ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

INTER-DOMAIN INTERMEDIATE SYSTEMS ROUTEING

ECMA TR/50

December 1989

Free copies of this document are available from ECMA,
European Computer Manufacturers Association
114 Rue du Rhône - CH-1204 Geneva (Switzerland)

Phone: + 41 22 735 36 34 Fax: + 41 22 786 52 31

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

INTER-DOMAIN INTERMEDIATE SYSTEMS ROUTEING

ECMA TR/50

December 1989

BRIEF HISTORY

This Technical Report addresses the standardization of a routeing protocol to be used between Intermediate Systems that serve to bridge Routeing Domains.

The Technical Report has the following objectives:

i) to state the ECMA position with regard to Inter-Domain Routeing,

ii) to serve as a vehicle for influencing decisions in other standards arenas,

iii) to formalize the work carried out by ECMA.

The intention of this Technical Report is to develop a Routeing Protocol that is based on existing routing technology, is compatible with other Routeing Protocols under development (e.g. the Intra-Domain Protocol under consideration in ISO), and exhibits maximal commonality with these protocols.

Accepted as an ECMA Technical Report by the General Assembly of 14th December 1989.

TABLE OF CONTENTS

SECTION I

GENERAL

1. SCOPE AND FIELD OF APPLICATION

This Technical Report describes a protocol for the exchange of Network Layer routeing information between intermediate systems in different routeing domains. It also describes the relationships which need to be established by administrative procedures to provide the framework within which the protocol can operate.

The content of this Technical Report can be applied to any intermediate system which explicitly participates in inter-domain routeing.

However, this Technical Report also includes the concept of grouping together sets of intermediate systems (concerned with closely-related domains) into clusters, such that the procedures are applied to communication between these clusters, while permitting a freer locally-determined exchange of information within clusters.

2. CONFORMANCE

Systems claiming Conformance with the Protocol defined in this Technical Report shall:

i) implement the functions defined in Clause 11, and

ii) construct Protocol Data Units (PDUs) according to Clause 12.

3. REFERENCES

ECMA TR/37	Framework for OSI Management
ECMA TR/38	End System Routeing
ISO 7498	Information Processing Systems - Open Systems Interconnection - Basic Reference Model
ISO 7498/Add1	Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Addendum Covering Connectionless-mode Transmission
ISO 7498-3	Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 3: Naming and Addressing
ISO/DIS 7498-4	Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 4: OSI Management Framework
ISO 8208	Information Processing Systems - X.25 Packet Level Protocol for Data Terminal Equipment
ISO 8348	Information Processing Systems - Telecommunications and Information Exchange between Systems - Network Service Definition
ISO 8348/Add1	Information Processing Systems - Data Communications - Network Service Definition - Addendum 1: Connectionless-mode Transmission

ISO 8348/Add2	Information Processing Systems - Data Communications - Network Service Definition - Addendum 2: Network Layer Addressing
ISO 8473	Information Processing Systems - Data Communications - Protocol for Providing the Connectionless Mode Network Service
ISO 8648	Information Processing Systems - Data Communications - Internal Organization of the Network Layer
ISO 8802/(1-6)	Information Processing Systems - Local Area Networks
ISO 9542	Information Processing Systems - Data Communications - End System to Intermediate System Routeing Exchange Protocol for use in Conjunction with ISO 8473
ISO TR 9575	OSI Routeing Framework
ISO/DP 10030	End System Routeing Information Exchange Protocol for use in conjunction with ISO 8878

4. DEFINITIONS

This Technical Report makes use of the following concepts defined in ISO 7498, Basic Reference Model:

- Network Layer,
- Network Service Access Point (NSAP),
- Network Service Access Point Address (NSAP Address),
- Network Entity,
- Routeing,
- Network Protocol,
- Network Protocol Data Unit (NPDU).

4.1 Network Layer Architecture Definitions

This Technical Report makes use of the following concepts defined in ISO 8648, Internal Organization of the Network Layer:

- Subnetwork (SN),
- End System (ES),
- Intermediate System (IS),
- Subnetwork Service.

4.2 Network Layer Addressing Definitions

This Technical Report makes use of the following concepts defined in ISO 8348/Add2, Addendum to the Network Service Definition Covering Network Layer Addressing:

- Network Entity Title (NET),
- Subnetwork Address,
- Subnetwork Point of Attachment (SNPA).

4.3 Additional Definitions

For the purposes of this Technical Report the following additional definitions apply:

4.3.1 Acquiree

A potential neighbour which is able to respond to an offer of an agreement, but cannot make an offer by itself.

4.3.2 Acquirer

A potential neighbour which is able to offer an agreement.

4.3.3 Cluster

A set of inter-domain ISs, deployed in order to connect a number of routeing domains or a number of clusters; different clusters do not overlap.

4.3.4 Cluster Identifier (CID)

A globally unique identifier associated with each cluster. It is a variable length octet string with a maximum length of 20 octets.

4.3.5 ES-entry

A unit of information which is conveyed by and processed by the protocol, describing paths to a given set of NSAPs.

4.3.6 Forwarding Information Base (FIB)

A Forwarding Information Base is a part of the RIB which is used to determine the next IS/ES to which an RNPU should be sent. In this document, FIB actually consists of two parts, FIB1 and FIB2 (see 8.5.7). There are as many FIBs as there are TGTs supported.

4.3.7 Global Routeing Domain

The routeing domain which has the knowledge to route, at least part-way, to all other routeing domains in the OSI environment (OSIE).

4.3.8 Management Information Base (MIB)

The conceptual repository for Management Information.

4.3.9 The concept of Neighbour

X, Y are inter-domain ISs. Neighbouring inter-domain ISs are ISs that exchange information. Y is a neighbour of IS X, if Y is a potential neighbour of X, and both ISs have agreed on becoming each others' neighbour.

4.3.10 Partitioned Cluster

A cluster within which there exist IS pairs that are unable to communicate.Then the ISs can be organized in equivalence classes, to be called partitions in what follows. Two ISs are in the same partition if and only if they are joined by intra-cluster paths.

4.3.11 Partition Identifier (Partition ID)

An identifier of a partition within a cluster. Typically it is the NET of some IS in the class that is algorithmically unique (e.g. least NET or maximum NET). More abstractly, the partition id is a function f defined over all subsets of ISs of a cluster C such that, for any two non-overlapping subsets C_1 and C_2, $f(C_1)$ is not equal to $f(C_2)$.

4.3.12 The concept of Potential Neighbours

Two inter-domain ISs are potential neighbours if they are directly linked and have been empowered to use the links between them.

4.3.13 Routed Network Protocol Unit (RNPU)

This is a protocol element to which routeing decisions are applied. When providing CLNS, each ISO 8473 PDU is an RNPU. When providing CONS, routeing decisions are made only on connection requests.

4.3.14 Routeing Domain

A set of ESs, ISs, and subnetworks. All systems within a Routeing Domain operate according to the same routeing procedures. Exchange of information between ISs within the same Routeing Domain is considered as intra-domain routeing, all others as inter-domain routeing. A formal definition of a Routeing Domain is given in Appendix D (D.2.1).

4.3.15 Routeing Information Base (RIB)

That part of the MIB which is concerned with routeing.

4.3.16 Routeing Metric

A unit of measure describing the relative quality of a given path or path segment. Commonly used metrics include hop counts, cost, delay, and congestion indicators.

4.3.17 Table Generation Tag (TGT)

An index that permits differentiation between FIBs to support multiple types of service (see 8.5.5).

SECTION II

GENERAL PRINCIPLES

5. INTRODUCTION

In the OSI environment (OSIE), the possibility exists for any end system (ES) to communicate with any other ES. The physical path, or paths, over which this communication takes place may:

- include multiple intermediate systems (ISs),
- include multiple subnetwork types, and
- traverse multiple organizations.

Furthermore, the communication may follow a different path for any given instance.

Within the Network Layer, the «Internal Organization of the Network Layer» (ISO 8648) identifies two functions, routeing and relaying, as being central to the ability for ESs to communicate through an arbitrary concatenation of subnetworks and ISs.

Part of the overall function of routeing and relaying is to allow ESs and ISs to find an appropriate path between two ESs.

Routeing is primarily concerned with path selection, potentially through multiple subnetworks and ISs, so that ESs may communicate.

The requirements for OSI Network Layer Routeing may be considered under two, largely separate, headings:

- those aspects of Network Layer Routeing concerned with communication between ESs and ISs on the same subnetwork, and
- those aspects that are concerned with communication among the ISs that connect multiple subnetworks.

The aspects concerning ESs, and ISs, on the same subnetwork are described in Technical Report ECMA TR/38 «End System Routeing».

The aspects concerning ISs fall under two headings:

- communication among ISs belonging to the same routeing domain, and
- communication among ISs belonging to different routeing domains.

This Technical Report describes a method by which IS Routeing among ISs belonging to different routeing domains may be employed to effect the OSI Routeing functions. Intra-domain IS interactions are considered to be independent and therefore are not part of this Technical Report.

Note 1:

Primarily this Technical Report is concerned with the needs of ECMA. Since the solution has, necessarily, to fit also into ISO requirements, no distinction has been made.

6. OVERALL PROBLEM DESCRIPTION

6.1 Functional Service Requirements

The OSI Routeing Framework document (ISO/TR 9575) decomposes the routeing problem into ES-IS and IS-IS operations. Furthermore, routeing between ISs is decomposed into three components:

i) Routeing within a routeing domain,

ii) Intra-administrative, inter-domain routeing, and

iii) Inter-administrative routeing.

In component (i), information or summary information is completely shared and all ISs run a single algorithm which attempts to construct optimal paths.

In component (ii), it is important to be able to raise firewalls between the routeing domains so that they can be protected from each other (see 6.2.1). In other words, it is undesirable that there be interactions between routeing domain algorithms or between a routeing domain algorithm and the intra-administrative routeing algorithm.

Finally, in component (iii), the issues of trust, security and policy-based routeing are raised (see 6.3). In this case, information is not always freely shared and the driving concern is to meet externally imposed restrictions, such as legal and contractual obligations, and administrative policies, rather than optimal routeing.

The requirements for inter-domain routeing (IDR) are a subset of the requirements for inter-administrative routeing. Moreover, it appears that whatever differences may exist between these two instances of routeing, they are differences of degree only. Therefore differentiation should not be made between inter-administrative and inter-domain routeing, and both of these instances of routeing should be addressed through the same protocol. The differentiating feature of this protocol is that it will primarily accommodate administrative, legal, contractual, and autonomy preserving concerns. The issues of routeing efficiency will never override the concerns just enumerated. Therefore this Technical Report addresses components (ii) and (iii) in a single protocol.

6.2 General Requirements and Performance Goals

It is to be taken as self-evident that any good protocol must provide service within the bounds of the actually existing or foreseeable technology without making unjustified assumptions or imposing arbitrary restrictions. The purpose of this section of the Technical Report is to enumerate a set of performance requirements and goals that IDR should satisfy, if it is to result in a feasible and useful protocol.

6.2.1 Architectural Issues

The architectural issues to be addressed are the choice of a model that satisfies the inter-domain routeing needs without unreasonable restrictions or unreasonable resource demands. In more detail:

i) There should be no unjustifiable topological restrictions.

ii) IDR should not impose unjustified requirements and restrictions on the type of intra-domain routeing protocol(s) to be used. Therefore, there should be a maximum degree of decoupling between the inter- and intra-domain routeing protocols and between the inter- and intra-domain routeing information databases.

iii) IDR should not result in a protocol that places unjustified and discriminatory requirements on the type of hardware and software to be used.

iv) The inter-domain routeing model should address the issue of how inter-domain links are realized and used for protocol and routeing purposes.

v) IDR should enforce firewalls so that:

(a) Routeing problems within a routeing domain should not affect routeing within other domains;

(b) Routeing problems within a routeing domain should not affect inter-domain routeing, unless the inter-domain links are affected;

(c) Inter-domain routeing shall not adversely affect intra-domain routeing.

vi) IDR should facilitate information compression/abstraction and information hiding (whereas information hiding should not be mandatory, routeing domains and routeing administrations may not wish to reveal to the world some of their NSAPs or other internal information).

vii) Whenever the addresses permit, IDR should exploit the possibilities that arise for compression/abstraction of the address information. But it will neither legislate nor depend on unjustified assumptions concerning the address structure.

6.2.2 Environmental Constraints

The routeing functions to be defined must be designed to operate without regard to any specific underlying technology or transmission medium, to the extent that they do not rely upon any technology-specific service for their correct operation. These functions must also be designed to operate correctly irrespective of the geographic distribution of ESs and ISs which comprise the global routeing domain (i.e. they are not topology-dependent).

The global OSIE in which ES data is to be transferred is assumed to consist of a very large number of ESs ($> 10^7$) which, in the most general situation, may be logically interconnected by means of paths consisting of concatenated ISs. The total number of ISs is assumed to be one to two orders of magnitude less than the number of ESs, but very large as well. Any routeing scheme adopted for OSI must be capable of near-infinite scaling.

Global routeing must by necessity be able to operate correctly under the distributed control of multiple organizations. Furthermore, the control of routeing within a single organization may be distributed, for example for reasons of efficiency, economy, or performance.

6.2.3 Routeing Performance Requirements

IDR must treat policy issues as being of higher priority than the problem of optimal route selection because:

i) Information abstraction and hiding are antithetical to the need for detailed and full information that optimal route selection necessitates.

ii) All other things being equal, any single link failure is bound to affect intra-domain routeing much more severely than it affects inter-domain routeing. Therefore, IDR routeing should de-emphasize the intra-domain concern for selecting optimal routes and emphasize static routeing features to the maximum degree possible.

iii) Nevertheless, route selection does not exhaust the performance issues. It is extremely important that the protocol developed does not require inordinate amounts of scarce resources, such as CPU, bandwidth, or memory, which are neither available now nor are likely to be in the foreseeable future.

iv) Furthermore, it is desirable to have a protocol that accommodates as many as possible of the following service features:

- Multipathing / loadsplitting.
- Cost minimization (both the cost of running the protocol and of routeing data traffic).
- It is desirable that the inter-domain routeing protocol be adaptative and converge rapidly; it should construct routes that tend to remain stable over long periods of time (no oscillations).
- It should provide adequate service in the presence of a very large number of routeing domains.

6.2.4 Compatibility and Migration Requirements

It is desirable that:

i) The inter-domain routeing protocol and the underlying model should be either compatible with the existing network protocols or require a limited amount of minor revisions and additional options.

ii) The inter-domain routeing protocol should be such that only cataclysmic changes in the intra-domain routeing protocol will necessitate inter-domain routeing protocol revisions.

6.3 On Policy-based Routeing

6.3.1 Policy-based Routeing

While it is important that the inter-domain routeing protocol should accommodate as many as possible of the preceding requirements, the fact is that the *raison d'être* of this protocol is to address the issues of providing firewalls as well as security and trust. The purpose of this section of the Technical Report is to delineate the routeing problems that arise when this type of administrative concern must be met.

It is necessary that an inter-domain routeing protocol subordinates optimal routeing to administrative concerns. Nevertheless, a routeing protocol should not be directly concerned with administrative concerns *per se*. The protocol should be based on a model such that any global policy can be decomposed into a set of local policies, i.e. a set of restrictions on ISs and on links. The protocol itself should be capable of interpreting and of implementing these local restrictions.

The effect of any administrative policy is to declare some paths legal and some illegal. That is, the intent of any given policy is to constrain routeing to use (or not use) a subset of the available paths. Therefore, it may be considered that at any time there is:

i) a set U consisting of all conceivable paths; a path is a sequence of links such that:

   (a) the first link starts at an ES and the last link ends at an ES;

   (b) the endpoint of any link but the last is the same as the starting point of the next link;

   Note 2:

   It is assumed that there might be multiple distinguishable links between any two ISs or between an IS and an ES.

ii) a set P consisting of all legal paths;

iii) a set F consisting of all feasible paths; a path is feasible if at time *t* all links in the path are operational.

Note 3:

U changes as equipment (nodes and links) is deployed or removed. Sets P and F are subsets of U. The first can change as U changes and/or when the policies to implement change. F changes as U changes and whenever the state of a node and/or a link changes. P was defined independently of F so that it would be independent of the operational state of the equipment.

From the available examples (see 6.3.4), it appears that there are cases in which a natural requirement for policy-based routeing is to be sensitive to the sender's NSAP and to the path that the RNPU has already traversed on its trek towards its destination. Moreover, sensitivity to the path already traversed is the most general form of policy-based routeing. Indeed, at any given IS, the forwarding function can operate only upon the knowledge the IS already has or can readily obtain. In other words:

i) its present and past knowledge of the network status,

ii) whatever information can be inferred from the header, and

iii) whatever information can be inferred from the underlying service, such as forwarding and receiving SNPAs.

It therefore follows that there are two matters that must be addressed:

i) what and how much information about the network must be kept in the ISs, and

ii) what information about the history of the packet is needed in order to route it along a legal path.

Given that the path traversed encompasses all the history of the RNPU, path discrimination is the most general request. Hence, it remains to address what information is to be kept in the ISs.

The following methods of defining P lead to inherently intractable routeing problems and should be explicitly disallowed:

i) P depends on future events (for example, P is legal at a time t_1 provided that at time t_2 ($t_2 > t_1$) some type of event will occur).

ii) P contains self-referential statements of the type «Path P_1 is in P if and only if P_2 is not»; such types of statements (declaring P_1 legal as soon as someone makes P_2 illegal) may introduce unending oscillations in P.

iii) P contains references to F.

The last point is significant since it is a natural requirement to stipulate that some path P_2 be used if and only if some primary path P_1 is not feasible. But, on the other hand, not all routeing schemes allow ISs to know at any given time what F is. Moreover, references to feasible paths and, by extension, to their performance characteristics, could greatly complicate the determination of what is legal. But there are other ways of addressing this problem, for example by assigning to P_2 such metrics that P_1, when feasible, will be preferred to P_2. In what follows it is assumed that P is indeed independent of F.

P should remain unchanged over long periods of time. Indeed, P should depend to the maximum degree possible on the policy restrictions. Whereas equipment additions/removal may add/remove legal paths, a path that is legal/illegal should remain so while the set of policies remains unchanged and the equipment that realizes this path remains in place.

Actually, it is desirable to group equipment into equivalence classes such that the legality of a path is independent of the particular element(s) of each class that participates in the path. If such equivalence classes have been defined then whenever new equipment is introduced it should be declared in which equivalence class it belongs. As an example of such equivalence classes, all ISs within some routeing domain may be seen as equivalent by all other entities. The addition of a new IS to a domain will create some new legal paths but the new paths are seen as equivalent to existing legal paths.

It follows that if P is so defined then the ISs should collect and propagate information about those feasible paths that happen to be legal and, upon receipt of an RNPU, route along such a path.

6.3.2 Tools for Policy-based Routeing

The model should be such that the policies be decomposable into local policy statements. Such statements are in a one-to-one correspondence with the tools used to implement them. These tools

i) group sets of ISs and announce policies that bind all ISs in the group (such as: the ISs in group X will relay only those RNPUs whose origin or destination NSAP matches some mask M).

ii) put filters on links connecting groups such as those just described. The filters can limit the type of information and of RNPUs that can cross the link.

iii) impose a structure on these groups so that administrative policies can be decomposed into a set of local policies and that the global effects of local policies can be computed.

iv) exploit whatever information is encoded in the RNPU Protocol Control Information (RNPU PCI).

Furthermore, there is another set of tools, namely address administration and equipment deployment. Clearly, a fixed amount of equipment can support only a limited number of policies. Often, the effect of multiple policies is to request that some piece of equipment must play multiple logical roles. In most instances this is achieved through a total separation of the logical roles (e.g. multiple ISs may be thought to co-reside in a single box, multiple logical links may be realized over the same physical link, and so on).

In theory, these tools are sufficient to implement any set of legal paths. It suffices to put in each box X as many logical entities $X_{(P)}$ as there are legal paths that cross X. Unfortunately, such a solution works only on paper given the number of paths that traverse a typical X. Nevertheless, this shows that the tools to implement any given policy exist. Therefore, any model that can accommodate a good number of policies at a reasonable cost is a workable model for policy-based routeing.

Finally, it is noted that restrictions on ISs and links are insufficient for the purpose of path discrimination. Therefore, a workable model needs some structure and the ability to split links and ISs into multiple logical entities.

6.3.3 Specific Requirements for Policy-based Routeing

In view of what precedes, a policy is any set of rules that provides an effective and efficient way for determining if a path is legal according to this policy or not. This path must either be realizable in the universe of existing equipment or in some well-described extension of the same (e.g. after new equipment is deployed).

A model for policy-based routeing makes it possible to see whether a policy can be implemented with minimal changes, if any, to the existing structures or whether new structures are needed.

Therefore, a model is needed in which the addition/removal of administrative policies can be accomplished with minimal impact on the given structures. Moreover, this impact should be easily assessable and not result in an open-ended sequence of adjustments. That is:

i) global policies can be decomposed into a set of local agreements;

ii) local agreements can be composed so that one can see the global policies that emerge.

The first requirement is needed in order to implement new policies; the second so that the new policies will not have unintended side effects.

6.3.4 Examples of Policy-based Routeing

The following examples show some policies and the routeing problems that these policies generate.

Example 1

A routeing domain R is under control of a corporation C. R is willing to route all RNPUs such that either the source or the destination are recognized to be affiliated with C. This calls for source-sensitive routeing. If some other routeing domain R_1 routes on the basis of this property of R, then the next routeing domain, R_2, must also be cognizant of this property; otherwise, inter-domain loops may result.

Example 2

Routeing domains A, B, C, D, U, V, W, and X are linked as shown in Figure 1. Routeing domain A does not trust routeing domain C while routeing domain X does. Therefore, when routeing to D, B must distinguish between RNPUs that originate in A and RNPUs that originate in X. It may also be necessary that other routeing domains, such as U, know of this routeing requirement (otherwise loops may result).

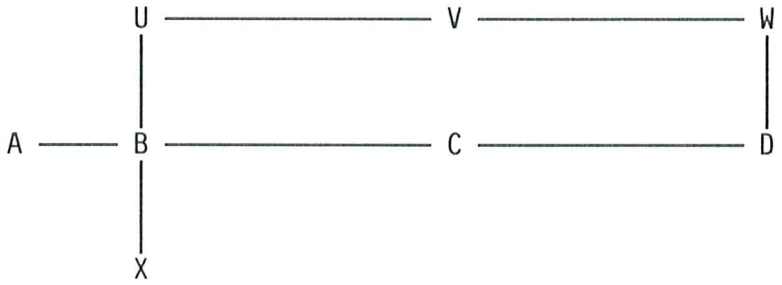


Figure 1 : Need for Source Discrimination

Example 3

Routeing domain W, see Figure 2, is perfectly willing to route traffic between routeing domains A and B. But, it may be barred from handling traffic that transited through routeing domain Y either for reasons of its own or because of contractual agreements with other routeing domains. Similarly, W may only wish to handle traffic between A and B if it transited through some other routeing domain X. In either case, the desired policy cannot be implemented unless IDR is capable of path discrimination.

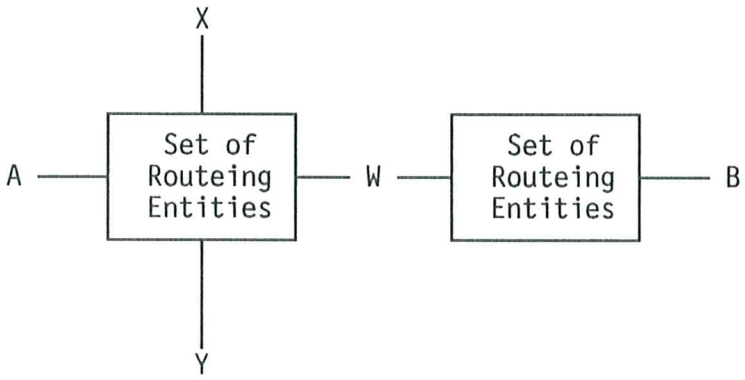


Figure 2 : Need for Path Discrimination

Example 4

Two overlapping routeing domains (common ISs and ESs) require that internal routes be always preferred to external routes. In this situation, this requirement is tantamount to asking that routeing be sensitive to the path traversed.

6.4 Implementation Assumptions

It is assumed that each IS that participates in this protocol is endowed, by means that lie outside this protocol, with the following knowledge:

i) the identity of the group it belongs to;

ii) a list of links, and the identity of the neighbour at the end of each link (including the identity of the group to which the neighbour belongs);

iii) a complete list of what information and RNPUs are supposed to cross the link in either direction and the means of establishing the link (via the protocol and at the neighbour acquisition phase) if its neighbour has a compatible view of the link.

It is also assumed that there is a set of ISs that are attached to routeing domains and that are capable of advertising the reachability of the attached routeing domains as if the domains in question consisted of a single ES.

This protocol should exploit, if and when present, whatever new information may be included in the RNPU PCI.

In addition it is assumed that:

i) routeing policies will tend to mirror administrative boundaries; and

ii) administrative boundaries will exhibit the topology of overlapping trees/hierarchies (i.e. partial ordering) with some equipment being in multiple administrations.

6.5 Conclusions

It appears, then, that the model that supports the policy-based routeing protocol should establish a partial order among the several groups of ISs which implement policy, with the routeing domains represented as leaves (minimal elements). In such a model the basic properties of hierarchical routeing can be preserved (see Clause 8).

7. CATEGORIES OF ROUTEING

Routeing within the OSI environment proceeds from two basic principles:

i) The aspects of routeing that are concerned with communication between ESs and ISs on the same subnetwork are, to a great extent, separable from the aspects that are concerned with communication among the ISs that connect multiple subnetworks (an ES reachable through more than one routeing domain may be an exception to this).

ii) Establishing communication among ISs to connect multiple subnetworks presents both technical and administrative challenges of a global nature. The technical issues have to do with establishing global connectivity and performing global routeing functions; the administrative issues have to do with controlling the way in which groups of systems managed by different administrative authorities are permitted to communicate.

These two principles lead to a decomposition of the global routeing function. The first principle establishes an initial distinction between local ES-IS operations and global IS-IS operations. The second principle establishes a further distinction between IS-IS operations within the purview of a single routeing domain (intra-domain routeing), and IS-IS operations that span routeing domains (inter-domain routeing). For further information about the structure of global OSI routeing, see Appendix D. This Technical Report concerns inter-domain routeing.

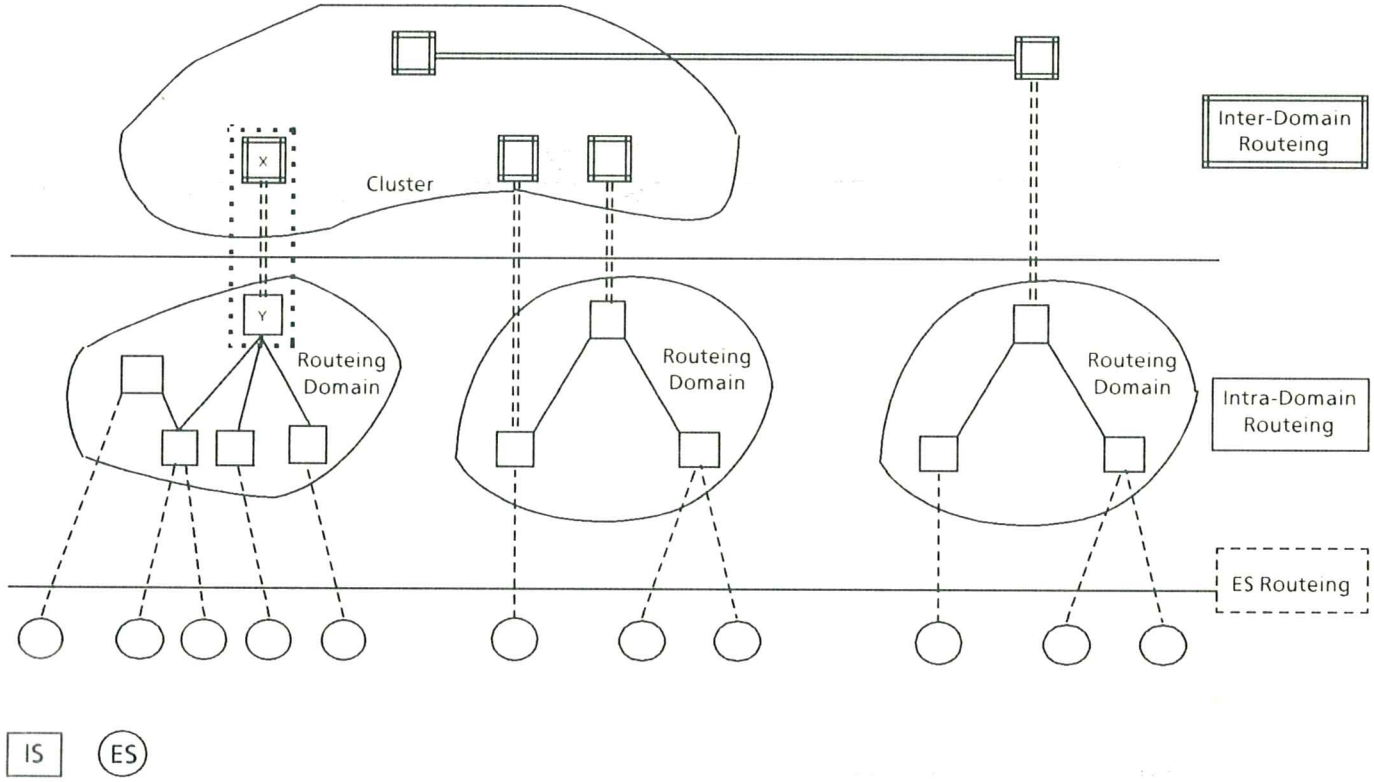


Figure 3 : Logical Associations according to Routeing Protocols of different Categories

Note 4:

Inter-domain routeing is conceptually the same as intra-domain routeing. ESs in intra-domain routeing correspond to routeing domains in inter-domain routeing, whereas the clusters (see 7.2) in inter-domain routeing correspond to ISs in intra-domain routeing.
Nevertheless, inter-domain routeing is expected to meet demands different from those of intra-domain routeing. Therefore, it is likely that the entities participating in inter-domain routeing will be organized on different principles than those entities that participate in intra-domain routeing.
For inter-domain routeing, the nature of the relationships will restrict the type and detail of routeing information available. More stringent procedures for authentication and propagation of routeing information may also be needed.

Note 5:

An IS operating as an inter-domain IS is a separate logical entity. It may reside within the same physical entity as the intra-domain IS (as shown by the dotted line enclosing ISs X and Y in Figure 3). If so, then the intra-domain to inter-domain exchanges can be implicit. If not, then there is an explicit exchange of information between the two ISs.

Note 6:

There may be groups of related ISs (clusters) that from the point of view of inter-domain routeing may be treated as equivalent. The routeing information exchange between these ISs (intra-cluster routeing) is outside the scope of this Technical Report. Therefore, in Figure 3 no lines are shown between ISs in the same cluster.

7.1 Routeing Domains

The global OSIE will of necessity be composed of multiple routeing domains which are under the responsibilities of different administrations.

A routeing domain is a set of ISs bound by a common routeing procedure, namely:

- they use the same set of routeing metrics,
- they use compatible metric measurement techniques,
- they use the same information distribution protocol, and
- they use the same path computation algorithm.

Note 7:

Two routeing domains may use the same routeing procedure, and consist of identical or overlapping sets of ISs.

7.2 Clustering Inter-domain Intermediate Systems

Inter-domain ISs are ISs that run the inter-domain protocol defined in this Technical Report. They are charged with:

- connecting routeing domains, and
- implementing issues of trust and security between these domains.

It is assumed that each inter-domain IS may at any time find out if a routeing domain is directly reachable by it and, if so, which parts are indeed reachable. Furthermore, a list of other measurements may also be mandated, such as: biggest distance between an IS and a reachable ES (expressed in number of hops, total delay, etc.), quality of service (QoS) supported by the routeing domain, as well as any other relevant factors. Since firewalls and/or tariffs are to be placed at this level, it should be possible to introduce new ISs as needed without secondary effects to other parties. It is assumed that the inter-domain ISs have a hierarchical structure, and that only a subset of the existing paths is used.

However, there may be groups of related inter-domain ISs between which information can be freely exchanged without restrictions, and which can be treated as equivalent from the point of view of the other inter-domain ISs. Such groups are referred to as *clusters*; further details about clusters are given in Clause 8 below.

Note 8:

The structures in inter-domain routeing are designed to permit:

- *viewing any routeing domain as member of as many administrations as desired (typically one),*
- *viewing any administrative unit as a sub-unit of as many administrative units as desired (again, most of the time, one), and*
- *placing of tariff structures and protection interfaces between clusters.*

8. PROTOCOL OVERVIEW

The purpose of this Clause is to provide a general description of the environment in which this protocol operates and of the underlying model upon which the protocol is based.

8.1 Inter-domain Protocol Exchanges

As shown in Figure 3, an inter-domain IS is potentially concerned with two types of protocol exchange. These are the exchange of:

- information with other inter-domain ISs, and
- information with the routeing domain(s) to which it is related.

The exchange of information between an inter-domain IS (cluster) and routeing domain(s) to which the inter-domain IS (cluster) is related is the equivalent of an ES-IS protocol in which most of the dynamic considerations are removed. This subject is covered in Clause 9.

8.2 Information provided

The information exchanges listed above convey two types of information between the network entities which support their operation:

- (logical) configuration information, and
- routeing information.

The information relating to a set of NSAPs is processed in the form of ES-entries which are described in 10.5.

8.3 Types of Subnetworks

This protocol is applicable for use in environments in which adjoining inter-domain ISs (clusters) communicate by the following means:

- point-to-point real subnetworks,
- broadcast real subnetworks, and
- general topology real subnetworks.

8.4 Types of Network Service

When providing the CLNS, a separate forwarding decision is made for each ISO 8473 PDU. When providing the CONS, routeing decisions are made only during connection establishment, and subsequent PDUs follow the route derived for the connection. This document uses the term routed network protocol unit (RNPU) to refer to those protocol elements on which routeing decisions are made, regardless of the mode of service.

8.5 General Model and Protocol Description

8.5.1 General Tasks to be performed by an Inter-domain IS

An inter-domain IS is expected to construct a Forwarding Information Base (FIB) that allows RNPUs to be routed. To accomplish this, an inter-domain IS must be able to perform the following tasks:

Neighbour acquisition :

An inter-domain IS must be able to acquire and maintain inter-domain IS neighbours. It may also participate in an inter-domain/intra-domain interface (see Clause 9).

Routeing information exchanges :

An inter-domain IS must be capable of supporting routeing information exchanges. These exchanges, as a rule, modify the RIB and, indirectly, the FIB. Modifications to either information base will, as a rule, trigger further information exchanges.

Raw data creation :

Inter-domain ISs are expected to assess their local environment and to create the raw data that trigger the RIB updates. Therefore, an inter-domain IS must:

- monitor the operational status of its links and of the ISs at the other end of said links; periodically, and when the operational status of such a link changes, the IS will modify its RIB and will so notify its neighbours;
- if it participates in an inter-domain / intra-domain interface, monitor the accessibility of the routeing domain at the other end; periodically, or when the accessibility changes, its RIB will be modified and the neighbours will be notified of this change.

8.5.2 Architectural Model for Inter-domain Routeing

The protocol inherent in this Technical Report assumes the following architectural and topological model:

i) The inter-domain ISs are grouped in clusters in such a way that each IS resides, as a logical entity, in a single cluster.

ii) The clusters form a partially ordered set (i.e. for any pair of clusters C_1 and C_2, either $C_1 < C_2$, or $C_2 < C_1$, or $C_1 = C_2$, or C_1 and C_2 are not comparable). The clusters have unique identifiers which are administered by a Registration Authority which also registers the relationship between the clusters. Existing relations can be repudiated and new ones instigated, provided that the new relations do not introduce inconsistencies in the partial order.

iii) The cluster identifiers exist purely to enable clusters to be uniquely distinguishable. It is not intended that the relationship between two clusters can be determined algorithmically from the values of their identifiers.

iv) Links between ISs in the same cluster are tagged as intra-cluster *(i)* links. There are no limitations for information exchange between ISs in the same cluster.

v) Links between clusters may be of two types:

- up/down-links, or
- jump-links.

Up/down-links only exist between comparable clusters. Jump-links (*(j)*-links) may exist by agreement between any two clusters.

If IS X is in C_1, IS Y is in C_2, and $C_1 < C_2$, then, with the exception of agreed jump-links, the links between X and Y are tagged as being up *(u)* in the direction from X to Y, and down *(d)* in the other direction.

vi) Inter-cluster links may have filters that limit the information exchanges and routeing along these links. Moreover, neither information nor PDUs will move in the *(u)*-direction or across a *(j)*-link after having crossed a *(j)*-link or a link in the *(d)*-direction.

Note 9:

This allows to route in such a way that there are no loops, unless they are intra-cluster-loops, and questions of trust/mistrust can be implemented by the relative position of the clusters and/or the bilateral agreements that may exist between clusters. The ordering of the clusters in fact implies a correspondence between each inter-domain IS and the NSAPs that are beneath it (reachable via (d.i) paths). An IS is not expected to forward a PDU unless either the sending or the receiving NSAP, or both, are beneath it.*

vii) Each routeing domain that participates in IDR will have links terminating in one or more clusters. Normally there will be only one exit cluster, but if multiple clusters exist then the decision of which to use for any given purpose is a local matter internal to the routeing domain.

viii) Inter-domain ISs will have access to a static part of the RIB that describes their cluster's view of the local environment as well as the IS's view of its local environment.

Note 10:

It is expected that as a rule the inter-cluster links will inherit their tag from the partial order. That is, a link that joins two non-comparable clusters will be tagged a jump and all others will be up in the direction from the lower to the higher cluster. But it is entirely possible to have jump links between comparable clusters. The usage of these links will be available only to the clusters in question and to their descendants.

8.5.3 General Protocol and Routeing Description

The protocol that constructs inter-domain routes is based on the following assumptions:

- that all ISs within a cluster will have the same information; for example, this could be achieved by a link state algorithm as described below.

- that externally each cluster will be seen as a single entity and that the mechanism for choosing the sequence of clusters to be traversed will be a distance vector scheme.
- that each inter-domain IS is provided, by means external to this protocol, with a static information base that fully describes the local environment of the IS and the allowed information exchanges between the IS in question and its neighbours.
- that each inter-domain IS maintains RIBs and FIBs (see Note 12); the entries of a FIB represent sets of NSAPs that may overlap.
- that each RNPU carries an RNPU-restriction tag that shows if jump-links or inter-cluster links in the downward direction have been crossed; further routeing of this RNPU will be restricted according to the value of this tag.

Note 11 provides additional information on these notions. This Technical Report does not mandate the choice of the intra-cluster algorithm to be used as long as the algorithm supports the functionality the inter-cluster algorithm needs (such as rapid convergence and partition identification). A well known example of such an algorithm is a link state algorithm. All further references to a link state algorithm are a short hand for "any intra-domain routeing protocol that provides the necessary functionality".

Note 11:

There are a number of considerations that lead to the choice of a link-state algorithm within a cluster but a distance vector algorithm between clusters. Rather briefly stated, these reasons are the following:

- *Link state is used within a cluster so as to be able to model the cluster as a single entity (and consequently a rapidly converging scheme is preferred), and to easily identify cluster partitions.*
- *The inter-cluster scheme uses distance vectors for the following reasons:*
  - *The inter-cluster path restrictions are such that, when the cluster topology is carefully maintained, there are no loops. Moreover, the intended topological structure is such that most equipment failures will only induce local perturbations.*
  - *A distance vector scheme allows clusters to provide service without necessarily revealing internal structure details which they may want to conceal (from all or some). By internal, read in this context all clusters beneath the cluster in question and the links between these clusters.*
  - *One of the driving concerns in the development of this protocol is to make it as independent as possible of the intra-domain routeing features. In this way, the inter-domain protocol would allow the gradual and graceful introduction of new intra-domain routeing schemes. However, it is also important to be able to introduce gradual changes to the inter-domain scheme. A scheme that uses link state information to derive pre-computed FIBs will not work well unless each cluster knows how a set of other clusters (i.e. those that lie on paths that join this cluster and routeing domains) construct their FIBs. A distance vector scheme does not impose this restriction. Since the path selection depends on the results of another cluster's computation, it does not need to know how this computation was per-*

formed. Thus, the proposed scheme provides a degree of flexibility that facilitates the introduction of new inter-domain routeing technologies.

- *A distance vector scheme does not impose externally-derived lower bounds on the amount of data that a cluster must maintain as the price of admission to inter-domain routeing; link state schemes, by contrast, do.*

8.5.4 Static Information of an IS

Each inter-domain IS X is assumed to have a static snapshot of its environment and of the environment of its cluster. This snapshot will not be altered by the operation of the inter-domain routeing protocol and will contain the following information:

- Local information about the cluster:
  - the identity of X's cluster, C;
  - the identities of all clusters that adjoin C;
  - the relation of C and the adjoining clusters;
  - for each adjoining cluster D, the costs that C assigns to traversing D. There is a separate cost for each supported TGT (see 8.5.5).
- Local information about X itself:
  - the NETs of all potential neighbours to X;
  - a complete list of the links (i.e. SNPA pairs) through which these neighbours can be reached, as well as the tags of these links (up/down, jump, or internal) and of the negotiated agreement that specifies what type of information and data PDUs can cross this link;
  - for each link whether to act as an Acquirer or Acquiree;
  - for each link the protocol parameters (AAC, AAT, DUC, DUT, HLT, KAT, RRC, RRT and WT, as defined in Clause 11).
- Local Parameters

  The values applicable to the underlying subnetwork as identified in 13.1.3 and 13.2.3.

8.5.5 Routeing Information Base (RIB)

All ISs within a cluster C are assumed to have RIBs containing the following information items:

- a complete list of all actual neighbours, active links, and such-like;
- the operational status, and all other pertinent information, on all links that either start or end in C;
- whatever information was offered by ISs in other clusters and retained by the border nodes of C;
- for each neighbour in a different cluster, whatever routeing information the IS has most recently offered to the neighbour in question.

In general, all information exchanged between clusters carries the following:

Table Generation Tag (TGT):

Indicates which FIB in an IS may use the information in question. An IS may construct multiple FIBs, one for each TGT it supports.

Functions such as F_1 and F_2 (Appendix E) are assumed to use a TGT value as an index to the database they wish to consult.

The following TGT values are currently assigned:

0 CO-mode, regardless of QoS values
1 CL-mode, regardless of QoS values.

Note 12:

The purpose of maintaining multiple FIBs is to support multiple routeing environments (e.g., Connection-Oriented Network Services and Connection-Less Network Services).

TGT values could also be used to accommodate options such as QoS. While multiple QoS metrics may be conveyed in the DU PDU, it is by no means certain that a path that accommodates a given QoS will be retained (and propagated) rather than one that does not. If it is deemed necessary that paths that satisfy some QoS be retained, this can be done by globally mapping this QoS onto a TGT value.

Restriction Tag :

The restriction tag determines both, how information is distributed and how it can be used in the forwarding decision. A cluster may freely share all information with the restriction tag «off» while information with the restriction tag «on» can be passed only in the *(d)*-direction. Correspondingly, RNPUs that have only travelled in the *(u)*-direction may use routes derived from all available information whose restriction tag is «off».

The restriction tag associated with information pertaining to a set of NSAPs is «off» if and only if it describes paths whereby all these NSAPs are reached by travelling only in the *(d)*-direction. Otherwise the restriction tag is «on».

Synthetization Index :

Reflects the levels of synthesizing in the information. Direct observations have synthetization index 0. When a metric is assigned as a result of a computation, the synthetization index of the result is 1 plus the maximum of the synthetization indices of the metrics used. Additionally, the Restriction tag is «off» if and only if all input information had the Restriction tag «off».

Note 13:

As the value of the synthetization index increases, the reliability of the information associated with it decreases. The value of the synthetization index is also a measure of the maximum number of clusters the information traversed.

8.5.6 Partition Identification

Each IS X in a cluster C can compute the numerically smallest NET that it can reach by crossing internal links only. This NET identifies the equivalence class of X. When X passes routeing information to neighbours in other clusters, it identifies not only itself, but also its cluster and its equivalence

class. In this way, the clusters adjoining C can assess if C is partitioned (assuming that they receive information from two or more ISs in C, and that these ISs do not all lie in the same partition).

8.5.7 FIB and Inter-Cluster Information Exchanges

As pointed out above, the routeing scheme of this protocol is a distance vector algorithm. The only link information that crosses cluster boundaries concerns links that join the clusters in question. The routeing scheme of this protocol assumes that the information that crosses inter-cluster links is of the form of ES-entries. In other terms, if ISs X and Y are in adjoining clusters C and D, Y may inform X that Y can reach a set of NSAPs M and that the metrics that apply to this set is m. Y will also indicate its equivalence class in D, and the metrics will be computed in such a way that all ISs in Y's equivalence class will report the same metrics for M.

Once information of this type is collected, the ISs in C will proceed as follows:

- Whenever inconsistent information emerges from the same partition, the worst possible inference will be drawn from the available data.
- The metrics will be modified by C's perception of the cost of traversing the clusters in question. Subsequently, C will choose for each set M, and for each value of the restriction tag, those entries that have the best metrics. A rounding factor may apply, in which case all entries within a range are retained and are used in subsequent computations as if they had the same metrics, equal to the worst metrics retained.
- C will extract its FIBs from the link state information and the ES-entries. FIB1 (TGT), is derived exclusively from information that has the restriction tag «off»; FIB2 (TGT), is derived from all the available information.
- C will use its own FIBs in order to construct its own ES-entries that it will send to its neighbours. The metrics reported for each such entry will not reflect the path portions in C and, therefore, the reported metrics will be independent of the reporting IS.

8.5.8 Minimal Set of Requirements

The only mandatory functions of an inter-domain IS are those that are explicitly needed for neighbour acquisition and maintenance, and for inter-cluster information exchanges. Therefore, the ISs in a cluster C may run internally any algorithm they wish, as long as the information provided externally to other clusters conforms to the following requirements:

- rapid convergence,
- ability to identify partitions,
- consistency with the information that would be generated by carrying out the operation described in 8.5.7.

Note 14:

This protocol can be extended to support global information distribution schemes so as to support alternate routeing algorithms; such extensions are left for further study.

9. INTER-DOMAIN / INTRA-DOMAIN INTERFACE

The inter-domain routeing protocol cannot route between routeing domains unless it is able to assess the accessibility of routeing domains. This is done by those inter-domain ISs that have links to intra-domain ISs. As a rule, the inter-domain ISs in question and the corresponding intra-domain ISs are expected to reside in the same physical equipment. However, this is not necessarily the case and, abstractly, the inter-domain / intra-domain interface will be modelled as a simplified ES routeing protocol.

The inter-domain ISs and the intra-domain ISs will attempt to establish a neighbourhood relationship exactly as if they were two inter-domain ISs. However, there will be no exchange of information across their links. The two ISs will only exchange periodic Hellos in order to maintain the neighbourhood relationship.

It is assumed that the inter-domain IS knows (static information) what routeing domain is reachable via its intra-domain neighbour, while the intra-domain neighbour can only assume that the inter-domain IS is an exit point to other routeing domains.

The implication of the preceding discussion is that all exit points of a routeing domain are treated as if they were equivalent. Conversely, all entry points to a routeing domain are treated as equivalent.

Note 15

Future versions of this protocol may wish to address the following points that, in this Technical Report, are left for further study:

- *Is it possible to leak within a routeing domain sufficient information about the outside world so that the ISs in the routeing domain can discriminate between adjoining inter-domain ISs ?*

  It would appear that the answer to this question will be dependent upon the routeing scheme used by the domain in question, and that the desired functionality can be obtained only by forcing the intra-domain ISs to implement some (or all) of the functionality of the inter-domain ISs.

- *Is it possible to use the inter-domain ISs to mend intra-domain partitions ?*

  Again, the answer seems to be dependent upon the routeing scheme used in the domain. For instance, if the clusters use a link state scheme internally, it is possible to advertise within the domain virtual links that are obtained by a concatenation of inter-domain internal links. Such a mechanism could also be used to mend cluster partitions.

10. ROUTEING FUNCTIONS BETWEEN INTER-DOMAIN ISs

10.1 Overview

As pointed out in Clause 8, each inter-domain IS has a RIB, and FIBs that are extracted from the RIB. Part of the RIB is obtained and updated by management protocols and within this Technical Report it will be treated as static informa-

tion. This Clause describes the structure of the static information available to this protocol, the dynamic information that this protocol maintains, and the procedures through which the dynamic information is obtained and updated. In other words:

- the means by which neighbourhood relations are obtained and maintained,
- the events which may trigger changes in the RIB and the FIBs of an IS, and the corresponding protocol procedures that effect these changes,
- the events which trigger updates to be sent to neighbours, and the corresponding procedures.

10.2 Local Environment Knowledge

The correct operation of this protocol is based on the assumption that each inter-domain IS can be provided with information that allows it to know which ISs are potential neighbours. This information can be available to the protocol described in this Technical Report either by pre-configured lists, or by locally-defined interactions with a management protocol, or by any other means outside the scope of this Technical Report. This protocol does not seek to establish if this information is globally consistent. However, when neighbourhood is established, the two ISs will ensure that they have compatible views of each other. That is, pairwise checks are performed during the neighbourhood establishment phase.

It is assumed that each cluster has a globally unique cluster identifier (CID).

Note 16

The specifics of how CIDs are obtained, as well as the relationships between the CID and the NETs of the ISs of a cluster, are for further study.

Each agreement between clusters has an agreement ID that has to be unique for the cluster pair in question.

Abstractly, each IS X can be thought of as having the following static information:

- the CID of its cluster,
- the CIDs of the clusters adjoining its own cluster, their relationship to its own, and a set of metrics that pertain to their diameters,
- a list of all potential neighbours $\{Y_1, Y_2, ..., Y_n\}$ and, for each such neighbour Y:
  - Y's NET,
  - the CID of Y's cluster,
  - the links $\{L_1, L_2, ..., L_k\}$ that exist between X and Y, and for each such link L:
    - the associated SNPA pair,
    - the metrics assigned to L,
    - L's tag (U, D, J, or I),

- a normalized description of the agreement that has been prearranged for X and Y (this agreement describes the allowable protocol exchanges, the format of the routeing information exchanges, and the values for the protocol parameters and timers),
- a unique agreement identifier,
- an agreement CRC that operates over the fields {X, X-cluster_id, Y, Y-cluster_id, L, normalized agreement_form, agreement_id}.

At negotiation time, X and Y will check if they have compatible views over a link L as follows.

The Acquirer, X, will encode a PDU that will contain, among others, the following information:

X / X-cluster_id / Y / Y-cluster_id / Relative-cluster-position /

L / tag(L) / agreement_id(L) / agreement CRC(L)

The Acquiree, Y, will check its static information and will either accept X as a neighbour over L, or it will not, and provide a reason as to why he does not.

Successfully negotiated links shall be maintained through the usage of timely exchanges of Hello PDUs (HL PDUs).

Note 17:

The tags of the links that join a pair of ISs X and Y must be compatible with the relative position of their clusters. That is, only jump links can join non-comparable clusters and when a U/D link joins two comparable clusters, the U-direction is from the lower to the higher cluster.

10.3 Standardized Form of a Negotiated Agreement

In its most general form, the routeing decision is the answer an inter-domain IS reaches on the basis of its RIB and of the RNPU PCI. Thus, the entire PCI can be seen as a key into the entries of the IS's FIBs. Usually, and this is the assumption made here, only the destination address is used. But, since the destination address is nothing more than a sequence of octets and the PCI is another, there is no conceptual difference between using the destination address only and using the whole PCI. It is nevertheless important that the size of the FIBs be kept manageable. This Technical Report assumes that the filters on a link (hence the agreements on same) are only destination address sensitive and that sensitivity to source addresses and other PCI options is obtained (when necessary) implicitly, through inter-cluster relationships, and by choosing the appropriate TGT.

Therefore, the nature of an agreement over a link consists in specifying who will offer what information for what destination addresses. Thus, the standardized form of an agreement over a link is:

- ID of Acquirer (1 octet NET length followed by NET value),
- ID of Acquiree (1 octet NET length followed by NET value),
- ID of link (1 octet),

- agreement identifier (4 octets),
- cumulative length of entries for Acquirer (4 octets),
- entries for Acquirer,
- cumulative length of entries for Acquiree (4 octets),
- entries for Acquiree.

Each Acquirer/Acquiree entry consists of a length identifier (1 octet), a TGT identifier, an NSAP identifier, and a mask identifier. It is assumed that:

i) the NSAP identifier is the smallest NSAP under this mask,

ii) within each category (Acquirer/Acquiree) the {NSAP, mask} pairs are in the lexicographic order,

iii) for each TGT the Acquirer (Acquiree) promises to provide routeing information for the sets of NSAPs described in the {NSAP, mask} pairs and to accept RNPUs destined for NSAPs that lie within one or more of the {NSAP, mask} pairs enumerated by the Acquirer (Acquiree).

The agreement checksum is calculated using a 32-bit Cyclic Redundancy Checking (CRC) calculation, based on the following standard generator polynomial.

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

10.4 Updating the RIB

As described in Clause 8, each IS X in a cluster C maintains locally-significant information, i.e.:

- information on all links that originate and/or end in C. This is unrestricted information whose synthetization index is 0;
- information on all synthesized ES-entries that it has received by its neighbours; the restriction tag of such information reflects the type of path(s) the neighbour uses to access the destinations in question.

Whenever a change occurs in X's RIB, X will pass on this information to its neighbours and will update its FIBs as needed. Moreover, if X is a border gateway, i.e. it has a neighbour Y in a cluster D other than C, then X will synthesize from its FIBs and communicate to Y as needed ES-entries that convey information about the NSAP sets enumerated in the agreement X and Y entered.

The following should be noted:

i) Each information item has an associated time-to-live field. When several information items $I_1, I_2, ..., I_k$ with associated time-to-live fields $T_1, T_2, ..., T_k$ are combined to derive information item I, its time-to-live field T is set to $T = \min \{T_1, T_2, ..., T_k\}$.

ii) When information I is exchanged between clusters, a holding timer HT is attached to the exchanged information. The value of HT should not exceed the time-to-live value, T, associated with I.

iii) Inter-cluster information exchanges can be triggered by timers, so that information item(s) in the receiving cluster will be refreshed (i.e. their time-to-live reset) before they are flushed.

iv) Inter-cluster links, even when they link the same ISs, will be treated independently. Consequently, if information over an NSAP set M can be offered over multiple links L_1, L_2, ..., L_n that join two IS's X and Y, this information will be offered over each and every link.

v) Information ordering over a link is obtained via PDU sequence numbers. At negotiation time the PDU sequence is initialized to zero. Subsequently, information is processed only in the order in which it was offered.

10.5 Derivation of the FIBs

The FIBs of an IS X in a cluster C are extracted from the locally significant information in the RIB and from the static information. To start, the synthesized ES-entries are of the form:

```
{   1:   provider's link PID;
    2:   set__of__NSAPs;
    3:   table generation tag;
    4:   restriction tag;
    5:   metrics;          5a: extent of coverage;
                           5b: administrative distance;
                           5c: synthetization index;
}
```

The field «table generation tag» (TGT) is an index that indicates which FIB can use this information. The IS in question will create different FIBs for each TGT it supports. In all subsequent sections it is assumed that information items with different TGTs are treated separately and independently.

The field «restriction tag» can assume two values, «off» and «on», whose semantics are as below:

off: when all paths used towards the advertised NSAP set cross inter-cluster links exclusively in the down direction;

on: when one or more of the advertized NSAPs are reached through paths that cross either jump links, or links in the up direction, or both.

The field «extent of coverage» can have three values: «all», «some», or «none». For a given set of NSAPs and a given restriction tag, this field describes the accessibility of the routeing domains that contain actual addresses within the set of NSAPs via paths that are compatible with the restriction tag.

The semantics of this field are thus:

all: if it is known all actual NSAPs lie in accessible routeing domains;

some: if it is believed that some actual NSAPs lie within accessible routeing domains and some do not;

none: if it is known that all actual NSAPs lie within inaccessible routeing domains.

Before the FIBs are derived, the IS will ensure that if two ISs report identical values for fields 1-4, then fields 5 are identical. If this is not true, then X will compute the pessimal value for fields 5 and will use this value in all further computations.

Once this is done, X will augment the metrics-field 5b by the diameter C assigns to the cluster of the reporting IS. Moreover, where X receives a number of ES-entries with similar metrics, it may choose to treat these metrics as if they were identical. If this is the case, X will perform its computations as if the pessimal of the retained values applied.

Then, X will proceed as follows:

- Among all synthetic ES-entries with the restriction tag «off», X will retain those that report the best metrics (best extent of coverage and, for the same extent of coverage, smallest administrative distance). Out of the retained ES-entries, X will construct FIB1 (TGT), that will be available to all but C's children.
- Among all ES-entries X will retain those with best metrics, regardless of the value of the restriction tag. Then it will construct FIB2 (TGT), that will be available only to C's children.

The information in either database is of the form:

{ set_of_NSAPs;
next IS;
metrics;
other;
}

10.6 Generation and Propagation of Synthetic ES-entries

If IS X in cluster C is neighbour to IS Y in cluster D, then X undertakes by virtue of the C-D agreement to send to Y information on several sets of NSAPs, M_1, M_2, ..., M_k for one or more TGTs. If X > Y is **not** in the *(d)*-direction, the information X sends to Y is obtained by use of FIB1 (TGT). If it is the *(d)*-direction, then FIB2 (TGT) is used.

Given a set M, it is first determined which of the entries of X's FIBs intersect M. By renumbering, if necessary, assume that the entries in question are E_1, E_2, ..., E_k. For each collection $T = \{E_{i1}, E_{i2}, ..., E_{in}\}$ of E_i's that cover M, let the associated metrics be D_{i1}, D_{i2}, ..., D_{in} and let:

$$D(T) = \text{pessimal}\,(D_{i1}, D_{i2}, ..., D_{in})$$

The pessimal operator returns in the extent of coverage field the value «all» if all input arguments are «all», the value «none» if all input arguments are «none», and the value «some» in all other cases.

For all other fields, the returned value is the maximum taken over the corresponding fields of the metrics for which the extent of coverage is not «none».

Among all coverings, X will choose the T for which D(T)'s extent of coverage field has the best value possible and, if it is practicable, it will attempt to choose T so that the resulting D(T) is optimal. All ISs in the same cluster must use the same ranking function.

Note 18:

- *For each T, D(T) is independent of the computing IS X.*
- *If there are few minimal covering sets, these sets can be precomputed. In that instance it suffices to compute D(T) for all these minimal covering subsets and to have some ranking rule so that T and D(T) be the same for all Xs.*
- *Within reason, it is not necessary to choose an optimal D(T). Any D(T) that gives the best possible extent of coverage is acceptable, provided that all Xs use the same algorithm and report the same metrics. Thus, if the minimal sets are too many to precompute, one can use any among the many straightforward algorithm that can produce a reasonable D(T).*
- *When the extent of coverage for a set E_i is «some», it is assumed that the appropriate value for the intersection of E_i and M is also «some». In reality, the true value may be «all» or «none». Nevertheless, as a rule it is impossible to know which value really applies and a conservative choice is to assume that the appropriate value is «some». An archconservative choice would be to assume that the value is «none». That would be consistent with the philosophy of assuming that when a distance D is advertised, the true distance is never worse than D. In that case one could also expect that when an extent of coverage is advertised, the true value should never be worse than the advertised value.*

Once the appropriate T is found, X will increase the synthetization index by one, and will report over each link L that joins X and Y, and for each TGT, the information on M as follows:

```
{   X's equivalence class (PID);
    set M;
    TGT of FIB used;
    restriction tag;
    metrics (= D(T));
}
```

with the restriction tag «on» unless the direction X->Y is up.

11. PROTOCOL SPECIFICATIONS

Figures 4 and 5 depict the state diagram for both, the Acquirer and Acquiree during the negotiation phase. This Clause provides a complete procedural description of the functions to be performed by the RDI protocol.

Note 19:

It is assumed that PDUs can and shall be constructed in such a way that segmentation will not occur.

11.1 Routeing Domain Interconnection (RDI) Restart Procedures

11.1.1 Restart Initiation Procedure

The restart initiation procedure is invoked upon the intervention of System Management functions and causes the IS entity to transition from whatever state it is currently in to the Initialization State (INIT_STATE).

Upon entering the INIT_STATE, the entity will clear any residues from any previously existing association, including purging the buffers, resetting the counters and timers. Subsequently, the entity generates a Restart Request PDU (RR PDU) and transmits it to the intended destination. In addition, the entity starts the Restart Request Timer (RRT) so that if a Restart Response PDU (RS PDU) is not received within the time range specified by the RRT, then the entity retransmits the RR PDU. The maximum number of times which the RR PDU can be retransmitted is specified by the Restart Request Counter (RRC). If the RRC is exceeded, the entity signals System Management and transitions to the Idle State (IDLE_STATE).

Upon receiving an RS PDU in response to an RR PDU, the entity assumes that the Restart Initiation Procedure has successfully completed, hence it invokes the Waiting Procedure. If, instead, the entity receives an RR PDU from a destination entity to which an RR PDU has been transmitted prior to receiving the RS PDU, then the entity responds with an RS PDU, assumes that the Restart Initiation Procedure has successfully completed and invokes the Waiting Procedure.

11.1.2 Restart Reception Procedure

The Restart Reception Procedure is invoked by the entity upon reception of an RR PDU if the entity is in any state other than the INIT_STATE.

Note 20:

The case where an RR PDU is received while in the INIT_STATE is addressed in 11.1.1. When invoked, this procedure causes clearing of any residues from any previously-existing association, including purging the buffers, resetting the counters and timers. Subsequently, an RS PDU is transmitted to the originator of the received RR PDU and the Waiting Procedure is invoked.

11.2 Routeing Domain Interconnection (RDI) Waiting Procedure

When invoked, the RDI Waiting Procedure sets the Waiting Timer (WT) to a value so as to ensure successful expiry of any outstanding RR PDU(s) or RS PDU transmitted and any other still living PDU from the previous association.

While in the Waiting State (WAIT_STATE), any PDU received, other than an RR PDU, is discarded. The reception of an RR PDU causes the invocation of the Restart Reception Procedure (see 11.1.2).

11.3 Routeing Domain Interconnection (RDI) Neighbour Acquisition Procedures

Upon expiry of the WT, an entity transitions into the Ready State (READY_STATE). Once in the READY_STATE, the actions to be taken are dependent on the designation of the particular entity.

For any given pair of neighbouring entities, one entity is designated as the Acquirer while the other is designated as the Acquiree. The Acquirer initiates an association, while the Acquiree decides whether or not it wants to participate in the association. The designation of Acquirer and Acquiree is left as a System Management function.

11.3.1 Neighbour Acquisition By Acquirer Procedure

Once in the READY_STATE, the Acquirer generates an Acquisition Request PDU (AR PDU) and forwards it to a designated Acquiree. The AR PDU conveys to the Acquiree the notion that an association is being requested for the purpose of exchanging information as described in some agreement A. The identifier for agreement A is contained within the AR PDU.

Upon forwarding an AR PDU, the Acquirer sets the Acquisition Attempt Timer (AAT). If the AAT expires prior to receiving an Acquisition Response PDU (AS PDU), the Acquirer retransmits the AR PDU. The maximum number of times an Acquirer retransmits an AR PDU is specified by the Acquisition Attempt Counter (AAC). If all attempts fail, the Acquirer invokes the RDI Restart Procedure (see 11.1).

When the Acquirer receives an AS PDU in response to an AR PDU containing the appropriate identifier for agreement A, the Acquirer assumes that the association has been correctly established, and invokes the Data Phase Procedure (see 11.4).

If the Acquirer receives an Acquisition Disconnect PDU (AD PDU) containing a different agreement identifier than it had requested, the Acquirer informs System Management, and enters the IDLE_STATE.

11.3.2 Neighbour Acquisition By Acquiree Procedure

When the Acquiree enters the READY_STATE, it waits for an AR PDU to arrive. Upon receiving an AR PDU, the Acquiree extracts the information concerning the proposed association (e.g. the agreement identifier, etc.), and decides whether the requested association is acceptable (i.e. whether the agreement identified in the AR PDU is the one expected). If so, the Acquiree generates an AS PDU, copies in it the same agreement identifier as requested by the Acquirer and forwards it to the Acquirer, hence entering the READY_FOR_DATA_STATE.

Once the Acquiree is in the READY_FOR_DATA_STATE, it is waiting for data exchanges to begin; any duplicate AR PDUs received while in this state will be answered with the appropriate AS PDUs.

If the agreement identified by the AR PDU is not that expected by the Acquiree, the Acquiree informs System Management, generates an Acquisition Disconnect PDU (AD PDU) containing what it believes to be the appropriate agreement identifier and sends it to the Acquirer. In this case, the Acquiree will enter the IDLE_STATE.

11.4 Routeing Domain Interconnection (RDI) Data Phase Procedures

Upon receiving an AS PDU, the Acquirer enters the DATA_EXCHANGE_STATE. It sets its send and receive counters for the link to zero. It then sends either a Data Update PDU (DU PDU) or a Hello PDU (HL PDU) to the Acquiree as described in 11.4.1 or 11.4.2.

Upon receiving a DU PDU, or a HL PDU, the Acquiree sets its send and receive counters for the link to zero and enters the DATA_EXCHANGE_STATE (from the READY_FOR_DATA_STATE). It then processes the received PDU as described in 11.4.3.

11.4.1 Sending a Data Update PDU (DU PDU)

Data Update PDUs are used to convey the information about ES-entries. One DU PDU can contain information for a number of ES-entries. Information for each entry is contained in a separate block, as shown in Figure 10.

The function of a DU PDU is to describe sets of NSAPs reachable through the sending cluster. The circumstances under which they shall be generated are described in 10.4.

When a DU PDU is generated the current value of the send counter is placed in the sequence number field of the PDU. The send counter shall then be incremented (mod 64K). The DU PDU shall then be transmitted on the link and a Data Update Timer (DUT) shall be started. If this timer expires before the PDU has been acknowledged the DU PDU shall be retransmitted. This is repeated until either the DU PDU is acknowledged or the number of transmissions of the DU PDU exceeds the Data Update Counter (DUC). If the DUC is exceeded, the transmitting entity shall signal its System Management and enter the IDLE_STATE on that link. A system that is waiting for 32K acknowledgements shall not send any further DU PDUs.

11.4.2 Sending a Hello PDU (HL PDU)

The purpose of HL PDUs is to maintain links on which no RDI data is being sent.

A HL PDU may be sent at any time by an entity in the DATA_EXCHANGE_STATE. It shall be sent on a link if no RDI PDUs of any type have been sent on that link for the period of the Hello Timer (HLT).

11.4.3 Receiving a PDU

On receiving a DU PDU the sequence number of the PDU is compared with the receiving entity's receive counter. If the sequence number is in the range *receive counter +1* to *receive counter +32K* (mod 64K) the DU PDU is not acknowledged and the data it contains is ignored. If the receive counter is equal to the sequence number then the data in the PDU is used to update the RIB as described in 10.4. If the sequence number is in the range *receive counter -32K +1* to *receive counter* (mod 64K) then it is acknowledged by sending a DK PDU with the same sequence number. The receive counter is then incremented (mod 64K).

If no RDI PDU of any type is received on a link for the period of the Keep Alive Time (KAT) the entity shall signal its System Management and enter the IDLE_STATE on that link. The KAT must be greater than twice the neighbours HLT.

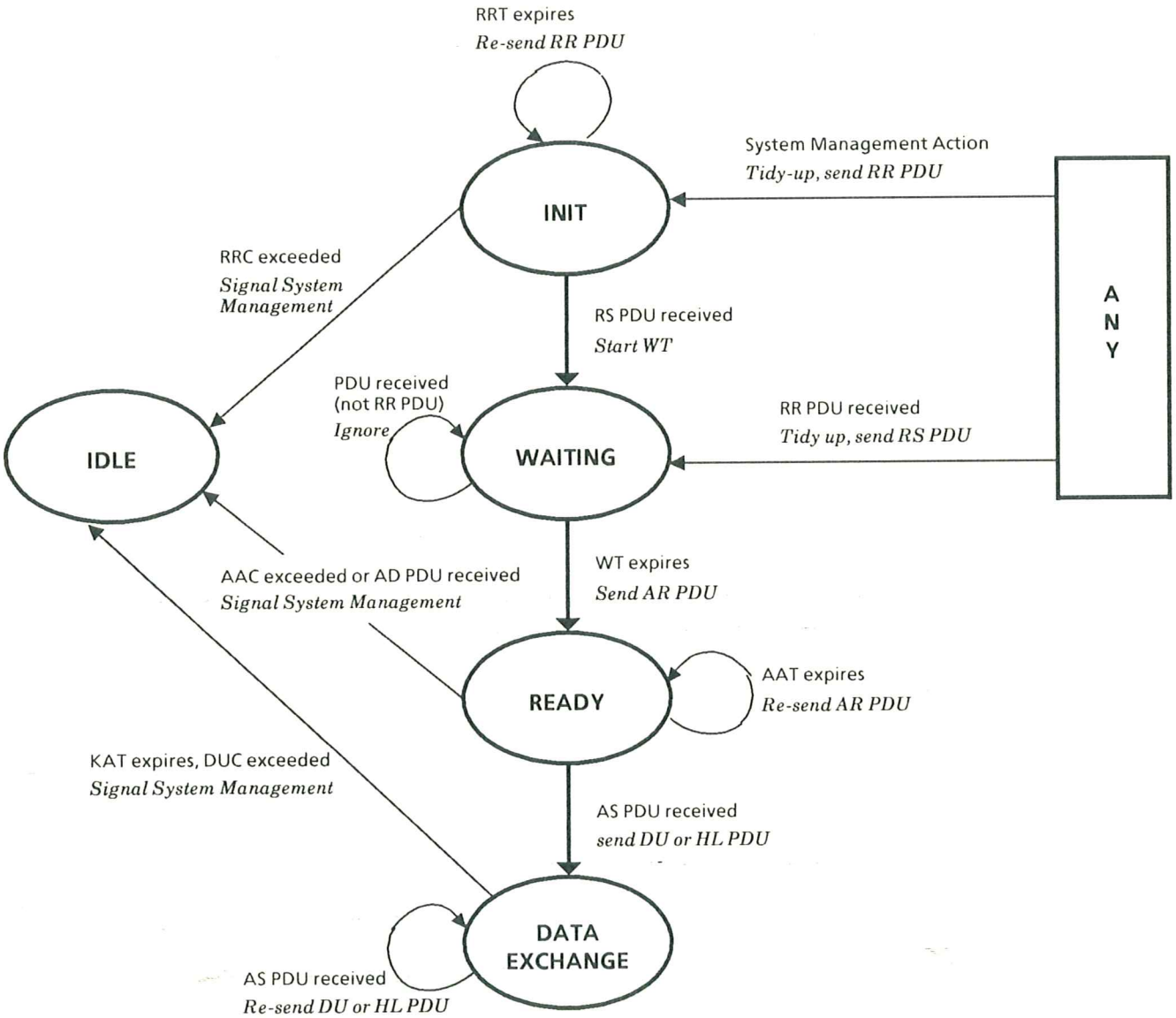


Figure 4 : Negotiation Phase, State Diagram for the Acquirer

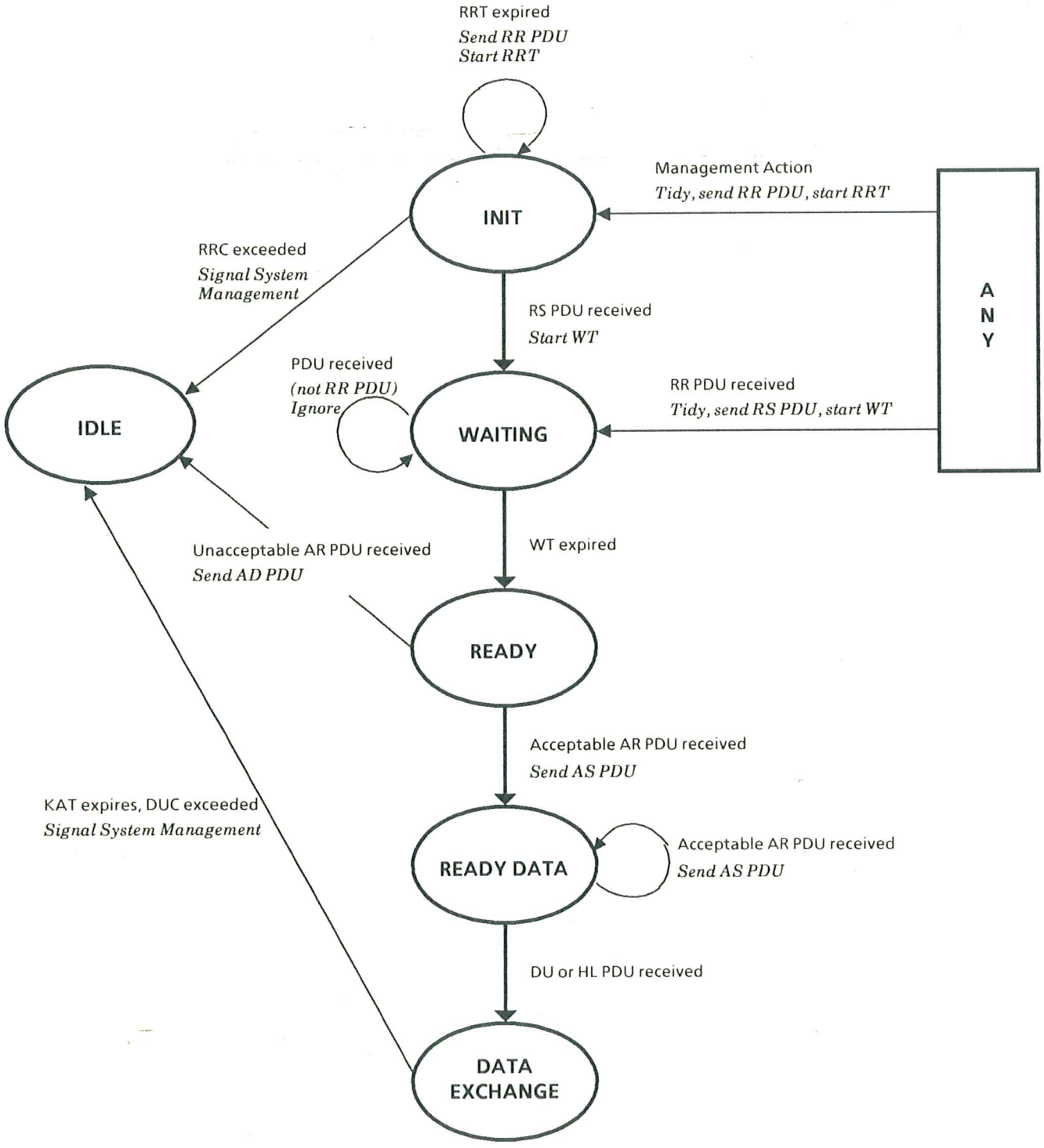


Figure 5 : Negotiation Phase, State Diagram for the Acquiree

11.5 Header Error Detection

The Header Error Detection function protects against failures of network entities due to the processing of erroneous information in the PDU header. The function is realized by a checksum computed on the entire PDU header. If the checksum calculation fails, the PDU must be discarded.

The use of the Header Error Detection function is optional and is selected by the originating network entity. If the function is not used, the checksum field of the PDU header is set to zero.

If the function is selected by the originating network entity, the value of the checksum field satisfies the following equations:

$$\left[\sum_{i=1}^{i=L} a_i \pmod{255} \right] = 0$$

$$\left[\sum_{i=1}^{i=L} (L - i + 1)\, a_i \pmod{255} \right] = 0$$

where L is the number of octets in the PDU header, and a_i the value of the octet at position i. The first octet in the PDU header is considered to occupy position $i = 1$.

When the function is in use, neither octet of the checksum field may be set to zero.

11.6 Protocol Error Processing Function

A PDU in which the Network Layer Protocol Identifier (NPID) field is present with the value defined in 12.2.1 and the version/protocol (V/P) identifier extension is present with the value defined in 12.2.3, and which is not discarded by the header error detection function, shall be considered a protocol error if its encoding does not comply with the remainder of the provisions of 12.2.1. Any such protocol error PDU shall be discarded.

Note 21:

PDUs in which the NPID has a value other than that in 12.2.1 or in which the V/P field has a value other than that in 12.2.3 are outside the scope of this Technical Report.

12. STRUCTURE AND ENCODING OF PROTOCOL DATA UNITS (PDUs)

This Clause describes the structure and encoding of protocol data units (PDUs) exchanged between peer RDI protocol entities.

12.1 Structure

All protocol data units shall contain an integral number of octets. The octets in a PDU are numbered in an increasing order starting from one (1). The bits in an octet are numbered from one (1) to eight (8), where bit one (1) is the low-order bit.

When consecutive octets are used to represent a binary number, the lower-numbered octet has the more significant value.

Note 22:

In this Clause, where encoding of the PDU is represented using a diagram, the following representation is used:

- *octets are shown with the lowest-numbered octet to the left (or to the top), higher-numbered octets being further to the right (or further down);*
- *within an octet, bits are shown with bit eight (8) to the left, and bit one (1) to the right.*

PDUs shall contain the following general components, in the order listed:

i) the Fixed part,
ii) the Network Address part, and
iii) the Data part, if present.

The structure of the PDU is shown in Figure 6.

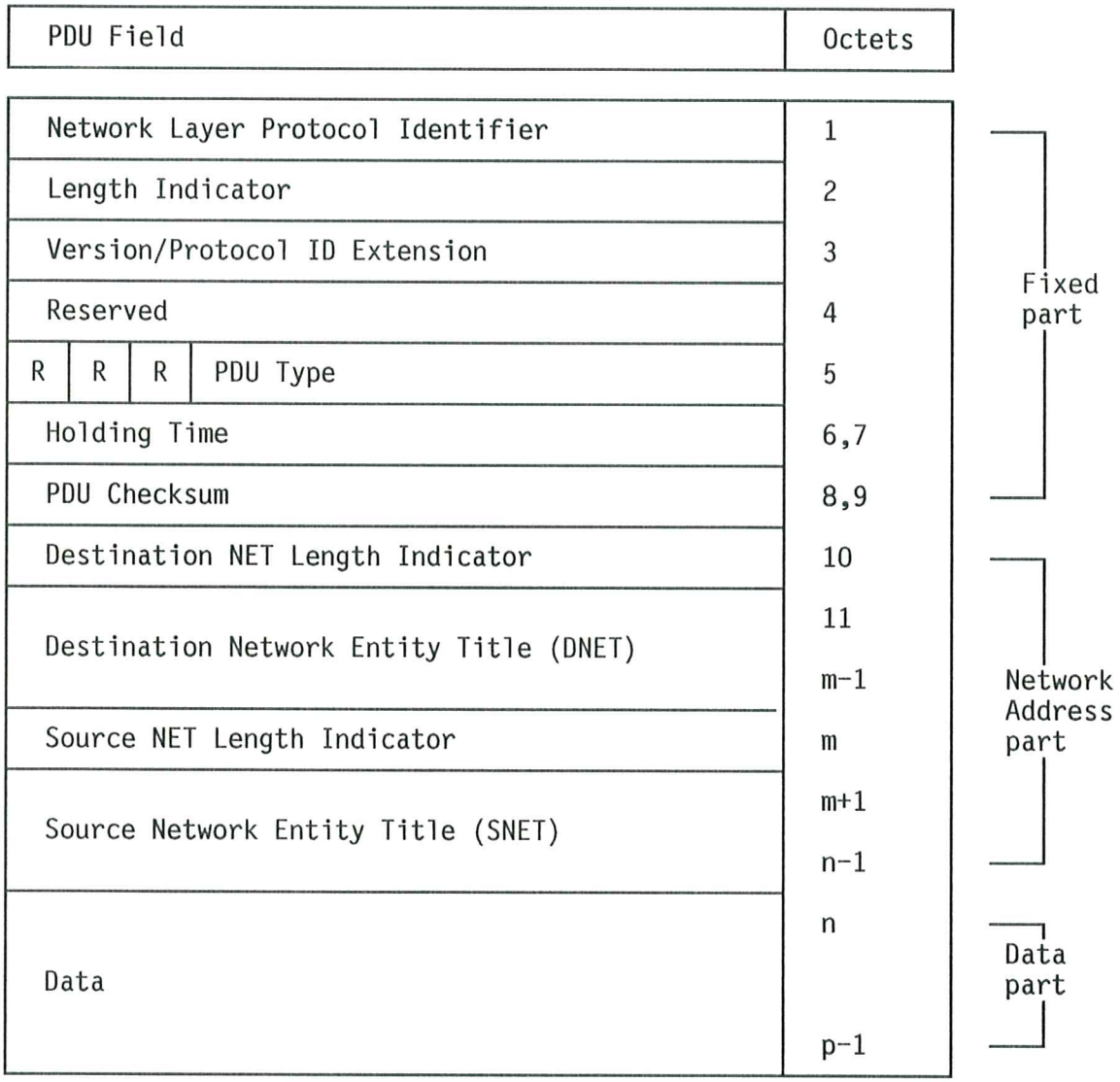
12.2 Fixed Part

The Fixed part contains frequently-occurring parameters including the Type Code of the PDU.

12.2.1 Network Layer Protocol Identifier

This field identifies the Network Layer protocol defined in this Technical Report as ECMA TR/*xx* «Inter-Domain Intermediate Systems Routeing».

The value of this field is to be determined.



R : reserved

Figure 6 : PDU General Structure

12.2.2 Length Indicator

The length is indicated by a binary number, with a maximum value of 254 (1111 1110). The length indicated is the length in octets of the PDU header. The value 255 (1111 1111) is reserved for possible future extensions.

12.2.3 Version / Protocol Identifier Extension

The value of this field is binary 0000 0001. This identifies a standard version of the protocol defined in this Technical Report, i.e. ECMA TR/*xx*.

12.2.4 PDU Type

The PDU Type field identifies the type of the PDU. Defined PDU types are given in Table 1.

Note 23:

Bits 6, 7, and 8 are reserved, which means that they are transmitted as zeros and ignored on receipt.

	PDU Type	Encoding
AD PDU	Acquisition_Disconnect	0 0 1 0 1
AR PDU	Acquisition_Request	0 0 0 1 1
AS PDU	Acquisition_Response	0 0 1 0 0
DK PDU	Data_Acknowledgement	0 0 1 1 1
DU PDU	Data_Update	0 0 1 1 0
HL PDU	Hello	0 1 0 0 0
RR PDU	Restart_Request	0 0 0 0 1
RS PDU	Restart_Response	0 0 0 1 0

Table 1 : Valid PDU Types

All other PDU types are reserved for future expansion.

12.2.5 Holding Time

The Holding Time field specifies the maximum time for the receiving network entity to retain the routeing information, if any, contained in this PDU. The Holding Time field is encoded as an integral number of seconds.

12.2.6 PDU Checksum

The checksum is computed on the entire PDU. The usage of this field is identical to that specified by ISO 8473.

12.3 Network Address Part

12.3.1 General

Address parameters are distinguished by their location. All PDU types convey the Destination Address followed by the Source Address of the communicating IS pair.

12.3.2 Network Protocol Address Information (NPAI) Encoding

The Destination and Source Addresses are Network Entity Titles (NETs) as defined in ISO 8348/Add2, Addendum to the Network Service Definition Covering Network Layer Addressing. These addresses are encoded as NPAI using the binary syntax defined in 8.3.1 of ISO 8348/Add2.

12.4 Data Part

The Data part of the PDU is structured as an ordered multiple of octets. Further structuring and semantics of the Data part are defined by individual PDU types.

12.4.1 Acquisition_Disconnect PDU (AD PDU)

The format of the Data part of the AD PDU is illustrated in Figure 7. Release reasons are to be defined.

PDU Fields	Octets
Data Part Length	n n+1
Reason for Disconnect	n+2 n+3

Figure 7 : AD PDU Data Part

12.4.2 Acquisition_Request PDU (AR PDU)

The format of the Data part of the AR PDU is illustrated in Figure 8.

PDU Fields	Octets
Data Part Length	n n+1
Proposer CID Length (p)	n+2
Proposer CID	n+3 n+p+2
Proposee CID Length (q)	n+p+3
Proposee CID	n+p+4 n+p+q+3
Relative CID Position	n+p+q+4
Tag	n+p+q+5
Agreement Identification	n+p+q+6 n+p+q+9
Agreement CRC	n+p+q+10 n+p+q+17

Figure 8 : AR PDU Data Part

12.4.3 Acquisition_Response PDU (AS PDU)

The AS PDU has no Data part.

12.4.4 Data_Acknowledgement PDU (DK PDU)

The format of the Data part of the DK PDU is illustrated in Figure 9.

PDU Fields	Octets
Data Part Length	n n+1
PDU Sequence Number Acknowledged	n+2 n+3

Figure 9 : DK PDU Data Part

12.4.5 Data__Update PDU (DU PDU)

The format of the Data part of the DU PDU is illustrated in Figure 10.

PDU Fields	Octets	
Data Part Length	n n+1	
PDU Sequence Number	n+2 n+3	
PID Length (p)	n+4	
PID	n+5 n+p+4	
NSAP Length (q)	n+p+5	Block 1
NSAP	n+p+6 n+p+q+5	
MASK	n+p+q+6 n+p+2q+5	
Table Generation Tag	n+p+2q+6	
Restriction Tag	n+p+2q+7	
Extent of Coverage	n+p+2q+8	
Administrative Distance	n+p+2q+9 n+p+2q+12	
Synthetization Index	n+p+2q+13 n+p+2q+14	
. . .		
Same structure as Block 1		Block x
. . .		

Figure 10 : DU PDU Data Part

12.4.6 Hello PDU (HL PDU)

The HL PDU has no Data part.

12.4.7 Restart_Request PDU (RR PDU)

The RR PDU has no Data part.

12.4.8 Restart_Response PDU (RS PDU)

The RS PDU has no Data part.

SECTION III

SUBNETWORK-DEPENDENT FUNCTIONS

13. PROTOCOL DEPENDENCIES

It is assumed that when supporting CLNS, underlying connectionless subnetwork procedures will be used and that when supporting CONS, underlying X.25 procedures will be used (this is not essential for the operation of the protocol, but is in line with usual network layer methods.

13.1 Protocol Dependencies for Use of CL Subnetwork Service

13.1.1 Facilities required from the Subnetwork Service

The subnetwork service required to support this protocol is defined by the primitives shown below:

Primitives	Parameters
SN_UNITDATA Request	SN_Destination_Address
SN_UNITDATA Indication	SN_Source_Address
	SN_Quality_of_Service
	SN_Userdata

Figure 11 : Subnetwork Service Primitives for Underlying CL Subnetwork Service

The mechanisms through which this service is provided are the same as those in ISO 8473.

Note 24:

This protocol is based on the assumption that more than one inter-domain IS may reside in the same equipment and that more than one logical link may exist between two inter-domain ISs (clusters). It is therefore assumed that:

- *one can distinguish, if necessary, between co-located ISs;*
- *it is possible to distinguish, when necessary, between multiple logical links.*

A mechanism for doing this may be based on multiple SNPAs. This subject is left for further study.

13.1.1.1 Subnetwork Addresses

The source and destination addresses specify the points of attachment to a public or private subnetwork(s) involved in the transmission SNPAs. Subnetwork addresses are defined in the service definition of each individual subnetwork.

The syntax and semantics of subnetwork addresses, except for the properties described above, are not defined in this Technical Report.

13.1.1.2 Subnetwork User Data

The SN_Userdata is an ordered multiple of octets, and is transferred transparently between the specified subnetwork service access points.

13.1.2 Interactions with ISO 8473

It is assumed that each RNPU to be forwarded contains a PCI that reflects the path traversed, which is the RNPU-restriction tag described in 8.5.3.

The initial value of this field is «off». It remains «off» until a jump-link or a link in the down direction is crossed. After that time, the value of this field is «on». While the value of this field is «off», each IS will forward the RNPU on the bases of FIB2 (TGT). Once the field is set, i.e. its value is «on», only FIB1 (TGT) will be consulted.

If an NPDU X is fragmented in NPDUs X_1, X_2, ..., X_K, then the RNPU-restriction tag in each of X_1, X_2, ..., X_K will be equal to the RNPU-restriction tag of X. Conversely, if during intermediate reassembly NPDUs X_1, X_2, ..., X_M are reassembled into NPDU X, the RNPU-restriction tag of X will be "on" unless the RNPU-restriction tag of X_1, X_2, ..., X_M are all "off". In the latter case the RNPU-restriction tag of X will be "off".

13.1.3 Local Parameters

There are no local parameters other than those defined in 8.5.4.

13.2 Protocol Dependencies for Use of CO Subnetwork Service

13.2.1 Procedures for Use of ISO 8208 Subnetworks

a) Each PDU defined in 12 is transmitted as a single n-bit sequence on an X.25 switched virtual circuit (SVC).Use of Permanent Virtual Circuits (PVCs) is for further study.

b) Each X.25 SVC is set up by means of a call request in which the first octet of the call user data has the value defined in 12.2.1.

c) Either IS may try to establish a Virtual Call (VC) if it has a PDU to send and there is no existing VC to send it to. If two VCs are established simultaneously (i.e. connect indication arrives before connect confirm) then that initiated by the Acquiree is disconnected.

d) Use of multiple circuits is for further study.

e) Diagnostic codes identified for use in clear request packets:

241	Circuit disconnected because not used for a long time (see also Closing Timer, 13.2.3)
245	reason unspecified, permanent condition
246	temporary congestion - try again later
248	Collision occurred (as described in c) above)

f) If a circuit cannot be established to send a PDU, the PDU should be discarded.

Note 25

It would be possible to describe optional transmission of PDUs in fast select call / clear user data. This method is for further study.

13.2.2 Interactions with ISO 8878 and ISO 8208

It is assumed that each RNPU to be forwarded contains a PCI that reflects the path traversed, which is the RNPU-restriction tag described in 8.5.3. One possible method for conveying this field could be as an X.25-DTE-facility.

The initial value of this field is «off». It remains «off» until a jump-link or a link in the down direction is crossed. After that time, the value of this field is «on». While the value of this field is «off», each IS will forward the RNPU on the bases of FIB2 (TGT). Once the field is set, i.e. its value is «on», only FIB1 (TGT) will be consulted.

13.2.3 Local Parameters

- Those defined in 8.5.4.
- A Closing Timer (CT) for closing down a circuit which has not been used recently.

APPENDIX A

AN EXAMPLE

Consider that a company X gets its addresses from distinct naming authorities. Further assume that X cannot efficiently operate intra-domain (L_1) routeing schemes over a routeing domain that uses both types of addresses, but can operate two routeing domains, one for each addressing scheme, rather well. In this case, a reasonable scenario may be that shown in Figure A-1.

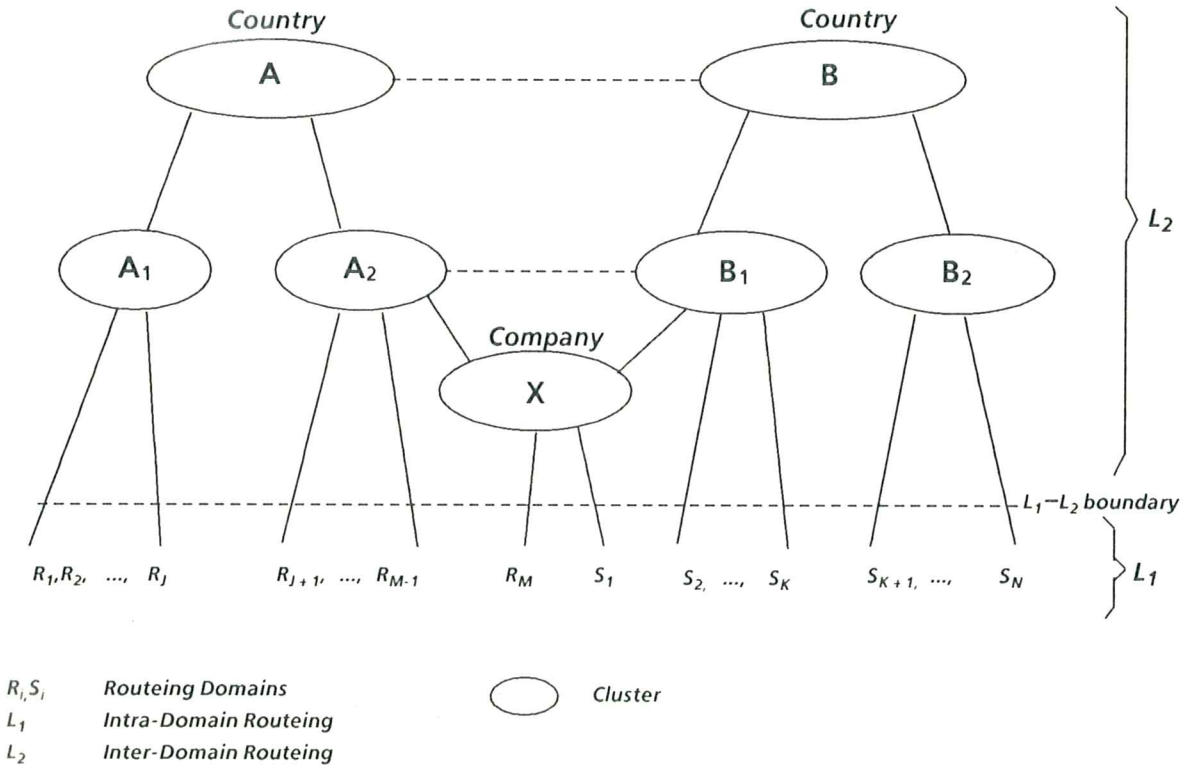


Figure A-1 : Example

In Figure A-1 solid lines show up-/down-movement while the dotted lines show a bilateral agreement. Each such single line represents one or more links between ISs. It is also assumed that R_1, ..., R_M are routeing domains using addressing scheme 1 while S_1, ..., S_N are using addressing scheme 2.

Cluster X serves to isolate R_M and S_1 from the rest of the world and to provide a reliable and efficient mechanism for routeing between R_M and S_1.

For the L_2-links that cross the L_1 - L_2 boundary two things are important:

- their existence and well being,
- the tariffs to be paid, if any, when R_M-originating traffic goes to S_1 and vice-versa.

Thus, each IS in X is expected to have the type of information database shown in Figure A-2.

ROUTEING DOMAIN / SET OF ROUTEING DOMAINS	DISTANCE	NEXT IS
R_M	distance	next IS
S_1	distance	next IS
R_{J+1}, ..., R_{M-1}	distance	next IS
R_1, ..., R_J	distance	next IS
S_2, ..., S_K	distance	next IS
S_{K+1}, ..., S_N	distance	next IS

Figure A-2 : Database of ISs belonging to Cluster X

The structure of this database is mostly determined by the model and by the topological structure shown in figure A-1. Other factors, such as tariffs and costs, may effect the final outcome.

Looking at cluster A_2, a different picture may emerge. A_2 may not be concerned with efficient ways to reach S_1, therefore its ISs may elect to use the database shown in Figure A-3.

ROUTEING DOMAIN / SET OF ROUTEING DOMAINS	DISTANCE	NEXT IS
$R_1, \ldots, R_J$	distance	next IS
R_{J+1}	distance	next IS
R_{J+2}	distance	next IS
	. . .	. . .
R_M	distance	next IS
$S_1, \ldots, S_K$	distance	next IS
$S_{K+1}, \ldots, S_N$	distance	next IS

Figure A-3 : Database of ISs belonging to Cluster A_2

Exchanges between A_2 and X will be dictated by the restrictions the model implies and the format of their databases:

The ISs in X are expected to provide distance and reachability information for R_M, but not necessarily for S_1 because A_2 has decided to see S_1, ..., S_K as a single entity and not to distinguish between S_1 and the rest.

The ISs in A_2 are expected to provide to X information on R_1, ..., R_J and R_{J+1}, ..., R_{M-1} , not on specific routes towards R_{J+1}, ..., R_{M-1} seen individually. Thus the ISs in A_2 will summarize their R_{J+1}, ..., R_{M-1} data

Examining now the exchange between A_2 and B_1:

A_2 can provide to B_1 only data on R_{J+1}, ..., R_M while B_1 can provide to A_2 data on S_1, ..., S_K. Depending on their limitations, it may be desirable that the information exchanged be for the whole range rather than on an individual basis.

Typical entries in each IS's database may look as shown in Figures A-4 and A-5.

ROUTEING DOMAIN / SET OF ROUTEING DOMAINS	RESTRICTION TAG	ADMINI-STRATIVE METRIC	INTRA-CLUSTER METRIC	NEXT IS
R_M	off	A_1	I_1	ID of next IS
S_1	off	A_2	I_2	ID of next IS
R_{J+1}, ..., R_{M-1}	on	A_3	I_3	ID of next IS
R_1, ..., R_J	on	A_4	I_4	ID of next IS
. . .				

Figure A-4 : FIB2(TGT)-Database entries of ISs belonging to Cluster X

Note A.1:

Figure A-4 assumes that the databases had time to converge and that there are no congestion, load balancing or other considerations that forces part of the traffic to be routed through one cluster while another part is routed through another cluster.

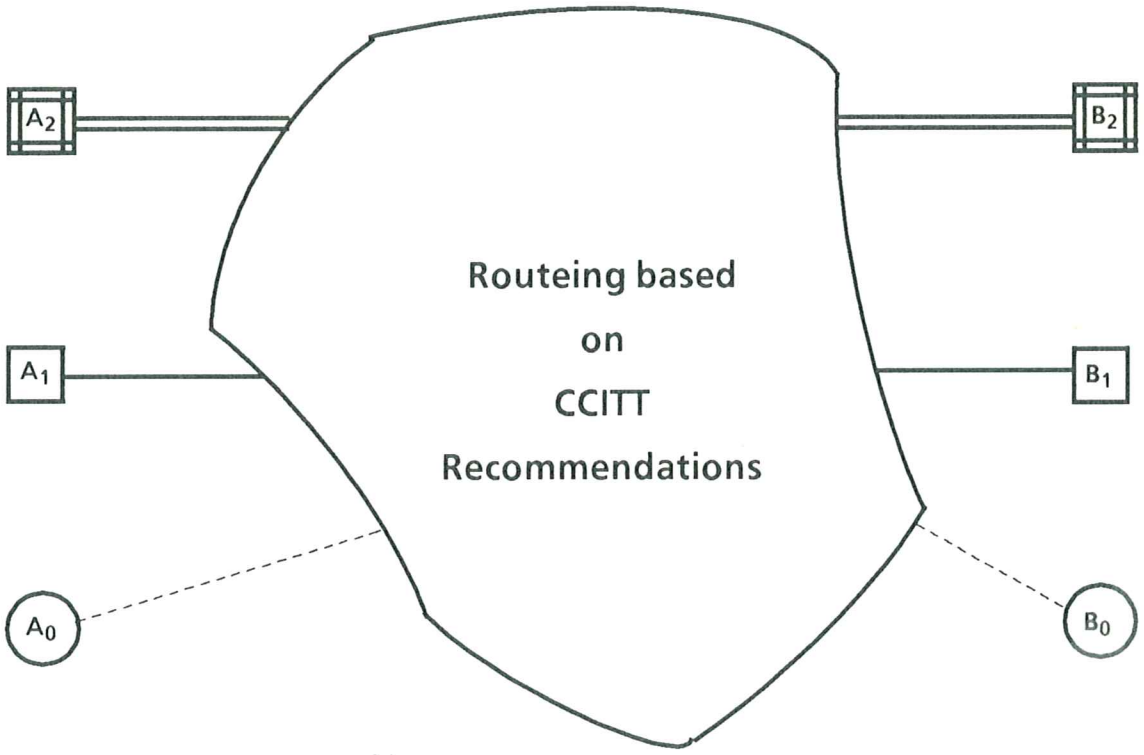
ROUTEING DOMAIN / SET OF ROUTEING DOMAINS	RESTRICTION TAG	ADMINI-STRATIVE METRIC	INTRA-CLUSTER METRIC	NEXT IS
R_{J+1}, ..., R_M	on	A_1'	I_1'	ID of next IS
R_1, ..., R_J	on	A_2'	I_2'	ID of next IS
. . .				

Figure A-5 : FIB2(TGT)-Database entries of ISs belonging to Cluster E

APPENDIX B

INTER-DOMAIN ROUTEING AND ENTITIES FOLLOWING CCITT RECOMMENDATIONS

CCITT networks can be used to pass routeing information at all levels, thus inter-domain routeing information may be transferred across CCITT networks. But CCITT networks may also be used to carry intra-domain and ES routeing information (see Figure B-1). The routeing carried out by the CCITT networks is invisible to the ESs and ISs using it. In particular, two inter-domain ISs connected through a CCITT network can be treated as directly connected for the purpose of inter-domain routeing.



A_0 and B_0: ESs
A_1 and B_1: Intra-Domain ISs
A_2 and B_2: Inter-Domain ISs

Figure B-1 : Routeing through CCITT Networks

Note B.1:

It is a subject for further study to decide if, and under which circumstances, redirects may be used to establish direct connectivity between systems in different domains and/or clusters. The resolution of this matter should not invalidate the basic principles established in this Technical Report.

APPENDIX C

A DISTRIBUTED NAMING AND REGISTRATION AUTHORITY

This Technical Report assumes that there is a Registration Authority that maintains the correct set of relationship (i.e. no loops). For obvious reasons, it is desirable that this authority be distributed. It is assumed that the agreements are of the form (X, Y, t, T, other) where X and Y (X parent to Y, i.e. there is a link from X to Y and the direction X to Y is down) are two clusters and [t, T] is the time interval over which the relationship is valid. A set of such relations is consistent if and only if there is no sequence X_1, X_2, ..., X_m such that (X_1, X_2, t_1, T_2, other), (X_2, X_3, t_2, T_3, other), ..., (X_i X_{i+1}, t_i, T_{i+1}, other), ..., (X_{m-1}, X_m, t_{m-1}, T_m, other), (X_m, X_1, t_m, T_1, other) all hold true at some time t'.

As a rule, in order to guarantee consistency, one new relationship has to be processed at a time. Moreover, if the introduction of a new entry is dependent upon the successful removal of other entries, the correct series of operations must be maintained. It therefore follows that a fully general solution to this problem may be extremely complicated. The usual "distributed-database" problems are to be solved, in which parts of the database must be frozen while an operation takes place, while at the same time a picture of the database must be maintained through future "alterations", and also a set of dependencies between relations to expire before other relations are introduced must be maintained.

This appears to be a complex problem and, unless database specialists indicate that some type of solution exists, this problem should be solved by providing something less than full generality.

What follows is a solution such that

- it does not preclude any valid set of relationships,
- as a rule will easily accommodate checks, and
- its cost is felt when drastic changes occur in the initial cluster topology.

Proposed solution:

The solution assigns levels to each cluster. Level assignment is a tool used to achieve ease and correctness of operations.

Consider a tree of naming/registration authorities that was created as follows:

(A) At time 0 the tree consists of a single node, the root, that can act as a naming/registration authority. An interval of cluster-levels [0, 2^{32}-1] is associated with this node.

(B) At any given time t an existing node D may acquire a child D*. At that time D will do the following:

(a) Part of the naming space that is under D's authority (i.e. has not been delegated or assigned) will be delegated to D*.

(b) If D is a leaf, then D's cluster-level interval is of the form [M, 2^{32}-1]. This interval will be apportioned between D and D* as [M, N] and [N+1, 2^{32}-1], respectively. The value N is arbitrarily chosen in [M, 2^{32}-2] by D.

If D is not a leaf, then its cluster-level interval is [M, N] with N<2^{32}-1. This interval will be apportioned between D and D* as [M, Z] and [Z+1, N], respectively. The value Z is arbitrarily chosen in [M, N-1] by D.

Note C.1:

Leaf nodes whose cluster-level interval consists of a single value [2^{32}-1] can never acquire children.

(C) The rules under which this distributed naming/registration authority operates are the following:

(1) Each cluster C is attached to a unique naming authority N(C) which will not change during the lifetime of the cluster. The NETs of the ISs in C will also be derived from N(C).

(2) Each cluster C is attached to a unique registration authority R(C). When C is created, R(C)=N(C), but R(C) may change in time.

(3) For each cluster C, N(C) will at all times know the value of R(C).

(4) Each cluster C has at any time an attached level L(C) whose value is within the range of values attached to R(C). This value will be registered with R(C).

(5) A cluster C may enter into a parent/child relationship with a cluster D (i.e. establish up/down links with the C-D direction being down) if and only if L(C)<L(D).

The relationship will be registered at R(C) and R(D).

(6) A cluster C may at any time decrease L(C) and, possibly change registration authority provided that:

(a) L(C) remains bigger than max {L(B)|B is a parent to C}, and

(b) the levels of clusters B as above do not increase while C's is decreasing.

(7) At any given time a cluster C may increase L(C) provided that L(C) remains smaller than min {L(D)|D is C's child} and no child D has initiated the procedure described in (6).

(8) If at any given time C wants to increase L(C) to a value V greater than or equal to min {L(D)|D is a child of C}, then C can initiate a change of levels as follows:

C's children will be asked to change their level to V+1 or more by some time t. Failing this, C will undo the parent/child relationship and will change its level to V. Such an initiation of a change of levels by C

may result in the related clusters themselves initiating a consequent change of levels.

(9) A decrease of L(C) to a value V that is not bigger than max {L(B)|B is a parent to C} will be actively discouraged. Nevertheless, it may be accomplished via a mechanism similar to the one described in (8) above. At the same time, provisions similar to those that appear in (6 (b)) must hold, so that no B, B parent of C will attempt to concurrently increase L(B) to V or more.

(10) All other operations will be handled via a sequence of the operations that are described above. For instance, a reversal of a parent/child relationship (from "C a parent of D" to "D a parent of C") can be handled as follows:

- the original relationship is undone,
- C's level is increased beyond D's,
- the D/C relationship is installed.

The practical effect of rules (1) to (10) is that the handling of relations is simple while the handling of levels is complicated. Therefore, it is expected that most cases can be handled with routine ease. But instances which require major topological changes will necessitate level changes and will be handled through more difficult and time consuming procedures.

APPENDIX D

THE STRUCTURE OF GLOBAL OSI ROUTEING

D.1 CATEGORIES OF ROUTEING

This Appendix summarizes the ISO Routeing Framework (ISO 9575).

As already outlined in Clause 7, the global routeing model is decomposed into three categories of routeing (see Figure D-1).

D.1.1 ES Routeing (Across a Single Subnetwork)

An ES routeing protocol (e.g. ISO/DIS 9542, End System to Intermediate System Routeing Exchange Protocol for use in conjunction with ISO 8473) may operate across individual subnetworks to establish connectivity and reachability between ESs and ISs on that subnetwork. ES routeing may also be used to establish connectivity and reachability within the subnetwork itself in those cases where this is not an inherent service of the subnetwork service provider (e.g. ISO 8802, Local Area Networks (LANs)).

The operation of an ES routeing protocol ensures that each ES knows about at least one IS that is directly reachable on each subnetwork to which the ES is attached, and that each IS knows about every ES reachable on each subnetwork to which the IS is attached.

D.1.2 Intra-domain Routeing (Within a Routeing Domain)

Intra-domain routeing is concerned with communication among ISs that are in the same routeing domain. Each routeing domain is assumed to be under the control of at least one administrative authority which takes responsibility for the assignment of NSAPs and subnetwork addresses, and the way in which the costs of operation are determined and recovered. When a routeing domain is under the control of multiple administrative authorities then such assignments and cost recovery procedures must be coordinated.

D.1.3 Inter-domain Routeing (Among Routeing Domains)

Inter-domain routeing is concerned with managing and controlling the exchange of information between ISs that are not within the same routeing domain. The issues of concern at this level are, for the most part, administrative: security, access control, national regulations, legal and political implications of transborder data flow, and others. The techniques used to accomplish the actual routeing function may be the same as those used at intra-domain routeing; the context in which they are employed is, however, fundamentally different for inter-domain routeing than for intra-domain routeing.

Inter-domain Routeing	IS-IS protocol between ISs which do not belong to the same routeing domain.
Intra-domain Routeing	IS-IS information exchange in the same routeing domain (the information exchanged concerns the common routeing domain)
ES Routeing	ES-IS protocol

Figure D-1 : Different Categories of Routeing

Note D.1:

This Technical Report concerns Inter-domain routeing.

D.2 ROUTEING DOMAINS

The global OSI Environment (OSIE) will of necessity be composed of multiple routeing domains which are under responsibility of different administrations.

A routeing domain is a set of ISs bound by a common routeing procedure, namely:

- use of the same set of routeing metrics,
- use of compatible metric measurement techniques,
- use of the same information distribution protocol, and
- use of the same path computation algorithm.

D.2.1 Formal Definition of a Routeing Domain

A routeing domain D can be defined formally as a couplet *(S, R)* where *S* is the set of ISs in the domain and *R* is the common routeing procedure. It is understood that:

- Every IS within a routeing domain D can determine if a given NSAP is reachable within D; if it is, then the routeing procedure is capable of deriving a path to that NSAP.
- An IS within a domain D has a means of ascertaining if another neighbouring IS participates in D.
- An IS may participate in more than one routeing domain. In such a case:
  - the IS will fully and completely, but independently, participate in the routeing procedures of each domain,
  - routeing information from one routeing domain will not be utilized in any way in the routeing procedures of the other, and
  - when an IS participating in two routeing domains D_a and D_b receives a PDU from an ES, the IS will have to determine in which domain this message will be routed.

Note D.2:

Two distinct routeing domains may use the same routeing procedure R and consist of overlapping sets of ISs.

D.2.2 Hierarchical Structure of Routeing Domains

As the number of ESs and ISs in a routeing domain increases, it becomes more difficult to maintain and process all of the information necessary to perform the routeing functions. Typically, the size of the RIB, the exchange of routeing update information, and the computation of routes may consume more resources than are allocated to route determination in the domain.

In order to reduce the overhead associated with route determination, it is often useful to introduce into a routeing domain a hierarchical structure which allows information to be summarized.

Furthermore, hierarchical structuring would greatly reduce the number of entries in the RIB maintained by a network entity. Typically, if the size of the RIB were m, then it will be of the order of $log\ m$ once the hierarchy is introduced. This reduction of the RIB results in a proportional reduction in the exchange of routeing update information and in turn reduces the load imposed by the computation of routes in the routeing domain.

D.3 ROUTEING PROCEDURES

OSI may adopt different routeing procedures for the different categories of routeing identified in Clause 7 of this Technical Report. The reasons for this are many:

- For ES Routeing, it is desirable to simplify the operation of ESs by offloading routeing functions to the ISs since:
  - the number of ESs is expected to be orders of magnitude greater than the number of ISs,
  - ISs may have more resources allocated to the functions of routeing and relaying, and
  - ESs are less likely to be attached to multiple subnetworks than ISs and hence have fewer routeing choices to make.
- For intra-domain routeing, significant benefits can be obtained from a dynamic routeing scheme which produces optimal routes with acceptable overhead. These routeing procedures may be non-standard, provided that the inter-domain/intra-domain interfaces are standard.
- For inter-domain routeing, the nature of the relationships will restrict the type and detail of routeing information available. More stringent procedures for authenticating and propagating routeing information may also be needed.

In order to analyze the strengths and weaknesses of various routeing procedures, it is useful to have a taxonomy which can be used to select from a number of techniques. Routeing algorithms may be classified according to how they accomplish the aspects of routeing defined in Clause 5, and according to what types of information are used to select routes. The following text provides such a taxonomy.

D.3.1 Static Routeing

In static routeing all routeing information known to a system is loaded into the RIB by System Management. This information is generally in pre-computed form, in that only the paths actually to be used are made available rather than all possible paths. In essence, static routeing performs the Decision Function of Figure E-1 in Appendix E in an off-line fashion and uses System Management protocols to communicate the resulting routeing tables to each system.

Static routeing has the advantage of permitting extremely sophisticated off-line optimization algorithms to be executed, since the route computation need not be done in real-time while PDUs are being relayed. It has the disadvantages of not being capable of «bootstrapping» the NS-providers since there is no information collection or distribution by the network entities themselves. Further, static routeing is not capable of reacting to changes in configuration, topology, or other, in an adaptive fashion since all paths are precomputed.

D.3.2 Quasi-static Routeing

Quasi-static routeing is similar to static routeing in that paths are computed off-line and loaded into the RIB through System Management. Rather than storing a single, highly-optimized path for each routeing metric, however, quasi-static routeing allows for alternate paths to be stored. This reduces the impact of failures by allowing the Forwarding Function to select a backup path if the best path is unavailable.

Quasi-static routeing has similar advantages and disadvantages to those of static routeing. It can, however, adapt to failures in a limited way, at the expense of an increase in the amount of information stored concerning backup paths.

D.3.3 Centralized Routeing

In centralized routeing, network entities report information about their local environment (NSAPs supported, SNPAs present, SNPAs operating, routeing metrics for each outbound path, and other information) to a centralized facility in their routeing domain. The centralized facility accumulates this information and periodically, or upon certain events, computes routes. It then sends this information to each of the systems in its routeing domain, which subsequently use it to forward RNPUs. In essence, the complete RIB only exists at the centralized facility, where the Decision Function is executed. The resulting FIBs are then returned to each network entity for use by the Forwarding Functions.

One way of viewing the operation of a centralized routeing procedure is to model it as a directory service, where the Information Collection Function is analogous to a directory service's update function, and the Information Distribution Function is analogous to a directory service's query function. The directory itself resides at the centralized facility.

A principal advantage of centralized routeing is that a powerful computing engine can be dedicated to determining routes in an optimal fashion. This

central facility can be made resilient against a single-point failure by redundancy and other backup techniques. Centralized routeing can also be relatively responsive to changes in configuration and topology since it uses real-time techniques for the information collection and information distribution aspects of routeing.

Centralized routeing has two substantial drawbacks. Firstly, a way must be found to route the routeing information to and from the centralized facility, since the routes calculated by the centralized facility cannot be used for this purpose. Often, static or quasi-static techniques are used, which limit the ability to respond to failures (failures on a path to or from the centralized facility are difficult to deal with). Further, the delays inherent in the propagation of the information to and from the centralized facility can be substantial and can lead to permanent mis-synchronization between the routes that are calculated and the routes actually in use.

D.3.4 Distributed Adaptive Routeing

In distributed adaptive routeing, systems dynamically sense their local environment, as with centralized routeing. They then exchange this information with other systems directly, using a routeing-specific information distribution protocol. Systems receive this information and store it in their Routeing Information Bases. Periodically, or upon certain events, each system computes new routes from the RIB and produces new FIBs which it uses to forward RNPUs.

Distributed adaptive routeing procedures have the major advantage of extreme robustness and the ability to adapt automatically, and quickly, to changes in configuration (due to failures and due to the addition of new systems and/or subnetworks). Some procedures also have the capability of dynamically adjusting routes based on changes in traffic patterns and congestion. Their most significant disadvantage is the complexity of their design, which can be substantial. A further disadvantage is that the real-time nature of the path computation algorithm often precludes the use of complex routeing metrics, and hence limits the optimality of the paths computed.

Distributed adaptive routeing procedures fall into two general categories, Distance Vector Routeing and Link State Routeing.

D.3.4.1 Distance Vector Routeing

In distance vector routeing, network entities learn about the configuration (topology, routeing metric values, and so on) from neighbouring systems (i.e. network entities which they can reach in a single subnetwork hop). They then compute routes based on this information. If the network entity changes the way it will route RNPUs, it informs its neighbours of the new routes. The precise information sent is a measure of the logical distance to each destination network entity in the routeing domain for each routeing metric in use, rather than the path itself. Neighbour network entities, on receipt of new routeing information, also recompute and, if they alter their routes, they inform their neighbours. The procedure converges when no network entity in a routeing domain changes its routes upon receipt of routeing information from its neighbours.

D.3.4.2 Link State Routeing

In link state routeing, network entities broadcast information about their local environment to all other network entities within the routeing domain. Each system thereby builds up a complete «topological map» of the entire routeing domain. Each network entity then independently computes routes using a graph-theoretic path minimization algorithm, such as Djikstra's SPF.

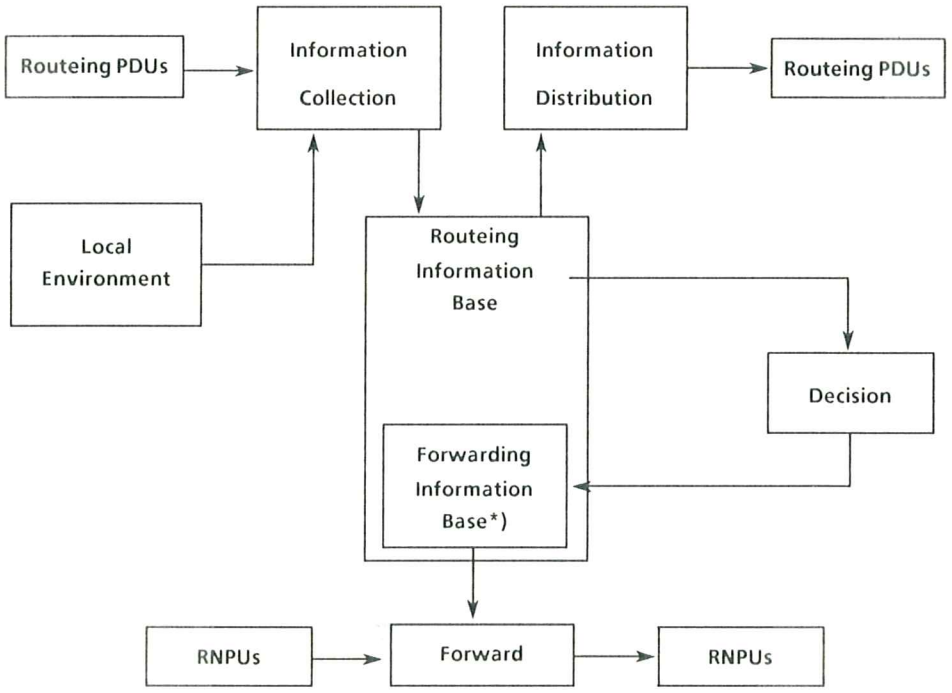
APPENDIX E

DECOMPOSITION OF THE ROUTEING FUNCTION

OSI Routeing can be decomposed into four different but interrelated aspects, which are described below. Figure E-1 illustrates their relationship.

The purpose of the division into these four aspects is to :

- conceptually clarify the functions of routeing,
- simplify the design of routeing protocols by breaking routeing into its component parts, and
- make the routeing functions as flexible as is practical by allowing for degrees of freedom in each aspect.



(*) multiple tables, dependent on certain constraints

Figure E-1 : Decomposition of the Routeing Function

E.1 THE ROUTEING INFORMATION BASE

The RIB comprises the complete information required by a particular ES or IS to accomplish routeing. Such information might include:

- **Next hop routeing tables**

  These are tables which relate destination NSAPs to the potential next subnetwork hops (e.g. local and remote SNPAs) which might be used to forward the PDU closer to the destination.

- **Lists of neighbour ESs and ISs**

  These lists enable an ES or IS to ascertain the local topology.

- **Measured QoS characteristics of a datalink or subnetwork path**

  These measurements allow the routeing functions to adapt to QoS changes.

- **Network maps**

  These are topological graphs of some portion of the global network. Such maps can be used to compute shortest paths to destination NSAPs using any of a number of routeing metrics.

E.2 INFORMATION COLLECTION

These are techniques which an ES or IS may use to build up its RIB. Some examples are : measurement protocols, policy input from System Management, directory lookup functions, and routeing protocols.

E.3 INFORMATION DISTRIBUTION

These are techniques which an ES or IS may use to inform other ESs and ISs of pertinent information in its RIB. Some examples are : routeing protocols and interactions through the Management Information Bases.

E.4 ROUTE CALCULATION AND MAINTENANCE

These are the internal functions executed by ESs and ISs on the RIB to accomplish routeing. The major function in this category is the generation of the FIBs which is used to relay RNPUs. This function is illustrated in Figure E-1 by the box labelled «Decision». Other examples of these internal functions include : timing functions such as ageing old RIB entries, and the functions F1 and F2 described below.

E.4.1 Network Entity Title Selection (Function F1) and SNPA Selection (Function F2)

The functions F1 and F2 are required by every ES and IS to route an RNPU.

The inputs to F1 are:

- the called NSAP address;
- the calling NSAP address;
- a source route (optional);

Note E.1:

A source route is a sequence of network entity titles which identify network relay systems. In a complete source route the next network entity title in the sequence is the output of F1. In a partial source route, the next network entity title in the sequence is used to determine the network entity title of a network relay system used to reach the network relay identified by the source route.

- quality of service (QoS) parameters (optional);
- the FIBs.

For each RNPU that is routed, F1 determines:

- the network entity title of a network relay system on the path to the destination NSAP, or else
- the title of the destination network entity, if no relay function is necessary to reach the destination. The title may be the same as the destination NSAP address.

The inputs to F2 are:

- the network entity title of the network relay or destination ES determined by F1;
- QoS;
- the FIBs.

This function is performed after F1 to determine which subnetwork point of attachment (SNPA) to use when sending an RNPU to the network relay or destination network entity. The information yielded by this function is:

- identification of the selected SNPA, and
- values of parameters which are input to the subnetwork service provider associated with that SNPA.

APPENDIX F

RELATIONSHIP OF ROUTEING TO OSI MANAGEMENT

The routeing function intersects with OSI Management through information stored in, and retrieved from, the MIB. Figure F-1 depicts this relationship. Routeing information is placed in the MIB either through the operation of the Network Layer or through interaction with System Management. Note that in general an IS does not provide the facilities to fulfil System Management functions.

Note F.1:

According to ISO/DIS 7498-4, layer operation is the set of facilities which control and manage a single instance of communication. These facilities can be embedded within an existing «normal» protocol exchange (as opposed to layer management protocol exchange).

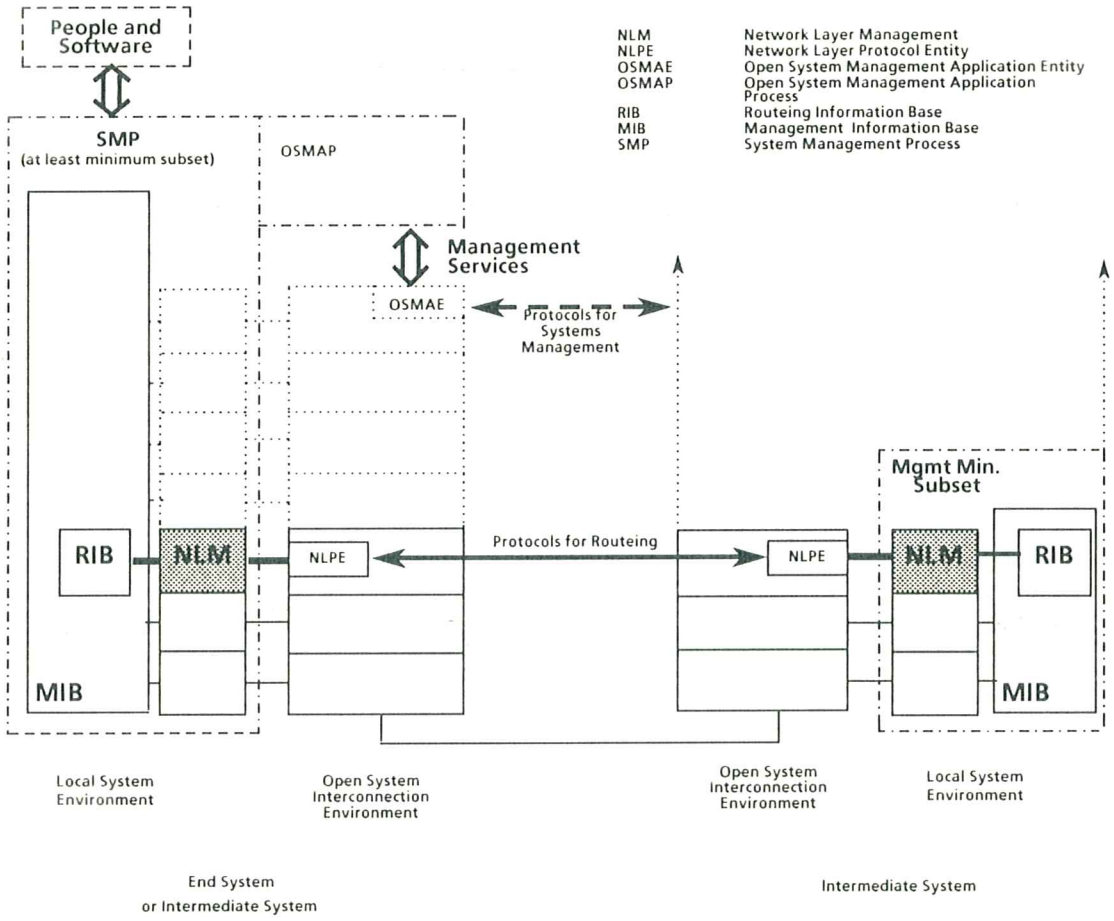


Figure F-1 : Relationship of Routeing to OSI Management

In general, it is desirable to obtain and exchange routeing information through the operation of the Network Layer itself rather than to rely on System Management exchanges at the Application Layer. Confining the generation, exchange, and synchronization of routeing information within the Network Layer keeps routeing a «closed system» and avoids difficult issues in cross-layer coordination.

Operation of the Network Layer, in fulfillment of the rôle assigned to it in the OSI Reference Model, requires shared knowledge concerning the location of NSAPs and routes through the available subnetworks. This information may be distributed automatically by the use of protocols.

These protocols cannot operate in the Application Layer because:

- by their nature they must operate even when the Network Layer is not fully able to offer the Network Service, i.e. when neither the location of NSAPs nor the possible routes to them are globally known;
- they typically use capabilities existing in the lower layers but not available in upper layer services, such as multicast;
- they necessarily operate in terms of subnetwork addresses, to discover where NSAPs are attached to the subnetworks, and cannot use a service which operates in terms of NSAP addresses.

Therefore, at least some aspects of information exchange for Network Layer routeing must take place completely within the Network Layer. Other aspects of this function, such as the distribution of routeing tables or parameters for the construction of routeing tables, may be performed by System Management protocols operating in the Application Layer.

There are, of course, circumstances under which it is desirable to exchange routeing information at the Application Layer through System Management protocols, e.g. routeing parameters and routeing tables. In general, it is likely that a complete and realistic solution to the global routeing problem in the OSIE will require a combination of techniques involving both Network Layer (management) protocols for routeing and the use of System Management protocols.

APPENDIX G

ACRONYMS AND ABBREVIATIONS

AAC	Acquisition Attempt Counter
AAT	Acquisition Attempt Timer
AD	Acquisition Disconnect
AR	Acquisition Request
AS	Acquisition Response
CID	Cluster Identifier
CL	Connectionless Mode
CLNS	Connectionless Mode Network Service
CO	Connection Oriented Mode
CONS	Connection Oriented Mode Network Service
CPU	Central Processor Unit
CRC	Cyclic Redundancy Checking
CT	Closing Timer
DH	Domain Hello
DK	Data Acknowledgement
DNET	Destination Network Entity Title
DTE	Data Terminal Equipment
DU	Data Update
DUC	Data Update Counter
DUT	Data Update Timer
ES	End System
FIB	Forwarding Information Base
HL	Hello
HLT	Hello Timer
HT	Holding Timer
ID	Identifier
IDR	Inter-Domain Routeing
IS	Intermediate System
ISH	Intermediate System Hello
ISO	International Standards Organization
KAT	Keep Alive Time
MIB	Management Information Base
NET	Network Entity Title
NLM	Network Layer Management

NLPE	Network Layer Protocol Entity
NPAI	Network Protocol Address Information
NPID	Network Layer Protocol Identification
NS	Network Service
NSAP	Network Service Access Point
OSI	Open Systems Interconnection
OSIE	OSI Environment
OSMAE	Open System Management Application Entity
OSMAP	Open System Management Application Protocol
PCI	Protocol Control Information
PDU	Protocol Data Unit
PID	Partition Identifier
PVC	Permanent Virtual Circuit
QCR	Query Configuration Request
QCS	Query Configuration Response
QoS	Quality of Service
RD	Redirect
RDI	Routeing Domain Interconnection
RI	Routeing / Configuration Information
RIB	Routeing Information Base
RIE	Routeing Information Exchange
RNPU	Routed Network Protocol Unit
RR	Restart Request
RRC	Restart Request Counter
RRT	Restart Request Timer
RS	Restart Response
SMP	System Management Protocol
SN	Subnetwork
SNET	Source Network Entity Title
SNPA	Subnetwork Point of Attachment
SNSDU	Subnetwork Service Data Unit
SPF	Shortest Path First
SVC	Switched Virtual Circuit
TGT	Table Generation Tag
VC	Virtual Call
V/P	Version/Protocol
WT	Waiting Timer