International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.1401

(02/2008)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS

Internet protocol aspects – Interworking

Principles of interworking

Recommendation ITU-T Y.1401

ITU-T

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	**Y.1400–Y.1499**
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Numbering, naming and addressing	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899

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

Recommendation ITU-T Y.1401

Principles of interworking

Summary

Recommendation ITU-T Y.1401 provides an architectural framework and general principles for transport stratum interworking in a next generation network (NGN) environment. It describes client/server and peer-partition interworking.

Source

Recommendation ITU-T Y.1401 was approved on 29 February 2008 by ITU-T Study Group 13 (2005-2008) under Recommendation ITU-T A.8 procedure.

Keywords

CAL, client/server, interworking, peer-partition.

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/.

CONTENTS

Recommendation ITU-T Y.1401

Principles of interworking

1 Scope

This Recommendation provides an architectural framework and general principles for transport stratum interworking in an NGN environment. It does not attempt to provide details for specific technology scenarios. Rather, it provides fundamental concepts and principles with illustrative examples. Specific scenarios are covered in other Recommendations.

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.809] Recommendation ITU-T G.809 (2003), *Functional architecture of connectionless layer* networks.

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

[ITU-T Y.1314] Recommendation ITU-T Y.1314 (2005), *Virtual private network functional decomposition*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 client/server relationship [ITU-T G.805]: 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.

3.1.2 logical external network-network interface (reference point) [ITU-T G.8080]: The E-NNI is the reference point between domains.

3.1.3 logical internal network-network interface (reference point) [ITU-T G.8080]: The I-NNI is the reference point within a domain between routing areas and, where required, between sets of control components within routing areas.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 interworking: The term "interworking" is used to express interactions between networks, between end systems, or between parts thereof, with the aim of providing a functional entity capable of supporting an end-to-end communication.

3.2.2 interworking function (IWF): These functions are referred to in the interworking definition, which include the conversion between protocols and the mapping of one protocol to another. The functionality required between networks can be separated from the functionality, if any, required in end systems.

3.2.3 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).

3.2.4 network interworking: Interworking between two similar (like) networks via an intermediary network with dissimilar characteristics.

3.2.5 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.

3.2.6 peer-partition interworking: In peer-partition interworking, a client network's characteristic information (CI) is transferred transparently over a concatenation of peered server layer networks.

3.2.7 plane: A category that identifies a collection of related objects, e.g., objects that execute similar or complementary functions; or peer objects that interact to use or to provide services in a class that reflects authority, capability, or time period.

3.2.8 service interworking: Interworking between two dissimilar (unlike) networks directly without the benefit of an intermediary network.

3.2.9 service level agreement (SLA): A negotiated agreement between an end user and the service provider. Its significance varies depending on the service offerings. The SLA may include a number of attributes such as, but not limited to, traffic contract, availability, performance, encryption, authentication, pricing and billing mechanism, etc.

3.2.10 service plane: The service plane comprises:

a) service presentation functionality being presented to the end user;

b) service implementation aspects with which the end user interacts, e.g., service invocation, control service level agreement function.

NOTE – Items a) and b) use the totality of the transfer capabilities including control and management functionalities.

3.2.11 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.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AP	Access Point
AS	Autonomous System
ASON	Automatically Switched Optical Network
ATM	Asynchronous Transfer Mode
BDI	Backward Defect Indication
CAL	Common Adaptation Layer
CC	Continuity Check

CI	Characteristic Information
cl-ps	connectionless packet-switched
co	connection-oriented
co-cs	connection-oriented circuit-switched
co-ps	connection-oriented packet-switched
CoS	Class of Service
CV	Connectivity Verification
E-NNI	External Network-Network Interface (as per [ITU-T G.8080])
FDI	Forward Defect Indication
I-NNI	Internal Network-Network Interface (as per [ITU-T G.8080])
IWF	Interworking Function
IWP	Interworking Path
MPLS	MultiProtocol Label Switching
NGN	Next Generation Network
OAM	Operations, Administration and Maintenance
PCI	Protocol Control Information
PNNI	Private Network Node Interface
QoS	Quality of Service
SAP	Service Access Point
SLA	Service Level Agreement
SLS	Service Level Specification
TCP	Termination Connection Point

5 Conventions

This Recommendation uses no specific conventions.

6 General aspects of interworking

The objective of interworking is to provide some end-to-end communications entity across a single server layer network or across a concatenation of (peered) server layer networks, which may consist of different networking technologies.

Throughout this Recommendation, an individual server layer network, within such a concatenation of server layer networks, is referred to as a server layer partition. The end-to-end communications entity may be a common higher layer network but may be a common application adaptation function.

Fundamentally, there are two interworking relationships, "client/server" and "peer-partition". They are sometimes known as "network" and "service" interworking. More complex interworking relationships can be derived from these two fundamental ones.

Client/server interworking describes and defines the way that end-to-end user traffic is transferred over a single server layer technology.

Peer-partition interworking describes and defines the way that end-to-end user traffic is transferred over a concatenation of discrete server layer technology partitions that may belong to different networking modes.

Interworking can take place between any two contiguous layer networks, regardless of their position in the total layer network stack. For example, there is interworking between transmission-media dependent technologies at the lowest level of the total stack and between the application signalling at the highest.

Combinations of layer networks in these two interworking relationships can produce complex networks.

Therefore, it is important to address interworking from the perspective of the end-to-end user traffic (the client layer) and not just from the perspective of the server layer(s). It is clear that in order to achieve this aim, a client/server relationship must be formed between each of the server layer(s) and their common client layer (i.e., the end-to-end user traffic).

The performance experienced by a client entity is dependant on the server layer partitions that it traverses and the performance experienced will ultimately depend on the networking technology and the networking mode that each server layer partition belongs to. More generally because a link connection in layer network N is supported by a trail in layer network N – 1, and this is a recursive relationship to the physical transmission media, the performance impairments experienced by layer network N are a sum (which is not necessarily linear) of the performance impairments inherited from all of the lower layer server networks (down to the physical transmission media) that support layer network N.

A given layer network can be defined in terms of the specific semantics and syntax of its user plane traffic units (including user plane OAM functions) and its control plane functions. Provided that the same functional semantics exists in two layer networks, both the user plane and the control plane can be interworked by passing on the semantics from one layer network to the other in the appropriate syntax.

The user plane semantics describes the purpose (or meaning) of a given user plane function in an abstract manner. The user plane syntax for a function describes how that function is encoded within the user plane traffic (i.e., how the bits used to encode that function are arranged). For example, both MPLS and Ethernet support connectivity verification (CV) functions. The semantics of a CV function could be described as "a function that monitors the availability of a path across a network". Although both MPLS and Ethernet support CV functions, the syntax that they use to encode the CV function differs: the format of an MPLS CV packet as described in [b-ITU-T Y.1711] differs from the format of an Ethernet ETH-CC frame as described in [b-ITU-T Y.1731]. In other words, MPLS and Ethernet share the same semantics for CV but they do not share the same syntax.

However, different layer networks may have differing functional semantics and syntax (the exact semantics and syntax will depend on the layer network's networking mode and technology), and therefore interworking is restricted to the functional semantics shared by all the peered server layer technologies used to support a given client layer entity. Similarly, all the functions and features of one server layer technology may not be available in another server layer technology, even if they belong to the same networking mode. Depending on the server layer partitions' network technology and mode, the set of common functions and features between different server layer partitions could be large or small.

It is also possible that a given application or client layer to server layer adaptation used by one server layer technology will not be the same as the one used by a different server layer technology. Therefore, it is possible that an attempt to map functions and features directly between server layer partitions will not be totally successful, and in some cases will have no meaning. When transferring a common client entity over a series of server layer networks, it is especially important for each server layer network to, wherever possible, transparently transfer all of the planes of its client entity.

The service plane concept allows the use of layer network transfer capabilities along with control and management related capabilities in the construction of services, as may be specified within an overall service level agreement (SLA) between the service provider and the end user. The service plane utilizes the capabilities provided by the underlying transfer functions, as well as the control and management plane functions. Consequently, the service plane incorporates more than just a layer service as defined in a service access point (SAP) in the protocol stack.

The interworking function (IWF) deals with the processing of the protocol layer functions in order to support the service across different networks. For some services, there is also the case in which the IWF deals with interworking between the application layer functions.

A layer network is defined by "the complete set of access groups of the same type which may be associated for the purpose of transferring information" [ITU-T G.805]. Within a physical network, there can be many layer networks. Since a layer network is defined by its access points, multiple layer networks can exist in the same physical network.

A layer network operates in one of three *modes:* connection-oriented circuit-switched (co-cs); connection-oriented packet-switched (co-ps); or connectionless packet-switched (cl-ps).

Within a layer network, there are one or more *planes.* The traditional three planes are the user plane (also known as the forwarding plane or data plane), the control plane, and the management plane. For further discussion of planes, see [ITU-T G.8080]. Layer networks are interworked in either a client/server or a peer-partition relationship (see clauses 7 and 9, respectively). Interworking is not only between different layers, but also between different planes within those layers.

7 Client/server relationships (network interworking)

7.1 General

In a client/server relationship (also known as network interworking), the client network characteristic information (CI) is transferred transparently over an intermediary server network. This implies that there is no functional coupling of the client and server layer networks. This is also true of the client layer control and management plane communications if they are coincident with the client user plane. Client/server relationships are used to create the topology of networks where an entity will lease capacity provided by server layer trails from one or more service providers. As there could be several different service providers and/or entities involved, there is no functional coupling in either a vertical (client/server) sense or a horizontal (peer-partition) sense between any of the parties' layer networks.

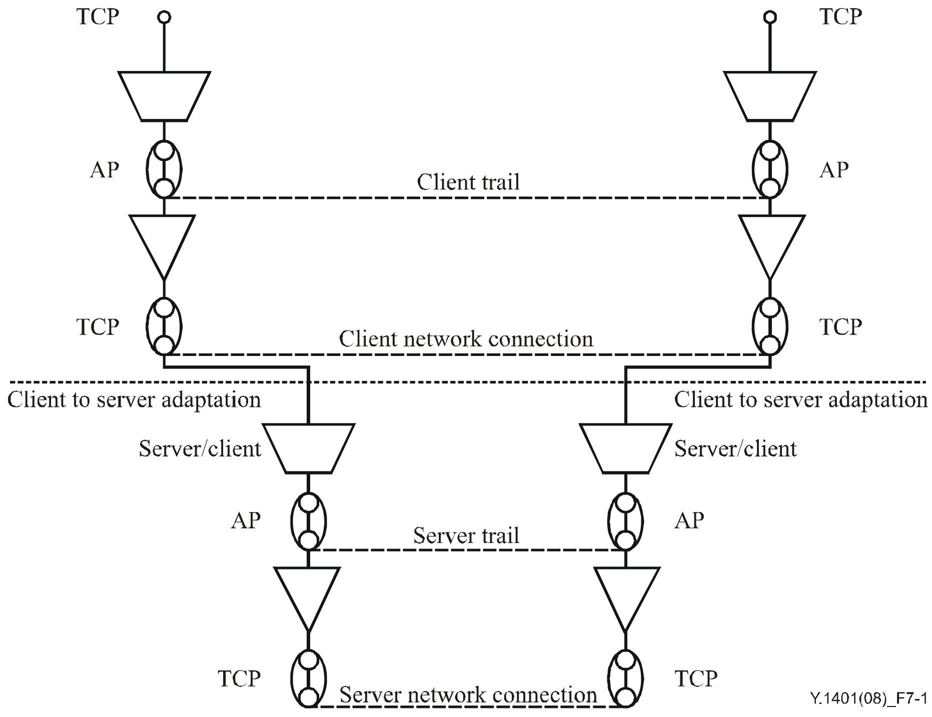

Figure 7-1 – Client/server user plane interworking between two connection-oriented layer networks

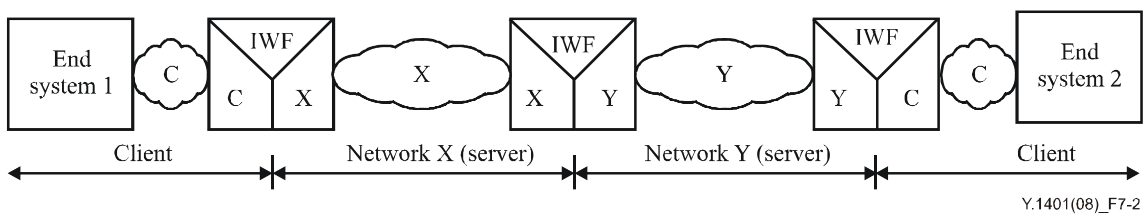

Figure 7-2 – Client/server interworking between two networks

A functional diagram showing an example of client/server user plane interworking between two connection-oriented layer networks is shown in Figure 7-1 using the conventions of [ITU-T G.805]. The client layer network and/or the server layer network may belong to any of the three networking modes. Examples of client/server user plane interworking between connection-oriented and connectionless layer networks are depicted in Figures 5, 6 and 7 of [ITU-T G.809].

Figure 7-1 can also be used to show that the client layer control and management plane communications are transparently transferred over their respective server layer networks in a client/server interworking relationship because 'client network connection' and the 'client trail' in Figure 7-1 could actually belong to any one of the user, control or management plane communications of the client layer network.

In a client/server relationship, for a given performance metric, the performance of the client layer is equal to the performance of the server layer plus any impairment introduced by the client layer itself. Therefore, in general, it is not possible for a client layer trail to provide better performance than the server layer trail over which it is transported. Due to this client/server performance inheritance, it is generally advisable to stack modes that normally exhibit poorer performance on top of modes that normally exhibit better performance. Table 5-3 of [ITU-T Y.1314] describes the client/server mode combinations that are possible and provides some information on their compatibility.

7.2 Principles of client/server relationships

The principles for user plane client/server layer network interworking are:

- The client and server layer networks are functionally independent.
- Whilst the link connection adaptation required will be different for each pair of client/server technologies (this depends on the networking mode of the client and server layer networks), the client layer network (all planes) must be transferred transparently over the user plane of the server layer trail.
- Any defects in a server layer network must be detected at the server layer trail termination points and may be mapped (in the server to client layer adaptation function) to the forward defect indication (FDI) syntax of the client layer network, unless the client layer network belongs to the cl-ps mode in which case FDI mapping makes little sense.

The principles of client/server interworking are, in general, also true when the client is a common application adaptation function (instead of a layer network).

7.3 OAM principles of client/server relationships

As there is no functional coupling of the client and server layer networks in a client/server relationship, each layer's OAM is independent and functions in isolation to the other layer.

However, any defects in a server layer network must be detected at the server layer trail termination points and mapped (in the server to client layer adaptation function) to the forward defect indication (FDI) syntax of the client layer network. Note that if the client layer network belongs to the cl-ps mode then FDI mapping makes little sense.

8 Common adaptation layer (CAL)

8.1 General

Traditionally, a specific encapsulation has been defined for each pair of client and server technologies. Therefore, if N different client layer technologies need to be interworked with M server layer technologies, then this leads to a requirement for N × M encapsulations. The number of specific encapsulations required can be reduced by introducing the concept of a common adaptation layer (CAL) that is independent of the server layer networks and therefore sits between the client layer and its server layer(s). A client layer is first encapsulated into the CAL and then the CAL is encapsulated into the server layer, thereby reducing the number of encapsulations that each server layer must support.

The use of a CAL also enables the modelling of an end-to-end entity across the server layer network(s) which in turn allows end-to-end entity defect monitoring, independent of the client and server layers, while transparently transferring the actual end-to-end user traffic. A CAL should therefore be able to operate over any server layer.

A functional diagram showing an example of the use of a CAL in client/server user plane interworking between two connection-oriented layer networks is shown in Figure 8-1, using the conventions of [ITU-T G.805]. The client layer network and/or the server layer network may belong to any of the three networking modes, subject to the comments related to performance mentioned in clause 7.

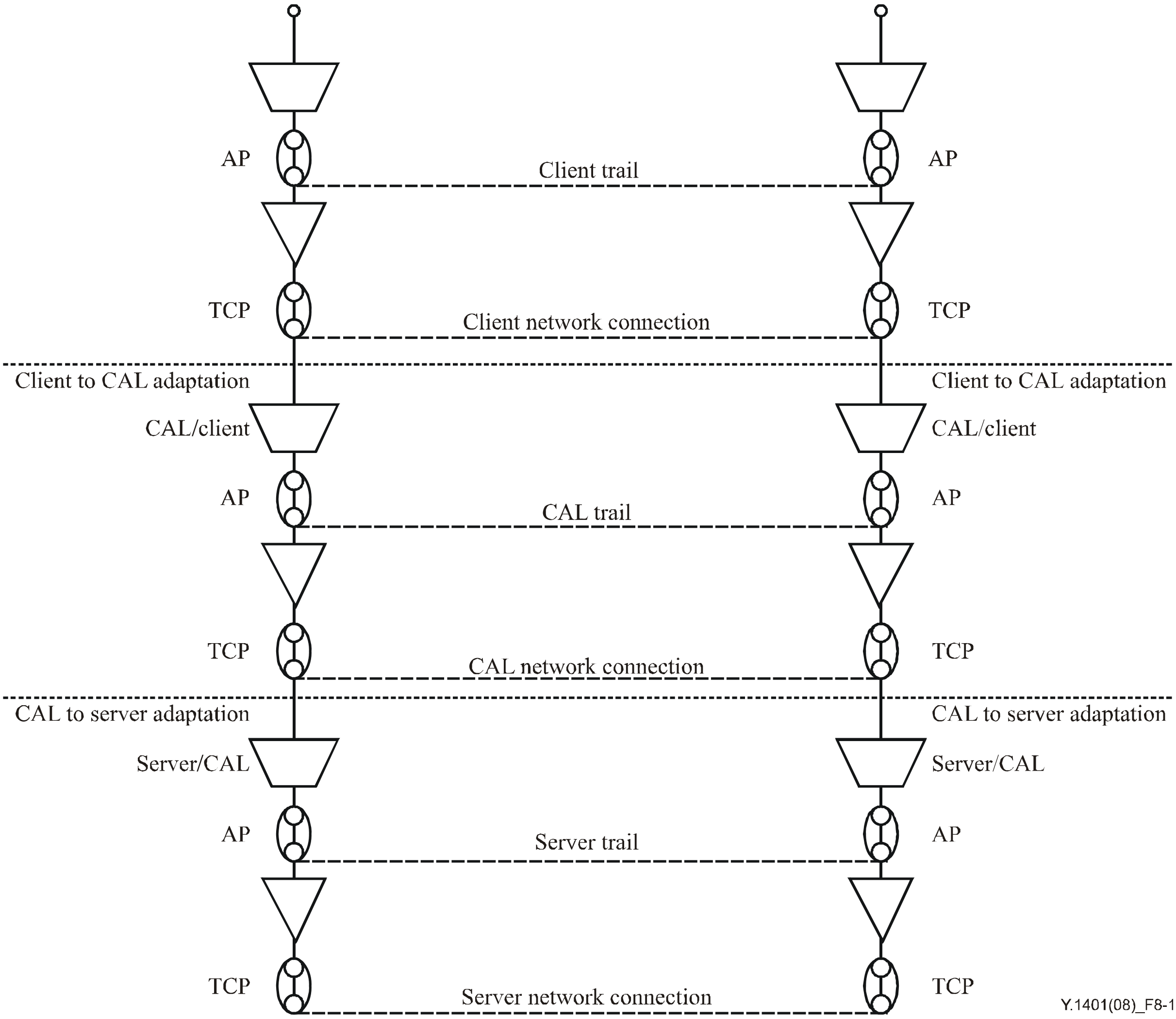

Figure 8-1 – Use of a CAL in client/server user plane interworking between two connection-oriented layer networks

When the client of a CAL is an application (as opposed to a layer network), the CAL acts as a common application adaptation over which the application's information is transferred across the server layer(s).

A CAL must be able to support all three information transfer categories (i.e., messages, open sequences and timed sequences) between its client's applications. The mechanisms used by a CAL to support these three information transfer categories are beyond the scope of this Recommendation.

A CAL is an abstract concept and may be implemented as a single server layer independent adaptation or as a number of server layer independent adaptations, for example by using a different adaptation for each information transfer category.

Although some functions provided by a CAL will retain the same syntax across all server layers and for all client layers, some functions provided by a CAL may vary depending on the client and server layers that are being adapted. For example the CAL trail termination functions are likely to retain the same syntax regardless of the client and server layers used, but the syntax used to adapt a client into the CAL and the CAL into a server layer partition may vary depending on the specific client or server layers used.

8.2 CAL principles

The principles for a CAL are:

- The CAL is expected to mediate between the various capabilities of the server layer(s) that it transits.
- The CAL should support its own OAM mechanisms in order to detect end-to-end faults that may not be detected by a single server layer partition. The CAL's OAM mechanism is expected to conform to generally accepted principles and requirements for OAM.
- The CAL is expected to support the transfer of fault indications (FDI) that have been mapped from a server layer partition.
- It is expected that the functions of the CAL are independent of the server layer(s) that it transits. This includes the server layers' user, control and management plane components (if relevant control and/or management plane components exist).
- CAL forward defect indication (FDI) should maintain the same semantics (but not necessarily the same syntax) as the equivalent fault indication in the affected server layer. In some specific cases, it may be necessary to map the semantics.
- At the CAL termination point, it is expected that defects detected by (or indicated to) the CAL will be mapped to appropriate defect indications in the affected client layers (if an appropriate defect indication exists in the affected client layers). It is expected that an appropriate reverse defect indication will be inserted by the same CAL termination point in the reverse direction.

8.3 CAL OAM principles

A CAL should provide its own OAM functionality in addition to that provided by its server layer(s). Specifically a CAL should support the following OAM functionality: CC, CV, FDI and backward defect indication (BDI), as appropriate. If a CAL does not provide its own OAM functionality, then the OAM messages and defect states can be mapped between server layer partitions. However, such mapping is extremely complex and should be avoided unless absolutely necessary. The OAM functionality provided by a CAL should always be the preferred option and OAM mapping between server layer partitions should only be used where no alternative exists.

Figure 8-2 shows a single CAL trail being transported across three peered server layer partitions; although the principles of CAL OAM remain the same if the CAL is only being transported over a single server layer network. Each server layer partition consists of a single subnetwork which is supporting a single trail along with that trail's respective adaptation and termination sources and sinks. In practice a server layer partition may consist of multiple subnetworks in which case the termination sink closest to the defect is responsible for detecting any defects in that server layer partition and taking the appropriate action as detailed below. Within Figure 8-2, the blue arrow shows the flow of OAM messages within the CAL and the green arrow shows the flow of OAM messages to the layers above the CAL.

Figure 8-2 and the text of the previous paragraph assume that the CAL and each server layer partition are connection-oriented. The principles and behaviour described remain the same if they are connectionless; however, the word 'trail' in the figures and text must be replaced with 'flow'.

The behaviour expected when the CAL detects a defect is shown in Figure 8-2 and described below. Any interactions with the management plane are outside the scope of this Recommendation.

1) In response to detecting a defect (in its own layer) or receiving an FDI (from a lower layer), the CAL termination sink should generate and propagate a BDI towards the CAL termination source (if the CAL supports BDI and BDI is considered appropriate for that CAL trail).

2) The CAL termination sink should map the defect it has detected into a defect (with the same semantics as an FDI) in all affected client layers (i.e., all the client layers that are supported by the affected CAL trail).

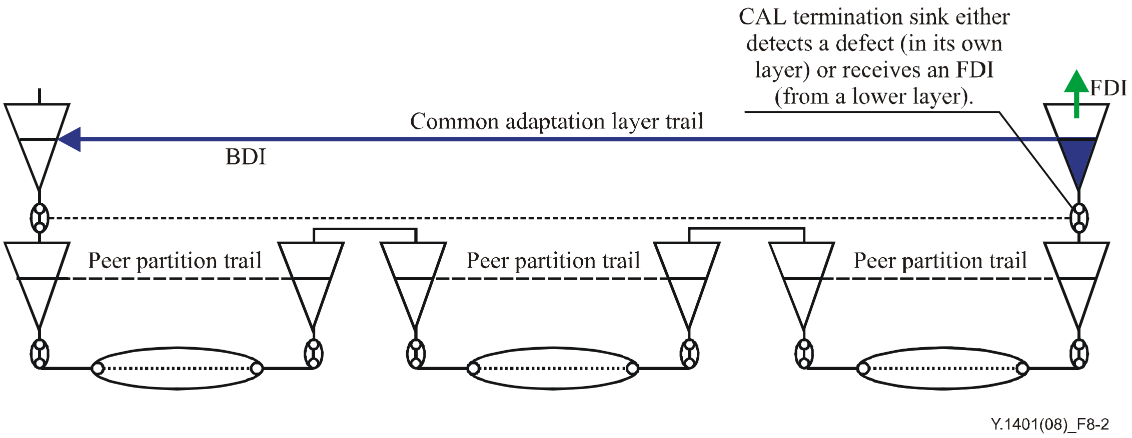

Figure 8-2 – Expected OAM behaviour when the common adaptation layer detects a defect

The only place where an alarm should be generated is at the trail termination of the CAL. Alarms should not be generated by the trail termination of the CAL in response to receiving an FDI. The client FDI should suppress any alarms in the client layer(s).

9 Peer-partition interworking (service interworking)

9.1 General

In peer-partition interworking (also known as service interworking), the client network characteristic information (CI) is transferred transparently over a concatenation of peered server layer networks. This implies that there is no functional coupling of the client with the various server layer networks. This is also true of the client layer control and management plane communications if they are coincident with the client user plane. However, there is likely to be at least some functional coupling between the various server layer networks.

A functional diagram showing an example of peer-partition user plane interworking between two peered server layer partitions is shown in Figure 9-1 where a client is transparently transferred over two server layer networks (X and Y) via a CAL. Figure 9-1 depicts both layer networks X and Y as connection-oriented technologies; however, they could each belong to any of the three networking modes of connection-oriented circuit switched (co-cs), connection-oriented packet switched (co-ps) and connectionless packet switched (cl-ps).

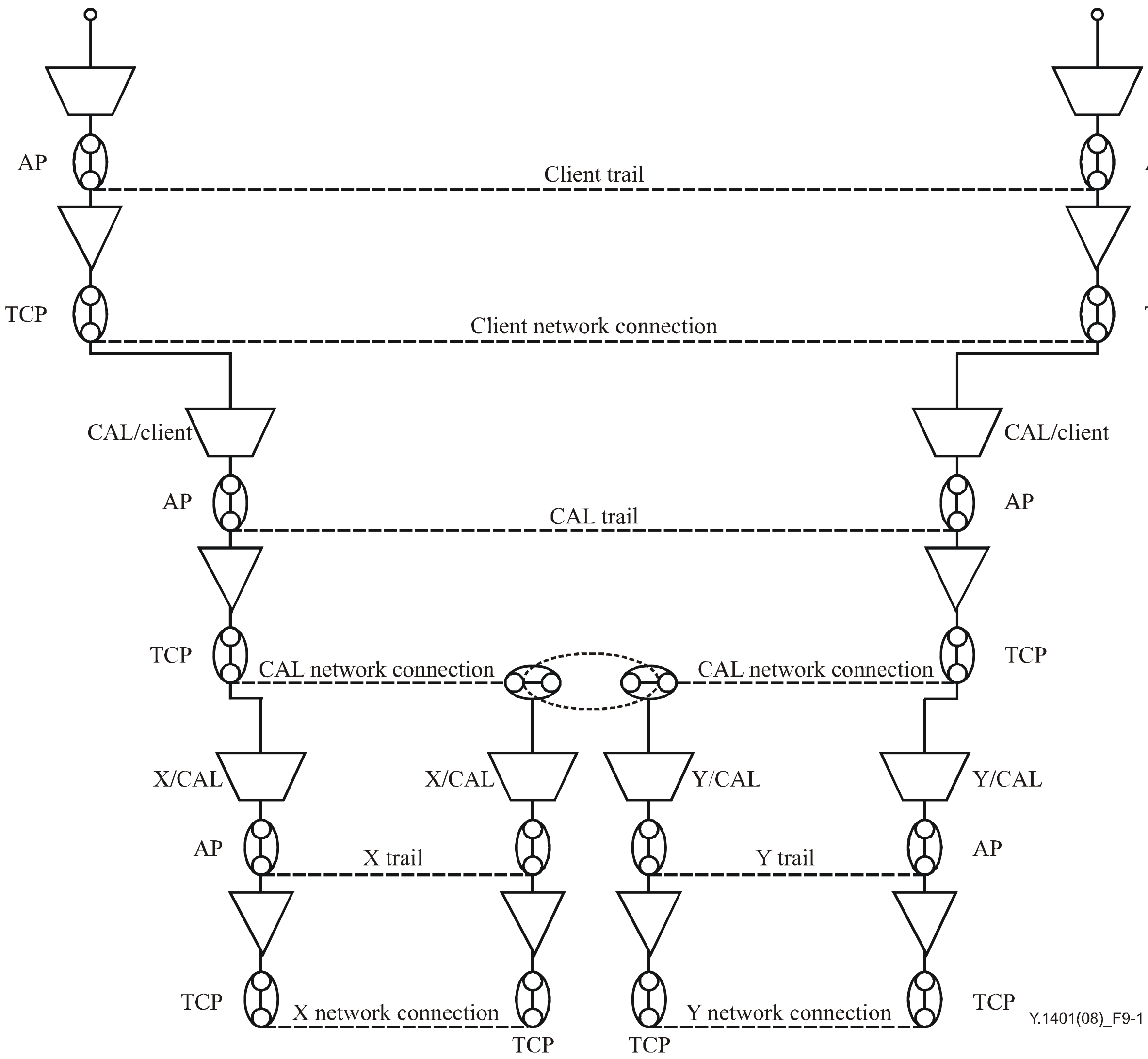

Figure 9-1 – Peer-partition user plane interworking between two peered server layer partitions via a CAL

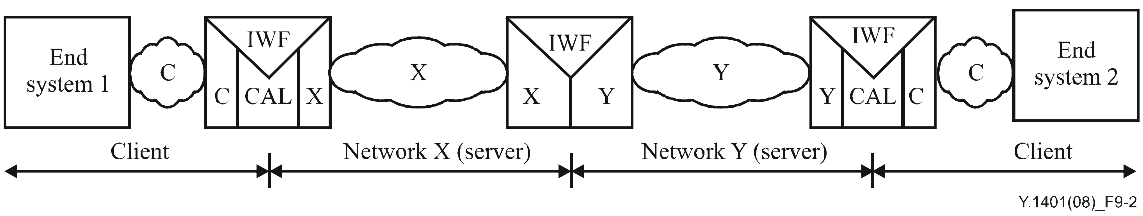

Figure 9-2 – Peer-partition user plane interworking between two server layer partitions

Although Figure 9-1 shows a CAL being used to facilitate interworking, peer-partition interworking does not mandate the use of a CAL and peer-partition interworking can be achieved without a CAL provided that functionality which is equivalent to that provided by a CAL (for example adaptation, OAM) is implemented in each server layer partition.

If a CAL is not used, then an IWF must exist between each pair of server layer partitions. This IWF must translate (i.e., map) the protocol control information (PCI) of the first server layer partition into the PCI of the second server layer partition for user, control and management plane functions to the extent possible. In general, since not all functions may be supported in one or other of the networks, the translation of PCI may be partial or non-existent. However, this should not result in any loss of user data since the payload is not affected by PCI conversion at the IWF.

A functional diagram showing an example of peer-partition user plane interworking between two peered server layer partitions is shown in Figure 9-3 where a client is transparently transferred over two server layer networks (X and Y) without the use of a CAL.

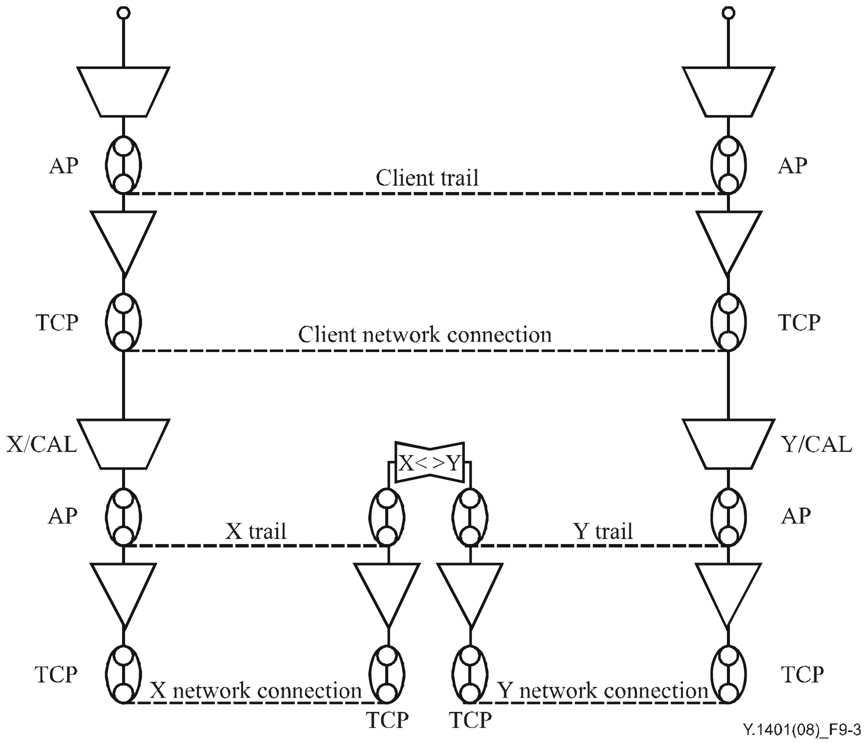

Figure 9-3 – Peer-partition user plane interworking between two peered server layer partitions without the use of a CAL

As a consequence of client/server performance inheritance, the common adaptation layer inherits the performance of all the server layer partitions. In practice, it may be that one partition is the dominant source of impairments due to the choice of mode and/or poor network design in that server layer network. Therefore, it is not possible for the common adaptation layer to provide (or receive) better performance than the worst performing (for a given performance metric) of the peered server layer partitions over which it is transported.

Therefore, it is necessary to carefully consider which server layer modes (and/or technologies) it is appropriate to peer with one another in order to provide the topmost client layer (the 'application' or 'service') with the performance that it requires. Due to the different performance provided by, and the different functions and features of each networking mode (i.e., cl-ps, co-ps or co-cs), it is generally advisable to only peer server layer partitions that belong to the same network mode with each other. The common mode case increases the likelihood of functional harmonization between peered server layer partitions and consequently will often result in interworking scenarios which are simpler and lower cost for service providers to manage. In the common mode case, when there is significant functional harmonization across both the traffic user plane and the control plane of peered server layer partitions, and in particular where these are identical in each partition, we have the E-NNI/I-NNI scenario of public network service interworking.

Although peer-partition interworking does not preclude the peering of any combination of server layer networking technologies and modes, it is clearly down to the specific service provider deploying peer-partition interworking to decide which combinations are sensible and which are not.

This decision is likely to be based on several criteria including (but not limited to) the requirements of the service being deployed, the service level specification (SLS) and service level agreement (SLA) agreed with the customer of the service and the common set of functions and features across the peered server layer partitions.

Figures 9-1 and 9-3 above only show user plane interworking between two peered server layer partitions. However, it will often be necessary to implement control plane interworking (and in some cases management plane interworking) between peered server layer partitions. The functional diagram depicting control plane interworking between peered server layer partitions looks similar to Figure 9-1 (if a CAL is used) or Figure 9-3 (if a CAL is not used) except that control plane messages (instead of user plane adapted information) must be mapped from the syntax used by one server layer partition into the syntax used by the next server layer partition.

Different layer networks may have differing control plane semantics (i.e., functions, features and message sets) which will likely depend on the layer network's networking mode and technology. Therefore, the degree of control plane interworking that can be achieved is restricted to the semantics shared by all the peered server layer technologies used to support a given client layer entity. Depending on the server layer partitions' network technology and mode, the set of common control plane functions and features between different server layer partitions could be large or small.

9.2 Principles of peer-partition interworking

Use of a CAL is the preferred solution for providing an end-to-end communications entity across the peered server layer partitions because it allows the possibility of end-to-end defect monitoring while transparently transferring the actual end-to-end user traffic in a simple manner without introducing excessive complexity. However peer-partition interworking does not mandate the use of a CAL and peer-partition interworking can be achieved without a CAL provided that functionality which is equivalent to that provided by a CAL (for example adaptation, OAM) is implemented in each server layer partition.

The following generic principles apply to peer-partition interworking (regardless of the networking modes of the peered server layer partitions being interworked).

- To provide an end-to-end trail to allow monitoring of user traffic, a CAL should be used. If a CAL is not used, it is expected that functionality which is equivalent to that provided by a CAL be implemented.
- Relevant defects detected in a server layer partition should be mapped to an appropriate FDI defect in the CAL or client layer (if a CAL is not used).
- It is expected that there is no direct feature/function mapping between peered server layer partitions. However, if a CAL is not used then some feature/function mapping between peered server layer partitions may be necessary.
- It is expected that the characteristic information (CI) of each peered server layer partition is terminated at the trail (or flow) termination sink of that layer network.

9.3 OAM principles of peer-partition interworking

A connection oriented (co) mode server layer partition should support continuity checks (CCs) and backward defect indication (BDI) for each direction of a trail. These are parts of what is referred to as operations, administration and maintenance (OAM) functions. These functions should be independent of each other and processed in a unidirectional sense in order to work correctly and in order for OAM to scale across both point-to-point and point-to-multipoint connections. A co mode server layer partition may also support a forward defect indication (FDI) as a means to try and suppress client layer alarms when the failure is in a lower layer network. However, an FDI function is not essential.

A cl-ps server layer partition may support a connectivity verification (CV) function, although it is more usual for it to support a 'ping-like' echo request/response behaviour.

If the CAL does not provide its own OAM functionality (or if a CAL is not used), then the OAM messages and defect states can be mapped between server layer partitions. However, such mapping is extremely complex and should be avoided unless absolutely necessary.

What should happen when a defect is detected in a server layer partition is dependent on the OAM capabilities of the CAL, as described below.

Figures 9-4 and 9-5 show a single CAL trail being transported across three peered server layer partitions. Each server layer partition consists of a single subnetwork which is supporting a single trail along with that trail's respective adaptation and termination sources and sinks. In practice, a server layer partition may consist of multiple subnetworks in which case the termination sink closest to the defect is responsible for detecting any defects in that server layer partition and taking the appropriate action as detailed below. Within each figure, the red arrows show the flow of OAM messages within the server layer partition that first detects a defect, the blue arrows show the flow of OAM messages within the CAL (Figure 9-4) or within the other server layer partitions (Figure 9-5) and the green arrows show the flow of OAM messages to the layers above the CAL.

Figures 9-4 and 9-5, the text of the previous paragraph assume that the CAL and each server layer partition are connection-oriented. The principles and behaviour described remain the same if they are connectionless; however, the word 'trail' in the figures and text must be replaced with 'flow'.

9.3.1 OAM principles when the CAL supports OAM functions

When a defect is detected in a server layer partition (which is supporting a CAL trail), any fault indications received or detected by that server layer's termination sink should be mapped into the equivalent of an FDI in the CAL to aid with alarm suppression. The equivalent of an FDI in the CAL shall maintain the same semantics but not necessarily the same syntax as the server layer partition FDI.

Only server layer partition FDI should be mapped into equivalent FDI in the CAL in order to aid with alarm suppression. All other server layer partition OAM should be terminated at that server layer partition's termination sink. Therefore the only semantics that need to be shared between a server layer partition and the CAL are those relating to FDI, examples include defect type, defect location and trail termination source (or connectionless trail termination source).

For example, an FDI generated by a server layer partition should be mapped into an FDI in the CAL. An FDI in the CAL does not have to share the same syntax (i.e., they do not have to share a common bit format/structure) as an FDI in a server layer partition. However, an FDI in the CAL shall share the same semantics as an FDI in a server layer partition (i.e., they are both FDI and therefore share the semantics "defect indication to aid alarm suppression").

The behaviour expected when a server layer detects a defect (or when the server layer receives an FDI) and the CAL supports OAM functions is shown in Figure 9-4 and described below. Any interactions with the management plane are outside the scope of this Recommendation.

1) In response to detecting a defect (in its own layer) or receiving an FDI (from a lower layer), the server layer partition termination sink should generate and propagate a BDI towards the server layer partition termination source (if the server layer partition supports BDI and BDI is considered appropriate for that server layer partition trail).

2) The server layer partition termination sink should map the defect detected (in its own layer) or the FDI received (from a lower layer) into an FDI in the CAL which should be propagated towards the CAL termination sink.

3) If the CAL supports BDI and BDI is considered appropriate for that CAL trail, the CAL termination sink should generate and propagate a BDI towards the CAL termination source in response to receiving an FDI.

4) The CAL termination sink should map the FDI it has received into a defect (with the same semantics as an FDI) in all affected client layers (i.e., all the client layers that are supported by the affected CAL trail).

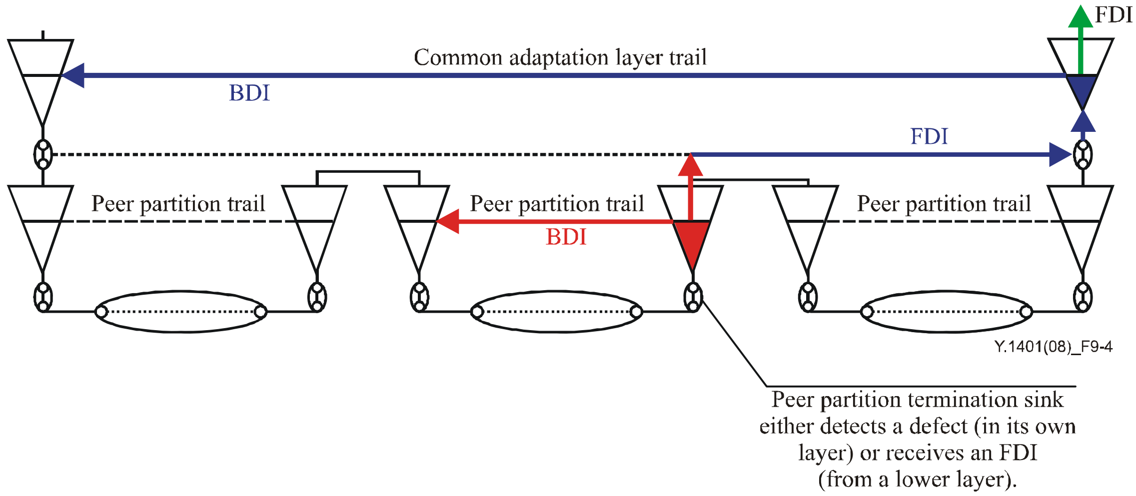

Figure 9-4 – Expected OAM behaviour when a server layer detects a defect

The only place where an alarm should be generated is at the trail termination of the server layer partition where the defect was originally detected. Alarms should not be generated by the trail termination of the server layer partition in response to receiving an FDI. The CAL FDI should suppress any alarms in the CAL and the client FDI should suppress any alarms in the client layer(s).

NOTE – When the CAL supports OAM functions, there should not be any OAM mappings between peered server layer partitions. Therefore if there are CV OAM flows active in the other peered partitions, then these continue unaffected.

9.3.2 OAM principles when the CAL does not support OAM functions

The behaviour expected when a server layer detects a defect and the CAL does not support OAM functions is shown in Figure 9-5 and described below. Any interactions with the management plane are outside the scope of this Recommendation.

1) In response to detecting a defect (in its own layer) or receiving an FDI (from a lower layer), the server layer partition termination sink should generate and propagate a BDI towards the server layer partition termination source (if the server layer partition supports BDI and BDI is considered appropriate for that server layer partition trail).

2) The server layer termination sink should map the defect detected (in its own layer) or the FDI received (from a lower layer) into an FDI in the downstream server layer partition. The downstream server layer partition should suppress its CV flow for the relevant peer-partition trail and propagate the FDI towards the appropriate termination sink in the downstream server layer partition.

3) In response to receiving a BDI, the server layer partition termination source should map the BDI it has received into a BDI in the upstream server layer partition. The upstream server layer partition should not suppress its CV flow for the reverse direction (i.e., the direction the BDI flows in) unless the defect is bidirectional and should propagate the BDI towards the appropriate termination source in the upstream server layer partition.

4) The above FDI and BDI mappings should continue until they reach the termination sink and termination source (respectively) in the final server layer partitions in each direction.

5) The final server layer partition termination sink should map the FDI it has received into a defect (with the same semantics as an FDI) in all affected client layers (i.e., all the client layers that are supported by the affected peer-partition trail).

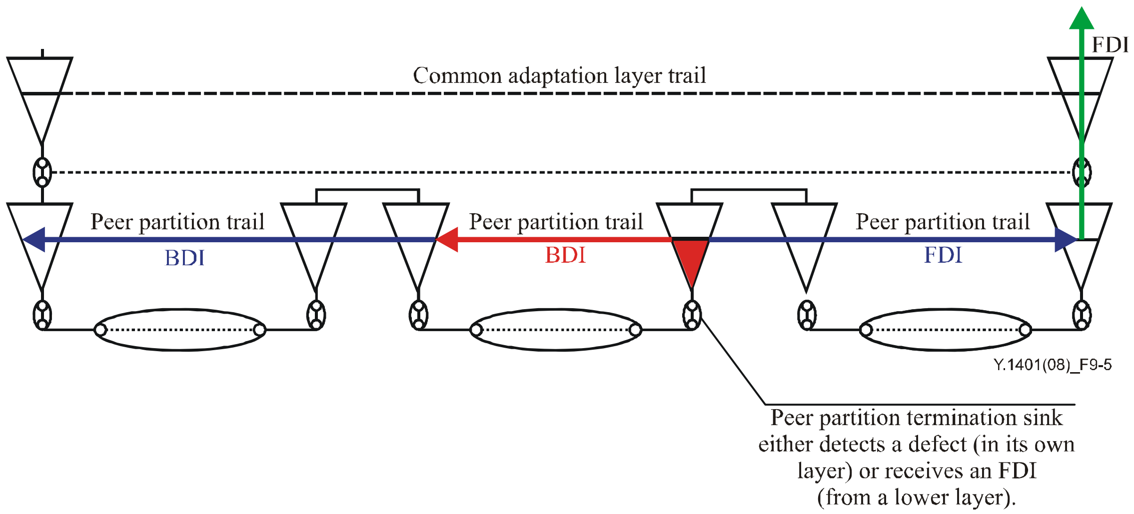

Figure 9-5 – Expected OAM behaviour when the CAL does not support OAM functions

The only place an alarm should be generated is at the trail termination of the server layer partition where the defect was originally detected. Alarms should not be generated by the trail termination of a server layer partition in response to receiving an FDI. The incoming server layer partition FDI should suppress any alarms in the other peer-partitions and in the client layer(s).

It may be considered useful to include a defect location field in the FDI and BDI functions, which could for example contain the autonomous system (AS) number of the layer network where the defect was originally detected. Such a field could be used by operations personnel in the other server layer partitions to identify the source of the defect.

10 Control plane interworking

10.1 General

This clause describes the interworking of the control planes that cooperate to establish the user plane connection or flow.

For connection-oriented networks (co-ps and co-cs), user plane interworking can only take place after a connection has been established. The control plane is mainly concerned with the topology of the user plane and has two major functions. These are routing or path computation and signalling to establish a connection.

For connectionless networks (cl-ps), user plane interworking can only take place if an end-to-end route (or path) exists. The control plane is mainly concerned with the topology of the user plane and its only major function is that of routing.

10.2 Network scenarios

For the purposes of this clause, the following scenario is assumed where both networks A and B operate in a connection-oriented (co) mode.

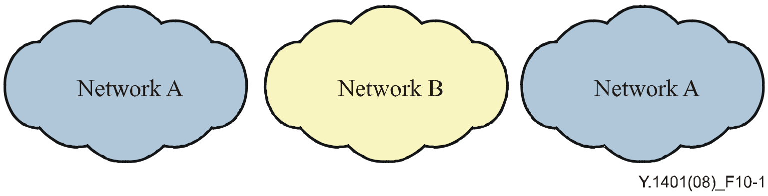

Figure 10-1 – Example network scenario

The control planes in networks A and B operate "as normal" in the context of networks A and B. The communication between them is supported by an interworking function. However, unlike user plane interworking, this interworking function is not always confined to the boundary between the networks in all cases.

For the purposes of the control plane, the user plane is abstracted as a set of subnetworks (that allow flexible connections), links (that represent the fixed connectivity between the subnetworks) and access points (where the user services attach to the network). These abstractions of the user plane entities are represented by components in the control plane.

To allow the functions of path computation and signalling to be performed, it is necessary to represent the complete network using the control plane components and name space of one of the technologies. For example, to allow control plane A to compute a path and signal it to establish a connection, the topology of network B must be abstracted and represented in the context of network A. This includes the use of the appropriate network A control plane entities and functions that represent the appropriate elements of the network B topology. This clause describes the case where network B is abstracted in the context of the network A control plane. It is also possible to abstract network A in the context of the network B control plane.

The topology of network B may be abstracted in three different ways as described below. The selection of the type of abstraction is constrained by the capabilities of the networks and the policy of the network operator.

10.3 Link representation

The user plane topology of the technology B network is abstracted as a set of (pre-provisioned) links between the technology A subnetworks as shown in the example in Figure 10-2. This representation provides the simplest case for control plane interworking but offers the least network flexibility. Note that the links are the abstraction presented to the control plane of network A and do not necessarily represent the full topology of network B.

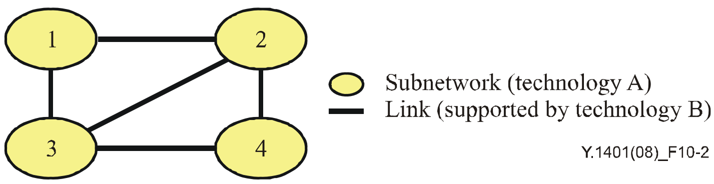

Figure 10-2 – User plane topology of technology B network abstracted as a set of links

The control plane representation of the edge of subnetwork 1, the edge of subnetwork 2 and the link between them is shown in Figure 10-3 below.

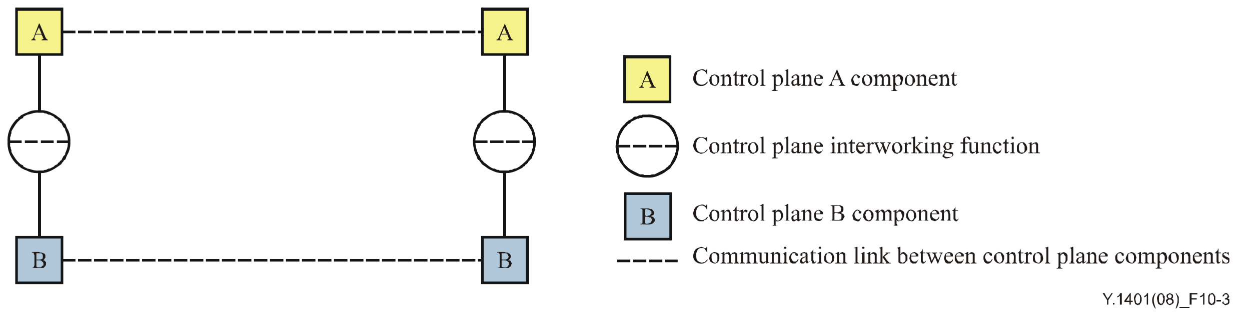

Figure 10-3 – Control plane representation between the edges of two networks

The control plane for technology A can compute paths across the (complete) network and must request the assignment of link connections for the technology B links. Technology B is not involved in path computation.

To perform these functions, the control adjacencies for control plane A (including those across the links provided by network B) must be pre-provisioned. The control plane interworking function translates the parameters of the network B links (e.g., cost, diversity, capacity) and name space into the semantics used by the control plane of technology A.

The control entities in subnetworks 1, 2, 3 and 4 must be able to exchange network topology information including the status of the technology B links.

When a connection is to be established, for example across the link between subnetworks 1 and 2, control plane A must request the assignment of a transport label and the allocation (or reservation) of the required bandwidth from the technology B link (note that if B is co-cs the assignment of the transport label (timeslot) implicitly reserves the capacity). To achieve this, the signalling message is intercepted by the control plane interworking function at the boundary of subnetwork 1 and the information relating to the network B link connection is extracted and used to allocate the link connection across network B. The part of the signalling message required to set up the connection across subnetwork 2 is encapsulated and transferred to subnetwork 2 along with the link label.

It is important that the network A signalling message is encapsulated and not translated into network B semantics, since unless the semantics of A and B are identical, information will be lost or corrupted when it is translated back into network A semantics. Call parameters need to be visible to the control plane interworking function as different connection segments exist on either side of the boundary. The policy for mapping the user service parameters (e.g., CoS, QoS) to the connection may be different for technology A and B.

In this case, the interworking functions are logically located at the boundary between the network A subnetwork and the network B link.

10.4 Subnetwork representation

If the user plane of network B can provide flexible connections at the granularity used by network A, then network B may be represented as a subnetwork interconnected via links as show in Figure 10-4. Note that the subnetwork and links are the abstraction presented to the control plane of network A, this does not necessarily represent the full topology of network B.

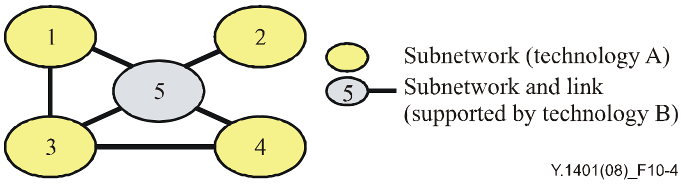

Figure 10-4 – User plane topology abstracted as a subnetwork interconnected via links

This representation provides more flexibility than the link representation described above. The limitation is that the view presented to any path computation function has limited detail and hence path computation results may not always be optimal. For example, should a path from subnetwork 1 to subnetwork 4 be routed via subnetwork 5 or 3? This is very similar to the case of hierarchical routing (e.g., as described by the ASON architecture in [ITU-T G.8080]) where the details of subnetwork 5 are not visible to the path computation function. It is similar to the ATM PNNI routing complex node representation.

Note that a single network element may offer flexibility in both technology A and technology B. In that case, the intersection between the technologies is within a single network element. However, it is simpler to represent this intersection as a link to which we can associate parameters such as cost and capacity. It also allows us to describe the translation between name spaces.

An example control plane representation for subnetworks 1, 2 and 5 and the links between them is shown in Figure 10-5.

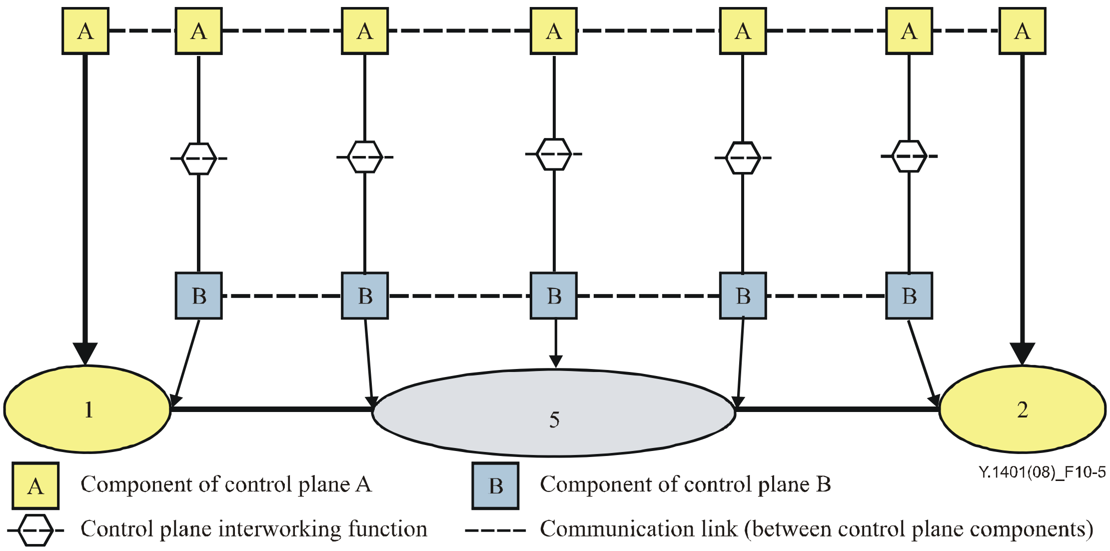

Figure 10-5 – An example control plane representation for subnetworks 1, 2 and 5 and the links between them

The control plane for technology A can compute paths across the (complete) network and must request the assignment of connections for the B technology links and subnetworks.

To perform these functions, the control adjacencies for control plane A (including those across the links and subnetworks provided by network B) must be pre-provisioned.

The control plane interworking function associated with the links supported by network B translates the parameters (e.g., cost, diversity, capacity) and name space into the semantics and name space used by the control plane of technology A. The control plane interworking function associated with the subnetwork supported by network B translates the name space of the subnetwork into the network A name space.

These control entities must be able to exchange network topology information including the status of the technology B links and the technology A reachable end points. Subnetwork 5 must present technology A control entities. This allows the technology A control plane to perform path computation and request connections.

The technology A control plane can compute a path (for example from subnetwork 1 to subnetwork 2) that transits subnetwork 5. To establish a connection, a signalling message must be sent from subnetwork 1 to subnetwork 2 via subnetwork 5. This signalling message is intercepted by the control plane interworking function at the boundary of subnetwork 1, the information relating to the technology B network is extracted and used to allocate the link connections and establish the subnetwork connection across subnetwork 5. The remainder of the signalling message (i.e., the part required to set up the connection across subnetwork 2) is encapsulated and transferred to subnetwork 2 along with the link labels.

Again, call parameters need to be visible at the boundary between the different networks as different connection segments exist on either side of the boundary. The policy for mapping the user service parameters (e.g., CoS, QoS) to the connection may be different for technology A and B.

In this case, one of the control plane interworking functions is (logically) located within network B.

10.5 Abstract network representation

If the user plane of network B can provide flexible connections at the granularity used by network A, then network B may be represented as a set of subnetworks interconnected via pre-provisioned links as show in Figure 10-6. The subnetwork and links are the abstraction presented to the control plane of network A. This does not necessarily represent the full topology of network B. The more detail that is provided about network B's topology, then the "better" the results of any path computation. Each of the technology B subnetworks may represent a single routing node or a complex node (as described in clause 10.4 above).

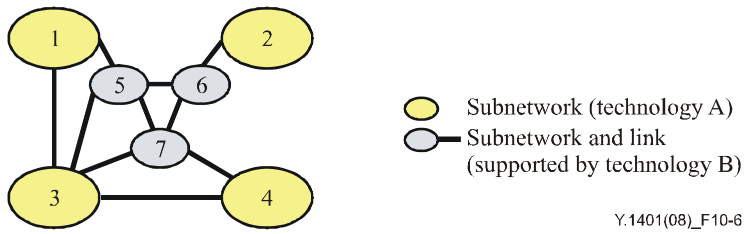

Figure 10-6 – User plane topology of network B represented as a set of subnetworks interconnected via links

An example control plane representation for subnetworks 1, 2, 5 and 6 and the links between them is shown in Figure 10-7.

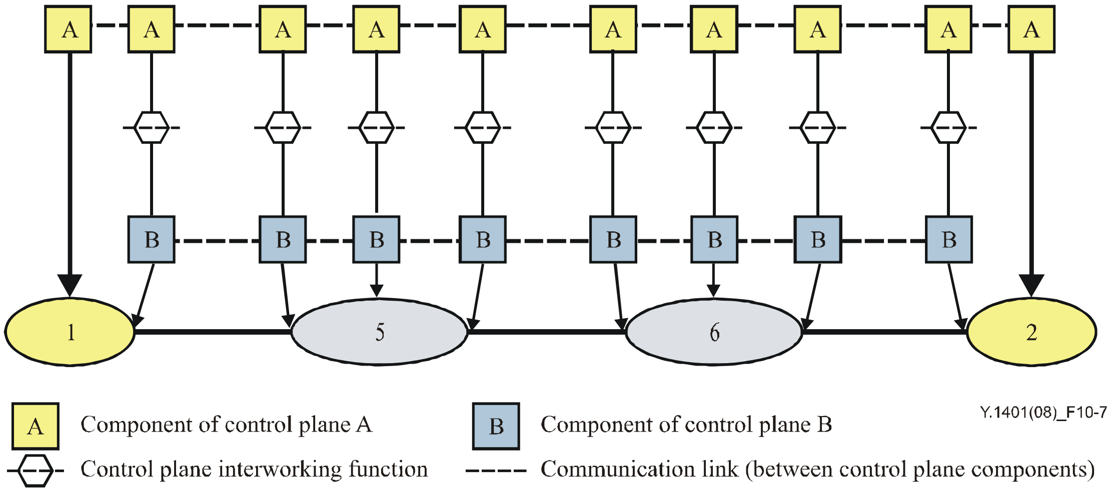

Figure 10-7 – An example control plane representation for subnetworks 1, 2, 5 and 6 and the links between them

The operation of this network is similar to the description provided in clause 10.5. The additional complexity incurred in this case is that each of the technology B subnetworks (5, 6 and 7) and the links between them must be represented by the appropriate (technology A) control entities with the associated control plane interworking functions. All of the control entities must cooperate to perform path computation and connection set-up (signalling).

As described in clause 10.5 when the connection request is signalled at the boundary of subnetwork 1, the signalling message is intercepted by the control plane interworking function at the boundary of subnetwork 1, the information relating to the technology B network is extracted and used to allocate the link connections and establish the subnetwork connections across subnetworks 5, 6 and 7. The remainder of the signalling message (i.e., the part required to set up the connection across subnetwork 2) is encapsulated and transferred to subnetwork 2 along with the link labels.

Again, call parameters need to be visible at the boundary between the different networks as different connection segments exist on either side of the boundary. The policy for mapping the user service parameters (e.g., CoS, QoS) to the connection may be different for technology A and B.

In this case, the control plane interworking functions representing subnetworks 5, 6 and 7 reside within network B and not on the network boundary.

10.6 Interworking with different network modes

Table 10-1 provides a summary of the major functions required to interwork between control planes within different layer networks.

Table 10-1 – A summary of the major functions required to interwork between control planes within different layer networks

A	B	Comments
co	co	As described above, must map the network B topology and name space into network A.
co	cl-ps	The co network must be provided a co "service" e.g., an interworking path (IWP) from the cl network thus the interworking has two parts: mapping as in the co/co case and IWP set-up.
cl-ps	cl-ps	Must map the network B topology and name space into network A.
cl-ps	co	The only service offered by the co layer is a topological link for the A cl-ps network. This is part of topology management for network A. It is not part of the information transfer process and therefore should not be considered for control plane interworking.

11 Security considerations

As this Recommendation deals with principles, security considerations have not been addressed in this Recommendation.

Bibliography

[b-ITU-T Y.1711] Recommendation ITU-T Y.1711 (2004), *Operation and maintenance mechanism for MPLS networks*.

[b-ITU-T Y.1731] Recommendation ITU-T Y.1731 (2006), *OAM functions and mechanisms for Ethernet based networks*.

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	Telephone transmission quality, telephone installations, local line networks
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, 2008