International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

M.3100

(04/2005)

SERIES M: TELECOMMUNICATION MANAGEMENT, INCLUDING TMN AND NETWORK MAINTENANCE

Telecommunications management network

Generic network information model

ITU-T Recommendation M.3100

ITU-T M-SERIES RECOMMENDATIONS

TELECOMMUNICATION MANAGEMENT, INCLUDING TMN AND NETWORK MAINTENANCE

Introduction and general principles of maintenance and maintenance organization	M.10–M.299
International transmission systems	M.300–M.559
International telephone circuits	M.560–M.759
Common channel signalling systems	M.760–M.799
International telegraph systems and phototelegraph transmission	M.800–M.899
International leased group and supergroup links	M.900–M.999
International leased circuits	M.1000–M.1099
Mobile telecommunication systems and services	M.1100–M.1199
International public telephone network	M.1200–M.1299
International data transmission systems	M.1300–M.1399
Designations and information exchange	M.1400–M.1999
International transport network	M.2000–M.2999
Telecommunications management network	**M.3000–M.3599**
Integrated services digital networks	M.3600–M.3999
Common channel signalling systems	M.4000–M.4999

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

ITU-T Recommendation M.3100

Generic network information model

Summary

This Recommendation provides a generic network information model. The model describes managed object classes and their properties that are generic and useful to describe information exchanged across all interfaces defined in M.3010 TMN architecture. These generic managed object classes are intended to be applicable across different technologies, architecture and services. The managed object classes in this Recommendation may be specialized to support the management of various telecommunications networks.

Source

ITU-T Recommendation M.3100 was approved on 29 April 2005 by ITU-T Study Group 4 (2005-2008) under the ITU-T Recommendation A.8 procedure.

This edition includes the modifications introduced by Corrigendum 1 to ITU-T Recommendation M.3100 (2005), approved on 13 November 2005 by ITU-T Study Group 4 (2005-2008) under the ITU-T Recommendation A.8 procedure.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. 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, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Page

ITU-T Recommendation M.3100

Generic network information model

1 Scope

This Recommendation provides a generic network information model. It identifies TMN object classes that are common to managed telecommunication networks; or are of a generic type that can be used to manage a network at a technology-independent level; or are super-classes of technology-specific managed objects in a telecommunication network; or management support objects that are required for the management of the telecommunication network. These objects are relevant to information exchanged across standardized interfaces defined in the M.3010 TMN architecture [1].

This Recommendation addresses generically the abstractions of those aspects of telecommunication resources (e.g., equipments, telecommunication services) required to manage the network. It also includes the abstractions related to the management services. ITU-T Rec. G.803, on the architecture of the transport network, is used as the basis in developing the transport aspects of this model.

This Recommendation does not address abstractions relevant to technology-specific areas or implementation-specific details.

1.1 Purpose

1.1.1 Interoperability

There will be a variety of TMN conformant management systems and managed systems concerning many technology-specific areas, such as switching and transmission. One purpose of this Recommendation is to provide a vehicle for management interoperability between such systems.

1.1.2 Technology-independent management

By introducing the concept of technology-independent management, it is possible to perform management of diverse equipment using common communication interfaces. In this manner, an "abstract" view over a set of network-elements can be achieved.

1.1.3 Facilitating information model development

This Recommendation also provides a framework from which technology-specific information models may be developed using the modelling principles defined in ITU-T Rec. X.720 [2].

1.2 Field of application

This Recommendation captures the generally applicable requirements of the technology-independent and technology-specific information models as well as information relating to TMN management services.

Through specialization, this Recommendation is applicable to technology-specific TMN information models. The mechanism for specialization is inheritance.

Even though technology-specific models may be derived from this Recommendation, some of the generic managed object classes in this Recommendation are instantiable in order to provide interoperability between equipment supporting information models derived from this Recommendation and equipment that only supports the information model in this Recommendation.

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.

[1] ITU-T Recommendation M.3010 (2000), *Principles for a telecommunications management network*.

[2] ITU-T Recommendation X.720 (1992) | ISO/IEC 10165-1:1993, *Information technology – Open Systems Interconnection – Structure of management information: Management information model*.

[3] ITU-T Recommendation X.722 (1992) | ISO/IEC 10165-4:1992, *Information technology – Open Systems Interconnection – Structure of management information: Guidelines for the definition of managed objects*.

[4] ITU-T Recommendation X.680 (2002) | ISO/IEC 8824-1:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*.

[5] ITU-T Recommendation X.721 (1992) | ISO/IEC 10165-2:1992, *Information technology – Open Systems Interconnection – Structure of management information: Definition of management information*.

[6] ITU-T Recommendation G.803 (2000), *Architecture of transport networks based on the synchronous digital hierarchy (SDH)*.

[7] ITU-T Recommendation Q.821 (2000), *Stage 2 and Stage 3 description for the Q3 interface – Alarm surveillance*.

[8] ITU-T Recommendation X.734 (1992) | ISO/IEC 10164-5:1993, *Information technology – Open Systems Interconnection – Systems management: Event report management function*.

3 Definitions

This Recommendation defines the following terms:

3.1 General definitions

3.1.1 management interface: Any managed entity interface that is defined for the purpose of management (e.g., OS interface, craft interface, LED indicator).

3.1.2 managed entity: A managed entity may be a managed system, a managed application, or a managed resource. This definition is dependent upon the context in which it is used.

3.1.3 managed resource: A specific component of a managed system/managed application (e.g., a specific circuit pack, termination point).

3.1.4 managed resource-specific/unit audible/visual indicator: An audible/visual alarm indicator that is specific to a single managed resource.

3.2 Alarm report-related definitions

3.2.1 alarm reporting: Process of alerting, for the purposes of management, external systems and users regarding alarms.

3.2.2 aggregate audible/visual indicators: An audible/visual alarm indicator that reflects information about a set of managed resources.

3.2.3 alarm reporting control: Involves the turning off of alarm reporting which includes inhibiting new autonomous alarm indication notification, and inhibiting the use of managed resource-specific/unit alarm information for the determination of aggregate audible/visual indicators. Autonomous alarm clear notification for previously reported alarms will not be suppressed. Alarm reporting "on" is supported by the "ALM" state. Alarm reporting "off" is supported by the "NALM-QI", "NALM-TI", and "NALM" states.

3.2.4 ARC interval: Generic term that applies to both the persistence and timed intervals.

3.2.5 inhibited: This term is used throughout this feature description to identify that reporting is off (or in other words, is not allowed).

3.2.6 persistence interval: A period of time for which a managed entity must be free of qualified problems.

3.2.7 timed interval: A period of time.

3.2.8 qualified problem: A problem that affects the operability of the managed entity and used to qualify transitions between the "NALM-NR" and "NALM-CD" states. Additional detail for this definition is managed resource specific and is to be defined by the managed resource.

3.3 ARC state definitions

3.3.1 ALM: ALarM reporting: Alarm Reporting is turned on.

3.3.2 NALM: No ALarM reporting: Alarm Reporting is turned off.

3.3.3 NALM-TI: No ALarM reporting, Timed Inhibit: Alarm Reporting is turned off for a specified timed interval.

3.3.4 NALM-QI: No ALarM reporting; Qualified Inhibit: Alarm Reporting is turned off until the managed entity is qualified problem-free for a specified persistence interval.

3.3.5 NALM-CD: No ALarM reporting, CountDown: This is a substate of NALM-QI and performs the persistence timing countdown function when the managed entity is qualified problem-free.

3.3.6 NALM-NR: No ALarM reporting, NotReady: This is a substate of NALM-QI and performs a wait function until the managed entity is qualified problem-free.

4 Abbreviations

This Recommendation uses the following abbreviations:

ANSI	American National Standards Institute
ASAP	Alarm Severity Assignment Profile
ASN.1	Abstract Syntax Notation One
ATIS	Alliance for Telecommunications Industry Solutions
AVC	Attribute Value Change Notification
CCITT	International Telegraph and Telephone Consultative Committee (replaced by ITU-T)
CMIP	Common Management Information Protocol
CMISE	Common Management Information Service Element
DCN	Data Communication Network

EFD	Event Forwarding Discriminator
ET	Event Time
GDMO	Guidelines for the Definition of Managed Objects
Ind	Indication
ISO	International Organization for Standardization
ISP	International Standardized Profile
ITU	International Telecommunication Union
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
MCS	Management Conformance Summary
MIB	Management Information Base
MICS	Management Information Conformance Statement
MIDS	Management Information Definition Statement
MIM	Management Information Model
MOC	Managed Object Class
MOCS	Managed Object Conformance Statement
MOI	Managed Object Instance
MRCS	Managed Relationship Conformance Statement
OS	Operations System
OSI	Open Systems Interconnection
PC	Probable Cause
PDU	Protocol Data Unit
PICS	Protocol Implementation Conformance Statement
RDN	Relative Distinguished Name
Req	Request
ROSE	Remote Operations Service Element
Rsp	Response
RTR	Reset Threshold Report
SCN	State Change Notification
SMAP	System Management Application Protocol
SP	Specific Problem
TMN	Telecommunication Management Network
TR	Threshold Report (a.k.a. quality of service alarm)

5 Conventions

There are no specific conventions defined in this Recommendation.

6 Overview of the model

A generic network information model is essential to the generation of uniform fault, configuration, performance, security, and accounting management standards. A common network model, identifying the generic resources that exist in a network and their associated attribute types, events, actions, and behaviours, provides a foundation for understanding the interrelationships between these resources and attributes, and may, in turn, promote uniformity in dealing with the various aspects of managing these resources and attributes.

Network resources may be customer- or provider-owned; the latter includes portions that may be assigned for exclusive use by specific customers. Resources may be physical or logical in nature. Physical resources include customer (e.g., PBXs) or provider (e.g., digital cross-connect systems) systems, their associated subsystems (e.g., a line card within a PBX) and also the links that interconnect these systems. Such systems are generally known as Network Elements (NEs). Logical resources include communication protocols, application programs, logs, and network services.

There may also exist (separate or integrated) Telecommunication Management Network (TMN) resources involved in operating a telecommunication network. These resources include the Operations Systems (OSs) closely associated with managing specific NEs, and OSs that have network-wide responsibilities.

Resources have attributes that allow the user to control and/or observe the behaviour of the resource. Attributes may also allow the user to control and/or observe the relationships between resources.

There is a need to represent the way resources, or entities can be combined and interrelated (relationships). In this version, UML class diagram techniques have been used to represent inter-object relationships.

These UML diagrams result in a high-level view (schema) of the model. This view can be used to derive information related to naming, to verifying consistency, and to ensuring completeness. For example, it ensures that sufficient information (i.e., relationships) is provided from a physical resource to identify the services that are dependent on that resource.

The information exchanged at the management interface is modelled using design principles outlined in ITU-T Rec. X.720 [2], "Management Information Model". Resources are modelled as objects, and the management view of a resource is called a managed object. Additional objects, called support managed objects, are defined to support the functions of managing a telecommunication network.

Objects with similar attributes and behaviours may be grouped into object classes. An object is characterized by its object class and object instance, and may possess multiple attribute types and associated values. Similarly, the terms managed object class and managed object instance apply specifically to objects that are being managed. This Recommendation specifies the properties of the resource (i.e., managed object) visible for management.

An object class may be a subclass of another object class. A subclass inherits attribute types and behaviours of its super-class, in addition to possessing its own specific attributes and properties.

Object classes and attribute types are defined only for the purpose of communicating network management messages between systems, and need not be related to the structuring of data within these systems. Some object classes defined in these issues (and future issues) of the model apply to many management functional areas, while others support specific functional areas.

Annex A contains an index of managed object classes, packages, attributes, notifications and actions defined in this Recommendation.

There are several different viewpoints of management information which may be defined for management purposes, with the Network Element level viewpoint, the Network level viewpoint and the Service level viewpoint defined below. These viewpoints are not restrictive but define the levels of abstraction of particular types of interfaces. That is, object class definitions are not forced into this categorization but are constructed to meet the needs of exchanging management information across TMN interfaces. Objects defined for a given viewpoint may be used in others, and any object may be used by any interface which requires it. The definition of viewpoint is a means of generating requirements, hence there is no implicit definition of interfaces or storage requirements. This information is defined for the purpose of management via an open interface.

The Network Element level viewpoint is concerned with the information that is required to manage a Network Element (NE). This refers to the information required to manage the Network Element Function (NEF) and the physical aspects of the NE. The information may be derived from open systems other than the NE.

The Network level viewpoint is concerned with the information representing the network, both physically and logically. It is concerned with how network element entities are related, topographically interconnected, and configured to provide and maintain end-to-end connectivity.

The Service level viewpoint is concerned with how Network level aspects (such as an end-to-end path) are utilized to provide a network service and, as such, is concerned with the requirements of a network service (e.g., availability, cost, etc.) and how these requirements are met through the use of the network, and all related customer information.

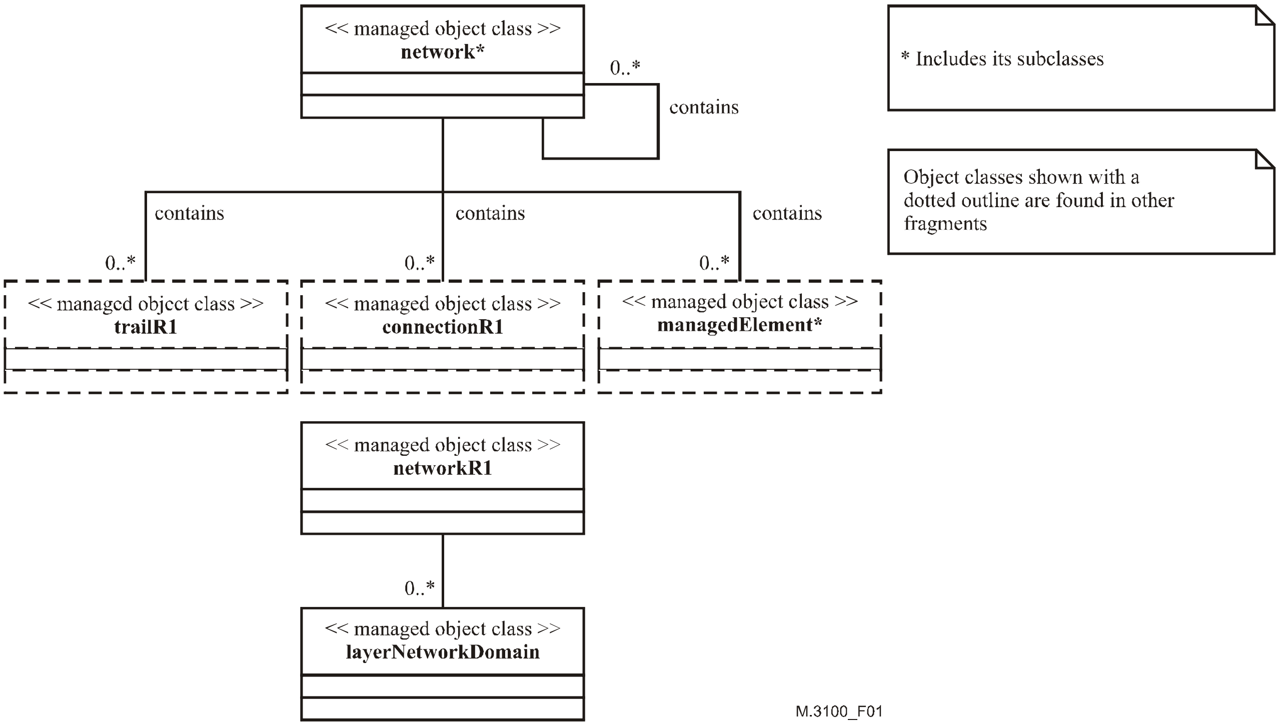
The object classes that form the basis for the generic network information model are grouped into fragments. The purpose of defining fragments is only to have a document that is easier to read by grouping a limited number of object class definitions. Each fragment deals with a particular subject (e.g., network, managed element, transmission, support objects) but object classes of each fragment will be usable in various models depending on the functional area managed and/or on the level viewpoint considered.

The following subclauses provide a high-level overview of each fragment with an emphasis on the containment relationship.

6.1 Network fragment

The network fragment is a grouping of object classes that represents collections of interconnected telecommunications and management objects (logical or physical) capable of exchanging information. These objects have one or more common characteristics, for example they may be owned by a single customer or provider, or associated with a specific service network. A network may be nested within another (larger) network, thereby forming a containment relationship.

Managed object classes in the network fragment are presented in Figure 1.

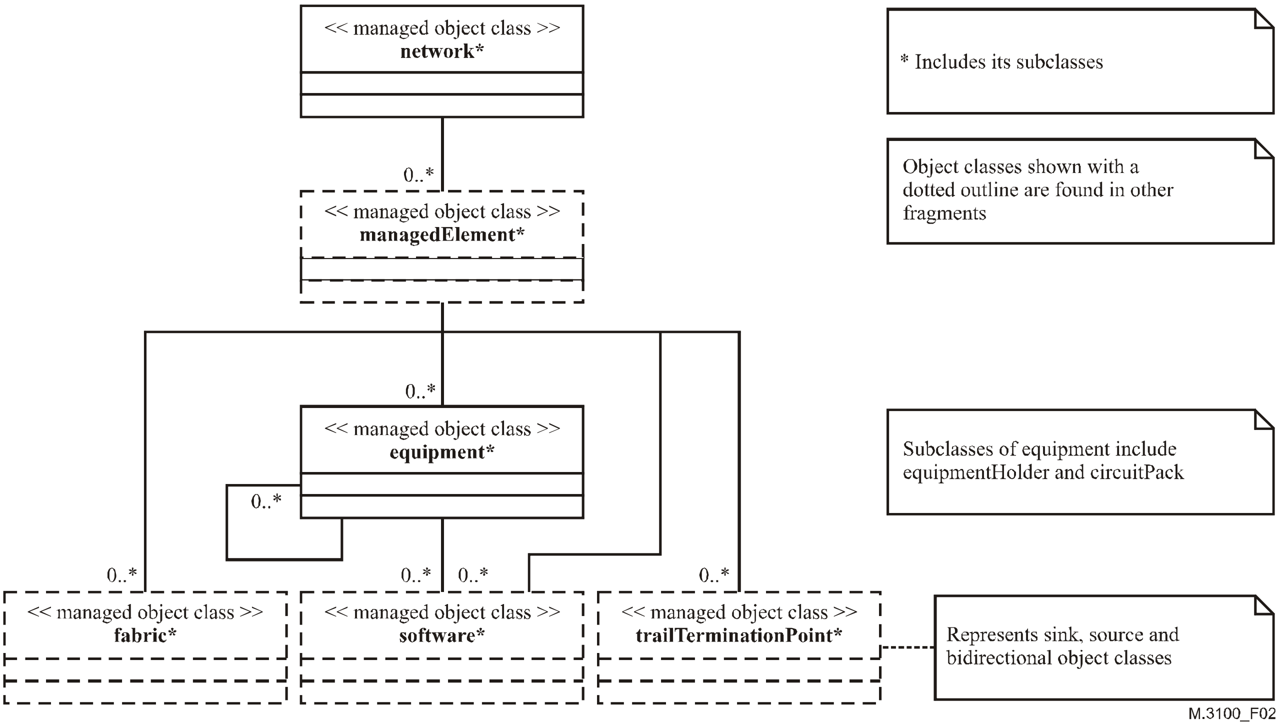


<< managed object class >>
network*
0..*
contains
* Includes its subclasses
Object classes shown with a dotted outline are found in other fragments
contains
contains
contains
0..*
0..*
0..*
<< managed object class >>
trailR1
<< managed object class >>
connectionR1
<< managed object class >>
managedElement*
<< managed object class >>
networkR1
0..*
<< managed object class >>
layerNetworkDomain
M.3100_F01

Figure 1/M.3100 – Network fragment

6.2 Managed element fragment

Managed object classes in managed element fragment are presented in Figure 2.

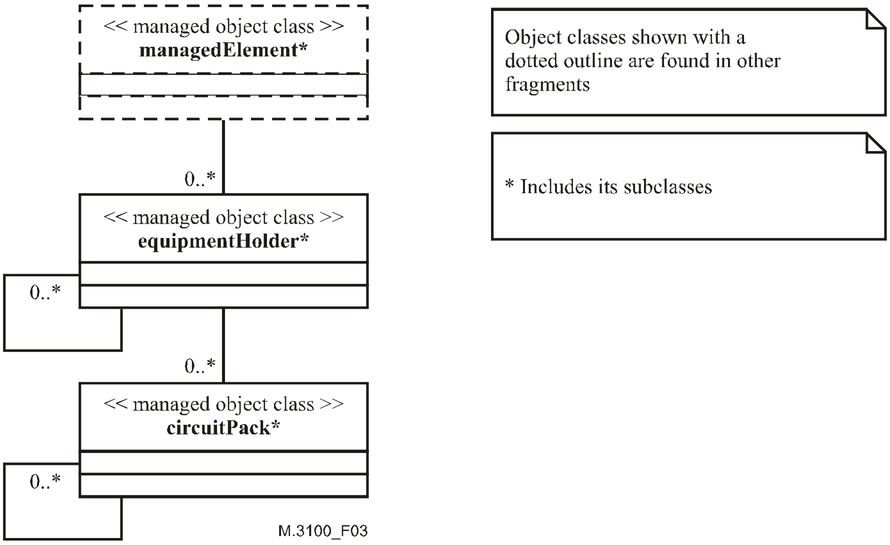


<< managed object class >>
network*
* Includes its subclasses
0..*
Object classes shown with a dotted outline are found in other fragments
<< managed object class >>
managedElement*
0..*
<< managed object class >>
equipment*
Subclasses of equipment include equipmentHolder and circuitPack
0..*
0..*
0..*
0..*
0..*
<< managed object class >>
fabric*
<< managed object class >>
software*
<< managed object class >>
trailTerminationPoint*
Represents sink, source and bidirectional object classes
M.3100_F02

Figure 2/M.3100 – Managed element fragment

6.3 Physical equipment fragment

Managed object classes in the physical equipment fragment are presented in Figure 3.

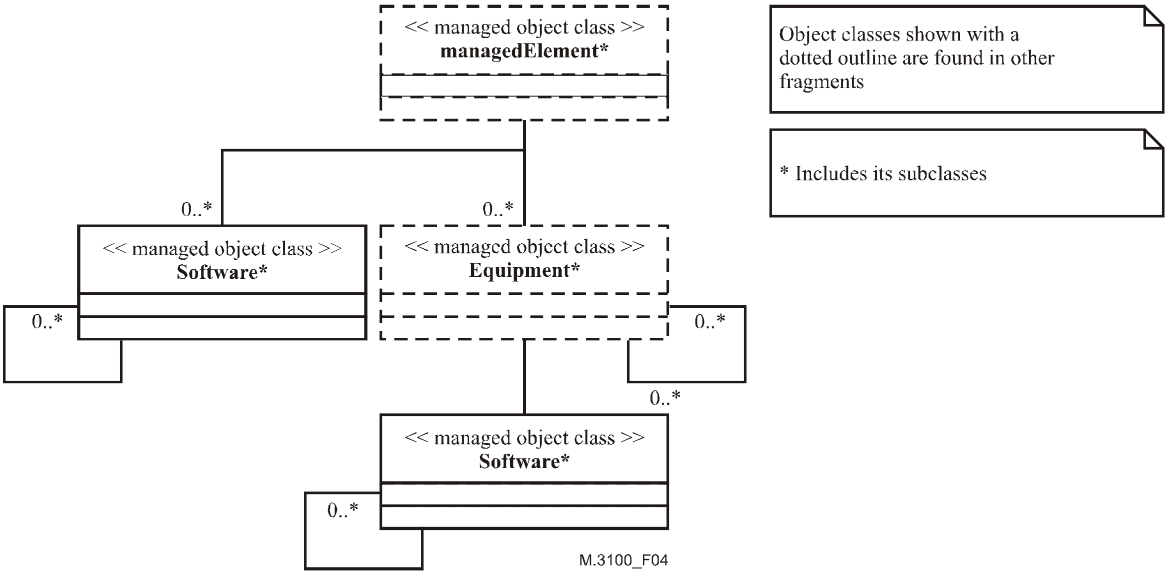


<< managed object class >>
managedElement*
0..*
<< managed object class >>
equipmentHolder*
0..*
0..*
<< managed object class >>
circuitPack*
0..*
M.3100_F03
Object classes shown with a dotted outline are found in other fragments
* Includes its subclasses

Figure 3/M.3100 – Physical equipment fragment

6.4 Logical equipment fragment

Managed object classes in the logical equipment fragment are presented in Figure 4.

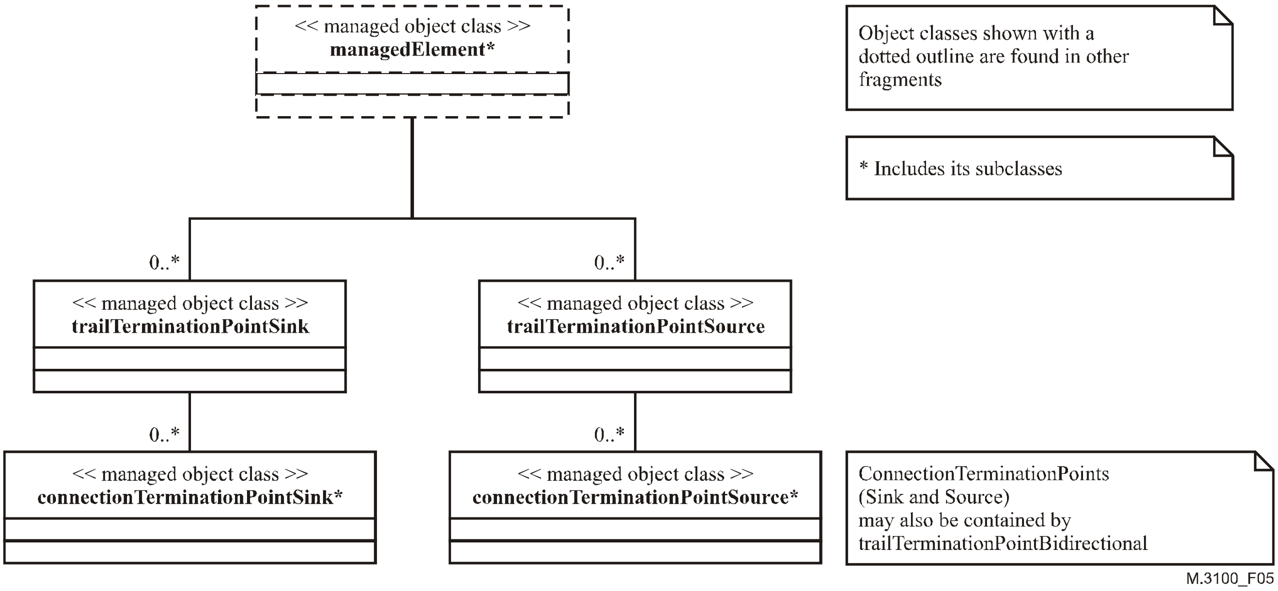


<< managed object class >>
managedElement*
0..*
0..*
<< managed object class >>
Software*
0..*
<< managed object class >>
Equipment*
0..*
0..*
<< managed object class >>
Software*
0..*
M.3100_F04
Object classes shown with a dotted outline are found in other fragments
* Includes its subclasses

Figure 4/M.3100 – Logical equipment fragment

6.5 Termination point fragment – Network element view

Managed object classes in termination point fragment for the element view are presented in Figure 5.

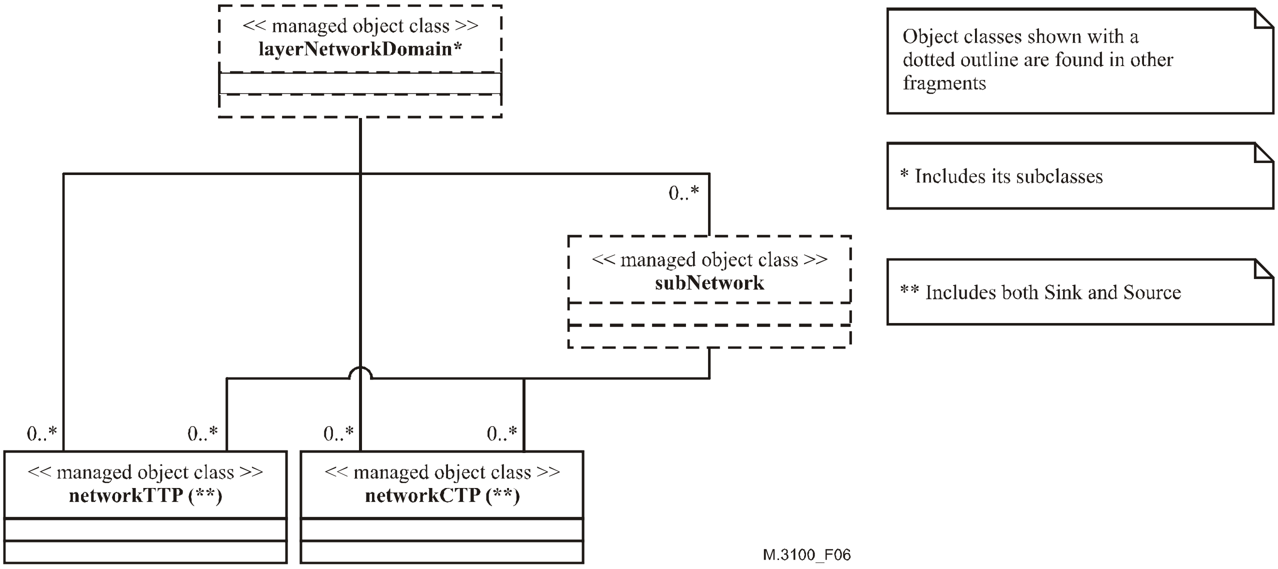


<< managed object class >>
managedElement*
Object classes shown with a dotted outline are found in other fragments
* Includes its subclasses
0..*
0..*
<< managed object class >>
trailTerminationPointSink
<< managed object class >>
trailTerminationPointSource
0..*
0..*
<< managed object class >>
connectionTerminationPointSink*
<< managed object class >>
connectionTerminationPointSource*
ConnectionTerminationPoints (Sink and Source) may also be contained by trailTerminationPointBidirectional
M.3100_F05

Figure 5/M.3100 – Termination point fragment – Network element view

6.6 Termination point fragment – Network view

Managed object classes in termination point fragment for the network view are presented in Figure 6.

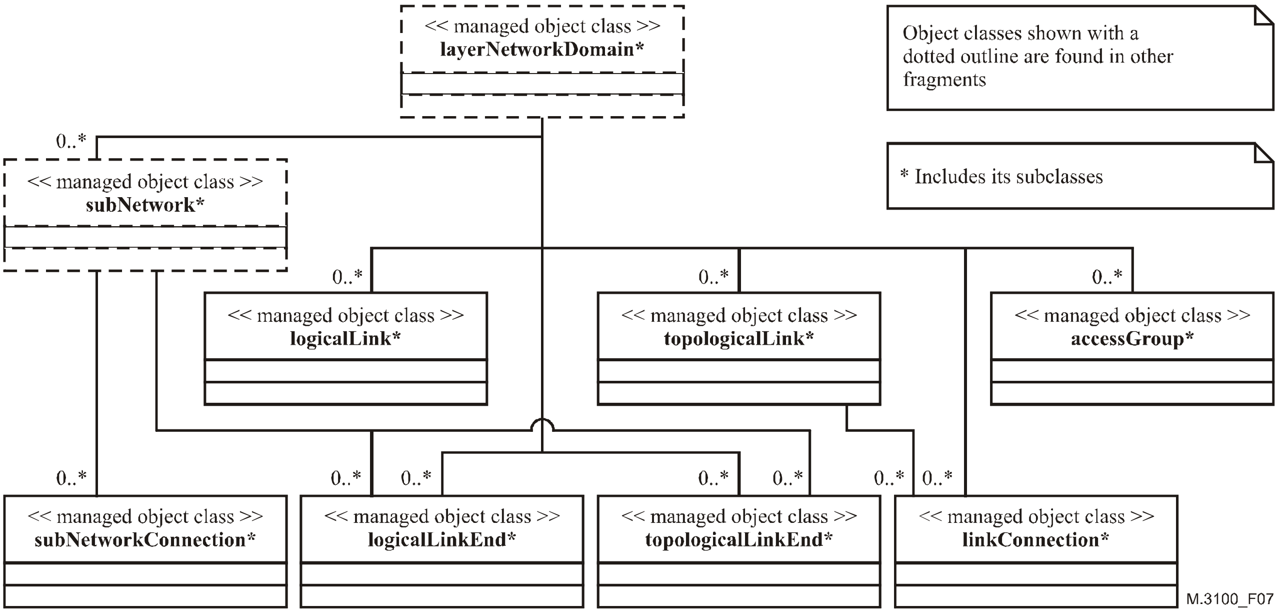


<< managed object class >>
layerNetworkDomain*
Object classes shown with a dotted outline are found in other fragments
* Includes its subclasses
0..*
<< managed object class >>
subNetwork
** Includes both Sink and Source
0..*
0..*
0..*
0..*
<< managed object class >>
networkTTP (**)
<< managed object class >>
networkCTP (**)
M.3100_F06

Figure 6/M.3100 – Termination point fragment – Network view

6.7 Topology and connectivity fragment – Network view

Managed object classes in the topology and connectivity fragment for the network view are presented in Figure 7.

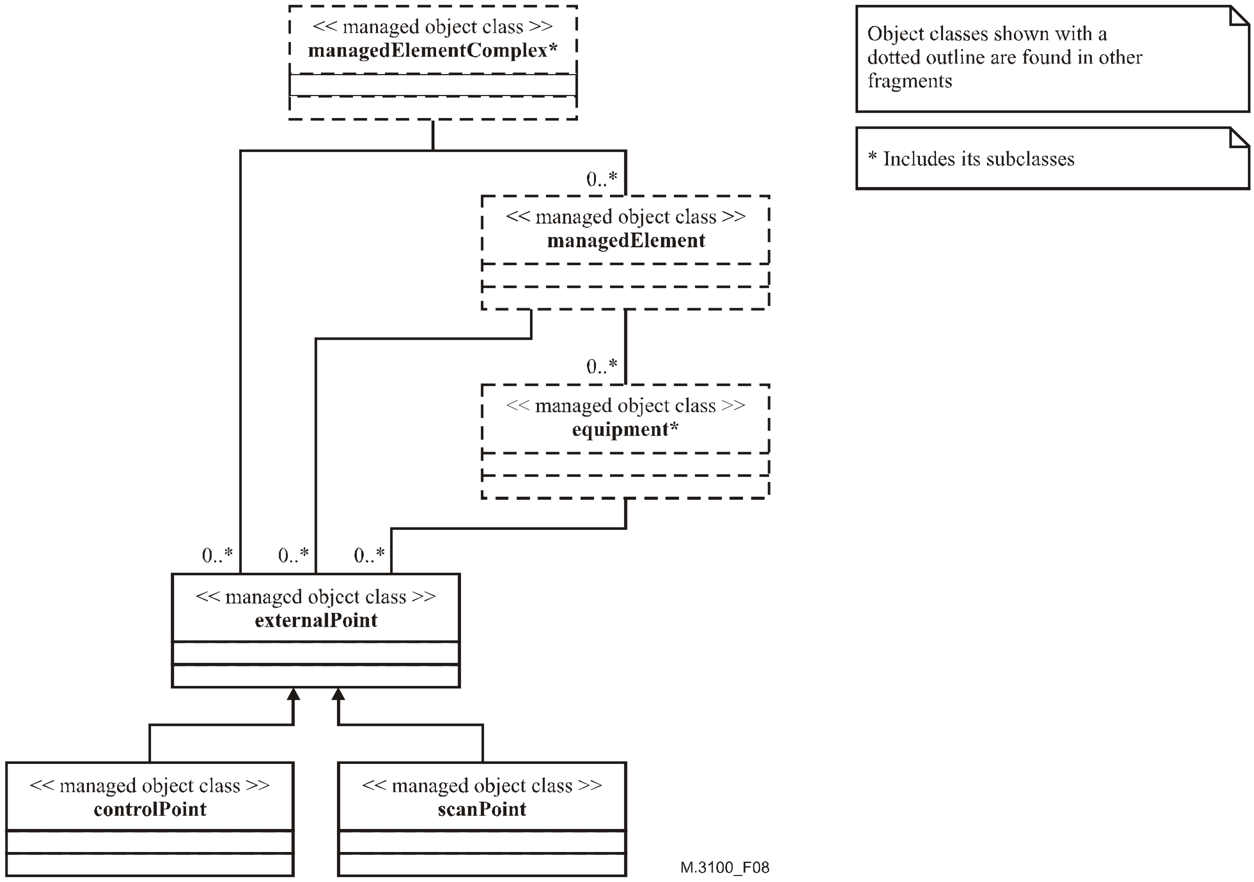


<< managed object class >>
layerNetworkDomain*
Object classes shown with a dotted outline are found in other fragments
* Includes its subclasses
0..*
<< managed object class >>
subNetwork*
0..*
<< managed object class >>
logicalLink*
0..*
<< managed object class >>
topologicalLink*
0..*
<< managed object class >>
accessGroup*
0..*
0..*
0..*
0..*
0..*
0..*
0..*
<< managed object class >>
subNetworkConnection*
<< managed object class >>
logicalLinkEnd*
<< managed object class >>
topologicalLinkEnd*
<< managed object class >>
linkConnection*
M.3100_F07

Figure 7/M.3100 – Topology and connectivity fragment

6.8 Telemetry fragment

Managed object classes in the telemetry fragment are presented in Figure 8.

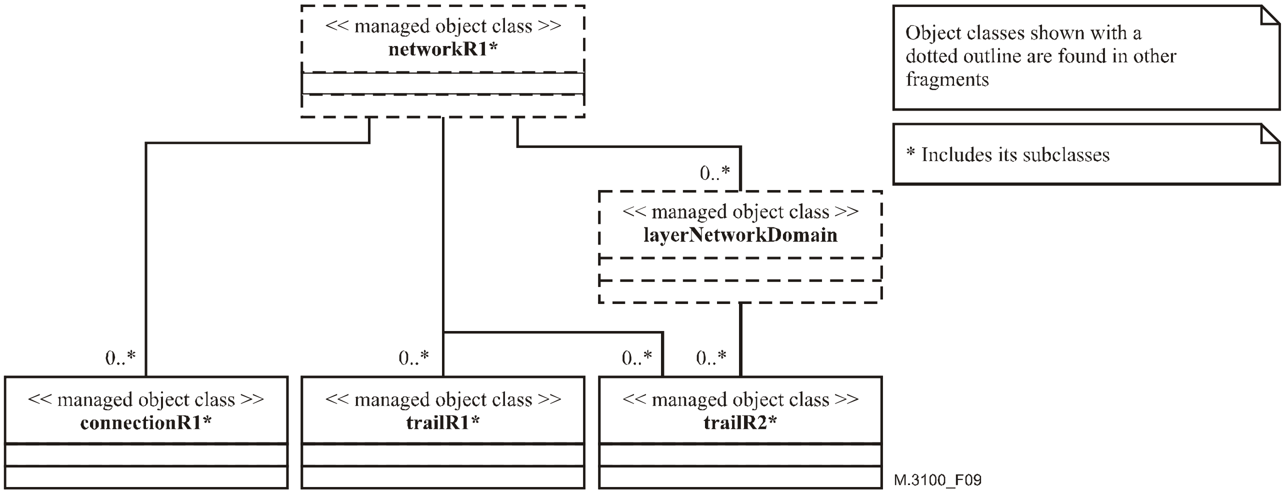


<< managed object class >>
managedElementComplex*
Object classes shown with a dotted outline are found in other fragments
* Includes its subclasses
0..*
<< managed object class >>
managedElement
0..*
<< managed object class >>
equipment*
0..*
0..*
0..*
<< managed object class >>
externalPoint
<< managed object class >>
controlPoint
<< managed object class >>
scanPoint
M.3100_F08

Figure 8/M.3100 – Telemetry fragment

6.9 Transmission fragment

Managed object classes in the transmission fragment are presented in Figure 9.

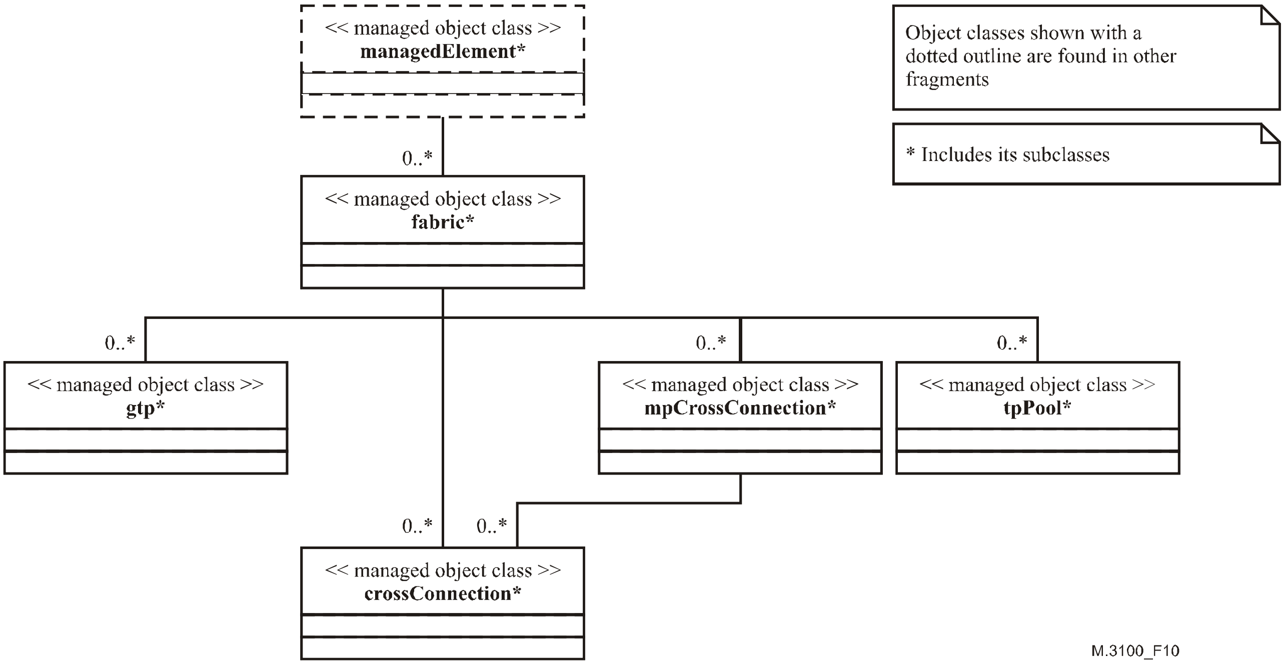


<< managed object class >>
networkR1*
Object classes shown with a dotted outline are found in other fragments
* Includes its subclasses
0..*
<< managed object class >>
layerNetworkDomain
0..*
0..*
0..*
0..*
<< managed object class >>
connectionR1*
<< managed object class >>
trailR1*
<< managed object class >>
trailR2*
M.3100_F09

Figure 9/M.3100 – Transmission fragment

6.10 Cross-connection fragment

Managed object classes in the cross-connect fragment are presented in Figure 10.



<< managed object class >>
managedElement*
Object classes shown with a dotted outline are found in other fragments
* Includes its subclasses
0..*
<< managed object class >>
fabric*
0..*
0..*
0..*
<< managed object class >>
gtp*
<< managed object class >>
mpCrossConnection*
<< managed object class >>
tpPool*
0..*
0..*
<< managed object class >>
crossConnection*
M.3100_F10

Figure 10/M.3100 – Cross-connect fragment

6.11 Functional area fragment

This fragment contains the following managed object classes:

– Alarm Record (defined in ITU-T Rec. X.721 [5]);

– Alarm Severity Assignment Profile;

– Attribute Value Change Record (defined in ITU-T Rec. X.721 [5]);

– Current Alarm Summary Control (defined in ITU-T Rec. Q.821 [7]);

– Discriminator (defined in ITU-T Rec. X.721 [5]);

– Event Forwarding Discriminator (defined in ITU-T Rec. X.721 [5]);

– Event Log Record (defined in ITU-T Rec. X.721 [5]);

– Log (defined in ITU-T Rec. X.721 [5]);

– Log Record (defined in ITU-T Rec. X.721 [5]);

– Management Operations Schedule (defined in ITU-T Rec. Q.821 [7]);

– Object Creation Record (defined in ITU-T Rec. X.721 [5]);

– Object Deletion Record (defined in ITU-T Rec. X.721 [5]);

– State Change Record (defined in ITU-T Rec. X.721 [5]).

7 GDMO definitions

This clause is available as an electronic file attached to this Recommendation.

8 ASN.1 modules

This clause is available as an electronic file attached to this Recommendation.

9 TMN application context

The object identifier value

> {itu-t recommendation m(13) gnm(3100) protocolSupport(1) applicationContext(0) tmnApplicationContextOne(1)}

is assigned to the application context that has the same capabilities as the systems management application context defined in ITU-T Rec. X.701, but also supports the integer values for ProbableCause. These integer value assignments are specified in this Recommendation.

Annex A

Index

A.1 Managed objects

Managed object class	Object identifier
AbstractLink	m3100ObjectClass 44
AbstractLinkEnd	m3100ObjectClass 45
abstractLinkEndR1	m3100ObjectClass 70
AccessGroup	m3100ObjectClass 46
AlarmReportingControlManager	m3100ObjectClass 78
AlarmSeverityAssignmentProfile	m3100ObjectClass 22
ArcIntervalProfile	m3100ObjectClass 66
AttributeRanges	m3100ObjectClass 75
CircuitEndPointSubgroup	m3100ObjectClass 31
CircuitPack	m3100ObjectClass 30
CircuitPackR1	m3100ObjectClass 43
ConnectionR1	m3100ObjectClass 23
ConnectionTerminationPointBidirectional	m3100ObjectClass 5
ConnectionTerminationPointSink	m3100ObjectClass 6
ConnectionTerminationPointSource	m3100ObjectClass 7
ControlPoint	m3100ObjectClass 41
CrossConnection	m3100ObjectClass 15
crossConnectionR1	m3100ObjectClass 37
Equipment	m3100ObjectClass 2
EquipmentR1	m3100ObjectClass 28
EquipmentR2	m3100ObjectClass 35
EquipmentHolder	m3100ObjectClass 32
ExternalPoint	m3100ObjectClass 40
Fabric	m3100ObjectClass 16
fabricR1	m3100ObjectClass 26
fabricR2	m3100ObjectClass 39

Managed object class	Object identifier
fabricR3	m3100ObjectClass 73
fabricR4	m3100ObjectClass 74
GenericTransportTTP	m3100ObjectClass 76
genericTransportTTPR1	m3100ObjectClass 80
Gtp	m3100ObjectClass 17
gtpR1	m3100ObjectClass 38
LayerNetworkDomain	m3100ObjectClass 47
layerNetworkDomainR1	m3100ObjectClass 69
LinkConnection	m3100ObjectClass 48
LogicalLink	m3100ObjectClass 49
LogicalLinkEnd	m3100ObjectClass 50
logicalLinkEndR1	m3100ObjectClass 71
ManagedElement	m3100ObjectClass 3
managedElementR1	m3100ObjectClass 27
managedElementR2	m3100ObjectClass 77
ManagedElementComplex	m3100ObjectClass 34
MpCrossConnection	m3100ObjectClass 18
mpCrossConnectionR1	m3100ObjectClass 36
NamedCrossConnection	m3100ObjectClass 19
namedMpCrossConnection	m3100ObjectClass 20
Network	m3100ObjectClass 1
networkR1	m3100ObjectClass 33
NetworkCTPBidirectional	m3100ObjectClass 51
NetworkCTPSink	m3100ObjectClass 52
NetworkCTPSource	m3100ObjectClass 53
NetworkTTPBidirectional	m3100ObjectClass 55
networkTTPBidirectionalR1	m3100ObjectClass 68
NetworkTTPSink	m3100ObjectClass 56
networkTTPSinkR1	m3100ObjectClass 67
NetworkTTPSource	m3100ObjectClass 57
NetworkTerminationPoint	m3100ObjectClass 54
PhysicalPort	m3100ObjectClass 79
Pipe	m3100ObjectClass 24
pipeR2	m3100ObjectClass 58
protectionGroupR2	m3100ObjectClass 64
protectionUnitR1	m3100ObjectClass 65
ScanPoint	m3100ObjectClass 42
Software	m3100ObjectClass 4
softwareR1	m3100ObjectClass 29
SubNetwork	m3100ObjectClass 59
SubNetworkConnection	m3100ObjectClass 60

Managed object class	Object identifier
TerminationPoint	m3100ObjectClass 8
TopologicalLink	m3100ObjectClass 61
TopologicalLinkEnd	m3100ObjectClass 62
topologicalLinkEndR1	m3100ObjectClass 72
TpPool	m3100ObjectClass 21
trailR1	m3100ObjectClass 25
trailR2	m3100ObjectClass 63
trailTerminationPointBidirectional	m3100ObjectClass 9
TrailTerminationPointSink	m3100ObjectClass 10
trailTerminationPointSource	m3100ObjectClass 11

A.2 Packages

Package	Object identifier
administrativeOperationalStatesPackage	m3100Package 1
AffectedObjectListPackage	m3100Package 2
alarmSeverityAssignmentPointerPackage	m3100Package 3
ArcPackage	m3100Package 94
arcRetrieveAlarmDetailPackage	m3100Package 95
attributeValueChangeNotificationPackage	m3100Package 4
audibleVisualLocalAlarmPackage	m3100Package 5
ChannelNumberPackage	m3100Package 6
characteristicInformationPackage	m3100Package 7
circuitPackConfigurationPackage	m3100Package 44
CircuitPackResetPackage	m3100Package 45
ClientCTPListPackage	m3100Package 49
clientConnectionListPackage	m3100Package 35
clientLinkConnectionPointerListPackage	m3100Package 50
clientLinkEndPointerPackage	m3100Package 51
ClientLinkPointerPackage	m3100Package 52
ClientTrailPackage	m3100Package 9
ComponentPointerPackage	m3100Package 53
CompositePointerPackage	m3100Package 54
configuredConnectivityPackage	m3100Package 55
connectivityPointerPackage	m3100Package 56
containedAccessGroupListPackage	m3100Package 57
containedBoardPackage	m3100Package 48
containedInSubNetworkListPackage	m3100Package 58
containedLinkEndListPackage	m3100Package 59
containedLinkListPackage	m3100Package 60
containedNetworkTPListPackage	m3100Package 61

Package	Object identifier
containedSubNetworkListPackage	m3100Package 62
createDeleteNotificationsPackage	m3100Package 10
crossConnectionPointerPackage	m3100Package 11
ctpInstancePackage	m3100Package 12
currentProblemListPackage	m3100Package 13
environmentalAlarmPackage	m3100Package 14
environmentalAlarmR1Package	m3100Package 36
environmentalAlarmR2Package	m3100Package 96
equipmentAlarmEffectOnServicePackage	m3100Package 38
equipmentsEquipmentAlarmPackage	m3100Package 15
equipmentsEquipmentAlarmR1Package	m3100Package 37
equipmentsEquipmentAlarmR2Package	m3100Package 97
externalTimePackage	m3100Package 16
layerConnectionListPackage	m3100Package 63
linkConnectionPointerListPackage	m3100Package 65
linkEndCapacityPackage	m3100Package 66
linkPointerListPackage	m3100Package 67
locationNamePackage	m3100Package 17
logicalLinkCapacityPackage	m3100Package 64
maximumLinkConnectionCountPackage	m3100Package 68
maximumNetworkCTPCountPackage	m3100Package 69
multicastConversionPkg	m3100Package 102
namedCrossConnectionPackage	Not registered
neAliasPackage	m3100Package 106
neAssignmentPackage	m3100Package 70
networkCTPPackage	m3100Package 72
networkCTPsInLinkEndListPackage	m3100Package 71
networkLevelPackage	m3100Package 18
networkTPPointerPackage	m3100Package 73
normalControlStatePackage	m3100Package 43
numberOfPortPackage	m3100Package 46
objectManagementNotificationsPackage	m3100Package 20
operationalStatePackage	m3100Package 19
portAssociationsPackage	m3100Package 47
potentialCapacityPackage	m3100Package 105
potentialLinkCapacityPackage	m3100Package 74
potentialLinkEndCapacityPackage	m3100Package 75
processingErrorAlarmPackage	m3100Package 21
processingErrorAlarmR1Package	m3100Package 39
processingErrorAlarmR2Package	m3100Package 98
protectedPackage	m3100Package 22

Package	Object identifier
protectionAlarmPkg	m3100Package 93
provisionedLinkCapacityPackage	m3100Package 76
provisionedLinkConnectionCountPackage	m3100Package 77
provisionedLinkEndCapacityPackage	m3100Package 78
provisionedNetworkCTPCountPackage	m3100Package 79
qualityOfConnectivityServicePackage	m3100Package 80
redlinePackage	m3100Package 42
relatedRoutingProfilePackage	m3100Package 81
resetAudibleAlarmPackage	m3100Package 23
serverConnectionListPackage	m3100Package 24
serverTTPPointerPackage	m3100Package 82
serverTrailListPackage	m3100Package 25
sncPointerPackage	m3100Package 83
SncpPkg	m3100Package 103
softwareProcessingErrorAlarmPackage	m3100Package 26
softwareProcessingErrorAlarmR1Package	m3100Package 40
softwareProcessingErrorAlarmR2Package	m3100Package 99
SplitJoinPkg	m3100Package 101
stateChangeNotificationPackage	m3100Package 28
subordinateCircuitPackPackage	m3100Package 41
supportableClientListPackage	m3100Package 27
supportedByPackage	m3100Package 84
systemTimingSourcePackage	m3100Package 29
tmnCommunicationsAlarmInformationPackage	m3100Package 30
tmnCommunicationsAlarmInformationR1Package	m3100Package 100
topologicalLinkCapacityPackage	m3100Package 85
topologicalLinkEndCapacityPackage	m3100Package 86
totalLinkCapacityPackage	m3100Package 87
totalLinkEndCapacityPackage	m3100Package 88
trafficDescriptorPackage	m3100Package 89
ttpInstancePackage	m3100Package 31
ttpPortIDPackage	m3100Package 104
unknownStatusPackage	m3100Package 90
usageCostPackage	m3100Package 91
usageStatePackage	m3100Package 92
userLabelPackage	m3100Package 32
vendorNamePackage	m3100Package 33
versionPackage	m3100Package 34

A.3 Attributes

Attribute	Object identifier
alarmSeverityAssignmentProfilePointer	m3100Attribute 5
a-TPInstance	m3100Attribute 1
Aend	m3100Attribute 85
aEndNetworkTPList	m3100Attribute 86
acceptableCircuitPackTypeList	m3100Attribute 58
accessGroupId	m3100Attribute 83
accessPointList	m3100Attribute 84
affectedObjectList	m3100Attribute 2
alarmReportingControlList	m3100Attribute 165
alarmReportingControlManagerId	m3100Attribute 166
alarmSeverityAssignmentList	m3100Attribute 3
alarmSeverityAssignmentProfileId	m3100Attribute 4
AlarmStatus	m3100Attribute 6
arcDefaultNALMTIInterval	m3100Attribute 148
arcDefaultNALMCDInterval	m3100Attribute 149
arcIntervalProfileId	m3100Attribute 150
arcIntervalProfilePointer	m3100Attribute 151
arcManagementRequestedInterval	m3100Attribute 152
arcProbableCauseList	m3100Attribute 153
ArcState	m3100Attribute 154
ArcQIStatus	m3100Attribute 155
arcTimeRemaining	m3100Attribute 156
assignedLinkEndCapacity	m3100Attribute 87
attributeRangesId	m3100Attribute 164
availableLinkEndCapacity	m3100Attribute 88
availableLinkCapacity	m3100Attribute 89
availableSignalRateList	m3100Attribute 77
CTPId	m3100Attribute 13
channelNumber	m3100Attribute 7
characteristicInformation	m3100Attribute 8
circuitDirectionality	m3100Attribute 66
circuitEndPointSubgroupId	m3100Attribute 61
circuitPackType	m3100Attribute 54
clientCTPList	m3100Attribute 90
clientConnectionList	m3100Attribute 53
clientLinkConnectionPointerList	m3100Attribute 93
clientLinkEndPointerList	m3100Attribute 91
clientLinkPointerList	m3100Attribute 92
ClientTrail	m3100Attribute 10

Attribute	Object identifier
componentPointers	m3100Attribute 94
compositePointer	m3100Attribute 95
configuredConnectivity	m3100Attribute 96
connectedTpCount	m3100Attribute 11
connectionId	m3100Attribute 12
connectionList	m3100Attribute 97
connectivityPointer	m3100Attribute 98
connectorType	m3100Attribute 170
containedAccessGroupList	m3100Attribute 99
containedInSubNetworkList	m3100Attribute 100
containedLinkEndList	m3100Attribute 101
containedLinkList	m3100Attribute 102
containedNetworkTPList	m3100Attribute 103
containedSubNetworkList	m3100Attribute 104
crossConnectionId	m3100Attribute 14
crossConnectionName	m3100Attribute 15
crossConnectionObjectPointer	m3100Attribute 16
currentControlState	m3100Attribute 71
currentProblemList	m3100Attribute 17
directionality	m3100Attribute 18
downstreamConnectivityPointer	m3100Attribute 19
equipmentHolderAddress	m3100Attribute 56
equipmentHolderType	m3100Attribute 57
EquipmentId	m3100Attribute 20
externalTime	m3100Attribute 21
externalPointId	m3100Attribute 74
externalPointMessage	m3100Attribute 76
FabricId	m3100Attribute 22
fromTermination	m3100Attribute 23
GtpId	m3100Attribute 24
HolderStatus	m3100Attribute 59
IdleTpCount	m3100Attribute 25
informationTransferCapabilities	m3100Attribute 65
Kind	m3100Attribute 157
labelOfFarEndExchange	m3100Attribute 63
layerNetworkDomainId	m3100Attribute 105
linkConnectionPointerList	m3100Attribute 106
linkDirectionality	m3100Attribute 107
LinkEndId	m3100Attribute 108
LinkId	m3100Attribute 109
LinkPointer	m3100Attribute 110

Attribute	Object identifier
linkPointerList	m3100Attribute 111
listOfCharacteristicInfo	m3100Attribute 26
locationName	m3100Attribute 27
lockedInCondition	m3100Attribute 145
logicalEndDirectionality	m3100Attribute 112
managedElementComplexId	m3100Attribute 68
managedElementId	m3100Attribute 28
managedElementType	m3100Attribute 158
maximumLinkConnectionCount	m3100Attribute 113
maximumNetworkCTPCount	m3100Attribute 114
ModelCode	m3100Attribute 159
mpCrossConnectionId	m3100Attribute 29
NeAliases	m3100Attribute 160
neAssignmentPointer	m3100Attribute 115
networkCTPsInLinkEndList	m3100Attribute 116
NetworkId	m3100Attribute 30
networkLevelPointer	m3100Attribute 31
networkTPPointer	m3100Attribute 117
normalControlState	m3100Attribute 72
numberOfCircuits	m3100Attribute 62
numberOfPorts	m3100Attribute 78
physicalPortAttribute	m3100Attribute 173
physicalPortId	m3100Attribute 168
physicalPortSignalRateAndMappingList	m3100Attribute 169
pointDirectionality	m3100Attribute 118
portAssociations	m3100Attribute 79
portSignalRateAndMappingList	m3100Attribute 80
potentialCapacity	m3100Attribute 161
potentialLinkCapacity	m3100Attribute 119
potentialLinkEndCapacity	m3100Attribute 120
Protected	m3100Attribute 32
protectionStatusR1	m3100Attribute 144
provisionedLinkCapacity	m3100Attribute 121
provisionedLinkConnectionCount	m3100Attribute 122
provisionedLinkEndCapacity	m3100Attribute 123
provisionedNetworkCTPCount	m3100Attribute 124
qualityOfConnectivityService	m3100Attribute 125
Ranges	m3100Attribute 162
Reach	m3100Attribute 171
Redline	m3100Attribute 33
relatedRoutingProfile	m3100Attribute 126

Attribute	Object identifier
reliableResourcePointerR1	m3100Attribute 146
Replaceable	m3100Attribute 34
serialNumber	m3100Attribute 69
serverConnectionList	m3100Attribute 35
ServerTrail	m3100Attribute 127
serverTrailList	m3100Attribute 36
serverTTPPointer	m3100Attribute 128
serviceAffected	m3100Attribute 75
SignalId	m3100Attribute 129
signallingCapabilities	m3100Attribute 64
SignalType	m3100Attribute 37
SoftwareId	m3100Attribute 38
subordinateCircuitPackSoftwareLoad	m3100Attribute 60
sub-partitionPointer	m3100Attribute 130
subNetworkConnectionId	m3100Attribute 131
subNetworkConnectionPointer	m3100Attribute 132
subNetworkId	m3100Attribute 133
superPartitionPointer	m3100Attribute 134
supportableClientList	m3100Attribute 39
supportedByObjectList	m3100Attribute 40
supportedTTPList	m3100Attribute 172
systemTimingSource	m3100Attribute 41
TTPId	m3100Attribute 48
toTermination	m3100Attribute 43
topologicalEndDirectionality	m3100Attribute 135
topologicalGroupPointer	m3100Attribute 136
topologicalPointId	m3100Attribute 137
totalLinkCapacity	m3100Attribute 138
totalLinkEndCapacity	m3100Attribute 139
totalTpCount	m3100Attribute 42
TpPoolId	m3100Attribute 44
TpsInGtpList	m3100Attribute 45
tpsInTpPoolList	m3100Attribute 46
trafficDescriptor	m3100Attribute 140
TrailId	m3100Attribute 47
transmissionCharacteristics	m3100Attribute 67
TtpPortID	m3100Attribute 163
TypeText	m3100Attribute 70
unreliableResourcePointerR1	m3100Attribute 147
upstreamConnectivityPointer	m3100Attribute 49
UsageCost	m3100Attribute 141

Attribute	Object identifier
UserLabel	m3100Attribute 50
validControlType	m3100Attribute 73
vendorName	m3100Attribute 51
Version	m3100Attribute 52
z-TPInstance	m3100Attribute 55
Zend	m3100Attribute 142
zEndNetworkTPList	m3100Attribute 143

A.4 Notifications

Notification	Object identifier
Protectionalarm	m3100Notification 1

A.5 Actions

Action	Object identifier
addCapacityToTopologicalLink	m3100Action 12
addCapacityToTopologicalLinkEnd	m3100Action 13
addTpsToGTP	m3100Action 1
addTpsToTpPool	m3100Action 2
allowAudibleVisualLocalAlarm	m3100Action 3
ArcControl	m3100Action 20
arcRetrieveAlarmDetail	m3100Action 21
assignLinkConnectionOnLogicalLink	m3100Action 14
assignNetworkCTPOnLogicalLinkEnd	m3100Action 15
bridgeRoll	m3100Action 22
circuitPackReset	m3100Action 11
connect	m3100Action 4
convertMulticastToPtoP	m3100Action 23
convertPtoPToMulticast	m3100Action 24
deassignLinkConnectionFromLogicalLink	m3100Action 16
deassignNetworkCTPFromLogicalLinkEnd	m3100Action 17
disconnect	m3100Action 5
externalControl	m3100Action 10
inhibitAudibleVisualLocalAlarm	m3100Action 6
joinXC	m3100Action 25
removeCapacityFromTopologicalLink	m3100Action 18
removeCapacityFromTopologicalLinkEnd	m3100Action 19
removeTpsFromGTP	m3100Action 7
removeTpsFromTpPool	m3100Action 8
splitXC	m3100Action 26
switchOver	m3100Action 9

A.6 Parameters

Parameter	Object identifier
affectedObjectListParameter	m3100Parameter 66
alarmEffectOnServiceParameter	m3100Parameter 1
alarmingResumedParameter	m3100Parameter 67
boundSubnetwork	m3100Parameter 6
channelsAlreadyProvisioned	m3100Parameter 7
circuitPackResetError	m3100Parameter 4
createErrorParameter	m3100Parameter 2
failureToAddLinkConnections	m3100Parameter 8
failureToAddNetworkCTPs	m3100Parameter 9
failureToAssociateLCs	m3100Parameter 10
failureToAssociateNetworkTTP	m3100Parameter 11
failureToBindLink	m3100Parameter 17
failureToBindLinkEnd	m3100Parameter 18
failureToBindTopologicalLink	m3100Parameter 19
failureToCreateAccessGroup	m3100Parameter 20
failureToCreateLCs	m3100Parameter 22
failureToCreateLink	m3100Parameter 21
failureToCreateLinkEnd	m3100Parameter 23
failureToCreateNetworkTTP	m3100Parameter 24
failureToCreateSubnetwork	m3100Parameter 25
failureToDeassignLinkConnection	m3100Parameter 12
failureToDeassignNetworkCTP	m3100Parameter 13
failureToDecreaseCapacity	m3100Parameter 14
failureToDisassociateNetworkTTP	m3100Parameter 26
failureToIncreaseCapacity	m3100Parameter 15
failureToRemoveAccessGroup	m3100Parameter 27
failureToRemoveLC	m3100Parameter 16
failureToRemoveNetworkCTPs	m3100Parameter 28
failureToRemoveNetworkTTP	m3100Parameter 29
failureToRemoveSubnetwork	m3100Parameter 30
failureToSetDirectionality	m3100Parameter 31
failureToSetLinkConnectionCallerId	m3100Parameter 32
failureToSetNetworkCTPCallerId	m3100Parameter 33
failureToSetUserIdentifier	m3100Parameter 34
failureToSupportLCs	m3100Parameter 35
generalErrorParameter	m3100Parameter 3
inconsistentDirectionality	m3100Parameter 36
inconsistentSignalIdentification	m3100Parameter 37
insufficientCapacity	m3100Parameter 38

Parameter	Object identifier
invalidChannelsNumber	m3100Parameter 39
invalidLinkConnection	m3100Parameter 40
invalidNetworkCTP	m3100Parameter 41
invalidServiceCharacteristicsRequested	m3100Parameter 42
invalidTPType	m3100Parameter 43
invalidTrafficDescriptorRequested	m3100Parameter 44
linkAndLinkConnectionNotCompatible	m3100Parameter 47
linkConnectionAlreadyAssigned	m3100Parameter 45
linkEndAndNetworkCTPNotCompatible	m3100Parameter 46
networkCTPAlreadyAssigned	m3100Parameter 48
networkTTPAndAccessGroupNotCompatible	m3100Parameter 49
networkTTPAndSubnetworkNotCompatible	m3100Parameter 50
networkTTPAssociatedWithAccessGroup	m3100Parameter 51
networkTTPAssociatedWithSubnetwork	m3100Parameter 52
networkTTPTerminatesTrail	m3100Parameter 54
networkTTPsExisting	m3100Parameter 53
newServiceCharacteristicsExistsAlready	m3100Parameter 55
newTrafficDescriptorExistsAlready	m3100Parameter 56
noLinkCapacity	m3100Parameter 57
noLinkEndCapacity	m3100Parameter 58
NoSuchLink	m3100Parameter 59
noSuchLinkEnd	m3100Parameter 60
notAssignedToCaller	m3100Parameter 61
notEnoughLinkConnections	m3100Parameter 62
notEnoughNetworkCTPs	m3100Parameter 63
notSupportedProbableCause	m3100Parameter 68
protectionStatusParameterR1	m3100Parameter 65
serviceAffectedErrorParameter	m3100Parameter 5
subnetworkInUse	m3100Parameter 64

A.7 Name bindings

Name binding	Object identifier
AccessGroup-layerNetworkDomain	m3100NameBinding 63
alarmSeverityAssignment-managedElement	m3100NameBinding 1
applicationProcess-managedElement	m3100NameBinding 54
arcIntervalProfile-managedElement	m3100NameBinding 90
arcIntervalProfile-managedElementComplex	m3100NameBinding 91
arcIntervalProfile-network	m3100NameBinding 92
attributeRanges-managedElement	m3100NameBinding 93
circuitPack-equipmentHolder-autoCreated	m3100NameBinding 32

Name binding	Object identifier
circuitPack-equipmentHolder-explicitlyCreated	m3100NameBinding 33
circuitPack-equipmentHolder-autoCreated-R1	m3100NameBinding 37
circuitPack-equipmentHolder-explicitlyCreated-R1	m3100NameBinding 46
CircuitPackR1-circuitPackR1-autoCreated	m3100NameBinding 89
CircuitPackR1-equipmentHolder-autoCreated-Delete	m3100NameBinding 59
CircuitPackR1-equipmentHolder-explicitlyCreated-Delete	m3100NameBinding 60
CircuitPackR1-equipmentHolder-autoCreated	m3100NameBinding 61
CircuitPackR1-equipmentHolder-explicitlyCreated	m3100NameBinding 62
ConnectionR1-network	m3100NameBinding 25
connectionTerminationPointSink-trailTerminationPointSink	m3100NameBinding 5
connectionTerminationPointSink-trailTerminationPointBidirectional	m3100NameBinding 6
connectionTerminationPointSource-trailTerminationPointSource	m3100NameBinding 3
connectionTerminationPointSource-trailTerminationPointBidirectional	m3100NameBinding 4
crossConnection-fabric	m3100NameBinding 7
crossConnection-fabric-R1	m3100NameBinding 39
crossConnection-mpCrossConnection	m3100NameBinding 8
crossConnection-mpCrossConnection-R1	m3100NameBinding 40
equipment-managedElement	m3100NameBinding 9
equipment-equipment	m3100NameBinding 10
equipment-managedElement-R1	m3100NameBinding 41
equipment-equipment-R1	m3100NameBinding 42
equipmentHolder-equipmentHolder	m3100NameBinding 31
eventForwardingDiscriminator-managedElement	m3100NameBinding 11
eventForwardingDiscriminator-managedElement-R1	m3100NameBinding 43
externalPoint-equipment	m3100NameBinding 56
externalPoint-managedElement	m3100NameBinding 57
externalPoint-managedElementComplex	m3100NameBinding 58
fabric-managedElement	m3100NameBinding 12
fabric-managedElement-R1	m3100NameBinding 44
genericTransportTTP-managedElement	m3100NameBinding 94
gtp-fabric	m3100NameBinding 13
layerNetworkDomain-networkR1	m3100NameBinding 64
linkConnection-layerNetworkDomain	m3100NameBinding 66
linkConnection-topologicalLink	m3100NameBinding 67
log-managedElement	m3100NameBinding 14
logicalLink-layerNetworkDomain	m3100NameBinding 65
logicalLinkEnd-layerNetworkDomain	m3100NameBinding 68
logicalLinkEnd-subNetwork	m3100NameBinding 69
managedElement-network	m3100NameBinding 15

Name binding	Object identifier
managedElement-organization	m3100NameBinding 27
managedElement-organizationalUnit	m3100NameBinding 28
managedElement-managedElementComplex	m3100NameBinding 34
managedElement-managedElementComplex-explicitlyCreated	m3100NameBinding 45
managedElementComplex-organization	m3100NameBinding 35
managedElementComplex-organizationalUnit	m3100NameBinding 36
managedElementComplex-network	m3100NameBinding 53
mpCrossConnection-fabric	m3100NameBinding 16
network-network	m3100NameBinding 17
network-organization	m3100NameBinding 29
network-organizationalUnit	m3100NameBinding 30
networkCTPSink-subNetwork	m3100NameBinding 72
networkCTPSink-layerNetworkDomain	m3100NameBinding 73
networkCTPSource-subNetwork	m3100NameBinding 74
networkCTPSource-layerNetworkDomain	m3100NameBinding 75
networkTTPSink-layerNetworkDomain	m3100NameBinding 76
networkTTPSink-subNetwork	m3100NameBinding 77
networkTTPSource-layerNetworkDomain	m3100NameBinding 79
networkTTPSource-subNetwork	m3100NameBinding 80
PhysicalPort-equipment	m3100NameBinding 95
PhysicalPort-managedElement	m3100NameBinding 96
scheduler-managedElement	m3100NameBinding 51
simpleScanner-managedElement	m3100NameBinding 49
software-equipment	m3100NameBinding 18
software-software	m3100NameBinding 19
software-managedElement	m3100NameBinding 20
SubNetwork-layerNetworkDomain	m3100NameBinding 81
subNetworkConnection-subNetwork	m3100NameBinding 82
subsystem-managedElement	m3100NameBinding 55
testActionPerformer-managedElement	m3100NameBinding 47
testObject-testActionPerfomer	m3100NameBinding 48
topologicalLink-layerNetworkDomain	m3100NameBinding 83
topologicalLinkEnd-layerNetworkDomain	m3100NameBinding 70
topologicalLinkEnd-subNetwork	m3100NameBinding 71
tpPool-fabric	m3100NameBinding 21
trailR1-network	m3100NameBinding 26
trailR2-layerNetworkDomain	m3100NameBinding 84
trailTerminationPointSink-managedElement	m3100NameBinding 24
trailTerminationPointSource-managedElement	m3100NameBinding 23
usageMeteringControl-managedElement	m3100NameBinding 50

Annex B

Network level model methodology

The network level managed object specifications were developed using a methodology for development of a GDMO Engineering Viewpoint. The GDMO definitions of these managed objects make reference to the communities from which the definitions were defined. These references are indicated in the behaviour clauses of the GDMO specifications by tags enclosed in angled brackets ('<' and '>').

In general, in GDMO, a single RDN (specified by the naming attribute of the managed object class and defined in its NAME BINDING) is used to uniquely identify an object instance relative to its parent. In some cases, this method of naming object instances is different from the definitions of the communities on which these managed objects are based, where multiple identifiers have been used. In such cases, the use of a single unique naming attribute is an optimization for the GDMO engineering viewpoint.

Annex C

Telemetry fragment

The telemetry fragment models external points (relays and contact closures) which are used to control external devices (generators, heaters, etc.) or monitor external conditions.

Table C.1/M.3100

Control point valid action type (Optional)	State before	Control action type	Action result	State after
momentary only	closed	close-continuously	error: invalid action type	closed
		open-continuously	error: invalid action type	closed
		close-momentarily	error: already in condition	closed
		open-momentarily	completed	open then closed
	open	close-continuously	error: invalid action type	open
		open-continuously	error: invalid action type	open
		close-momentarily	completed	closed then open
		open-momentarily	error: already in condition	open
continuous only	closed	close-continuously	error: already in condition	closed
		open-continuously	completed	open
		close-momentarily	error: invalid action type	closed
		open-momentarily	error: invalid action type	closed
	open	close-continuously	completed	closed
		open-continuously	error: already in condition	open
		close-momentarily	error: invalid action type	open
		open-momentarily	error: invalid action type	open
momentary and continuous	closed	close-continuously	error: already in condition	closed
		open-continuously	completed	open
		close-momentarily	error: already in condition	closed
		open-momentarily	completed	open then closed
	open	close-continuously	completed	closed
		open-continuously	error: already in condition	open
		close-momentarily	completed	closed then open
		open-momentarily	error: already in condition	open

Annex D

Circuit pack fragment

The Circuit pack fragment models external points (relays and contact closures) which are used to control external devices (generators, heaters, etc.) or monitor external conditions.

The model supports the following circuit pack functionality:

- request re-initialization of a circuit pack;
- for a circuit pack that supports multiple physical ports, indicate the associated entity of the ports;
- indicate the available signal rates of a circuit pack;
- indicate and configure the signal rate and payload mapping for the port(s) of a circuit pack.

The circuitPackR1 object is subclassed from equipmentR2 instead of circuitPack, in order to use the attribute values of the availabilityStatus besides "notInstall", including "degrade" for indicating that only a subset of the ports is not functioning.

The textType attribute inherited from equipmentR2 is used to indicate the type of the circuit pack (the syntax of textType is GraphicString, and the syntax of the circuitPackType attribute is printableString).

The comment field of the ASN.1 data type SignalRate is an OID which reflects the rate and format.

Annex E

Generic protection fragment

The generic protection fragment describes an information model for generic protection switching (PS) of resources such as circuit pack. The object classes defined in this fragment are useful to describe information exchanged across interfaces defined in M.3010 Telecommunication Management Network (TMN) architecture. Two protection switching object classes are, namely protectionGroupR2, which is a subclass of the G.774.3 protectionGroupR1 object class, and protectionUnitR1, which is a subclass of the X.721 top object class.

E.1 Protection Group R2

The protectionGroupR2 managed object class is used to represent the various manageable aspects of a protection system within a Network Element (NE). Notifications of protection switch events and management system control of lockouts, forced switches, and manual switches are the primary management functions supported by this entity. This object class is a subclass of the protectionGroupR1 object class defined in ITU-T Rec. G.774.3 (2001).

Instances of this object class may be automatically created in an agent, e.g., immediately following the initialization of the NE resources involved in the protection system, according to the make-up and mode of the NE. Instances of this object class may be automatically deleted in the agent.[1]

[1] Instances of protectionGroupR2 may also be created or deleted as the result of management operations to the protectionCoordinator object (defined in ITU-T Rec. G.774.9) of an NE, such as the establishProtection and dismissProtection actions.

Multiple instances of the protectionGroupR2 object may exist in an NE (one for each protection system supported by the NE). An instance of the protectionGroupR2 object would contain two or more instances of the protectionUnitR1 object.

This object class inherits the following attributes from its superclass protectionGroupR1:

Protection Group ID: This read-only attribute provides a unique name for the Protection Group instance in the NE.

Operational State: This read-only attribute identifies whether or not the protection mechanism represented by this instance is capable of performing its normal functions.

Protection Group Type: This read-write attribute identifies whether the protection scheme used is 1+1 or M:N.

Revertive: This read-write attribute identifies whether or not the protection scheme used is revertive. The default value for this attribute shall indicate revertive operation, but this attribute should be able to be set to indicate non-revertive operation by a command from the manager.

Wait To Restore Time: This read-write attribute identifies the amount of time, in seconds, that the protection system should wait after a fault clears before switching back to the protected resource. This attribute is only relevant for revertive system operation.

This object class inherits the invokeProtection and releaseProtection actions from its superclass protectionGroupR1.

Invoke Protection: This action is used to request a lockout, a forced switch, or a manual switch on one or more of the resources involved in the protection system. The following input parameters are included in the Invoke Protection action:

– Switch Type (Manual, Forced or Lockout).

– Protection Entity (Optional): ID(s) of the protected and/or protecting Protection Unit entity to which the request applies. If not present, the request is meant to apply to all such entities in the Protection Group.

Release Protection: This action is used to release a lockout, a forced switch, or a manual switch on one or more of the resources involved in the protection system. The following input parameters are included in the Release Protection action:

– Switch Type (Manual, Forced or Lockout).

– Protection Entity (Optional): ID(s) of the protected and/or protecting Protection Unit entity to which the request applies. If not present, the request is meant to apply to all such entities in the Protection Group.

This object class also inherits the protectionSwitchReporting, stateChange, objectCreation, objectDeletion, and attributeValueChange notifications from protectionGroupR1.

Protection Switch Reporting: This notification is emitted from the Protection Group entity to report any protection switch events. The following parameters are included:

– The ID of the Protection Group entity reporting the notification.

– Time and date the protection switch event was detected.

– The ID of the Protection Unit (within the protection group) involved in the switch event.

– Protection Status (old and new) according to the following rules:

  - If the switch event entails a switch from protected resource to protecting resource (or vice versa) and has been done without preempting an existing switch, the old and new Protection Status parameters in the notification shall match the old and new values of the Protection Status attribute of the protecting Protection Unit.

- If the switch is performed by preempting an existing one, the old and new Protection Status parameters in the notification shall match the old and new values of the Protection Status attribute of the protecting Protection Unit.
- If an auto-switch condition exists on a resource, but the auto-switch cannot be served due to the unavailability of the resource that otherwise protects it, the old and new Protection Status parameters in the notification shall match the old and new values of the Protection Status attribute of the Protection Unit on which the auto-switch condition arises. The exception is when that resource is already forced or locked out, in which case, no notification is sent.
- If the switch event entails a protected resource being locked out or released from lockout without modifying any existing switch, the old and new Protection Status parameters in the notification shall match the old and new values of the Protection Status attribute of the protected Protection Unit which has been locked out.
- If the switch event entails a protecting resource being locked out or released from lockout without modifying any existing switch, the old and new Protection Status parameters in the notification shall match the old and new values of the Protection Status attribute of the protecting Protection Unit which has been locked out.

State Change: This notification is used to report changes to the Operational State attribute of this entity. The notification identifies the state attribute that changed, its old value, and its new value.

Object Creation: This notification is used to report the creation of an instance of this entity.

Object Deletion: This notification is used to report the deletion of an instance of this entity.

Attribute Value Change: This notification is used to report a change in a value of a given attribute. The notification identifies the attribute that changed, its old value, and its new value. Supported attributes are: Protection Group Type, Revertive, and Wait To Restore Time.

E.2 Protection Unit R1

The protectionUnitR1 managed object class is used to manage the protected (i.e., working, regular, or preferred) or protecting (i.e., backup or standby) resource in a protection system. It relates the resources (e.g., circuit packs) involved in the protection system and keeps track of the protection switching status of the resources.

Instances of this entity may be automatically created in the agent, e.g., immediately following the initialization of the NE resources (e.g., such as circuit pack) involved in the protection system, according to the make-up and mode of the NE. Instances of this entity may be deleted by the agent, e.g., upon the deletion of the corresponding resource objects. The agent may also create and delete instances of this object class in order to reflect local modifications in the protection schemes.[2]

Two or more instances of the protectionUnitR1 object may exist within an instance of the protectionGroupR2 object.

An instance of the protectionUnitR1 object could contain an instance of the protectionCurrentData object (defined in ITU-T Rec. G.774.1).

A protectionUnitR1 instance is related to instances of resource entities (e.g., Circuit Pack) via the Unreliable Resource Pointer attribute. If the function of the resource entities (e.g., timing function, transport termination point function, etc.) is explicitly modelled as object instances in the NE, then a

[2] Instances of this object class may also be created or deleted as the result of management operations to the protectionCoordinator object (defined in ITU-T Rec. G.774.9) of an NE, such as the establishProtection, dismissProtection and modifyProtection actions.

protectionUnitR1 instance is also related to instances of the modelled function entity via the Reliable Resource Pointer attribute.

This object class has the following attributes:

Protection Unit ID: This read-only attribute provides a unique name for the Protection Unit instance within the containing Protection Group object.

Protecting: This read-only attribute identifies whether or not the protection unit is associated with a resource providing a protecting ("true") or protected ("false") role in the protection system.

Unreliable Resource Pointer: This read-only attribute identifies the unreliable resource (e.g., circuit pack entity) associated with the Protection Unit object (e.g., the actual protected or protecting resource). The syntax of this attribute is set-valued and could point to multiple instances of unreliable resources when a set of resources forms an atomic unit in the protection system.

Reliable Resource Pointer: This read-only attribute identifies the reliable resource (i.e., the functional entity), if there is any, associated with the Protection Unit. The value of this attribute of a protection unit (PU) will change when the PU is involved in a protection switch or release. For a protected PU, when it is not switched, this attribute is pointing to the associated reliable resource (i.e., the functional object) and when it is switched, this attribute points to NULL. For a protecting PU, when it is not switched, this attribute is pointing to NULL, and when it is switched, this attribute is pointing to the associated reliable resource (i.e., the functional object). The syntax of this attribute is set-valued and could point to multiple instances of reliable resource when a set of functional objects form an atomic unit in the protection system. Examples of usage of this attribute are provided in E.3.

Priority: This read-write attribute specifies the priority of the service carried on the resource associated with the protection Unit instance. Valid values for this attribute are integers, where the value 1 indicates the highest priority, and a larger value indicates a lower priority.

This object class is defined with a status attribute.

Protection Status R1: This read-only attribute indicates the status of the protection switch in a Protection Unit object. The following behaviour applies to this attribute:

– This attribute must be capable of indicating pending as well as active switching requests relative to the protection unit. However, only one of the values lockout, forced switch, or manual switch can be present at the same time.

– A protection system may support only a subset of the allowable values of this attribute. The subset of values to be supported by a system is implementation-specific.

– The syntax of this attribute includes a subfield "relatedUnit" which is of ASN.1 CHOICE of "fromProtectionUnitNumber" and "toProtectionUnitNumber". This subfield is used to indicate on which unit the service is carried.

  - For a protected PU, both the fromProtectionUnitNumber (fromPU#) and the toProtectionUnitNumber (toPU#) hold the ID of the related protecting PU. When switching to the protecting PU (i.e., service will be carried by the protecting PU), the toProtectionUnitNumber choice is used. When switching back to the protected PU (service will be carried by the protected PU), the fromProtectionUnitNumber choice is used.

  - For a protecting PU, both the fromProtectionUnitNumber (fromPU#) and the toProtectionUnitNumber (toPU#) hold the ID of the related protected PU. When switching to the protected PU (i.e., service will be carried by the protected PU), the toProtectionUnitNumber choice is used. When switching to the protecting PU (service will be carried by the protecting PU), the fromProtectionUnitNumber choice is used.

– If a system can support protection switching due to Resource Degrade (RD) besides Resource Fail (RF), protection switching of RD is similar to that in the subsequent description for RF.

– The following allowable Protection Status values are associated with each **protected** Protection Unit (PU).

  - **No Request**: No switch request is present on the unit. *In this case, service is on the protected PU, status syntax is noRequest. For non-revertive system, the status syntax of the related protecting PU is also noRequest.*
  - **Manual Switch to Protecting Unit Complete**: The unit has completed a Manual Switch. *In this case, service is on the related protectingPU, status syntax of the protected PU is manualSwitch (switchStatus: completed; relatedUnit: toPU#). Status syntax of the related protecting PU is manualSwitch (switchStatus: completed; relatedUnit: fromPU#).*
  - **Release Failed**: A time-out occurs while waiting for a release. *In this case, service is still on the protectingPU, status syntax is releaseFailed plus the previous status, such as manualSwitch (switchStatus: completed; relatedUnit: toPU#). Status syntax of the related protecting PU is still the previous status, such as manualSwitch (switchStatus: completed; relatedUnit: fromPU#).*
  - **Automatic Switch (RF) Pending:** The unit has a Fail condition present and the protecting unit is unavailable. *In this case, service is still on the protectedPU, status syntax is autoSwitch (switchStatus: pending; relatedUnit: toPU#; reason: RF). Status syntax of the related protecting PU is autoSwitch (switchStatus: pending; relatedUnit: fromPU#; reason: RF) plus its previous status.*
  - **Automatic Switch (RF) Complete**: The unit has completed an Automatic Switch to the protecting unit due to an Equipment Fail condition. *In this case, service is on the related protectingPU, status syntax of the protected PU is autoSwitch (switchStatus: completed; relatedUnit: toPU#; reason: RF). Status syntax of the related protecting PU is autoSwitch (switchStatus: completed; relatedUnit: fromPU#; reason: RF).*
  - **Automatic Switch (RF) Present, Operate failed**: An automatic switch (RF) request is in progress and a time-out occurs while waiting for completion. *In this case, service is still on the protectedPU, status syntax is autoSwitch (switchStatus: failed; relatedUnit: toPU#; reason: RF). Status syntax of the related protecting PU is autoSwitch (switchStatus: pending; relatedUnit: fromPU#; reason: RF) plus its previous status.*
  - **Force Switch Complete, Automatic Switch (RF) Pending**: The unit has completed a Force Switch. Additionally, the unit has an automatic switch (RF) pending. *In this case, service is on the related protectingPU, status syntax of the protected PU is forceSwitch (switchStatus: completed; relatedUnit: toPU#) plus autoSwitch (switchStatus: pending; relatedUnit: toPU#; reason: RF). Status syntax of the related protecting PU is forceSwitch (switchStatus: completed; relatedUnit: fromPU#) plus autoSwitch (switchStatus: pending; relatedUnit: fromPU#; reason: RF).*
  - **Automatic Switch Complete, Wait-To-Restore (revertive only)**: The unit has completed an Automatic Switch to the protecting unit. *In this case, service is on the related protectingPU, status syntax of the protected PU is autoSwitch (switchStatus: completed; relatedUnit: toPU#; reason: WTR). Status syntax of the related protecting PU is autoSwitch (switchStatus: completed; relatedUnit: toPU#; reason: WTR).*
  - **Force Switch Complete**: The unit has completed a Force Switch to the protecting unit. *In this case, service is on the related protectingPU, status syntax of the protected PU is forceSwitch (switchStatus: completed; relatedUnit: toPU#). Status syntax of the related protecting PU is forceSwitch (switchStatus: completed; relatedUnit: fromPU#).*

  - **Protected Unit Lockout Completed**: The unit has been locked out from the protecting unit. *In this case, service is on the protectedPU, status syntax is lockout (switchStatus: completed).*
  - **Protected Unit Lockout, Operate failed**: The unit has been locked out from the protecting unit, and, the previously completed switch could not be released within the expected time-out. When the switch is released, the Operate failed status is removed. *In this case, service is still on the related protectingPU, status syntax of the protected PU is lockout (switchStatus: completed) plus releaseFailed. Status syntax of the related protecting PU is still the previous status, such as manualSwitch (switchStatus: completed; relatedUnit: fromPU#).*
  - **Locked In**: The unit is in the locked-in condition. This is caused by excessive protection switching events. *In this case, service is on the protectedPU, status syntax is locked-in.*

– A **non-revertive protected** Protection Unit has the following additional status values:

  - **Do Not Revert**: The protected unit has been switched to the protecting unit and the request to do so has been released. The switch to the protecting unit is maintained. *In this case, service is on the related protectingPU, status syntax of the protected PU is doNotRevert. Status syntax of the related protecting PU is doNotRevert.*
  - **Manual Switch to Protected Unit Complete**: The unit has completed a Manual Switch from the protecting unit to the protected unit. *In this case, service is on the protectedPU, status syntax is manualSwitch (switchStatus: completed; relatedUnit: fromPU#). Status syntax of the related protecting PU is manualSwitch (switchStatus: completed; relatedUnit: toPU#).*
  - **Force Switch to Protected Unit Complete**: The unit has completed a Force Switch from the protecting unit to the protected unit. *In this case, service is on the protectedPU, status syntax is forceSwitch (switchStatus: completed; relatedUnit: fromPU#). Status syntax of the related protecting PU is forceSwitch (switchStatus: completed; relatedUnit: toPU#).*
  - **Automatic Switch (RF) to Protected Unit Complete**: The protecting unit has an Equipment Fail condition present and the protected unit is now being utilized. *In this case, service is on the protectedPU, status syntax is autoSwitch (switchStatus: completed; relatedUnit: fromPU#; reason: RF). Status syntax of the related protecting PU is autoSwitch (switchStatus: completed; relatedUnit: toPU#; reason: RF).*
  - **Force Switch from Protecting Unit Complete, Automatic Switch (RF) Pending**: The unit has completed a Force Switch from the protecting unit to the protected unit. Additionally, the protected unit has an automatic switch (RF) condition present. *In this case, service is on the protectedPU, status syntax is forceSwitch (switchStatus: completed; relatedUnit: fromPU#) plus autoSwitch (switchStatus: pending; relatedUnit: toPU#; reason: RF). Status syntax of the related protecting PU is forceSwitch (switchStatus: completed; relatedUnit: toPU#) plus autoSwitch (switchStatus: pending; relatedUnit: fromPU#; reason: RF).*

– The following allowable Protection Status values are associated with each **protecting** Protection Unit:

  - **No Request**: No switch request is present on the protecting unit. *In this case, service is not on the protecting PU, status syntax is noRequest. For non-revertive system, the status syntax of the related protected PU is noRequest.*

- **Manual Switch to Protecting Unit Complete**: A protected unit has completed a Manual Switch. *In this case, service is on the protectingPU, status syntax is manualSwitch (switchStatus: completed; relatedUnit: fromPU#). Status syntax of the related protected PU is manualSwitch (switchStatus: completed; relatedUnit: toPU#).*
- **Automatic Switch (RF) Pending**: A protected unit has an Equipment Fail condition present and the protecting unit is unavailable for this request. *In this case, service is still on the protectedPU. Status syntax of the protecting PU is autoSwitch (switchStatus: pending; relatedUnit: fromPU#; reason: RF) plus its previous status, which causes its unavailability. Status syntax of the related protected PU is autoSwitch (switchStatus: pending; relatedUnit: toPU#; reason: RF).*
- **Automatic Switch Complete (RF) to Protecting Unit**: A protected unit has completed an automatic switch (RF) to the protecting unit. *In this case, service is on the protectingPU, status syntax is autoSwitch (switchStatus: completed; relatedUnit: fromPU#; reason: RF). Status syntax of the related protected PU is autoSwitch (switchStatus: completed; relatedUnit: toPU#; reason: RF).*
- **Automatic Switch (RF) to Protecting Complete, Wait-To-Restore (revertive only)**: The unit has completed an Automatic Switch from the protected unit. *In this case, service is on the protectingPU, status syntax is autoSwitch (switchStatus: completed; relatedUnit: fromPU#; reason: WTR). Status syntax of the related protected PU is autoSwitch (switchStatus: completed; relatedUnit: fromPU#; reason: WTR).*
- **Protecting Unit RF Present**: The protecting unit has an Equipment Fail condition present. *Status syntax of the protecting PU is resourceFailed.*
- **Force Switch Complete to Protecting Unit**: The unit has completed a Force Switch from a protected unit to the protecting unit. *In this case, service is on the protectingPU, status syntax is forceSwitch (switchStatus: completed; relatedUnit: fromPU#). Status syntax of the related protected PU is forceSwitch (switchStatus: completed; relatedUnit: toPU#).*
- **Protecting Unit Locked Out**: The protecting unit has been locked out. *In this case, service is not on the protectingPU, status syntax is lockout (switchStatus: completed).*
- **Protecting Unit Release Lock Out Failed**: A release of a lockout is in progress and a time-out occurs waiting for the lockout condition to clear. *In this case, service is not on the protectingPU, status syntax is lockout (releaseFailed).*

– A **non-revertive protecting** Protection Unit has the following additional status values:

- **Do Not Revert**: A protected unit has been switched to the protecting unit and the request to do so has been released. The switch to the protecting unit is maintained. *In this case, service is on the protectingPU, status syntax is doNotRevert. Status syntax of the related protected PU is doNotRevert.*
- **Manual Switch to Protected Unit Complete**: The unit has completed a Manual Switch from the protecting unit to the protected unit. *In this case, service is on the protectedPU. Status syntax of the protecting PU is manualSwitch (switchStatus: completed; relatedUnit: toPU#). Status syntax of the related protected PU is manualSwitch (switchStatus: completed; relatedUnit: fromPU#).*
- **Force Switch to Protected Unit Complete**: The protecting unit has completed a forced switch to the protected unit. *In this case, service is on the protectedPU. Status syntax of the protecting PU is forceSwitch (switchStatus: completed; relatedUnit: toPU#). Status syntax of the related protected PU is forceSwitch (switchStatus: completed; relatedUnit: fromPU#).*

- **Force Switch to Protected Unit Complete, Protecting Unit Equipment Failed**: The protecting unit has completed a forced switch to the protected unit. Additionally, there is an Equipment Fail condition on the protecting unit. *In this case, service is on the protectedPU. Status syntax of the protecting PU is forceSwitch (switchStatus: completed; relatedUnit: toPU#) plus equipmentFailed. Status syntax of the related protected PU is forceSwitch (switchStatus: completed; relatedUnit: fromPU#).*
- **Automatic Switch (RF) to Protected Unit Complete**: The protecting unit has an Equipment Fail condition present and the protected unit is now being utilized. *In this case, service is on the protectedPU. Status syntax of the protecting PU is autoSwitch (switchStatus: completed; relatedUnit: toPU#; reason: RF). Status syntax of the related protected PU is autoSwitch (switchStatus: completed; relatedUnit: fromPU#; reason: RF).*

The following table provides a mapping of the protection status of a protection unit to the syntax of the attribute. In the table, the following abbreviate notations are used:

- AS = Auto Switch.
- MS = Manual Switch.
- FS = Forced Switch.
- RF = Resource Failed.
- WTR = Wait To Restore.
- SwitchStatus: completed, pending, failed.
- FromAndToPU: toPU#, fromPU#.
- AutoSwitchReason: waitToRestore, resourceDegrade, resourceFail.

	Scenario	**Value of the Protection Status attribute**
Cases for the **Protected** Protection Unit	No Request	noRequest( )
	MS to protecting Complete	manualSwitch(completed,toProtectionUnitNumber)
	Release Failed	releaseFailed( ) and previous status
	AS (RF) Pending	autoSwitch(pending,toProtectionUnitNumber, resourceFail)
	AS (RF) to protecting Complete	autoSwitch(completed,toProtectionUnitNumber, resourceFail)
	AS (RF) Present, Operate failed	autoSwitch(failed,toProtectionUnitNumber, resourceFail)
	FS Complete, AS (RF) Pending	forcedSwitch(completed,toProtectionUnitNumber) andautoSwitch(pending,toProtectionUnitNumber, resourceFail)
	AS to protecting Complete, WTR (revertive only)	autoSwitch(completed,toProtectionUnitNumber, waitToRestore)
	FS to protecting Complete	forcedSwitch(completed,toProtectionUnitNumber)
	Protected Unit Lockout Complete	lockout(completed)
	Protected Unit Lockout Complete Operate failed	lockout(completed) andreleaseFailed( )
	Locked In	lockedIn ( )

	Scenario	Value of the Protection Status attribute
Additional Cases for the Non-Revertive **Protected** Protection Unit	Do Not Revert	doNotRevert( )
	MS to Protected Unit Complete	manualSwitch(completed,fromProtectionUnitNumber)
	FS to Protected Unit Complete	forcedSwitch(completed,fromProtectionUnitNumber)
	AS (RF) to Protected Unit Complete	autoSwitch(completed,fromProtectionUnitNumber, resourceFail)
	FS from Protecting Unit Complete, AS (RF) Pending	forcedSwitch(completed,fromProtectionUnitNumber) and autoSwitch(pending,toProtectionUnitNumber, resourceFail)
Cases for the **Protecting** Protection Unit	No Request	noRequest( )
	MS to Protecting Unit Complete	manualSwitch(completed,fromProtectionUnitNumber)
	AS (RF) to Protecting Complete	autoSwitch(completed,fromProtectionUnitNumber, resourceFail)
	AS (RF) to Protecting Pending	autoSwitch(pending,fromProtectionUnitNumber, resourceFail)
	AS Complete (RF) to Protecting, WTR (revertive)	autoSwitch(completed,fromProtectionUnitNumber, waitToRestore)
	Protecting Unit RF Present	resourceFailed( )
	FS Complete to Protecting Unit	forcedSwitch(completed,fromProtectionUnitNumber)
	Protecting Unit Locked Out	lockout(completed)
	Protecting Unit Locked Out, Release lockOut Failed	lockout(releaseFailed)
Additional Cases for the Non-Revertive **Protecting** Protection Unit	Do Not Revert	doNotRevert( )
	MS to Protected Unit Complete	manualSwitch(completed,toProtectionUnitNumber)
	FS to Protected Unit Complete	forcedSwitch(completed,toProtectionUnitNumber)
	FS to Protected Unit Complete, Protecting Unit RF	forcedSwitch(completed,toProtectionUnitNumber) andresourceFailed( )
	AS (RF) to Protected Unit Complete	autoSwitch(completed,toProtectionUnitNumber, resourceFail)

NOTE – A protection system may support only a subset of the allowable status values listed in the above table. The subset of values to be supported by a system is implementation-specific.

This object class inherits the following notification from its superclass:

Attribute Value Change: This notification is used to report a change in a value of a given attribute. The notification identifies the attribute that changed, its old value, and its new value. Supported attributes are: Reliable Resource Pointer, Protection Status, and Priority.

E.3 Example protection applications

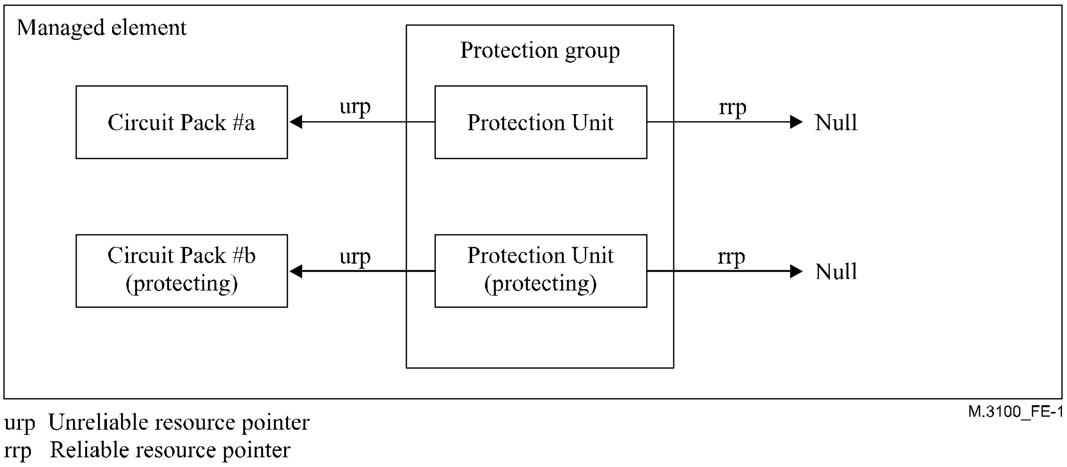
Three protection schemes are illustrated equipment resource (i.e., circuit pack):

- 1+1 Protection without explicitly modelled functions on the circuit packs, e.g., duplicated controllers;
- 1+1 Protection involving explicitly modelled functions on the circuit pack, e.g., Transport packs;
- 1xN Protection, where 1 Protecting pack is available for N normal packs, e.g., in case of a DS3 Port Unit.

It should be noted that the protection function is independent of the protection function of the explicitly modelled functionality. For instance, if the explicitly modelled functionality is termination point and termination point protection switching is supported (e.g., SDH Multiplex-Section Protection), then in addition to the resource protection model for the resources, the transport protection switching model should be used for the transport objects.

E.3.1 1+1 Equipment protection, no explicitly modelled functionality

The 1+1 protection schemes of circuit packs are modelled as shown in Figure E.1.


Managed element
Protection group
Circuit Pack #a
urp
Protection Unit
rrp
Null
Circuit Pack #b (protecting)
urp
Protection Unit (protecting)
rrp
Null
M.3100_FE-1
urp Unreliable resource pointer
rrp Reliable resource pointer

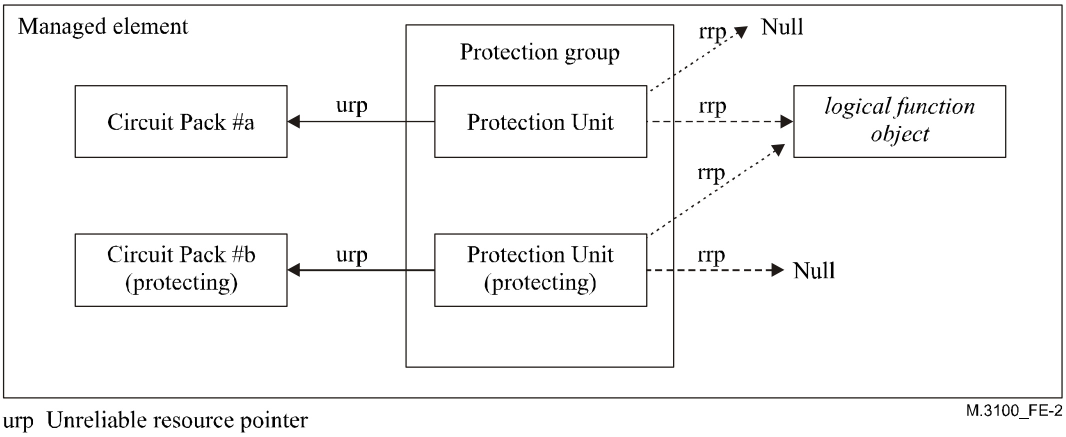
NOTE 1 – Containment relationship for the Circuit Pack is not shown.
NOTE 2 – If the unit of redundancy is not a single circuit pack but a set of circuit packs, the resource pointers will point to all the circuit packs in the set.

Figure E.1/M.3100 – 1+1 Protection, no explicitly modelled functionality

Note that if the unit of redundancy is not a single circuit pack unit but a set of circuit pack, the resource pointers of the Protection Units will point to all the circuit packs in the set. However, the number of Protection Unit is still two.

E.3.2 1+1 Equipment Protection, Explicitly Modelled Functionality

This 1+1 protection schemes apply if circuit packs are associated with explicitly modelled functionality, such as Timing/Synchronization/Termination objects. A similar protection scheme is present as described in E.3.1; however, the Reliable Resource Pointer is now pointing to the functional objects that are being protected. See Figure E.2.



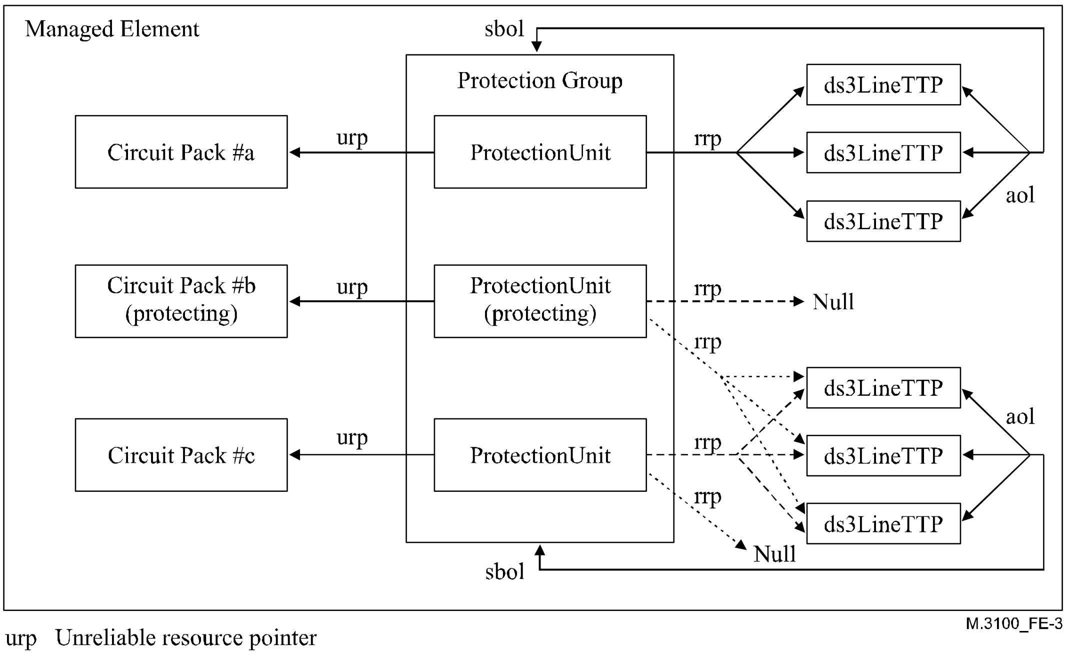
urp Unreliable resource pointer
rrp Reliable resource pointer

NOTE 1 – Containment relationship for the Circuit Pack is not shown.
NOTE 2 – On a protection switch from Circuit Pack #a to Circuit Pack #b, all dashed arrows (----▶) are replaced by the dotted arrows (·······▶).
NOTE 3 – If the unit of redundancy is not a single object but a set of objects, the resource pointers will point to all the objects in the set.

Figure E.2/M.3100 – 1+1 Protection, explicitly modelled functionality

E.3.3 1xN equipment protection, explicitly modelled functionality

The 1xN circuit pack protection schemes of circuit packs that have manageable entities related to them (e.g., DS3 Termination) are modelled as shown in Figure E.3.



urp Unreliable resource pointer
rrp Reliable resource pointer
sbol Supported by object list
aol Affected object list

NOTE 1 – Containment relationship for the Circuit Pack is not shown.
NOTE 2 – On a protection switch from Circuit Pack #c to Circuit Pack #b, all dashed arrows (----▶) are replaced by the dotted arrows (·········▶).

Figure E.3/M.3100 – 1xN protection, explicitly modelled functionality

Annex F

Generic alarm reporting control (ARC) feature

F.1 Business requirements

This clause describes the generic Alarm Reporting Control business requirements.

F.1.1 High-level use cases

The terminology used in the use cases is based on terminology defined in this Recommendation and on terminology defined in ITU-T Rec. M.3400, *TMN management functions*.

The set of use cases provided here is not exhaustive and is left as an exercise to the reader. Only that which was deemed necessary to clarify the need and the feature requirements is included.

This use case has been developed to provide a business context for ARC. (See Figure F.1.)

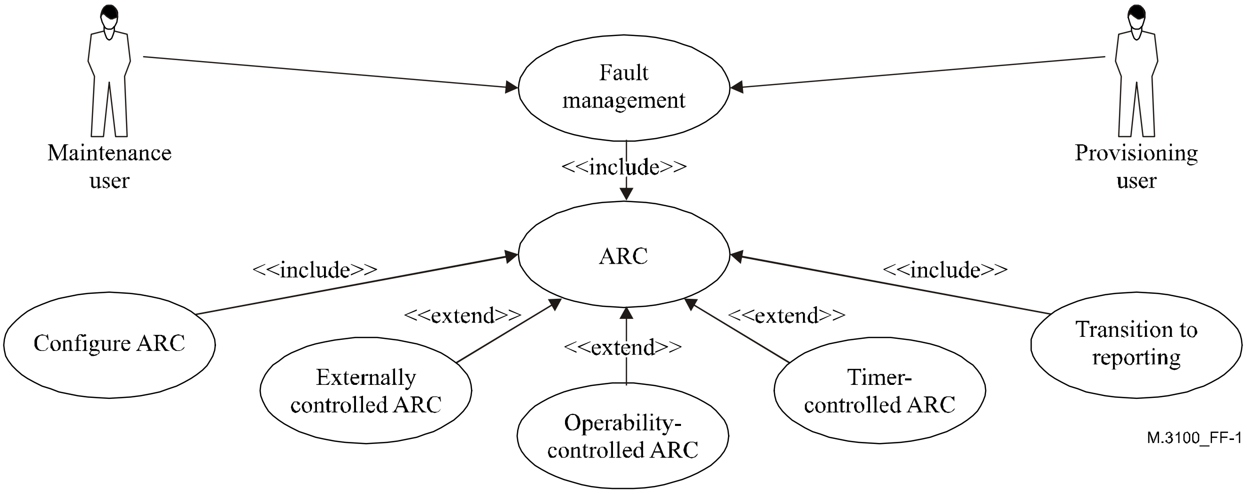


Figure F.1/M.3100 – Feature context use case

F.1.1.1 Fault Management

This use case represents all the functions provided by Fault Management as described in ITU-T Rec. M.3400, *TMN management functions*. ARC functions provide additional capability beyond those already defined for Fault Management in other ITU-T Recommendations.

Maintenance and Provisioning Users are expected to make use of the ARC Fault Management capabilities to enable alarm-free setup/teardown, alarm-free provisioning, or alarm-free repair. These three functions are generalizations of the first business requirement.

F.1.1.2 ARC

In order to provide the needed capabilities, and in order to provide these capabilities within different operational environments, the following use cases have been identified: "Configure ARC" and "Transition to reporting" (core functions needed for all operational environments), and "Externally controlled ARC", "Operability-controlled ARC", or "Timer-controlled ARC" (need dependent upon operational environment and/or function).

F.1.1.3 Configure ARC

This high-level use case represents all of the configuration use cases for ARC.

F.1.1.4 Externally controlled ARC

This use case describes the case of an external managing entity determining and controlling when resource alarm reporting is to be turned on after having been turned off.

F.1.1.5 Operability-controlled ARC

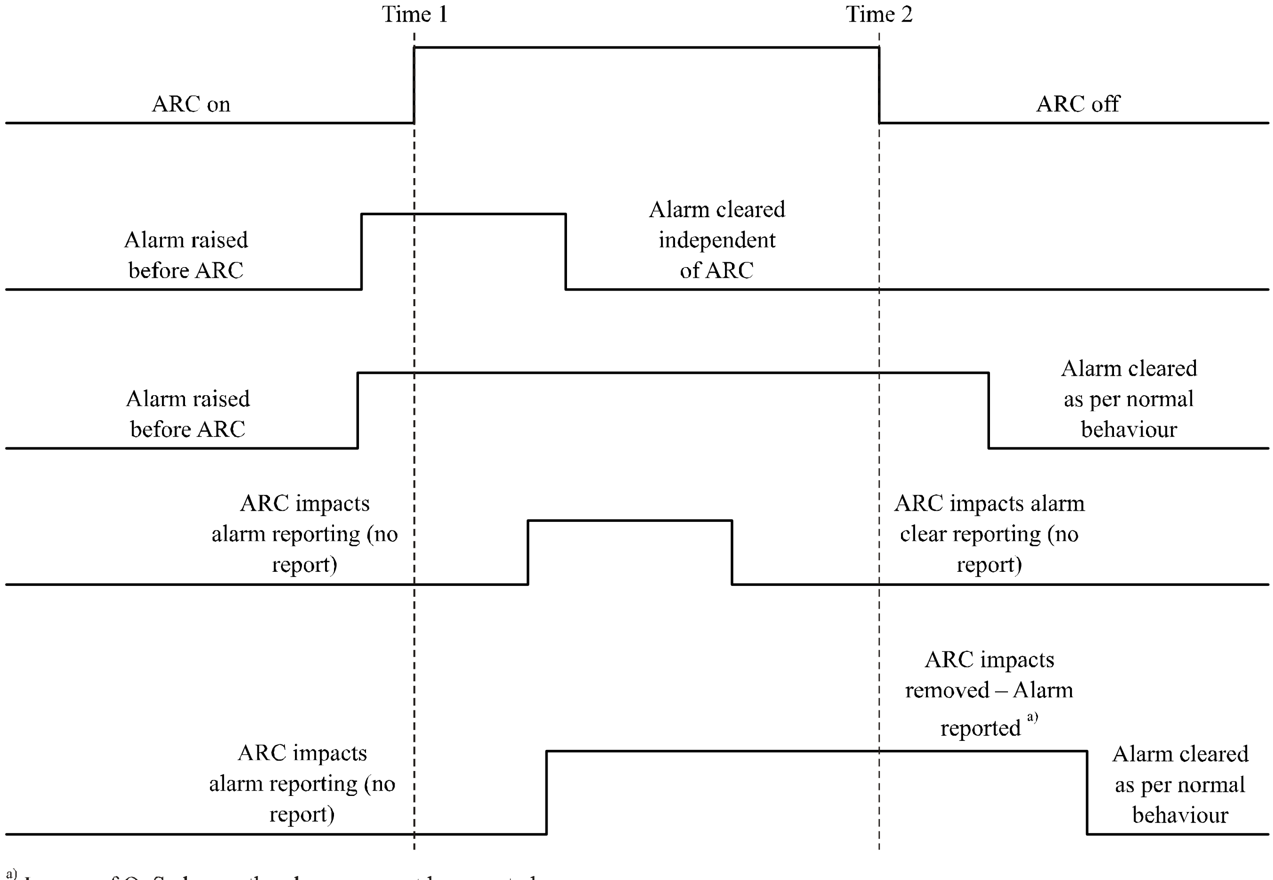
This use case describes the case of the operability of the resource determining and controlling when resource alarm reporting is to be turned on after having been set to a qualified inhibit. The criteria used to determine the operability of a given resource is technology-specific.

F.1.1.6 Timer-controlled ARC

This use case describes the case of an internal timer determining and controlling when resource alarm reporting is to be turned on after having been set to a timed inhibit.

F.1.1.7 Transition to reporting

This high-level use case contains all of the transition to reporting scenarios for ARC. Possible transition behaviours for a transition to reporting are illustrated in Figure F.2.


Time 1
Time 2
ARC on
ARC off
Alarm raised before ARC
Alarm cleared independent of ARC
Alarm raised before ARC
Alarm cleared as per normal behaviour
ARC impacts alarm reporting (no report)
ARC impacts alarm clear reporting (no report)
ARC impacts removed – Alarm reported a)
ARC impacts alarm reporting (no report)
Alarm cleared as per normal behaviour

[a] In case of QoS alarms, the alarm may not be reported.

M.3100_FF-2

Figure F.2/M.3100 – Alarm notification reference trace

F.1.2 Business requirements list

1) The Alarm Reporting Control feature shall support the following use cases:
   - alarm-free setup (and modification, breakdown) of lines, sections, and paths;
   - alarm-free modification of payload structures;
   - alarm-free intervals for some installation and maintenance activities.

2) A managed entity shall support the ability to turn on/off alarm reporting over its management interfaces.

3) When alarm reporting is turned off for any managed resource that reports alarms, performance monitoring threshold crossing alerts for the managed resource shall be inhibited. This requirement applies to counters. Gauges are for further study. One example of sending threshold crossing alerts is via the Quality of Service alarm.

4) When alarm reporting is turned off for any managed resource that reports alarms, performance measurement shall continue to be updated normally.

5) When alarm reporting is turned on for any managed resource that reports alarms after having been turned off, all associated performance monitoring threshold crossing alerts shall be allowed if inhibited by alarm report control. This requirement applies to counters. Gauges are for further study.

6) When a managed system/managed application has alarm reporting turned off for a managed resource, current alarm information and performance monitoring data shall be available over management interfaces by management request. Current alarm information shall identify, at minimum, the probable cause.

7) When alarm reporting is turned off, actions triggered, based on alarm monitoring information, shall continue to occur with the obvious exception of alarm reporting itself. For example, protection switching, operational state transitions, forward defect indication, backward defect indication, etc., shall continue to behave in the same manner as when alarm reporting is turned on.

F.1.3 Reference trace for ARC impact on alarm notifications

Figure F.2 first illustrates alarm reporting being inhibited for a period of time beginning with Time 1 and ending with Time 2. Then it illustrates what would happen in the event that an alarm existed prior to Time 1 and cleared before Time 2. It can be seen here that the alarm clear would be reported for this case when the clear occurred. The next event trace shows an alarm being raised before Time 1 and clearing after Time 2. The raise and clear for this alarm will be reported normally as these events are occurring outside of the Alarm Reporting Control window. The fourth trace illustrates the case where an alarm occurs and clears within the Alarm Reporting Control window. In this case, neither the raising nor the clearing of the alarm will be reported. The final trace shows the case of an alarm occurring during after Time 1 but before Time 2 and clearing after Time 2. In this case, the alarm will be reported at Time 2 and will have a timestamp indicating the actual time the event occurred. The clear will be reported as it is normally, as it is outside of the Alarm Reporting Control window.

In summary, the following points can be noted with regard to the provided traces:

1) Alarm clears for alarms that were reported prior to entering an Alarm Reporting Control mode shall not be inhibited.

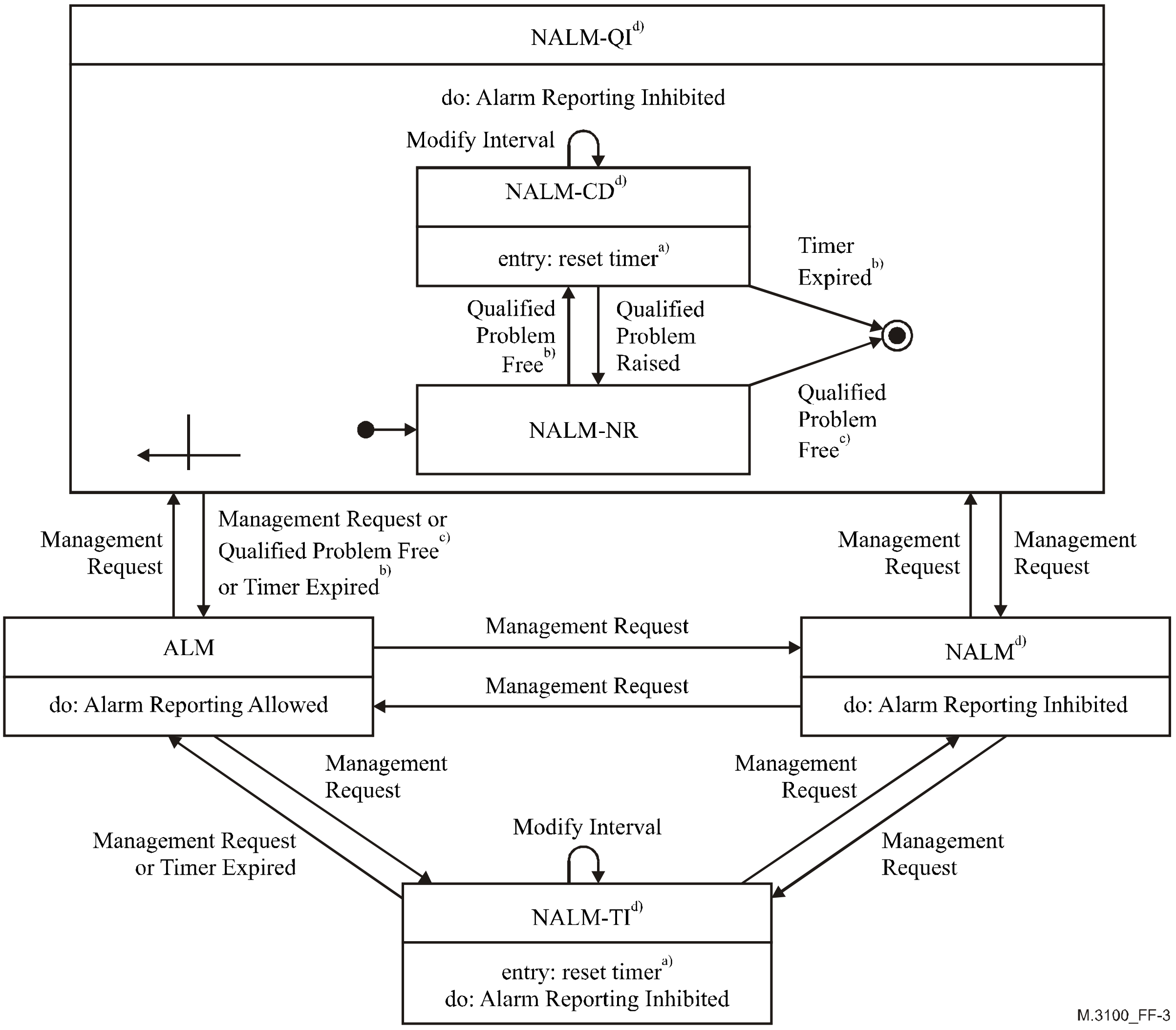
2) Alarm indications and associated clears that occur within an Alarm Reporting Control window shall be inhibited.

3) Alarms that occur in an Alarm Reporting Control window, and still existing when no longer in an Alarm Reporting Control mode, shall be reported in the transition to the normal reporting mode. These reports shall correctly identify the time the alarm was raised. In the case of quality of service alarms, the alarm may not be reported.

F.2 Analysis

F.2.1 ARC state diagram

This clause provides an illustration of the possible alarm reporting states for each managed resource providing the Alarm Reporting Control feature.



ALM	Alarm Reporting Allowed/Turned On
NALM	Alarm Reporting Inhibited/Turned Off (i.e. No Alarm Reporting)
TI	Timed Inhibit
CD	Count Down
NR	Not Ready
QI	Qualified Inhibit

[a] The interval may be set to zero.
[b] If NALM-CD is supported.
[c] If NALM-CD is NOT supported.
[d] Support for this state is optional at the generic level.

Figure F.3/M.3100 – Alarm reporting control state transition diagram

F.2.2 ARC state requirements list

This clause describes the generic Alarm Reporting Control requirements. Further details such as the default state and the list of states required to be supported is considered technology specific and will need to be addressed by technology-specific information model definitions.

1) Upon management requested creation of a managed resource, the ability to specify the state shall be provided.

2) The "ALM" state is required and at least one of the "NALM-T1", "NALM-QI" or "NALM" states.

3) If "NALM-QI" is supported, then the support of "NALM-NR" is required and "NALM-CD" is optional.

4) If the "NALM" state is supported, then a management request is required to change the resource to another state.

5) If the "NALM-QI" state is supported, then upon creation of a management representation of a managed resource, unless otherwise specified in a creation request, the managed system/managed application shall place the managed resource in the "NALM-QI" state and shall not report alarms for the managed resource over its management interfaces until the managed resource is in the "ALM" state.

6) The managed system/managed application shall not autonomously transition a managed resource from the "ALM" state. A management requested change from this state is required.

7) If the "NALM-CD" state is supported, a persistence interval should be provided to facilitate transitioning to the "ALM" state from the "NALM-QI" state.

8) If the "NALM-CD" state is supported, when the ARC interval timer expires in the "NALM-CD" state, the managed resource shall transition from the "NALM-QI" state to the "ALM" state.

9) If the "NALM-CD" state is not supported but the "NALM-QI" state is supported, when the managed entity becomes qualified problem-free, the managed resource shall transition from the "NALM-QI" state to the "ALM" state.

10) The time remaining for the persistence interval in the "NALM-QI" state shall be retrievable.

11) The default persistence interval should be programmable on a per managed system/managed application basis at minimum. If the persistence interval is programmable, the default persistence interval default value shall be documented in system management interface specifications.

12) When the ARC interval timer expires in the "NALM-TI" state, the managed resource shall transition from the "NALM-TI" state to the "ALM" state.

13) The time remaining for the timed interval in the "NALM-TI" state shall be retrievable.

14) The default timed interval for the "NALM-TI" state should be programmable on a per managed system/managed application basis at minimum. If the timed interval is programmable, the default timed interval default value shall be documented in system management interface specifications.

15) There shall be separate defaults for the "NALM-CD" and "NALM-TI" ARC intervals.

16) If the "NALM-QI" state is supported, in the management request to turn reporting off (i.e., when transitioning to the "NALM-QI" state), the manager shall be able to specify a persistence interval. This value is in effect until changed by another management request or until the state is exited. If a persistence interval is not specified in the management request, the default persistence interval shall be used.

17) If the "NALM-TI" state is supported, in the management request to turn reporting off (i.e., when transitioning to the "NALM-TI" state), the manager shall be able to specify a timed interval. This value is in effect until changed by another management request or until the state is exited. If a timed interval is not specified in the management request, the default timed interval shall be used.

18) If the "NALM-QI" state is supported, the persistence interval for a single managed entity shall be able to be modified via a management request while it is in the "NALM-QI" state. This value is in effect until changed by another management request or until the state is exited.

19) If the "NALM-TI" state is supported, the timed interval for a single managed entity shall be able to be modified via a management request while it is in the "NALM-TI" state. This value is in effect until changed by another management request or until the state is exited.

20) Upon auto creation of a managed resource and when the default state is either "NALM-QI" or "NALM-TI" and the ARC interval is programmable, the default interval (timed or persistence accordingly) shall be used.

21) The timed and persistence intervals shall be programmable between 0 and 99 hours with a one-minute granularity.

22) Queries of the time remaining shall be rounded up to the nearest minute.

23) The ARC interval timer(s) shall be accurate within ±10 seconds.

24) The managed resource shall support transitions directly to the "ALM" state from any other state via management request.

25) If the "NALM-TI" state is supported, the ability to place a managed resource that is in "ALM" state into "NALM-TI" state via management request shall be provided.

26) A managed resource shall not automatically transit into the "NALM-TI" state.

27) A timed interval shall be able to be specified with the management request to place a managed resource in the "NALM-TI" state.

28) Unless otherwise requested via management request, the managed resource that is placed in the "NALM-TI" state shall remain in that state, until the ARC interval timer expires, at which time it shall transit into the "ALM" state.

29) When a managed resource is manually placed in the "NALM", "NALM-QI" or "NALM-TI" state, the managed resource shall emit an autonomous message indicating that the alarm reporting of the managed resource is turned off. There shall be a different message for each ARC state (i.e., "NALM", "NALM-QI" and "NALM-TI").

30) When the managed resource transits into "ALM" state, an autonomous message shall be sent indicating that the managed resource's alarm reporting is turned on.

31) The managed entity shall support the ability to configure the list of probable causes (i.e., off-normal condition types) that will be inhibited by the Alarm Reporting Control. The default value for this list shall be all probable causes applicable for the managed entity.

32) If the "NALM-CD" state is supported and the managed entity is in the "NALM-NR" state and the managed entity becomes qualified problem-free, the managed entity shall transition to the "NALM-CD" state.

33) If the managed entity is in the "NALM-CD" state and a qualified problem occurs, the managed entity shall transition back to the "NALM-NR" state.

34) In the transition from the "ALM" state to any Alarm Reporting Control state, the controlled probable causes for the managed entity will be removed from the list of inputs for aggregate audibles/visuals.

35) In the transition to the "ALM" state from any Alarm Reporting Control state, alarms that had not been reported due to ARC but still present shall be reported. In addition, these previously controlled probable causes for the managed entity will be added to the list of inputs for aggregate audibles/visuals.

36) If timestamps are supported for alarm reports, the timestamp on any alarm report shall be the time the alarm event occurred. This means the timestamp when the alarm is sent on entry into the "ALM" state is the same as the timestamp would have been if the resource had been in the "ALM" state when the event occurred.

F.2.3 ARC state table

Table F.1/M.3100 – Alarm reporting control state event matrix

Event/State	ALM	NALM	NALM-TI	NALM-NR	NALM-CD
Managed resource becomes qualified problem-free	Clear alarm(s) as normal, Remain in ALM	Remain in NALM	Remain in NALM-TI	Transition to NALM-CD if NALM-CD supported; otherwise transition to ALM	
Qualified problem raised	Raise alarm(s) as normal, Remain in ALM	Remain in NALM	Remain in NALM-TI	Remain in NALM-NR	Transition to NALM-NR
Manager request to transition to ALM	Reject Request, Remain in ALM	Report existing alarms raised during ARC, Transition to ALM	Report existing alarms raised during ARC, Transition to ALM	Report existing alarms raised during ARC, Transition to ALM	Report existing alarms raised during ARC, Transition to ALM
Manager request to transition to NALM	Transition to NALM	Reject Request NALM	Transition to NALM	Transition to NALM	Transition to NALM
Manager request to transition to NALM-TI, interval not provided in request	if NALM-TI supported set timed interval to default timed interval and Transition to NALM-TI; otherwise Reject Request and Remain in ALM	if NALM-TI supported set timed interval to default timed interval and Transition to NALM-TI; otherwise Reject Request and Remain in NALM	Reject Request, Remain in NALM-TI	Reject Request, Remain in NALM-NR	Reject Request, Remain in NALM-CD
Manager request to transition to NALM-QI, interval not provided in request	if NALM-QI supported set persistence interval to default persistence interval and Transition to NALM-QI; otherwise Reject Request and Remain in ALM	if NALM-QI supported set persistence interval to default persistence interval and Transition to NALM-QI; otherwise Reject Request and Remain in NALM	Reject Request, Remain in NALM-TI	Reject Request, Remain in NALM-NR	Reject Request, Remain in NALM-CD

Table F.1/M.3100 – Alarm reporting control state event matrix

Event/State	ALM	NALM	NALM-TI	NALM-NR	NALM-CD
Manager request to transition to NALM-TI, interval provided in request	if NALM-TI supported set timed interval and Transition to NALM-TI; otherwise Reject Request and Remain in ALM	if NALM-TI supported set timed interval and Transition to NALM-TI; otherwise Reject Request and Remain in NALM	Reject Request, Remain in NALM-TI	Reject Request, Remain in NALM-NR	Reject Request, Remain in NALM-CD
Manager request to transition to NALM-QI, interval provided in request	if NALM-QI supported set persistence interval, Transition to NALM-NR; otherwise Reject Request and Remain in ALM	if NALM-QI supported set persistence interval, Transition to NALM-NR; otherwise Reject Request and Remain in NALM	Reject Request, Remain in NALM-TI	Reject Request, Remain in NALM-NR	Reject Request, Remain in NALM-CD
Timer expires			Report existing alarms raised during ARC, Transition to ALM		Report existing alarms raised during ARC, Transition to ALM
Manager request to modify persistence interval default	If current value specifies no adjustment, Reject Request; otherwise Change default, 1st potential use is next transition to NALM-QI	If current value specifies no adjustment, Reject Request; otherwise Change default, 1st potential use is next transition to NALM-QI	If current value specifies no adjustment, Reject Request; otherwise Change default, 1st potential use is next transition to NALM-QI	If current value specifies no adjustment, Reject Request; otherwise Change default, 1st potential use is next transition to NALM-QI	If current value specifies no adjustment, Reject Request; otherwise Change default, 1st potential use is next transition to NALM-QI
Manager request to modify timed interval default	If current value specifies no adjustment, Reject Request; otherwise Change default, 1st potential use is next transition to NALM-TI	If current value specifies no adjustment, Reject Request; otherwise Change default, 1st potential use is next transition to NALM-TI	If current value specifies no adjustment, Reject Request; otherwise Change default, 1st potential use is next transition to NALM-TI	If current value specifies no adjustment, Reject Request; otherwise Change default, 1st potential use is next transition to NALM-TI	If current value specifies no adjustment, Reject Request; otherwise Change default, 1st potential use is next transition to NALM-TI

Table F.1/M.3100 – Alarm reporting control state event matrix

Event/State	ALM	NALM	NALM-TI	NALM-NR	NALM-CD
Manager request to change ARC interval	Reject Request, Remain in ALM	Reject Request, Remain in NALM	Change Timed Interval, re-enter NALM-TI	Change Persistence Interval if NALM-CD supported; otherwise Reject Request, Remain in NALM-NR	Change Persistence Interval, re-enter NALM-CD
Manager request to modify ARC probable cause list	Modify list, Remain in ALM	Modify list, send alarms for existing alarms no longer inhibited, Remain in NALM	Modify list, send alarms for existing alarms no longer inhibited, Remain in NALM-TI	Modify list, send alarms for existing alarms no longer inhibited, re-determine if qualified problem-free, Remain in NALM-NR	Modify list, send alarms for existing alarms no longer inhibited, re-determine if qualified problem-free, Remain in NALM-CD

F.2.4 ARC object model

F.2.4.1 ARC class

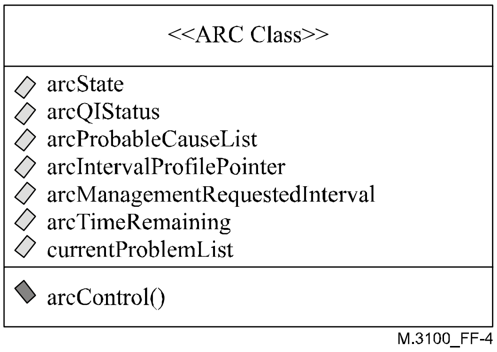


Figure F.4/M.3100 – ARC class

F.2.4.2 ARC interval profile class

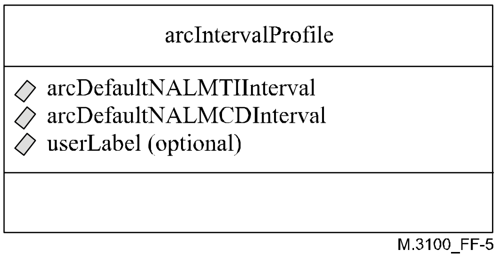


Figure F.5/M.3100 – ARC interval profile class

F.2.4.3 ARC retrieve alarm detail class

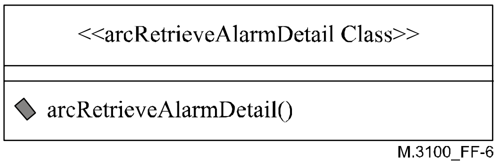


Figure F.6/M.3100 – ARC retrieve alarm detail class

F.2.5 ARC functional model description

This functional model description (in Figure F.7) has been included to show, within a typical managed system, the flow of information relating to a detected failure or probable cause. It has also been included to illustrate ARC impacts to the functional model. ARC causes certain probable causes to be marked as "not reported".

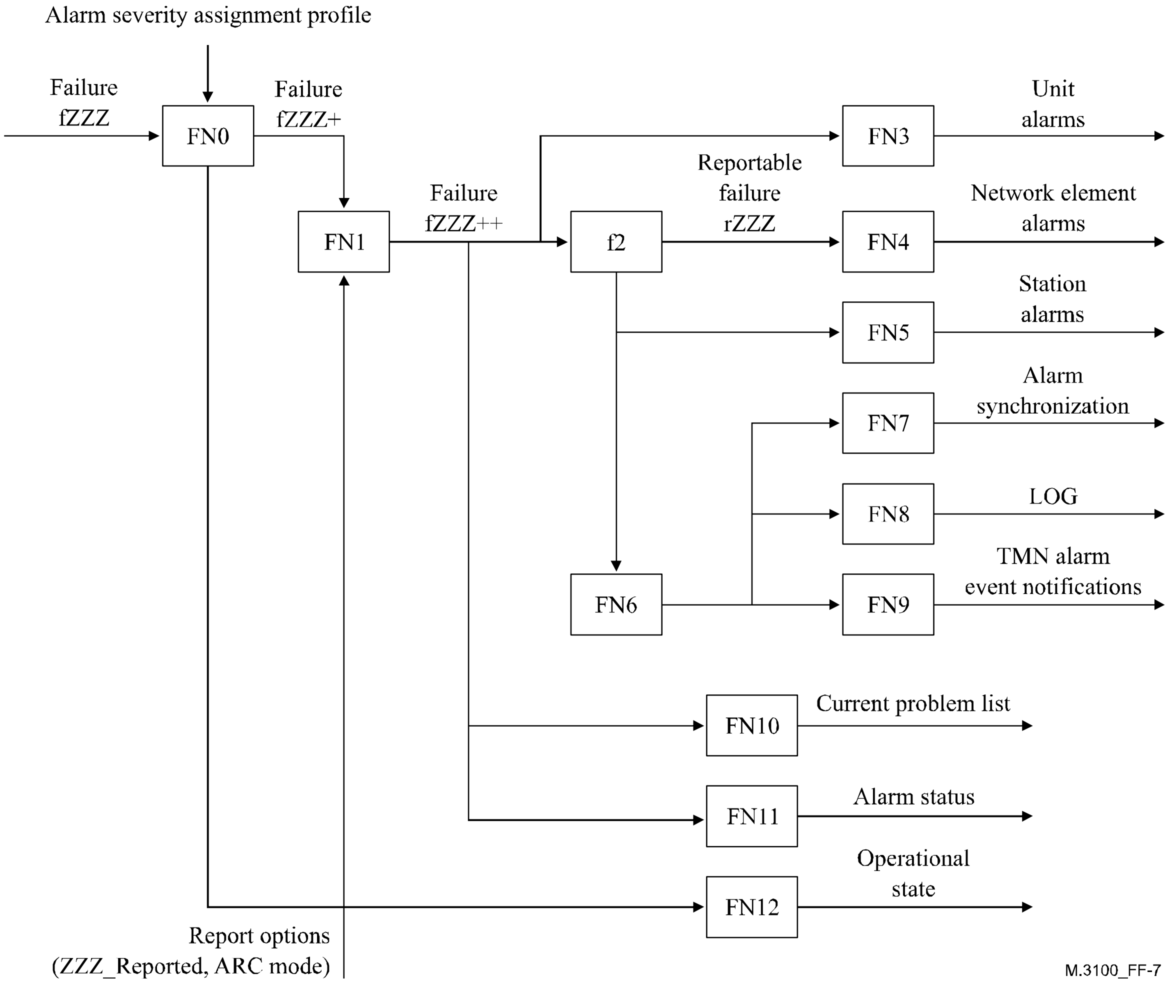


Figure F.7/M.3100 – Alarm flow functional model

- Function FN0 is responsible for assigning a severity for a given probable cause. The probable cause and its marked severity and other alarm information (including specific problem, backup status, trend indication, threshold info, event time, additional text, additional information, state change information, proposed repair action, monitored attributes, event type, managed object class, and managed object instance) are forwarded to function FN1. This alarm information is recorded at the time the alarm occurs.

- Function FN1 is responsible for marking a probable cause as "reported" or "not reported" for ARC. A probable cause is marked as "not reported" when the ARC information specified the probable cause to be "not reported". The "Failure fZZZ++" indicates the alarm status of the probable cause in addition to all object centric alarm information received from the FN0 function including: severity, specific problem, backup status, trend indication, threshold info, event time, additional text, additional information, state change information, proposed repair action, monitored attributes, event type, managed object class, and managed object instance. The output of the FN1 function is broadcast to FN3, f2, FN10, FN11 and FN12.
- Function f2 is a filter that forwards only probable cause indications that have been identified as reportable alarms by FN1. The output of the f2 filter is broadcast to functions FN4, FN5 and FN6.
- Function FN3 is responsible for determining whether or not unit audible/visual indicators need to be updated. The effect of ARC upon audible/visual indicators is left undefined in this Recommendation. It is only illustrated here to show that alarm information is forwarded to this function for application-specific processing.
- Function FN4 is responsible for determining whether or not aggregate audible/visual indicators need to be updated.
- Function FN5 is responsible for determining whether or not aggregate station audible/visual indicators need to be updated.
- Function FN6 is the TMN event pre-processing function. The output of the FN6 function is broadcast to functions FN7, FN8 and FN9.
- Function FN7 is responsible for storing all current reportable alarm information.
- Function FN8 is responsible for determining whether or not the event notification needs to be logged.
- Function FN9 is responsible for forwarding event notifications over the TMN.
- Function FN10 is responsible for updating the current problem list.
- Function FN11 is responsible for updating the alarm status.
- Function FN12 is responsible for updating operational state.

F.2.6 Alarm reporting parameters

Upon a transition from ARC, all alarm notification parameters in an alarm notification that need to be reported and that had occurred during ARC (other than notification identifier and correlated notifications) should reflect the values as defined in Table F.2.

Table F.2/M.3100 – Alarm reporting parameters table

Data in alarm notification	Set upon alarm occurrence/notification
Perceived severity	Occurrence
Probable cause	Occurrence
Specific problems	Occurrence
Backup status	Occurrence
Back up object	Occurrence
Trend indication	Occurrence
Threshold info	Occurrence
Event time	Occurrence

Table F.2/M.3100 – Alarm reporting parameters table

Data in alarm notification	Set upon alarm occurrence/notification
Additional text	Occurrence
Additional info	Occurrence
Notification identifier	Notification
Correlated notifications	Notification
State change information	Occurrence
Proposed repair action	Occurrence
Monitored attributes	Occurrence
Event type	Occurrence
Managed object class	Occurrence
Managed object instance	Occurrence

F.2.7 Relationship between ASAP, alarm status, and perceived severity

This clause discusses the relationship between alarm severity assignment specified in the alarm severity assignment profile and the perceived severity and alarm status values that are assigned to a probable cause both when in ARC (alarm reporting is turned off) and when not in ARC (alarm reporting is turned on).

When both in ARC and not in ARC, the perceived severity for a probable cause is assigned the same way. In addition, in the case that the alarm severity assignment profile is supported, this assignment is done based on the assignments made in the alarm severity assignment profile as indicated in Table F.3. However, when a probable cause is under ARC, the alarm status for that probable cause is always set to Pending.

Table F.3/M.3100 – Alarm severity and status table

Alarm severity assignment profile	Perceived severity	Alarm status	Alarm status in ARC
NA (Not Alarmed)	<unassigned>	Pending	Pending
WN (Warning)	WN	WN	Pending
MN (Minor)	MN	MN	Pending
MJ (Major)	MJ	MJ	Pending
CR (Critical)	CR	CR	Pending
<unassigned>	Indeterminate	Indeterminate	Pending
<any>	<unassigned>	Pending	Pending

F.2.8 ARC relationship to ITU-T Rec. Q.821

In the context of ARC, a current alarm is an outstanding problem (i.e., probableCause) and current alarm summary control returns current **reportable** (i.e., non-pending) alarms only. Alarms under alarm reporting control are not considered reportable alarms and, therefore, will not be included in alarm synchronization.

F.3 Design

F.3.1 CMIP/CMIS/CMISE

F.3.1.1 ARC management information model overview

The Alarm Reporting Control management information model is defined to overcome limitations in the ITU-T Rec. X.721 [5] and ITU-T Rec. X.734 [8] definition of EFD and limitations, in the definition of audibleVisualLocalAlarmPackage and resetAudibleAlarmPackage, when temporary alarm reporting control is needed, such as during some cases of controlled maintenance and provisioning.

Reasons why these mechanisms are considered inadequate include the following:

1) This feature requires temporary inhibition of reports for all managers. While EFD can do this, it is somewhat awkward for a manager to do this on another manager's behalf without an understanding of the other manager's EFD(s).

2) This feature requires that not only should alarms for a resource not be forwarded to a manager, but they should also not be included in aggregate audible/visual indicators. The EFD does not control audible/visual indicators. While there are controls for audible/visual indicators in the managedElement class and subclasses, these are controls for the behavior of the aggregate itself and do not control the information being fed into the aggregate.

3) This feature introduces methods to allow a resource to automatically transition from a non-reporting mode to a reporting mode. The EFD and the audible/visual functions do not support this capability.

This information model introduces several new definitions including:

- a new managed object class;
- a new package that may be included in the definition of any object class that supports alarms;
- a new parameter for clarifying the discrepancy between the time of the alarm and the time of the alarm notification when alarm reporting is resumed. This required a revision to all of the alarm reporting packages.

F.3.1.2 ARC managed object class

F.3.1.2.1 arcIntervalProfile

The Alarm Reporting Control interval profile managed object class provides the ability to configure default persistence and timed intervals for the "NALM-QI" and the "NALM-TI" states respectively. Association with this class implies that timing for both the "NALM-QI" and the "NALM-TI" states are supported by the resource. A class figure, using UML, has been provided in Figure F.8.

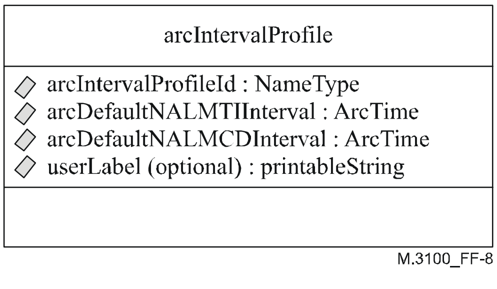

arcIntervalProfile
arcIntervalProfileId : NameType
arcDefaultNALMTIInterval : ArcTime
arcDefaultNALMCDInterval : ArcTime
userLabel (optional) : printableString
M.3100_FF-8

Figure F.8/M.3100 – Alarm reporting control interval profile object class

The Alarm reporting control interval profile managed object class, as shown in Figure F.8 has four attributes defined. Even though Figure F.8 does not show it, this class also supports some notifications.

F.3.1.2.2 Alarm reporting control interval profile inheritance hierarchy

Figure F.9 contains the inheritance hierarchy for the managed object class. Alarm reporting control interval profile is a concrete class (i.e., one that is expected to be used to instantiate managed objects). This class is a subclass of "top".

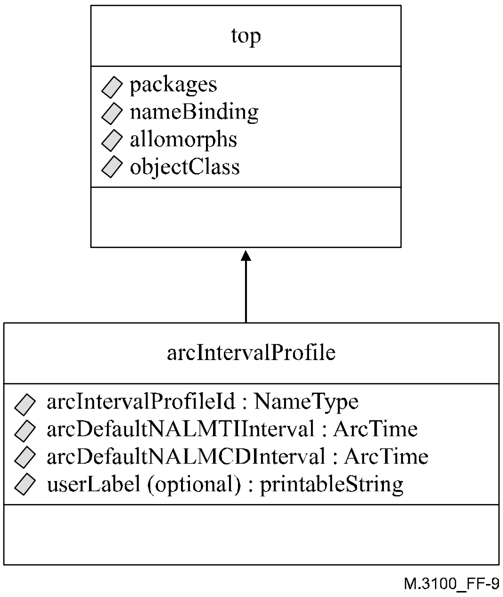

top
packages
nameBinding
allomorphs
objectClass
arcIntervalProfile
arcIntervalProfileId : NameType
arcDefaultNALMTIInterval : ArcTime
arcDefaultNALMCDInterval : ArcTime
userLabel (optional) : printableString
M.3100_FF-9

Figure F.9/M.3100 – Inheritance hierarchy

F.3.1.2.3 Name bindings

Multiple name bindings have been defined for this class to support the use of ARC in various types of systems (see Figure F.10).

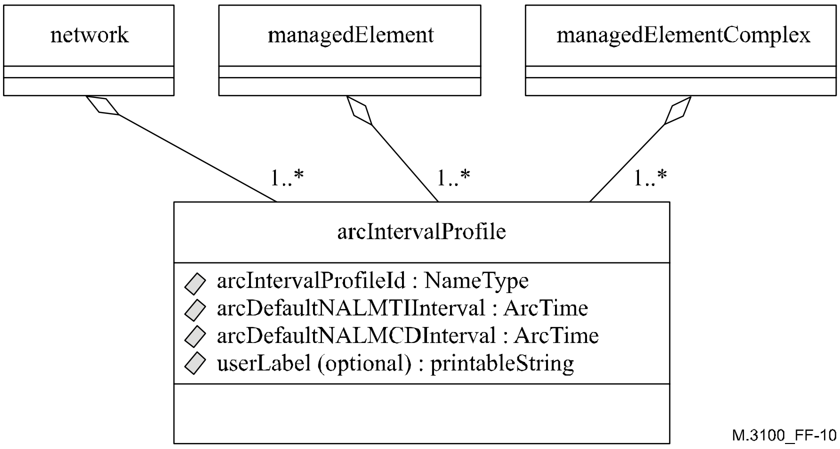
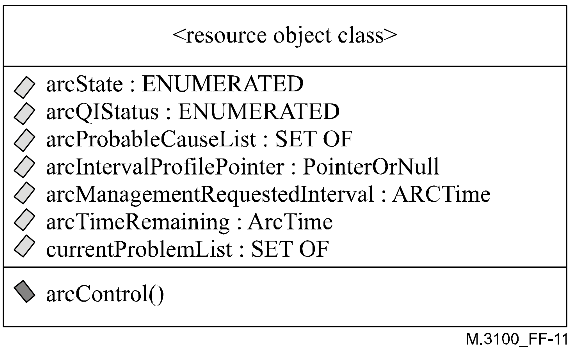

network
managedElement
managedElementComplex
1..*
1..*
1..*
arcIntervalProfile
arcIntervalProfileId : NameType
arcDefaultNALMTIInterval : ArcTime
arcDefaultNALMCDInterval : ArcTime
userLabel (optional) : printableString
M.3100_FF-10

Figure F.10/M.3100 – Naming tree hierarchy

The following namebindings will be provided: arcIntervalProfile-managedElement, arcIntervalProfile-managedElementComplex, arcIntervalProfile-network.

F.3.1.3 ARC package

This package has been defined to be included in object class definitions for objects that support alarm reporting. The characteristics illustrated (see Figure F.11) are in addition to the other characteristics defined for an object class.



<resource object class>
arcState : ENUMERATED
arcQIStatus : ENUMERATED
arcProbableCauseList : SET OF
arcIntervalProfilePointer : PointerOrNull
arcManagementRequestedInterval : ARCTime
arcTimeRemaining : ArcTime
currentProblemList : SET OF
arcControl()
M.3100_FF-11

Figure F.11/M.3100 – Alarm reporting control package

F.3.1.4 ARC retrieve alarm detail package

This package has been defined to be included in object class definitions for objects that support alarm reporting. The characteristics illustrated (see Figure F.12) are in addition to the other characteristics defined for an object class.

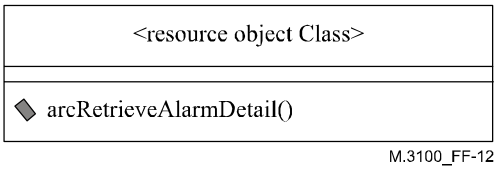


Figure F.12/M.3100 – ARC retrieve alarm detail package

F.3.1.5 Example application

This clause provides an example scenario of a given application of this information model.

Application Scenario 1

The default state is "NALM-QI" for all objects, the default persistence interval is 5 minutes, the default value for arcProbableCauseList is empty for all objects, the arcIntervalProfilePointer is not NULL and points to the object that defined the default persistence interval for all objects. All object classes support the currentProblemList and alarmStatus attributes. All object classes use the operationalState to determine operability for the managed resource.

1) A circuit pack is plugged-in.

2) The circuitPack object is auto created in the "NALM-QI" state. The arcTimeRemaining and arcManagementRequestedInterval are set to 5 minutes.

3) The circuitPack object creation causes the auto creation of the supported termination point(s). The termination point(s) are created in the "NALM-QI" state. The arcTimeRemaining and arcManagementRequestedInterval are set to 5 minutes.

4) The circuitPack is determined to be failed, the probableCause is added to the currentProblemList as pending and the alarmStatus is updated accordingly. In addition, the circuitPack waits to be qualified problem-free. Because the supported termination point(s)

are also inoperable due to their dependency on the circuitPack, they are also waiting to be qualified problem-free. The failed circuit pack is replaced.

5) It is determined that the failure has cleared and the circuitPack is now operationally enabled (i.e., operationalState = enabled). The circuitPack object begins counting down the persistence interval.

6) The supported Line termination points are detected to all have LOS failures. The LOS is added to the currentProblemList as pending and the alarmStatus is updated accordingly. The termination points wait to be qualified problem-free.

7) The persistence interval expires for the circuitPack, so it transitions to the "ALM" state.

Application Scenario 2

The default state is "NALM" for all objects, the default value for arcProbableCauseList is empty for all objects, the arcIntervalProfilePointer is NULL for all objects. All object classes support the currentProblemList and alarmStatus attributes. All object classes use the operationalState to determine operability for the managed resource. The path termination point (trail termination sink function) does not receive a signal (no signal, or unequipped signal). In this case, the path termination point is considered inoperable.

1) A multi-termination point circuit pack is plugged in and the termination points for that circuit pack are automatically created as a result in the NALM state. The transmitted trace identifiers are provisioned during the creation of the termination points.

2) A bidirectional connection is set up in the network and terminated at a termination point on that circuit pack; the expected trace identifier is provisioned at both termination points of the bidirectional connection.

3) When the connection is set up, both termination endpoints are queried for their fTIM status or for their received trace identifier values. If both fTIMs are cleared, or both received trace identifiers match the expected values, the termination point ARC state will be changed from NALM into ALM by means of a management request.

Application Scenario 3: Turn-up scenario

There is no signal present for the termination point. The default persistence interval is set to 0. The default state for the termination point is "NALM-NR". The default state for the circuit pack is "ALM".

1) A single port circuit pack is plugged in and the termination point for that circuit pack is automatically created as a result in the "NALM-NR" state.

2) The termination point managed resource becomes free of qualified problems, so it transitions to the "NALM-CD" state.

3) Because the default persistence interval value is set to zero, the termination point managed resource transitions to the "ALM" state immediately.

Application Scenario 4: Turn reporting off indefinitely

Reporting is turned on. The managed resource is free of qualified problems.

1) A management request to turn off reporting indefinitely (i.e., set state to "NALM"). As a result, the managed resource transitions from the "ALM" state to the "NALM" state.

Because there is no qualifying or strictly time-based criteria relevant for remaining in this state, the managed resource remains in this state indefinitely.

Application Scenario 5: Turning off reporting for 2 hours regardless of the failure state

An LOS is present on a termination point. The unavailable seconds (UAS) count is near threshold. UAS is a condition that does not persist and no clear is sent. The Craft is assigned 2 hours to repair trouble.

1) A management request is issued to place the termination point in "NALM-TI" state with a timed interval set to 2 hours.
2) An unavailable seconds (UAS) count exceeds its threshold; however, the notification is suppressed.
3) The maintenance personnel proceeds to fix the problem that caused the LOS.
4) At the end of about 1.5 hours, the maintenance personnel determines that more time is needed, and then he/she contacts the manager to extend the timed interval for 2 more hours.
5) The manager issues a command to reset the timed interval to 2 hours.
6) The LOS clears and an alarm clear notification is sent.
7) Approximately 3.5 hours after the termination point was first placed in "NALM-TI" state, the managed resource automatically transitions to the "ALM" state.

Application Scenario 6: Using NALM for alarm-free path setup

The default state is "NALM" for all managed entities, the default value for the configurable list of probable causes is set to all probable causes, the default ARC intervals are not configurable. The path termination point (trail termination sink function) does not receive a signal (no signal, or unequipped signal). In this case, the path termination point is considered inoperable.

1) A multi-termination point circuit pack is plugged in and the termination points for that circuit pack are automatically created as a result in the "NALM" state. The transmitted trace identifiers are provisioned during the creation of the termination points.
2) A bidirectional connection is set up in the network and terminated at a termination point on that circuit pack; the expected trace identifier is provisioned at both termination points of the bidirectional connection.
3) When the connection is set up, both termination endpoints are queried for their fTIM status or for their received trace identifier values. If both fTIMs are cleared, or both received trace identifiers match the expected values, the termination point ARC state will be changed from "NALM" into "ALM" by means of a management request.

F.3.1.6 Compliance

Managed object class definitions support the function of this Alarm Reporting Control (ARC) feature by incorporating the ARC package. Inclusion of the ARC package indicates that the managed object class supports the "ALM" state and at least one other state of the set "NALM", "NALM-TI", and "NALM-QI".

The definition of the managed object class including the ARC package shall specify in the behaviour clause which of the optional and conditional characteristics are to be utilized and any further restrictions on their use and their values. In particular, the behaviour of the managed object class shall clarify the following definitions:

1) The set of required ARC state values for the class.
2) The factors that determine operability for the class. For example, the definition for "qualified problem" may be augmented by the class definition by specifying a list of probable causes that affect operability.
3) The class shall specify whether or not existing quality of service alarms that were inhibited due to ARC must be reported when the resource transitions to the "ALM" state from any other ARC state.

4) If it is determined necessary to constrain the default ARC state value for the managed entity, the ARC state default value(s) allowed for the class shall be provided in the class definition.

Annex G

Bridge-and-roll cross-connect feature

G.1 Business requirements

This clause describes the generic bridge-and-roll cross-connect business requirements.

The bridge-and-roll process is used to move traffic from one facility to another facility without disruption. While the bridge-and-roll functions are separately managed within each NE, the overall process requires coordination across multiple network elements to ensure that the traffic is not disrupted. The logic of the bridge/roll/release is similar to protection switching, however, the applications are different. Protection switching is for restoration (recovery) and could be accomplished automatically (through signalling) or manually. Bridge/roll/release is for reconfiguration and accomplished manually, i.e., through management operations.

The bridge-and-roll functions are described in terms of three facilities on the network element:

- The *unchanged facility* is part of the current connection and will be part of the new connection.
- The *from facility* is part of the current connection, and will not be part of the new connection.
- The *to facility* is not part of the current connection, but will be part of the new connection.

The operations involved in a bridge-and-roll are defined as follows:

- The *bridge* operation causes traffic from the unchanged facility to be bridged to both the "from facilities" and "to facilities". The unchanged facility still receives traffic from the "from facility". This operation is applicable only to the source side of the signal.
- The *roll* operation causes the "unchanged facility" to receive traffic from the "to facility". This operation is applicable only to the sink side of the signal.
- The *release bridge* operation drops the connection between the "unchanged facility" and the "from facilities". This operation is applicable only to the source side of the signal.

G.2 High-level use cases

The terminology used in the use cases is based on terminology defined in this Recommendation and terminology defined in ITU-T Rec. M.3400, *TMN Management Functions*.

The set of use cases provided here is not exhaustive and is left as an exercise to the reader. Only that which was deemed necessary to clarify the need and the feature requirements is included.

This clause describes an example scenario where it is necessary to perform each of the three steps independently to ensure that the traffic is not disrupted. The key to the scenario is that there are two simultaneous bridge-and-roll processes that need to be coordinated to avoid a traffic disruption.

G.2.1 Use case 1 – two network elements, two facilities

The simplest case is two network elements, with two facilities connecting them. While this may not be particularly realistic, it does serve to illustrate the issue of coordination between the bridge-and-roll process in the two network elements. More realistic configurations would involve additional elements, but they do not change the fundamental process that is described in this example. In the diagrams that follow (see Figure G.1), boxes represent network elements, thick lines represent facilities, and thin lines with arrows represent connectivity within the network elements.

Traffic is initially on one facility, and is to be moved to the other.

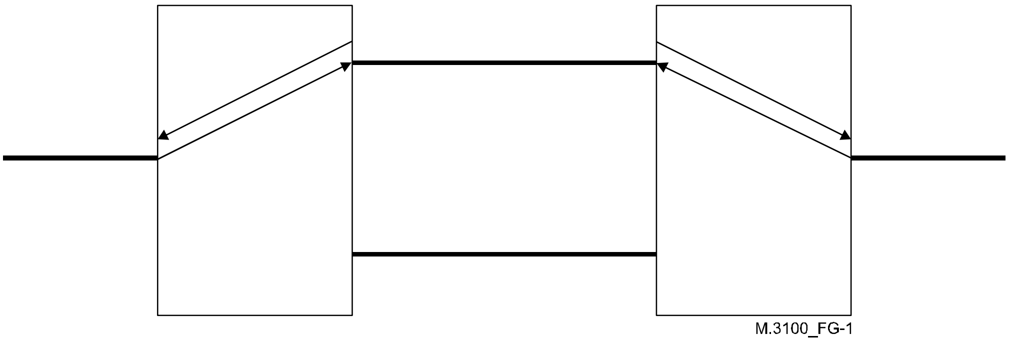


Figure G.1/M.3100 – Initial configuration

Using the present switchover action for fabricR1, traffic is lost unless there is precise coordination between the network elements. Unless the switchover command is executed at exactly the same time in both NEs, traffic is lost between the time that one NE is switched over and the time that the other NE is switched over from the *from facility* to the *to facility.* See Figure G.2.

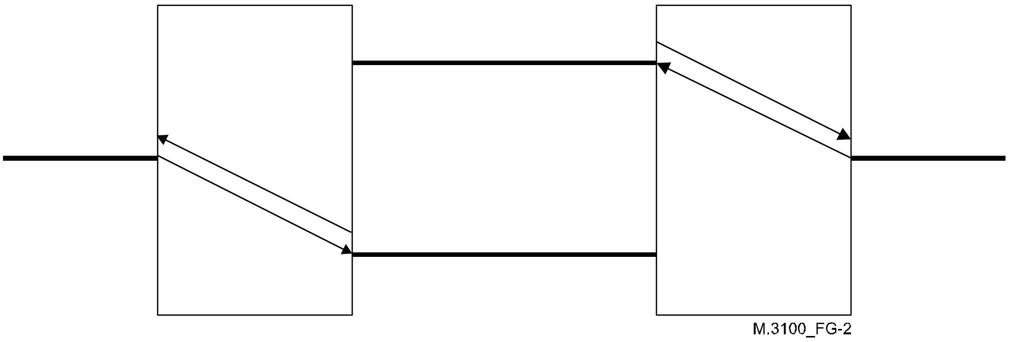


Figure G.2/M.3100 – Coordination problem when using switchover action

Using the bridge-and-roll process, traffic is first bridged to the new facility in both network elements. Note that both network elements still receive traffic from the top facility. See Figure G.3.

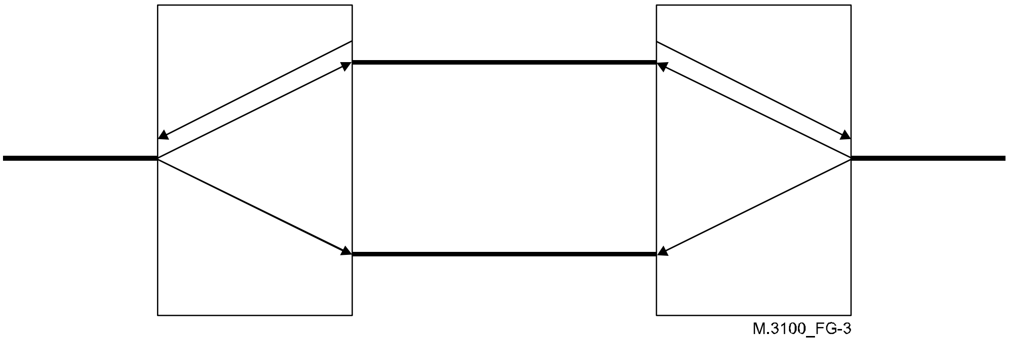


Figure G.3/M.3100 – Traffic bridged to new facility

If the roll-and-release operations are combined for the bridge-and-roll process, the flow of traffic will be disrupted unless the roll/release command is executed at exactly the same time in both network elements. Figure G.4 shows the result of the roll/release being preformed in only one of the network elements. The network element on the left is still receiving traffic; the NE on the right is not, since it has not yet rolled, and the left NE has dropped its bridge.

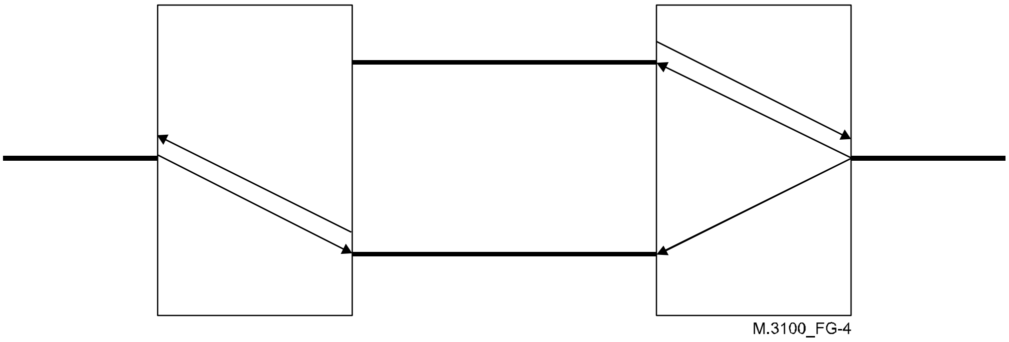


Figure G.4/M.3100 – Coordination problem when roll and release are combined

If the roll-and-release are separate operations, a roll can be performed at each network element, and then a release can be performed in each element. This eliminates the synchronization issue and ensures that the traffic is not disrupted. Figure G.5 shows the configuration after the roll operation has been performed in one network element. Traffic has not been disrupted, since the bridge is still in place. From here, the roll is done in the second NE, and then both bridges are released.

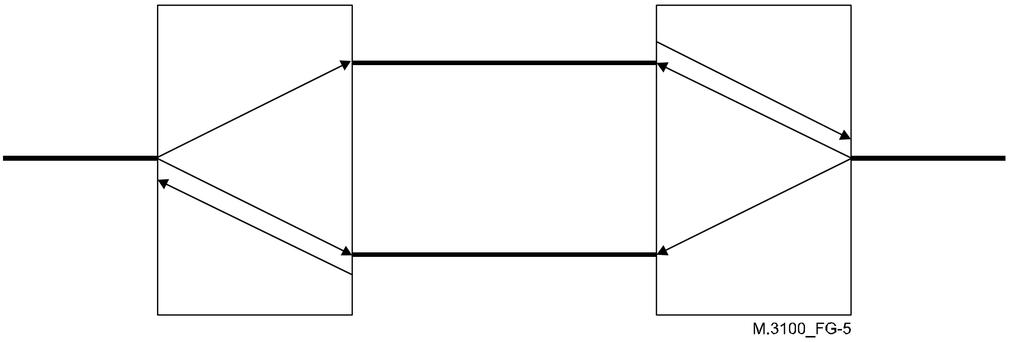


Figure G.5/M.3100 – Configuration after roll operation in one NE

G.3 Analysis

This clause shows how the bridge-and-roll model affects the configuration of the NE. The specific details of how the model works vary with the type of connection, based on the directionality, the connection type, and which end of the connection is being bridged.

G.3.1 Bidirectional connection

Figures G.6 to G.9 depict the series of steps to bridge-and-roll a bidirectional point-to-point connection within a single NE (CTP containers are omitted for clarity).

Figure G.6 shows a cross-connection between two CTPs, and a third unused CTP:

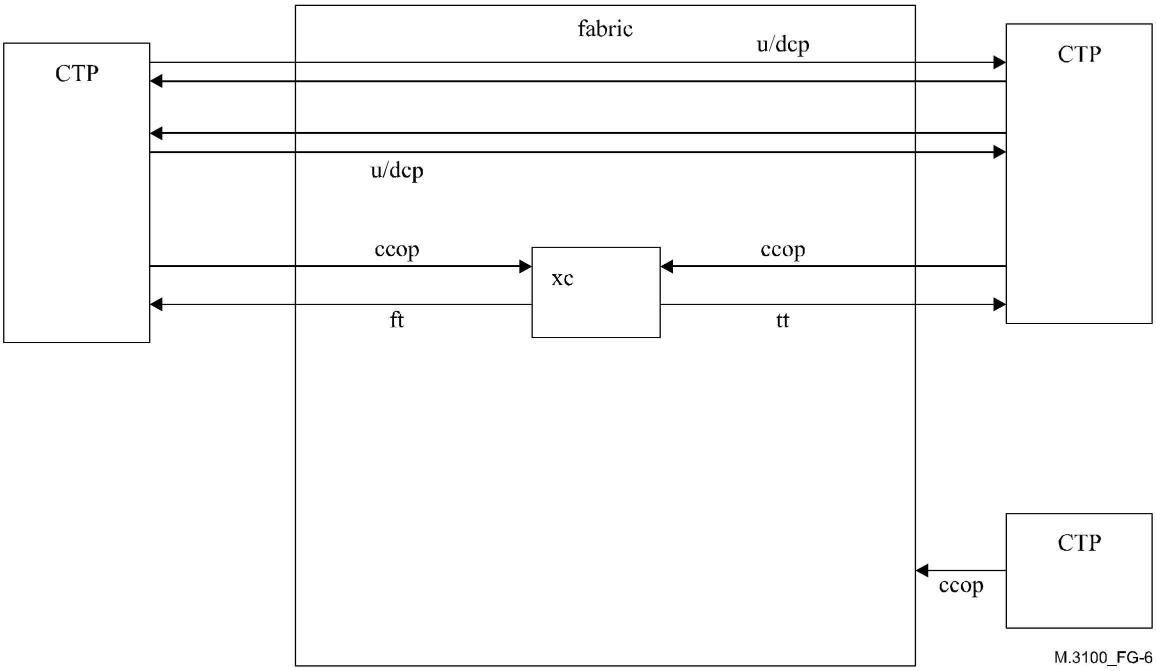


Figure G.6/M.3100 – Initial configuration (point-to-point bidirectional)

Figure G.7 shows the configuration after the bridge operation. A new one-way cross-connection has been created between the CTP on the left and the lower CTP on the right. The configuration of the bidirectional cross-connection is unaffected.

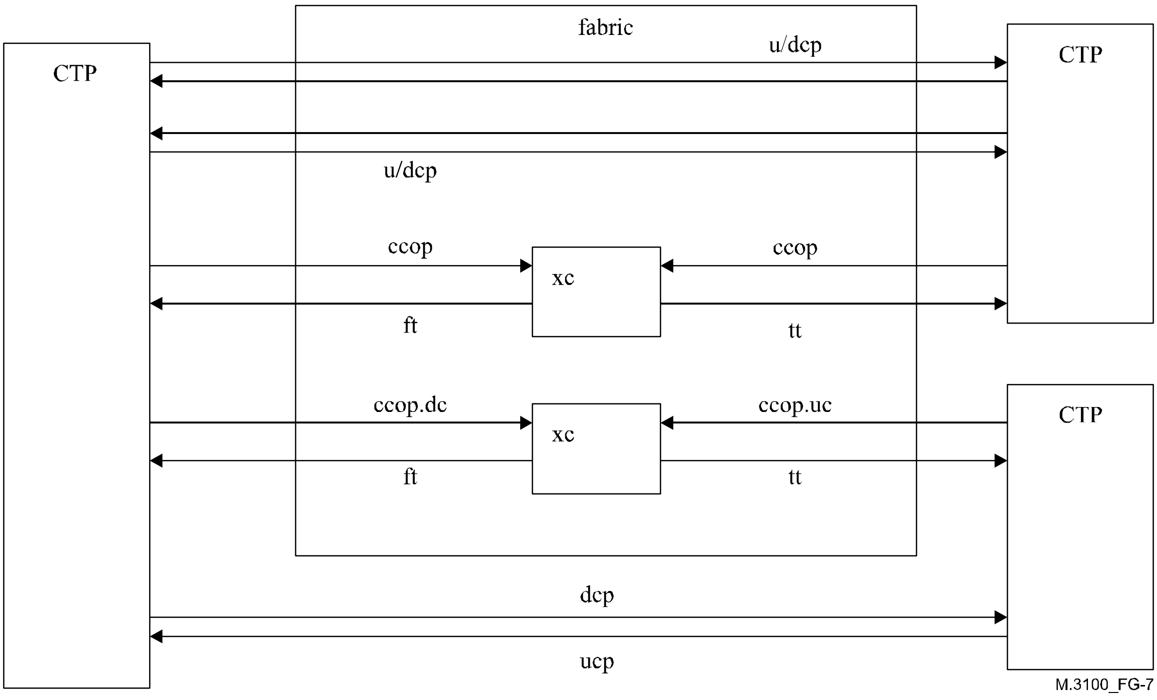


Figure G.7/M.3100 – Bridge operation (point-to-point bidirectional)

Figure G.8 shows the configuration after the roll operation. The top cross-connection is now unidirectional, and the bottom one is bidirectional.

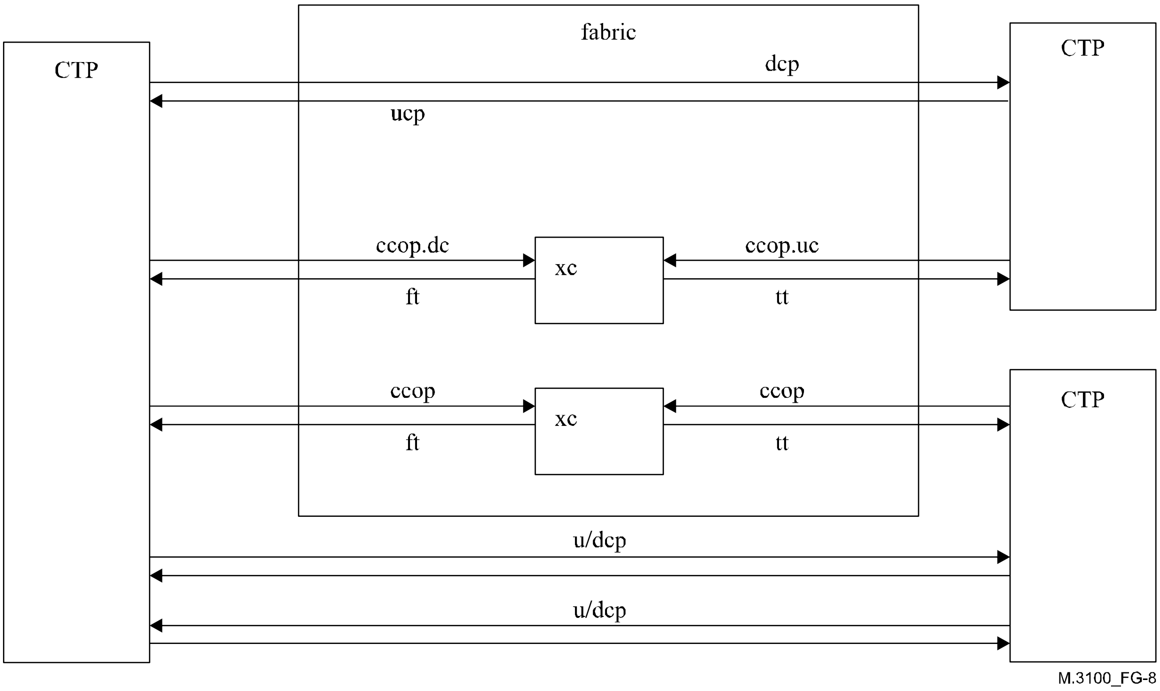


Figure G.8/M.3100 – Roll operation (point-to-point bidirectional)

Figure G.9 shows the configuration after the release operation. The unidirectional cross-connection has been deleted, and traffic now flows only between the left CTP and lower right CTP.

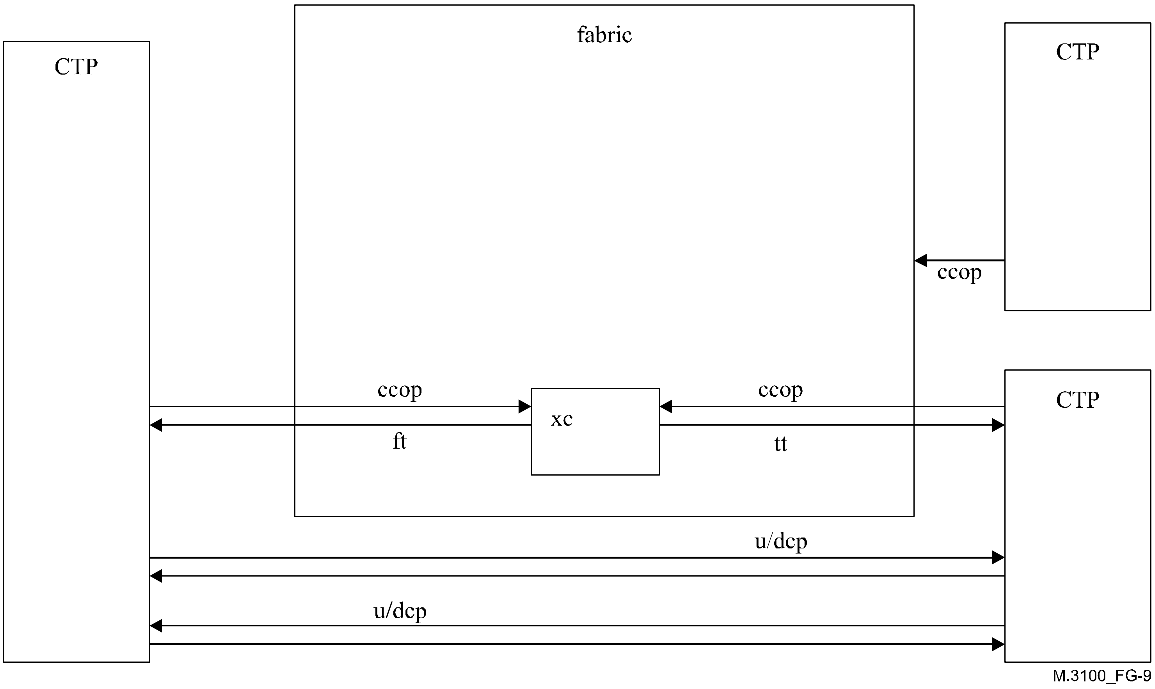


Figure G.9/M.3100 – Release bridge operation (point-to-point bidirectional)

G.3.2 Unidirectional connection

This clause describes the behaviour for unidirectional connections (both point-to-point and multicast) during the bridge-and-roll procedure. With unidirectional connections, the behaviour is different at each end of the connection. Figure G.10 shows a simple unidirectional connection that involves two network elements. For purposes of the discussion that follows, NE1 is called the source end, and NE2 is called the sink end.

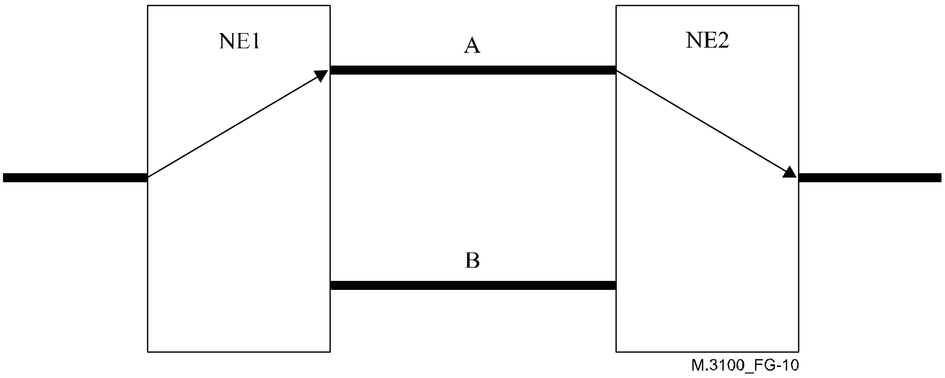


Figure G.10/M.3100 – Unidirectional cross-connection

G.3.3 Source end of unidirectional point-to-point connection

At the source end of a unidirectional point-to-point connection, the roll operation has no meaning, and is thus an optional step in the process (there is no data related to this connection flowing in that direction). The bridge and release bridge operations have the same meaning as they do in the bidirectional point-to-point case.

G.3.4 Source end of multicast connection

As is the case with the source of a unidirectional point-to-point connection, the roll operation has no meaning, and is optional. Moreover, the bridge operation is really no different than adding a leg to the multicast, and the release bridge operation is no different than removing a leg from the multicast. The bridge-and-roll process provides those functions for completeness (a typical application is to bridge all connections from one facility to another one, and it is desirable that the manager has a single interface for this application).

G.3.5 Sink end of unidirectional point-to-point connection or multicast connection

At the sink end of a unidirectional connection, the bridge operation does not cause any change in the flow of traffic. However, it is still important in some network elements to reserve the resources that will be used in the roll operation, thus, for some network elements, the bridge is a required operation. To enhance interoperability, it is proposed that the bridge request be accepted by all network elements, whether or not resources are reserved. When a network element does reserve resources, it is proposed that this reservation be indicated by pointing the ccop of the termination point to the existing cross-connection.

The roll operation consists of replacing one termination with another (i.e., there is no new cross-connection that gets created in this case), and the release bridge releases the reservation on the resources that are no longer used. This is shown in Figures G.11 to G.14 that follow (the point-to-point case is shown; the multicast case is very similar).

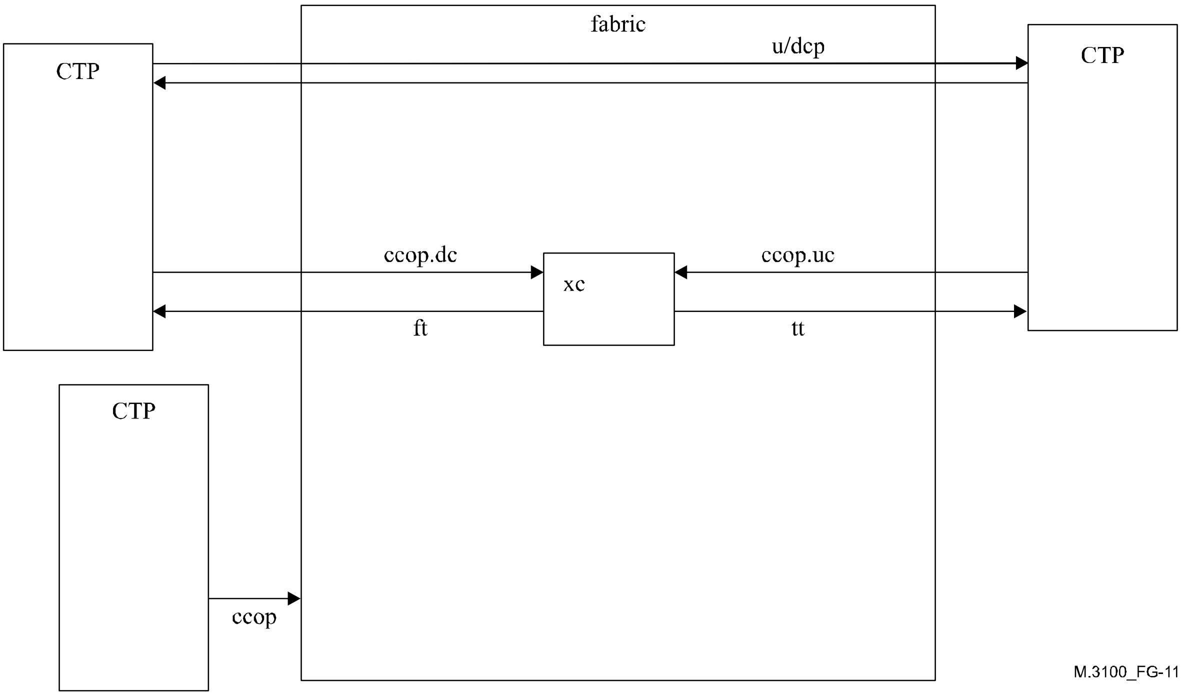


Figure G.11/M.3100 – Initial configuration (point-to-point unidirectional sink end)

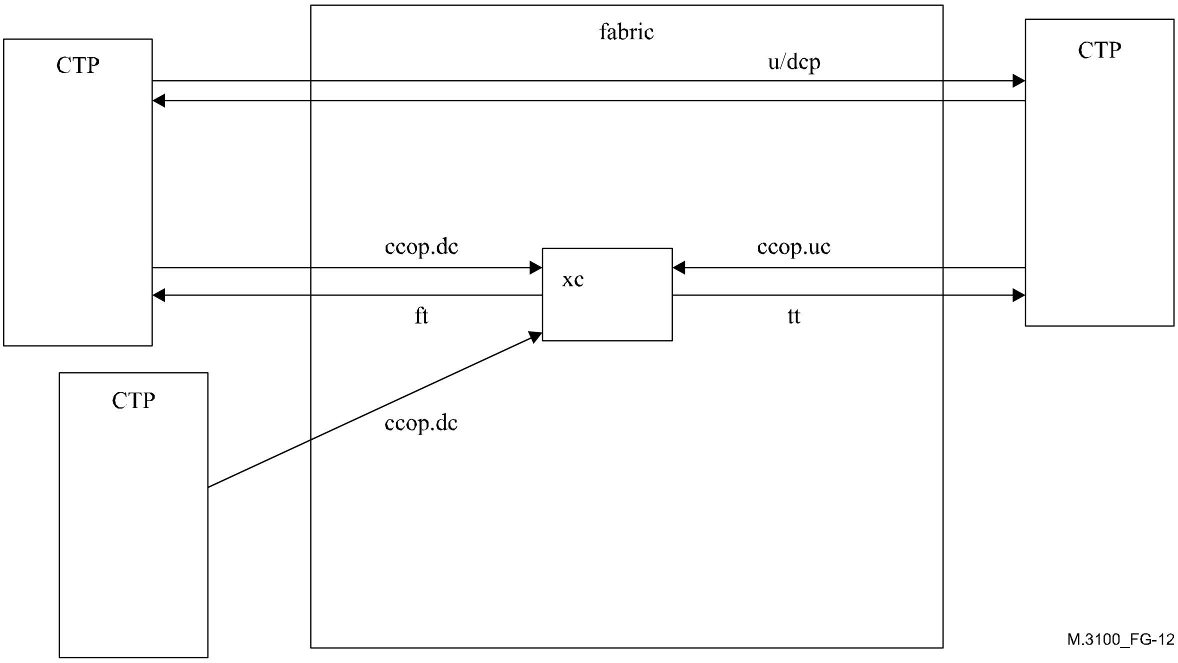


Figure G.12/M.3100 – Bridge operation (point-to-point unidirectional sink end)

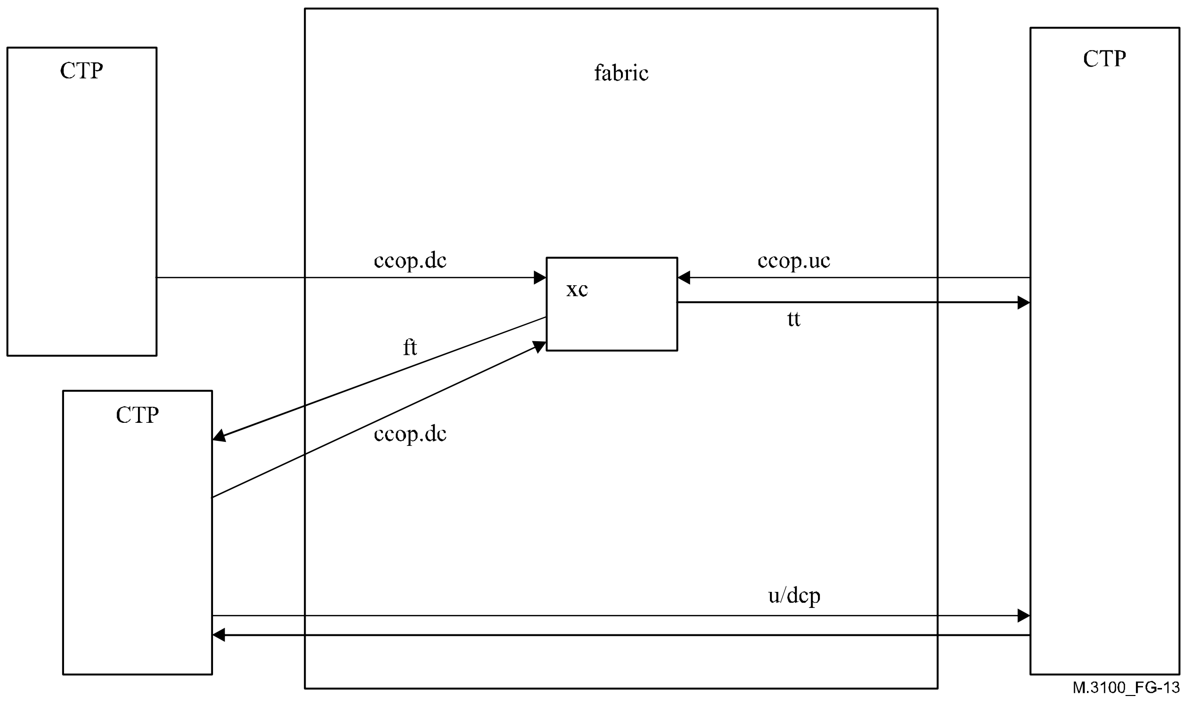


Figure G.13/M.3100 – Roll operation (point-to-point unidirectional sink end)

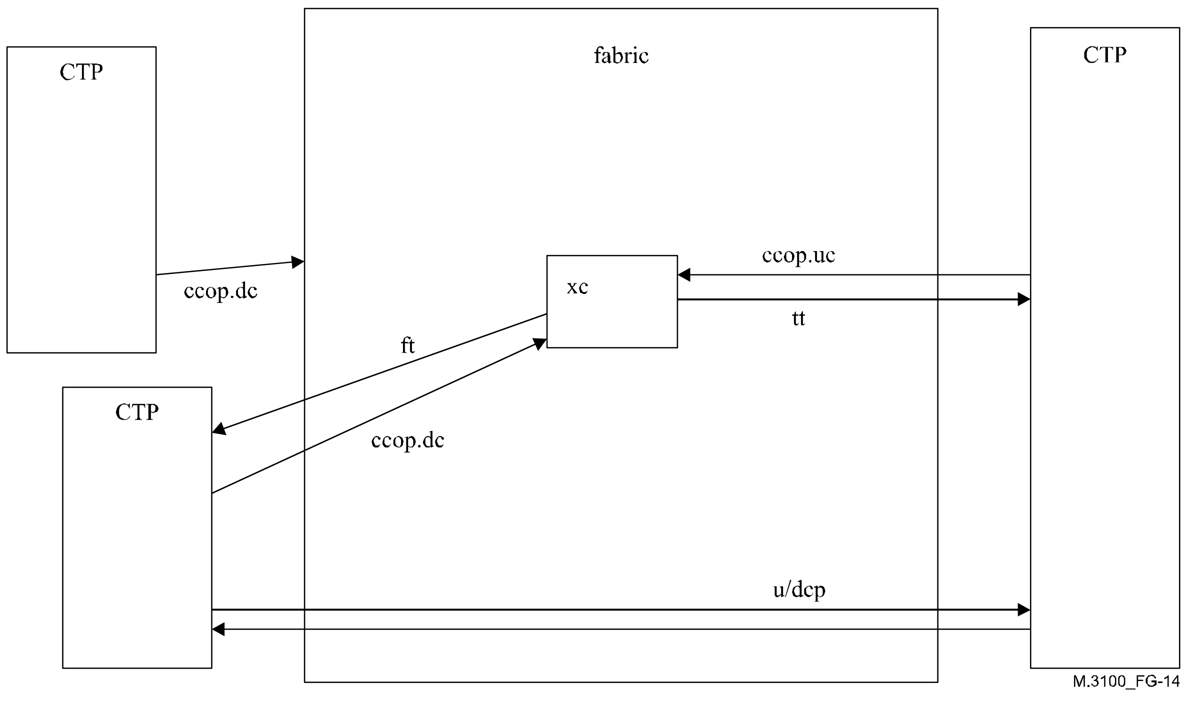


Figure G.14/M.3100 – Release operation (point-to-point unidirectional sink end)

G.3.6 Modelling alternatives

As stated in the introduction, none of the existing standards supports a 3-step bridge-and-roll process.

To support a three-step process, a new fabric object class is needed. There are two alternatives, depending on how important it is to preserve compatibility with the existing models:

- Derive a new fabric from fabricR2; this fabric would support a new action that allows either the 2-step or 3-step process.

- Derive a new fabric from X.721:top or fabric; this fabric would only support the 3-step process. If a 1- or 2-step process is desirable, the manager can provide such an interface, using the primitives from the 3-step process.

The choice between the two alternatives depends on how important the 2-step process is; if there is a need for an atomic roll and release operation, the first alternative is necessary. If not, the second alternative is perhaps simpler.

Annex H

Enhanced cross-connect feature

H.1 Business requirements

This clause describes the enhanced cross-connect business requirements. The current cross-connect model as defined by fabric (including revisions R1-R3) needs to be enhanced to support the following features:

1) Splitting a bidirectional cross-connect into two unidirectional cross-connects;

2) Changing from unidirectional cross-connect to broadcast;

3) Changing from broadcast to unidirectional cross-connect; and

4) Joining two unidirectional cross-connect into a single bidirectional cross-connect.

H.2 High-level use cases

The terminology used in the use cases is based on terminology defined in this Recommendation and terminology defined in ITU-T Rec. M.3400, *TMN Management Functions*.

The set of use cases provided here is not exhaustive and is left as an exercise to the reader. Only that which was deemed necessary to clarify the need and the feature requirements is included.

This clause describes some considerations related to interconnecting rings within a single network element, as well as some proposed additions to the fabric model related to interconnecting rings.

H.2.1 Use case 1 – Interconnection considerations

In general, ring interconnection is accomplished by using the same objects that are used for other cross-connections. Dual-homed, drop-and-continue, same-side interconnection requires some special handling; to create the necessary connections for that scheme, unidirectional connections must be used.

Figure H.1 shows the interconnection scenario from a network view. The solid line is the primary circuit; the dashed line is the secondary circuit. Thus, node C performs two independent connection protection functions in this scheme. In the D → H direction of transmission, the traffic is selected from node D or node F and in the H → D direction connections must be used, as shown in Figures H.2 and H.3.

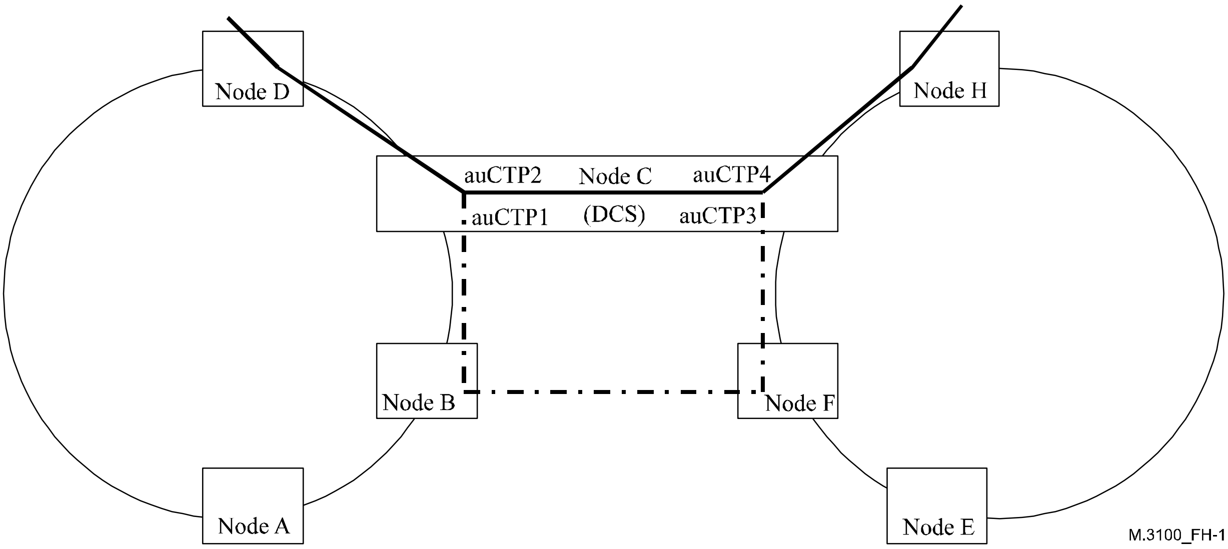


Figure H.1/M.3100 – Network view of same-side interconnection

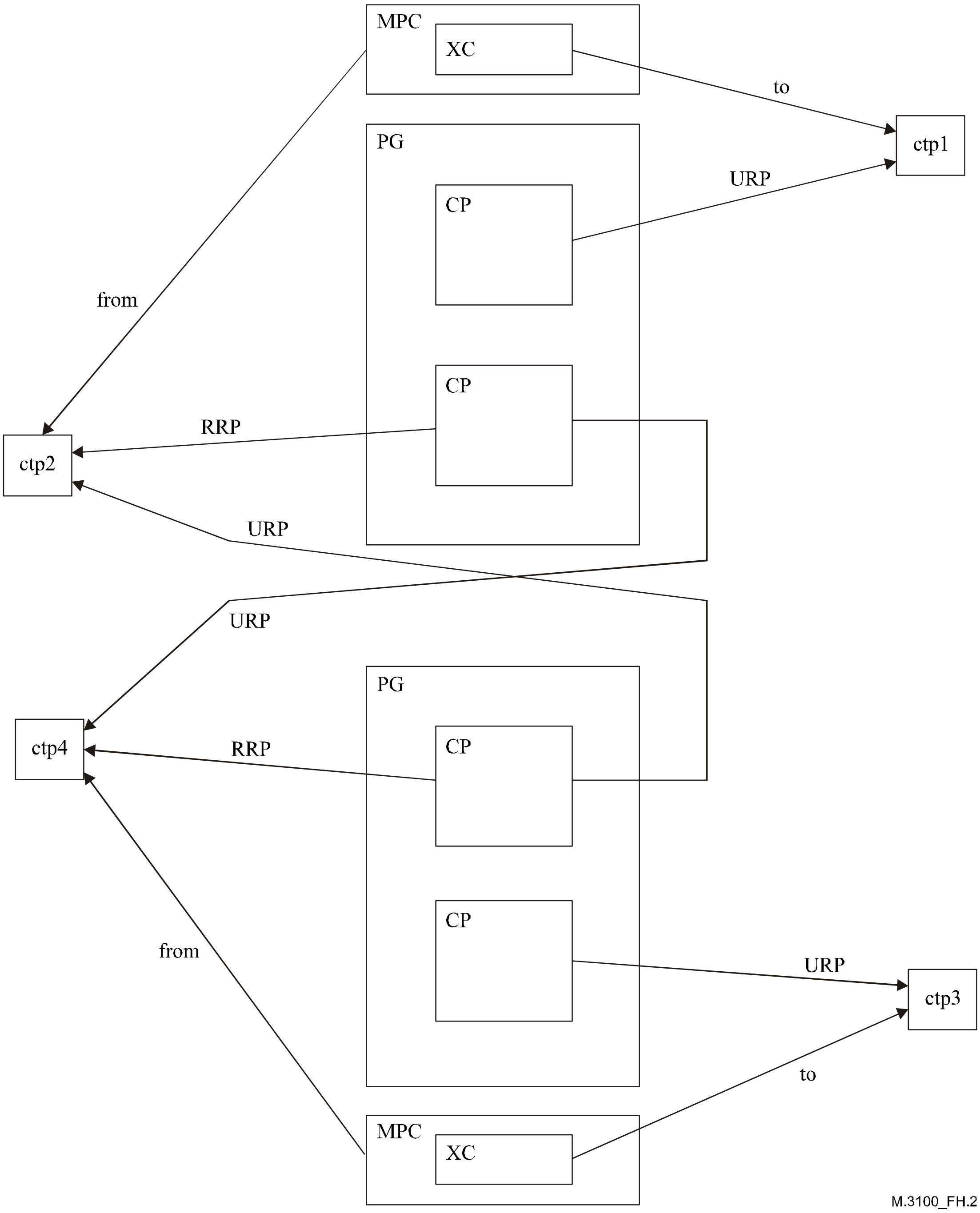


Figure H.2/M.3100 – Connections in Node C, connection pointers view

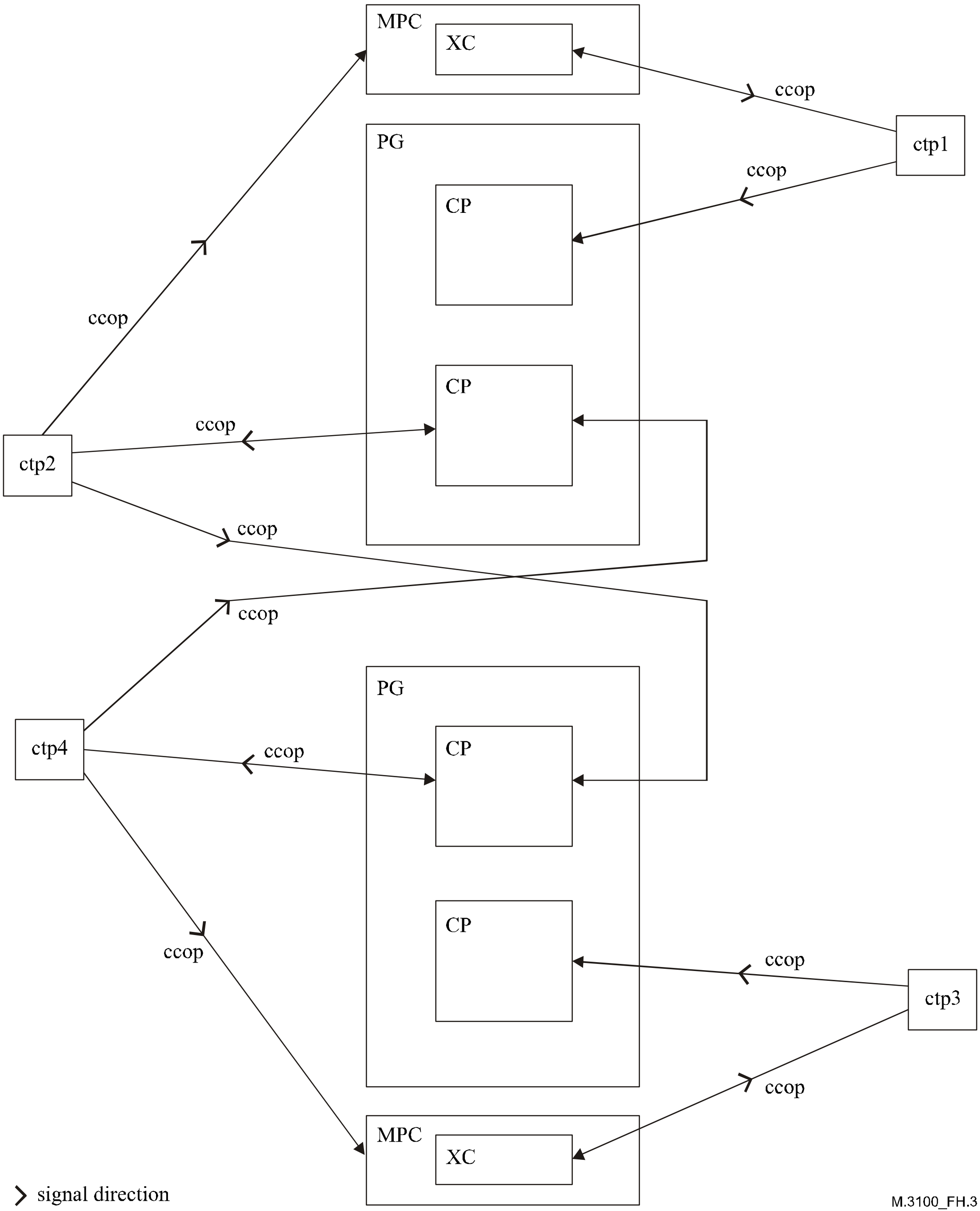


Figure H.3/M.3100 – Connection in Node C, termination point pointers

H.3 Analysis

The requirement to use unidirectional connections for certain interconnection schemes poses a problem, in that it is not possible with the current fabric model (fabricR2) to migrate from single-homed interconnection (which would be created as a simple bidirectional crossConnectionR1) to dual-homed, drop-and-continue, same-side interconnection (which uses unidirectional connections) without disrupting traffic.

To solve that problem, and create a fully flexible set of connection management functions, the following extensions are proposed to the fabric:

– ability to convert a bidirectional connection into a pair of unidirectional connections;

- ability to convert a pair of unidirectional connections between the same termination points into a single bidirectional connection;
- ability to convert a unidirectional point-to-point connection into a multicast connection with a single leg;
- ability to convert a multicast connection with a single leg into a unidirectional point-to-point connection.

All of these operations would be accomplished by deleting and creating the appropriate objects, without affecting traffic.

H.4 Design

This clause defines GDMO and CORBA IDL designs (both fine-grained and coarse-grained) for the enhanced cross-connect capability.

NOTE – In this version of this Recommendation, only the GDMO model is available.

Annex I

Attribute value ranges feature

This attribute value ranges feature provides a mechanism that would allow managed systems to automatically report acceptable value ranges for attributes associated with a network element in the model. Such a mechanism would be a valuable asset for equipment discovery and configuration, since a managing system would automatically be aware of the acceptable value ranges for each configurable parameter in the network before attempting to set these values.

For this mechanism to be implemented, we define a new AttributeRanges object class. The AttributeRanges class allows the managed system to report the minimum and maximum values a certain attribute accepts, as well as the granularity, or step increments, of the range. Each AttributeRanges instance contains ranges for attributes belonging to one object class. The "*kind*" attribute in AttributeRanges denotes the object class for which ranges are being defined. "*attributeName*" specifies the name of the attribute for which a range is being defined. The range is then defined using the "*minimum*", "*maximum*", and "*granularity*" attributes.

For each ManagedElement instance representing a network element, one or more AttributeRanges instances may be created. AttributeRanges instances are bound to the ManagedElement instance via a containment relationship.

Ranges are defined per ManagedElement instance. This allows for an attribute to have different ranges when it belongs to different network elements. In other words, the scope of each AttributeRanges instance is the relevant objects associated with the ManagedElement which contains the AttributeRanges instance. The managed system instantiates one AttributeRanges instance per class per Managed Element instance.

In order to set ranges for attributes defined inside data structures, the dot notation is used. For instance, consider the following data structure:

```
SampleStructureType::= SEQUENCE {
                                          xyz          REAL,
                                          abc          REAL,
                                          def          REAL
                                          }
```

In order to set an attribute range on attribute xyz, we may refer to attribute xyz by setting the attributeName attribute in the Ranges data structure to "*SampleStructureType.xyz*".

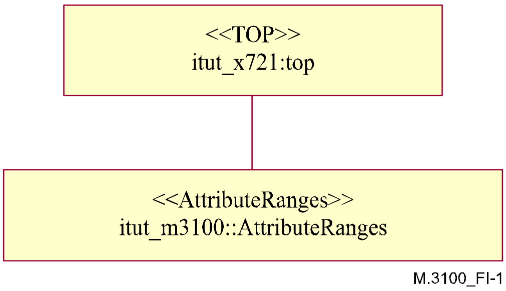


Figure I.1/M.3100 – AttributeRanges inheritance relationship

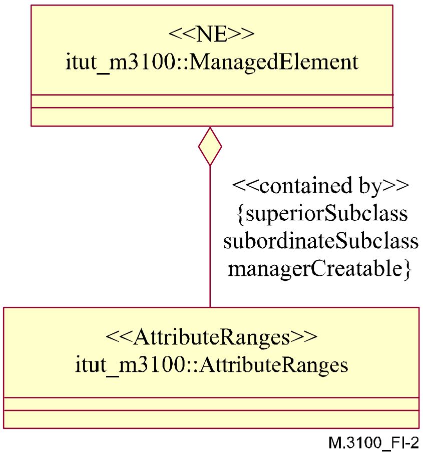


Figure I.2/M.3100 – AttributeRanges containment relationship

Annex J

Generic transport TTP feature

The generic transport TTP object class is used to represent a physical port or endpoints of transport connections. It may be used by technology-specific models as an abstraction of an underlying transport layer.

A new GenericTransportTTP interface is defined. This object is a subclass of NetworkTP. It is related to ManagedElement using a containment relationship. It is associated with CircuitPack using the PortAssociationList attribute, and with LinkEnd using the ClientLinkEndPointerList attribute.

Figures J.1 and J.2 show the inheritance, containment, and association relationships.

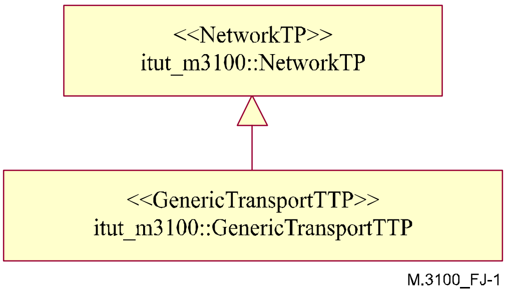


Figure J.1/M.3100 – Generic Transport TTP inheritance relationship

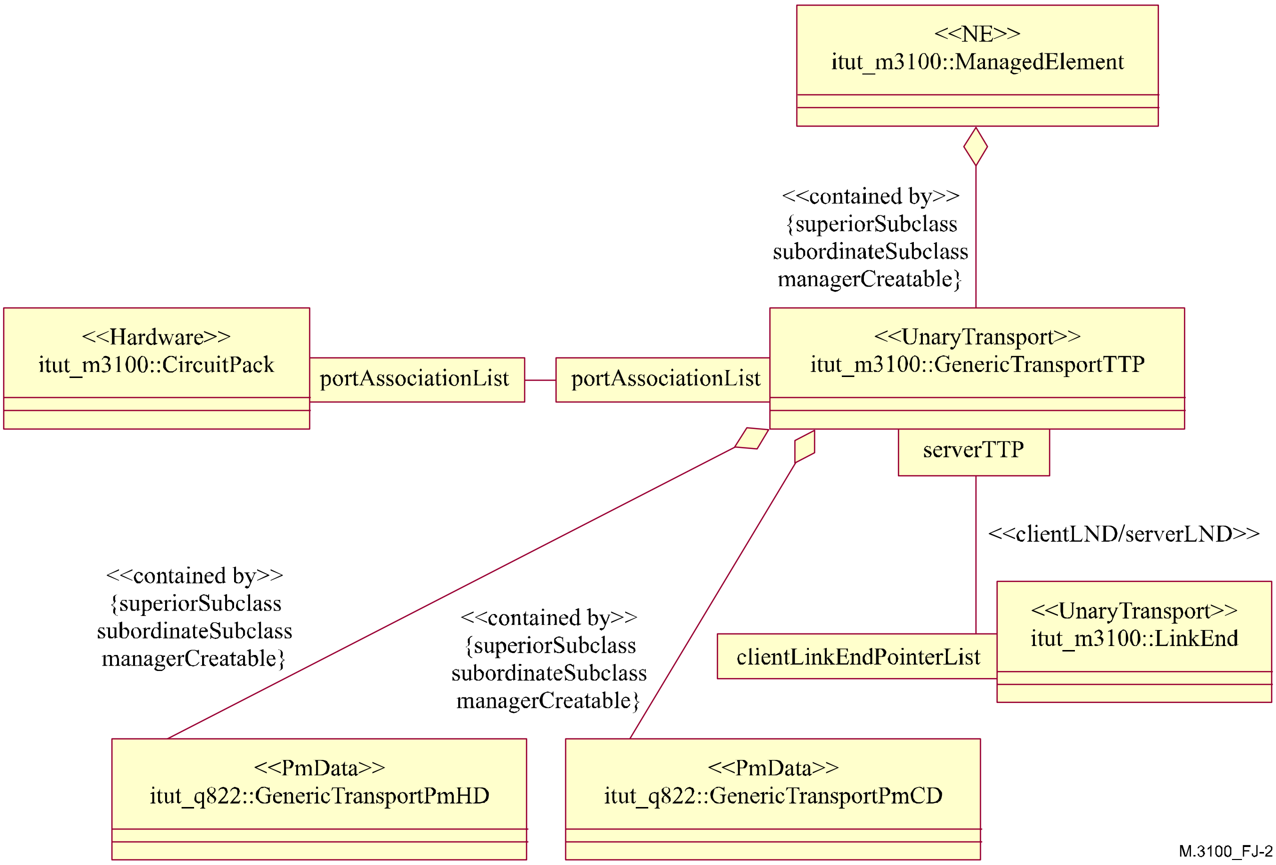


Figure J.2/M.3100 – Generic transport TTP containment and association relationships

Appendix I

User guidelines

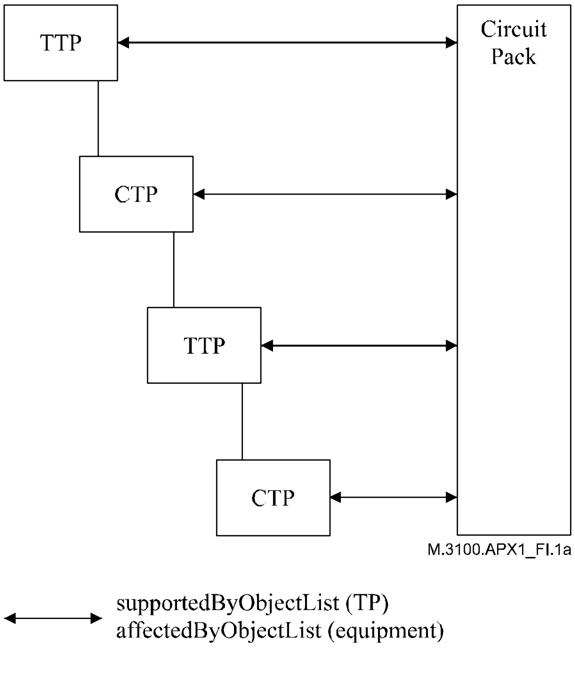
I.1 Introduction

This appendix contains user guidelines to clarify the use of object classes and attributes defined in this Recommendation. The clarification provided here to aid the users of this Recommendation in gaining further understanding of the model. In some cases, examples are provided on how to use the model for specific technology.

I.2 Use of supported by object list

This attribute is used to represent a dependency of that object that is contained in other objects. For example if contained in termination point, this attribute may point to a power source. Changes of state (e.g., disabled) of these other objects may affect the state of the object containing this attribute. In the above example, the termination point may become disabled because of the power source.

All termination point instances will be related to equipment when they are providing a service. The supportedByObjectList attribute in the TP object instance will point to an instance of circuit pack, and the circuit pack will point to all related TPs using the affectedObjectList attribute. The TPs indicate relationships with each other through the connectivity pointer attributes and naming, the supportedByObjectList attribute is generally not be used to indicate relationships between TPs (although exceptions exist, such as in ITU-T Rec. I.751). This is illustrated in the following figure:



I.3 Use of upstream and downstream connectivity pointers

I.3.1 Downstream connectivity pointer

This attribute indicates the instance(s) of termination point from which information (traffic) is received by the termination point object containing this attribute. This attribute points to (one or many) termination point(s) in the same managed element. However, a value of NULL can be used when the related object is in a different managed element or when the termination point is not connected. This attribute is read-only and cannot be modified directly. It will be updated as a side-effect of operations modifying the connectivity in the element (such as connect, disconnect operations on the fabric).

I.3.2 Upstream connectivity pointer

This attribute indicates the instance(s) of termination point to which information (traffic) is sent by the termination point object containing this attribute. This attribute points to (one or many) termination point(s) in the same managed element. However, a value of NULL can be used when the related object is in a different managed element or when the termination point is not connected. This attribute is read-only and cannot be modified directly. It will be updated as a side-effect of operations modifying the connectivity in the element (such as connect, disconnect operations on the fabric).

Figure I.1 gives an example on the use of these pointers in a unidirectional configuration composed of three network elements.

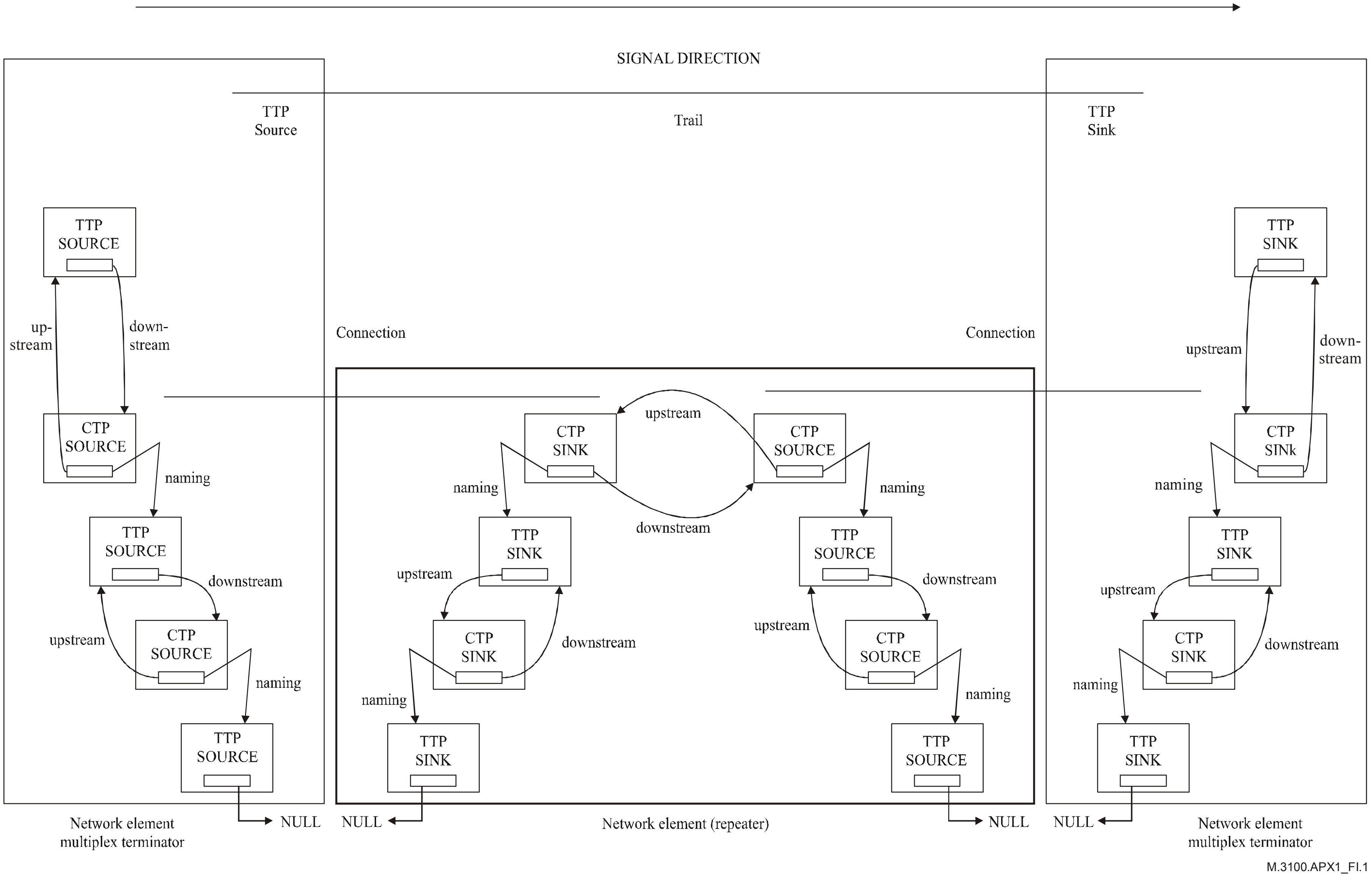


Figure I.1/M.3100 – Downstream, upstream pointer use example

I.4 Use of cross-connection objects

A connection between two termination points should be modelled using a cross-connection object when the assignment is flexible and can be modified through the management interface. This assignment can then be modified by using the connect/disconnect action on the fabric, and/or deleting the cross-connection objects directly.

In the cases where the connection cannot be modified through the management interface (e.g., the assignment is not flexible or human intervention is required), cross-connection objects should not be used. In those cases, the connection will simply be modelled by the connectivity pointers.

I.5 Cross-connection use examples

The goal of this clause is to provide examples of cross-connection modelling using the managed object classes defined in the cross-connection fragment (see 6.10). These examples are presented in the form of annotated figures. In these figures, thin lines represent connectivity pointers and thick, shaded lines represent CrossConnectionObject pointers and pointers in associated objects such as Cross-Connection and GTP. Generic examples are first presented, followed by examples applied to a specific technology (SDH).

I.5.1 Point-to-point cross-connection between two termination points

Figure I.2 provides an example of the simplest type of cross-connection: a point-to-point cross-connection between two termination points. The Upstream and Downstream connectivity pointers in each termination point point to the other termination point, the cross-connection object pointers in the termination points point to the cross-connection object and the From and To termination pointers in the cross-connection object point to the termination points.

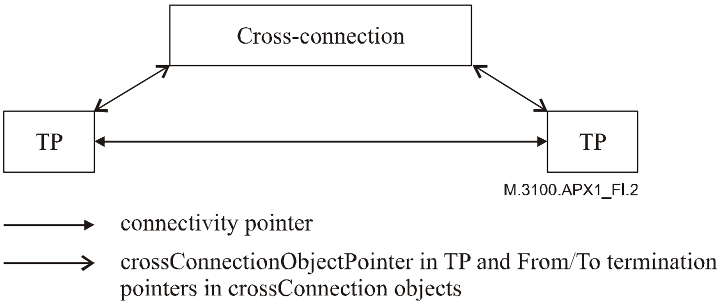

Cross-connection
TP
TP
M.3100.APX1_FI.2
connectivity pointer
crossConnectionObjectPointer in TP and From/To termination pointers in crossConnection objects

Figure I.2/M.3100 – Point-to-point cross-connection between two termination points

I.5.2 Cross-connection between two groups of termination points

There are cases where a group of termination points must be treated as a single entity and cross-connected as such. In such cases, the termination points to be grouped are included in a GTP (Group of Termination Points) object and it is the GTP objects that are cross-connected. Figure I.3 shows an example of such a configuration where each group contains two termination points.

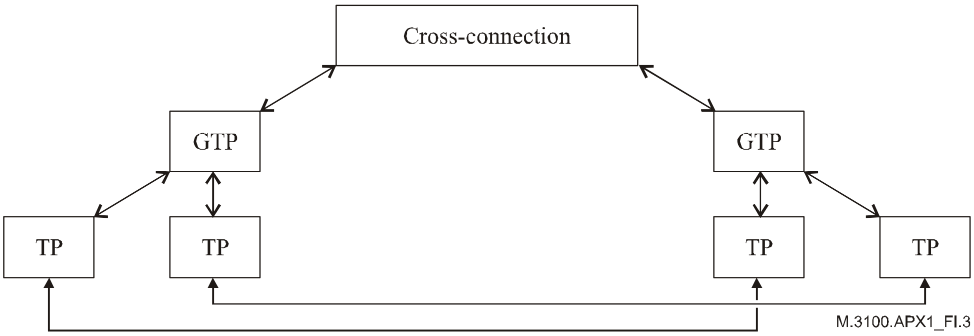


Figure I.3/M.3100 – Cross-connection with GTP

I.5.3 Modelling a broadcast

A Multipoint Cross-Connection object is used to represent a broadcast (or point-to-multipoint) cross-connection. The Multipoint Cross-Connect object only has a pointer to the originator of the broadcast. The destination of the broadcast is reflected in a pointer in the Cross-Connection Objects that are contained in the Multipoint Cross-Connection object. Figure I.4 shows an example of a broadcast configuration.

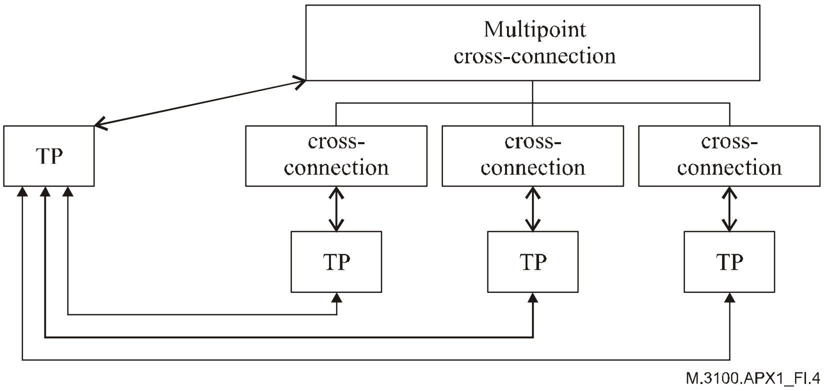


Figure I.4/M.3100 – Broadcast

I.5.4 Suspending a cross-connection

The model allows set-up of a cross-connection in a state which prevents traffic from flowing through it. For example, a cross-connection may be set up and tested but the telephone company wants to prevent traffic from flowing through it before the service is billed to the customer. This can be done either by putting the cross-connection in an intrusive test configuration and allowing a test signal to flow through it, or by locking the cross-connection, in which case the termination points will generate a 'not equipped' signal. The model supports the administrative state in the cross-connection objects to allow this situation. In this case, the crossConnectionObject pointers will remain the same, but the connectivity pointers in the termination points will be set to NULL. See Figure I.5.

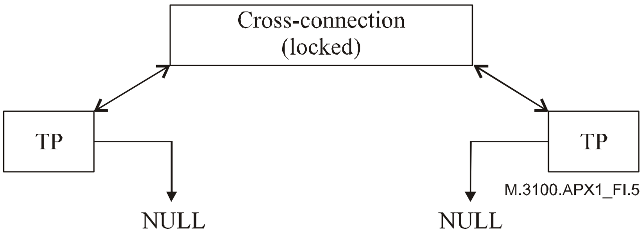


Figure I.5/M.3100 – Locked cross-connection

Applications of the cross-connection model in an SDH context

NOTE – For a detailed description of the SDH format, please refer to ITU-T Recs G.707/Y.1322, G.708, and G.709/Y.1331.

I.5.5 Queries of cross-connections

Given the M.3100 model, it is very easy to retrieve information on cross-connections. Queries based on the state of a cross-connection, its name, one of its endpoints or another attribute of the cross-connection can be performed by simple filtering on the cross-connection objects.

Also, to determine if a termination point is involved in a cross-connection, one simply has to look at the CrossConnectionObject pointer. If the CrossConnectionObject pointer points to the Fabric, the termination point is neither cross-connected nor reserved for cross-connection (assigned to a group). If the CrossConnectionObject pointer does not point to the Fabric, the termination point is assigned to a cross-connection or reserved. In this case, the source of the signal is indicated by the connectivity pointer.

I.5.6 Unidirectional cross-connection using GTPs

The model presents the cross-connection of groups of unidirectional termination points in an intuitive manner and permits knowledge of the connectivity between the termination points by issuing a single M-GET to retrieve the connectivity pointer in the Termination Point object. Figure I.6 provides an example of how such a cross-connection would be modelled. In this example, GTP objects are used to group two TU-11 (1.728 Mbit/s) termination points that are to be treated as a single entity for management purposes.

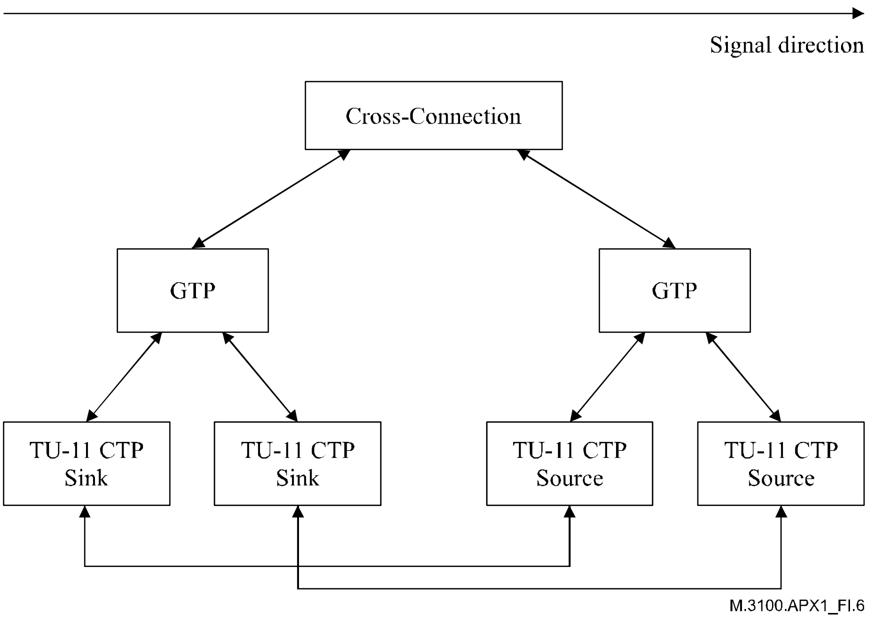


Figure I.6/M.3100 – Unidirectional 2xTU-11 point-to-point cross-connection with proposed model

I.5.7 Cross-connection of concatenated payloads

The SDH hierarchy allows for concatenated payloads, that is, several payloads of a lower rate can be combined to form a synchronous payload of a higher rate.

Cross-connection of concatenated payloads is a special case. A termination point whose traffic is carried by *n* concatenated payloads can be cross-connected to exactly *n* termination points of the lower rate. In all other cases, a point-to-point cross-connection will be established between termination points of the same characteristic information type, and each sink or bidirectional termination point will be the sink of exactly one termination point.

Since a sink, or bidirectional termination point, whose traffic is carried by *n* concatenated payloads may be the sink of zero or *n* source or bidirectional termination points, its connectivity pointer must be special. The connectivity pointer in such a termination point will point to either zero or *n* source or bidirectional termination points. All the other sink or bidirectional termination points will have a connectivity pointer that can point only to zero or one termination point.

Figure I.7 is an example of this situation using the VC-4 according to the North American practice. The VC-4 (155 Mbit/s) is carried by three AU-3s (52 Mbit/s each) and, as such, can be cross-connected to three AU-3s. Another configuration is to connect the VC-4 directly to an AU-4 (see Figure I.8).

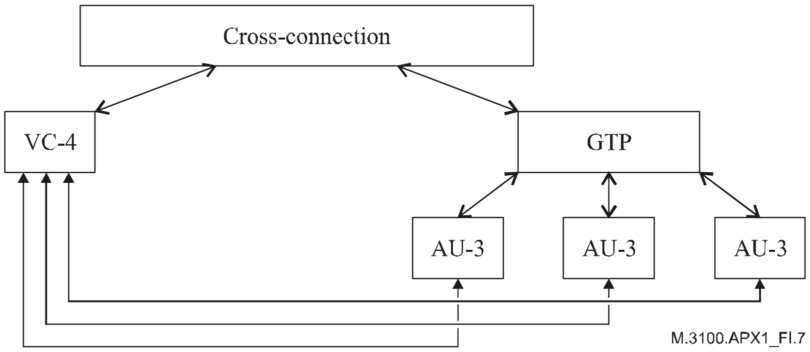


Figure I.7/M.3100 – VC-4 connected to three AU-3s

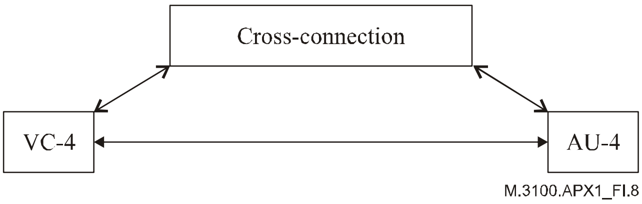


Figure I.8/M.3100 – VC-4 connected to one AU-4

I.5.8 Cross-connection of indirect adaptors

The SDH model introduces the concept of "indirect adaptors" representing an intermediate step in the multiplexing process. One such adaptor is the TUG-2 at 6.9 Mbit/s which represents the multiplexing of either four TU-11 (1.7 Mbit/s), three TU-12 (2.3 Mbit/s), or one TU-2 (6.9 Mbit/s).

The model allows for cross-connection of the termination points contained in indirect adaptors. The manager does not have to be aware of the actual content of the group but can still manage the cross-connection of the group as a whole. An example of this situation is the cross-connection of TUG-2s where the content of the TUG-2 may change while the cross-connection remains undisturbed. The connect action requests the connection of two TUG-2s. Two GTPs are automatically created to reflect the content of the TUG-2s. A cross-connection is established between these GTPs (see Figure I.9). If the content of the TUG-2s changes (for example from three TU-12s to one TU-2) the content of the GTPs will also change to reflect this, but the management view of the cross-connection will not change (see Figure I.10).

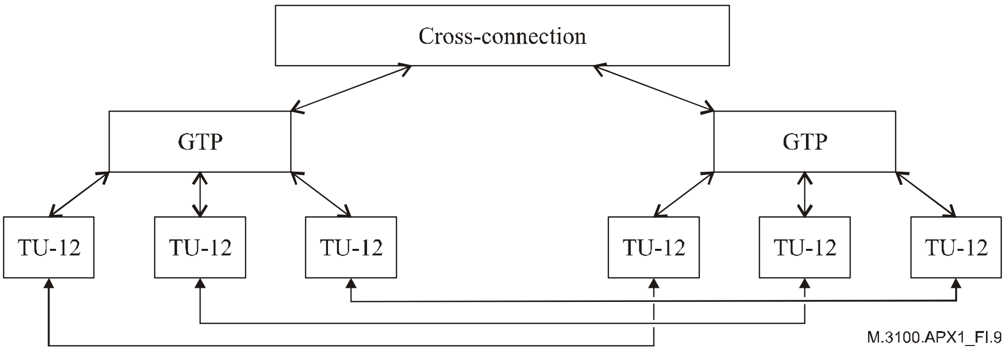


Figure I.9/M.3100 – Cross-connection of two TUG-2s containing three TU-12s

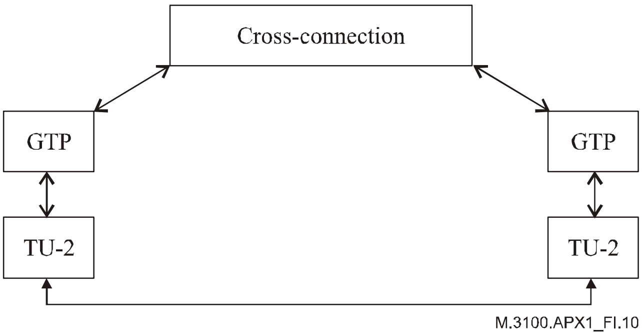


Figure I.10/M.3100 – The two TUG-2s now contain one TU-2

I.5.9 Cross-connection of arbitrary groups

The model allows for the cross-connection of arbitrary GTPs. The only restriction is that the GTPs must be composed of compatible termination points. (See Figure I.11.)

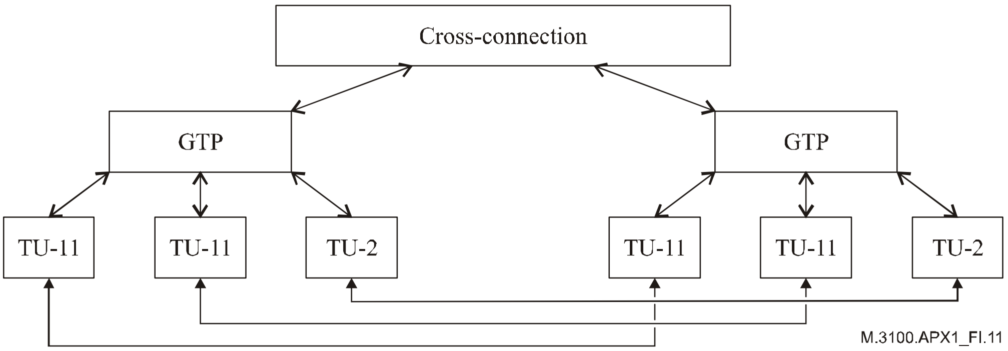


Figure I.11/M.3100 – Cross-connection of GTPs composed of two TU-11s and one TU-2

I.5.10 Use of cross-connection object pointer

The MultipleConnections choice of the crossConnectionObjectPointer attribute syntax is used in SDH when a bidirectional termination point is connected in both directions, using unidirectional cross-connection object in each direction to two other TPs (see Figure I.12) or one other bidirectional TP. This choice is also used in ITU-T Rec. G.774.4 for protection of a broadcast where part of the legs can be protected and others are not protected. In this case, two different objects are used, the first one is an mpCrossConnection object that holds the individual unprotected legs, and the other is an mpCrossConnectionProtection that holds all protected legs.

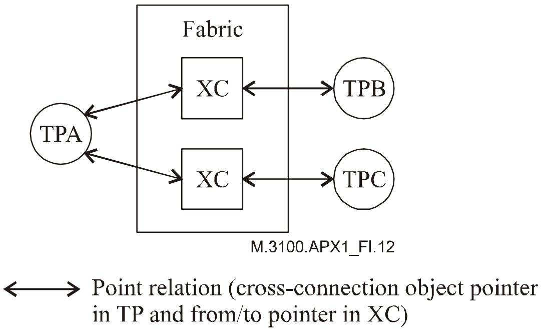

Fabric
XC
XC
TPA
TPB
TPC
M.3100.APX1_FI.12
Point relation (cross-connection object pointer in TP and from/to pointer in XC)

Figure I.12/M.3100 – Cross-connection object pointer

I.6 Object classes and logical layering

Any object class may be used in any layer of the Logical Layered Architecture (LLA).

I.7 Mandatory naming attribute

ITU-T Rec. X.720 | ISO/IEC 10165-1 imposes that naming attributes should be defined as mandatory for instantiable managed object classes. Object classes such as TTPSource/Sink/Bidirectional have the naming attribute in a conditional package. When these classes are instantiated, this results in the condition evaluating to true (making the naming attribute mandatory).

I.8 Interoperability between ITU-T Rec. M.3100 (1992) and this Recommendation

This clause is no longer applicable (text removed).

I.9 Support for multipoint trails

Multipoint trails are supported in the model by abstracting each leg as a trail. Note that in this configuration, multiple trails may share one end point.

I.10 Use of topological link

Whether to use topologicalLink or not is relevant to two kinds of Name Binding that linkConnection has. Moreover, as it can be hard to understand how to use logicalLink, the following guidelines are offered:

a) If you do not want to use topologicalLink, Name Binding between layerNetworkDomain and linkConnection is used. The pointer relationship between linkConnection and server trail is established by serverTrailList and clientLinkConnectionList attributes, respectively.

b) If you want to use topologicalLink, Name Binding between topologicalLink and linkConnection is used. The pointer relationship between linkConnection and server trail is not established. Instead, the pointer relationship between topologicalLink and server trail is established by serverTrail and clientLinkPointer attributes, respectively.

In either case, if you want to use logicalLink, the pointer relationship between linkConnection and logicalLink is established by linkConnectionPointerList of logicalLink.

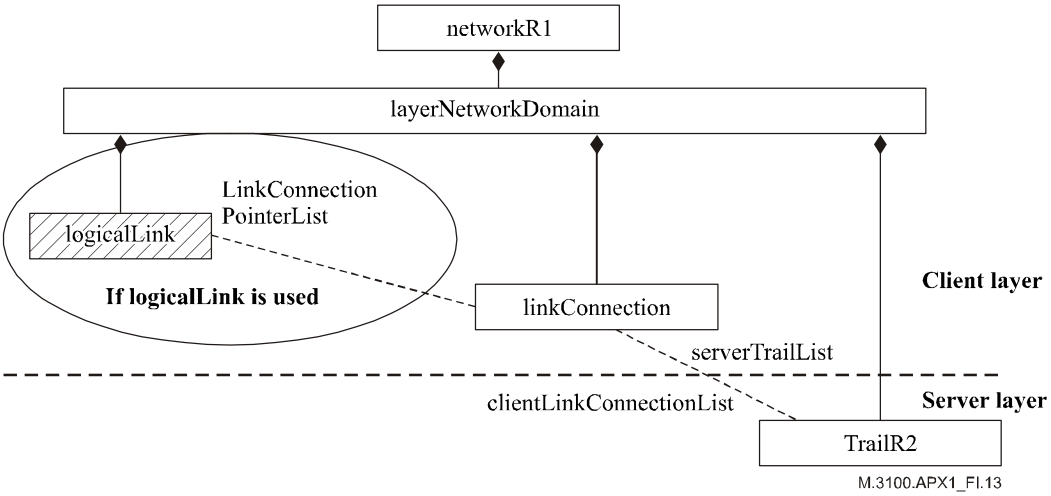


Figure I.13/M.3100 – Name binding when topologicalLink is not used

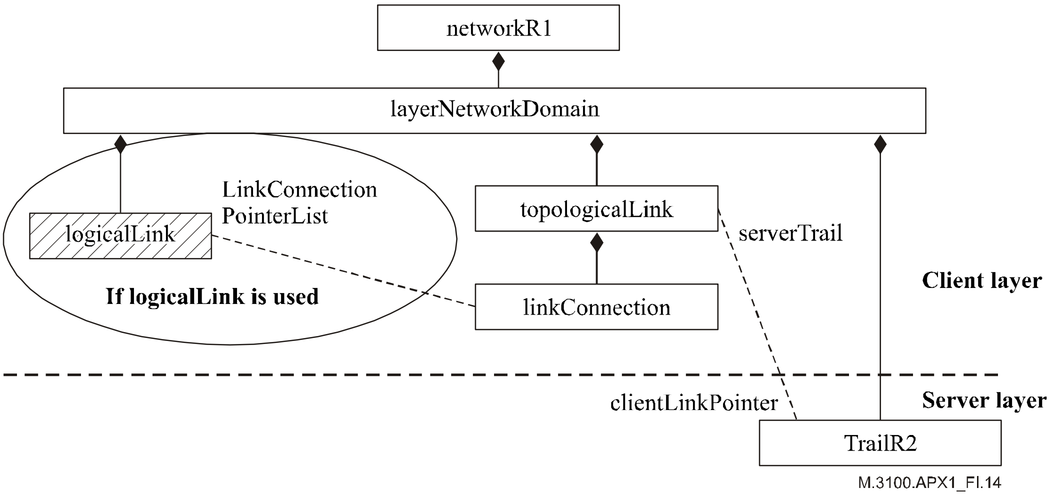


Figure I.14/M.3100 – Name binding when topologicalLink is used

I.11 System initialization

Having object creation notifications as mandatory may give the impression that when the system is set up for the first time, and the management information tree is created, the management network will be flooded with object creation notifications of any autoinstantiated objects. However, at that point, the event forwarding discriminators are either not yet created or, if created, will need a manager to configure the destination for the event reports sent. Therefore, in practice, there will not be a flood of event reports as the event forwarding discriminators are not yet fully functional.

In normal operation, after the event forwarding discriminators have been configured with appropriate destinations, the management systems set as destinations will get the object creation notifications when e.g., new equipment is added. In case there is a need to do major re-equipping to the system, then the management system may suspend the event forwarding during the equipping period, in order to avoid the flood of notifications, and then resume the event forwarding once the changes have been done.

When a system is initialized, it is expected that either the network object, the managedElement object, or the managedElementComplex object is auto-created, so that subtending objects be either auto-created, or created by the manager.

I.12 Use of equipmentHolder acceptableCircuitPackList attribute

This attribute can be used to know what circuit packs can be supported by a given equipmentHolder. When coming-up initially, the default value should be the list of all circuit packs the equipmentHolder can support. At that time, the manager can query the agent to retrieve the set of circuit packs supported by the equipmentHolder.

In cases where a software addition allows the agent to support new types of packs over the same equipmentHolder, an attribute value change notification will then be sent out.

Appendix II

User guidelines – Network topology

This non-normative appendix provides information that illustrates the use of the network topology fragment in assembling usable network level information models. The topology fragment model, while comprised of a singular set of object classes, offers a limited number of alternative relationships between the objects via optional name bindings and conditional packages. These alternatives address different modelling optimizations and, when taken together, reflect more than a single model architecture. In fact, when considered as a whole, the number of possible combinations of alternative elements could be quite large.

In order to provide guidance to the users of the topology fragment, examples that illustrate some of the more common combinations of model components are given. Each example model is internally consistent and does not exhibit the redundancies apparent in the topology fragment in its entirety.

Clause II.1 discusses general design aspects concerning inter-layer relationships. Clause II.2 describes aspects of intra-layer topology. Clauses II.3 and II.4 give two different example assemblies of model components.

II.1 Inter-layer relationship alternatives

The aggregation of object classes, that may have numerous instances, such as termination points, into containers or pools and higher level aggregates, is needed for both inter-layer relationships (representing adaptation functions) and intra-layer relationships (i.e., for subnetwork topology). For both types of aggregation, alternative approaches are supported.

Figure I.1 shows a view of a set of basic resource entities that demonstrates inter-layer relationships between server layer networkTTPs and client layer topological components. These client layer components include networkCTP, topological link end and subnetwork. In the context of Figure I.1, the networkTTP is in one layerNetworkDomain (server) and the remaining components are in another layerNetworkDomain (client). In this view, two basic options are indicated for relating network termination points to client layer components:

A pointer relation to topologicalLinkEnd and naming relation of topologicalLinkEnd to layerNetworkDomain. Naming relationships are used to bind networkCTP to topologicalLinkEnd and thence to layerNetworkDomain;

B pointer relation to client layer networkCTP, naming relation of networkCTP to subnetwork, and naming relation of subnetwork to layerNetworkDomain.

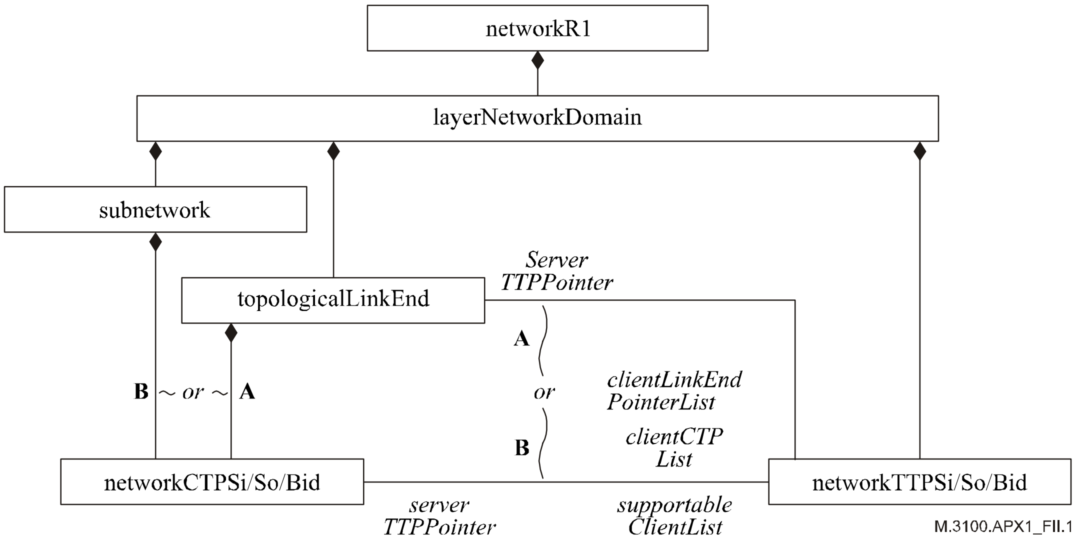


Figure II.1/M.3100 – Alternative entity-relationships for inter-layer associations

As indicated by the *or* conditions, a given implementation might use only those relationships marked as "A" or those marked as "B," without mixing elements of each.

II.2 Intra-layer topology alternatives

Aggregation within a given layer topology can be done using a hierarchical scheme. This scheme has two levels. The first level of aggregation associates termination points with link ends or access groups. The second level associates these structures with larger structures, i.e., subnetworks.

Alternatively, the termination points may be associated with subnetworks directly, and pools formed by grouping sets of termination points. These approaches are included as part of the two following example models.

II.3 Example #1

An entity-relationship diagram for the first example model is shown in Figure II.2. GDMO name bindings are indicated by lines with diamond-shaped tips. Other types of aggregation or association relationships are indicated by plain lines. Pointer attribute names are indicated by italicized text next to the object classes with which the attributes are associated. Inter-layer aggregation uses the scenario described as "A" above. Within a given layer network domain, network termination points are aggregated by either topologicalLinkEnd or accessGroup objects. Two-way pointers associate subnetwork objects with topologicalLinkEnd and accessGroup objects. In this example, only the *topological* subclasses of abstractLink and abstractLinkEnd objects have been used for simplicity. A topologicalLink joins subnetworks together via topologicalLinkEnd objects.

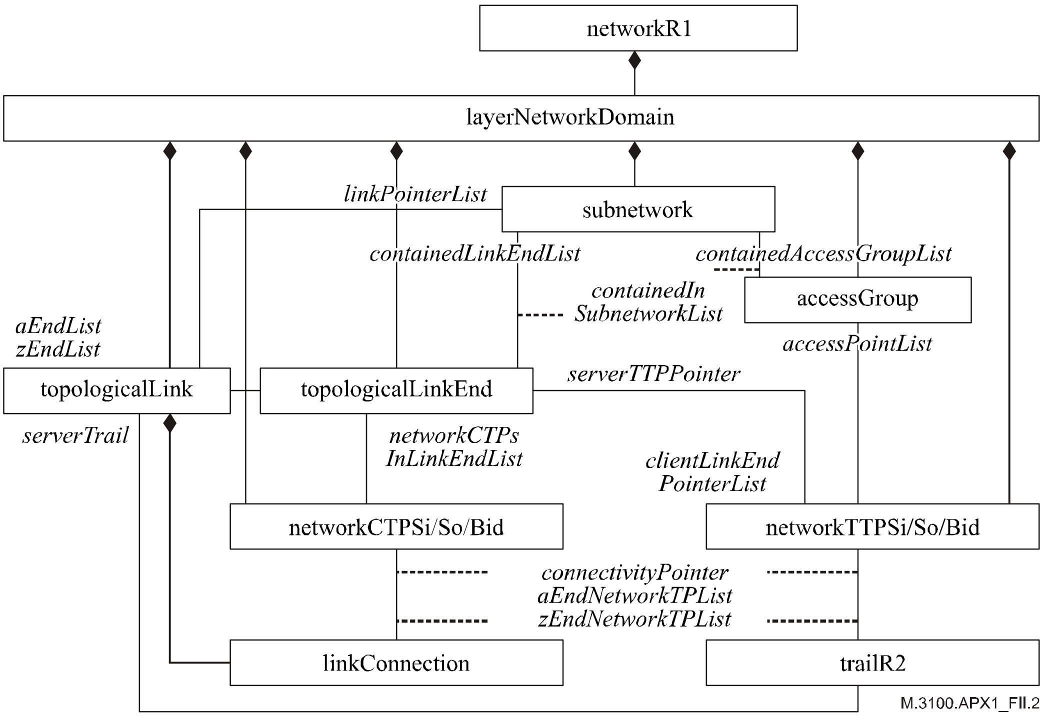


Figure II.2/M.3100 – First example of entity – relationship diagram

II.4 Example #2

In the second example assembly, inter-layer aggregation uses the scenario described as "B" above. Within a given layer network domain, network termination points are bound to a given subnetwork via GDMO name bindings. In this case, name bindings to subnetwork apply to only one level of partitioning (usually the lowest); pointers may be used to relate higher levels of partitioning (not shown). Termination points may be aggregated into either topologicalLinkEnd or accessGroup objects, but not for the same purpose as in example #1, i.e., not to associate to subnetwork objects.

In this example, both subclasses of abstractLink and abstractLinkEnd objects are used. Either topologicalLink or logicalLink objects join subnetwork objects together without involving subclasses of abstractLinkEnd. The abstractLinkEnd subclass objects are used to provide a topological point of view of links that join together different administrative domains defined by different instances of networkR1. These links are not modelled directly (point of view used also in example #1).

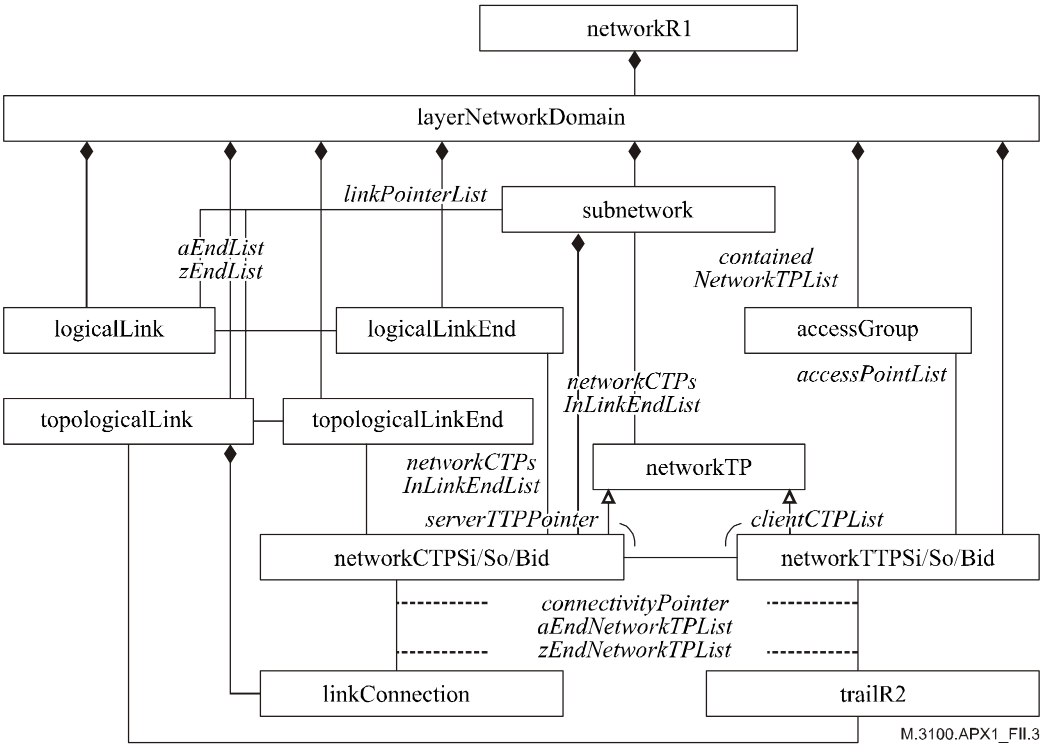


Figure II.3/M.3100 – Second example of entity – relationship diagram

Connectivity object classes are for the most part similar in the two examples. The linkConnection object joins networkCTP subclasses; trail joins networkTTP subclasses. Subnetwork connections may be established between networkTP objects.

Appendix III

Alarm Report Control (ARC) – Additional considerations

The following items have been noted for further study. Resolutions to these study points may be addressed in future revisions of this Recommendation.

III.1 Business requirement considerations

1) There is no indication in the PM objects that its threshold crossing alerts are being suppressed.
2) It is not known whether or not this is a problem, but this model does not discriminate between security alarms and any other type of alarm. Additional restrictions with regard to security alarms may be applied at some future time.
3) Some further study is needed with regard to performance monitoring gauge behaviours during periods of Alarm Reporting Control.

III.2 GDMO/ASN.1 design considerations

1) In order for this feature to become widely used, the core classes in this Recommendation (e.g., termination point objects, equipment objects) will need to be updated.

2) The arcManagementRequestedInterval, arcDefaultPersistenceInterval, and arcDefaultTimedInterval attributes manifest strange behavior in that this attribute can only be modified in certain ARC states. Similarly, the interval override in the arcControl action can only be specified with certain ARC states and is not allowed for others.

3) Additional parameter definitions for errors on actions need to be specified.

4) The current conformance reflects only the need to include MOCS. Whether one or more service definitions and functional units are required for the ARC feature needs further study.

5) The arcState behaviour may be overcomplicated by the fact that the manager is not allowed to change between the "NALM-QI" and the "NALM-TI" state. Of course, allowing these transitions is also complicated because of the semantic differences in meaning of the persistence and timed intervals.

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, 2005