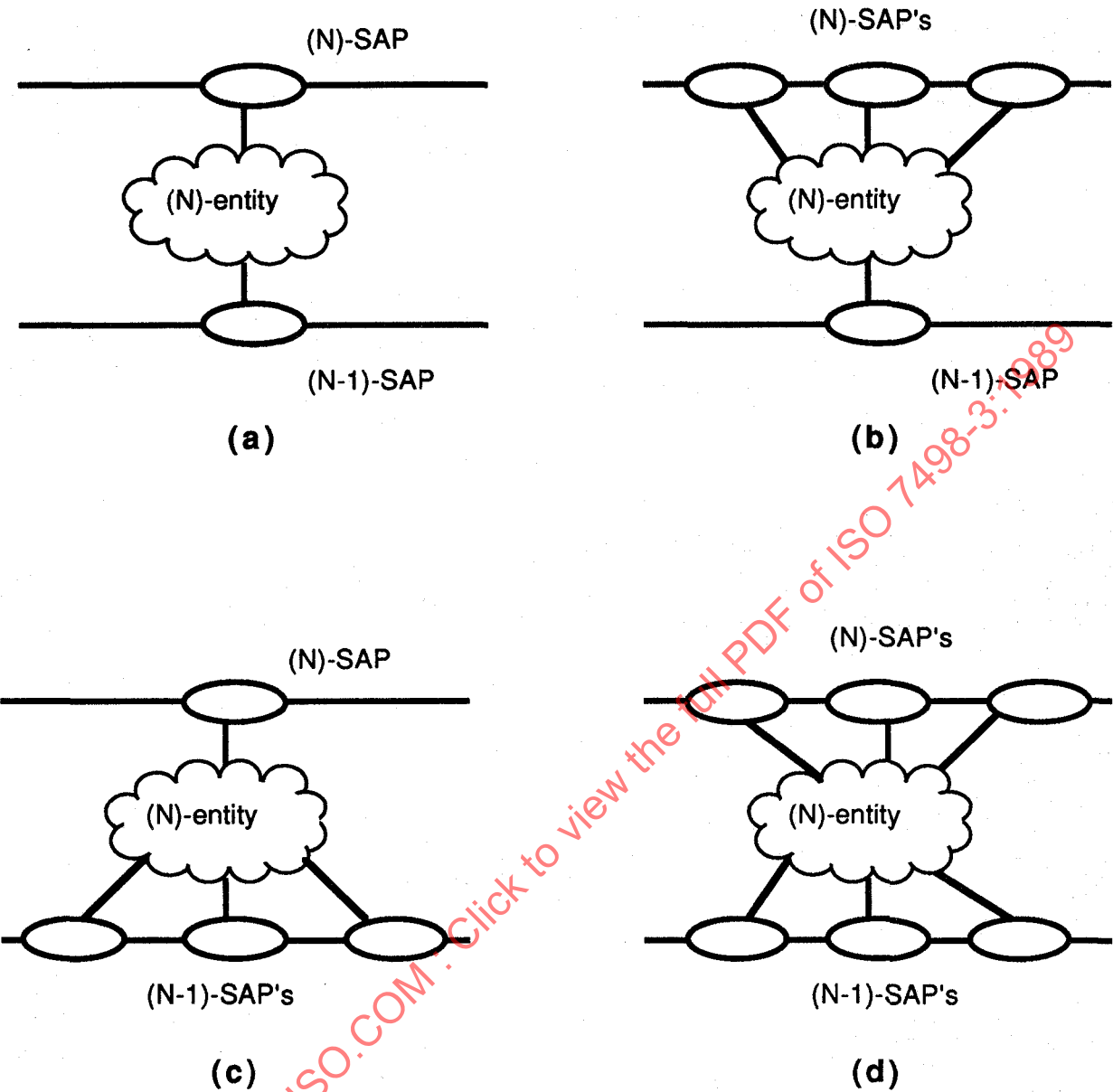


Figure 1 — Relationships of (N)-entity to (N)-SAPs and to (N-1)-SAPs

NOTE - The relevant addressing structure allows a presentation-address to identify the location of an application-entity without constraining the structure of presentation-, session-, and transport-subsystems within an open system; and allows the definition of a single set of addressing information for use in establishing communication with an application-entity in a recipient system.

7.3.2 An (N)-address identifies a set of (N)-SAPs all located at the boundary of a single (N)-subsystem. The exact membership of the set is an issue local to that (N)-subsystem. The set membership is not known to other open systems and may change over time.

7.3.3 The set of (N)-SAPs identified by an (N)-address may consist of:

- a) a single (N)-SAP bound to an (N+1)-entity; or
- b) multiple (N)-SAPs that are bound to a single (N+1)-entity; or
- c) multiple (N)-SAPs that are bound to different (N+1)-entities.

7.3.4 When an (N)-address is used as a called-(N)-address in a service primitive, the recipient (N)-subsystem will select a single (N)-SAP from the set identified by the (N)-address. The selection mechanism is a local issue transparent to the (N)-initiator.

7.3.5 Open systems are configured to ensure that all of the (N)-SAPs in the set identified by an (N)-address are bound to (N+1)-entities that are of the same type and, therefore, provide the same functions.

7.3.6 It is important to distinguish between the semantics of an (N)-address and the syntax used to represent an (N)-address within a given open system. (N)-addresses are passed across layer boundaries within open systems as parameters of (N)-service primitives. For (N)-service request/response primitives, the semantics of (N)-addresses are conveyed to the peer (N)-subsystem and passed across the layer boundary as parameters of the (N)-service indication/confirm primitives. Only the semantics of an (N)-address are conveyed by the (N)-service. The syntax of an (N)-address is a local issue and different representations may be used in different open systems.

7.3.7 When an (N+1)-entity establishes an (N)-connection with another (N+1)-entity, each (N+1)-entity is given an (N)-connection-endpoint-identifier [(N)-CEPI] by its supporting (N)-entity. (See 5.4.2 of ISO 7498). An (N)-CEPI is a local identifier determined at connection establishment time. An (N)-CEPI cannot be used as a substitute for an (N)-address. In the case that the calling-(N)-address and called-(N)-address of an (N)-connection are identical, the (N)-connection has two (N)-connection-end-points and two (N)-CEPIs (connection of an (N)-entity to itself). How the two (N)-CEPIs in the

(N)-subsystems are distinguished is an entirely local matter.

7.4 (N)-directory-functions and directory facilities

7.4.1 (N)-directory-functions process (N)-addresses, (N-1)-addresses, (N)-entity-titles, and (N)-PAI to provide mappings among these categories of information. Information used for these mappings is held by a Directory Facility. It is a local system management responsibility to access the Directory Facility to retrieve the information and make it available to an (N)-directory-function.

7.4.2 Some of this information represents the logical structure of the local end system and influences local operation. This information is stored locally. Other information represents the logical structure of the remote end system and influences the generation of (N)-PAI. This information may be stored locally or remotely. If it is stored remotely, OSI protocols are used to access that information.

8 Addressing information and (N)-services

8.1 Introduction

8.1.1 This clause provides a layer independent description of the use of (N)-addresses in (N)-service primitives.

8.1.2 (N+1)-entities use (N)-services by issuing (N)-service primitives at (N)-SAPs. Issuing an (N)-service request/response primitive may cause an (N)-service indication/confirm primitive to be issued at an (N)-SAP to which a peer (N+1)-entity is attached.

8.1.3 The (N)-address derived from information provided by a Directory Facility may be invalid. An (N)-address derived from the calling/responding-(N)-address parameter of a previously received (N)-service indication/confirm primitive has to be valid at the time it is issued, but no guarantee can be given for a subsequent use of this address. Therefore, an (N+1)-entity using an (N)-address should, under all circumstances, check that it has led to communication with the desired correspondent in the (N+1)-layer. It is normally sufficient to do this in the Application Layer by exchanging application-entity-titles.

8.1.4 The use of an (N)-address is not sufficient by itself to identify a particular (N+1)-entity-invocation. An (N+1)-entity may be satisfied to communicate with any (N+1)-entity-invocation of the desired (N+1)-entity at the (N)-address. In some (N+1)-layers, it may be nec-

essary to reference the (N+1)-entity-invocation using the (N+1)-entity-invocation-identifier.

8.2 Address parameters

8.2.1 It is important to distinguish between the (N)-addresses passed as called-(N)-address parameters and those passed as calling-(N)-address or responding-(N)-address parameters.

8.2.2 Called-(N)-addresses are used in the initiation of communication between (N+1)-entity-invocations. The (N+1)-initiator provides the called-(N)-address the semantics of which are conveyed to the peer (N+1)-recipient.

8.2.3 Calling- and responding-(N)-addresses are primarily used for identification and recall purposes and may identify the specific (N)-SAPs used in an instance of communication.

8.3 Called-(N)-address

8.3.1 The called-(N)-address parameter in connection-mode service primitives is equivalent to the destination (N)-address parameter in connectionless-mode service primitives.

8.3.2 The called-(N)-address is provided by the (N+1)-initiator. The semantics of the (N)-address are conveyed to the recipient (N)-subsystem and passed to the (N+1)-subsystem in an (N)-service indication primitive.

8.3.3 The called-(N)-address conveyed in the (N)-service indication primitive parameter is not restricted to be the same address as was specified in the associated request primitive. However, an (N)-service definition may impose such a restriction.

8.3.4 Above the Network Layer, address processing is confined to end systems:

- a) at the initiating open system, the processing of the called-(N)-address is not dependent on the complexity of the address structures supported by the recipient open system; and
- b) at the recipient open system, the processing of the called-(N)-address depends on the complexity of the address structures supported by this system.

8.3.5 In the Network Layer, although some processing of the called-(N)-address may occur in an intermediate system, this processing is not dependent on the complexity of the address structures supported by the recipient open system.

8.3.6 The called-(N)-address identifies a set of (N)-SAPs at the recipient-(N)-subsystem. Any one of the (N)-SAPs in this set may be used to support the communication. The resolution of the address to select a particular (N)-SAP is the responsibility of the recipient (N)-subsystem.

8.3.7 The called-(N)-address may have been derived from information obtained from the Directory Facility. In this case, the semantics of the (N)-address are related to a directory entry published on behalf of the recipient system. The attributes associated with the Directory Facility entry are known within the recipient system. The called-(N)-address identifies a set of (N)-SAPs that provide access to (N+1)-entities which support communication in a manner that is consistent with the information obtained from the Directory Facility.

8.3.8 The called-(N)-address may have been previously passed as a calling- or responding-(N)-address parameter by the recipient (N)-subsystem in a previous instance of communication. In this case, the (N)-address identifies the set of (N)-SAPs consistent with the requirements concerning calling- or responding-(N)-addresses described in clauses 8.4 and 8.5.

8.3.9 The called-(N)-address may have been obtained by private arrangement. In this case, the called-(N)-address identifies a set of (N)-SAPs that provide access to (N+1)-entities that support communication in a manner consistent with that private arrangement.

8.4 Calling-(N)-address

8.4.1 The calling-(N)-address parameter in connection-mode service primitives is equivalent to the source (N)-address parameter in connectionless-mode service primitives.

8.4.2 The calling-(N)-address is provided by the (N+1)-initiator. The semantics of the (N)-address are conveyed to the recipient (N)-subsystem and passed to the (N+1)-subsystem in an (N)-service indication primitive.

8.4.3 Provided (N)-protocol specifications and OSI management standards do not impose constraints, the recipient (N+1)-subsystem may use the calling-(N)-address in any of the following ways:

- a) as a called-(N)-address in a subsequent request primitive that is not related to the initial communication;
- b) as a called-(N)-address in a subsequent request primitive that is related to the initial communication, in order to facilitate connection re-establishment or splitting;

- c) as a called-(N)-address which is forwarded to some other open system;
- d) for management purposes.

8.4.4 While valid, the calling-(N)-address will identify a set of (N)-SAPs at the initiating (N)-entity. The set of (N)-SAPs identified may be constrained by layer specific requirements concerning calling-(N)-addresses. For example, a layer may require a calling-(N)-address to identify the single (N)-SAP actually used to support the original communication.

8.4.5 When the calling-(N)-address received in an (N)-service indication primitive is used by the recipient (N)-subsystem as a called-(N)-address in a subsequent (N)-service request primitive, this subsystem should be aware of the possibility that this address may be no longer valid in the sense defined in 8.4.4 and should take appropriate measures.

8.5 Responding-(N)-address

8.5.1 The responding-(N)-address is used in (N)-service response/confirm primitives.

NOTE - In some OSI service definitions the term "called address" is used in response and confirm primitives to denote the responding-(N)-address parameter.

8.5.2 The responding-(N)-address is provided by the (N+1)-recipient. The semantics of the (N)-address are conveyed to the initiating (N)-subsystem and passed to the initiating (N+1)-subsystem in a confirm service primitive.

8.5.3 Provided (N)-layer protocol specifications and OSI management standards do not impose constraints, the initiating (N+1)-subsystem may use the responding-(N)-address in any of the following ways:

- a) as a called-(N)-address in a subsequent request primitive not related to the initial communication;
- b) as a called-(N)-address in a subsequent request primitive related to this communication, e.g. to facilitate connection re-establishment or splitting;
- c) as a called-(N)-address which is forwarded to some other open system;
- d) for management purposes.

8.5.4 The responding-(N)-address may differ from the called-(N)-address specified in the related (N)-service indication primitive.

8.5.5 While valid, the responding-(N)-address will identify a set of (N)-SAPs at the recipient (N)-entity. The set of (N)-SAPs identified may be constrained by

layer specific requirements concerning responding-(N)-addresses. For example, a layer may require a responding-(N)-address to identify the single (N)-SAP actually used to support the communication.

8.5.6 When the responding-(N)-address received in an (N)-service confirm primitive is used by the initiating (N)-subsystem as a called-(N)-address in a subsequent (N)-service request primitive, this subsystem should be aware of the possibility that this address may be no longer valid in the sense defined in 8.5.5 and should take appropriate measures.

9 Addressing information and (N)-protocols

9.1 Introduction

This clause provides a layer independent description of the use of addressing information in (N)-Protocol-Addressing-Information [(N)-PAI]. (N)-PAI are those elements of (N)-PCI which contain addressing information.

9.2 Addressing information in (N)-PAI

9.2.1 The semantics of an (N)-address are conveyed by means of (N)-protocol exchanges between (N)-entity-invocations. For some layers, the full semantics of (N)-addresses are conveyed in (N)-PAI. In other layers, it is not necessary for the full semantics of (N)-addresses to be represented in the (N)-PAI and, for these layers, the full semantics of (N)-addresses are conveyed by a combination of:

- a) the exchange of (N)-PAI; and
- b) local information about the scope of the (N)-PAI.

NOTES

1 For example, the network-entities exchange network-addresses. In this case, the Network-PAI includes the network-address.

2 The values of (N)-PAI may include information relevant to the operation of both the (N)-layer and the (N+1)-layer. Nevertheless, a given layer only makes use of the information relevant to that layer.

9.2.2 Below the Network Layer, communicating (N)-entities are confined to a single subnetwork. The (N)-PAI exchanged need not be globally applicable, since it can be interpreted within the scope of the subnetwork.

9.2.3 At the Network Layer, communicating (N)-entities may be attached to different subnetworks. Consequently, the (N)-PAI exchanged has to be globally applicable. For this reason, the Network-PAI alone provides for the exchange of the complete semantics of a network-address.

9.2.4 Above the Network Layer, the scope of the (N)-PAI is restricted to the communicating end systems. At these layers, the semantics of the (N)-address comprise:

- a) the identification of a set of (N)-SAPs; this identification is unambiguous within the scope of the (N)-subsystem which contains the (N)-SAPs and is provided by (N)-selectors exchanged in (N)-PAI and local information about the scope of the (N)-selectors within the (N)-subsystem; and
- b) the identification of the end systems, derived from the Network Layer exchange of network-addresses.

NOTE - At the Application Layer, titles and identifiers, not addressing information, are exchanged.

9.3 Assignment of values to elements of (N)-PAI

9.3.1 Layer protocol specifications define elements of (N)-PAI to be used for the exchange of addressing information. Different elements of (N)-PAI are used to convey the semantics of:

- called-(N)-addresses,
- calling-(N)-addresses, and
- responding-(N)-addresses.

9.3.2 The values of the elements used to convey calling-(N)-address semantics are provided by the (N)-initiator. These values may be retained by the recipient (N)-subsystem and used in a subsequent request primitive to convey called-(N)-address semantics to the original initiating (N)-subsystem.

9.3.3 The values of the elements used to convey responding-(N)-address semantics are provided by the recipient (N)-subsystem. These values may be retained by the initiating (N)-subsystem and used in a subsequent (N)-service request primitive to convey called-(N)-address semantics to the recipient (N)-subsystem.

9.3.4 The values of the elements used to convey called-(N)-address semantics may be obtained:

- a) from a Directory Facility,
- b) by private arrangement, or
- c) from previously issued (calling-) responding-(N)-addresses.

9.4 Network-addresses and network-PAI

The complete semantics of the network-address is conveyed in the Network-PAI. The network-address is globally applicable and is issued by the appropriate registration authority.

9.5 (N)-Addresses and (N)-PAI above the Network Layer

9.5.1 (N)-selectors are unambiguous within the scope of an (N)-subsystem. The values of (N)-selectors are chosen by the local administration of an open system and there is no requirement for an OSI addressing authority, although the chosen values has to be known by systems which want to communicate. When an (N)-selector identifies a set of (N)-SAPs, the resolution of that (N)-selector is the responsibility of the recipient (N)-subsystem.

NOTES

1 All (N)-entities refer to a particular set of (N)-SAPs in the same way, (i.e. the (N)-selector value which is associated with this set of (N)-SAPs is the same regardless of the (N)-entity which handles the communication).

2 The way unambiguity is achieved is a local matter. Local administration of the open system may decide to achieve that goal by defining (N)-selectors that are unique within the scope of the (N)-subsystem. In such a case, the semantics of the (N)-selector is directly derived from the value carried in the (N)-PAI, regardless of the (N)-entity which handles the communication. Where (N)-selectors are unambiguous within the scope of the (N)-subsystem, without being unique within that scope, additional information local to the recipient open system, is necessary (in particular, the semantics of the (N)-selector depend on the (N)-entity which handles the communication).

9.5.2 Protocol specifications may designate (N)-PAI as optional and therefore it may be absent. Since the (N)-PAI in an (N)-protocol is the (N)-selector, the (N)-selector may be absent. There is no distinction between the absence of an (N)-selector and the presence of a nil (N)-selector value in connectionless-mode operation. In connection mode operation, the absence of an (N)-selector is equivalent to the presence of a nil (N)-selector value for calling- and called-(N)-addresses. For responding-(N)-addresses, the absence of an (N)-selector indicates that the responding-(N)-address is equivalent to the called-(N)-address.

NOTE - Within layers which use the Type-Length-Value (TLV) encoding technique:

- a) "absence of selector" means that no parameter of the type used to convey this selector is present;
- b) the "nil selector value" corresponds to a zero value for the length field of the parameter which is of the type used to convey this selector; and
- c) if the parameter type which corresponds to the type used to convey the selector is present and if the associated parameter length is not zero, then the selector value is not considered as nil whatever the encoding of this value might be.

9.5.3 The nil (N)-selector value (or the absence of a value) is only used in (N)-PAI to convey called-(N)-address semantics when the nil value has been specified:

- a) by a value from an entry in a Directory Facility;
or
- b) as the (N)-PAI used to convey the semantics of a previously issued calling/responding-(N)-address;
or
- c) by private arrangement.

9.5.4 The (N)-recipient uses the nil (N)-selector value according to local information to select an (N)-SAP.

NOTE - The use of a nil (N)-selector value does not preclude the use of other (N)-selector values by the local administration of an open system.

9.6 Obtaining (N)-PAI

9.6.1 Information about application-entities is obtained from the Application Title Directory Facility (see clause 14). Included in this information is a single tuple specifying the addressing (N)-PAI values required to access the application-entities through a PSAP. The tuple is of the form:

(P-selector, S-selector, T-selector, list of network-addresses)

NOTE - Each of the (N)-PAI values derived from the tuple may be used by the relevant (N)-recipient to identify a set of (N)-SAPs. The fact that the (N)-addressing information may identify a set of (N)-SAPs is known only by the recipient (N)-subsystem.

9.6.2 All of the network-addresses of the list belong to a single open system. At the initiating open system, one of the network-address values is selected by local system management for a given instance of communication.

9.6.3 The T-selector is the single T-selector value that, when used in transport-PAI, identifies the set of TSAPs at the open system to which the set of network-addresses in the tuple applies. The selector value is equally valid regardless of which network-address is used.

9.6.4 The S-selector is the single S-selector value that, when used in session-PAI, identifies the set of SSAPs at the open system to which the set of network-addresses in the tuple applies. The selector value is equally valid regardless of which network-address is used.

9.6.5 The P-selector is the single P-selector value that, when used in presentation-PAI, identifies the set of PSAPs at the open system to which the set of network-addresses in the tuple applies. The selector value is equally valid regardless of which network-address is used.

10 (N)-directory-functions

10.1 Introduction

10.1.1 (N)-directory-functions process (N)-addresses, (N-1)-addresses, (N)-entity-titles, (N)-PAI, and possibly routing information to provide mappings among these categories of information. These functions are performed by the (N)-entity within the (N)-layer during connection establishment or connectionless data transmission:

- a) when an (N)-service request primitive is received from the (N+1)-layer, or an (N-1)-service confirmation primitive is received from the (N-1)-layer (the initiator (N)-directory-functions); and
- b) when an (N-1)-service indication primitive is received from the (N-1)-layer, or an (N)-service response primitive is received from the (N+1)-layer (the recipient (N)-directory-functions).

10.1.2 Information on these mappings may be held by local system management and made available for access by (N)-directory-functions or it may be held by a Directory Facility. If information is needed from a Directory Facility, it is obtained by local system management and made available to (N)-directory-functions.

10.2 The initiator (N)-directory-functions

10.2.1 The parameters of the initiator (N)-directory-functions used for connection establishment or connectionless data transmission are:

- a) the called-(N)-address supplied by the (N+1)-layer, (CALLED-(N)-ADDRESS);
- b) the calling-(N)-address supplied by the (N+1)-layer, (CALLING-(N)-ADDRESS);
- c) the called-(N)-entity-title supplied by the (N)-layer, (CALLED-(N)-ENTITY-TITLE);
- d) the called-(N-1)-address generated by an initiator (N)-directory-function, (CALLED-(N-1)-ADDRESS);
- e) the responding-(N)-PAI supplied by the (N-1)-layer, (RESPONDING-(N)-PAI);
- f) the responding-(N-1)-address supplied by the (N-1)-layer, (RESPONDING-(N-1)-ADDRESS); and
- g) information (LOCAL) made available by local system management to the (N)-directory-functions on loading, quality of service requirements, and other local information.

NOTE - A layer need not use all of these parameters for its initiator (N)-directory-functions.

10.2.2 Taking these parameters as input the initiator (N)-directory-functions generate the following information:

- a) the called-(N)-PAI to be carried in (N)-PCI, (CALLED-(N)-PAI);
- b) the calling-(N)-PAI to be carried in (N)-PCI, (CALLING-(N)-PAI);
- c) the calling-(N-1)-address to be passed in the (N-1)-service request primitive, (CALLING-(N-1)-ADDRESS);

NOTE - The choice of the (N-1)-SAP at which this (N-1)-service request primitive is issued is a local matter. This choice has to be consistent with the calling-(N-1)-address.

- d) the called-(N-1)-address to be passed in the (N-1)-service-request-primitive, (CALLED-(N-1)-ADDRESS); and
- e) the responding-(N)-address to be passed in the (N)-service confirmation primitive, (RESPONDING-(N)-ADDRESS);
- f) routing information, (ROUTING INFORMATION).

NOTE - The nature of the ROUTING INFORMATION and its use by the (N)-layer depend on the detailed architecture of the routing function within the (N)-layer.

10.2.3 There are seven initiator (N)-directory-functions.

- a) The Initiator Addressing Function 1 - IAF1. For this function,
 - 1) the input parameters are: CALLED-(N)-ENTITY-TITLE and LOCAL;
 - 2) the output is: CALLED-(N-1)-ADDRESS.
- b) The Initiator Addressing Function 2 - IAF2. For this function,
 - 1) the input parameters are: CALLED-(N)-ADDRESS and LOCAL;
 - 2) the output is: CALLED-(N-1)-ADDRESS.
- c) The Initiator Addressing Function 3 - IAF3. For this function,
 - 1) the input parameters are: CALLED-(N-1)-ADDRESS, CALLING-(N)-ADDRESS and LOCAL;
 - 2) the output is: CALLING-(N-1)-ADDRESS.

d) The Initiator Addressing Function 4 - IAF4. For this function,

- 1) the input parameters are: RESPONDING-(N-1)-ADDRESS and RESPONDING-(N)-PAI;
- 2) the output is: RESPONDING-(N)-ADDRESS.

e) The Initiator PAI Function 1 - IPF1. For this function,

- 1) the input parameter is: CALLED-(N)-ADDRESS;
- 2) the output is: CALLED-(N)-PAI.

f) The Initiator PAI Function 2 - IPF2. For this function,

- 1) the input parameters is: CALLING-(N)-ADDRESS;
- 2) the output is: CALLING-(N)-PAI.

g) The Initiator Routing Function 1 - IRF1. For this function,

- 1) the input parameters are: CALLED-(N)-ADDRESS and LOCAL;
- 2) the output is: ROUTING INFORMATION.

10.3 The recipient (N)-directory-functions

10.3.1 The parameters of the recipient (N)-directory-functions used for connection establishment or connectionless data transmission are:

- a) the called-(N-1)-address supplied by the (N-1)-layer, (CALLED-(N-1)-ADDRESS);
- b) the calling-(N-1)-address supplied by the (N-1)-layer, (CALLING-(N-1)-ADDRESS);
- c) the called-(N)-PAI carried in the (N)-PCI, (CALLED-(N)-PAI);
- d) the calling-(N)-PAI carried in the (N)-PCI, (CALLING-(N)-PAI);
- e) the responding-(N)-address supplied by the (N+1)-layer, (RESPONDING-(N)-ADDRESS); and
- f) information (LOCAL) locally known, identifying the scope of (N)-PAI.

NOTE - A layer need not use all of these parameters for its recipient (N)-directory-functions.

10.3.2 Taking these parameters as input the recipient (N)-directory-functions generate the following information:

- a) the called-(N)-address to be passed in the (N)-service indication primitive, (CALLED-(N)-ADDRESS);

NOTE - The choice of the (N)-SAP at which this (N)-service indication primitive is issued is a local matter. This choice must be consistent with the called-(N)-address.

- b) the calling-(N)-address to be passed in the (N)-service indication primitive (CALLING-(N)-ADDRESS);
- c) the responding-(N-1)-address to be passed in the (N-1)-service response primitive, (RESPONDING-(N-1)-ADDRESS); and
- d) the responding-(N)-PAI carried in (N)-PCI, (RESPONDING-(N)-PAI).

10.3.3 There are four recipient (N)-directory-functions.

- a) The Recipient Addressing Function 1 - **RAF1**. For this function,

- 1) the input parameters are: CALLED-(N)-PAI, CALLED-(N-1)-ADDRESS, and LOCAL;
- 2) the output is: CALLED-(N)-ADDRESS.

- b) The Recipient Addressing Function 2 - **RAF2**. For this function,

- 1) the input parameters are: CALLING-(N)-PAI and CALLING-(N-1)-ADDRESS;
- 2) the output is: CALLING-(N)-ADDRESS.

- c) The Recipient Addressing Function 3 - **RAF3**. For this function,

- 1) the input parameters are: CALLED-(N-1)-ADDRESS and LOCAL;
- 2) the output is: RESPONDING-(N-1)-ADDRESS.

- d) The Recipient PAI Function - **RPF1**. For this function,

- 1) the input parameter is: RESPONDING-(N)-ADDRESS;
- 2) the output is: RESPONDING-(N)-PAI.

11 Addressing in specific OSI layers

11.1 Application-processes and the Application Layer

This clause is concerned with the naming of elements of Application-processes and the Application Layer. A complete description of these elements is given in ISO 9545.

11.1.1 Application-processes and Application Layer elements

11.1.1.1 Application-processes are identified by application-process-titles that are unambiguous throughout the OSIE. An application-process-title is a single name which, for convenience, may be structured internally. In particular, for some application-processes the internal structure of the application-process-title may be based on system-title.

NOTES

1 The purpose of structuring an application-process-title from a system-title is to make it possible to register application-processes within the system on which they reside once its system-title has been registered.

2 Application-process-titles can have synonyms. That is, an application-process may be known to one or more application-processes by different application-process-titles.

11.1.1.2 Application-entities are identified by application-entity-titles that are unambiguous throughout the OSIE. An application-entity-title is composed of an application-process-title and an application-entity-qualifier. This division into two components permits the user of the application-entity-title to obtain information specific to the application-process or to the application-entity. The application-entity-qualifier is unambiguous within the scope of the application-process. Each application-entity-title is associated with a presentation-address.

NOTE - Application-entity-titles can have synonyms. That is, an application-entity may be known to one or more application-entities by different application-entity-titles.

11.1.1.3 Where application-process-invocations have to be identified, this is done by means of application-process-invocation-identifiers that are unambiguous within the scope of an application-process. An application-process-invocation is identified unambiguously within the OSIE by the application-process-invocation-identifier qualified by the application-process-title.

11.1.1.4 Where application-entity-invocations have to be identified, this is done by means of application-entity-invocation-identifiers that are unambiguous within the scope of a pair: (application-process-invocation, application-entity). An application-entity-invocation is identified unambiguously within the OSIE by the application-entity-invocation-identifier qualified by the appli-

cation-entity-qualifier, the application-process-invocation-identifier, and the application-process-title.

NOTE - The following table provides a summary of the identifiers described above:

ITEM	Identified by AE	APT	APII	AEO	AEII
Appl. Process		+			
Appl. Process Invocation		+	+		
Appl. Entity		+		+	
Appl. Entity Invocation		+	+	+	+

APT = application-process-title
APII = application-process-invocation-identifier
AEO = application-entity-qualifier
AEII = application-entity-invocation-identifier

11.1.1.5 Where application-associations have to be identified, this is done by means of application-association-identifiers that are unambiguous within the scope of the application-entity-invocations at the endpoints of the association.

11.1.1.6 Where application-process-types have to be identified, this is done by means of an application-process-type-title that is unambiguous throughout the OSIE. An application-process-type-title may be used to denote the distributed processing capabilities of an application-process.

11.1.1.7 Where application-entity-types have to be identified, this is done by means of an application-entity-type-title that is unambiguous throughout the OSIE. An application-entity-type-title may be used to denote the communications capabilities of an application-entity.

11.1.1.8 At any instant of time, each application-entity-title is bound to a single presentation-address that identifies the set of PSAPs to which the application-entity is attached. This binding is recorded in the Application Title Directory Facility (see clause 14).

11.1.2 Application-association

11.1.2.1 In order for an application-entity-invocation to establish an application-association with another application-entity-invocation, it uses the presentation-address of the called application-entity to establish a presentation-connection or to use the presentation connectionless-mode service. This presentation-address may be obtained from the application-directory-function, IAF1, using the called-application-entity-title.

11.1.2.2 If it is necessary to confirm that the required application-entity is still attached to the PSAP identified by the presentation-address, then the initiating application-entity-invocation may pass the called-application-entity-title as a part of the application-PCI exchanged to establish the application-association.

11.1.2.3 Application-entity-invocations may exchange calling- and responding-application-entity-titles

for use in future communication. Such titles may be recognized by a recipient system as naming specific responding-application-entities. This is a matter for agreement between the communicating applications.

11.1.2.4 If required in establishing an application-association, the application-entity-invocations may exchange the following identifiers as part of the application-PCI exchanged to establish the application-association:

- application-process-invocation-identifier,
- application-entity-invocation-identifier,
- application-association-identifier.

11.1.3 Application Layer use of (N)-directory-functions

11.1.3.1 In the initiating system, on request from the application-process, the application-entity :

- a) uses IAF1 to derive the called-presentation-address (which is passed in a presentation-service request primitive) from the called-application-entity-title and from local information; and
- b) uses IAF3 to derive the calling-presentation-address (which is passed in a presentation-service request primitive) and the local PSAP through which the presentation-service request primitive is issued from the called-presentation-address and from local information.

NOTES

- 1 The choice of the PSAP at which this primitive is issued is a local matter. This choice has to be consistent with the calling-presentation-address.
- 2 At the Application Layer, the calling-(N)-address parameter does not apply.

11.1.3.2 The initiator application-directory-function, IAF1, yields a single presentation-address. Where a generic application-entity-title (e. g., application-entity-type-title) is used as the called-application-entity-title, resolution of this title to a single presentation-address is the responsibility of the recipient local system.

11.1.3.3 In the recipient system, on receipt of a presentation-service indication primitive, the application-entity uses RAF3 to derive the responding-presentation-address from the called-presentation-address and from local information.

11.2 Presentation Layer

11.2.1 In the initiating system, on receipt of a presentation-service request primitive, the presentation-entity:

a) uses IAF2 to derive a called-session-address (which is passed in a session-service request primitive) from the called-presentation-address and from local information;

b) uses IAF3 to derive the calling-session-address (which is passed in a session-service request primitive) and the local SSAP through which the session-service request primitive is issued from the called-session-address, the calling-presentation-address, and local information;

NOTE - The choice of the SSAP at which this primitive is issued is a local matter. This choice has to be consistent with the calling-session-address.

c) uses IPF1 to derive the called-presentation-selector, (which is sent in presentation-PAI), from the called-presentation-address; and

d) uses IPF2 to derive the calling-presentation-selector, (which is sent in presentation-PAI), from the calling-presentation-address.

11.2.2 In the recipient system, on receipt of a session-service indication primitive, the presentation-entity:

a) uses RAF1 to derive the called-presentation-address from the called-presentation-selector (which is received in presentation-PAI) the called-session-address, and from local information;

NOTE - Local information may be used to resolve a called-presentation-selector, that refers to a set of PSAPs to a single PSAP.

b) uses RAF2 to derive the calling-presentation-address from the calling-presentation-selector (which is received in presentation-PAI) and from the calling-session-address;

c) uses RAF3 to derive the responding-session-address from the called-session-address and from local information; and

d) uses RPF1 to derive the responding-presentation-selector (which is sent in presentation-PAI) from the responding-presentation-address (which is received in a presentation-service response primitive).

11.2.3 In the initiating system for connection-mode operation, on receipt of a session-service confirmation primitive, the presentation-entity uses IAF4 to derive the responding-presentation-address from the responding-presentation-selector (which is received in presentation-PAI) and from the responding-session-address received from the session-service confirmation primitive.

11.3 Session Layer

11.3.1 In the initiating system, on receipt of a session-service request primitive, the session-entity:

a) uses IAF2 to derive a called-transport-address (which is passed in a transport-service request primitive) from the called-session-address and from local information;

b) uses IAF3 to derive the calling-transport-address (which is passed in a transport-service request primitive) and the local TSAP through which the transport-service request primitive is issued from the called-transport-address, the calling-session-address and local information;

NOTE - The choice of the TSAP at which this primitive is issued is a local matter. This choice has to be consistent with the calling-transport-address.

c) uses IPF1 to derive the called-session-selector (which is sent in session-PAI) from the called-session-address; and

d) uses IPF2 to derive the calling-session-selector (which is sent in session-PAI) from the calling-session-address.

11.3.2 In the recipient system, on receipt of a transport-service indication primitive, the session-entity:

a) uses RAF1 to derive the called-session-address from the called-session-selector (which is received in session-PAI), the called-transport-address, and from local information;

NOTE - Local information may be used to resolve a called-session-selector that refers to a set of session-SAPs to a single session-SAP.

b) uses RAF2 to derive the calling-session-address from the calling-session-selector (which is received in session-PAI) and from the calling-transport-address;

c) uses RAF3 to derive the responding-transport-address from the called-transport-address and from local information; and

d) uses RPF1 to derive the responding-session-selector (which is sent in session-PAI) from the responding-session-address, which is received in a session-service response primitive.

11.3.3 In the initiating system for connection-mode operation, on receipt of a transport-service confirmation primitive, the session-entity uses IAF4 to derive the responding-session-address from the responding-session-selector (which is received in session-PAI) and from the responding-transport-address which is received in the transport-service confirmation primitive.

11.4 Transport Layer

11.4.1 In the initiating system, on receipt of a transport-service request primitive, the transport-entity:

- a) uses IAF2 to derive a called-network-address (which is passed in a network-service request primitive) from the called-transport-address and from local information;
- b) uses IAF3 to derive the calling-network-address (which is passed in a network-service request primitive) and the local NSAP through which the service primitive is issued from the called-network-address, the calling-transport-address and local information;

NOTE - The choice of the NSAP at which this primitive is issued is a local matter. This choice has to be consistent with the calling-network-address.

- c) uses IPF1 to derive the called-transport-selector (which is sent in transport-PAI) from the called-transport-address; and
- d) uses IPF2 to derive the calling-transport-selector (which is sent in transport-PAI) from the calling-transport-address.

NOTE - The initiator transport-directory-function IAF2 always yields a single network-address. Where the addressing information provided by the Application Title Directory Facility or by private arrangement specifies a list of network-addresses, resolution of these addresses to a single network-address is the responsibilities of local system management. (See 9.6.2).

11.4.2 In the recipient system, on receipt of a network-service indication primitive, the transport-entity:

- a) uses RAF1 to derive the called-transport-address from the called-transport-selector (which is received in a transport-PAI), the called-network-address, and from local information;

NOTE - Local information may be used to resolve a called-transport-selector that refers to a set of TSAPs to a single TSAP.

- b) uses RAF2 to derive the calling-transport-address from the calling-transport-selector (which is received in transport-PAI) and from the calling-network-address;
- c) uses RAF3 to derive the responding-network-address from the called-network-address and from local information; and
- d) uses RPF1 to derive the responding-transport-selector (which is sent in transport-PAI) from the responding-transport-address (which is received in a transport-service response primitive).

11.4.3 In the initiating system for connection-mode operation, on receipt of a network-service confirmation primitive, the transport-entity uses IAF4 to derive the responding-transport-address from the responding-transport-selector (which is received in transport-PAI) and from the responding-network-address received in the network-service confirmation primitive.

11.5 Network Layer

11.5.1 Introduction

11.5.1.1 The internal architecture of the Network Layer is complex. At each higher layer of the OSI architecture, an instance of communication (i.e. communication over a connection-mode, or a connectionless-mode data transmission) involves only a single pair of peer entities, which are located in end systems and communicate by means of a peer protocol. Within the Network Layer, however, communication often requires the participation of network-entities not only in the end systems but in intermediate systems as well. The necessary interactions between network-entities may be achieved through the operation of single protocols between pairs of network-entities, or may need more complicated combinations of layered protocols within the Network Layer.

11.5.1.2 The task of the network-directory-functions is, for any instance of communication, to use the called- and calling-network-addresses, together with other information, to determine which network-entities participate in the communication (this determination may be achieved by other methods).

11.5.2 Properties of network-addresses

11.5.2.1 Introduction

The properties are stated in terms of network-addresses. Since network-addresses refer to sets of NSAP-addresses, they extend naturally to a single NSAP-address.

11.5.2.2 Global non-ambiguity

At any instant of time, a network-address identifies just one set of NSAPs in the global domain. A set of NSAPs may have more than one network-address, i.e., synonyms may exist.

11.5.2.3 Global applicability

It is possible at any set of NSAPs to identify any other set of NSAPs, in any end system, by means of the network-address of the other set of NSAPs. If the other set of NSAPs has synonymous network-addresses, any of the synonyms will identify the set of NSAPs. In particular, therefore,