International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.800

(02/2012)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

Digital networks – General aspects

Unified functional architecture of transport networks

Recommendation ITU-T G.800

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TRANSMISSION MEDIA AND OPTICAL SYSTEMS CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
General aspects	**G.800–G.809**
Design objectives for digital networks	G.810–G.819
Quality and availability targets	G.820–G.829
Network capabilities and functions	G.830–G.839
SDH network characteristics	G.840–G.849
Management of transport network	G.850–G.859
SDH radio and satellite systems integration	G.860–G.869
Optical transport networks	G.870–G.879
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
MULTIMEDIA QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DATA OVER TRANSPORT – GENERIC ASPECTS	G.7000–G.7999
PACKET OVER TRANSPORT ASPECTS	G.8000–G.8999
ACCESS NETWORKS	G.9000–G.9999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T G.800

Unified functional architecture of transport networks

Summary

Recommendation ITU-T G.800 describes a unified functional architecture for transport networks that use connection-oriented circuit switching (CO-CS), connection-oriented packet switching (CO-PS), and connectionless packet-switching (CL-PS) in a technology-independent way.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T G.800	2007-09-06	15
1.1	ITU-T G.800 (2007) Amd. 1	2009-03-09	15
1.2	ITU-T G.800 (2007) Amd. 2	2010-09-06	15
2.0	ITU-T G.800	2012-02-13	15

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Page

Recommendation ITU-T G.800

Unified functional architecture of transport networks

1 Scope

A telecommunication network is a complex network which can be described in a number of different ways depending on the particular purpose of the description. This Recommendation describes the network as a transport network from the viewpoint of the information transfer capability.

This Recommendation provides a set of constructs (definitions and diagrammatic symbols) and the semantics that can be used to describe such a viewpoint.

A transport network transfers user information from a sender at one location to a receiver at another location. A transport network can also transfer various kinds of network control information such as signalling, and operations and maintenance information for the control functional group.

This Recommendation describes the functional architecture of transport networks in a technology-independent way. The generic functional architecture of transport networks provides the basis for a harmonized set of functional architecture Recommendations for specific layer network technologies, including those that use connection-oriented circuit switching (commonly referred to as CO-CS) or connection-oriented packet switching (commonly referred to as CO-PS), and connectionless packet switching (commonly referred to as CL-PS), and a corresponding set of Recommendations for management, performance analysis, and equipment specifications.

This Recommendation provides the definition of the architectural constructs that are required to describe a transport network. The theoretical material that was used as the basis for these definitions is provided in Annex A (concepts), Annex B (properties of systems), and Annex C (properties of communications).

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T G.805] Recommendation ITU-T G.805 (2000), *Generic functional architecture of transport networks*.

[ITU-T G.806] Recommendation ITU-T G.806 (2012), *Characteristics of transport equipment – Description methodology and generic functionality*.

[ITU-T G.8080] Recommendation ITU-T G.8080/Y.1304 (2006), *Architecture for the automatically switched optical network*.

[ITU-T X.200] Recommendation ITU-T X.200 (1994), *Information technology – Open Systems Interconnection – Basic Reference Model: The basic model*.

3 Definitions

This Recommendation uses the following terms defined elsewhere:

3.1 Terms imported from ITU-T G.805

This Recommendation uses the following terms defined in [ITU-T G.805]. The definitions are given here for the readers' convenience.

3.1.1 access group: A group of co-located "trail termination" functions that are connected to the same "subnetwork" or "link".

3.1.2 access point (AP): A "reference point" that consists of the pair of co-located "unidirectional access" points, and therefore represents the binding between the trail termination and adaptation functions.

3.1.3 adaptation: A "transport processing function" that consists of a co-located adaptation source and sink pair.

3.1.4 adaptation sink: A "transport processing function" which presents the client layer network characteristic information at its output by processing the information presented at its input by the server layer network trail.

3.1.5 adaptation source: A "transport processing function" which accepts client layer network characteristic information at its input and processes it to allow transfer over a trail (in the server layer network).

3.1.6 adapted information (AI): A signal which is transferred on "trails". The specific formats will be defined in the technology specific Recommendations.

3.1.7 architectural component: Any item used in this Recommendation to generically describe transport network functionality.

3.1.8 binding: A direct relationship between a "transport processing function" or "transport entity" and another "transport processing function" or "transport entity" which represents the static connectivity that cannot be directly modified by management action.

3.1.9 characteristic information (CI): A signal with a specific format, which is transferred on "network connections". The specific formats will be defined in the technology specific Recommendations.

3.1.10 client/server relationship: The association between layer networks that is performed by an "adaptation" function to allow the link connection in the client layer network to be supported by a trail in the server layer network.

3.1.11 connection: A "transport entity" which consists of an associated pair of "unidirectional connections" capable of simultaneously transferring information in opposite directions between their respective inputs and outputs.

3.1.12 connection point (CP): A "reference point" that consists of a pair of co-located "unidirectional connection points" and therefore represents the binding of two paired bidirectional "connections".

3.1.13 layer network: A "topological component" that represents the complete set of access groups of the same type which may be associated for the purpose of transferring information.

3.1.14 link: A "topological component" which describes a fixed relationship between a "subnetwork" or "access group" and another "subnetwork" or "access group".

3.1.15 matrix: It represents the limit to the recursive partitioning of a subnetwork.

3.1.16 matrix connection: A "transport entity" that transfers information across a matrix, it is formed by the association of "ports" on the boundary of the matrix.

3.1.17 network: All of the entities (such as equipment, plant, facilities) which together provide communication services.

3.1.18 network connection: A transport entity formed by a series of contiguous "link connections" and/or "subnetwork connections" between "termination connection points".

3.1.19 port: It consists of a pair of unidirectional ports.

3.1.20 reference point: An architectural component, which is formed by the binding between inputs and outputs of transport processing functions and/or transport entities.

3.1.21 sublayer: A set of additional transport processing functions and reference points encapsulated within a layer network. It is created by decomposition of transport processing functions or reference points.

3.1.22 subnetwork: A topological component used to effect routing of a specific characteristic information.

3.1.23 subnetwork connection: A "transport entity" that transfers information across a subnetwork, it is formed by the association of "ports" on the boundary of the subnetwork.

3.1.24 topological component: An architectural component, used to describe the transport network in terms of the topological relationships between sets of points within the same layer network.

3.1.25 trail: A "transport entity" which consists of an associated pair of "unidirectional trails" capable of simultaneously transferring information in opposite directions between their respective inputs and outputs.

3.1.26 trail termination: A "transport processing function" that consists of a co-located trail termination source and sink pair.

3.1.27 trail termination sink: A "transport processing function" which accepts the characteristic information of the layer network at its input, removes the information related to "trail" monitoring and presents the remaining information at its output.

3.1.28 trail termination source: A "transport processing function" which accepts adapted "characteristic information" from a client layer network at its input, adds information to allow the "trail" to be monitored and presents the characteristic information of the layer network at its output. The trail termination source can operate without an input from a client layer network.

3.1.29 transport: The functional process of transferring information between different locations.

3.1.30 transport network: The functional resources of the network which conveys user information between locations.

3.1.31 transport processing function: An architectural component defined by the information processing which is performed between its inputs and outputs. Either the input or output must be inside a layer network; the corresponding output or input may be in the Management Network (e.g., output of a monitor function).

3.1.32 unidirectional access point: A "reference point" where the output of a "trail termination sink" is bound to the input of an "adaptation" sink or the output of an "adaptation" source function is bound to an input of a "trail termination source".

3.1.33 unidirectional connection: A "transport entity" which transfers information transparently from input to output.

3.1.34 unidirectional connection point: A "reference point" that represents the binding of the output of a "unidirectional connection" to the input of another "unidirectional connection".

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 access transport entity: A transport entity responsible for the transfer of information from the access ports at the inputs of a set of termination sources to the access ports at the outputs of a set of termination sinks. The integrity of the information transfer may be monitored. It is formed by combining a set of termination functions and a network transport entity.

3.2.2 aggregation: Combining several instances of communications onto a single transport entity, with sufficient labelling to distinguish and later separate them, without providing full client/server layer network independence (i.e., without full information independence) or creating a new communication.

3.2.3 communication: A body of information produced by a sender and intended, in its entirety, to reach a particular receiver or set of receivers.

3.2.4 forwarding function: A transport processing function that supports a transport entity in a subnetwork.

3.2.5 forwarding point (FP): The binding of an output forwarding port and an input forwarding port.

3.2.6 forwarding port (FPt): An input or output of a transport entity or layer processor function, input of an adaptation source function or termination sink function, or output of an adaptation sink function or termination source function. The forwarding port on a transport entity is coincident with (corresponds to) a forwarding port on the transport processing function that directly supports that transport entity.

3.2.7 information system: A system that processes only information.

3.2.8 layer processor function: A transport processing function that accepts layer network characteristic information at its input forwarding port and delivers layer network characteristic information at its output forwarding port and provides specific transport functionality by reading, modifying or inserting layer information.

3.2.9 link connection: A transport entity that exists within a link, with a single forwarding rule that transfers information present at the input forwarding port to the output forwarding port.

3.2.10 multiplexing: Combining instances of client layer communications onto a server layer access transport entity, with sufficient labelling to distinguish and later separate them, providing full client/server layer network independence (i.e., full information independence). This creates a new communication.

3.2.11 network transport entity: A transport entity formed by binding a set of subnetwork transport entities and/or link connections to provide connectivity between a set of forwarding end points at the output of a set of termination sources to the forwarding end points at the input of a set of termination sinks.

3.2.12 server subnetwork: A topological component that describes a fixed relationship among a set of two or more subnetworks and/or access groups.

3.2.13 subnetwork transport entity: A transport entity that exists within a subnetwork (examples of a subnetwork transport entity are subnetwork connection protected subnetwork connection, etc.).

3.2.14 transitional link: A topological component that consists of the link port at the edge of one subnetwork and a corresponding link port at the edge of another subnetwork that operates on different instances of characteristic information or whose characteristic information is the same but with different Layer Information. A transitional link (topological component) is supported by or implemented by layer processors and/or adaptation/termination functions (transport processing functions). A transitional link can be partitioned into parallel transitional links, or a concatenation of transitional links. It can also be partitioned into a concatenation of transitional links and zero or more links.

3.2.15 transport entity (TE): An architectural component that exists within a topological component (link, subnetwork or layer network) which transfers information between its input ports and output ports. The information transfer is controlled by forwarding rules. Forwarding between an input port and one or more output ports is controlled by one or more forwarding rules. A transport entity contains one or more forwarding rules.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AI	Adapted Information
AP	Access Point
APt	Access Port
ATM	Asynchronous Transfer Mode
BIP	Bit Interleaved Parity
CC	Connectivity Check
CI	Characteristic Information
CRC	Cyclic Redundancy Check
CL-PS	Connectionless Packet-Switching
CO-CS	Connection-Oriented Circuit Switching
CO-PS	Connection-Oriented Packet Switching
CP	Connection Point
ENNI	External Network to Network Interface
FEC	Forward Error Correction
FP	Forwarding Point
FPt	Forwarding Port
ITE	Interoperation Transport Entity
LC	Link Connection
LI	Layer Information
LPt	Link Port
MP	Maintenance Point
MP-MP	Multipoint to Multipoint
MPt	Maintenance Port
NI	Network Interface
NNI	Network to Network Interface
NTE	Network Transport Entity
ODU	Optical channel Data Unit
OAM	Operations, Administration and Maintenance
P-MP	Point to MultiPoint

QoS	Quality of Service
RCSI	Remote Customer Service Interface
RMP	Rooted Multipoint
SDH	Synchronous Digital Hierarchy
SSN	Server Subnetwork
SSNTE	Server Subnetwork Transport Entity
S-VLAN	Service Virtual Local Area Network
TE	Transport Entity
TTL	Time To Live
UML	Unified Modelling Language
UNI	User to Network Interface
VC	Virtual Channel
VCI	Virtual Channel Identification
VP	Virtual Path
VPI	Virtual Path Identification
VPN	Virtual Private Network
VS	Virtual Section
WDM	Wavelength Division Multiplexing

5 Conventions

A number of diagrammatic conventions have been developed to support the descriptions that follow and these are illustrated in Figures 5-1 to 5-3.

A number of terminological conventions have been developed as follows:

Input port: The port, viewed from inside the boundary of an information system, at which information to be processed enters the system.

Output port: The port, viewed from inside the boundary of an information system, at which information that has been processed leaves the system.

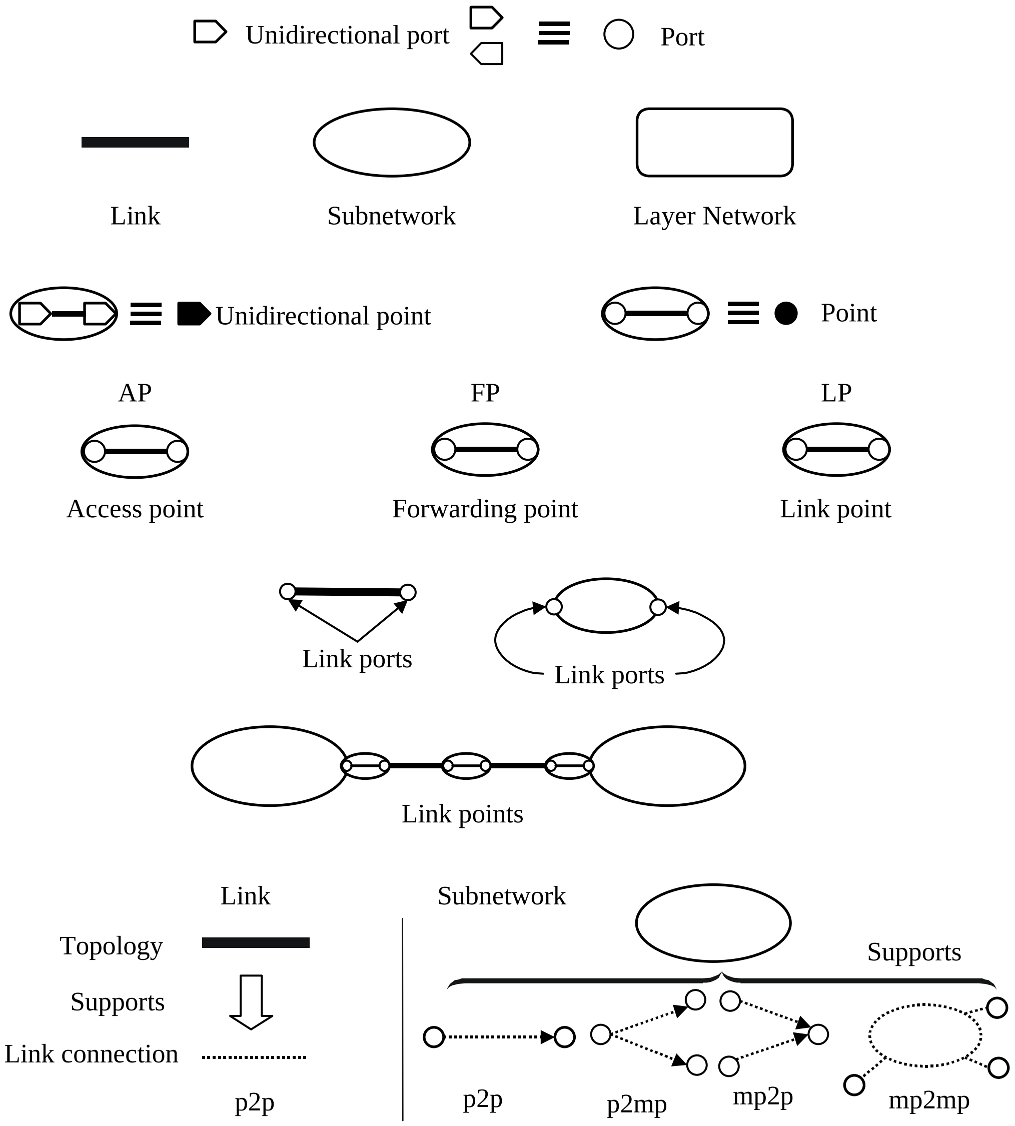
Receiver: The role of an information system that consumes information from another information system.

Sender: The role of an information system that originates information to be processed by another information system.

Sink: The port, viewed from outside the boundary of an information system that accepts information to be processed.

Source: The port, viewed from outside the boundary of an information system, from which information is emitted.

NOTE – Ingress and Egress are synonymous with Input and Output and may be used interchangeably.



point-to-point (p2p), point-to-multipoint (p2mp),

multipoint-to-point (mp2p), multipoint-to-multipoint (mp2mp

Figure 5-1 – Diagrammatic conventions

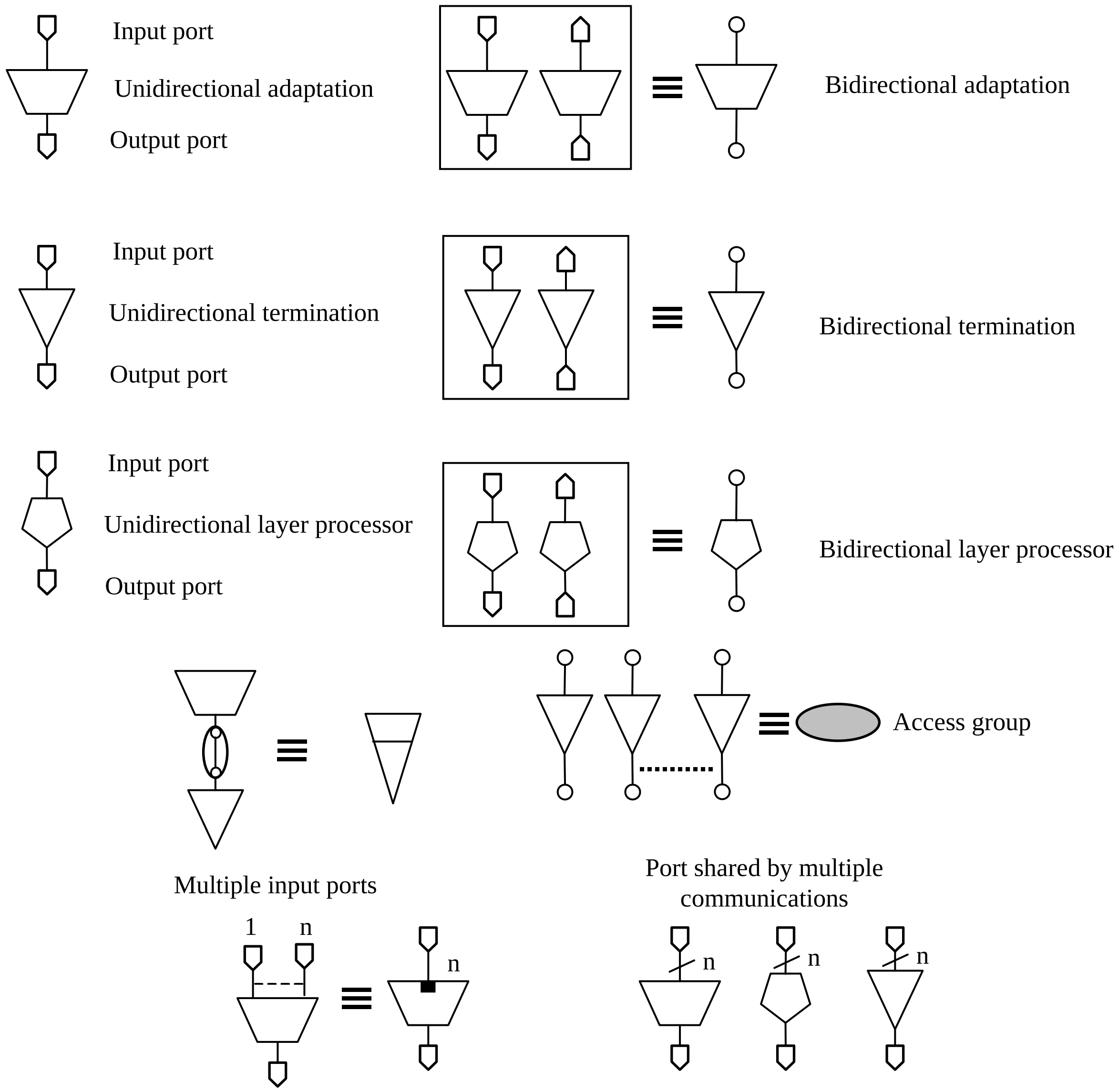

Input port
Unidirectional adaptation
Output port
Bidirectional adaptation
Input port
Unidirectional termination
Output port
Bidirectional termination
Input port
Unidirectional layer processor
Output port
Bidirectional layer processor
Access group
Multiple input ports
1
n
n
Port shared by multiple communications
n
n
n

Figure 5-2 – Further diagrammatic conventions

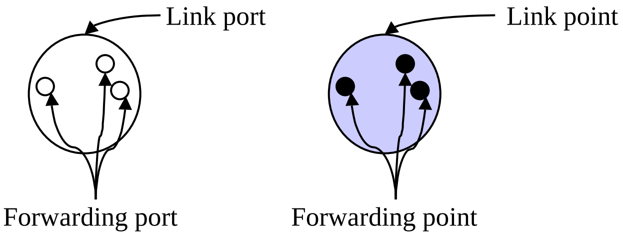


Figure 5-3 – Further diagrammatic conventions

6 Functional architecture of transport networks

The various functions which constitute a telecommunications network can be classified into two broad functional groups. One is the transport functional group which transfers any telecommunications information from one point to another point(s) and this forms the transport plane. The other is the control functional group which realizes various ancillary services and operations and maintenance functions. This Recommendation is concerned with the transport functional group.

A transport network transfers user information from a sender at one location to a receiver at another location. A transport network can also transfer various kinds of network control information such as signalling, and operations and maintenance information for the control functional group. Since the transport network is a large, complex network with various components, an appropriate network model with well-defined functional entities is essential for its design and management. The transport network can be described by defining the associations between points in the network. In order to simplify the description, a transport network model, based on the concepts of layering and partitioning within each layer network, is used in a manner which allows a high degree of recursion. A layer network describes the generation, transport, and termination of a particular characteristic information.

The layer networks which have been identified in the transport network functional model should not be confused with the layers of the OSI Reference Model [ITU-T X.200]. An OSI layer offers a specific service using one protocol among different protocols. On the contrary, each layer network (in this Recommendation) offers the same service using a specific protocol (the characteristic information). It is recommended that this method be used for describing the transport network.

The transport network has been analysed and generic functionality, which is independent of implementation technology, has been identified. This provides a means to describe network functionality in an abstract way in terms of a small number of architectural components. These are defined by the functions that they perform, in information processing terms, and by the relationships they describe between other architectural components. In general, these functions act on information presented at one or more inputs and present information at one or more outputs. They are defined and characterized by the information process between their inputs and outputs. The architectural components are associated together in particular ways to form the network elements from which real networks are constructed. The reference points of the transport network architecture are the result of binding the inputs and outputs of processing functions and transport entities.

This Recommendation describes a transport network as a set of interconnected systems. A full description of the properties of such systems is provided in Annex B. The relationship between the architectural entities described in this Recommendation and those described in [ITU-T G.805] is provided in Appendix II. Considerations on the complexity and scalability of these systems are provided in Appendix III.

6.1 Topological components

The topological components provide the most abstract description of a network in terms of the topological relationship between sets of like reference points. Four topological components have been distinguished: these are the layer network, the subnetwork, the link, and the access group. Using these components, it is possible to completely describe the logical topology of a layer network.

6.1.1 Layer network

A layer network is a topological component that represents the finite non-empty set of access groups of the same type which may be associated for the purpose of transferring information. The structures within and between layer networks are described by the components defined below.

6.1.2 Subnetwork

A subnetwork exists within a single layer network. It is defined by the set of link ports which are available for the purpose of transferring characteristic information. It represents a point of flexibility where relationships between the forwarding ports (within the link ports) at the edge of a subnetwork may be created and broken. These relationships allow characteristic information to be transferred across the subnetwork. In general, subnetworks may be partitioned into smaller subnetworks interconnected by links; this is described in clause 6.7.

6.1.3 Link

A link consists of a link port at the edge of one subnetwork or access group and a corresponding link port at the edge of another subnetwork or access group that are associated for the purpose of transferring characteristic information. The link represents the topological relationship between a pair of subnetworks. Multiple links may exist between any pair of subnetworks. A link may contain zero or more link connections. Link connections are functionally supported by adaptation to a server layer access transport entity (and ultimately to a physical resource). The potential for creating link connections represents the transport capacity available to the link.

6.1.4 Access group

An access group is a group of co-located termination functions. It is bounded by a link port (LPt) that contains the individual forwarding ports (FPts) and the set of individual access ports (APt) of each of the termination functions. When the link port is bound to a subnetwork or link, it forms a link point.

6.2 Derived network constructs for topological viewpoints

This subclause describes a number of constructs that can be used when providing a topological viewpoint for purposes such as routing.

One of two topology views may be used to show the relationship between a set of link points in a layer network (client layer) that are, or may be, related by transport provided by a server layer network.

1) The first view would include at least the set of link points of interest and indicate the set relationship among these points. If the server layer transport has been provisioned (and is presumed fixed) attributes related to the communication capacity and behaviour of the server layer transport may be indicated in addition to the set relationship. The simplest form of this view is as a single subnetwork (simple set relation).

   NOTE – If further partitioning of the topological relationship between the link points of interest is provided, any additional subnetworks or links provided in this (client layer) topology serve to represent attributes related to the communication capacity and behaviour of the server layer transport. They need not have any direct relation to the topology of the server layer network. Link points added in a further decomposed topology do not represent actual forwarding point sets in the client layer (as this partitioned topology is actually representing server layer transport).

2) The second view would include the set of link points of interest and indicate the topological relationship between these points by showing a transitional link from each client layer link point to a server layer link point at the boundary of a server layer topology. This server layer topology may be as simple as a single subnetwork or may be further partitioned (as may be done with any layer network topology).

6.2.1 Transitional link

A transitional link consists of the link port at the edge of one subnetwork and a corresponding link port at the edge of another subnetwork that operates on different instances of characteristic information, or whose characteristic information is the same but with different layer information (LI). A transitional link (topological component) is supported by, or implemented by, layer processors and/or adaptation/termination functions (transport processing functions). A transitional link can be partitioned into parallel transitional links, or a concatenation of transitional links. It can also be partitioned into a concatenation of transitional links and zero or more links.

The use of a transitional link implies a sequence of adaptations between layers, or a sequence of layer processors between sublayers. Any directed ingress transformations must be reversed by corresponding egress transformations.

An example is illustrated in Figure 6-1 below.

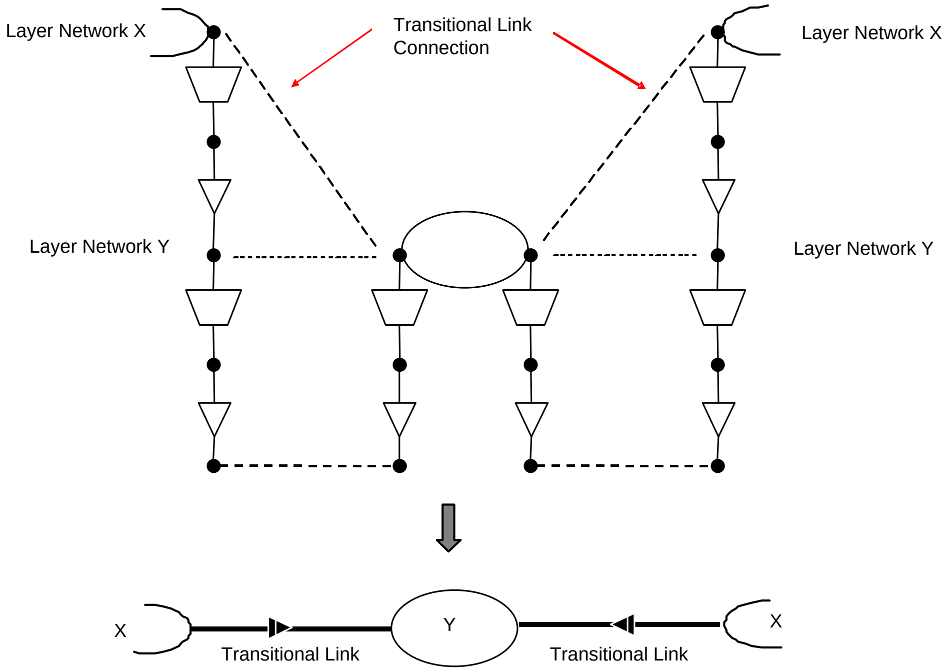


Figure 6-1 – Transitional link between layer networks X and Y

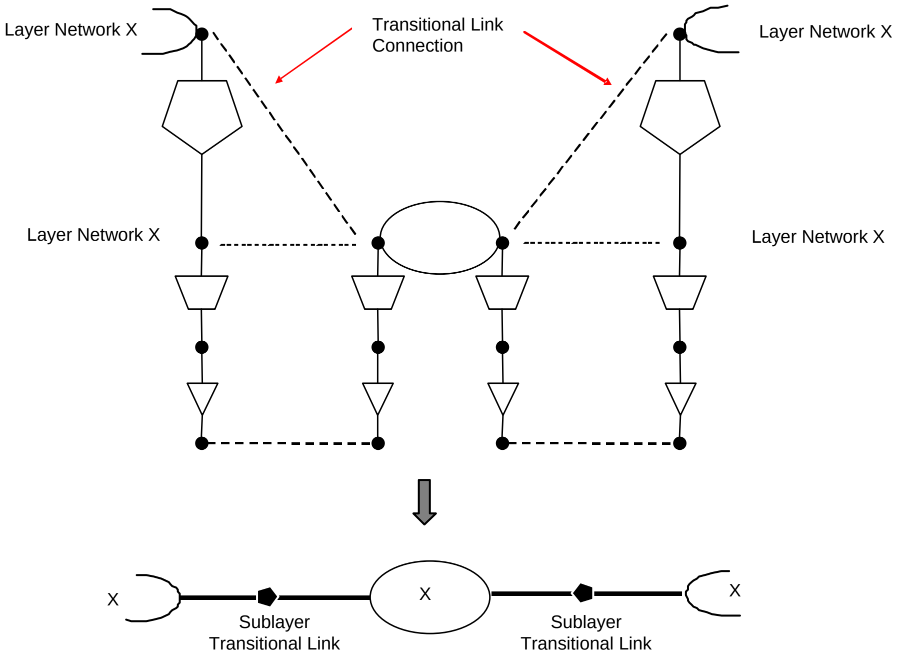

Layer Network X
Transitional Link Connection
Layer Network X
Layer Network X
Layer Network X
X
Sublayer Transitional Link
X
Sublayer Transitional Link
X

Figure 6-2 – Transitional link between sublayers

Figure 6-1 shows a transitional link pair between layer X and layer Y. That the X and Y subnetworks are remote from each other is shown by the Y layer server trail. Note that the ingress Y/X adaptation is undone by the egress Y/X adaptation. Figure 6-2 shows the same situation but within the same layer. In this case, the X and Y subnetworks refer to sublayers.

Figure 6-3 shows the diagrammatic convention for a transitional link between independent layer networks. Figure 6-4 shows the diagrammatic conventions for transitional link between sublayers within a layer network.

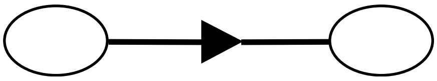


Figure 6-3 – Symbol for layer transitional link

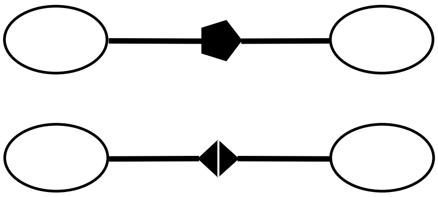


Figure 6-4 – Symbols for sublayer transitional links

The upper diagram is used to depict a directed transformation, which needs to be undone within a trail. An example is inserting a sublayer operations, administration and maintenance (OAM) field. The lower diagram is used to depict a non-directed operation, such as setting a parameter value.

6.2.2 Server subnetwork

A server subnetwork (SSN) consists of two or more link ports that are associated for the purpose of transferring characteristic information. A server subnetwork represents a topological relationship among multiple subnetworks. A subnetwork may be bound to multiple SSNs. As illustrated in Figure 6-5a, a server subnetwork is rendered using the same line weight as a link.

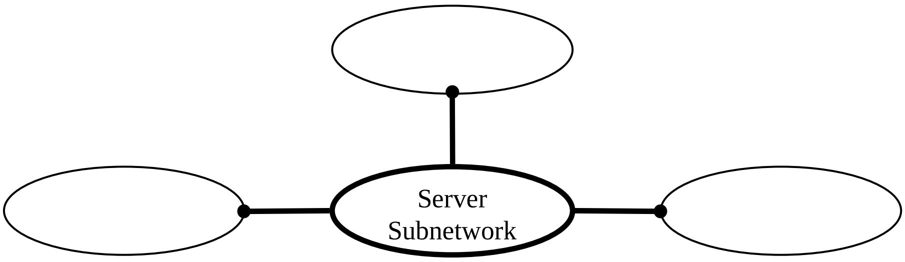


Figure 6-5a – Client layer view

A server subnetwork may contain zero or more server subnetwork transport entities (SSNTEs). SSNTEs are functionally supported by adaptation to a server layer access transport entity (and ultimately to a physical resource). The potential for creating SSNTEs represents the transport capacity available to the SSN.

A server subnetwork cannot be further partitioned within the client layer network and it is not a matrix. There is no unresolved flexibility in the server layer represented by the SSN since the SSN represents a fixed relationship between link points. There may be flexibility in the creation of SSNTEs arising from flexibility in the configuration of adaptation functions at the boundary of the server access transport entity supporting the SSN.

If there is unresolved flexibility in the server layer network, a multi-layer topology can be shown using transitional links between the link ports in the client layer and link ports in the server layer (as described in clause 6.2.1 above). This is illustrated in Figure 6.5b. In this case, the server layer topology may be further partitioned as described in clause 6.7.1.

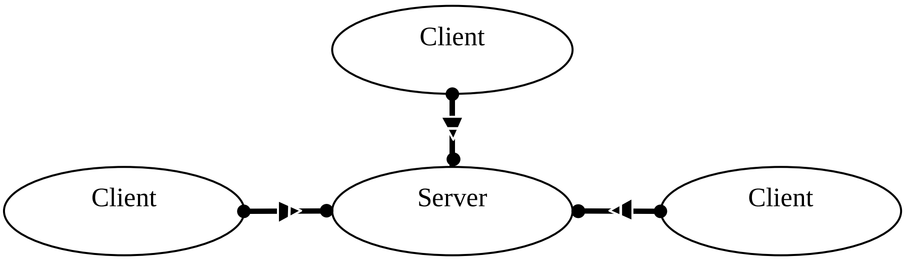


Figure 6-5b – Multi-layer view

6.3 Transport entities and their properties

Transport entities provide the means to transfer information across the network, between reference points, and are configured within topological components. A transport entity is a generalized channel with a number of ports that transfers information from its input port(s) to its output port(s). The information transferred is characteristic of the layer network and specific resources are assigned to accomplish the information transfer. The information transfer relationship provided by a transport entity is distinct from that of other transport entities. The properties of a topological component are unchanged by the addition or configuration of a transport entity. The transport entity can only further restrict any topological or resource constraints that are initially present.

A transport entity exists independently of the information/symbols it transfers. To create a transport entity the set of forwarding points associated with the transport entity must be identified and a set of forwarding rules must be defined. At least one forwarding rule must be present to enable a transport entity to transfer information. For some transport entities forwarding rules may be dynamically added or modified to further control the transfer of symbols, for example to optimize the use of resources or to respond to faults in the network.

6.3.1 Forwarding

Information transfer is the net behaviour resulting from the forwarding rules associated with a transport entity's ingress forwarding points and egress forwarding points, with any policy related to these points. The selection of the ingress and egress points to be used may be based on a policy that considers, for example, the condition of a link connected to the subnetwork for the purpose of protection switching. The creation of a transport entity and associated forwarding rules binds the forwarding ports on the subnetwork to the forwarding ports on a link, server subnetwork, or access group. This binding creates the forwarding points (FP). Two types of forwarding are possible:

Destination forwarding: Symbols presented at an ingress forwarding port are selectively forwarded to zero or more egress forwarding ports. The forwarding function requires control information to identify the output port(s) to which a symbol is destined. This control information is carried by the symbol being forwarded (commonly in the form of a destination address). The resulting network behaviour is traditionally known as "connectionless".

Channel forwarding: All symbols on all ingress forwarding ports are forwarded to all egress forwarding ports. No additional control information is required with the symbol. When there is a single ingress forwarding port the transport entity behaviour is equivalent to a subnetwork connection in [ITU-T G.805].

6.3.2 Subnetwork transport entity

A subnetwork transport entity (SNTE) is a transport entity created in a subnetwork by the configuration of the forwarding function represented by that subnetwork. The SNTE may provide channel forwarding or destination forwarding behaviour according to the capabilities of the forwarding function represented by the subnetwork. If the SNTE provides destination forwarding behaviour, the symbols (CI) (characteristic information) of the layer network must contain destination selection information (e.g., destination address). A network transport entity is bounded by forwarding points (FPs), i.e., it is an SNTE in the maximal subnetwork of a layer network. When the SNTE uses channel forwarding and there is a single ingress forwarding port, the transport entity is a subnetwork connection as defined in [ITU-T G.805].

6.3.3 Server subnetwork transport entity

A server subnetwork transport entity (SSNTE) is a transport entity created by the configuration of adaptation functions bounding a server layer access transport entity. The SSNTE may provide channel forwarding or destination forwarding behaviour. If the SSNTE provides destination forwarding behaviour, additional information to select destination server layer access port(s) must accompany symbols passed to the SSNTE.

6.3.4 Link connection

A link connection is the transport entity that is created when a forwarding function is configured in a link. The link connection has exactly one ingress forwarding port and one egress forwarding port. Any symbol presented at the ingress is delivered to the egress port (i.e., it uses channel forwarding). Resources are reserved for that link connection and there is no possibility of further reservation of capacity. This is a link connection as defined in [ITU-T G.805]. A bidirectional link connection is a pair of link connections in opposite directions in the same bidirectional link. The link connection can be created in the link either before the FPt is bound to another FPt or at the time the binding is created. Changing the binding can only create or delete a link connection; it cannot modify an existing link connection.

6.3.5 Connection

A connection is a channel forwarding transport entity with the added constraint that all the link connection resources have been reserved for a specific communication. A connection has only one ingress forwarding port. Further, the user of a connection has complete control over the allocation of the capacity of the connection. The allocation is only controlled locally, so the local allocation decisions can be instantaneous, deterministic and flexible. A bidirectional connection is a pair of unidirectional connections between the same bidirectional FPs (in opposite directions). A unidirectional connection may have multiple egress forwarding ports, each of which receives in principle the same communication. This construct is called a point-to-multipoint (P-MP) connection. A network connection is a connection that has an FP at each end.

6.3.6 Differentiated connection

A differentiated connection is a transport entity that transfers information belonging to multiple communications between ports across a subnetwork. A differentiated link connection is a special case of a differentiated connection that exists in the context of a link. In a differentiated connection message, contents are interpreted to identify (sets of) communications which receive different treatment. The sets of communications may be distinguished by the forwarding identifier or other layer information. Order is not necessarily preserved between messages belonging to sets of communications receiving different treatment. Sets of communications may be identified for purposes such as traffic conditioning or preserving communication message order.

6.3.7 Multipoint transport entities

A multipoint transport entity is a subnetwork transport entity or server subnetwork transport entity that provides destination forwarding behaviour. Two specific forms of multipoint transport entity are multipoint to multipoint (MP-MP) and rooted multipoint (RMP). An MP-MP transport entity provides bidirectional transfer of symbols among a set of bidirectional forwarding ports. An RMP transport entity provides bidirectional transfer of symbols among a set of root forwarding points and leaf forwarding points, where a root can exchange symbols with any other root or leaf and a leaf can exchange symbols with any root but not with another leaf.

6.3.8 Access transport entity

An access transport entity is a transport entity provided between a set of access ports at the boundary of a layer network. An access transport entity cannot be partitioned. The access transport entity may be established either before or after the termination is bound to an adaptation, i.e., it may be bounded by access ports or access points, or a combination. Modifications to the bindings do not change the access transport entity. The access transport entity is supported by a corresponding subnetwork transport entity in the largest subnetwork (i.e., a network transport entity).

In a network that uses channel forwarding, the access transport entity is supported by a network connection, i.e., it is equivalent to a trail in [ITU-T G.805]. The access transport entities provided by such a network are channel based.

In a network that uses destination-based forwarding, the access transport entity is supported by corresponding destination forwarding in the largest subnetwork. The access transport entity provided by such a network may be channel- or destination-based.

6.3.9 Trails

A trail represents the monitored transfer of information between access points at the boundary of a layer network or between sublayer access points within a layer network. It is delimited by two access points, one at each end of the trail. It represents the association between the ends of the trail. The monitoring provided may verify any combination of:

- layer information in received traffic units;
- the continuity of the intervening transport entity;
- the connectivity of the intervening transport entity; and
- the integrity of the information transferred.

It is possible that no monitoring is provided.

6.3.10 Transitional link connection

A transitional link connection is the transport entity that is created when a forwarding function is configured in a transitional link. The transitional link connection has exactly one ingress forwarding port and one egress forwarding port. Any symbol presented at the ingress is delivered to the egress port with a change in CI type due to an adaptation and termination process or a change in layer information (LI) value due to layer processors. A bidirectional transitional link connection is a pair of transitional link connections in opposite directions in the same bidirectional transitional link.

6.3.11 Partitioning of transport entities

A network transport entity (bounded by termination functions) may be partitioned into parts (smaller transport entities) controlled by different administrations, and within an administrative domain, into parts supported by different resources (links or matrices). Each transport entity has a topology defined by its forwarding points and the transport entities provided between these points. When additional functionality (e.g., tandem monitoring, subnetwork protection, etc.) is provided for a portion of a transport entity, this creates additional forwarding points, adding to the topology of the transport entity. In some cases, only a subset of the parts of a transport entity are used to transfer client traffic at any given time.

When a subnetwork that contains a connection is partitioned to reveal the internal structure of the connection, the subnetworks contain only subnetwork connections and the links or server subnetworks contain link connections.

When a subnetwork that contains a destination forwarding SNTE is partitioned to reveal the internal structure of the SNTE, the subnetworks contain destination or channel forwarding SNTEs and the links or server subnetworks contain link connections or SSNTEs.

6.3.12 Reservation and allocation

In the network, resources are represented by links and these resources can be reserved for a particular transport entity, or a set of transport entities, for supporting a particular communication (a connection) or set of communications. A portion of a resource is reserved for (or assigned to) a transport entity when the transport entity is created. A resource is allocated to a communication only when the communication is using the resource. Resources are limited by the installed capacity and allocations must be within this capacity.

In networks using packet switching, a resource is only allocated when a symbol is present. Therefore, it is possible that the total of the reservations exceeds the capacity of the link. This overbooking may cause link congestion, in which case some symbols may suffer increased delay and in extreme cases may be discarded. Note that policing functions are derived from contracts and are not part of resource reservation. However, they may be used to ensure that the resource allocated does not exceed the reservation. Communications transported by destination forwarding may be subjected to policies even when there is no explicit resource reservation in the network.

In networks using circuit switching, the resource allocation takes place at the time the reservation is made, i.e., when the transport entity is provisioned, and the allocated resource is used even in the absence of any meaningful communication.

6.4 Transport information entities

Transport information entities are entities that are constructed by a network to convey a communication between a sender and receivers. They are formed by the combination of client information with appropriate labels and equivocation overhead. They are the information entities of the transport plane.

Transport information entities, being themselves instances of information, exist separately as three forms of information:

a) *Client information*: This is the communication that the client requires to be transported transparently and accurately.

b) *Adapted information (AI)*: Adapted information is the information that is transported transparently across a server layer network. Adapted information is the client information encoded in such a manner that it is transportable across the layer network. This encoding can include labelling of the client information in order to distinguish the client information within the context of a single instance of adapted information. Adapted information is the construct that allows independence between client and server.

c) *Characteristic information (CI)*: This is the combination of the adapted information with additional information (layer information) that is transported across the network. Some of the layer information can remain unchanged across the network, though it can be read within the network, while other layer information may be altered within the network.

The only information which can be read (and by implication understood) within the layer network is the layer information which is added to the adapted information to form the characteristic information and is added by a termination function. This is irrespective of whether or not the symbols encoding the information are changed within the layer network. For example, if the layer network needs to read a field for its operation, this field is not transparent to the layer network and cannot be part of the adapted information. This field must be treated as belonging to the termination function.

The client may pass information to the server layer network, and vice versa, which must be understood by both. Payload type, destination, and QoS marking are examples. These must be passed as parameters between the client and the layer network along with the adapted information, and may be carried as part of the CI or may be carried 'out of band'.

This Recommendation notes the significant difference between information that is carried transparently (adapted information), and information that is necessary to the operation of the layer (layer information). Although [ITU-T G.805] notes information transparency, it does not explicitly note the role played by layer information and, in particular, it does not recognize the role played by parameters.

Because of this, some layer information and parameter processing have been assigned to the adaptation function. The effect of this choice is that the access point [ITU-T G.805] is unwittingly moved inside the adaptation function. With respect to equipment models, this choice is not critical because there is no open interface at the access point. However, management and control plane models are much more involved with transparency (and view the access point as the strict inter-layer delimitation point), and are affected by that choice.

Because layer information is information that is inspected or modified by the layer, it is quite easy to recognize layer information in existing uses of [ITU-T G.805] and transcription is straightforward. See the examples in Appendix IV.

6.5 Transport processing functions

A transport processing function may be considered, without implying an implementation, as a universal algorithmic state machine and "firmware" information, or as the programme which defines the specific behaviour of the entity. This "firmware" information is pertinent to the design of the entity. The behaviour of the entity is now controlled by the information passing into it through the ports.

This Recommendation is only concerned with configuration information that is material to the external behaviour of the entity as a specific labelling and encoding entity.

Four generic processing functions of adaptation, termination, layer processor and forwarding, together with labelling and encoding are described below.

6.5.1 Adaptation function

6.5.1.1 Adaptation source function

The adaptation source function is a labelling and encoding entity that takes one or more client communications passing through its client facing input port(s), and combines them into instances of adapted information. The adaptation source function also adds sufficient labelling in order to distinguish each client communication from all others within the scope of the access point to which the adaptation source is bound. The adaptation source function may provide parameters which are required for the operation of the server layer. The instances of adapted information are passed through the server-facing port.

6.5.1.2 Adaptation sink function

The adaptation sink function is a labelling and encoding entity that receives adapted information at its server-facing port, identifies the labels for client communications that are intended to be received by the adaptation sink function, and ignores all others. It may also accept parameters from the server layer. It then reconstructs the client communications and passes these out through its client-facing output port(s).

6.5.1.3 Adaptation and information constructs

The relationship between an adaptation function and information entities is illustrated in Figure 6-6.

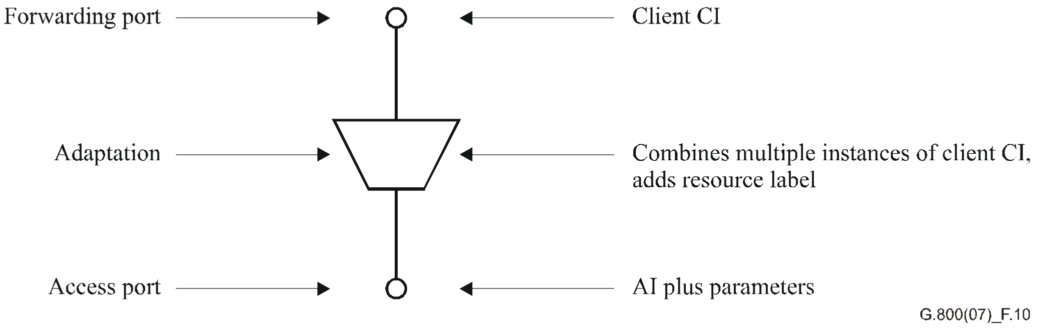


Figure 6-6 – Relationship between adaptation and information entities

6.5.2 Termination function

6.5.2.1 Termination source function

The termination source function adds layer specific information (e.g., encoding, labelling, fields for sublayer OAM) to create the layer characteristic information. The added information is the layer information (LI), such that characteristic information is equal to the adapted information plus the layer information. It may also accept parameters from the server layer. The operation of the termination function is independent of the client layer network. In accordance with the ability to offer transparency only, the layer information may be interpreted or modified within a layer network. Further fields cannot be added to the characteristic information symbol unless they have been predefined by the termination function that creates the characteristic information.

6.5.2.2 Termination sink function

The termination sink function extracts layer specific information (e.g., encoding, labelling, fields for sublayer OAM) to create the layer adapted information. The extracted information is the layer information (LI), such that the adapted information is equal to the characteristic information minus the layer information. The operation of the termination function is independent of the client layer network. The termination sink function also creates parameters that are passed to the adaptation function. In accordance with the ability to offer transparency only the layer information may be processed within a layer network. Further fields cannot be extracted from the characteristic information symbol unless they have been predefined by the termination function that outputs the adapted information.

6.5.2.3 Termination and information constructs

The relationship between a termination function and information entities is illustrated in Figure 6-7.

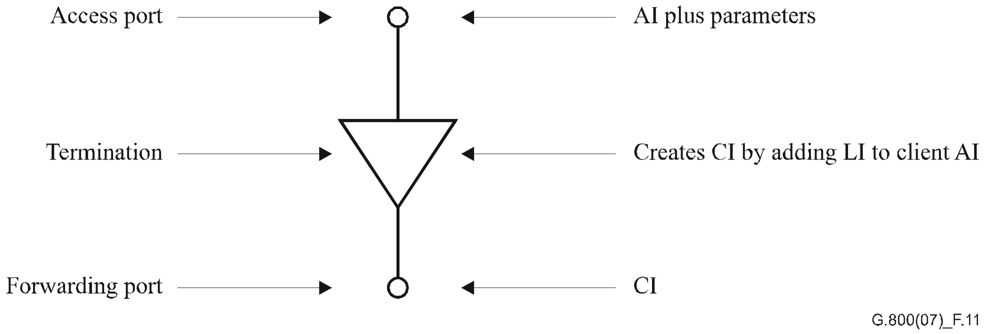


Figure 6-7 – Relationship between termination and information entities

6.5.3 Layer processor function

A layer processor function is a transport processing function that operates within a single layer network. To preserve the transparency of the AI, this type of function can, by definition, only read and modify the LI. The relationship between a layer processor function and information entities is illustrated in Figure 6-8.

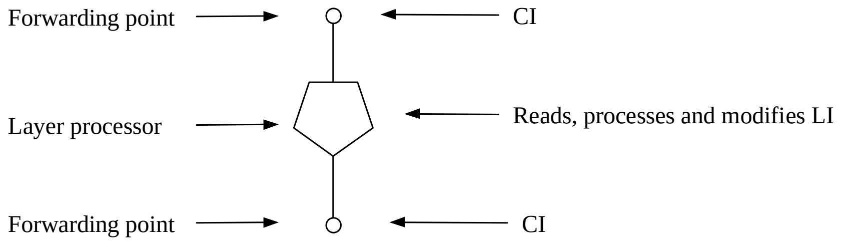
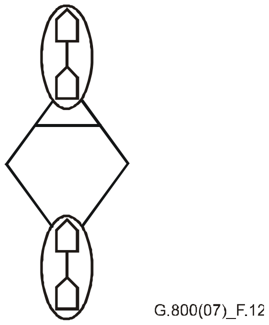


Figure 6-8 – Relationship between a layer processor function and information entities

An example of a layer processor function is a traffic conditioning function, which can also be denoted as shown in Figure 6-9. The traffic conditioning function accepts the characteristic information of the layer network at its input, classifies the traffic units according to configured rules, meters each traffic unit within its class to determine its eligibility, polices non-conformant traffic units, and presents the remaining traffic units at its output as characteristic information of the layer network.



Note that the diagram illustrates a unidirectional function. It is also possible to have a bidirectional function.

Figure 6-9 – Traffic conditioning function

6.5.4 Forwarding function

A forwarding function is a transport processing function that is configured in a subnetwork or link[1] topological component. It is responsible for transferring symbols presented at its ingress ports to its egress ports based on control information and configuration policy. During configuration, the following information must be provided:

– the ingress and egress forwarding ports[2];

– the type of forwarding that is supported (channel or destination);

[1] The forwarding function in a link is normally present as a default and does not need to be configured.

[2] In the case of a subnetwork, normally the forwarding port on a link that is bound to the subnetwork is referenced (instead of the forwarding point on the boundary of the subnetwork); this creates a forwarding point directly.

– port selection policy (may be null);

– forwarding policy (may be null).

6.6 Reference points

Reference points are formed by the binding between input ports and output ports of transport processing functions and/or transport entities. The allowable bindings and resultant specific types of reference points are shown in Table 1.

Table 1 – Allowable bindings and resulting reference points

Architectural components				**Reference point**	
Adaptation	Source output	Termination	Source input	AP	Uni
	Sink input		Sink output		Uni
	Source/sink pair		Source/sink pair		Bi
Termination	Source output	LC	uni input	FP	Uni
	Sink input		uni output		Uni
	Source/sink pair		Source/sink pair		Bi
Termination	Source output	SNTE	uni input	FP	Uni
	Sink input		uni output		Uni
	Source/sink pair		Source/sink pair		Bi
LC	uni input	SNTE	uni output	FP	Uni
	uni output		uni input		Uni
	Source/sink pair		Source/sink pair		Bi
LC	uni input	LC	uni output	FP	Uni
	uni output		uni input		Uni
	Source/sink pair		Source/sink pair		Bi
Adaptation	Source input	Adaptation	Sink output	FP	Uni
	Sink output		Source input		Uni
	Source/sink pair		Source/sink pair		Bi
Layer processor	Source output	Adaptation	Source input	FP	Uni
	Sink input		Sink output		Uni
	Source/sink pair		Source/sink pair		Bi
Termination	Source output	Layer processor	Source input	FP	Uni
	Sink input		Sink output		Uni
	Source/sink pair		Source/sink pair		Bi

AP	Access Point	LC	Link Connection
bi	Bidirectional		
uni	Unidirectional	FP	Forwarding Point

6.7 Topology

The network topology of interest is in the plane that transfers the characteristic information (CI), and this is represented by the largest subnetwork. This plane is the transport plane. This is illustrated in Figure 6-10 below.

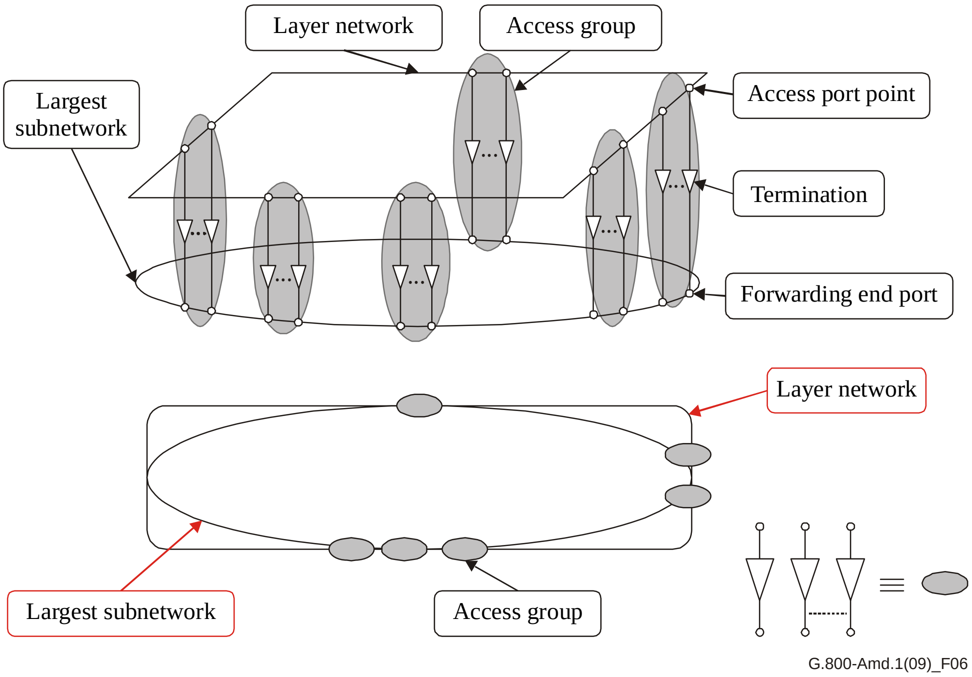


Figure 6-10 – Layer network

The internal structure of this plane can be further described by partitioning the largest subnetwork into smaller subnetworks (points of flexibility) and the links that interconnect them. The binding between a link and a subnetwork results in a link point.

The representation of a layer network as subnetworks and links is equivalent to a graph theory representation where a subnetwork corresponds to a node (or vertex) on the graph and a link corresponds to an arc (or edge) on the graph.

A subnetwork may be partitioned into smaller subnetworks interconnected by links. In addition, subnetworks and the links that interconnect them can be aggregated into a larger (containing) subnetwork. In this case, the details of the contained links and subnetworks are not visible.

An example of recursive partitioning is provided in Figure 6-11 below.

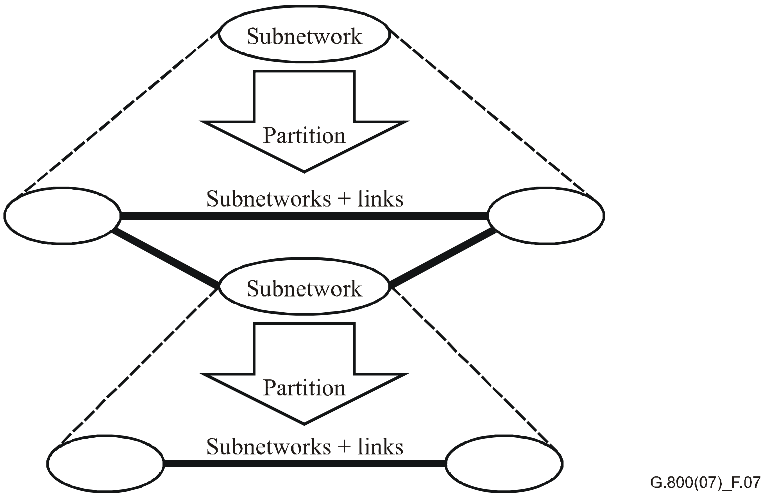


Figure 6-11 – Example of recursive partitioning

Partitioning of a layer network allows different subnetworks to be administered by different organizations (as required by axiom 5).

The links in a client layer network are supported by trails in a server layer network; this is illustrated in Figure 6-12 below. The transport entities and the components that support these relationships are described later in this clause.

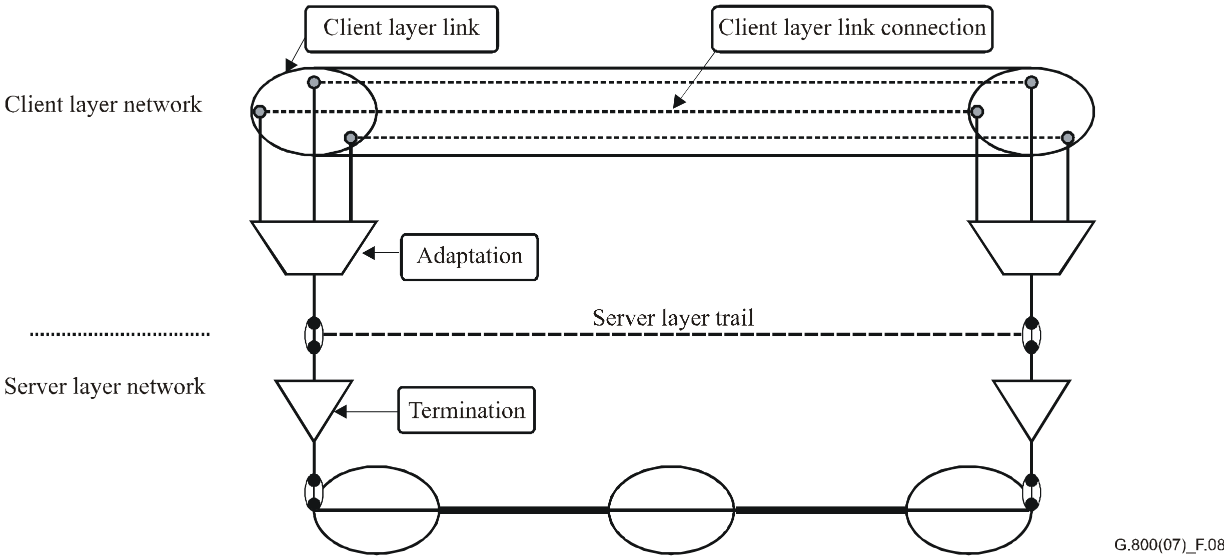


Figure 6-12 – Client-server relationship

6.7.1 Partitioning of a topological component

When addressing the partitioning of topological components, both the topological and resource aspects of the components must be considered. A subnetwork may be recursively partitioned to a degree such that the resultant subnetwork is considered to be in a single (spatial) location, and therefore further partitioning does not result in the ability to add any more precision to the location information. At this level of partitioning, the subnetwork is considered to be a node. Note that a node is not necessarily the limit of recursive partitioning, of a subnetwork. Further, depending on the network implementation, it may not be possible to partition a subnetwork to the level of a node.

For example, if a subnetwork belongs to a different administration and the policy of that administration is not to allow other administrations to see internal structure, or where the subnetwork is supported by a server layer that uses destination-based forwarding as described in clause 6.3.2.

6.7.2 Resource considerations

When considering the support of communications, it is necessary to examine the resource and connectivity restriction aspects of topological components.

The resource aspects of a link are the capacity, support (or not) of capacity reservation and temporal characteristics (e.g., delay, jitter) and other impairment characteristics (e.g., symbol loss after mitigation). The link inherits the characteristics of the server layer trail (or trails) supporting it. Some of these characteristics may be mitigated as described in clause 7. The inheritance of characteristics applies recursively over all layer networks down to the physical infrastructure.

A subnetwork may have some restrictions on the forwarding capabilities that it supports:

– It may not offer full flexibility because of restrictions imposed on the forwarding that it supports between some or all of its ports.

These flexibility restrictions may be imposed by the supporting hardware, the network configuration or the policy of the network operator. These restrictions can be described as a set of constraints that may be attached to the subnetwork. Partitioning of the subnetwork may yield simpler constraint rules.

The underlying resources that support the subnetwork may have capacity limitations because of either:

- the capacity of the links that interconnect any contained subnetworks; or
- the physical forwarding hardware has capacity limitations.

It is also possible to partition a node into multiple independent subnetworks, each of which is under the control of an independent control functional group (e.g., different routing areas, see [ITU-T G.8080]).

We can consider a matrix as the limit of recursion of subnetwork partitioning, which need not be further partitioned to expose connectivity restrictions or location information (i.e., the subnetwork is within a node) and is non-blocking, i.e., it has the resource capacity to:

- accept any request to configure a transport entity; and
- ensure that any symbol offered for transfer on a configured transport entity is guaranteed resource allocation within the matrix.

NOTE – It may not always be possible to partition a subnetwork to a set of such matrices.

Most common path computation algorithms expect nodes to be non-blocking and expect blocking or congestion on links. Thus, for the purposes of path computation, the network should be partitioned to the level of non-blocking matrices and links. This allows a network planning application to observe link utilization and adjust the link capacity to reduce the blocking or congestion to an acceptable level.

Transferring information between layers or sublayers also has an associated resource property. For this Recommendation, it is assumed that whenever this occurs, the adaptation/termination or layer processor functions have adequate resources for the link/subnetworks they source or sink.

6.7.3 Assignment of topology to organizations or communications

Partitioning allows different organizations to administer different links and subnetworks. It is also possible to allow multiple organizations to administer resources within the same subnetwork. An example application is the support of virtual private networks (VPNs) using a common set of network resources. This is achieved by dividing the subnetwork (including the contained links and subnetworks) into domains and assigning control of a domain to an organization. The representation of the capability to share resources between multiple organizations is outside the scope of this Recommendation (see [ITU-T G.8080]). Domains are also used to model semantically different networks using common underlying resources, thereby allowing for mixed networks using a common hardware platform.

A domain may be further divided into subdomains that support communications for a single user.

A subnetwork domain is formed from a subset of the ports on the containing subnetwork and inherits all of the properties of the containing subnetwork.

A sublink is formed from a subset of the ports on the containing link and inherits all of the properties of the containing link. A specific portion of the capacity of the containing link is assigned to a sublink.

From the perspective of the organization that has control over the resources, a subnetwork domain is a subnetwork and a sublink is a link.

6.8 Layer relationships

The interlayer relationship allows us to "build" a client layer topology in terms of links and subnetworks that are supported by transport entities in a server layer. The topology of the server layer network is used to construct transport entities which in turn support the topology of a client layer network. Each instance of a client layer network inherits the mitigated characteristics of the transport entities of its server layer network. The characteristics of this layer network are combined with the inherited mitigated characteristics and are presented as the characteristics of this layer network when it is acting as a server layer network.

To describe the characteristics of a layer network, it must be partitioned to an appropriate degree. In some cases this may not be possible, for example, due to an administrative policy or because of the nature of the server layer network.

If the layer network uses channel forwarding, at its access points, the layer network provides a trail transport entity. This is represented as a link in the client layer network topology.

If the server layer network uses destination forwarding, at its access points, the server layer network offers a multipoint to multipoint access transport entity and expects the client adaptation to provide destination information (as parameters) with each message. At the ingress access point, the interlayer adaptation function must translate the destination information, provided as parameters, associated with each symbol into a primitive that identifies the intended egress access point. The termination function maps this primitive into the server layer network destination address. We have two cases for the client layer network:

- If the client layer network uses destination forwarding, the server layer transport entity may be represented as a subnetwork in the client layer network topology. The ingress adaptation function maps the destination information carried by each message into the primitive that identifies the intended egress access point. However, this subnetwork cannot be partitioned within the client layer network and in general is not a node since it may have some geographic distribution.
- If the client layer network uses channel forwarding, then the server layer transport entity may be represented as a link in the client layer topology. In this case, the ingress adaptation function must map the intended link end point into the primitive that identifies the intended egress access point. This creates (from the perspective of the client) a persistent point to point transport entity. However, this transport entity cannot be decomposed into a predetermined concatenation of subnetwork connections and link connections.

6.8.1 Inheritance of geographic properties

In order to predict the degree to which a layer network can offer resilience against resource failures (caused for example by the failure of a cable or physical site), the topology must include geographic data, and therefore it can be partitioned to the level of nodes and links. Each link must be supported by a transport entity (in the server layer network) that can also be partitioned into nodes and links. This recursive relationship must be supported down to the physical infrastructure.

6.8.2 Links supported by multiple server layer trails

Three different cases exist, these are described below:

Multiple parallel links between the same subnetworks can be bundled together into a single composite link. Each component of the composite link is independent in the sense that each component link is supported by a separate server layer trail. The composite link conveys communication information using different server layer trails, thus the sequence of symbols crossing this link may not be preserved. This is illustrated in Figure 6-13.

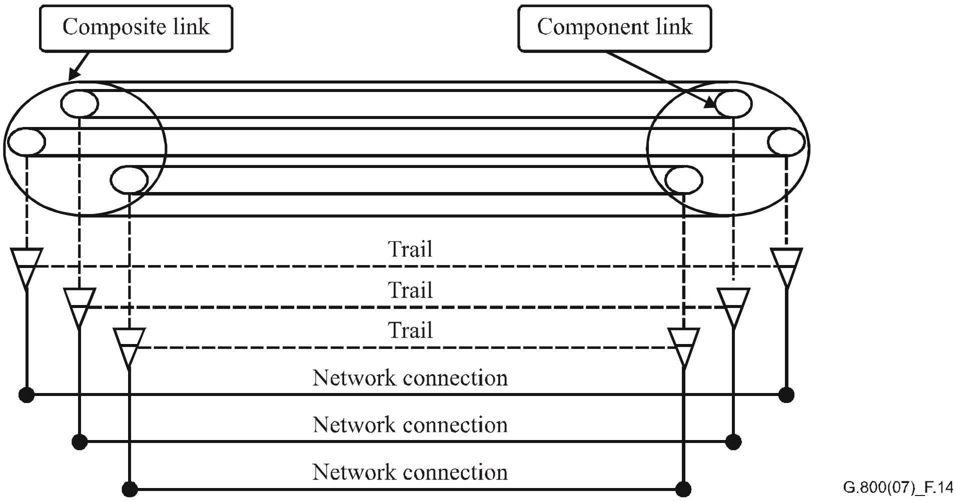


Figure 6-13 – Composite link supported by multiple independent server layer trails

Multiple server layer trails can be combined using the inverse multiplexing technique described in [ITU-T G.805]. This creates a new composite rate trail with a capacity that is the sum of the capacity of the component trails. The link in the client layer is supported by this composite trail. This link may support a single link connection and it preserves the sequence of any symbols that use this link connection. This is illustrated in Figure 6-14. Note that the composite trail is not visible in the network.

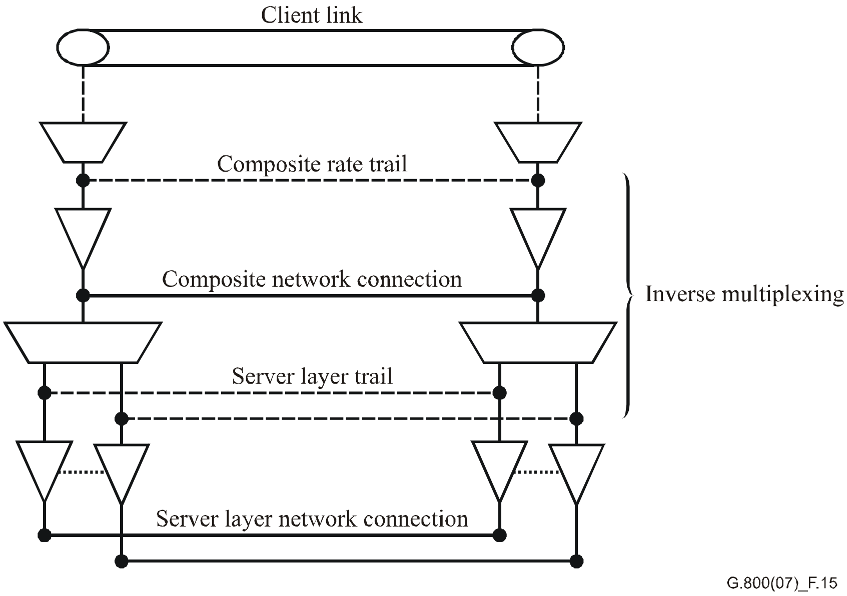


Figure 6-14 – Client link supported by inverse multiplexing

A link can also be constructed by a concatenation of component links and configured subnetwork connections. This is illustrated in Figure 6-15.

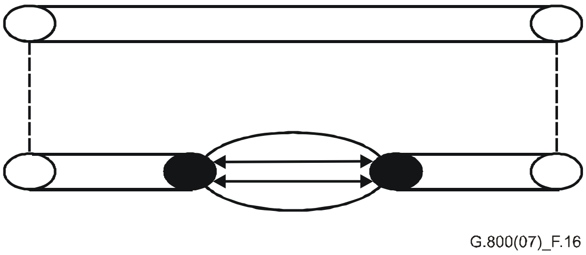


Figure 6-15 – Serial compound composed of component links and a subnetwork with configured subnetwork connections

The subnetwork connections must have a 1:1 correspondence to the link connections that will be provided by the client link. In this case, it is not possible to fully infer the status of the link by observing the server layer trails visible at the ends of the link.

6.9 Transport over composite links

6.9.1 Transport over composite link for transport resilience

If a differentiated link connection is configured over a composite link, its ingress can distribute individual communications over component links based on a distribution algorithm. If the differentiated link connection has reserved spare capacity on the composite link, it can redistribute impacted communications to other available component links when a component link fails or is degraded. A distribution and redistribution algorithm can use component link attributes, available information on communications, and policies for distribution decision. As a result, a composite link can be used to gain transport resilience.

6.9.2 Faults and their supervision in a composite link

Component links in a composite link can fail independently, which causes composite link capacity reduction. The scenario is referred to as a composite link constituent fault.

A constituent fault on a composite link can affect its link connections in two distinct ways, depending on the distribution function used by the composite link. If the composite link distributes each link connection it supports to only one component link, then a partial fault may cause some link connections to fail completely and others to be unaffected. The failed link connections may be redistributed over other working (non-failed) component links. If a composite link treats a link connection as a differentiated link connection and distributes communications to different component links, then a partial fault causes the link connection traffic to be redistributed over the remaining working (non-failed) component links. In both cases, the redistribution can reduce the link capacity available to these or other link connections supported by the composite link.

Whether a composite link function must send status information indicating reduced capacity or failure of link connections depends on the recovery mechanisms in use. If the composite link function can fully recover the lost connectivity locally by redistributing traffic across the working (non-failed) component links, it may not be necessary to send status information. If local action is not sufficient to recover lost connectivity, status information may be sent on selected connections depending on recovery policy. For example, some link connections may be selectively shut down to avoid others being affected or several link connections may signal reduced capacity to their connection endpoints to allow connectivity for all (fair reduction).

7 Transparency and impairments

This Recommendation describes the network in terms of recursive layer networks that offer "transparent" information transfer. In general, a layer network is not capable of transferring information without imparting some impairments.

As described in clause 6.4, when the CI of a client layer is transported over a server layer network, the symbols from a client layer CI lexicon are mapped into the AI lexicon of the server layer by an inter layer adaptation function. The inter layer adaptation function may multiplex multiple instances of client CI into a single instance of AI. The client layer CI must be transferred with the required degree of transparency. The mapping between the lexicon of the client layer CI and the server layer AI must include the information required to both demultiplex the individual instances of client CI and mitigate any AI transfer impairments (e.g., loss of order or corruption of timing information) to a degree that is acceptable to the client. The server layer does not have the ability to interpret the meaning (semantics) of the AI symbols. The transparency of the information transferred across a layer network between its access points is defined in terms of:

– The AI lexicon from which the inter layer adaptor can select symbols that are to be transferred.

– The integrity of the symbols that are transferred.

– The inherent information in a sequence of symbols in an open sequence or timed sequence.

Figure 7-1 below illustrates the transparency that is required to provide layer independence.

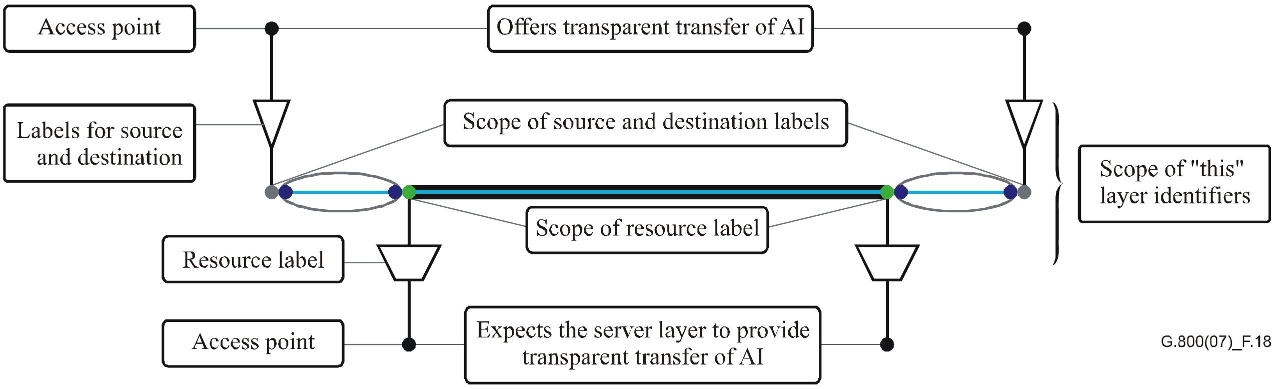

Access point
Offers transparent transfer of AI
Labels for source and destination
Scope of source and destination labels
Scope of "this" layer identifiers
Scope of resource label
Resource label
Access point
Expects the server layer to provide transparent transfer of AI
G.800(07)_F.18

Figure 7-1 – Scope of identifiers and transparency

The access point provides an abstraction barrier that isolates the client from the server. This allows arbitrary stacking of layer networks.

If a server layer relies on some specific information that is encoded by the client layer, then those layer networks are no longer independent and must always be deployed as a pair of layer networks. In a network with recursive layer networks that do not support the transparency property as defined, all possible combinations of layer networks must be defined.

There are several aspects of symbol impairment within a transport entity:

– symbol value degeneration (e.g., bit errors);

– symbol order degeneration (deviation from original order);

– symbol timing degeneration (deviation from original time position);

– symbol delivery degeneration (deviation from the intended output port or ports or delivery of unintended symbols to the designated ports);

– non delivery of symbols (e.g., caused by resource exhaustion (resulting in the failure to allocate a resource to a symbol), or symbol corruption).

The server layer may encode the AI symbols to allow mitigation of impairments incurred within the layer or its server layer.

Besides degeneration of the symbols, an access, network, subnetwork, matrix, server-subnetwork or link transport entity may experience unintended modifications:

- ingress port with attached sender is added (unexpected symbols inserted);
- egress port with attached receiver is added (misdelivery of symbols);
- ingress port with attached sender is removed;
- egress port with attached receiver is removed;
- failure or mis-configuration of the resources supporting the transport entity (resulting in a short or long break in the transport entity);
- combinations of the above.

The techniques applied to the detection and mitigation of some or all of these impairments are described in technology-specific Recommendations, and are based on deployment of additional, well known OAM information symbols, which complement the adapted client symbols within the monitored transport entity. Mitigation of those impairments is provided via repair actions of such impaired transport entities.

The adaptation function or a function in the server layer may encode the symbols to allow mitigation of impairments incurred during the transfer of the symbols.

7.1 Inter-layer information dependency

In a strict sense there should be no information dependency between layer networks. Where there is, this may be decomposed into several different aspects:

a) Visibility and use of client information at the access point (i.e., in the adaptation function). The adaptation function reads the client "control" information and presents this as parameters across the access point. The termination function encodes these parameters into the syntax of the server layer[3]. This maximizes independence between the client and server symbol sets. The syntactic and semantic information dependency is resolved at the access point. In order to preserve the fidelity of transfer the appropriate server layer encoding must be chosen.

b) Use of encoded client information between the forwarding end points. Within the server layer some of the client information that has been encoded into the syntax of the server layer (i.e., server layer LI) is used. Three general cases in which client control information may be used in a server layer network are:

   - Symbol Destination(s) for a destination forwarding server layer network transport entity.

     NOTE – Transport entity destination information is resolved for a channel forwarding layer network when the transport entity is set up.

   - Symbol Urgency (e.g., queuing priority) for a packet transport network transport entity.

     NOTE – This is resolved for a circuit network when the transport entity is set up.

   - Symbol Importance (e.g., drop precedence) for a packet transport network transport entity.

     NOTE – This is resolved for a circuit network when the transport entity is set up, e.g., protection mode, and may be resolved similarly in a packet network as well.

[3] The encoded syntax of the parameters used by the adaptation function may match the syntax used by the server layer, in this case the encoding in the termination is a null function.

However, if channel-based forwarding is used, this may result in impairment of the fidelity of transfer. For example, in a packet network where a transport entity utilising channel-based forwarding has multiple client priorities, each mapped into different server priorities, then the order of symbols with different (client) priorities will not be preserved. Such impairment may not be significant if the client does not expect that the order of symbols with different priorities is preserved during transfer.

c) Visibility and use of the information contained in client symbols (i.e., client LI) between forwarding end points. Functions in "this layer" read and interpret the syntax and semantics of the client "control" information that was passed across the access point as adapted information (AI). In this case "this layer" is not transparent within the ITU-T G.800 definition since "this layer" is assuming the presence of some information element that is provided directly by the client.

8 Sublayers

It is often useful to identify sublayers within a specific layer network in order to identify additional reference points. The ITU-T G.805 definition of sublayer mentions sublayer creation by decomposition of function or expansion of reference points. ITU-T G.800 slightly expands that idea to describe the sublayer fields within the layer CI definition. To be able to activate a sublayer, the definition of the CI must already include the fields that will be used by the sublayer. A sublayer is activated within the context of an existing transport entity.

A sublayer provides a set of reference points that are unique within the scope of the transport entity. These "sublayer reference points" are typically used for OAM or protection switching. The terms maintenance point (MP) or maintenance port (MPt) are used to differentiate the sublayer reference points from the layer network access points (AP). Maintenance points (or ports) are only visible or accessible within the transport entity within which the sublayer has been activated.

Sublayer OAM information may be added or removed by introducing a layer processing function and a sublayer OAM termination function at a termination forwarding point or a forwarding point[4] in a transport entity. The type of forwarding (channel or destination) that the layer network uses creates different cases for the sublayer.

8.1 Channel forwarding

The sublayer (OAM) information is added within the channel used by the client information. The OAM information may either be added to the original message (for example the tandem connection overhead in a synchronous digital hierarchy (SDH) network); or, it may be contained in a new message with an identical envelope. The latter method can only be used in a packet-switched network and must be taken into account by the resource reservation/allocation process.

8.1.1 Insertion/removal at a forwarding point

Figure 8-1 shows the insertion of sublayer processing functions at an FP.

[4] [ITU-T G.805] describes this as "decomposing the trail termination function or connection point and describes this by inserting sublayer adaptation and termination functions".

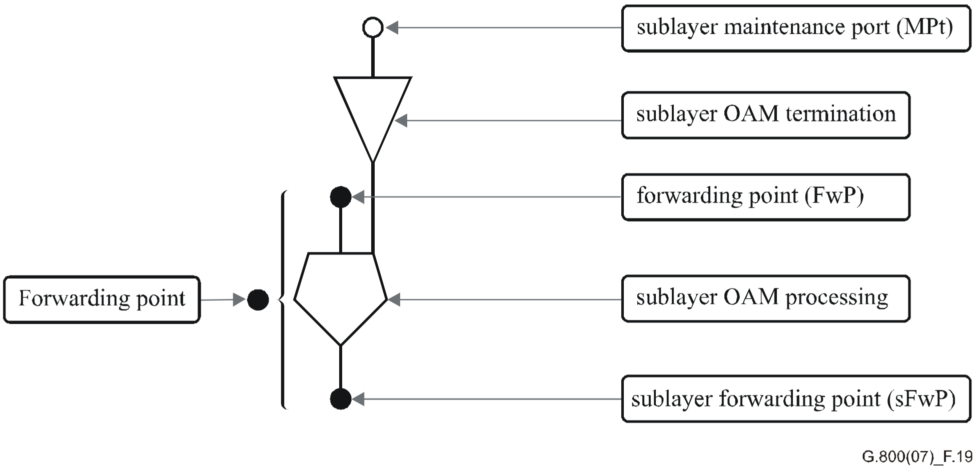


Figure 8-1 – Insertion at a forwarding point

The sublayer OAM termination source function originates the sublayer OAM information. It must also insert an identifier for the maintenance port. The scope of the maintenance port identifier is limited to that of the original transport entity. The layer processing source function monitors the client traffic presented at the FP, passes the appropriate information to the sublayer OAM termination function. It multiplexes the sublayer OAM information (from the termination) with the CI.

The layer processing sink function monitors the traffic at the sublayer FP and passes the appropriate information to the sublayer OAM termination function. It demultiplexes the sublayer OAM information from the CI. The sublayer OAM information is passed to the termination sink function where it is terminated.

8.1.2 Insertion/removal at a forwarding end point

Figure 8-2 shows insertion of a sublayer processing function at FP.

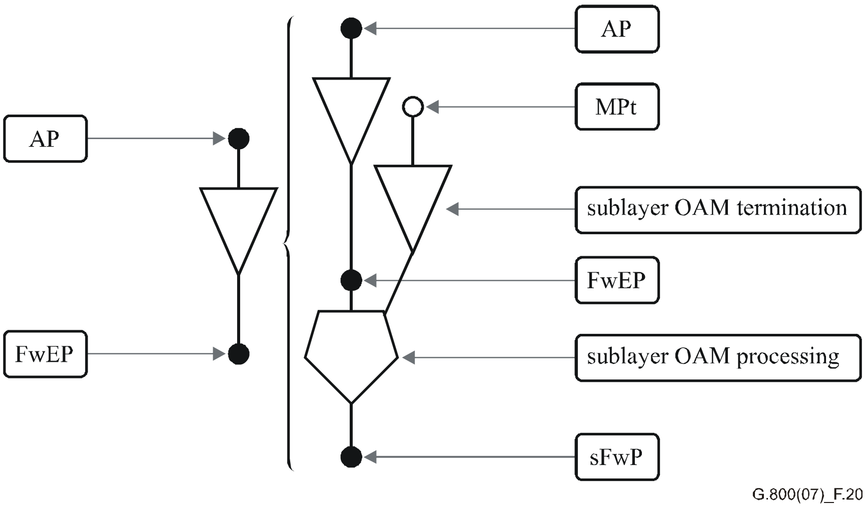


Figure 8-2 – Insertion at a forwarding end point

The operation of the sublayer termination function and layer processing function is identical to that described in clause 8.1.1.

8.1.3 Sublayer trail

The sublayer trail exists between the source and destination maintenance ports. The sublayer trail uses the same transport entities as the client signal. Therefore, the symbols conveying OAM information follow the same path across the network (and are exposed to the same impairments) as the symbols conveying client traffic AI. The lifetime of the sublayer trail is tied to the lifetime of the transport entity. This tight coupling between the sublayer trail and the client communication makes the insertion of OAM indications (such as forward defect indication) relatively simple. A non-intrusive monitor may be used to allow the sublayer OAM information to be observed at intermediate points. In this case, a layer processing sink function monitors the traffic at the sublayer FP and passes the appropriate information, including the sublayer OAM information, to the sublayer OAM termination function.

The visibility of the maintenance ports is inherently limited to the scope of the transport entity within which it has been activated. The sublayer OAM information may include the MP source and destination identifiers. Use of these identifiers may allow multiple sublayer OAM trails to exist in the same channel forwarding transport entity.

8.2 Destination forwarding

The sublayer (OAM) information is added within the destination-forwarding transport entity used by the client information. The OAM information is inserted in a new message that conforms to the CI of the layer network. Since destination forwarding is used, each message carries a forwarding identifier that identifies the intended destination. The messages carrying OAM information must also contain a forwarding identifier that identifies the intended OAM destination or destinations (within the destination forwarding transport entity). This (maintenance) forwarding identifier may be independent of any client communications. Therefore, OAM messages and client messages may be subject to independent forwarding decisions. The exchange of OAM messages allows the functionality of the segment of the transport entity between the source and sink functions to be verified. However, in a destination forwarding transport entity only a subset of the forwarding points may be used by the messages that carry client communications (i.e., client AI). This should be taken into account when selecting a forwarding point to insert/remove OAM messages.

8.2.1 Insertion/removal at a forwarding point

When sublayer processing functions are added to allow the insertion of OAM information at a forwarding point, we have two possibilities, either:

– the sublayer OAM information can be inserted in the context of a specific client communication; this is equivalent to insertion at the FP described above. Given the nature of the communication (i.e., independent messages), monitoring individual client communications is not practical; or

– the sublayer OAM information is carried in the same transport entity by a new message that includes the forwarding identifier for the destination MP. This is illustrated in Figure 8-3 below.

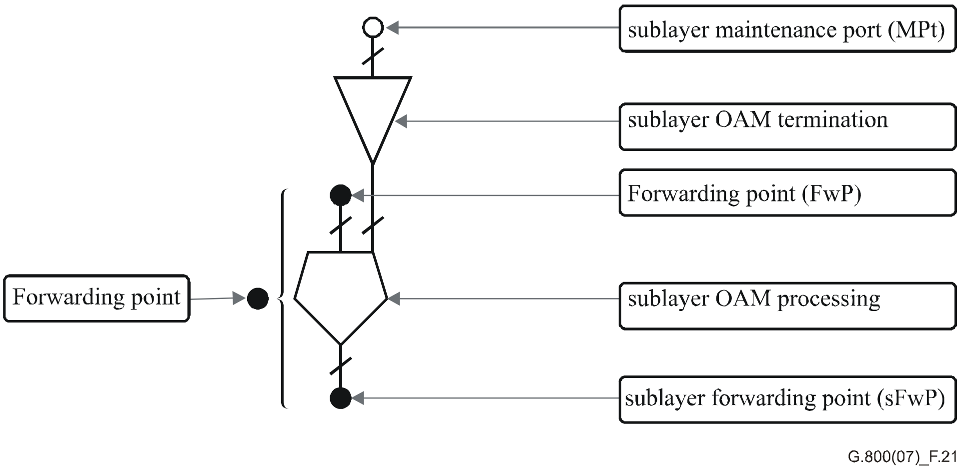


Figure 8-3 – Insertion at a forwarding point

This results in the generation of new messages, to carry the OAM information, with forwarding identifiers that are independent of the messages conveying client AI.

The sublayer OAM termination source function originates the sublayer OAM information. This must include the target (destination) maintenance port in the form of a forwarding identifier. It must also insert an identifier for the local (source) maintenance port. The scope of the maintenance port identifier is limited to that of the original destination forwarding transport entity. The layer processing source function monitors the messages presented at the FP, passes the appropriate information to the sublayer OAM termination function and multiplexes the sublayer OAM messages with the client traffic.

The layer processing sink function monitors the messages at the sublayer FP, passes the appropriate information to the sublayer OAM termination function and demultiplexes the client traffic from the sublayer OAM messages. The sublayer OAM messages are passed to the termination sink function where they are terminated.

The operation of the sublayer termination function and layer processing function is identical to that described in the previous clause.

8.2.2 Sublayer OAM maintenance relationship

The sublayer OAM maintenance relationship exists between the source and destination maintenance points, within the destination forwarding transport entity used by the client information. It is supported by a sublayer OAM transport entity. Note that if a multicast address is used, a single OAM message may be directed to multiple destination maintenance points within the destination forwarding transport entity. The sublayer OAM forwarding is independent[5] of the messages conveying client AI since they have independent forwarding identifiers. The sublayer OAM transport entity may be used to monitor the ability to forward client messages that transit the same sequence of forwarding points between the maintenance points. This monitoring is only valid if the implementation of the transport entity forces all of the messages to follow a common path. It may also be used to monitor the integrity of the transfer of the aggregate of the client messages provided that the transport entity does not merge in any other messages along this path, and maintains the order of messages in the aggregate.

[5] The OAM messages may contain information about the client messages, but the forwarding of OAM messages is independent of the forwarding of the client messages.

It is possible to activate a non-intrusive monitor at an intermediate point on the sublayer OAM transport entity. However, an update in the forwarding tables may cause the sublayer OAM messages and the messages carrying client AI to transit a different set of FPs.

The lifetime of the sublayer OAM transport entity is dependent on the lifetime of the transport entity within which it is created. It is independent of any messages conveying client AI (client communications). This makes the insertion of OAM indications into client communications (e.g., FDI) somewhat complex. A further degree of complexity is that outside the context of the sublayer OAM transport entity there are no inherent constraints on the routing of the messages conveying client AI (client communications). Since each sublayer OAM communication is essentially independent of any other communications, it is possible to have multiple sublayer OAM communications within the same transport entity.

9 Transport network availability enhancement techniques

In [ITU-T G.805], several transport network availability enhancement techniques are described. This clause describes additional techniques which may be used to enhance the availability of a transport network. These techniques include differentiated connection protection and composite link protection.

9.1 Differentiated connection protection

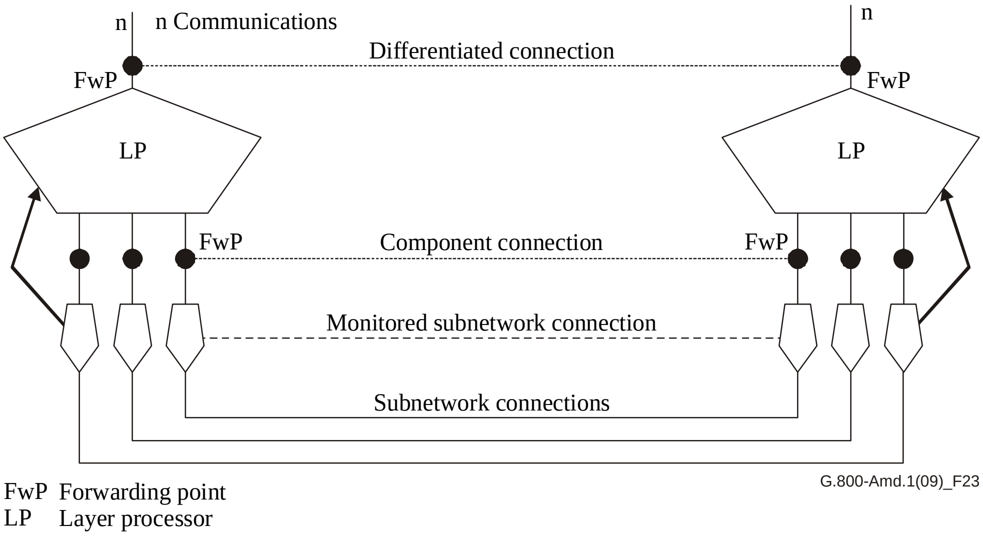


Figure 9-1 – Transport model for a differentiated connection

A differentiated connection, defined in clause 6.3.4, can provide transport resilience. Figure 9-1 illustrates a differentiated connection transport model in a layer network. A differentiated connection has several component connections and is configured on layer processors residing at the differentiated connection ingress and egress. Each component connection is a connection as defined in clause 6.3.3. The component connections can have different capacities. The differentiated connection ingress and egress each have a single forwarding point (FPt). The component connections are subnetwork connections and may be routed through different paths in the layer network. The layer processor at differentiated connection ingress distributes traffic units to the component connections. Each component connection is independently monitored.

The differentiated connection carries multiple communications. The ingress layer processor can distinguish communications or sets of communications by examining datagram LI contents and distributes each communication to a single component connection. The egress layer processor takes datagram from each component connection and delivers them to the differentiated connection egress port. Each subnetwork connection will preserve the packet sequence of the communications it carries. The distribution function can consider communication traffic attributes and generate a mapping table between communications and component connections. In subnetwork connection failure situations, the distribution function implements a new distribution relationship. When a failure is detected, the layer processor implements a new mapping relationship. As a result, the differentiated connection provides resilient transport without using connection protection. Spare capacity must be reserved in the component connections to support this transport resilience mechanism.

A differentiated connection and its component connections may be configured as bidirectional. The two differentiated connection layer processors may independently distribute communications. Thus, a bidirectional client communication may be transported over different subnetwork connections in each direction. If bidirectional communications are required to be transported over a single bidirectional subnetwork connection, the two layer processors must use the same mapping relationship. In this case, one distribution function provides the mapping policy to both layer processors. Each layer processor executes the provided distribution policy. For more rapid recovery, the distribution function can pre-calculate the different failure scenarios and recovery plans, and provide multiple distribution policies to the layer processors. When a component connection fails or is repaired, some information exchange is necessary to ensure both ends use the same distribution policy.

9.2 Composite link protection

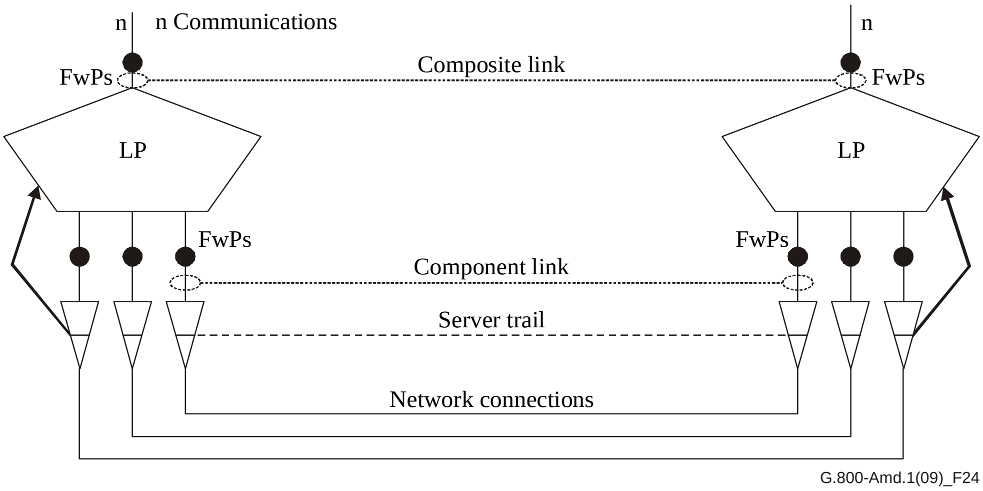


Figure 9-2 – Transport model for composite link (destination forwarding network)

A composite link defined in clause 6.9 can provide transport resilience without dedicated protection links. The transport model is shown in Figure 9-2. Multiple component links are bundled together into a single composite link. The component links are supported by independent server trails that are supported by individual server layer network connections. Layer processors reside at the composite link ingress and egress. All packets arriving at the composite link ingress are transported to the composite link egress but order may not be preserved between packets traversing different component links. To preserve packet sequence for individual communications, the distribution function in the layer processor uses LI in the packet to distinguish communications and sends each communication or set of communications over a single component link. For example, a destination address may be used by the distribution algorithm to ensure individual communications traverse a single component link. For finer distribution, other fields may be used as well. A component link failure will trigger the distribution algorithm to change the distribution to use only the remaining active component links. This provides transport resilience. In general, the distribution algorithm may not support traffic engineering due to lack of traffic engineering information for individual communications. However, if the network is aware of communication traffic characteristics, the layer processor can perform the distribution based on this traffic engineering information.

Annex A

Fundamental concepts used in Recommendation ITU-T G.800

(This annex forms an integral part of this Recommendation.)

This annex provides the Axioms, a description of the concepts of labelling and encoding, and the identifiers used in the main body of this Recommendation

A.1 Axioms of the unified architecture

The unified architecture of transport networks is derived from the following axioms:

Axiom 1

Telecommunication networks are concerned with the conveyance of information between senders and receivers when the senders and receivers are separated geographically.

NOTE 1 – A body of information produced by a sender and intended, in its entirety, to reach a particular receiver or receivers is called a communication.

Axiom 2

The means by which communications can be conveyed by telecommunications networks (resources) is normally limited and therefore needs to be shared amongst many communications.

Axiom 3

A telecommunications network needs to be able to select (and therefore identify) a sender of a communication and to select (and therefore identify) the intended receivers of that communication.

Axiom 4

The information content of a communication conveyed by the telecommunications network is sometimes subject to loss.

NOTE 2 – Loss of information includes:

- the corruption of symbols;
- loss of symbols;
- insertion of symbols;
- or other impairments whereby the intended receiver does not correctly receive the communication that was sent.

Axiom 5

The resources of a telecommunications network are administered by one or more organizations.

A.2 Information

There are two widely recognized definitions of information: the first, communication information, is relevant to the communication of information as it is defined in terms of the passing of information between entities; whereas the second, algorithmic information, is defined in terms of the complexity of a computation machine. Communication information is defined as a message selected from a possible set of messages, weighted by the probability of that message within the set, passed between a sending entity and one or more receiving entities. This is the definition of information as set out originally by Claude Shannon [b-Shannon]. Algorithmic information is defined as the smallest programme required for a Universal Turing Machine to construct a required bit sequence of information. This architecture utilizes both forms of information and describes their application, interrelationship and use in specification of functions.

The properties of information as described in this Recommendation are:

Property 1

The measure of information: For both communication information and algorithmic information, the measure of information is the binary "bit". The amount of information is the smallest length sequence of binary bits (with the assumption that for each bit, "1" and "0" are equally probable) needed to encode the information.

Property 2

Copying of information: Information can be arbitrarily copied without loss of information.

Property 3

Merging or combining of communication information: Any merging or combining of communication information will result in a fundamental loss of information unless information is added to distinguish the instances of communication information that have been merged or combined.

There are three basic types of communication information:

a) *Message*

A symbol selected from a finite set of symbols.

NOTE – The set of symbols from which a symbol can be chosen is a lexicon (or dictionary).

b) *Open Sequence (File)*

An open sequence is an open-ended sequence of messages, i.e., each symbol value is selected from a finite set of possible values for the symbol. The order of messages in the open sequence carries (implicit) information. An open sequence communication preserves the order of the messages.

c) *Timed Sequence (Stream)*

A timed sequence is an open-ended sequence of messages where the timing of each message relative to another is significant. The sequence and the relative time of each symbol carries (implicit) information. A timed sequence communication preserves both the order of the messages and the timing between each message.

These forms of communication information are illustrated in Figure A.1.

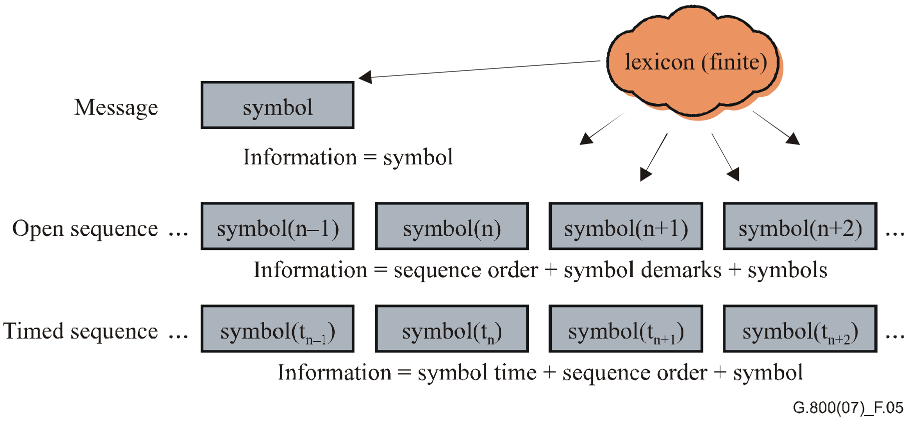


Figure A.1 – Types of communication information

A symbol is a recursive construct in that a new symbol can be constructed from a sequence of symbols. Similarly a symbol can be decomposed into a sequence of smaller symbols. The smallest possible symbol is a bit.

A consequence of axiom 3 and property 3 is that the telecommunications network must create and use its own information in order to distinguish communications.

A.3 Encoding and labelling

Labelling and encoding entities are information processing entities and have the following ports:

– a client facing port;

– a server facing port;

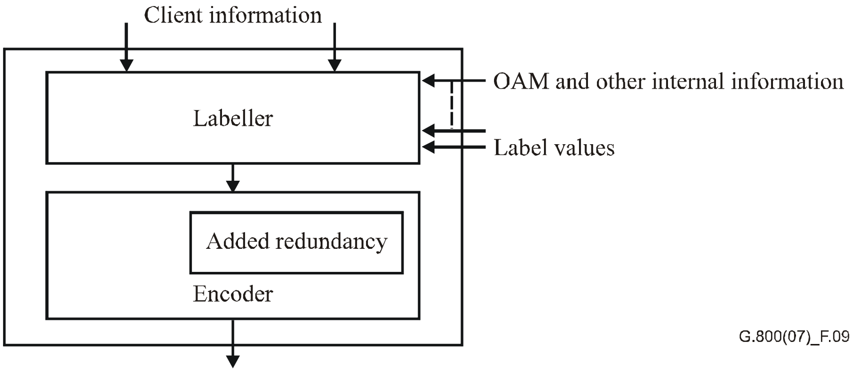
– a configuration and control port;

– an OAM message port.

In the case of the labeller, the value in any added fields is independent of the value of client information. A key characteristic of encoding which clearly distinguishes it from labelling is that the values of any added fields by the encoder (resulting from redundancy in the encoding) are dependent on the value of the client information.

The order in which encoding and labelling may take place inside an adaptation function or a termination function is such that either order is possible and that it is essential to the unified model that this flexibility is clear and unfettered. Figure A.2 shows a possible configuration to which many existing adaptation and termination functions can be mapped.

For any particular adaptation or termination function, the order does matter and the order is part of the definition of the behaviour of the function which is essential for the interface specification and interoperability. For example, the SDH termination functions specify clearly the bytes over which the BIP is calculated.



NOTE – A transport processing function may contain more than one labeller and encoder. The arrows external to the information system represent information that is presented via ports.

Figure A.2 – Labelling and encoding

A.3.1 Distinguishing and identifying individual communications on shared resources

A label provides a means of providing added information for the purpose of distinguishing and identifying individual communications within a communication which is formed to convey a combination of communications.

The terms "distinguish" and "identify" have a particular meaning in this Recommendation. To "distinguish" is essential from information theory in order to separate out communications from each other when they have been combined. To "identify" is essential to manage a communication.

The following types of label are used in this Recommendation:

Resource label: A resource label is the information required to *distinguish* a communication within a combination of communications.

Source label: A label which is used to identify the access point that a sender of information is attached to.

Destination label: A label which is used to identify the access point that the intended receiver of information is attached to.

Connectivity label: A connectivity label is the information required to identify the sender and intended receivers of an information instance.

Further details on these labels are provided in clause A.4.

Two cases are commonly used to describe the act of combining communications.

Aggregation is used to describing combining communications that already have sufficient labelling to distinguish and later separate them into individual communications. Aggregation does not create a new communication. Because the traffic units being aggregated already have sufficient labelling, further labelling at the aggregation point is not necessary.

Multiplexing is used to describe combining instances of communications, with sufficient new labelling added at the multiplexing point to distinguish and later separate them. Multiplexing creates a new communication.

A.3.2 Equivocation overhead

Equivocation is an example of the use of encoding processes to add overhead information that is used to ameliorate for the possible loss of information in the process of its transfer from a sender to a receiver. The following forms of equivocation overhead are defined in this Recommendation:

a) *Communication information equivocation overhead*: This is information that is coupled to the communication information. This coupling may be achieved in one of two ways (or in combination):

   – Overhead information that is derived from the communication information itself, in that its value is dependent on the communication information.

     NOTE 1 – Examples include cyclic redundancy check (CRC), bit interleaved parity (BIP), and forward error correction (FEC) schemes.

   – Overhead information that is known and deterministic but which is indistinguishable from communication information when transferred and is, therefore, subject to the same information loss mechanisms.

     NOTE 2 – Examples include connectivity check (CC) flows in packets/cell networks and frame alignment words in circuit networks.

     The purpose of communication information equivocation overhead is to allow a receiver to reliably monitor loss of client information.

b) *Control information equivocation overhead*: This is information that ameliorates for the loss of label information and/or loss of configuration and control information.

   NOTE 3 – Examples include path trace, signal label/protocol identifier (ID), and time to live (TTL).

c) *Forwarding equivocation overhead*: This is a communication that can be injected in order to make available a monitor for a transport entity in the transport plane. This overhead communication is independent of any other communications and is also distinguished from any client communications. Normally, forwarding equivocation overhead will give a good indication of the performance of other communications also using the transport entity; however, this overhead can be subject to systematic failures where another communication is impaired but the forwarding equivocation overhead will not detect the impairment, even in a statistical way.

   NOTE 4 – Examples include routing protocol "hello" messages.

A.4 Identity and identifiers

Each entity within a transport network has a unique identity but does not necessarily have a visible globally unique identifier. The various applications that control, use, or observe the transport network (including entities within the transport network itself) need the ability to identify some entities (e.g., reference points, communications, functions) within the transport network. Each of these applications requires an identifier for each of the transport network entities that are of interest. These identifiers are from the name space of that application and must be unique within the context of that application. In general, multiple applications may reference the same network entity; in this case, multiple identifiers will exist. Identifiers are an alias to the identity of the entity within the context of the mechanism of identification.

In some cases, an entity may make some implicit characteristic visible, e.g., a timeslot in an SDH frame or a wavelength in a wavelength division multiplexing (WDM) layer. These implicit characteristics are considered to be a label for the resource.

A.4.1 Identifiers

As defined in Axiom 2, the resources of the layer network are shared by multiple users. To allow the network to be configured to support multiple communications that are delivered to only the intended receiver(s), the configuration application and the layer network must be able to distinguish the topological components, the resources and the individual communications. The identifiers described below are used for this purpose; Figure A.3 is an example of a layer network with a single connection.

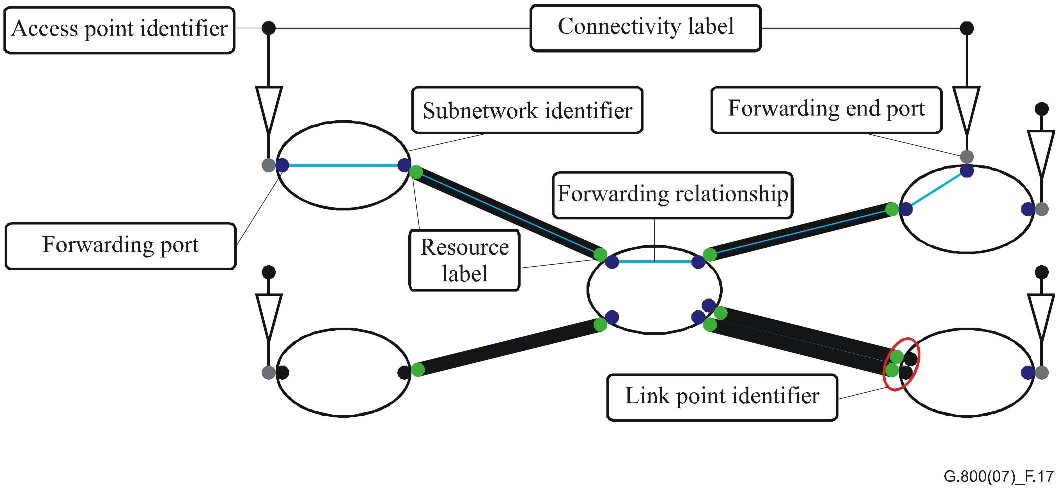


Figure A.3 – Identifiers in a layer network

All networks (independent of forwarding topology) require the same set of identifiers, the forwarding topology only impacts the scope and encoding of those identifiers:

Access point identifier: This identifier must be unique within the context of the layer network. This is used as the source or destination label described in clause 6.5.1.1.

Connectivity label: Identifies the access points between which AI symbols are transferred.

Resource label: This identifier allows the symbols belonging to one communication to be distinguished from the symbols that belong to another communication on the same link. This identifier is logically associated with the link not the symbols (or communications) being carried. It must be unique in (at least) the context of the link. This identifier must be encoded with the symbols (since by definition it is not local information). This label is used to identify the forwarding port (FPt) of a link connection. The resource label is frequently used as the identifier for the forwarding point (FP) that results from the binding of the link connection to a transport entity.

Forwarding end port identifier: This label identifies the FPt on a termination function. This identifier is frequently used for the forwarding end point (FP) that results from binding the termination to a transport entity.

Forwarding port identifier: Identifies the forwarding ports (FPt) on the boundary of a subnetwork. The identifier for a forwarding port must be unique within the context of the subnetwork.

Forwarding identifier: This identifier is used by the forwarding function to deliver a symbol from an ingress forwarding point to the appropriate egress forwarding point(s). The forwarding identifier is logically associated with a set of communications.

Link point identifier: Allows the links that terminate on a subnetwork to be identified. It is unique in the context of a subnetwork.

Subnetwork identifier: Allows the subnetwork within a layer network to be identified. It must be unique in the context of the containing subnetwork.

The use of these identifiers is described in Appendix I.

Annex B

Definition and properties of a system

(This annex forms an integral part of this Recommendation.)

B.1 Introduction

This annex provides a definition of a system and its properties. A telecommunications network is considered to be a system.

B.1.1 Definition of a system

A system delivers outputs by performing a prescribed function based on inputs to the system. It has input ports through which all inputs enter the system and output ports through which all outputs leave the system.

The specification of a system regards the inputs as independent variables and the outputs as dependent variables. In addition, a system may have internal dependent variables called state variables which will have initial values which are independent variables. The system itself is a transfer function where the output variables are an invariant function of the input variables and the state variables. Note that output variables may also be state variables. The state of the system is defined by the value of the state variables.

The value of the inputs can change over time and this will result in a change in the state of the system. The speed at which the change of state propagates through the system is finite and this sets limiting characteristics on the ability of a system to respond to changing inputs. A system is shown in Figure B.1.

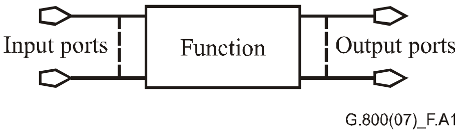


Figure B.1 – System

B.1.2 System binding

Systems can be connected together such that the output of one system feeds the input of another system. This is called a binding. A binding has the property that only one output port from one system is bound to only one input port of another system. A binding is shown in Figure B.2.

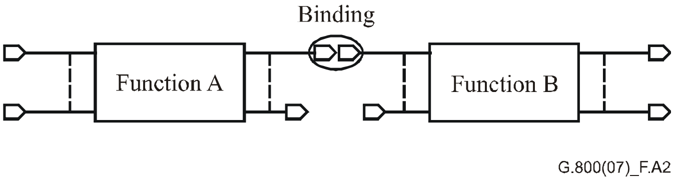


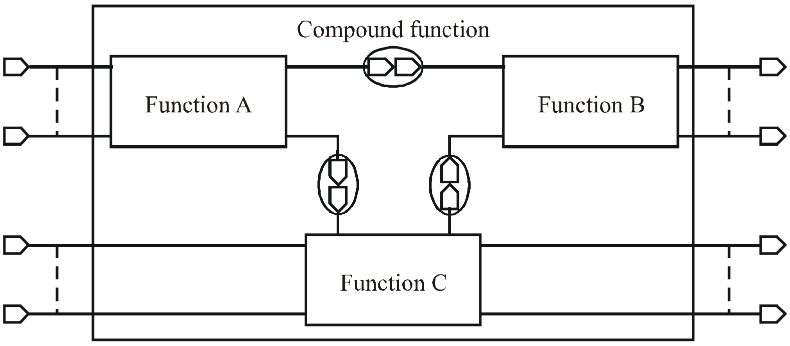
Figure B.2 – System binding

B.1.3 Compound system or aggregation

Binding is used to create more complex systems by combining a number of systems into a single aggregate system. Such a system is called a compound system, or equivalently, an aggregation. The constituent systems within the compound system are called subsystems. The process of binding functions into a compound system is called aggregation while the process of viewing the subsystems and their binding within a compound system is called decomposition.

The compound system is itself a system and has all the properties of a system. Similarly, a subsystem is a system and has all the properties of a system. Aggregation and decomposition are therefore recursive properties of systems.

Figure B.3 shows an example of a compound system.



G.800(07)_F.A3

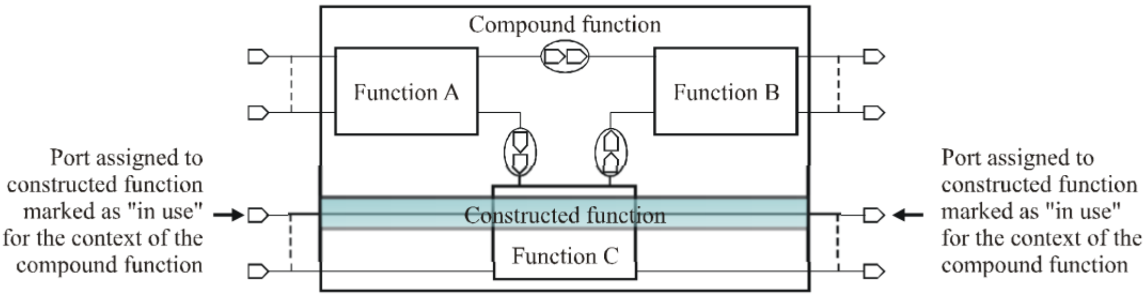
Figure B.3 – An example of a compound system or aggregation

Within a compound system, the bindings are static and become part of the transfer function of the compound system.

B.1.4 Configuration and construction of systems

The formal approach of systems includes the mechanism which we commonly call configuration and construction. Configuration of a system is the ability to take input information that is held in state variables and remains unchanged until such time as it is desired to reconfigure the system. These configuration state variables control the configurable properties of the transfer function. While a particular configuration persists, the state variables are effectively part of the transfer function.

Construction of a system is defined here as the re-interpretation of state variables as a transfer function. In this way inputs may be passed into state variables, thereby defining the transfer function. Formally, configuration and construction is the same thing. The re-interpretation of state variables as a transfer function must always take place in a context of constructing a system. Practically speaking, configuration is re-interpretation of state variables within the context of constructing a system which is already specific to the system which is to be constructed and normally configuration is about the specific purposing of a system. The construction of a system normally describes the re-interpretation of state variables as transfer functions across many subsystems within a compound system. This is illustrated in Figure B.4.


Compound function
Function A
Function B
Constructed function
Function C
Port assigned to constructed function marked as "in use" for the context of the compound function
Port assigned to constructed function marked as "in use" for the context of the compound function

G.800(07)_F.A4

Figure B.4 – Configuration/construction of a system

B.2 Information systems

An information system is a system that processes only information. The inputs and outputs to an information system are information; the state variables of an information system are also information, while the transfer function relates the state information and output information to the input information using an algorithmic state machine which can be described using a logical algebra.

B.2.1 Forms of information

While the variables of an information system and the transfer function of an information system are both information, there is an important difference between the information of the variables and the information of the transfer function. The values of the variables are open, and at the time of the construction of the system are essentially unknown to the system. On the other hand, the value of the information which defines the transfer function is known to the system at the time of its construction/configuration.

The form of information associated with the variables of an information system is information as defined by Shannon [b-Shannon]. In particular:

- Information is data with a certain semantic where data is a representation of a concept.
- The unit of data is a symbol.
- The enumerable set of symbols is the lexicon.
- Communication can be understood as the passing of information from one location to another.
- A message is an object of communication: It is something that provides information, it can also be information itself.
- A message is a sequence of symbols taken from an enumerable set of possibilities.

Note that it is possible for state variables to hold the previous symbols received on an input port. In this way, it is possible for an information system to operate on a sequence of symbols on an input port and be equivalent to a system which accepted a single larger symbol equivalent to the sequence of smaller symbols.

Information associated with the transfer function of the information system is unique for a particular system. This form of information is called algorithmic information and is in line with the definition set out by Kolmogorov [b-Watanabe]. Note that Kolmogorov information, as normally defined, describes only synchronous state information systems; however, the extension to asynchronous state information systems can in principle be made.

B.2.2 State synchronous and state asynchronous information systems

A system where, following a change in input variables, all consequent changes to state variables and output variables have occurred and stabilized within the system before any new change in input variables are accepted, is called a state synchronous system. The transfer function of a state synchronous system can be defined using a process calculus such as λ calculus and can be defined with a single thread of execution. It is possible for such systems to have flexible (i.e., input variable driven) and deterministic behaviour. The speed at which output variables are set in response to input variables is determined by the speed with which the system can synchronize the state and output variables.

A system where, following a change in input variables, some consequent changes to state variables or output variables have not occurred and stabilized within the system before any new change in input variables are accepted is called a state asynchronous system. The transfer function of a state asynchronous system can be defined using a process calculus such as π calculus and can only be defined with multiple threads of execution, for example as presented by Petri-Nets. It is possible for such a system to have unavoidable non-deterministic behaviour including "race" and "deadlock" conditions.

As a consequence of the above, when a compound system is made up from subsystems which are spatially dispersed and synchronization of state requires, at a minimum, speed of light delay between the subsystems, such a spatial-distributed compound system can be:

– Flexible in response to input variables, deterministic in its behaviour, but requires a minimum time between changes to inputs;

– Flexible in response to input variables, accept effectively instantaneous changes in input, state, and/or output variables, but have non-deterministic behaviour;

– Flexible in response to input variables, accept effectively instantaneous changes in input, state, and/or output variables, but cannot accept changes in input variables.

Any telecommunications network, by its very definition, is a system to carry information between spatial-separated end points, subjected to the constraints above.

B.3 Basic transfer function and telecommunications network

B.3.1 General

A telecommunications network is a system. The basic transfer function of the telecommunications network relays the sequence of symbols on a selected input port, and transfers them as the same sequence of symbols on selected output port(s).

In the following clauses, the information instance corresponds to a communication.

B.3.2 Transparency and labelling

An information system accepts a sequence of symbols through an input port. The behaviour of the system can depend on the sequence of the symbols – the sequence is part of the input information passed through the port.

However, an information system may have more than one input port. The behaviour of the system can depend on which port the sequence of symbols passes to the system – the selection of input port is therefore also part of the input information passed to the system.

Therefore, where the information system has more than one port, the selection of the input port must be included as part of the overall information. For an information system with one input port to be equivalent to an information system with many input ports, the input information must include labelling of the symbols which tells the information system the equivalent of which port each symbol enters the system. This is shown in Figure B.5.

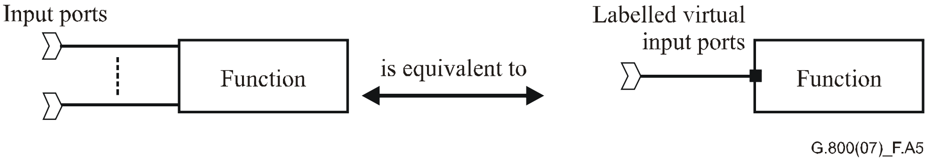


Figure B.5 – Multiple input ports as equivalent to labelling of input symbols

The purpose of a telecommunications network is to transfer information from an input port to an output port. Since the user is transferring information, we may assume that the user is sending information from one information subsystem to another information subsystem. Ideally, to the user of the service, the input port to the network should appear to behave as if it were the input of the user's subsystem at the far end of the network. This means that the sequence of symbols at an input port is to be replicated at an output port. The service is transparent when, given an agreement on the symbol size and the method of inter-symbol demarcation between the user and the network:

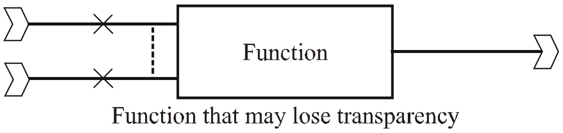
– the same sequence of symbols is output through the output port as sent through the input port;

  NOTE – Adding or deleting symbols will change the sequence of symbols.

– the user has complete freedom over the choice of symbols he/she sends and has also complete freedom in the meaning he/she attaches to each symbol in the set.

This implies that the network, for it to be transparent, must not add, subtract or alter symbols, nor must the network take any functional actions based on the value of any symbol other than those that are derived purely from the shared knowledge of the symbol set and inter-symbol demarcation.

Some subsystems may themselves not be transparent and lose information, for example by merging symbols from multiple inputs. An example is shown in Figure B.6.



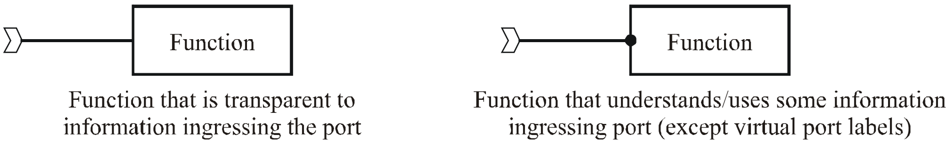
G.800(07)_F.A6

Figure B.6 – A system which loses information, for example by merging symbols from multiple input ports

B.3.3 Passing information to control an information system

As well as passing user information, information systems need to receive input information to construct, configure, and control its operation as well as issue output information for the construction, configuration and control of other information systems.

The distinction between input ports intended only to input user information from ports carrying construction, configuration and control information is shown in Figure B.7.


Function
Function that is transparent to information ingressing the port
Function
Function that understands/uses some information ingressing port (except virtual port labels)

G.800(07)_F.A7

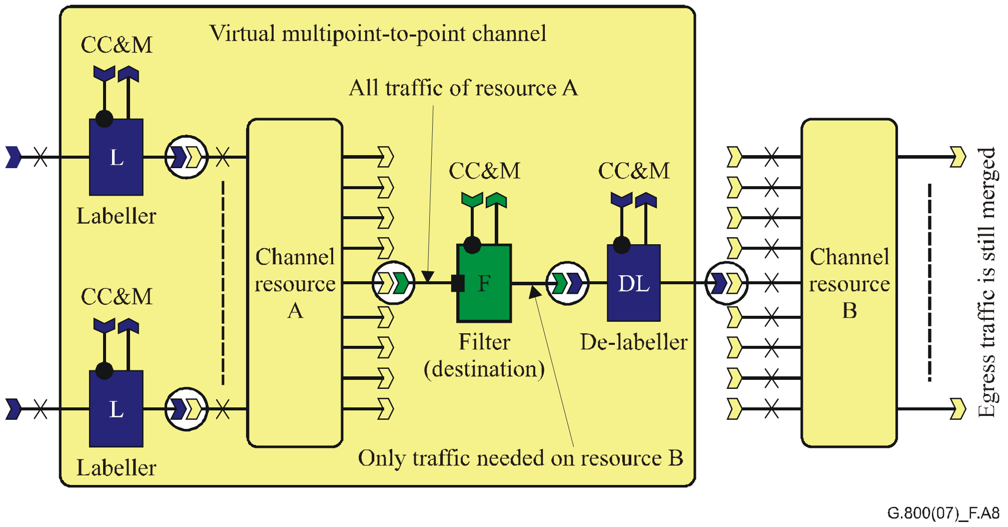
Figure B.7 – Inputs carrying user information and input ports carrying construction configuration and control information

B.3.4 Examples of forwarding and demerging

We can illustrate the use of the formal systems approach to two of the basic processes of the unified model – forwarding and demerging (i.e., demultiplexing).

B.3.4.1 Forwarding

The need for forwarding arises from the second axiom. As resources do not scale, it is necessary to filter only traffic which needs to use a resource onto the resource. Figure B.8 shows the basic operation of labelling and filtering which achieves this forwarding process when based on a multipoint-to-multipoint broadcasting channel.


CC&M
Virtual multipoint-to-point channel
All traffic of resource A
L
Labeller
CC&M
L
Labeller
Channel resource A
CC&M
CC&M
F
DL
Filter (destination)
De-labeller
Channel resource B
Egress traffic is still merged
Only traffic needed on resource B
G.800(07)_F.A8

CC&M Configuration, Control and Management

NOTE – It is possible to the scope of labelling to extend over more than one channel resource.

Figure B.8 – Forwarding using a multipoint-to-multipoint channel resource

B.3.4.2 Demerging

The demerging process is required to recover individual sequences of symbols after they have been merged with other sequences of symbols. The individual sequences must be labelled in order to overcome the loss of information inherent in the merging.

In the case where the information instance is an open sequence of symbols, the demerging process results in the construction of a point-to-point connection.

This is illustrated in Figure B.9 below.

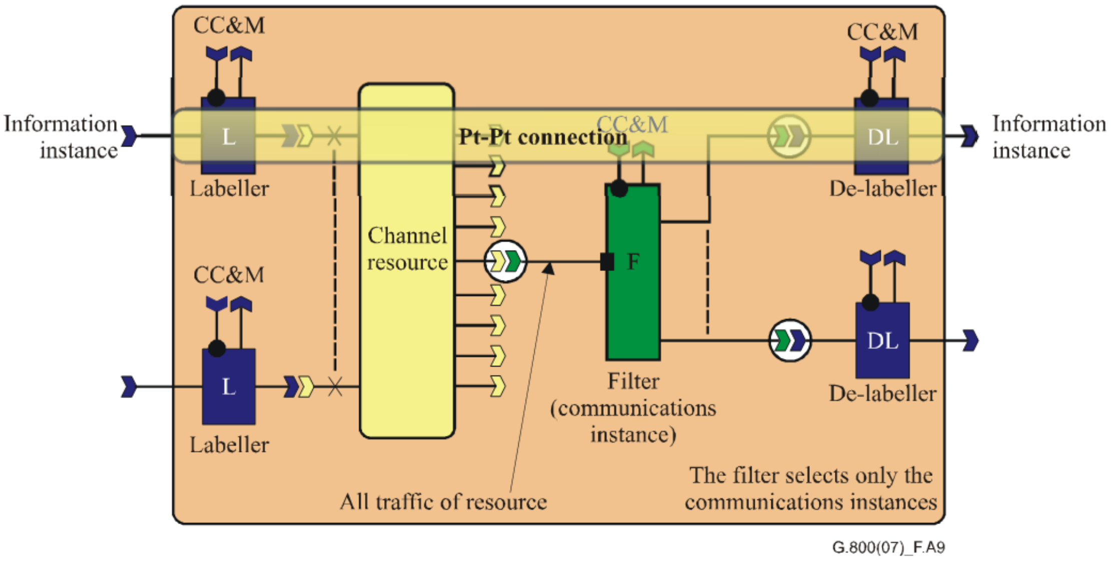


Figure B.9 – Demerging and the construction of a point-to-point connection

B.3.4.3 Connection-oriented and connectionless examples

In the above examples, a new channel function has been constructed by the configuration of labelling and filtering. In connection-oriented networks, the multipoint-to-point channels and the point-to-point connections are formed by the above construction processes as enduring constructions. In the case of connectionless networks, some of the channels are transient and only last for the duration of the connectionless packet. However, in both cases, symbols ultimately are transferred using channel forwarding within channels that are constructed by the configuration of labelling and filtering.

Annex C

Communication

(This annex forms an integral part of this Recommendation.)

The purpose of a network is to deliver communications among a number of parties. From a service perspective, the necessary specification is the set of all senders and all destinations, together with all the set of relationships that those senders and receivers can use to unidirectionally exchange communications amongst themselves. This specification is called a communications matrix. Bidirectional communication is specified by means of additional sources and destinations.

A communications matrix may be written as:

$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & \cdots & c_{1,n} \\ c_{2,1} & \cdots & \cdots & \cdots & \vdots \\ \vdots & & \ddots & & \\ \vdots & & & \ddots & \\ c_{m,1} & \cdots & \cdots & \cdots & c_{m,n} \end{bmatrix}$$

where $c_{i,j}$ is the communication between source S_i and destination D_j. In the general case, m does not equal n. For a communication, $c_{i,j}$ equals one, and for no communication $c_{i,j}$ equals zero.

Note that in the matrix, all desired communications are explicitly specified with a directionality from a source, such that:

{S_1 communicates with D_1} and {S_1 communicates with D_2} specify individual communications and do *not* specify a multicast communication, which must be specified explicitly as {S_1 communicates with {D_1 , D_2}}. Also note that a broadcast communication must be specified as {S_1 communicates with *all destinations*}. If a multicast or broadcast communication is to be supported, then it must be identified by an additional column in the communication matrix that identifies the set of destinations.

The communications relationship applies to points around a subnetwork and is the only property that can be stated about a service at the subnetwork level.

Each communication can be further specified by attributes (for example, delay, availability, bandwidth policy, etc.). These attributes can be attached to the communications relationship, and the meaning of such an attachment is that the attribute applies to *all* communications. This indicates how attributes fit into the model.

It is also possible to attach relationship attributes such as "relationship name."

C.1 Service configuration

The configuration of service proceeds in steps.

The first step, usually associated with routing, chooses the links and inner subnetworks that will be used to instantiate the service. This step also transforms the original communications matrix into appropriate individual sub-matrices, one for each inner subnetwork.

The second step, often associated with provisioning, applies each sub-matrix to its associated inner subnetwork and may reserve or allocate (link) a resource to the service instance. Note that reservation information need not necessarily be kept on any network element.

This process continues recursively until configuration is complete. Note that communications attributes may affect how and when resources are reserved, and these attributes may affect the inner structure that is selected.

A transport entity is created when a communication matrix is used to configure a forwarding function (in a subnetwork).

Diagrammatically we have

$$\{ S:D \} \text{->} \{ \{ S:D \}_1 , \{ S:D \}_2 \ ....,\{ S:D \}_n \ \}$$

where { S:D } applies to the subnetwork and $\{ S:D \}_n$ applies to the n^{th} contained subnetwork.

Note that $\{S:D\}_1$ is transformed from { S:D } and the result of the transformation depends on the actual links and physical locations of the S_1..j. This transformation is not a decomposition, and cannot be reversed. That is, {S:D} cannot be derived from inspection of $\{S:D\}_1$ alone.

As more demands are put on the network, and each demand is transformed into an $\{ S:D \}_{n,}$ the transform matrices become combined, making it even less likely that {S:D} can be derived from inspection of $\{S:D\}_1$ alone.

The communications matrix defined is necessary to fully specify the service in terms of allowed communications, but it is not sufficient to fully dimension the internal link resources. Additional parameters may be attached to each communication in order to specify additional properties, such as traffic profile and importance.

C.2 Forwarding modes of a transport entity

Channel based forwarding supports 1:all communications, with 1:1 as a special case. Destination based forwarding supports 1:n communications with 1:1 and 1:all as special cases.

	Destination forwarding	**Channel forwarding**
Point-to-Point	As a special case of 1:n	Only for a single egress
Point-to-Multipoint	Required at a minimum of one subnetwork	May or may not be present in network for supporting single egress
Multipoint-to-Multipoint	Required at a minimum of one subnet	May or may not be present in network for supporting single egress
Full broadcast	As discussed in clause 6.3.1	

C.3 Monitoring of a transport entity

The transport entity, derived from a communications matrix, supports a set of senders and receivers that exchange information.

It may be possible to identify when a member sender or receiver leaves or a non-member sender or receiver joins the transport entity.

Individual pair-wise communication $\{ S_j \text{->} D_k\}$ may be monitored.

If the communications matrix only specified a single sender and receiver, then the individual communication is the complete specification of the entire transport entity.

Annex D

Transport entity roles

(This annex forms an integral part of this Recommendation.)

The objective of describing the roles is to provide a consistent description of how transport entities are being used within a network, i.e., whether they are being used to directly support a customer service, or to provide aggregation within a transport network, or to provide direct connectivity between points of flexibility. Examples of the use of roles are provided in Appendix V.

Use of the roles is optional within an operational domain.

D.1 Overview of transport entity roles

An operational domain is a collection of transport resources falling within the scope of control of a single operator or administrative entity within an operator. The transport service demarcation point between operational domains occurs on a link (the link may be virtual) between the operational domains and is a user to network interface (UNI) or external network to network interface (ENNI) depending on the service relationship between the domains.

Transport networks are built to provide transport services, i.e., connectivity service for customers' networks. These provide topology for a customer network.

A transport entity's role relates to the purpose it serves in an operational domain. Roles are defined from the perspective of a single operational domain and an instance of a transport entity (TE) may play different roles in different domains. Roles apply to TE instances, not to complete layer networks. Three roles are described in this clause.

D.2 Roles of the transport entity within an operational domain

D.2.1 Role of the virtual channel

The role of the virtual channel (VC) is to provide connectivity for a single service instance within an operational domain.

To provide topology for a customer's (client) layer network the (server) trail terminations supporting that topology must be in the customer's network. Therefore the transport network provides a service in the form of a subnetwork connection (SNC) or subnetwork transport entity (SNTE) provided between the demarcation points at the domain boundary.

At the granularity of individual services there are two ways an operational domain may support a service instance. If the transport service provider operates a network in the same layer the customer wants to interconnect, the VC SNC/SNTE may be switched natively. If greater transparency than that provided by a native SNC/SNTE is required then the customer's SNC/SNTE can be adapted (1:1) to a server transport entity in a layer that the service provider switches. A service instance may be carried natively in one part of an operational domain and adapted (1:1) to a server TE in another part of the domain, or using different server layers in different parts of an operational domain.

The simplest and most common service is a single SNC or SNTE; however, there are also cases where the customer service is a set of transport entities that are intended to be managed as a single service instance. These are called bundled services (the "bundle" being the set of TEs) and the VC comprises the set of TEs. Treating a set of TEs as a single service instance can present challenges for service instance monitoring because there is not a simple operational state for the set.

D.2.2 Role of the virtual path

The role of the virtual path (VP) is to provide connectivity for multiple service instances within an operational domain. The operator of the operational domain has control of the route and extent of a transport entity that is playing the VP role.

The use of virtual paths is at the discretion of the service provider so zero or more VP levels may be employed in a transport network. A VP may carry a combination of VCs and other VPs.

D.2.3 Role of the virtual section

The role of the virtual section (VS) is to provide connectivity directly between subnetworks.

Transmission of information between subnetworks requires a section layer (i.e., a layer comprising all the functions which provide for the transfer of information between locations in a path layer network). A TE playing the virtual section role provides connectivity between subnetworks operating on TEs that support the VC and/or VP roles.

D.3 Roles of the transport entity across multiple operational domains

Different segments of a transport entity (TE) may play different roles if the transport entity spans multiple operational domains. For example a TE that is playing the VP or VS role in one domain may play the VC role in another domain.

Appendix I

Use of identifiers

(This appendix does not form an integral part of this Recommendation.)

I.1 Use of identifiers

At the boundary of the layer network, the termination function adds the required fields to the adapted information (AI) symbol presented at the access point to allow this layer network to insert the identifiers that it requires. Note that in some cases, the encoding of the resource label is provided by the characteristics of the server layer network, e.g., a timeslot in an SDH frame.

The text below describes the insertion, removal and use of each identifier with the assumption that each identifier is encoded into an independent field in the layer network LI. In most practical layer networks, the same label field within the LI is reused for several purposes. In general, any identifier with a scope that is equal to or greater than the scope required for the purpose may be reused. When an existing identifier is reused, the insertion process is null and the removal process is implemented as a read.

Resource label: This is used to deliver symbols between the ingress and egress forwarding points in the context of a link. It is injected by the client to server adaptation function[6] at the ingress link point and removed by the server to client adaptation at the egress link point. It allows the symbols to be delivered to the correct egress forwarding point. The scope of this identifier must be (at least) the link point on the subnetwork and must be large enough to distinguish all of the link connections that appear within that link point.

NOTE 1 – Examples of a resource label are: a timeslot in an SDH, an ATM VPI or VCI.

Forwarding point identifier: This identifier is used by the forwarding function in a subnetwork to deliver a symbol from the ingress point to the egress point (or points) of the transport entity. Note that the resource label of the link connection is commonly used for this identifier.

Forwarding identifier: This identifier is inspected at the ingress of a transport entity. It provides an index to an entry in the forwarding table (within the forwarding function) that identifies the target egress FP (or FPs). Typically a forwarding identifier that targets a specific access point (or set of access points) is used. The message could be broadcast to all of the access points in the layer network and discarded if the destination does not match the target. However, this is an inefficient use of the layer network resources. The path that a specific message (communication) will take across the network is not pre-provisioned for each (potential) communication. The commonly used forwarding identifier is an address (with network wide scope). However, it is also possible to use a sequence of next hops (e.g., egress link point). These semantics impart different network properties. For example, if an address is used, even under fault conditions, delivery may still be possible and misdelivery is impossible. A sequence of next hops does not have these properties.

The forwarding identifier may also include other information that is encoded in the LI, e.g., type of application, QoS/Priority.

Address: This has a network wide scope and identifies the destination access point. This identifier is carried within the LI and is inserted by the termination at the ingress to the layer network.

NOTE 2 – In general, the address is a subset of the information contained in the connectivity label.

NOTE 3 – This may be via a gateway if address translation is used along the path.

[6] In some cases, the value of the resource label value may be modified by an intra layer transport processing function.

NOTE 4 – A multicast address is an alias to a set of destinations. A destination may appear in more than one multicast address group, also the composition of a multicast address group may change.

Sequence of next hops: Each entry (i.e., link point and forwarding point) in the list has limited scope (target subnetwork) and is carried by the LI. It is inserted by the termination at the ingress to the layer network as an explicit route, i.e., a sequence of the subnetworks that the message will transit.

– The next hop may only define the egress link point, this allows the selection of a specific forwarding point to be determined locally.

– In hybrid cases, the LI may contain both the destination address and the "next hop" or "sequence of hops" to allow intermediate nodes to compute the path.

Communication instance identifier: The instantiation of this identifier can be split into two cases depending on the topology:

– Point-to-point: The topology of the transport entity provides a 1:1 relationship between the source and destination. Therefore, communication instance can be identified by the relationship between the access points in the control/management plane. The communication instance identifier need not be present in the LI.

– Multipoint-to-point or multipoint-to-multipoint: The connectivity of the transport entity provides an n:1 relationship between the source and destination access points. This arrangement does not allow a specific source to be identified. Therefore, the transport plane must carry a connectivity label. Typically the transport plane identifier for the source and destination access points are used for this purpose. However, an explicit route (to the destination) and a source access point identifier may also be used.

NOTE 5 – In the case of MPLS, since communications from multiple sources may share the same mp2p transport entity and a link scope resource label is used, the connectivity label is not supported.

Access point identifier: This identifier is used by both the client layer and the server layer. The client layer uses the access point identifier to indicate to the server layer the intended destination for the communication.

Use of the access point identifier within a layer network can be split into two cases depending on the type of forwarding used:

– Channel forwarding: The access point identifier is only used by the routing process in the control/management plane to configure the channel forwarding transport entity. It need not be represented as a label in the LI.

– Destination forwarding: The transport plane identifier for the destination access point is used in different ways: As (part of) the connectivity label and either, (part of) the forwarding identifier; or to compute a route from the source. This identifier must be passed to the server layer as a parameter across the access point along with the AI symbol.

Appendix II

Relationships between architectural entities

(This appendix does not form an integral part of this Recommendation.)

This appendix describes the relationships between the architectural entities described in this Recommendation, and those described in [ITU-T G.805]. These are described in Table II.1.

Table II.1 – Relationships between architectural entities in ITU-T G.800, ITU-T G.805

Unified architecture (ITU-T G.800)	**ITU-T G.805**
Topological Components:	
Layer network	Layer network
Subnetwork	Subnetwork
Link	Link
Access group	Access group
Transport entities:	
Access transport entity	Trail (in the pt-pt case)
Channel forwarding transport entity (single source)	Subnetwork connection
Channel forwarding transport entity (multiple sources)	Not applicable
Destination forwarding transport entity	Not applicable
Link connection	Link connection
Transport processing functions:	
Adaptation	Adaptation
Termination	Trail termination
Layer processor	Not described
Forwarding	Not described
Reference points:	
Access point	Access point
Forwarding point	Connection point
Forwarding end point	Termination connection point

Appendix III

Complexity and scalability of systems

(This appendix does not form an integral part of this Recommendation.)

Many human endeavours, including engineering, show characteristics of complexity. 'Complexity' has many definitions; however, the definition used in this Recommendation is the following:

- *Complexity* is the ratio between the number of interactions between individuals and total number of those individuals.

If this ratio grows with the number of individuals, then the endeavour can be said to become complex as it gets bigger. This can be seen in many diverse places, including business organizations, industry organizations, systems of government administrations, as well as in engineering.

In engineering, the discipline of systems engineering has evolved to deal with complex engineering problems and, in the language of systems engineering, we call these *complex systems*. There are two important properties which characterize many complex systems, with some systems exhibiting one or other characteristic and some exhibiting both characteristics:

- *Chaotic behaviour* – Complex systems can have unexpected, emergent behaviour that often runs counter to the basic objectives of good engineering where predictable behaviour is essential – we expect that the response of a bridge to a cross wind has been correctly predicted when we drive across the bridge.
- *Non-linear cost of expansion* – In a complex system, when there is an inherent cost of interactions between individuals, the costs of the overall system can grow more than linearly as the system is expanded.

Generally speaking, complexity is an undesirable characteristic for a system. The discipline of systems engineering largely deals with the good engineering practice for managing the development of systems which have a given and unavoidable level of complexity.

However, the complexity of a system is often a matter of choice. One particular system architecture may be highly cost effective at a small scale; however, if the system needs to be expanded, its inherent complexity means that costs will increase dramatically with the expansion. An alternative system architecture may be more expensive at small scale; however, because it has less inherent complexity, the increase in costs when it is expanded is much lower.

Although chaotic behaviour can be beneficial in a very few systems, it is highly undesirable in the engineering of the great majority of systems. At a minimum, chaotic behaviour can increase the overall costs as the final costs of mitigating the unexpected behaviour, many of which may not have need expected at the design stage, must be added to the overall system. A minimum consequence of chaotic behaviour is to increase the non-linearity of the cost of expansion.

Frequently, the practical consequence of complexity within a system architecture is the scalability of the system. The scalability can be usefully defined as the following:

- The *scalability* of a system is the power exponent of the total costs as a function of expansion. Generally speaking, a power exponent of 1 (or suitably close to 1) is called *scalable* while an exponent significantly greater than 1 is said to be *non-scalable*. The *scaling limit* of a system is said to be the scale at which the exponent becomes significantly greater than 1.

Achieving an architecture for a large system which is scalable, and which does not have a clear scaling limit, is normally not straightforward. Generally, scalable systems arise when close attention is paid to the way in which interactions grow as the overall system grows.

III.1 Independence of subsystems

In a system comprised of subsystems, complexity arises out of the interactions between subsystems. If we follow the consequences of a scalable system, we can see that:

- When a new subsystem is added to a system, for the system to be scalable, the number of interactions generated by the new subsystem must be fixed with the new subsystem and not depend on the total number of subsystems in the overall system.

This observation leads to an analysis of dependencies between subsystems. A subsystem has a dependency on another subsystem if it has any interaction with it. Analysis of dependency is one of the features built into modelling languages, notably the unified modelling language (UML).

Once a dependency has been identified, the nature and frequency of interaction can be categorized. Examples include the following:

- A dependency between subsystems may exist only to manage the process of adding and deleting a subsystem.
- A dependency between subsystems may exist for occasional ad hoc interactions; for example, the dependency between an individual web browser and an individual web server.
- A dependency between subsystems may require time-critical, high-volume, state-locked interactions.

However, a simple conclusion of scalability is that for a large system to be scalable, most subsystems must work independently of each other. Maximizing independence between subsystems is at the heart of scalability.

III.2 Independence within a lifecycle and between lifecycles

The engineering of a system involves a lifecycle which broadly comprises – requirements capture, architectural design, component design, component development, deployment, live operation, operational support and repair, and decommissioning.

The discipline of systems engineering has given considerable attention to managing the dependencies between these different stages in the engineering lifecycle. This is generally a *forward dependency*. For example, live operation and operational support and repair may well depend on decisions made in design stages. In this case, the observation has been that complexity has been generated by assuming perfection in the early stages with an assumption that there are no forward dependencies but, in practice, this turned out not to be the case. The development process associated with this is often referred to as the 'waterfall' method. In practice, some dependency is largely unavoidable so complexity is managed better by acknowledging and planning for the dependencies. This has led to the current systems engineering development process of 'iterative' cycles.

Importantly for telecommunications, the live operation phase cannot be readily 'turned off'. This means that when a new system is introduced, it must be integrated into existing systems – the old and the new form one large super system. This means that there are now not merely dependencies between the stages of a particular development lifecycle, but that there are dependencies between different lifecycles. This can take many forms. Examples include the following:

- Between the architectural design stages of each lifecycle, there is a new set of potential subsystem dependencies between the subsystems of each development.
- The live operation of the existing system has a dependency on the architectural design of the new system. The design of the new system may well affect the level of operational disruption caused to the existing system when the new system is deployed.
- There may be a dependency between the deployment of the new system and the decommissioning of the old system.

III.3 Transparency of telecommunications network services

Telecommunication systems introduce a further aspect of dependency. Broadly speaking, this is "when a client uses a server layer network, is there a dependency between client and server?" We can consider this in two stages: first, clarify what is meant by dependency and what is independence in the specific case of telecommunications; second, examine specific examples to show the extent and characteristics of dependency between client and server.

When a client uses a network to transfer information between end points, as set out in the axioms of ITU-T G.800, there is one basic independency setup and five basic forms of dependency setup.

- Symbol selection independency – this is the extent to which the client can select symbols from a lexicon at will without creating any dependency interaction in the server. This is defined as the *transparency* of the server.
- Lexicon dependency – this is the delineation of traffic units which the client wishes to transfer. The client has full freedom and independence in the selection from the lexicon but the lexicon itself and the demarcation between sequenced symbols from the lexicon are a dependency between client and server.
- Attachment control information dependency – this is the interaction between client and server to establish an access point to the server network and assign it an address which is shared information.
- Communication control information dependency – this is the interaction between client and server requesting a particular transfer of information between access points. This normally involves the client passing the server a set of destination addresses and a source address which may be implicit by the location of the request.
- Performance control information dependency – this is the interaction between client and server indicating the performance requirement and its scope may vary, for example, the scope may be a communication or it may be a symbol/traffic unit.
- Transfer performance dependency – the success of the communication cannot be totally guaranteed and so there is a performance dependency setup between client and server.

Having established these dependencies, it is possible to characterize different forms of a server network according to the characteristics of these dependencies. Some examples are illustrated in the table below.

Table III.1 – Examples of dependency for server layer networks

	Independency	Dependency				
	Transparent symbol selection	Lexicon and symbol demark	Attachment control information	Communication control information	Performance control information	Transfer performance
Fixed rate leased line	Binary bit	Unit interval	At 'subscription' time	At 'subscription' time	None	EP, AP, SR
Packet PVC with uniform diffserv model	Packet payload	Frame length field	At 'subscription' time	At 'subscription' time	Code point with every packet	EP, PL, AP, SR
PSTN	Analogue amplitude/ time quanta	Real time	At 'subscription' time	With every connection request	None	Noise, AP
Public Internet	Packet payload	Frame length field	At 'subscription' time (may be PPP over server connection)	Destination address with every packet	None	EP, PL, AP
Private IPVPN service with uniform diffserv model	Packet payload	Frame length field	'Subscription' time interaction per end point of VPN	Destination address with every packet	Code point with every packet	EP, PL, AP, SR
EP – error performance; PL – packet loss; AP – availability performance; SR – shared risk of common mode failure between sets of communications.						

As can be seen, the number and rate of interactions associated with the dependencies vary greatly. The choice of level of interaction for any client/server relationship will inevitably affect the overall scalability and complexity of a telecommunications network.

It is also possible to include further dependencies. The server can use further information from the client, for example to control routing choices. However, three factors must be considered when constructing further dependencies between client and server layers.

- This inevitability increases the basic complexity of the overall telecommunication network and may well fundamentally undermine scalability.
- This is likely to generate an inter-lifecycle dependency between the development lifecycle of the client and all its possible servers as well as between the server and all its possible clients. This may have profound consequences for the complexity of evolution for the overall network.
- If the dependency is based on symbol selection over which the client thought it had full and independent choice, then the introduction of this feature fundamentally reduces the transparency offered by the server to the client.

Appendix IV

Access points in architecture and equipment Recommendations

(This appendix does not form an integral part of this Recommendation.)

This appendix describes the relationship between access points as defined and used in functional architecture Recommendations (based on Recommendations ITU-T G.800 and [ITU-T G.805]) and access points as used in equipment Recommendations (based on [ITU-T G.806]).

[ITU-T G.805] defines the access point (AP) as the binding between an adaptation function and a termination function, but does not explicitly define principles for the division of responsibility between these functions. Adapted information (AI) is defined as being "transferred on trails", where a trail is a transport entity that provides transparent information transfer between layer network reference points. This implies that AI comprises the adapted client layer characteristic information that is transferred transparently from one AP to another.

ITU-T G.800 further explores and defines the relationship between layer networks on the basis of information dependence. In this Recommendation the AP resides between the client dependent adaptation function and the client independent termination function of the server layer network. This defines the information boundary for the server layer network at which AI is transferred to and from a trail. However, there may be information that is processed or held by the adaptation function that is also processed in the server layer network (layer information or LI). This information may (or may not) be transferred from one AP to another, but this transfer is not transparent due to the use of the information by the server layer. LI passed at the AP is called "parameters" to distinguish it from the AI defined in [ITU-T G.805] and to indicate that the server layer defines the information encoding.

Note that it is straightforward to recognize layer information and parameters. If information is read or modified by the server layer, then it is layer information. If that information has a dependence on client layer adaptation, then it is a parameter and is passed across the access point without being embedded in the AI. The initial circuit technology architecture models developed using [ITU-T G.805] did not provide detailed information structure models and did not distinguish parameters from AI at an AP. The need for this distinction became more apparent with packet technologies and, in particular, MAC bridging technology which has a rich layer network information structure.

[ITU-T G.806] defines a modelling methodology for equipment models. In this methodology (building on [ITU-T G.805]) AI is redefined as "The information passing across an AP" which includes all information passing this reference point, regardless of whether it is transferred across the trail. This methodology also defines the functional division between adaptation and termination based on type of function, assigning only monitoring functions to termination and all other functions to adaptation. This leads to a different position for the AP in [ITU-T G.806] models from that in ITU-T G.800 models. This difference is shown in Figure IV.1.

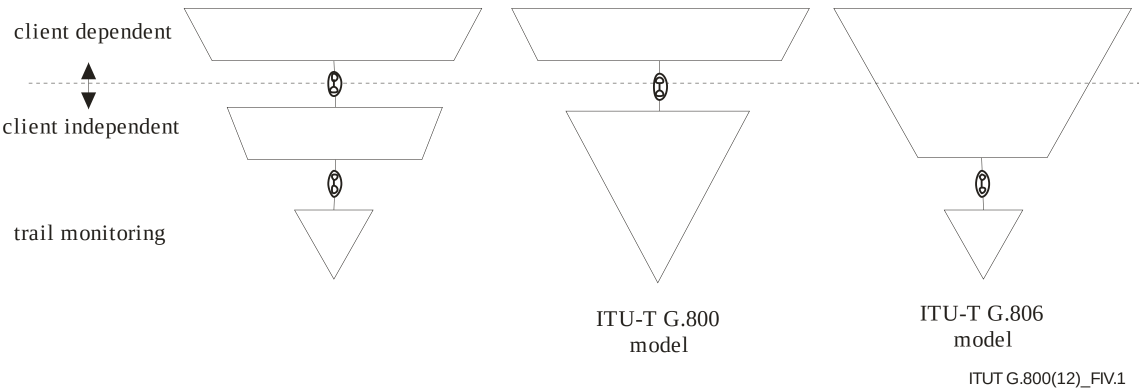


Figure IV.1 – Relationship of access points to functional partitioning

This difference in the location of the AP and definition of AI does not result in conflicting functional models; however, the models may appear to differ since they use the same terms in different ways. Understanding the relationship between these terms as used in architecture and equipment Recommendations clarifies how the models are aligned.

Appendix V

Examples of roles of the transport entity

(This appendix does not form an integral part of this Recommendation.)

V.1 Examples of roles of the transport entity (TE) within an operational domain

V.1.1 Switched-service instance (virtual channel)

A transport entity providing connectivity for a single-service instance within an operational domain, plays the virtual channel (VC) role. For example, an Ethernet S-VLAN providing a point-to-point port-based service between an Ethernet UNI and an Ethernet over an SDH NNI is shown in Figure V.1. This is an example of a natively-switched service.

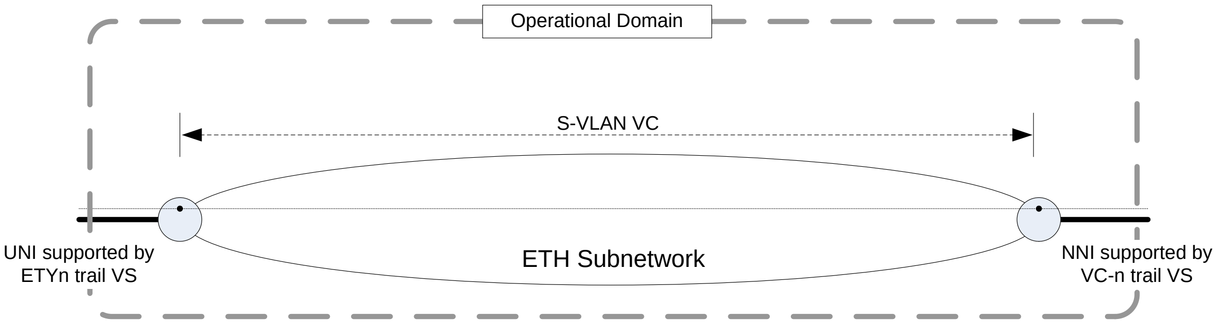

Operational Domain
S-VLAN VC
UNI supported by ETYn trail VS
ETH Subnetwork
NNI supported by VC-n trail VS

Figure V.1 – Example of virtual channel (transport-service instance)

In some cases, the transport network does not switch at the layer of the service to be provided. For example, Figure V.2 shows an ODU transport network providing a transparent Ethernet service (ETCn). The service signal must be adapted at the network edge to an ODUflex signal that can be switched in the ODU subnetwork. Thus, the VC managed in the transport network is the ODUflex and the ETC connection is only accessible at the domain boundaries where it is mapped between an ETYn UNI and the ODUflex VC. This is an example of a mapped VC service.

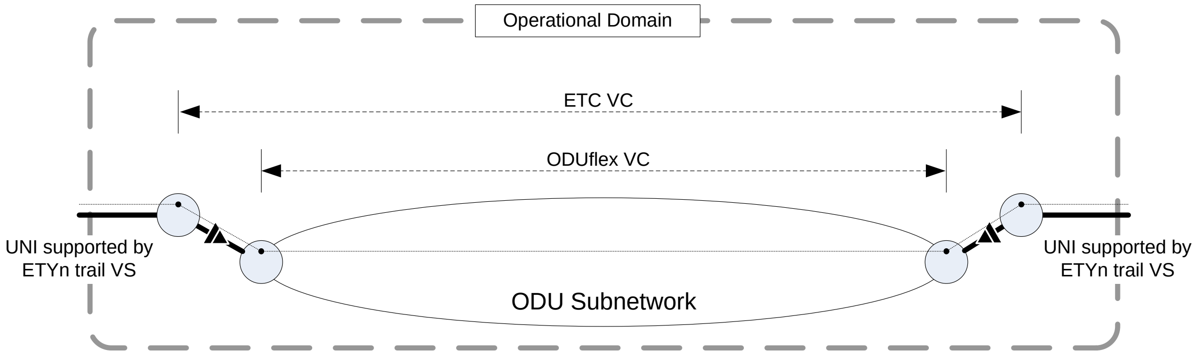

Operational Domain
ETC VC
ODUflex VC
UNI supported by ETYn trail VS
ODU Subnetwork
UNI supported by ETYn trail VS

Figure V.2 – Example of mapped virtual channel

V.1.2 Service aggregation using a single switched TE (virtual path)

A transport entity within an operational domain that is providing connectivity for multiple service instances plays the virtual path (VP) role. Figure V.3 shows an ODU3 VP carrying three VCs, two GE mapped to ODU0, and one STM-64 mapped to ODU2.

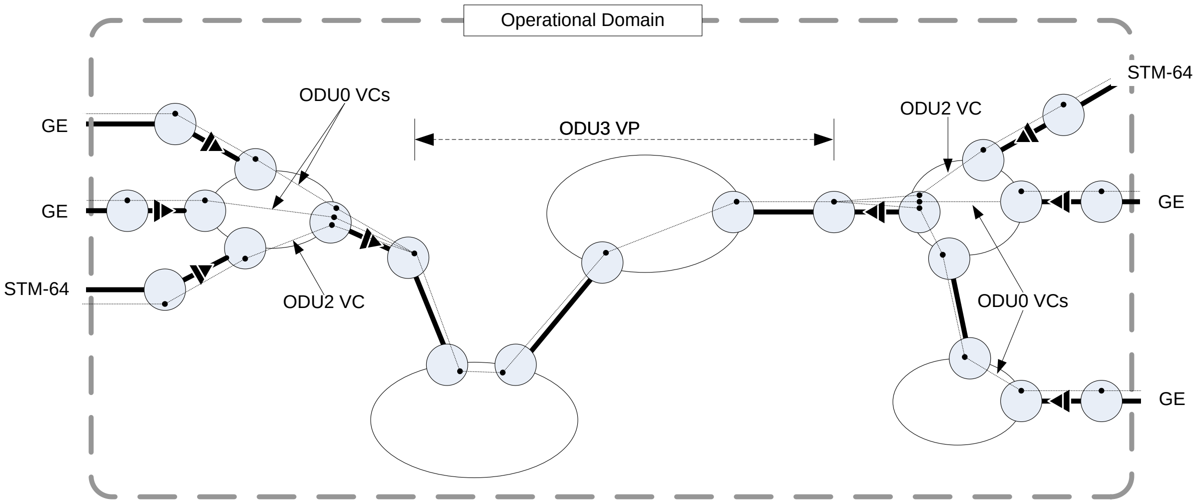


Figure V.3 – Example of virtual path (aggregation of service instances)

V.2 Roles of TE's that span multiple operational domains

Different parts of a network transport entity (NTE) may play different roles if the NTE spans multiple operators' networks (or multiple administrative domains in one operator's network).

V.2.1 TE roles in two operational domains

A physical section interconnecting elements in two different operational domains supports a network interface (NI) as illustrated in Figure V.4. Depending on the relationship between the operators, this may be a user to network interface (UNI) between a service provider and his customer, or a network to network interface (NNI) between peering network operators or peering domains within one operator's network.

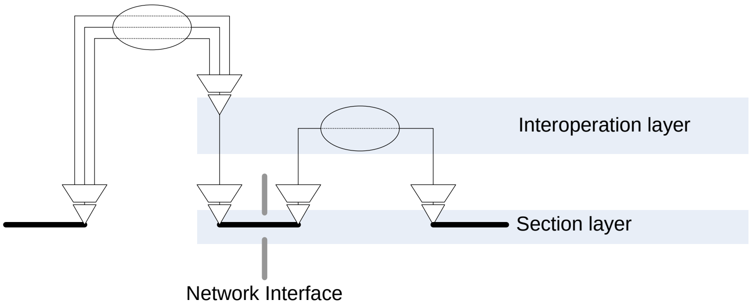


Figure V.4 – Network interface

Above the physical section at an NI, there must be one or more TEs that both operators manage and that extend beyond the NI in at least one of the operational domains. These TEs form the "interoperation layer" for the NI[7]. The role played by an interoperation TE (ITE) may be different in each operational domain. For example, the ITE may be a VP in one operational domain and a VC in the other. If one network operator uses the services of another network to interconnect switches, the ITE may be a VS in one operational domain and a VC in the other. For example, Figure V.5 shows an ODU3 that plays the VS role in operational domain A and is a VC in operational domain B.

[7] There may be zero or more multiplexing or adaptation layers between the VS and interoperation layers at an NI.

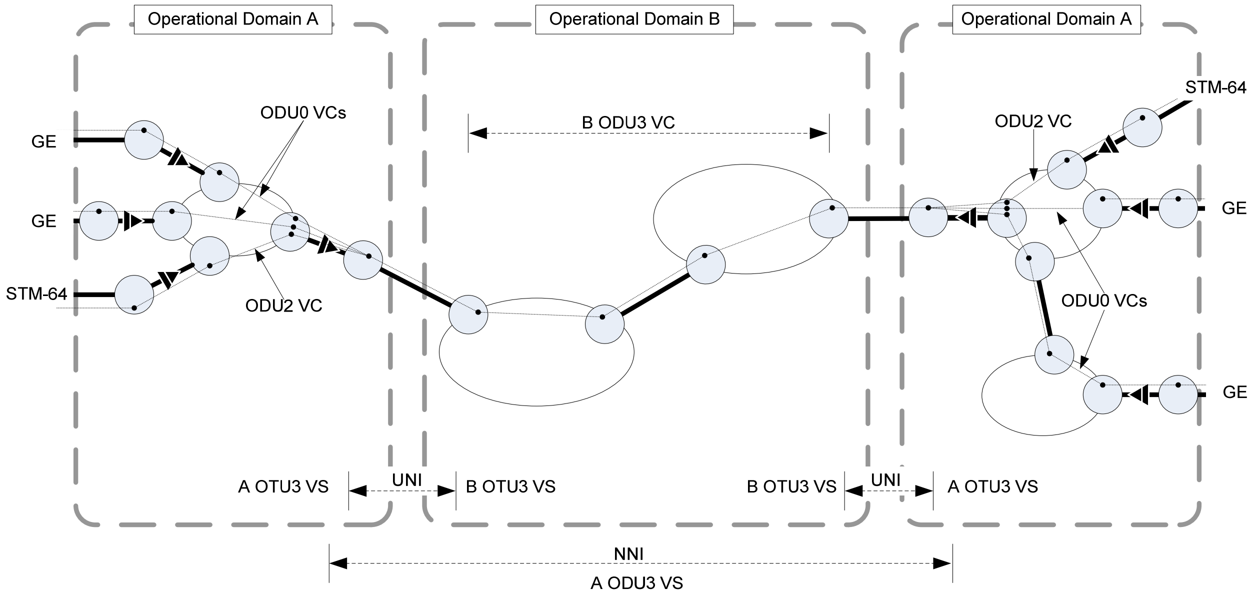


Figure V.5 – Two operator example

V.2.2 TE roles between three operational domains (remote network interface)

A more complex arrangement may exist between three network operators: a service provider, their customer, and an access provider in between. There are multiple arrangements possible in this situation. An example is shown in Figure V.6 of Ethernet Remote Customer Service Interfaces (IEEE Std 802.1Qbc^{tm}-2011). In this example, multiple S-VLAN TEs supported by a single physical section (NNI) are used to provide virtual sections (virtual UNIs) connecting operational domain A with customer domains C1, C2, and C3. These S-VLAN TEs are VCs in operational domain B, an access service provider in this example, and VSs to operational domains A, C1, C2, and C3. Operational domain A terminates these S-VLAN TEs and provides services (S-VLAN VCs) at the (virtual) UNI provided by each S-VLAN VS supported by the NNI.

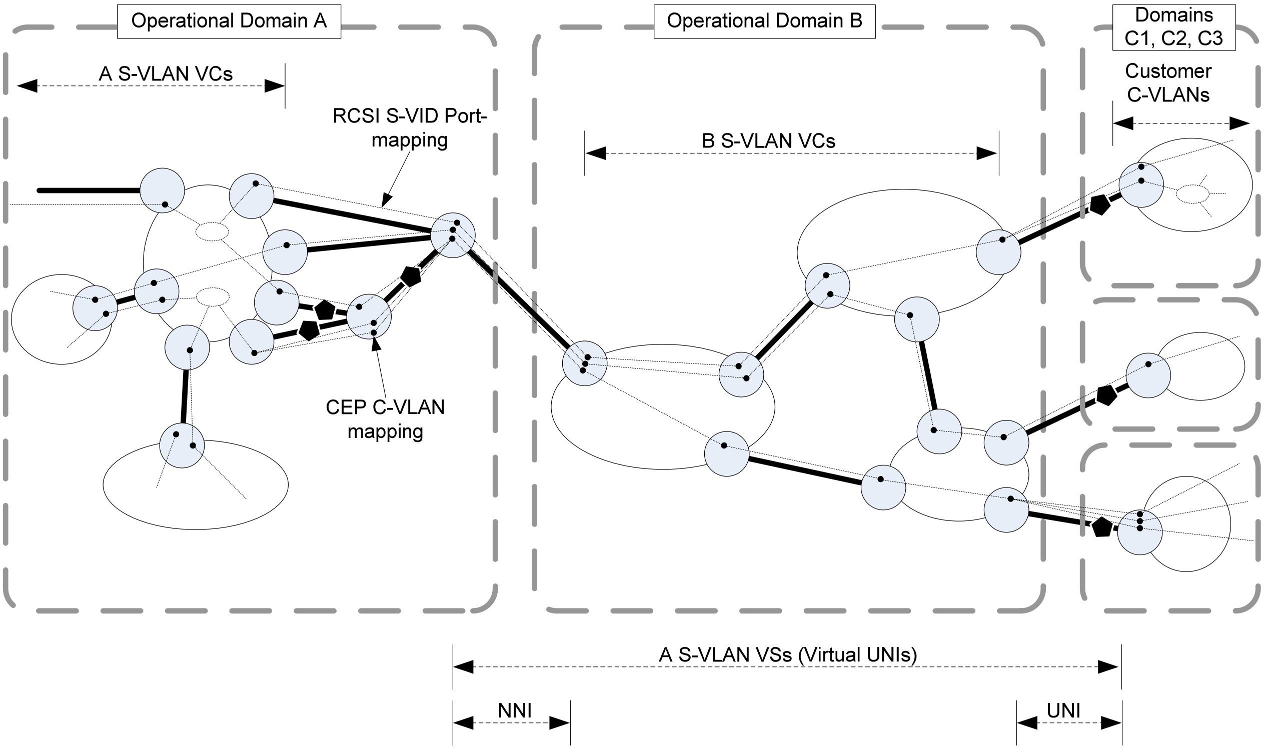


Figure V.6 – Three operator example – RCSI

Bibliography

[b-Shannon] Shannon, Claude E. (1948), *A mathematical theory of communication*, Bell System Technical Journal (July and October), 27, pp. 379-423 and 623-656.

[b-Trudeau] Trudeau, R.J. (2004), *Introduction to Graph Theory*, Dover Publications.

[b-Watanabe] Watanabe, O. (1992), *Kolmogorov Complexity and Computational Complexity*, Springer Verlach, October.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	**Transmission systems and media, digital systems and networks**
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2012