



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.854.3

(03/99)

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

Digital transmission systems – Digital networks –
Management of transport network

Computational viewpoint for topology management

ITU-T Recommendation G.854.3

(Previously CCITT Recommendation)

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
INTERNATIONAL ANALOGUE CARRIER SYSTEM	
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TESTING EQUIPMENTS	
TRANSMISSION MEDIA CHARACTERISTICS	G.600–G.699
DIGITAL TRANSMISSION SYSTEMS	
TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
General aspects	G.800–G.809
Design objectives for digital networks	G.810–G.819
Quality and availability targets	G.820–G.829
Network capabilities and functions	G.830–G.839
SDH network characteristics	G.840–G.849
Management of transport network	G.850–G.859
SDH radio and satellite systems integration	G.860–G.869
Optical transport networks	G.870–G.879
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999

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

COMPUTATIONAL VIEWPOINT FOR TOPOLOGY MANAGEMENT

Summary

The topology management community is used to managing the topology of a layer network domain and the relationships between the resources of the layer network domain being managed. The service provided by the community allows for the creation and deletion of the following resources inside a layer network domain: subnetwork, link, topological link, link end, topological link end and access group. The service also provides a set of reporting actions to advise potential notification receivers of creation and deletion of resources in the community. The service is available between one single caller and one single provider.

The partitioning of subnetworks and links is not addressed in this community.

Enterprise community actions covered by this specification:

- create access group;
- delete access group;
- create link;
- delete link;
- create link end;
- delete link end;
- create subnetwork;
- delete subnetwork;
- create topological link;
- delete topological link;
- create topological link end;
- delete topological link end;
- report access group creation;
- report access group deletion;
- report link creation;
- report link deletion;
- report link end creation;
- report link end deletion;
- report subnetwork creation;
- report subnetwork deletion;
- report topological link creation;
- report topological link deletion;
- report topological link end creation;
- report topological link end deletion.

Source

ITU-T Recommendation G.854.3 was prepared by ITU-T Study Group 4 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 26th of March 1999.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The 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 Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 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 term *recognized operating agency (ROA)* includes any individual, company, corporation or governmental organization that operates a public correspondence service. The terms *Administration*, *ROA* and *public correspondence* are defined in the *Constitution of the ITU (Geneva, 1992)*.

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The 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, the 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.

© ITU 1999

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

CONTENTS

	Page
1 Scope	1
2 References	1
3 Definitions	1
4 Abbreviations	1
5 Conventions.....	2
6 Label references.....	2
7 Interfaces	3
7.1 Query interfaces.....	3
7.2 Operational interfaces.....	6
7.2.1 Common resource interface	6
7.2.2 Topology resources provisioning interface.....	7
7.3 Report interfaces.....	20
7.3.1 Common report resourceId change interface	20
7.3.2 Topology resources reporting interface.....	21
7.4 ASN.1 supporting productions	30

Recommendation G.854.3

COMPUTATIONAL VIEWPOINT FOR TOPOLOGY MANAGEMENT

(Geneva, 1999)

1 Scope

This computational viewpoint specification is related to the topology management enterprise specification defined in Recommendation G.852.3 and the topology management information specification defined in Recommendation G.853.3.

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; all 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.

- [1] ITU-T Recommendation G.851.1 (1996), *Management of the transport network – Application of the RM-ODP framework.*
- [2] ITU-T Recommendation G.853.1 (1999), *Common elements of the information viewpoint for the management of a transport network.*
- [3] ITU-T Recommendation G.852.3 (1999), *Enterprise viewpoint for topology management.*
- [4] ITU-T Recommendation G.853.3 (1999), *Information viewpoint for topology management.*

3 Definitions

None.

4 Abbreviations

This Recommendation uses the following abbreviations:

AG	Access Group
ASN.1	Abstract Syntax Notation One
CTP	Connection Termination Point
Id	Identifier
Ifce	Interface
inv	invariant
ITU	International Telecommunication Union
layerND	layer Network Domain
LE	Link End
LND	Layer Network Domain

NTTP	Network Trail Termination Point
RM-ODP	Reference Model for Open Distributed Processing
SN	Subnetwork
SNC	Subnetwork Connection
SNTP	Subnetwork Termination Point
TL	Topological Link
TLE	Topological Link End
topman	topology management
TP	Termination Point
TTP	Trail Termination Point

5 Conventions

In order to increase the readability of the behaviour in the operations:

- parameters are written in **bold**;
- elements defined in the information specification are written in *italic*.

6 Label references

Full label reference	Local label reference
<"Rec. G.853.1", INFORMATION_OBJECT:linkConnection>	linkConnection
<"Rec. G.853.1", INFORMATION_OBJECT:networkCTP>	networkCTP
<"Rec. G.853.1", INFORMATION_OBJECT:subnetworkConnection>	subnetworkConnection
<"Rec. G.853.3", INFORMATION_OBJECT: topmanAccessGroup>	topmanAccessGroup
<"Rec. G.853.3", INFORMATION_OBJECT: topmanLayerNetworkDomain>	topmanLayerNetworkDomain
<"Rec. G.853.3", INFORMATION_OBJECT: topmanLink>	topmanLink
<"Rec. G.853.3", INFORMATION_OBJECT: topmanLinkEnd>	topmanLinkEnd
<"Rec. G.853.3", INFORMATION_OBJECT: topmanNetworkTTP>	topmanNetworkTTP
<"Rec. G.853.3", INFORMATION_OBJECT: topmanSubnetwork>	topmanSubnetwork
<"Rec. G.853.3", INFORMATION_OBJECT: topmanSubnetworkTP>	topmanSubnetworkTP
<"Rec. G.853.3", INFORMATION_OBJECT: topmanTopologicalLink>	topmanTopologicalLink
<"Rec. G.853.3", INFORMATION_OBJECT: topmanTopologicalLinkEnd>	topmanTopologicalLinkEnd
<"Rec. G.853.1", ATTRIBUTE: linkDirectionality>	linkDirectionality
<"Rec. G.853.1", ATTRIBUTE: pointDirectionality>	pointDirectionality
<"Rec. G.853.1", ATTRIBUTE: resourceId>	resourceId
<"Rec. G.853.1", ATTRIBUTE: signalIdentification>	signalIdentification
<"Rec. G.853.1", ATTRIBUTE: topologicalEndDirection>	topologicalEndDirection
<"Rec. G.853.1", ATTRIBUTE: userLabel>	userLabel
<"Rec. G.853.1", INFORMATION_RELATIONSHIP: accessGroupIsMadeOfNetworkTTPs>	accessGroupIsMadeOfNetworkTTPs

Full label reference	Local label reference
<"Rec. G.853.1", INFORMATION_RELATIONSHIP: layerNetworkDomainIsMadeOf>	layerNetworkDomainIsMadeOf
<"Rec. G.853.1", INFORMATION_RELATIONSHIP: linkBinds>	linkBinds
<"Rec. G.853.1", INFORMATION_RELATIONSHIP: linkConnectionIsTerminatedByTopologicalEntities>	linkConnectionIsTerminatedByTopologicalEntities
<"Rec. G.853.1", INFORMATION_RELATIONSHIP: linkEndHasNetworkCTPs>	linkEndHasNetworkCTPs
<"Rec. G.853.1", INFORMATION_RELATIONSHIP: linkEndIsBoundTo>	linkEndIsBoundTo
<"Rec. G.853.1", INFORMATION_RELATIONSHIP: linkHasLinkConnections>	linkHasLinkConnections
<"Rec. G.853.1", INFORMATION_RELATIONSHIP: subnetworkHasSubnetworkConnections>	subnetworkHasSubnetworkConnections
<"Rec. G.853.1", INFORMATION_RELATIONSHIP: subnetworkIsDelimitedBy>	subnetworkIsDelimitedBy
<"Rec. G.853.1", INFORMATION_RELATIONSHIP: subnetworkTPIsRelatedToExtremity>	subnetworkTPIsRelatedToExtremity
<"Rec. G.853.1", INFORMATION_RELATIONSHIP: topologicalLinkEndIsSupportedByNetworkTTP>	topologicalLinkEndIsSupportedByNetworkTTP
<"Rec. G.853.1", INFORMATION_RELATIONSHIP: topologicalLinkIsSupportedByTrail>	topologicalLinkIsSupportedByTrail

Full ASN.1 production reference	Local label reference
<"Rec. X.721:1992: Attribute-ASN1Module" : SimpleNameType>	SimpleNameType
<"Rec. X.680:1997": GraphicString>	GraphicString

7 Interfaces

This specification does not define operations which change the userLabel of the resources or report this change to the notification receiver because they do not change the state of the system. Their exact signature will be developed as part of the engineering viewpoint, with the concerned technology.

7.1 Query interfaces

This specification refers to interfaces that allow get access to identification and properties of resources involved in the "topology management" community. As the invocation of contained operations do not modify any state, there is no interest to develop them explicitly. Their exact signature will be developed as part of the engineering viewpoint, with the concerned technology. These interfaces are listed in the following table with the information they allow access to.

Interface name	Information object	Attributes and relationships
topmanAccessGroupQueryIfce	<topmanAccessGroup>	<resourceId> <signalIdentification> <topologicalEndDirection> <userLabel> <accessGroupIsMadeOfNetworkTTPs, ROLE: elementTTP> <linkBinds, ROLE: transferCapacityLink> <layerNetworkDomainIsMadeOf, ROLE: containerLND>
topmanLayerNetworkDomainQueryIfce	<topmanLayerNetworkDomain>	<resourceId> <signalIdentification> <layerNetworkDomainIsMadeOf, ROLE: element>
topmanLinkEndQueryIfce	<topmanLinkEnd>	<resourceId> <signalIdentification> <topologicalEndDirection> <userLabel> <layerNetworkDomainIsMadeOf, ROLE: containerLND> <linkEndIsBoundTo, ROLE: topologicalEntity> <linkEndHasNetworkCTPs, ROLE: elementCTP>
topmanLinkQueryIfce	<topmanLink>	<resourceId> <signalIdentification> <linkDirectionality> <userLabel> <linkBinds, ROLE: a_endTopological, ROLE: z_endTopological> <layerNetworkDomainIsMadeOf, ROLE: containerLND> <linkHasLinkConnections, ROLE: elementLC>
topmanNetworkTTPQueryIfce	<topmanNetworkTTP>	<resourceId> <signalIdentification> <subnetworkTTPsRelatedToExtremity, ROLE: abstractionSNTP> <layerNetworkDomainIsMadeOf, ROLE: containerLND> <accessGroupIsMadeOfNetworkTTPs, ROLE: containerAG>

Interface name	Information object	Attributes and relationships
topmanSubnetworkQueryIfce	<topmanSubnetwork>	<resourceId> <signalIdentification> <userLabel> <linkBinds, ROLE: transferCapacityLink> <subnetworkIsDelimitedBy, ROLE: elementSNTP> <linkEndIsBoundTo, ROLE: transferCapacityLE> <subnetworkHasSubnetworkConnections, ROLE: elementSNC> <linkConnectionIsTerminatedByTopological Entities, ROLE: transportEntityLC> <layerNetworkDomainIsMadeOf, ROLE: containerLND>
topmanSubnetworkTPQueryIfce	<topmanSubnetworkTP>	<resourceId> <signalIdentification> <pointDirectionality> <subnetworkIsDelimitedBy, ROLE: containerSN> <subnetworkTPIsRelatedToExtremity, ROLE: extremity> <layerNetworkDomainIsMadeOf, ROLE: containerLND>
topmanTopologicalLinkEndQueryIfce	<topmanTopologicalLinkEnd>	<resourceId> <signalIdentification> <topologicalEndDirection> <userLabel> <layerNetworkDomainIsMadeOf, ROLE: containerLND> <linkEndIsBoundTo, ROLE: topologicalEntity> <topologicalLinkEndIsSupportedByNetwork TTP, ROLE: serverTTP>
topmanTopologicalLinkQueryIfce	<topmanTopologicalLink>	<resourceId> <signalIdentification> <linkDirectionality> <userLabel> <linkBinds, ROLE: a_endTopological, ROLE: z_endTopological> <layerNetworkDomainIsMadeOf, ROLE: containerLND> <topologicalLinkIsSupportedByTrail, ROLE: serverTrail>

7.2 Operational interfaces

7.2.1 Common resource interface

The common resource interface provides functionality for changing the resource identifier of the resources by using the operation "change resource identifier". It satisfies the enterprise requirements stated in <"Rec. G.852.2", COMMUNITY: Transport Network Enterprise Model, General policy>.

COMPUTATIONAL_INTERFACE commonResourceIfce {

 OPERATION <changeResourceIdentifier>; }

7.2.1.1 Change resource identifier

<"Rec. G.852.2", COMMUNITY: Transport Network Enterprise Model, General policy>

OPERATION changeResourceId {

 INPUT_PARAMETERS

 layerND: LayerNetworkDomainId ::= TopmanLayerNetworkDomainQueryIfce;
 involvedResource: ResourceChoice
 newResourceIdentifier: UserIdentifier ::= SimpleNameType;

 OUTPUT_PARAMETERS

 agreedResourceIdentifier: UserIdentifier ::= SimpleNameType;

 RAISED_EXCEPTIONS

 invalidResource: NULL;
 newResourceIdentifierNotUnique: newResourceIdentifier;
 failureToChangeId: NULL;

BEHAVIOUR

SEMI_FORMAL

PARAMETER_MATCHING

 layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
 involvedResource: <INFORMATION OBJECT networkInformationTop>;
 newResourceIdentifier: <INFORMATION ATTRIBUTE: resourceId>;
 agreedResourceIdentifier: <INFORMATION ATTRIBUTE: resourceId>;

PRE_CONDITIONS

 inv_existingInvolvedResource

 "**involvedResource** shall refer to *element* in the same <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

 inv_uniqueResourceId

 "**newResourceIdentifier** must not be equal to <resourceId> of any *element* in the same <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

POST_CONDITIONS

 inv_idChanged

 "<resourceId> of the resource referred to by **involvedResource** has to be equal to **newResourceIdentifier** and **agreedResourceIdentifier**.";

EXCEPTIONS

 IF PRE_CONDITION inv_existingInvolvedResource NOT_VERIFIED RAISE_EXCEPTION
 invalidResource;

 IF PRE_CONDITION inv_uniqueResourceId NOT_VERIFIED RAISE_EXCEPTION
 newResourceIdentifierNotUnique;

```

        IF POST_CONDITION inv_idChanged NOT_VERIFIED RAISE_EXCEPTION failureToChangeId;
    ;}

```

7.2.2 Topology resources provisioning interface

The Topology Resources Provisioning Interface provides functionality for the provisioning of the resources defined in <"Rec. G.852.2", COMMUNITY: Transport Network Enterprise Model>.

```

COMPUTATIONAL_INTERFACE topologyResourcesProvisioningIfce {
    OPERATION
        <createAccessGroup>;
        <deleteAccessGroup>;
        <createLink>;
        <deleteLink>;
        <createLinkEnd>;
        <deleteLinkEnd>;
        <createSubnetwork>;
        <deleteSubnetwork>;
        <createTopologicalLink>;
        <deleteTopologicalLink>;
        <createTopologicalLinkEnd>;
        <deleteTopologicalLinkEnd>; }

```

7.2.2.1 Create access group

<COMMUNITY: Topology Management, ACTION: create access group>

```

OPERATION createAccessGroup {
    INPUT_PARAMETERS
        layerND: LayerNetworkDomainId;
        suppliedUserIdentifier: UserIdentifier;
            -- zero length string or 0 implies none supplied.
        suppliedUserLabel: GraphicString;
            -- zero length string implies none supplied.
        suppliedDirection: topologicalEndDirection;

```

```

    OUTPUT_PARAMETERS
        accessGroup: AccessGroupId;

```

```

    RAISED_EXCEPTIONS
        userIdentifierNotUnique: suppliedUserIdentifier;
        failureToSetUserIdentifier: NULL;
        failureToSetUserLabel: NULL;
        failureToCreateAccessGroup: NULL;
        failureToSetDirection: NULL;

```

BEHAVIOUR

SEMI_FORMAL

```

    PARAMETER_MATCHING
        layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
        suppliedUserIdentifier: <INFORMATION ATTRIBUTE: resourceId>;
        suppliedUserLabel: <INFORMATION ATTRIBUTE: userLabel>;
        accessGroup: <INFORMATION OBJECT: topmanAccessGroup>;
        suppliedDirection: <INFORMATION ATTRIBUTE: topologicalEndDirection>;

```

PRE_CONDITIONS

```

inv_uniqueUserIdentifier
    "The suppliedUserIdentifier value shall not be equal to the <resourceId> value of any element in the
    <layerNetworkDomainIsMadeOf> relationship where the layerND refers to containerLND."
```

POST_CONDITIONS

inv_agreedUserIdentifier

"The <resourceId> value of <topmanAccessGroup> referred to by **accessGroup** is equal to the **suppliedUserIdentifier** value, if it is supplied."

inv_agreedUserLabel

"The <userLabel> value of <topmanAccessGroup> referred to by **accessGroup** is equal to the **suppliedUserLabel** value, if it is supplied."

inv_existingAccessGroup

"**accessGroup** refers to *element* in the <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

inv_agreedDirection

"The <topologicalEndDirection> value of <topmanAccessGroup> referred to by **accessGroup** is equal to the **suppliedDirection** value, if it is supplied, or set to 'undefined' if not supplied."

EXCEPTIONS

IF PRE_CONDITION inv_uniqueUserIdentifier NOT_VERIFIED RAISE_EXCEPTION
userIdentifierNotUnique;

IF POST_CONDITION inv_agreedUserIdentifier NOT_VERIFIED RAISE_EXCEPTION
failureToSetUserIdentifier;

IF POST_CONDITION inv_agreedUserLabel NOT_VERIFIED RAISE_EXCEPTION
failureToSetUserLabel;

IF POST_CONDITION inv_existingAccessGroup NOT_VERIFIED RAISE_EXCEPTION
failureToCreateAccessGroup;

IF POST_CONDITION inv_agreedDirection NOT_VERIFIED RAISE_EXCEPTION
failureToSetDirection;

;

7.2.2.2 Delete access group

<COMMUNITY: Topology Management, ACTION: delete access group>

OPERATION deleteAccessGroup {

INPUT_PARAMETERS

layerND: LayerNetworkDomainId;
accessGroup: AccessGroupId;

OUTPUT_PARAMETERS

-- none

RAISED_EXCEPTIONS

incorrectAccessGroup: accessGroup;
networkTTPsExisting: NULL;
failureToDeleteAccessGroup: NULL;

BEHAVIOUR

SEMI_FORMAL

PARAMETER_MATCHING

layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
accessGroup: <INFORMATION OBJECT: topmanAccessGroup>;

PRE_CONDITIONS

inv_existingAccessGroup

"**accessGroup** refers to *element* in the <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

inv_noNetworkTTPs

"**accessGroup** shall not refer to any *containerAG* in a <accessGroupIsMadeOfNetworkTTPs> relationship."

POST_CONDITIONS
 inv_noAccessGroup
 "**accessGroup** does not refer to any *element* in the <layerNetworkDomainIsMadeOf> relationship."

EXCEPTIONS

IF PRE_CONDITION inv_existingAccessGroup NOT_VERIFIED RAISE_EXCEPTION
 incorrectAccessGroup;
 IF PRE_CONDITION inv_noNetworkTTPs NOT_VERIFIED RAISE_EXCEPTION networkTTPsExisting;
 IF POST_CONDITION inv_noAccessGroup NOT_VERIFIED RAISE_EXCEPTION
 failureToDeleteAccessGroup;

;}

7.2.2.3 Create link

<COMMUNITY: Topology Management, ACTION: create link>

OPERATION createLink {

INPUT_PARAMETERS

 layerND: LayerNetworkDomainId;
 a_End: LinkEndChoice;
 z_End: LinkEndChoice;
 suppliedUserIdentifier: UserIdentifier;
 -- zero length string or 0 implies none supplied.
 suppliedUserLabel: GraphicString;
 -- zero length string implies none supplied.
 suppliedDirectionality: linkDirectionality;

OUTPUT_PARAMETERS

 link: LinkId;

RAISED_EXCEPTIONS

 incorrectLinkEnds: LinkEndChoice;
 userIdentifierNotUnique: suppliedUserIdentifier;
 failureToSetUserIdentifier: NULL;
 failureToCreateLink: NULL;
 failureToBindLink: NULL;
 failureToSetUserLabel: NULL;
 failureToSetDirectionality: NULL;

BEHAVIOUR

SEMI_FORMAL

PARAMETER_MATCHING

 layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
 subnetwork: <INFORMATION OBJECT: topmanSubnetwork>;
 accessGroup: <INFORMATION OBJECT: topmanAccessGroup>;
 suppliedUserIdentifier: <INFORMATION ATTRIBUTE: resourceId>;
 suppliedUserLabel: <INFORMATION ATTRIBUTE: userLabel>;
 suppliedDirectionality: <INFORMATION ATTRIBUTE: linkDirectionality>;
 link: <INFORMATION OBJECT: topmanLink>;

PRE_CONDITIONS

inv_sameLayerND

 "**a_End** and **z_End** shall both refer to *element* in the same <layerNetworkDomainIsMadeOf> relationship
 where **layerND** refers to *containerLND*."

inv_uniqueUserIdentifier

 "The **suppliedUserIdentifier** value shall not be equal to the <resourceId> value of any *element* in the
 <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

POST_CONDITIONS

inv_existingLink

"**link**, **a_End** and **z_End** refer to *element* in the <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

inv_linkBinds

"**a_End** and **z_End** respectively refer to *a_endTopological* and *z_endTopological* in the <linkBinds> relationship where **link** refers to *transferCapacityLink*."

inv_agreedUserIdentifier

"The <resourceId> value of <topmanLink> referred to by **link** is equal to the **suppliedUserIdentifier** value, if it is supplied."

inv_agreedUserLabel

"The <userLabel> value of <topmanLink> referred to by **link** is equal to the **suppliedUserLabel** value, if it is supplied."

"inv_agreedDirectionality

"The <linkDirectionality> value of <topmanLink> referred to by **link** is equal to the **suppliedDirectionality** value, if it is supplied, or set to 'undefined' if not supplied."

EXCEPTIONS

IF PRE_CONDITION inv_sameLayerND NOT_VERIFIED RAISE_EXCEPTION incorrectLinkEnds;

IF PRE_CONDITION inv_uniqueUserIdentifier NOT_VERIFIED RAISE_EXCEPTION
userIdentifierNotUnique;

IF POST_CONDITION inv_existingLink NOT_VERIFIED RAISE_EXCEPTION failureToCreateLink;

IF POST_CONDITION inv_linkBinds NOT_VERIFIED RAISE_EXCEPTION failureToBindLink;

IF POST_CONDITION inv_agreedUserIdentifier NOT_VERIFIED RAISE_EXCEPTION
failureToSetUserIdentifier;

IF POST_CONDITION inv_agreedUserLabel NOT_VERIFIED RAISE_EXCEPTION failureToSetUserLabel;

IF POST_CONDITION inv_agreedDirectionality NOT_VERIFIED RAISE_EXCEPTION
failureToSetDirectionality;

;}

7.2.2.4 Delete link

<COMMUNITY: Topology Management, ACTION: delete link>

OPERATION deleteLink {

INPUT_PARAMETERS

layerND: LayerNetworkDomainId;
link: LinkId;

OUTPUT_PARAMETERS

-- none

RAISED_EXCEPTIONS

incorrectLink: link;
linkConnectionExisting: NULL;
failureToDeleteLink: NULL;

BEHAVIOUR

SEMI_FORMAL

PARAMETER_MATCHING

layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
link: <INFORMATION OBJECT: topmanLink>;

```

    PRE_CONDITIONS;
    inv_existingLink
        "link refers to the element in the <layerNetworkDomainIsMadeOf> relationship where layerND refers to
        containerLND."

    inv_noLinkConnection
        "link shall not refer to any containerLink of a <linkHasLinkConnections> relationship."

    POST_CONDITIONS
    inv_noLink
        "link does not participate in any <layerNetworkDomainIsMadeOf> AND <linkBinds> relationships."

    EXCEPTIONS
    IF PRE_CONDITION inv_existingLink NOT_VERIFIED RAISE_EXCEPTION incorrectLink;
    IF PRE_CONDITION inv_noLinkConnection NOT_VERIFIED RAISE_EXCEPTION
        linkConnectionExisting;
    IF POST_CONDITION inv_noLink NOT_VERIFIED RAISE_EXCEPTION failureToDeleteLink;
;}

```

7.2.2.5 Create link end

This operation creates a link end bound to a subnetwork.

<COMMUNITY: Topology Management, ACTION: create link end>

```

OPERATION createLinkEnd {
    INPUT_PARAMETERS
        layerND: LayerNetworkDomainId;
        subnetwork: SubnetworkId;
        suppliedUserIdentifier: UserIdentifier;
            -- zero length string or 0 implies none supplied.
        suppliedUserLabel: GraphicString;
            -- zero length string implies none supplied.
        suppliedDirection: topologicalEndDirection;

    OUTPUT_PARAMETERS
        linkEnd: LinkEndId;

    RAISED_EXCEPTIONS
        incorrectSubnetwork: subnetwork;
        userIdentifierNotUnique: suppliedUserIdentifier;
        failureToCreateLinkEnd: NULL;
        failureToBindLinkEnd: NULL;
        failureToSetUserIdentifier: NULL;
        failureToSetUserLabel: NULL;
        failureToSetDirection: NULL;
}

```

BEHAVIOUR

SEMI_FORMAL

```

PARAMETER_MATCHING
    layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
    subnetwork: <INFORMATION OBJECT: topmanSubnetwork>;
    suppliedUserIdentifier: <INFORMATION ATTRIBUTE: resourceId>;
    suppliedUserLabel: <INFORMATION ATTRIBUTE: userLabel>;
    suppliedDirection: <INFORMATION ATTRIBUTE: topologicalEndDirection>;
    linkEnd: <INFORMATION OBJECT: topmanLinkEnd>;

```

```

PRE_CONDITIONS
  inv_subnetworkExisting
    "subnetwork shall refer to element in the <layerNetworkDomainIsMadeOf> relationship where layerND
    refers to containerLND."

  inv_uniqueUserIdentifier
    "The suppliedUserIdentifier value shall not be equal to the <resourceId> value of any element in the
    <layerNetworkDomainIsMadeOf> relationship where layerND refers to containerLND.";

POST_CONDITIONS
  inv_existingLinkEnd
    "linkEnd and subnetwork refer to element in the same <layerNetworkDomainIsMadeOf> relationship
    where layerND refers to containerLND."

  inv_boundLinkEnd
    "linkEnd refers to transferCapacityLE in the <linkEndIsBoundTo> relationship where subnetwork refers
    to topologicalEntity."

  inv_agreedUserIdentifier
    "The <resourceId> value of <topmanLinkEnd> referred to by linkEnd is equal to the
    suppliedUserIdentifier value, if it is supplied."

  inv_agreedUserLabel
    "The <userLabel> value of <topmanLinkEnd> referred to by linkEnd is equal to the suppliedUserLabel
    value, if it is supplied."

  inv_agreedDirection
    "The <topologicalEndDirection> value of <topmanLinkEnd> referred to by linkEnd is equal to the
    suppliedDirection value, if it is supplied, or set to 'undefined' if not supplied."

EXCEPTIONS
  IF PRE_CONDITION inv_subnetworkExisting NOT_VERIFIED RAISE_EXCEPTION
    incorrectSubnetwork;
  IF PRE_CONDITION inv_uniqueUserIdentifier NOT_VERIFIED RAISE_EXCEPTION
    userIdentifierNotUnique;
  IF POST_CONDITION inv_existingLinkEnd NOT_VERIFIED RAISE_EXCEPTION
    failureToCreateLinkEnd;
  IF POST_CONDITION inv_boundLinkEnd NOT_VERIFIED RAISE_EXCEPTION failureToBindLinkEnd;
  IF POST_CONDITION inv_agreedUserIdentifier NOT_VERIFIED RAISE_EXCEPTION
    failureToSetUserIdentifier;
  IF POST_CONDITION inv_agreedUserLabel NOT_VERIFIED RAISE_EXCEPTION
    failureToSetUserLabel;
  IF POST_CONDITION inv_agreedDirection NOT_VERIFIED RAISE_EXCEPTION failureToSetDirection;
;}
```

7.2.2.6 Delete link end

<COMMUNITY: Topology Management, ACTION: delete link end>

```

OPERATION deleteLinkEnd {
  INPUT_PARAMETERS
    layerND: LayerNetworkDomainId;
    linkEnd: LinkEndId;

  OUTPUT_PARAMETERS
    -- none

  RAISED_EXCEPTIONS
    incorrectLinkEnd: linkEnd;
    networkCTPExisting: NULL;
    failureToDeleteLinkEnd: NULL;
```

BEHAVIOUR

SEMI_FORMAL

PARAMETER_MATCHING

layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;

linkEnd: <INFORMATION OBJECT: topmanLinkEnd>;

PRE_CONDITIONS

inv_existingLinkEnd

"**linkEnd** refers to *element* in the <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

inv_noNetworkCTPs

"**linkEnd** shall not refer to any *containerLE* in a <linkEndHasNetworkCTPs> relationship."

POST_CONDITIONS

inv_noLinkEnd

"**linkEnd** does not refer to any *element* in the <layerNetworkDomainIsMadeOf> relationship."

EXCEPTIONS

IF PRE_CONDITION inv_existingLinkEnd NOT_VERIFIED RAISE_EXCEPTION incorrectLinkEnd;

IF PRE_CONDITION inv_noNetworkCTPs NOT_VERIFIED RAISE_EXCEPTION networkCTPExisting;

IF POST_CONDITION inv_noLinkEnd NOT_VERIFIED RAISE_EXCEPTION failureToDeleteLinkEnd;

;}

7.2.2.7 Create subnetwork

<COMMUNITY: Topology Management, ACTION: create subnetwork>

OPERATION createSubnetwork {

INPUT_PARAMETERS

layerND: LayerNetworkDomainId;

suppliedUserIdentifier: UserIdentifier;

-- zero length string or 0 implies none supplied.

suppliedUserLabel: GraphicString;

-- zero length string implies none supplied.

OUTPUT_PARAMETERS

subnetwork: SubnetworkId;

networkTTPs: SetOfNetworkTTPs;

-- the networkTTPs parameter value is provided when <PERMISSION: successReturnTTPs> is

-- supported.

RAISED_EXCEPTIONS

userIdentifierNotUnique: suppliedUserIdentifier;

failureToSetUserIdentifier: NULL;

failureToSetUserLabel: NULL;

failureToCreateSubnetwork: NULL;

failureToCreateNTTP: NULL;

failureToAssociateNTTP: NULL;

BEHAVIOUR

SEMI_FORMAL

PARAMETER_MATCHING

layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;

suppliedUserIdentifier: <INFORMATION ATTRIBUTE: resourceId>;

suppliedUserLabel: <INFORMATION ATTRIBUTE: userLabel>;

subnetwork: <INFORMATION OBJECT: topmanSubnetwork>;

networkTTPs ELEMENTS: <INFORMATION OBJECT: topmanNetworkTTP>;

PRE_CONDITIONS

inv_uniqueUserIdentifier

"The **suppliedUserIdentifier** value shall not be equal to the *<resourceId>* value of any *element* in the *<layerNetworkDomainIsMadeOf>* relationship where **layerND** refers to *containerLND*."

POST_CONDITIONS

inv_agreedUserIdentifier

"The *<resourceId>* value of the *<topmanSubnetwork>* referred to by **subnetwork** is equal to the **suppliedUserIdentifier** value, if it is supplied."

inv_agreedUserLabel

"The *<userLabel>* value of the *<topmanSubnetwork>* referred to by **subnetwork** is equal to the **suppliedUserIdentifier** value, if it is supplied."

inv_existingSubnetwork

"**subnetwork** refers to *element* in the *<layerNetworkDomainIsMadeOf>* relationship where **layerND** refers to *containerLND*."

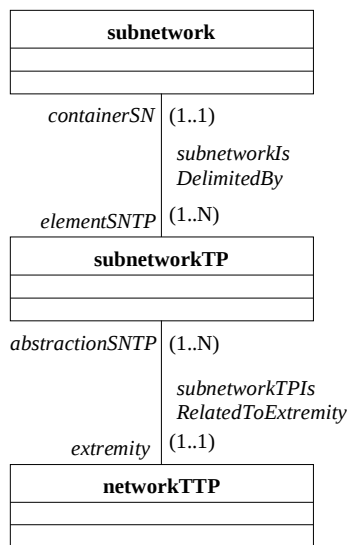
-- NOTE – The following invariants are only needed if the *networkTTPs* appear automatically when the *subnetwork* is created.

inv_existingNetworkTTP

"**networkTTPs** refers to *element* in the *<layerNetworkDomainIsMadeOf>* relationship where **layerND** refers to *containerLND*."

inv_nttpAssociated

"**networkTTPs** refers to *extremity* of the *<subnetworkTTPsRelatedToExtremity>* relationship where the *abstractionSNTP* (which is a *subnetworkTP*) is also *elementSNTP* of a *<subnetworkIsDelimitedBy>* relationship where **subnetwork** refers to *containerSN*."



";

EXCEPTIONS

IF PRE_CONDITION inv_uniqueUserIdentifier NOT_VERIFIED RAISE_EXCEPTION

userIdentifierNotUnique;

IF POST_CONDITION inv_agreedUserIdentifier NOT_VERIFIED RAISE_EXCEPTION

failureToSetUserIdentifier;

IF POST_CONDITION inv_agreedUserLabel NOT_VERIFIED RAISE_EXCEPTION failureToSetUserLabel;

IF POST_CONDITION inv_existingSubnetwork NOT_VERIFIED RAISE_EXCEPTION

failureToCreateSubnetwork;

IF POST_CONDITION inv_existingNetworkTTP NOT_VERIFIED RAISE_EXCEPTION

failureToCreateNTTP;

```

        IF POST_CONDITION inv_nttpAssociated NOT_VERIFIED RAISE_EXCEPTION failureToAssociateNTTP;
    ;}

```

7.2.2.8 Delete subnetwork

<COMMUNITY: Topology Management, ACTION: delete subnetwork>

OPERATION deleteSubnetwork {

INPUT_PARAMETERS

layerND: LayerNetworkDomainId;
subnetwork: SubnetworkId;

OUTPUT_PARAMETERS

-- none

RAISED_EXCEPTIONS

incorrectSubnetwork: subnetwork;
subnetworkInUse: NULL;
subnetworkContainsSubnetworkConnections: NULL;
boundSubnetwork: NULL;
failureToDeleteSubnetwork: NULL;

BEHAVIOUR

SEMI_FORMAL

PARAMETER_MATCHING

layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
subnetwork: <INFORMATION OBJECT: topmanSubnetwork>;

PRE_CONDITIONS

inv_existingSubnetwork

"**subnetwork** refers to *element* in the <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

inv_noSubnetworkTP

"**subnetwork** shall not refer to any *containerSN* in a <subnetworkIsDelimitedBy> relationship."

inv_noSubnetworkConnections

"**subnetwork** shall not refer to any *containerSN* in a <subnetworkHasSubnetworkConnections> relationship."

inv_noBinding

"**subnetwork** shall not refer to any *a_endTopological* or *z_endTopological* in a <linkBinds> or <linkConnectionIsTerminatedByTopologicalEntities> relationship or to any topologicalEntity in a <linkEndIsBoundTo> relationship."

POST_CONDITIONS

inv_noSubnetwork

"**subnetwork** must not refer to any *element* in a <layerNetworkDomainIsMadeOf> relationship."

EXCEPTIONS

IF PRE_CONDITION inv_existingSubnetwork NOT_VERIFIED RAISE_EXCEPTION

incorrectSubnetwork;

IF PRE_CONDITION inv_noSubnetworkTP NOT_VERIFIED RAISE_EXCEPTION subnetworkInUse;

IF PRE_CONDITION inv_noSubnetworkConnection NOT_VERIFIED RAISE_EXCEPTION

subnetworkContainsSubnetworkConnections;

IF PRE_CONDITION inv_noBinding NOT_VERIFIED RAISE_EXCEPTION boundSubnetwork;

IF POST_CONDITION inv_noSubnetwork NOT_VERIFIED RAISE_EXCEPTION

failureToDeleteSubnetwork;

```

;}

```

7.2.2.9 Create topological link

<COMMUNITY: Topology Management, ACTION: create topological link>

OPERATION createTopologicalLink {

INPUT_PARAMETERS

layerND: LayerNetworkDomainId;
a_End: LinkEndChoice;
z_End: LinkEndChoice;
suppliedUserIdentifier: UserIdentifier;
-- zero length string or 0 implies none supplied.
suppliedUserLabel: GraphicString;
-- zero length string implies none supplied.
suppliedDirectionality: linkDirectionality;

OUTPUT_PARAMETERS

topologicalLink: TopologicalLinkId;

RAISED_EXCEPTIONS

incorrectLinkEnds: LinkEndChoice;
userIdentifierNotUnique: suppliedUserIdentifier;
failureToSetUserIdentifier: NULL;
failureToCreateTopologicalLink: NULL;
failureToBindTopologicalLink: NULL;
failureToSetUserLabel: NULL;
failureToSetDirectionality: NULL;

BEHAVIOUR

SEMI_FORMAL

PARAMETER_MATCHING

layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
subnetwork: <INFORMATION OBJECT: topmanSubnetwork>;
accessGroup: <INFORMATION OBJECT: topmanAccessGroup>;
suppliedUserIdentifier: <INFORMATION ATTRIBUTE: resourceId>;
suppliedUserLabel: <INFORMATION ATTRIBUTE: userLabel>;
suppliedDirectionality: <INFORMATION ATTRIBUTE: linkDirectionality>;
topologicalLink: <INFORMATION OBJECT: topmanLink>;

PRE_CONDITIONS

inv_sameLayerND

"**a_End** and **z_End** shall both refer to *element* in the same <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

inv_uniqueUserIdentifier

"The **suppliedUserIdentifier** value shall not be equal to the <resourceId> value of any *element* in the <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

POST_CONDITIONS

inv_existingTopologicalLink

"**link**, **a_End** and **z_End** refer to *element* in the <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

inv_linkBinds

"**a_End** and **z_End** respectively refer to *a_endTopological* and *z_endTopological* in the <linkBinds> relationship where **topologicalLink** refers to *transferCapacityLink*."

inv_agreedUserIdentifier

"The <resourceId> value of <topmanTopologicalLink> referred to by **topologicalLink** is equal to the **suppliedUserIdentifier** value, if it is supplied."

inv_agreedUserLabel

"The <userLabel> value of <topmanTopologicalLink> referred to by **topologicalLink** is equal to the **suppliedUserLabel** value, if it is supplied.

"inv_agreedDirectionality

"The <linkDirectionality> value of <topmanTopologicalLink> referred to by **topologicalLink** is equal to the **suppliedDirectionality** value, if it is supplied, or set to 'undefined' if not supplied."

EXCEPTIONS

IF PRE_CONDITION inv_sameLayerND NOT_VERIFIED RAISE_EXCEPTION incorrectLinkEnds;

IF PRE_CONDITION inv_uniqueUserIdentifier NOT_VERIFIED RAISE_EXCEPTION
userIdentifierNotUnique;

IF POST_CONDITION inv_existingTopologicalLink NOT_VERIFIED RAISE_EXCEPTION
failureToCreateTopologicalLink;

IF POST_CONDITION inv_linkBinds NOT_VERIFIED RAISE_EXCEPTION failureToBindTopologicalLink;

IF POST_CONDITION inv_agreedUserIdentifier NOT_VERIFIED RAISE_EXCEPTION
failureToSetUserIdentifier;

IF POST_CONDITION inv_agreedUserLabel NOT_VERIFIED RAISE_EXCEPTION failureToSetUserLabel;

IF POST_CONDITION inv_agreedDirectionality NOT_VERIFIED RAISE_EXCEPTION
failureToSetDirectionality;

;}

7.2.2.10 Delete topological link

<COMMUNITY: Topology Management, ACTION: delete topological link>

OPERATION deleteTopologicalLink {

INPUT_PARAMETERS

layerND: LayerNetworkDomainId;

topologicalLink: TopologicalLinkId;

OUTPUT_PARAMETERS

-- none

RAISED_EXCEPTIONS

incorrectTopologicalLink: topologicalLink;

serverTrailExisting: NULL;

failureToDeleteTopologicalLink: NULL;

BEHAVIOUR

SEMI_FORMAL

PARAMETER_MATCHING

layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;

topologicalLink: <INFORMATION OBJECT: topmanTopologicalLink>;

PRE_CONDITIONS;

inv_existingTopologicalLink

"**topologicalLink** refers to the *element* in the <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

inv_noServerTrail

"**topologicalLink** shall not refer to any *clientTL* of a <topologicalLinkIsSupportedByTrail> relationship."

POST_CONDITIONS

inv_noTopologicalLink

"**topologicalLink** does not participate in any <layerNetworkDomainIsMadeOf> AND <linkBinds> relationships."

EXCEPTIONS

```
IF PRE_CONDITION inv_existingTopologicalLink NOT_VERIFIED RAISE_EXCEPTION
    incorrectTopologicalLink;
IF PRE_CONDITION inv_noServerTrail NOT_VERIFIED RAISE_EXCEPTION serverTrailExisting;
IF POST_CONDITION inv_noTopologicalLink NOT_VERIFIED RAISE_EXCEPTION
    failureToDeleteTopologicalLink;
```

```
;}

```

7.2.2.11 Create topological link end

This operation creates a topological link end bound to a subnetwork.

<COMMUNITY: Topology Management, ACTION: create topological link end>

OPERATION createTopologicalLinkEnd {

INPUT_PARAMETERS

```
layerND: LayerNetworkDomainId;
subnetwork: SubnetworkId;
suppliedUserIdentifier: UserIdentifier;
    -- zero length string or 0 implies none supplied.
suppliedUserLabel: GraphicString;
    -- zero length string implies none supplied.
suppliedDirection: topologicalEndDirection;
```

OUTPUT_PARAMETERS

```
topologicalLinkEnd: TopologicalLinkEndId;
```

RAISED_EXCEPTIONS

```
incorrectSubnetwork: subnetwork;
userIdentifierNotUnique: suppliedUserIdentifier;
failureToCreateTopologicalLinkEnd: NULL;
failureToBindTopologicalLinkEnd: NULL;
failureToSetUserIdentifier: NULL;
failureToSetUserLabel: NULL;
failureToSetDirection: NULL;
```

BEHAVIOUR

SEMI_FORMAL

PARAMETER_MATCHING

```
layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
subnetwork: <INFORMATION OBJECT: topmanSubnetwork>;
suppliedUserIdentifier: <INFORMATION ATTRIBUTE: resourceId>;
suppliedUserLabel: <INFORMATION ATTRIBUTE: userLabel>;
suppliedDirection: <INFORMATION ATTRIBUTE: topologicalEndDirection>;
topologicalLinkEnd: <INFORMATION OBJECT: topmanTopologicalLinkEnd>;
```

PRE_CONDITIONS

inv_subnetworkExisting

"**subnetwork** shall refer to *element* in the <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

inv_uniqueUserIdentifier

"The **suppliedUserIdentifier** value shall not be equal to the <resourceId> value of any *element* in the <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*.";

POST_CONDITIONS

inv_existingTopologicalLinkEnd

"**topologicalLinkEnd** and **subnetwork** refer to *element* in the same <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

inv_boundTopologicalLinkEnd
"topologicalLinkEnd refers to *transferCapacityLE* in the <linkEndIsBoundTo> relationship where
subnetwork refers to *topologicalEntity*."

inv_agreedUserIdentifier
"The <resourceId> value of <topmanTopologicalLinkEnd> referred to by **topologicalLinkEnd** is equal to
the **suppliedUserIdentifier** value, if it is supplied."

inv_agreedUserLabel
"The <userLabel> value of <topmanTopologicalLinkEnd> referred to by **topologicalLinkEnd** is equal to
the **suppliedUserLabel** value, if it is supplied."

inv_agreedDirection
"The <topologicalEndDirection> value of <topmanTopologicalLinkEnd> referred to by
topologicalLinkEnd is equal to the **suppliedDirection** value, if it is supplied, or set to 'undefined' if not
supplied."

EXCEPTIONS

```
IF PRE_CONDITION inv_subnetworkExisting NOT_VERIFIED RAISE_EXCEPTION
    incorrectSubnetwork;
IF PRE_CONDITION inv_uniqueUserIdentifier NOT_VERIFIED RAISE_EXCEPTION
    userIdentifierNotUnique;
IF POST_CONDITION inv_existingTopologicalLinkEnd NOT_VERIFIED RAISE_EXCEPTION
    failureToCreateTopologicalLinkEnd;
IF POST_CONDITION inv_boundTopologicalLinkEnd NOT_VERIFIED RAISE_EXCEPTION
    failureToBindTopologicalLinkEnd;
IF POST_CONDITION inv_agreedUserIdentifier NOT_VERIFIED RAISE_EXCEPTION
    failureToSetUserIdentifier;
IF POST_CONDITION inv_agreedUserLabel NOT_VERIFIED RAISE_EXCEPTION failureToSetUserLabel;
IF POST_CONDITION inv_agreedDirection NOT_VERIFIED RAISE_EXCEPTION
    failureToSetDirection;
;}
```

7.2.2.12 Delete topological link end

<COMMUNITY: Topology Management, ACTION: delete topological link end>

```
OPERATION deleteTopologicalLinkEnd {
    INPUT_PARAMETERS
        layerND: LayerNetworkDomainId;
        topologicalLinkEnd: TopologicalLinkId;

    OUTPUT_PARAMETERS
        -- none

    RAISED_EXCEPTIONS
        incorrectTopologicalLinkEnd: topologicalLinkEnd;
        serverTTPExisting: NULL;
        failureToDeleteTopologicalLinkEnd: NULL;
```

BEHAVIOUR

SEMI_FORMAL

```
PARAMETER_MATCHING
    layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
    topologicalLinkEnd: <INFORMATION OBJECT: topmanTopologicalLinkEnd>;

PRE_CONDITIONS
inv_existingTopologicalLinkEnd
    "topologicalLinkEnd refers to element in the <layerNetworkDomainIsMadeOf> relationship where  
layerND refers to containerLND."
```

```

inv_noServerTTP
  "topologicalLinkEnd shall not refer to any clientTLE in a
  <topologicalLinkEndIsSupportedByNetworkTTP> relationship."

```

```

POST_CONDITIONS
inv_noTopologicalLinkEnd
  "topologicalLinkEnd does not refer to any element in the <layerNetworkDomainIsMadeOf> relationship."

```

```

EXCEPTIONS
IF PRE_CONDITION inv_existingTopologicalLinkEnd NOT_VERIFIED RAISE_EXCEPTION
  incorrectTopologicalLinkEnd;
IF PRE_CONDITION inv_noServerTTP NOT_VERIFIED RAISE_EXCEPTION serverTTPExisting;
IF POST_CONDITION inv_noTopologicalLinkEnd NOT_VERIFIED RAISE_EXCEPTION
  failureToDeleteTopologicalLinkEnd;

```

```

;}

```

7.3 Report interfaces

7.3.1 Common report resourceId change interface

The common report resourceId change interface provides functionality for reporting the change of the resource identifier of the resources by using the operation "report resourceId change". It satisfies the enterprise requirements stated in <"Rec. G.852.2", COMMUNITY: Transport Network Enterprise Model, General policy>.

```

COMPUTATIONAL_INTERFACE commonReportResourceIdChangeIfce {
  OPERATION      <reportResourceIdChange>; }

```

7.3.1.1 Report resourceId change

<"Rec. G.852.2", COMMUNITY: Transport Network Enterprise Model, General policy>

```

OPERATION reportResourceIdChange {
  INPUT_PARAMETERS
    layerND: LayerNetworkDomainId;
    involvedResource: ResourceChoice
    oldResourceIdentifier: UserIdentifier ::= SimpleNameType;
    newResourceIdentifier: UserIdentifier ::= SimpleNameType;

  OUTPUT_PARAMETERS
    -- none

  RAISED_EXCEPTIONS
    -- none

```

BEHAVIOUR

SEMI-FORMAL

```

PARAMETER_MATCHING
  layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
  involvedResource: <INFORMATION OBJECT networkInformationTop>;
  oldResourceIdentifier: <INFORMATION ATTRIBUTE: resourceId>;
  newResourceIdentifier: <INFORMATION ATTRIBUTE: resourceId>;

```

TRIGGERING_CONDITIONS

PRE_CONDITIONS

```

inv_existingInvolvedResource
  "involvedResource shall refer to element in the same <layerNetworkDomainIsMadeOf> relationship
  where layerND refers to containerLND."

```

```

    inv_oldResourceIdentifier
        "<resourceId> of the resource referred to by involvedResource must be equal to oldResourceIdentifier.";

    inv_differentNewResourceIdentifier
        "<resourceId> of the resource referred to by involvedResource must be different from
        newResourceIdentifier.";

    POST_CONDITIONS
        inv_newResourceIdentifier
            "<resourceId> of the resource referred to by involvedResource must be equal to newResourceIdentifier.";

    EXCEPTIONS
        -- none
;}
```

7.3.2 Topology resources reporting interface

The Topology Resources Reporting Interface provides functionality for the reporting of the creation and deletion of the resources defined in <"Rec. G.852.2", COMMUNITY: Transport Network Enterprise Model>.

```

COMPUTATIONAL_INTERFACE topologyResourcesReportingIfce {
    OPERATION
        <reportAccessGroupCreation>;
        <reportAccessGroupDeletion>;
        <reportLinkCreation>;
        <reportLinkDeletion>;
        <reportLinkEndCreation>;
        <reportLinkEndDeletion>;
        <reportSubnetworkTPCreation>;
        <reportSubnetworkTPDeletion>;
        <reportSubnetworkCreation>;
        <reportSubnetworkDeletion>; }
        <reportTopologicalLinkCreation>;
        <reportTopologicalLinkDeletion>;
        <reportTopologicalLinkEndCreation>;
        <reportTopologicalLinkEndDeletion>; }
```

7.3.2.1 Report access group creation

<COMMUNITY: Topology Management, ACTION: report access group creation>

```

OPERATION reportAccessGroupCreation {
    INPUT_PARAMETERS
        layerND: LayerNetworkDomainId;
        accessGroup: AccessGroupId;

    OUTPUT_PARAMETERS
        -- none

    RAISED_EXCEPTIONS
        -- none

    BEHAVIOUR
    SEMI-FORMAL
        PARAMETER_MATCHING
            layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
            accessGroup: <INFORMATION OBJECT: topmanAccessGroup>;
```

```

    TRIGGERING_CONDITIONS
    PRE_CONDITIONS
        -- none

    POST_CONDITIONS
    inv_existingAccessGroup
        "accessGroup refers to element in the <layerNetworkDomainIsMadeOf> relationship where layerND refers
        to containerLND."

    EXCEPTIONS
        -- none
;}
```

7.3.2.2 Report access group deletion

<COMMUNITY: Topology Management, ACTION: report access group deletion>

```

OPERATION reportAccessGroupDeletion {
    INPUT_PARAMETERS
        layerND: LayerNetworkDomainId;
        accessGroup: AccessGroupId;

    OUTPUT_PARAMETERS
        -- none

    RAISED_EXCEPTIONS
        -- none

BEHAVIOUR
SEMI-FORMAL
    PARAMETER_MATCHING
        layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
        accessGroup: <INFORMATION OBJECT: topmanAccessGroup>;

    TRIGGERING_CONDITIONS
    PRE_CONDITIONS
    inv_existingAccessGroup
        "accessGroup refers to element in the <layerNetworkDomainIsMadeOf> relationship where layerND refers
        to containerLND."

    inv_noNetworkTTPs
        "accessGroup shall not refer to any containerAG in an <accessGroupIsMadeOfNetworkTTPs>
        relationship."

    POST_CONDITIONS
    inv_noAccessGroup
        "accessGroup shall not refer to any element in the <layerNetworkDomainIsMadeOf> relationship."

    EXCEPTIONS
        -- none
;}
```

7.3.2.3 Report link creation

<COMMUNITY: Topology Management, ACTION: report link creation>

```

OPERATION reportLinkCreation {
    INPUT_PARAMETERS
        layerND: LayerNetworkDomainId;
        link: LinkId;
```

a_End: LinkEndChoice;
z_End: LinkEndChoice;

OUTPUT PARAMETERS
-- none

RAISED EXCEPTIONS
-- none

BEHAVIOUR

SEMI-FORMAL

PARAMETER_MATCHING
layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>; link: <INFORMATION OBJECT:
topmanLink>;
subnetwork: <INFORMATION OBJECT: topmanSubnetwork>;
accessGroup: <INFORMATION OBJECT: topmanAccessGroup>;

TRIGGERING_CONDITIONS
PRE_CONDITIONS
-- none

POST_CONDITIONS
inv_existingLink
"link refers to *element* in the <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to
containerLND."

inv_linkBinds
"a_End and z_End respectively refer to a_endTopological and z_endTopological in the <linkBinds>
relationship where **link** refers to transferCapacityLink."

EXCEPTIONS
-- none

;}

7.3.2.4 Report link deletion

<COMMUNITY: Topology Management, ACTION: report link deletion>

OPERATION reportLinkDeletion {
INPUT_PARAMETERS
layerND: LayerNetworkDomainId;
link: LinkId;

OUTPUT_PARAMETERS
-- none

RAISED_EXCEPTIONS
-- none

BEHAVIOUR

SEMI-FORMAL

PARAMETER_MATCHING
layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
link: <INFORMATION OBJECT: topmanLink>;

TRIGGERING_CONDITIONS
PRE_CONDITIONS

inv_existingLink
"link refers to *element* in the <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to
containerLND."

```

inv_noLinkConnection
    "link shall not refer to any containerLink of a <linkHasLinkConnections> relationship."

POST_CONDITIONS
inv_noLink
    "link does not refer to any element in the <layerNetworkDomainIsMadeOf> relationship."

EXCEPTIONS
    -- none
;}
```

7.3.2.5 Report link end creation

<COMMUNITY: Topology Management, ACTION: report link end creation>

```

OPERATION reportLinkEndCreation {
    INPUT_PARAMETERS
        layerND: LayerNetworkDomainId;
        linkEnd: LinkEndId;
        subnetwork: SubnetworkId;

    OUTPUT PARAMETERS
        -- none

    RAISED EXCEPTIONS
        -- none

BEHAVIOUR
SEMI-FORMAL
    PARAMETER_MATCHING
        layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
        linkEnd: <INFORMATION OBJECT: topmanLinkEnd>;
        subnetwork: <INFORMATION OBJECT: topmanSubnetwork>;

    TRIGGERING_CONDITIONS
    PRE_CONDITIONS
        -- none

    POST_CONDITIONS
inv_existingLinkEnd
    "linkEnd and subnetwork refer to element in the same <layerNetworkDomainIsMadeOf> relationship
    where layerND refers to containerLND."

inv_boundLinkEnd
    "linkEnd refers to transferCapacityLE in the <linkEndIsBoundTo> relationship where subnetwork refers to
    topologicalEntity."

    EXCEPTIONS
        -- none
;}
```

7.3.2.6 Report link end deletion

<COMMUNITY: Topology Management, ACTION: report link end deletion>

```

OPERATION reportLinkEndDeletion {
    INPUT_PARAMETERS
        layerND: LayerNetworkDomainId;
        linkEnd: LinkEndId;
```

OUTPUT PARAMETERS

-- none

RAISED EXCEPTIONS

-- none

BEHAVIOUR

SEMI-FORMAL

PARAMETER_MATCHING

layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;

linkEnd: <INFORMATION OBJECT: topmanLinkEnd>;

TRIGGERING_CONDITIONS

PRE_CONDITIONS

inv_existingLinkEnd

"**linkEnd** refers to *element* in the <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

inv_existingLinkEndBinding

"**linkEnd** refers to *topologicalEntity* in the <linkEndIsBoundTo> relationship."

inv_noNetworkCTPs

"**linkEnd** shall not refer to any *containerLE* in a <linkEndHasNetworkCTPs> relationship."

POST_CONDITIONS

inv_noLinkEnd

"**linkEnd** does not refer to any *element* in the <layerNetworkDomainIsMadeOf> relationship where **layerND** refers to *containerLND*."

EXCEPTIONS

-- none

;}

7.3.2.7 Report subnetwork creation

<COMMUNITY: Topology Management, ACTION: report subnetwork creation>

OPERATION reportSubnetworkCreation {

INPUT_PARAMETERS

layerND: LayerNetworkDomainId;

subnetwork: SubnetworkId;

networkTTPs: SetOfNnetworkTTPs;

-- the networkTTPs parameter value is provided when <PERMISSION: informTTP s> is supported.

OUTPUT_PARAMETERS

-- none

RAISED_EXCEPTIONS

-- none

BEHAVIOUR

SEMI-FORMAL

PARAMETER_MATCHING

layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;

subnetwork: <INFORMATION OBJECT: topmanSubnetwork>;

networkTTPs ELEMENTS: <INFORMATION OBJECT: topmanNetworkTTP>;

TRIGGERING_CONDITIONS

PRE_CONDITIONS

-- none

```

    POST_CONDITIONS
    inv_existingSubnetwork
        "subnetwork refer to the element in the <layerNetworkDomainIsMadeOf> relationship where layerND
        refers to containerLND."
    ;

    EXCEPTIONS
        -- none
;}
```

7.3.2.8 Report subnetwork deletion

<COMMUNITY: Topology Management, ACTION: report subnetwork deletion>

```

OPERATION reportSubnetworkDeletion {
    INPUT_PARAMETERS
        layerND: LayerNetworkDomainId;
        subnetwork: SubnetworkId;

    OUTPUT_PARAMETERS
        -- none

    RAISED_EXCEPTIONS
        -- none

    BEHAVIOUR
    SEMI-FORMAL
        PARAMETER_MATCHING
            layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
            subnetwork: <INFORMATION OBJECT: topmanSubnetwork>;

    TRIGGERING_CONDITIONS
    PRE_CONDITIONS
    inv_existingSubnetwork
        "subnetwork must refer to element in a <layerNetworkDomainIsMadeOf> relationship where layerND
        refers to containerLND."

    inv_noSubnetworkTP
        "subnetwork shall not refer to any containerSN in a <subnetworkIsDelimitedBy> relationship."

    inv_noBinding
        "subnetwork shall not refer to any a_endTopological or z_endTopological in a <linkBinds> relationship or
        to any topologicalEntity in a <linkEndIsBoundTo> relationship."

    POST_CONDITIONS
    inv_noSubnetwork
        "subnetwork does not participate in the <layerNetworkDomainIsMadeOf> relationship."

    EXCEPTIONS
        -- none
;}
```

7.3.2.9 Report subnetworkTP creation

<COMMUNITY: Topology Management, COMMUNITY POLICY: architecturalConstraints>

```

OPERATION reportSubnetworkTPCreation {
  INPUT_PARAMETERS
    layerND: LayerNetworkDomainId;
    subnetworkTP: SubnetworkTPId ::= SimpleNameType;

  OUTPUT_PARAMETERS
    -- none

  RAISED_EXCEPTIONS
    -- none

BEHAVIOUR
SEMI-FORMAL
  PARAMETER_MATCHING
    layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
    subnetworkTP: <INFORMATION OBJECT: topmanSubnetworkTP>;

  TRIGGER CONDITION
  PRE_CONDITIONS
    -- none

  POST_CONDITIONS
    inv_existingSubnetworkTP
      "subnetworkTP refer to the element in the <layerNetworkDomainIsMadeOf> relationship where layerND
      refers to containerLND."
    ;

  EXCEPTIONS
    -- none
;}

```

7.3.2.10 Report subnetworkTP deletion

<COMMUNITY: Topology Management, COMMUNITY POLICY: architecturalConstraints>

```

OPERATION reportSubnetworkTPDeletion {
  INPUT_PARAMETERS
    layerND: LayerNetworkDomainId;
    subnetworkTP: SubnetworkTPId ::= SimpleNameType;

  OUTPUT_PARAMETERS
    -- none

  RAISED_EXCEPTIONS
    -- none

BEHAVIOUR
SEMI-FORMAL
  PARAMETER_MATCHING
    layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
    subnetworkTP: <INFORMATION OBJECT: topmanSubnetworkTP>;

  TRIGGER CONDITION
  PRE_CONDITIONS
    inv_existingSubnetworkTP
      "subnetworkTP must refer to element in a <layerNetworkDomainIsMadeOf> relationship where layerND
      refers to containerLND."

```

```

    POST_CONDITIONS
    inv_noSubnetworkTP
    "subnetworkTP does not participate in the <layerNetworkDomainIsMadeOf> relationship."

    EXCEPTIONS
    -- none
;}

```

7.3.2.11 Report topological link creation

<COMMUNITY: Topology Management, ACTION: report topological link creation>

```

OPERATION reportTopologicalLinkCreation {
    INPUT_PARAMETERS
    layerND: LayerNetworkDomainId;
    topologicalLink: TopologicalLinkId;
    a_End: LinkEndChoice;
    z_End: LinkEndChoice;

    OUTPUT PARAMETERS
    -- none

    RAISED EXCEPTIONS
    -- none

    BEHAVIOUR
    SEMI-FORMAL
    PARAMETER_MATCHING
    layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
    topologicalLink: <INFORMATION OBJECT: topmanTopologicalLink>;
    subnetwork: <INFORMATION OBJECT: topmanSubnetwork>;
    accessGroup: <INFORMATION OBJECT: topmanAccessGroup>;

    TRIGGERING_CONDITIONS
    PRE_CONDITIONS
    -- none

    POST_CONDITIONS
    inv_existingTopologicalLink
    "topologicalLink refers to element in the <layerNetworkDomainIsMadeOf> relationship where layerND
    refers to containerLND."

    inv_linkBinds
    "a_End and z_End respectively refer to a_endTopological and z_endTopological in the <linkBinds>
    relationship where link refers to transferCapacityLink."

    EXCEPTIONS
    -- none
;}

```

7.3.2.12 Report topological link deletion

<COMMUNITY: Topology Management, ACTION: report topological link deletion>

```

OPERATION reportTopologicalLinkDeletion {
    INPUT_PARAMETERS
    layerND: LayerNetworkDomainId;
    topologicalLink: TopologicalLinkId;

```

```

OUTPUT_PARAMETERS
    -- none

RAISED_EXCEPTIONS
    -- none

BEHAVIOUR
SEMI-FORMAL
    PARAMETER_MATCHING
        layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
        topologicalLink: <INFORMATION OBJECT: topmanTopologicalLink>;

    TRIGGERING_CONDITIONS
        PRE_CONDITIONS
            inv_existingTopologicalLink
                "topologicalLink refers to element in the <layerNetworkDomainIsMadeOf> relationship where layerND
                refers to containerLND."
            inv_noLinkConnection
                "topologicalLink shall not refer to any containerLink of a <linkHasLinkConnections> relationship."

        POST_CONDITIONS
            inv_noLink
                "topologicalLink does not refer to any element in the <layerNetworkDomainIsMadeOf> relationship."

    EXCEPTIONS
        -- none
;}
```

7.3.2.13 Report topological link end creation

<COMMUNITY: Topology Management, ACTION: report topological link end creation>

```

OPERATION reportTopologicalLinkEndCreation {
    INPUT_PARAMETERS
        layerND: LayerNetworkDomainId;
        topologicalLinkEnd: TopologicalLinkEndId;
        subnetwork: SubnetworkId;

    OUTPUT_PARAMETERS
        -- none

    RAISED_EXCEPTIONS
        -- none

    BEHAVIOUR
    SEMI-FORMAL
        PARAMETER_MATCHING
            layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
            topologicalLinkEnd: <INFORMATION OBJECT: topmanTopologicalLinkEnd>;
            subnetwork: <INFORMATION OBJECT: topmanSubnetwork>;

        TRIGGERING_CONDITIONS
            PRE_CONDITIONS
                -- none

            POST_CONDITIONS
                inv_existingLinkEnd
                    "topologicalLinkEnd and subnetwork refer to element in the same <layerNetworkDomainIsMadeOf>
                    relationship where layerND refers to containerLND."
```

inv_boundTopologicalLinkEnd
"topologicalLinkEnd refers to *transferCapacityLE* in the <linkEndIsBoundTo> relationship where
subnetwork refers to *topologicalEntity*."

EXCEPTIONS
 -- none

;}

7.3.2.14 Report topological link end deletion

<COMMUNITY: Topology Management, ACTION: report topological link end deletion>

OPERATION reportTopologicalLinkEndDeletion {
 INPUT_PARAMETERS
 layerND: LayerNetworkDomainId;
 topologicalLinkEnd: TopologicalLinkEndId;

OUTPUT PARAMETERS
 -- none

RAISED EXCEPTIONS
 -- none

BEHAVIOUR

SEMI-FORMAL

PARAMETER_MATCHING
 layerND: <INFORMATION OBJECT: topmanLayerNetworkDomain>;
 topologicalLinkEnd: <INFORMATION OBJECT: topmanTopologicalLinkEnd>;

TRIGGERING_CONDITIONS

PRE_CONDITIONS

inv_existingTopologicalLinkEnd
"topologicalLinkEnd refers to *element* in the <layerNetworkDomainIsMadeOf> relationship where
layerND refers to *containerLND*."

inv_existingTopologicalLinkEndBinding
"topologicalLinkEnd refers to *topologicalEntity* in the <linkEndIsBoundTo> relationship."

 inv_noNetworkCTPs
"topologicalLinkEnd shall not refer to any *containerLE* in a <linkEndHasNetworkCTPs> relationship."

POST_CONDITIONS
 inv_noTopologicalLinkEnd
"topologicalLinkEnd does not refer to any *element* in the <layerNetworkDomainIsMadeOf> relationship
 where **layerND** refers to *containerLND*."

EXCEPTIONS
 -- none

;}

7.4 ASN.1 supporting productions

In this specification, when an interface name is used within an ASN.1 production, the same label will be used, starting with a capital letter. The complete ASN.1 type definition for this query interface (e.g. use of ObjectIdentifier, INTEGER, ...) will be developed as part of the engineering viewpoint, with the concerned technology.

AccessGroupId ::= CHOICE {	
accessGroupQueryIfce	TopmanAccessGroupQueryIfce
userIdentifier	UserIdentifier};
LayerNetworkDomainId ::= CHOICE {	
layerNetworkDomainQueryIfce	TopmanLayerNetworkDomainQueryIfce,
userIdentifier	UserIdentifier };
LinkEndChoice ::= CHOICE {	
subnetwork	SubnetworkId,
accessGroup	AccessGroupId };
LinkEndId ::= CHOICE {	
linkEndQueryIfce	TopmanLinkEndQueryIfce,
userIdentifier	UserIdentifier };
LinkId ::= CHOICE {	
linkQueryIfce	TopmanLinkQueryIfce,
userIdentifier	UserIdentifier };
ResourceChoice ::= CHOICE {	
resourceId	ResourceIfce,
userIdentifier	SimpleNameType};
SetOfNetworkTTPs ::= SET OF CHOICE {	
networkTTPId	TopmanNetworkTTPQueryIfce,
userIdentifier	SimpleNameType}
SubnetworkId ::= CHOICE {	
subnetworkQueryIfce	TopmanSubnetworkQueryIfce,
userIdentifier	UserIdentifier};
TopologicalLinkEndId ::= CHOICE {	
topologicalLinkEndQueryIfce	TopmanTopologicalLinkEndQueryIfce,
userIdentifier	UserIdentifier };
TopologicalLinkId ::= CHOICE {	
topologicalLinkQueryIfce	TopmanTopologicalLinkQueryIfce,
userIdentifier	UserIdentifier };
UserIdentifier ::= SimpleNameType;	

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
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	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	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
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 and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems