



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.854.1

(11/96)

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

Digital transmission systems – Digital networks – SDH
network characteristics

**Management of the transport network –
Computational interfaces for basic transport
network model**

ITU-T Recommendation G.854.1

(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
TRANSMISSION MEDIA CHARACTERISTICS	G.600–G.699
DIGITAL TRANSMISSION SYSTEMS	
TERMINAL EQUIPMENTS	G.700–G.799
General	G.700–G.709
Coding of analogue signals by pulse code modulation	G.710–G.719
Coding of analogue signals by methods other than PCM	G.720–G.729
Principal characteristics of primary multiplex equipment	G.730–G.739
Principal characteristics of second order multiplex equipment	G.740–G.749
Principal characteristics of higher order multiplex equipment	G.750–G.759
Principal characteristics of transcoder and digital multiplication equipment	G.760–G.769
Operations, administration and maintenance features of transmission equipment	G.770–G.779
Principal characteristics of multiplexing equipment for the synchronous digital hierarchy	G.780–G.789
Other terminal equipment	G.790–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.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
General	G.900–G.909
Parameters for optical fibre cable systems	G.910–G.919
Digital sections at hierarchical bit rates based on a bit rate of 2048 kbit/s	G.920–G.929
Digital line transmission systems on cable at non-hierarchical bit rates	G.930–G.939
Digital line systems provided by FDM transmission bearers	G.940–G.949
Digital line systems	G.950–G.959
Digital section and digital transmission systems for customer access to ISDN	G.960–G.969
Optical fibre submarine cable systems	G.970–G.979
Optical line systems for local and access networks	G.980–G.999

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

ITU-T RECOMMENDATION G.854.1

MANAGEMENT OF THE TRANSPORT NETWORK – COMPUTATIONAL INTERFACES FOR BASIC TRANSPORT NETWORK MODEL

Summary

This Recommendation contains the computational viewpoint specification for managing a transport network from a network viewpoint. In particular, this Recommendation details the computational viewpoint for subnetwork connection management. This Recommendation was written in accordance with G.851.1 Management of the transport network – Application of the RM-ODP framework.

Source

ITU-T Recommendation G.854.1 was prepared by ITU-T Study Group 15 (1993-1996) and was approved under the WTSC Resolution No. 1 procedure on the 8th of November 1996.

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 expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

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/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 1997

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	Computational interfaces for basic transport network model.....	2
	Annex A – Computational interfaces for simple subnetwork connection configuration.....	2
A.0	Computational interfaces to satisfy simple subnetwork connection configuration community enterprise requirements.....	2
A.1	Label references.....	2
A.2	simple SNC performer interface.....	3
	A.2.1 ssc set-up SNC.....	3
	A.2.2 ssc release SNC.....	5
A.3	ssc SN interface.....	6
	A.3.1 query SN for snTPs.....	7
	A.3.2 query SN for SNCs.....	7
A.4	snTP interface.....	8
	A.4.1 query snTP for SNC.....	8
	A.4.2 query snTP for subnetwork the point delimits.....	9
	A.4.3 query snTP for nTP.....	9
A.5	Service characteristics interface.....	10
A.6	nTP interface.....	11
	A.6.1 query nTP for SnTPs	11
A.7	SNC interface.....	12
	A.7.1 get user label.....	12
	A.7.2 query SNC terminating points	13
	A.7.3 get directionality	14
	A.7.4 query SNC for Transport Service Characteristics.....	15
	Annex B – Computational interfaces for simple monitored subnetwork connection configuration.....	16
B.0	Computational interfaces to satisfy simple monitored subnetwork connection configuration community enterprise requirements.....	16
B.1	Label references.....	16
B.2	monitored SNC performer interface – derived from simpleSncPerformerIfce.....	16

	Page
B.2.1 suspend SNC monitor reporting	16
B.2.2 resume SNC monitor reporting.....	17
B.3 SNC monitor interface	18
B.3.1 Get SNC operational State.....	18
B.3.2 Get reportFailureStatus.....	19
B.4 SNC monitor reporting interface	20
B.4.1 report SNC failure notification	20
B.4.2 report SNC recovery notification.....	20

Recommendation G.854.1

MANAGEMENT OF THE TRANSPORT NETWORK – COMPUTATIONAL INTERFACES FOR BASIC TRANSPORT NETWORK MODEL

(Geneva, 1996)

1 Scope

This Recommendation describes the computational viewpoint specification for computational interfaces for basic transport network model.

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 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. A list of the currently valid ITU-T Recommendations is regularly published.

- ITU-T Recommendation G.805 (1995), *Generic functional architecture of transport networks.*
- ITU-T Recommendation G.851.1 (1996), *Management of the transport network – Application of the RM-ODP framework.*
- ITU-T Recommendation G.852.1 (1996), *Management of the transport network – Enterprise viewpoint for simple subnetwork connection management.*
- ITU-T Recommendation G.853.1 (1996), *Common elements of the information viewpoint for the management of a transport network.*
- ITU-T Recommendation G.853.2 (1996), *Subnetwork connection management information viewpoint.*
- ITU-T Recommendation M.3100 (1995), *Generic network information model.*

3 Definitions

None.

4 Abbreviations

This Recommendation uses the following abbreviations.

Ifce Interface

ITU-T International Telecommunication Union – Telecommunication Standardization Sector

RM-ODP Reference Model for Open Distributed Processing

5 Computational interfaces for basic transport network model

Computational interfaces are specified using engineering neutral computational templates, defined in Recommendation G.851.1. These computational interface templates may be translated to templates associated with more than one engineering realization for use in particular communication domains.

A complete computational specification requires the definition of computational objects and the scenarios which describe how the computational objects are used. The specification of computational objects is for further study.

This is not the complete computational viewpoint for transport networks, and these interfaces may be satisfied by a profile or subset of a more comprehensive interface description.

An implementation which claims support of an interface in this Recommendation, as a server, shall implement all of the operations of that interface.

The inheritance structure of the computational viewpoint is not necessarily the same as the inheritance structure of the engineering viewpoint.

ANNEX A

Computational interfaces for simple subnetwork connection configuration

A.0 Computational interfaces to satisfy simple subnetwork connection configuration community enterprise requirements

The simple subnetwork connection community enterprise requirements are met by the interfaces specified in this Annex.

A reference to a snTP interface type is equivalent to a reference to a nTP interface type.

A.1 Label references

The following information relationships, static schema, and ASN.1 productions are referenced in this Annex:

Fully-qualified label reference	Local reference used
< "Rec. G.853.1", INFORMATION_RELATIONSHIP:subnetworkIsDelimitedBy >	<subnetworkIsDelimitedBy>
< "Rec. G.853.1", INFORMATION_RELATIONSHIP:subnetworkHasSubnetworkConnections>	<subnetworkHasSubnetworkConnections>
< "Rec. G.853.1", INFORMATION_RELATIONSHIP:extremitiesTerminateTransportEntity>	<extremitiesTerminateTransportEntity>
< "Rec. G.853.1", INFORMATION_RELATIONSHIP:subnetworkTPIsRelatedToExtremity>	<subnetworkTPIsRelatedToExtremity>
< "Rec. G.853.2", INFORMATION_RELATIONSHIP:subnetworkConnectionHasTSC >	<subnetworkConnectionHasTSC>
<"Rec. G.853.2", STATIC_SCHEMA:ssccNotConnected >	<ssccNotConnected>
<"Rec. G.853.2", STATIC_SCHEMA:ssccConnected >	<ssccConnected>
Fully-qualified ASN.1 production reference	Local reference used
"M.3100 : 199x : ASN1DefinedTypesModule>::Failed	Failed
"M.3100 : 199x : ASN1DefinedTypesModule>::Directionality	Directionality
"M.3100 : 199x : ASN1DefinedTypesModule>::UserLabel	UserLabel

A.2 simple SNC performer interface

The simple subnetwork performer manages the set-up and release of subnetwork connections.

The simple SNC performer interface is required to satisfy the enterprise requirements stated in:

```
<"Rec. G.852.1", COMMUNITY:sscc, ACTION:sscc1 > ,  
<"Rec. G.852.1", COMMUNITY:sscc, ACTION:sscc2 > .
```

The simple subnetwork connection performer interface provides basic connection set-up functionality. The operation `ssccSetupSubnetworkConnection` sets up a subnetwork connection, and the operation `ssccReleaseSubnetworkConnection` removes the subnetwork connection.

```
COMPUTATIONAL_INTERFACE simpleSncPerformerIfce {  
    OPERATION      <ssccSetupSubnetworkConnection>;  
                  <ssccReleaseSubnetworkConnection>;  
}
```

A.2.1 ssc set-up SNC

This operation sets up a simple subnetwork connection between a single A-End snTP or nTP, and a single Z-End snTP or nTP.

```
OPERATION      sscSetupSubnetworkConnection {
```

INPUT_PARAMETERS

```
    subnetwork : SubnetworkId ::= (ssccSnIfce);
```

```
-- The subnetwork parameter is used to indicate the subnetwork across which the performer is setting  
-- up the SNCs. This parameter is used, for example, when a given performer can set up SNCs in  
-- many subnetworks. If the performer is associated with a single subnetwork, the subnetwork  
-- parameter of this operation is redundant, and may be removed as an engineering optimization.
```

```
    snpa : SnTPId ::= (snTPIfce);
```

```
    snpz : SnTPId ::= (snTPIfce);
```

```
    dir : Directionality;
```

```
    suppliedUserLabel : UserLabel ;
```

```
        -- zero length string implies none supplied
```

```
    serviceCharacteristics: CharacteristicsId ::= (serviceCharacteristicsIfce) ;
```

```
        -- reference can be used to determine any QOS or routing characteristics ;
```

OUTPUT_PARAMETERS

```
    newSNC : SNCId ::= (sncIfce) ;
```

```
    agreedUserLabel : UserLabel ;
```

RAISED_EXCEPTIONS

```
    invalidTransportServiceCharacteristics: NULL;
```

```
    incorrectSubnetworkTerminationPoints : SEQUENCE OF SnTPId;
```

```
-- the list contains one element when only point is incorrect.
```

```
    subnetworkTerminationPointsConnected : SEQUENCE OF SnTPId;
```

```
-- the list contains one element when only one subnetworkTerminationPoint remains
```

```
-- connected
```

```
    failure : Failed;
```

```
    wrongDirectionality : Directionality;
```

```
    userLabelInUse : UserLabel;
```

BEHAVIOUR

```
    INFORMAL
```

```
!
```

This operation sets up a subnetwork connection between a given A-End snTP or nTP and a given Z-End snTP or nTP. The subnetwork termination points or network termination points to be

connected are specified by explicitly identifying the subnetwork network termination points or network termination points.

The client may supply a unique user label. If not supplied (i.e. string length zero) the provider assigns a user label for the connection.

A single point-to-point unidirectional, or point-to-point bidirectional, unpartitioned subnetwork connection object will be created. The subnetwork connection object will have one A-End and one Z-End.

The subnetwork connection will have a directionality (unidirectional or bidirectional) as specified in the operation parameters.

If used, the service characteristics specify one predetermined set of transport parameters which the server may offer.

The operation replies for set-up includes full information about the reasons in case the request could not be satisfied.

PRE_CONDITIONS

This operation will fail if any of the subnetwork termination points or network termination points specified is already involved in a subnetwork connection. The exception "subnetworkTerminationPointsConnected" will be generated.

This operation will fail if the subnetwork termination points or network termination points are not contained within the domain of the subnetwork. The exception "incorrectSubnetworkTerminationPoints" will be generated.

This operation will fail if the serviceCharacteristics requested is not supported by the computational object which executes the operation. The exception "invalidTransportServiceCharacteristics" will be generated.

POST_CONDITIONS

If any subnetwork connection input parameters cannot be met by the server, the operation will fail.

This operation will fail if the value of the userLabel of the SubnetworkConnection is zero or is not unique within the domain of the containing subnetwork. The exception "userLabelInUse" will be generated.

!

SEMI_FORMAL

PARAMETER_MATCHING

```
subnetwork: < sscNotConnected , ROLE:involvedSubnetwork > AND
    < sscConnected , ROLE:involvedSubnetwork > ;
snpa : < sscNotConnected , ROLE:potentialAEnd > AND
    < sscConnected , ROLE:connectedAEnd > ;
snpz : < sscNotConnected , ROLE: potentialZEnd > AND
    < sscConnected , ROLE:connectedZEnd > ;
dir : < sscConnected, ROLE: involvedSubnetwork , ATTRIBUTE: directionality > ;
newSNC : < sscConnected, ROLE: involvedSubnetwork > ;
suppliedUserLabel : < sscConnected, ROLE:involvedSubnetwork, ATTRIBUTE: userLabel >
    OR <> ; -- The user does not have to supply a user label value
agreedUserLabel : < sscConnected, ROLE:involvedSubnetwork, ATTRIBUTE: userLabel > ;
serviceCharacteristics : < sscConnected , ROLE:involvedServiceCharacteristics > ;
```

PRE_CONDITIONS < sscNotConnected > ;

```
-- The sscNotConnected schema defines a schema type with two non-connected
-- networkTP information objects subtypes candidates to the point-to-point connection
-- management service.
```

POST_CONDITIONS < sscConnected > ;

--The sscConnected schema defines the schema type of two connected networkTP
-- information objects candidates to the point-to-point connection management service.

EXCEPTIONS

IF PRE_CONDITION <inv_1> NOT_VERIFIED RAISE_EXCEPTION
incorrectSubnetworkTerminationPoints ;
IF PRE_CONDITION <inv_2> NOT_VERIFIED RAISE_EXCEPTION
subnetworkTerminationPointsConnected ;
IF PRE_CONDITION <inv_3> NOT_VERIFIED RAISE_EXCEPTION
subnetworkTerminationPointsConnected ;
IF POST_CONDITION <inv_1> NOT_VERIFIED RAISE_EXCEPTION failure ;
IF POST_CONDITION <inv_2> NOT_VERIFIED RAISE_EXCEPTION failure ;
IF POST_CONDITION <inv_3> NOT_VERIFIED RAISE_EXCEPTION failure ;
IF POST_CONDITION <inv_4> NOT_VERIFIED RAISE_EXCEPTION userLabelInUse ;

;
}

A.2.2 ssc release SNC

This operation releases a given subnetwork connection.

OPERATION sscReleaseSubnetworkConnection {

INPUT_PARAMETERS

subnetwork : SubnetworkId ::= (ssccSnIfce) ;

-- The subnetwork parameter is used to indicate the subnetwork across which the performer is
-- releasing the SNCs. This parameter is used, for example, when a given performer can release SNCs
-- in many subnetworks. If the performer is associated with a single subnetwork, the subnetwork
-- parameter of this operation is redundant, and may be removed as an engineering optimization.

connection: ConnectionId ::= CHOICE { userLabel UserLabel , sncId SNCId } ;

OUTPUT_PARAMETERS

userLabel : UserLabel ;

RAISED_EXCEPTIONS

noSuchConnection : ConnectionId ;

subnetworkTerminationPointsConnected : SEQUENCE OF SnTPId ;

-- the list contains one element when only one subnetworkTerminationPoint remains
-- connected

failure : Failed ;

invalidUserLabel : UserLabel ;

BEHAVIOUR

INFORMAL

!

This operation releases a given point-to-point subnetwork connection. This subnetwork connection is between two existing snTP or nTPs and is not partitioned.

PRE_CONDITIONS

This operation will fail if the subnetwork connection is not terminated by valid network termination points. The exception "failure" will be generated.

This operation will fail if the subnetwork connection id is not within the domain of the object which executes this operation. The exception "noSuchConnection" will be generated. Note that this exception is raised on a computational identifier, sncid, and is thus not defined in the EXCEPTION clause.

This operation will fail if the value of the userLabel supplied for the subnetwork connection is zero or is not within the domain of the containing subnetwork. The exception "invalidUserLabel" will be generated.

POST_CONDITIONS

This operation will fail if the userLabel returned does not match the userLabel supplied or is not within the domain of the object which executes this operation. The exception "failure" with a value "logical problem" will be generated.

If the requested subnetwork connection is still connected to a subnetwork termination point, the exception "subnetworkTerminationPointConnected" is raised.

```
!  
SEMI_FORMAL  
PARAMETER_MATCHING  
subnetwork: < sscNotConnected , ROLE:involvedSubnetwork > AND  
           < sscConnected, ROLE:involvedSubnetwork > ;  
connection : < sscConnected, ROLE: involvedPointToPointSubnetworkConnection,  
ATTRIBUTE:userLabel > OR < sscConnected, ROLE: involvedPointToPointSubnetworkConnection > ;  
  
PRE_CONDITIONS <ssccConnected> ;  
-- The sscConnected schema defines the schema type of two connected snTP or nTP  
-- information object candidates to the point-to-point connection management service.  
  
POST_CONDITIONS <ssccNotConnected> ;  
--The sscNotConnected schema defines a schema type with two non-connected  
--nTP information objects subtypes candidates to the point-to-point connection  
  
EXCEPTIONS  
-- The exception "noSuchConnection" is raised on a computational identifier and is thus not defined  
-- in the EXCEPTION clause.  
  
IF PRE_CONDITION <inv_1> NOT_VERIFIED RAISE_EXCEPTION failure ;  
IF PRE_CONDITION <inv_2> NOT_VERIFIED RAISE_EXCEPTION failure ;  
IF PRE_CONDITION <inv_3> NOT_VERIFIED RAISE_EXCEPTION failure ;  
IF PRE_CONDITION <inv_4> NOT_VERIFIED RAISE_EXCEPTION invalidUserLabel ;  
IF POST_CONDITION <inv_1> NOT_VERIFIED RAISE_EXCEPTION failure ;  
IF POST_CONDITION <inv_2> NOT_VERIFIED RAISE_EXCEPTION  
subnetworkTerminationPointsConnected ;  
IF POST_CONDITION <inv_3> NOT_VERIFIED RAISE_EXCEPTION  
subnetworkTerminationPointsConnected ;  
;  
}
```

A.3 ssc SN interface

The ssc SN interface is required to satisfy the enterprise requirements stated in:

<"Rec. G.852.1", COMMUNITY:sscc, OBLIGATION:OBLG_1>

The subnetwork represents logical collections of subnetwork termination points or network termination points.

```
COMPUTATIONAL_INTERFACE sscSnIfce {  
    OPERATION <querySnForSnTPs>;  
             <querySnForSNCs>;  
}
```

A.3.1 query SN for snTPs

This operation retrieves the list of snTP or nTPs that are associated with the given subnetwork.

OPERATION querySnForSnTPs {

INPUT_PARAMETERS

subnetwork : SubnetworkId ::= (ssccSnIfce)

*-- The subnetwork parameter is used to indicate which interface serves queries for that subnetwork.
-- This parameter is used, for example, when a given interface can serve queries for many
-- subnetworks. If the interface is associated with a single subnetwork, the subnetwork parameter of
-- this operation is redundant, and may be removed as an engineering optimization.*

OUTPUT_PARAMETERS

delimitingSNTPs : SnTPIs ::= SEQUENCE OF (snTPIfce)

-- list may be empty for an unprovisioned subnetwork

BEHAVIOUR

INFORMAL

!

This operation retrieves the list of snTP or nTPs that delimit the given subnetwork.

!

SEMI_FORMAL

PARAMETER_MATCHING

subnetwork : <subnetworkIsDelimitedBy,ROLE: container> ;

delimitingSNTPs ELEMENTS :< subnetworkIsDelimitedBy,ROLE: element> ;

;

}

A.3.2 query SN for SNCs

This operation retrieves the list of subnetwork connections that are contained within the given subnetwork.

OPERATION querySnForSNCs {

INPUT_PARAMETERS

subnetwork : SubnetworkId ::= (ssccSnIfce);

*-- The subnetwork parameter is used to indicate which interface serves queries for that subnetwork.
-- This parameter is used, for example, when a given interface can serve queries for many
-- subnetworks. If the interface is associated with a single subnetwork, the subnetwork parameter of
-- this operation is redundant, and may be removed as an engineering optimization.*

OUTPUT_PARAMETERS

containedSNCs : SNCs ::= SEQUENCE OF (SNCIfce); -- list may be empty

BEHAVIOUR

INFORMAL

!

This operation retrieves the list of subnetwork connections that are contained within the given subnetwork.

!

SEMI_FORMAL

PARAMETER_MATCHING

subnetwork : < subnetworkHasSubnetworkConnections , ROLE:container >;

containedSNCs ELEMENTS : < subnetworkHasSubnetworkConnections , ROLE:element >;

;

}

A.4 snTP interface

The snTP query interface is required to satisfy the enterprise requirements stated in:

```
<"Rec. G.852.1", COMMUNITY:sscc, OBLIGATION:OBLG_1 > .
```

The snTP delimits a subnetwork and provides an association to a nTP.

```
COMPUTATIONAL_INTERFACE snTPIfce {  
    OPERATION      <querySNTPforSNC>;  
                  <querySNTPforSN>;  
                  <querySNTPforNTP>;  
}
```

A.4.1 query snTP for SNC

This operation retrieves the subnetwork connection, if any, that is associated with the given subnetwork termination point.

```
OPERATION querySNTPforSNC {  
    INPUT_PARAMETERS  
        snTP : SnTPId ::= (snTPIfce);  
  
    -- The snTP parameter is used to indicate which interface serves queries for that snTP. This  
    -- parameter is used, for example, when a given interface can serve queries for many snTPs. If the  
    -- interface is associated with a single snTP, the snTP parameter of this operation is redundant, and  
    -- may be removed as an engineering optimization.
```

```
    OUTPUT_PARAMETERS  
        associatedSNC : SNCId ::= (snIfce);
```

```
    RAISED_EXCEPTIONS  
        unconnectedSNTP : SnTPId;
```

```
    BEHAVIOUR  
        INFORMAL
```

!

This operation retrieves the subnetwork connection that is associated with the given subnetwork termination point.

The subnetwork termination point must terminate at most one subnetwork connection.

An exception is generated if this operation fails.

```
!  
    SEMI_FORMAL  
        PARAMETER_MATCHING  
  
        associatedSNC : < extremitiesTerminateTransportEntity,ROLE: transportEntity>;  
  
        snTP : < extremitiesTerminateTransportEntity, ROLE: A_end>  
              OR < extremitiesTerminateTransportEntity , ROLE:Z_end>;  
  
        PRE_CONDITIONS  
        "inv1:<PARAMETER_MATCHING: snTP >";  
        POST_CONDITIONS  
        <PRE_CONDITIONS:inv1 >;  
  
        EXCEPTIONS  
        IF PRE_CONDITION <inv1> NOT_VERIFIED RAISE_EXCEPTION unconnectedSNTP;  
    ;  
}
```

A.4.2 query snTP for subnetwork the point delimits

This operation retrieves the subnetwork that is associated with the given subnetwork termination point.

OPERATION querySNTPforSN {

INPUT_PARAMETERS

snTP : SnTPId ::= (snTPIfce);

-- The snTP parameter is used to indicate which interface serves queries for that snTP. This parameter is used, for example, when a given interface can serve queries for many snTPs. If the interface is associated with a single snTP, the snTP parameter of this operation is redundant, and may be removed as an engineering optimization.

OUTPUT_PARAMETERS

containingSN : SubnetworkId ::= (ssccSnIfce);

RAISED_EXCEPTIONS

invalidSNTP : SnTPId ;

BEHAVIOUR

INFORMAL

!
This operation retrieves the subnetwork that is delimited by the given subnetwork termination point. The subnetwork termination point is contained in exactly one subnetwork.

An exception is generated if this operation fails.

!

SEMI_FORMAL

PARAMETER_MATCHING

containingSN : < subnetworkIsDelimitedBy , ROLE:container>;

snTP : < subnetworkIsDelimitedBy , ROLE:element>;

PRE_CONDITIONS

"inv1: <PARAMETER_MATCHING:snTP>" ;

POST_CONDITIONS

<PRE_CONDITIONS:inv1> ;

EXCEPTIONS

IF PRE_CONDITION <inv1> NOT_VERIFIED RAISE_EXCEPTION invalidSNTP ;

;

}

A.4.3 query snTP for nTP

This operation retrieves the network termination point that the subnetwork termination point is representing.

If the snTP is at the lowest level of partitioning, and is also an nTP, then engineering optimizations may eliminate this operation.

OPERATION querySNTPForNTP {

INPUT_PARAMETERS

snTP : SnTPId ::= (snTPIfce) ;

-- The snTP parameter is used to indicate which interface serves queries for that snTP. This parameter is used, for example, when a given interface can serve queries for many snTPs. If the interface is associated with a single snTP, the snTP parameter of this operation is redundant, and may be removed as an engineering optimization.

OUTPUT_PARAMETERS

nTP : NtpId ::= (nTPIfce) ;

RAISED_EXCEPTIONS

invalidSnTP : snTPIId ;

BEHAVIOUR**INFORMAL**

!

This operation retrieves the networkTTP, networkCTP, or link connection that is associated with a subnetwork termination point.

A bidirectional subnetwork termination point is associated with a bidirectional networkTTP, networkCTP, or link connection.

A subnetwork termination point source is associated with a networkTTP source, networkCTP sink, or unidirectional link connection.

A subnetwork termination point sink is associated with a networkTTP sink, networkCTP source, or unidirectional link connection.

An exception is generated if this operation fails.

!

SEMI_FORMAL**PARAMETER_MATCHING**

snTP : < subnetworkTPIsRelatedToExtremity , ROLE: abstraction > ;

nTP : < subnetworkTPIsRelatedToExtremity , ROLE:extremity > ;

PRE_CONDITIONS

inv1: <PARAMETER_MATCHING: snTP> ;

POST_CONDITIONS

<PRE_CONDITIONS:inv1> ;

EXCEPTIONS

IF PRE_CONDITION <inv1> NOT_VERIFIED RAISE_EXCEPTION invalidSnTP ;

;

}

A.5 Service characteristics interface

The service characteristics query interface is required to satisfy the enterprise requirements stated in:

<"Rec. G.852.1", COMMUNITY:sscc, ACTION:sscc1, PERMISSION:PERM_3> .

This interface is a superclass for technology specific interface subclasses, which can be used to allow the client to pass, by reference, an interface identifier which can be queried to determine the client service characteristic requirements for a requested subnetwork connection, e.g. transport QOS and route restriction characteristics.

This interface may be specialized through inheritance to add operations to conveying service characteristics appropriate for a specific technology. Subclasses of this interface type may also have technology specific modify operations, allowing the client to modify the requested service characteristics.

NOTE – Transport technology specific subclasses of this interface are for further study.

```

COMPUTATIONAL_INTERFACE serviceCharacteristicsIfce {
    OPERATION --none defined
}

```

A.6 nTP interface

The common nTP interface is required to satisfy the enterprise requirements stated in:

<"Rec. G.852.1", COMMUNITY:sscc, OBLIGATION:OBLG_1 > .

The common nTP interface represents the termination of a transport entity.

This interface type is intended to be subclassed to add operations for particular TP specializations (e.g. nTTP, nCTP).

```

COMPUTATIONAL_INTERFACE nTPIfce {
    OPERATION <queryNTPForSnTPs> ;
}

```

A.6.1 query nTP for SnTPs

This operation retrieves the list of subnetwork termination points that are associated with a networkTTP, networkCTP, or link connection at different levels of partitioning.

```

OPERATION queryNTPForSnTPs {
    INPUT_PARAMETERS
        ntp : NTPIId ::= (nTPIfce) ;
    -- The nTP parameter is used to indicate which interface serves queries for that nTP. This parameter
    -- is used, for example, when a given interface can serve queries for many nTPs. If the interface is
    -- associated with a single nTP, the nTP parameter of this operation is redundant, and may be
    -- removed as an engineering optimization.
}

    OUTPUT_PARAMETERS
        snTPs : SnTPids ::= SEQUENCE OF (snTPIfce) ;

    RAISED_EXCEPTIONS
        uncomposedNTP : NTPIId ;

```

BEHAVIOUR INFORMAL

!

A bidirectional subnetwork termination point is associated with a bidirectional networkTTP, networkCTP, or link connection.

A subnetwork termination point source is associated with a networkTTP source, networkCTP sink, or unidirectional link connection.

A subnetwork termination point sink is associated with a networkTTP sink, networkCTP source, or unidirectional link connection.

An exception is generated if this operation fails.

!

SEMI_FORMAL PARAMETER_MATCHING

```

nTP : < subnetworkTPIsRelatedToExtremity , ROLE: extremity > ;
snTPs ELEMENTS : < subnetworkTPIsRelatedToExtremity , ROLE: abstraction >
AND NOT <nTP>;

```

```

    PRE_CONDITIONS
    "inv1: <PARAMETER_MATCHING: nTP >" ;

    POST_CONDITIONS
    <PRE_CONDITIONS:inv1> ;

    EXCEPTIONS
    IF PRE_CONDITION <inv1> NOT_VERIFIED RAISE_EXCEPTION uncomposedNTP ;
;
}

```

A.7 SNC interface

The SNC interface is required to satisfy the enterprise requirements stated in:

<"Rec. G.852.1", COMMUNITY:sscc, OBLIGATION:OBLG_1 > .

The Subnetwork Connection associates the Network CTP(s), Network TP(s), or subnetworkTPs identified in the A-End and the Network CTP(s), Network TP(s), or subNetwork TP(s) identified in the Z-End. For all types of subnetwork connection, the network termination points at the A-End are related to the network termination points at the Z-End in such a way that traffic can flow between the network termination points in a unidirectional or bidirectional manner as indicated by the directionality attribute.

```

COMPUTATIONAL_INTERFACE sncIfce {
    OPERATION      <get_userLabel> ;
                  <querySncTerminatingPoints> ;
                  <get_directionality> ;
                  <querySncForTSC> ;
}

```

A.7.1 get user label

This operation retrieves the user label attribute value associated with the subnetwork connection. This operation may be implemented by direct manipulation of an attribute if the engineering viewpoint language supports attributes.

```

OPERATION      get_userLabel {
    INPUT_PARAMETERS  -- none
    OUTPUT_PARAMETERS
        userLabel_val : UserLabel ;

    RAISED_EXCEPTIONS
        cannotDetermineValue : NULL;

    BEHAVIOUR
        INFORMAL
!

```

This operation retrieves the value of a user label for the subnetwork connection.

An exception is generated if this operation fails.

```

!
    SEMI_FORMAL
    PARAMETER_MATCHING

    userLabel_val : <ATTRIBUTE: userLabel> ;

```

```

PRE_CONDITIONS
"inv_1 - value of userLabel attribute of SNC is obtainable and valid";

POST_CONDITIONS --none

EXCEPTIONS

IF PRE_CONDITION <inv_1> NOT_VERIFIED RAISE_EXCEPTION
    cannotDetermineValue ;
;
}

```

A.7.2 query SNC terminating points

This operation returns the subnetwork termination points or network termination points that terminate the given subnetwork connection.

```

OPERATION querySncTerminatingPoints {
    INPUT_PARAMETERS
        subnetworkConnection : SNCId ::= (sncIfce) ;
    -- The subnetworkConnection parameter is used to indicate which interface serves queries for that
    -- subnetworkConnection. This parameter is used, for example, when a given interface can serve
    -- queries for many subnetworkConnection. If the interface is associated with a single
    -- subnetworkConnection, the subnetworkConnection parameter of this operation is redundant, and
    -- may be removed as an engineering optimization.

    OUTPUT_PARAMETERS
        aEndTP : SnTP ::= (snTPIfce) ;
        zEndTP : SnTP ::= (snTPIfce) ;

    RAISED_EXCEPTIONS
        unterminatedSNC : SNCId ;

    BEHAVIOUR
    INFORMAL
    !

```

This operation returns the subnetwork termination points or network termination points that terminate the A-End and the Z-End of a given subnetwork connection.

An A-End may be a subnetworkTP source, or subnetworkTP bidirectional.

A Z-End may be a subnetworkTP sink, or subnetworkTP bidirectional.

A point-to-point unidirectional subnetwork connection can be established between one of snTP (or nTP) source and snTP (or nTP) sink.

If the SNC is bidirectional, A-End snTP or nTP and Z-End snTP or nTP are bidirectional.

For all types of subnetwork connection, the subnetwork network termination point which terminates the A-End is related to the subnetwork termination point which terminates the Z-End attribute in such a way that traffic can flow between these subnetwork terminations in a unidirectional or bidirectional manner as indicated by the directionality attribute.

An exception is raised if the subnetwork connection has no A-End subnetwork termination point or no Z-End subnetwork termination point.

An exception is generated if this operation fails.

!

```

SEMI_FORMAL
PARAMETER_MATCHING

```

```

subnetworkConnection: < extremitiesTerminateTransportEntity , ROLE: transportEntity> ;
aEndTPs ELEMENTS: < extremitiesTerminateTransportEntity , ROLE:A_end > ;
zEndTPs ELEMENTS : < extremitiesTerminateTransportEntity , ROLE:Z_end > ;

```

PRE_CONDITIONS

```
"inv1: <PARAMETER_MATCHING: subnetworkConnection > " ;
```

POST_CONDITIONS

```
<PRE_CONDITIONS:inv1> ;
```

EXCEPTIONS

```
IF PRE_CONDITION <inv1> NOT_VERIFIED RAISE_EXCEPTION unterminatedSNC ;
```

INFORMAL

```
"." ;
```

```
}
```

A.7.3 get directionality

This operation retrieves the directionality attribute value associated with the subnetwork connection. This operation may be implemented by direct manipulation of an attribute if the engineering viewpoint language supports attributes.

```

OPERATION    get_directionality {
  INPUT_PARAMETERS -- none
  OUTPUT_PARAMETERS
    directionality_val : Directionality ;

```

RAISED_EXCEPTIONS

```
cannotDetermineValue : NULL ;
```

BEHAVIOUR

INFORMAL

```
!
```

This operation retrieves the directionality attribute value associated with a transport connection. The value of the directionality is fixed during the lifetime of the transport connection.

An exception is generated if this operation fails.

```
!
```

SEMI_FORMAL

PARAMETER_MATCHING

```
directionality_val : <ATTRIBUTE: directionality> ;
```

PRE_CONDITIONS

```
"inv_1 - value of directionality attribute of SNC is obtainable and valid ";
```

POST_CONDITIONS --none

EXCEPTIONS

```
IF PRE_CONDITION <inv_1> NOT_VERIFIED RAISE_EXCEPTION
  cannotDetermineValue ;
```

```
;
```

```
}
```

A.7.4 query SNC for Transport Service Characteristics

OPERATION querySncForTSC {

INPUT_PARAMETERS

subnetworkConnection : SNCId ::= (sncIfce) ;

*-- The subnetworkConnection parameter is used to indicate which interface serves queries for that
-- subnetworkConnection. This parameter is used, for example, when a given interface can serve
-- queries for many subnetworkConnections. If the interface is associated with a single
-- subnetworkConnection, the subnetworkConnection parameter of this operation is redundant, and
-- may be removed as an engineering optimization.*

OUTPUT_PARAMETERS

tsc : TSCId ::= (serviceCharacteristicsIfce) ;

RAISED_EXCEPTIONS

uncharacterizedSNC : SNCId ;

invalidTSC : NULL;

BEHAVIOUR

INFORMAL

!

This operation returns the transport service characteristics that qualify the given subnetwork connection.

Several subnetwork connections may share the same transport service characteristics but a given subnetwork connection may only have one set of transport service characteristics.

An "uncharacterizedSNC" exception will be generated if this operation is directed towards a subnetwork connection which has no transport service characteristics.

An "invalidTSC" exception is generated if this operation fails.

!

SEMI_FORMAL

PARAMETER_MATCHING

subnetworkConnection : < subnetworkConnectionHasTSC , ROLE: transportQualified > ;

tsc: < subnetworkConnectionHasTSC , ROLE: transportQualifier > ;

PRE_CONDITIONS

" inv1: <PARAMETER_MATCHING: subnetworkConnection > ,

inv2: <PARAMETER_MATCHING:tsc > " ;

POST_CONDITIONS

<PRE_CONDITIONS:inv1> ;

<PRE_CONDITIONS:inv2> ;

EXCEPTIONS

IF PRE_CONDITION <inv1> NOT_VERIFIED RAISE_EXCEPTION uncharacterizedSNC ;

IF PRE_CONDITION <inv2> NOT_VERIFIED RAISE_EXCEPTION invalidTSC ;

;

}

ANNEX B

Computational interfaces for simple monitored subnetwork connection configuration

B.0 Computational interfaces to satisfy simple monitored subnetwork connection configuration community enterprise requirements

The simple monitored subnetwork connection community enterprise requirements are met by the interfaces specified in Annex A, along with the additional interfaces specified in this Annex.

B.1 Label references

This Annex uses the same label references as those used in Annex A together with the following:

Fully-qualified ASN.1 production reference	Local reference used
"M.3100 : 199x : ASN1DefinedTypesModule"::OperationalState	OperationalState

B.2 monitored SNC performer interface – derived from simpleSncPerformerIfce

The monitored SNC performer interface is required to satisfy the enterprise requirements stated in:

```
<"Rec. G.852.1", COMMUNITY:smscc, OBLIGATION:OBLG_6 > ,  
<"Rec. G.852.1", COMMUNITY:smscc, ACTION:sfm3 > ,  
<"Rec. G.852.1", COMMUNITY:smscc, ACTION:sfm4 > .
```

This interface suspends and resumes failure monitoring for a given subnetwork connection.

```
COMPUTATIONAL_INTERFACE monitoredSncPerformerIfce {  
    DERIVED FROM <simpleSncPerformerIfce> ;  
    OPERATION <suspendSNCMonitorReporting> ;  
              <resumeSNCMonitorReporting> ;  
}
```

B.2.1 suspend SNC monitor reporting

```
OPERATION suspendSNCMonitorReporting {  
    INPUT_PARAMETERS  
        subnetwork : SubnetworkId ::= (ssccSnIfce);  
    -- The subnetwork parameter is used to indicate which subnetwork the performer is monitoring. This  
    -- parameter is used, for example, when a given performer can monitor SNCs in many subnetworks. If  
    -- the performer is associated with a single subnetwork, the subnetwork parameter of this operation is  
    -- redundant, and may be removed as an engineering optimization.  
  
        userLabel : UserLabel ;  
  
    OUTPUT_PARAMETERS  
        --none  
  
    RAISED_EXCEPTIONS  
        suspendFailure : NULL ;  
        notConnected : NULL ;  
  
    BEHAVIOUR  
        INFORMAL
```

This operation suspends failure reporting for a given subnetwork connection.

PRE_CONDITIONS

An exception will not be generated if the reportFailureStatus attribute has the value "off".

The user label which is supplied as part of this notification must identify a subnetwork connection which has a unique Id in the domain of the containing subnetwork.

If the SNC does not exist, the exception "notConnected" is generated. If any other pre-condition is not satisfied, the exception "suspendFailure" is generated.

POST_CONDITIONS

The reportFailureStatus attribute has the value "Off", otherwise the exception "suspendFailure" is generated.

SEMI_FORMAL

PARAMETER_MATCHING

```
subnetwork : <ssccConnected, ROLE:involvedSubnetwork> ;
userLabel : <ssccConnected, ROLE:involvedPointToPointSubnetworkConnection, ATTRIBUTE:
userLabel > ;
```

```
PRE_CONDITIONS <ssccConnected> ;
```

```
POST_CONDITIONS
```

```
"inv_1: <ATTRIBUTE:reportFailureStatus, STATE:reportFailureOff >" ;
```

EXCEPTIONS

```
IF PRE_CONDITION <inv_1> NOT_VERIFIED RAISE_EXCEPTION notConnected ;
IF PRE_CONDITION <inv_2> NOT_VERIFIED RAISE_EXCEPTION suspendFailure ;
IF PRE_CONDITION <inv_3> NOT_VERIFIED RAISE_EXCEPTION suspendFailure ;
IF PRE_CONDITION <inv_4> NOT_VERIFIED RAISE_EXCEPTION suspendFailure ;
IF POST_CONDITION <inv_1> NOT_VERIFIED RAISE_EXCEPTION
suspendFailure ;;
```

```
}
```

B.2.2 resume SNC monitor reporting

```
OPERATION resumeSNCMonitorReporting {
```

INPUT_PARAMETERS

```
subnetwork : SubnetworkId ::= (ssccSnIfce);
```

```
-- The subnetwork parameter is used to indicate which subnetwork the performer is monitoring. This
-- parameter is used, for example, when a given performer can monitor SNCs in many subnetworks. If
-- the performer is associated with a single subnetwork, the subnetwork parameter of this operation is
-- redundant, and may be removed as an engineering optimization.
```

```
userLabel : UserLabel ;
```

OUTPUT_PARAMETERS

```
--none
```

RAISED_EXCEPTIONS

```
resumeFailure : NULL ;
```

```
notConnected : NULL ;
```

BEHAVIOUR

INFORMAL

This operation resumes failure reporting for a given subnetwork connection.

PRE_CONDITIONS

An exception will not be generated if the reportFailureStatus attribute has the value "on".

The user label which is supplied as part of this notification must identify a subnetwork connection which has a unique Id in the domain of the containing subnetwork.

If the SNC does not exist, the exception "notConnected" is generated. If any other pre-condition is not satisfied, the exception "suspendFailure" is generated.

POST_CONDITIONS

The reportFailureStatus attribute has the value "On", otherwise the exception "suspendFailure" is generated".

SEMI_FORMAL

PARAMETER_MATCHING

subnetwork : <ssccConnected, ROLE: involvedSubnetwork > ;

userLabel : <ssccConnected, ROLE: involvedPointToPointSubnetworkConnection,
ATTRIBUTE : userLabel > ;

PRE_CONDITIONS <ssccConnected> ;

POST_CONDITIONS

"inv_1: < ATTRIBUTE:reportFailureStatus, STATE: reportFailureOn>" ;

EXCEPTIONS

IF PRE_CONDITION <inv_1> NOT_VERIFIED RAISE_EXCEPTION notConnected ;

IF PRE_CONDITION <inv_2> NOT_VERIFIED RAISE_EXCEPTION resumeFailure ;

IF PRE_CONDITION <inv_3> NOT_VERIFIED RAISE_EXCEPTION resumeFailure ;

IF PRE_CONDITION <inv_4> NOT_VERIFIED RAISE_EXCEPTION resumeFailure ;

IF POST_CONDITION <inv_1> NOT_VERIFIED RAISE_EXCEPTION resumeFailure ;;

}

B.3 SNC monitor interface

The SNC monitor control interface is required to satisfy the enterprise requirements stated in

<"Rec. G.852.1", COMMUNITY:smscc, OBLIGATION:OBLG_6 > ,

COMPUTATIONAL_INTERFACE sncMonitorIfce {

DERIVED FROM <sncIfce> ;

OPERATION <get_snc_operationalState> ;

<get_reportFailureStatus> ;

}

B.3.1 Get SNC operational State

This operation may be implemented by direct manipulation of an attribute if the engineering viewpoint language supports attributes.

OPERATION get_snc_operationalState {

INPUT_PARAMETERS -- none

OUTPUT_PARAMETERS

operationalState_val : OperationalState ;

RAISED_EXCEPTIONS

cannotDetermineValue : NULL ;

BEHAVIOUR

INFORMAL

"This operation retrieves the value of the OperationalState attribute value of a given subnetwork connection.

An exception is generated if this operation fails."

```

SEMI_FORMAL
PARAMETER_MATCHING

operationalState_val : <ATTRIBUTE: operationalState> ;

PRE_CONDITIONS
"inv_1: value of operationalState of SNC is obtainable and valid";

POST_CONDITIONS -- none

EXCEPTIONS
IF PRE_CONDITION <inv_1> NOT_VERIFIED RAISE_EXCEPTION
    cannotDetermineValue ;
;
}

```

B.3.2 Get reportFailureStatus

This operation may be implemented by direct manipulation of an attribute if the engineering viewpoint language supports attributes.

```

OPERATION    get_reportFailureStatus {
    INPUT_PARAMETERS -- none
    OUTPUT_PARAMETERS
        reportFailureStatus_val : ReportFailureStatusType ::=
            ENUMERATED {
                reportFailureOn (0),
                reportFailureOff (1) };

    RAISED_EXCEPTIONS
        cannotDetermineValue : NULL ;
}

```

BEHAVIOUR

INFORMAL

"This operation retrieves the value of the report failure status attribute.

If this value is one, there is no failure reporting for the entity, if this value is zero then failure reporting for the entity is enabled.

An exception is generated if this operation fails."

```

SEMI_FORMAL
PARAMETER_MATCHING

reportFailureStatus_val : <ATTRIBUTE: reportFailureStatus> ;

PRE_CONDITIONS
"inv 1: value of reportFailureStatus of SNC is obtainable and valid " ;

POST_CONDITIONS -- none

EXCEPTIONS

IF PRE_CONDITION <inv_1> NOT_VERIFIED RAISE_EXCEPTION
    cannotDetermineValue ;
;
}

```

B.4 SNC monitor reporting interface

SNC client notification interface. This interface generates notifications when the operational state of the SNC changes.

The SNC monitor reporting interface is required to satisfy the enterprise requirements stated in

```
<"Rec. G.852.1", COMMUNITY:smscc, ACTIONS:sfm1 > ,  
<"Rec. G.852.1", COMMUNITY:smscc, ACTIONS:sfm2 > .
```

```
COMPUTATIONAL_INTERFACE sncMonitorReportingIfce {  
    OPERATION      <reportSNCFailureNotification> ;  
                  <reportSNCRestoreNotification> ;  
}
```

B.4.1 report SNC failure notification

```
OPERATION      reportSNCFailureNotification {  
    INPUT_PARAMETERS  
        existingSNCLabel : UserLabel ;  
    OUTPUT_PARAMETERS -- none  
    RAISED_EXCEPTIONS -- none  
    BEHAVIOUR  
        INFORMAL "If the operational state of the SNC changes from enabled to disabled, a notification is  
        generated. The user label which is supplied as part of this notification must identify a subnetwork  
        connection which has a unique Id in the domain of the containing subnetwork."  
        SEMI_FORMAL  
            PARAMETER_MATCHING  
                existingSNCLabel: <ssccConnected, ROLE: involvedPointToPointSubnetworkConnection,  
ATTRIBUTE:userLabel > ;  
  
        TRIGGERING_CONDITIONS <"Rec. G.853.2", DYNAMIC_SCHEMA:  
reportFailureOnEnabledToDisabled > ;  
  
        EXCEPTIONS -- none;  
}
```

B.4.2 report SNC recovery notification

```
OPERATION      reportSNCRecoveryNotification {  
    INPUT_PARAMETERS  
        existingSNCLabel : UserLabel ;  
    OUTPUT_PARAMETERS -- none  
    RAISED_EXCEPTIONS -- none  
    BEHAVIOUR  
        INFORMAL "If the operational state of the SNC changes from disabled to enabled, a notification is  
        generated. The user label which is supplied as part of this notification must identify a subnetwork  
        connection which has a unique Id in the domain of the containing subnetwork."  
        SEMI_FORMAL  
            PARAMETER_MATCHING  
                existingSNCLabel: <ssccConnected, ROLE: involvedPointToPointSubnetworkConnection,  
ATTRIBUTE:userLabel > ;  
        TRIGGERING_CONDITIONS <"Rec. G.853.2", DYNAMIC_SCHEMA:  
reportFailureOnDisabledToEnabled> ;  
        EXCEPTIONS -- none;  
}
```

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 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 communication
- Series Z Programming languages