

ICS 33.040

M 11

**YD**

# 中华人民共和国通信行业标准

YD/T 1767.5-2009

---

自动交换光网络（ASON）

网络管理技术要求

第5部分：基于IDL/IIOP技术的  
EMS—NMS 接口通用信息模型

Technical requirements for automatically switched optical network  
(ASON) management

Part 5: EMS—NMS interface information model based on IDL/IIOP

2009-12-11 发布

2010-01-01 实施

---

中华人民共和国工业和信息化部 发布

## 目 次

前 言	II
1 范围	1
2 规范性引用文件	1
3 术语、定义和缩略语	1
3.1 术语和定义	1
3.2 缩略语	1
4 接口规范	2
4.1 EMS管理模块 (module emsMgr)	2
4.2 终端点模块 (module terminationPoint)	11
4.3 MLRA管理模块 (module multiLayerSubnetwork)	11
4.4 多层子网点池模块 (module mLSNPP)	43
4.5 多层子网点池链路模块 (module mLSNPPLink)	50
4.6 呼叫模块 (module callSNC)	61
4.7 ASON连接模块 (module subnetworkConnection)	72
4.8 控制通道管理模块 (module scnLink)	76
4.9 发现管理模块 (module discoveringMgr) (可选)	80
4.10 故障管理模块	83
4.11 性能管理模块	83
附录A (规范性附录) 对象管理相关定义	84
附录B (资料性附录) 通知格式定义	85

## 前 言

《自动交换光网络（ASON）网络管理技术要求》分为 5 个部分：

- 第 1 部分：基本原则
- 第 2 部分：NMS 系统功能
- 第 3 部分：EMS—NMS 接口功能
- 第 4 部分：EMS—NMS 接口通用信息模型
- 第 5 部分：基于 IDL/IOP 技术的 EMS—NMS 接口信息模型

本部分为第 5 部分。

本部分附录 A 为规范性附录，附录 B 为资料性附录。

本部分由中国通信标准化协会提出并归口。

本部分起草单位：中兴通讯股份有限公司、北京邮电大学、北京市天元网络技术股份有限公司、武汉邮电科学研究院、华为技术有限公司、工业和信息化部电信研究院。

本部分主要起草人：年庆飞、司 昕、范小磊、强国武、蒙向阳、夏文君、肖延明、张国颖。

广东省网络空间安全协会受控资料

## 自动交换光网络（ASON）网络管理技术要求

### 第5部分：基于IDL/IIOP技术的EMS—NMS接口信息模型

#### 1 范围

本部分规定了ASON网络管理体系EMS-NMS之间基于IDL/IIOP技术的接口规范。

本部分适用于ASON网络管理系统的多厂商互通。

#### 2 规范性引用文件

下列文件中的条款通过本部分的引用而成为本部分的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本部分。然而，鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本部分。

GB/T 21645.1-2008	自动交换光网络（ASON）技术要求 第1部分：体系结构和总体要求
GB/T 21645.2-2008	自动交换光网络（ASON）技术要求 第2部分：术语和定义
YD/T 1767.1-2008	自动交换光网络（ASON）网络管理技术要求 第1部分：基本原则
YD/T 1767.2-2008	自动交换光网络（ASON）网络管理技术要求 第2部分：NMS系统管理功能
YD/T 1767.3-2009	自动交换光网络（ASON）网络管理技术要求 第3部分：EMS—NMS系统接口功能
YD/T 1767.4-2009	自动交换光网络（ASON）网络管理技术要求 第4部分：EMS—NMS接口通用信息模型
ITU-T G.7718	自动交换光网络（ASON）管理框架
TMF513	多技术网络管理事务协定 NML—EML接口 版本3.5
TMF608	多技术网络管理信息协定 NML—EML接口 版本3.5
TMF814	多技术网络管理解决方案 NML—EML接口 版本3.5
TMF814A	多技术网络管实现声明模板和指南 NML—EML接口 版本3.5

#### 3 术语、定义和缩略语

##### 3.1 术语和定义

GB/T 21645.2-2008《自动交换光网络（ASON）技术要求 第2部分：术语和定义》和YD/T 1767.2-2008《自动交换光网络（ASON）网络管理技术要求 第2部分：NMS系统功能》确立的术语和定义适用于本部分。

##### 3.2 缩略语

下列缩略语适用于本部分。

ASON	Automatically Switched Optical Network	自动交换光网络
ASAP	Alarm Severity Assignment Profile	告警级别配置模板
AVC	Attribute Value Change Notification	属性改变通知
CORBA	Common Object Request Broker Architecture	公共对象请求代理结构
EMS	Element Management System	网元管理系统

E-NNI	External Network Node Interface	外部网络节点接口
IDL	Interface Description Language	接口描述语言
IOP	Internet Inter-ORB Protocol	互联网内部对象请求代理协议
I-NNI	Internal Network Node Interface	内部网络节点接口
LayeredSNPP	Layered Subnetwork Point Pool	分层 SNPP
LayeredSNPP Link	Layered Subnetwork Point Pool Link	分层 SNPP 链路
MLRA	MultiLayer Routing Area	多层路由区
NMS	Network Management System	网络管理系统
PC	Permanent Connection	永久连接
RA	Routing Area	路由区
SC	Swiched Connection	交换连接
SCN	Signalling Communications Network	信令通信网络
SDH	Synchronous Digital Hierarchy	同步数字体系
SNP	Subnetwork Point	子网点
SNPP	Subnetwork Point Pool	子网点池
SPC	Soft Permanent Connection	软永久连接
SRG	Shared Risk Group	共享风险组
TNA	Transport Network Assigned	传送网络分配
TP	Termination Point	终端点
UNI	User Network Interface	用户网络接口

#### 4 接口规范

##### 4.1 EMS 管理模块 (module emsMgr)

###### 4.1.1 EMSMgr\_I 接口 (从 common::Common\_I 接口继承)

###### 4.1.1.1 获取 EMS 管理的所有顶级多层路由区 (getAllTopLevelMLRAs)

本操作同TMF814中定义的“getAllTopLevelSubnetworks”。

###### 4.1.1.2 获取 EMS 管理的所有顶级多层路由区名称 (getAllTopLevelMLRANames)

本操作同TMF814中定义的“getAllTopLevelSubnetworkNames”。

## 4.1.1.3 查询 EMS 管理的所有 MLRA (getAllMLRAs)

定 义	
<pre>void getAllMLRAs (     in unsigned long      how_many,     out multiLayerSubnetwork::SubnetworkList_T  mLRAList,     out multiLayerSubnetwork::SubnetworkIterator_I  sIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询 EMS 管理的所有多层路由区
输入参数	how_many: 第一批返回的多层路由区数上限
输入/输出参数	无
输出参数	mLRAList: 第一批返回的多层路由区的列表 sIt: 查询剩余多层路由区信息的迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_TOO_MANY_OPEN_ITERATORS EXCPT_NE_COMM_LOSS

广东省网络空间安全协会受控资料

4.1.1.4 查询所有使用被指定终端点的 MLSNPPLink(getAllMLSNPPLinksWithTP) (可选)

定 义	
<pre>void getAllMLSNPPLinksWithTP(     in globaldefs::NamingAttributes_T tPName,     in boolean sNPListRequested,     in unsigned long how_many,     out MLSNPPLink::MLSNPPLinkList_T mLSNPPLinkList,     out MLSNPPLink::MLSNPPLinkIterator_I mLSNPPLinkIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定终端点，查询所有使用该终端点的多层子网点池链路
输入参数	tPName: 指定的终端点名称 sNPListRequested: 是否必须返回子网点 how_many: 第一批返回的多层子网点池链路数上限
输入/输出参数	无
输出参数	mLSNPPLinkList: 第一批返回的多层子网点池链路列表 mLSNPPLinkIt: 查询剩余多层子网点池链路信息的迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

## 4.1.1.5 查询指定 MLRA 内所有 MLSNPPLink(getAllMLSNPPLinksWithMLSNs)(可选)

定 义	
<pre>void getAllMLSNPPLinksWithMLSNs(     in globaldefs::NamingAttributesList_T mLRANames,     in boolean sNPListRequested,     in unsigned long      how_many,     out MLSNPPLink::MLSNPPLinkList_T mLSNPPLinkList,     out MLSNPPLink::MLSNPPLinkIterator_I mLSNPPLinkIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定 MLRA，查询所有多层子网点池链路
输入参数	<p>mLRANames: 指定的多层路由区的名称列表，返回的多层子网点池链路连接了这里指定的 MLRA 对</p> <p>sNPListRequested: 是否必须返回子网点</p> <p>how_many: 第一批返回的多层子网点池链路数上限</p>
输入/输出参数	无
输出参数	<p>mLSNPPLinkList: 第一批返回的多层子网点池链路列表</p> <p>mLSNPPLinkIt: 查询剩余多层子网点池链路信息的迭代器</p>
操作异常	<p>EXCPT_NOT_IMPLEMENTED</p> <p>EXCPT_INTERNAL_ERROR</p> <p>EXCPT_UNABLE_TO_COMPLY</p> <p>EXCPT_INVALID_INPUT</p> <p>EXCPT_ENTITY_NOT_FOUND</p> <p>EXCPT_NE_COMM_LOSS</p>



4.1.1.6 查询所有与指定 TNA 匹配的 MLSNPPLink(getAllMLSNPPLinksWithTNAs) (可选)

定 义	
<pre>void getAllMLSNPPLinksWithTNAs(     in MLSNPP::TNAList_T tNAList,     in boolean sNPListRequested,     in unsigned long how_many,     out MLSNPPLink::MLSNPPLinkList_T MLSNPPLinkList,     out MLSNPPLink::MLSNPPLinkIterator_I MLSNPPLinkIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定 TNA 名称，查询所有与该 TNA 名称相匹配的多层子网点池链路
输入参数	tNAList: TNA 列表 sNPListRequested: 是否必须返回子网点 how_many: 第一批返回的多层子网点池链路数上限
输入/输出参数	无
输出参数	mLSNPPLinkList: 第一批返回的多层子网点池链路列表 mLSNPPLinkIt: 查询剩余多层子网点池链路信息的迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_NE_COMM_LOSS

## 4.1.1.7 查询 EMS 中所有的多层子网点池(getAllMLSNNPs)

定 义	
<pre>void getAllMLSNNPs(     in boolean sNPListRequested,     in unsigned long how_many,     out mLSNPP::MLSNNPList_T mLSNNPList,     out mLSNPP::MLSNNPIterator_I mLSNNPIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询 EMS 中所有的多层子网点池
输入参数	sNPListRequested: 是否必须返回子网点 how_many: 第一批返回的多层子网点池数上限
输入/输出参数	无
输出参数	mLSNNPList: 第一批返回的多层子网点池列表 mLSNNPIt: 查询剩余多层子网点池信息的迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_NOT_IMPLEMENTED EXCPT_NE_COMM_LOSS

4.1.1.8 查询使用指定终端点的所有多层子网点池(getAllMLSNNPsWithTP)(可选)

定 义	
<pre>void getAllMLSNNPsWithTP(     in globaldefs::NamingAttributes_T tPName,     in boolean sNPListRequested,     in unsigned long how_many,     out MLSNPP::MLSNPPList_T mLSNPPList,     out MLSNPP::MLSNPPIterator_I mLSNPPIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定终端点，查询使用该终端点的所有多层子网点池
输入参数	tPName: 指定的终端点名称 sNPListRequested: 是否必须返回子网点 how_many: 第一批返回的多层子网点池数上限
输入/输出参数	无
输出参数	mLSNPPList: 第一批返回的多层子网点池列表 mLSNPPIt: 查询剩余多层子网点池信息的迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

## 4.1.1.9 查询所有使用指定 TNA 的多层子网点池(getAllMLSNNPsWithTNA)(可选)

定 义	
<pre>void getAllMLSNNPsWithTNA(     in MLSNPP::TNAList_T tNAList,     in boolean sNPListRequested,     in unsigned long how_many,     out MLSNPP::MLSNNPList_T MLSNNPList,     out MLSNPP::MLSNNPIterator_I MLSNNPIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定 TNA 名称，查询所有使用该 TNA 名称的多层子网点池
输入参数	tNAList: TNA 列表 sNPListRequested: 是否必须返回子网点 how_many: 第一批返回的多层子网点池数上限
输入/输出参数	无
输出参数	MLSNNPList: 第一批返回的多层子网点池列表 MLSNNPIt: 查询剩余多层子网点池信息的迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_NE_COMM_LOSS

4.1.1.10 查询 EMS 中所有的 MLSNPPLink(getAllMLSNPPLinks)

定 义	
<pre>void getAllMLSNPPLinks (     in boolean sNPListRequested,     in unsigned long how_many,     out MLSNPPLink::MLSNPPLinkList_T MLSNPPLinkList,     out MLSNPPLink::MLSNPPLinkIterator_I MLSNPPLinkIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询 EMS 中所有的多层子网点池链路，包括边界的和内部的
输入参数	sNPListRequested: 是否必须返回子网点 how_many: 第一批返回的多层子网点池链路数上限
输入/输出参数	无
输出参数	mLSNPPLinkList: 第一批返回的多层子网点池链路列表 mLSNPPLinkIt: 查询剩余多层子网点池链路信息的迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_NOT_IMPLEMENTED EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

## 4.2 终端点模块 (module terminationPoint)

终端点模块是在符合TMF814的基础上，针对于ASON网络管理而新增的。

在TerminationPoint\_T的附加信息 (additionalInfo) 字段中，在已有字段基础上，增加以下定义：

名 称	取 值	说 明
“allocatedToControlPlane”	“True”表示资源被分配到控制平面， “False”表示资源未被分配到控制平面	表示资源是否被分配到控制平面

## 4.3 MLRA 管理模块 (module multiLayerSubnetwork)

### 4.3.1 MLRA

本部分采用TMF814中定义的MultiLayerSubnetwork\_T来表示MLRA，MLRA在multiLayerSubnetwork模块中实现，在MultiLayerSubnetwork\_T的附加信息 (additionalInfo) 字段中，在已有字段基础上，增加以下定义来描述MLRA：

名 称	取 值	说 明
“RoutingAreaLevel”	“TopLevelRA”表示顶级路由区， “InterMediateRA”表示中间级路由区 “MLRN”表示最低级别的多层路由区（路由节点）	表示多层路由区级别
“SupportingMEName”	按层次顺序填写网元名称的名值对，即为 “/name=/value=/name=/value=”形式	当路由点为网元时其所在的网元名称
”SRG”	格式为“/type=value”，其中的 type 和 value 可替换为具体的 SRG 类型及对应的值，可以为空或多个 SRG	共享风险组列表
“LayeredRoutingAreaList”	格式为“/LayerRate=rate-RAid=id”，其中的 rate 和 id 可替换为整数，可以为空或多个 例如：/LayerRate=sts3c_au4-RAid=123	每层的 RA 列表
“SuperiorMLRA”	格式要求同“SupportingMEName”	上一层 MLRA 名称
“SupportingMLSNs”	格式要求同“SupportingMEName”	多层子网名称，当多层子网作为 MLRA 时使用

4.3.2 RAIDList\_T

定 义	
typedef sequence<string> RAIDList_T;	
说 明	
对象说明	RA 标识列表

4.3.3 RouteAndRouteType\_T

定 义	
<pre>struct RouteAndRouteType_T {     RAIDList_T rAIDList;     callSNC::RouteType_T routeType;     string RouteDescription; };</pre>	
说 明	
对象说明	路由和路由类型信息
属性名	属性说明
rAIDList	参与到连接路由的 RA 标识列表
routeType	路由类型
RouteDescription	路由描述符

4.3.4 RoutePerRouteType\_T

定 义	
typedef sequence<RouteAndRouteType_T> RoutePerRouteType_T;	
说 明	
对象说明	路由和路由类型列表

## 4.3.5 MultiLayerSubnetworkMgr\_I 接口（从 common::Common\_I 接口继承）

## 4.3.5.1 查询所有呼叫和顶级连接（getAllCallsAndTopLevelConnections）

定 义	
<pre>void getAllCallsAndTopLevelConnections(     in globaldefs::NamingAttributes_T subnetName,     in unsigned long how_many,     out callSNC::CallAndTopLevelConnectionsList_T callAndTopLevelConnectionsList;     out callSNC::CallAndTopLevelConnectionsIterator_I callAndTopLevelConnectionsIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询由 EMS 管理的、在顶级的 MLRA 范围内的所有的呼叫和每个呼叫的顶级连接
输入参数	subnetName: MLRA 名称 how_many: 第一批返回数据数目
输入/输出参数	无
输出参数	callAndTopLevelConnectionsList: 表示首次查询返回的呼叫和顶级连接列表 callAndTopLevelConnectionsIt: 呼叫和顶级连接迭代查询接口
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS



4.3.5.2 查询所有呼叫和顶级连接及关联的子网连接 (getAllCallsAndTopLevelConnectionsAndSNCs)  
(可选)

定 义	
<pre>void getAllCallsAndTopLevelConnectionsAndSNCs(     in globaldefs::NamingAttributes_T subnetName,     in unsigned long how_many,     out callSNC::CallAndTopLevelConnectionsAndSNCsList_T callAndTopLevelConnectionsAndSNCsList,     out callSNC::CallAndTopLevelConnectionsAndSNCsIterator_I callAndTopLevelConnectionsAndSNCsIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询由 EMS 管理的、在顶级的 MLRA 范围内的所有的呼叫和每个呼叫的顶级连接及关联的子网连接
输入参数	subnetName: MLRA 名称 how_many: 第一批返回数据数目
输入/输出参数	无
输出参数	callAndTopLevelConnectionsAndSNCsList: 表示首次查询返回的呼叫和顶级连接列表及关联的子网连接列表 callAndTopLevelConnectionsAndSNCsIt: 呼叫和顶级连接及关联的子网连接信息迭代查询接口
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

## 4.3.5.3 指定网元查询呼叫和顶级连接 (getAllCallsAndTopLevelConnectionsWithME) (可选)

定 义	
<pre>void getAllCallsAndTopLevelConnectionsWithME(     in globaldefs::NamingAttributes_T subnetName,     in globaldefs::NamingAttributes_T meName,     in unsigned long how_many,     out callSNC::CallAndTopLevelConnectionsList_T callAndTopLevelConnectionsList,     out callSNC::CallAndTopLevelConnectionsIterator_I callAndTopLevelConnectionsIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询由 EMS 管理的、在顶级的 MLRA 范围内的呼叫和顶级连接，其中至少有一个顶级连接的端点关联的终端点在指定网元中
输入参数	subnetName: MLRA 名称 meName: 网元名称 how_many: 第一批返回数据数目
输入/输出参数	无
输出参数	callAndTopLevelConnectionsList: 表示首次查询返回的呼叫和顶级连接列表 callAndTopLevelConnectionsIt: 呼叫和顶级连接迭代查询接口
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

4.3.5.4 指定网元查询呼叫和顶级连接及子网连接 ( getAllCallsAndTopLevelConnectionsAndSNCsWithME ) (可选)

定 义	
<pre>void getAllCallsAndTopLevelConnectionsAndSNCsWithME(     in globaldefs::NamingAttributes_T subnetName,     in globaldefs::NamingAttributes_T meName,     in unsigned long how_many,     out         callSNC::CallAndTopLevelConnectionsAndSNCsList_T callAndTopLevelConnectionsAndSNCsList,     out         callSNC::CallAndTopLevelConnectionsAndSNCsIterator_I callAndTopLevelConnectionsAndSNCsIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询由 EMS 管理的、在顶级的 MLRA 范围内的呼叫和顶级连接及关联的子网连接，其中至少有一个顶级连接的端点关联的终端点在指定网元中
输入参数	subnetName: MLRA 名称 meName: 网元名称 how_many: 第一批返回数据数目
输入/输出参数	无
输出参数	callAndTopLevelConnectionsAndSNCsList: 表示首次查询返回的呼叫和顶级连接列表及关联的子网连接列表 callAndTopLevelConnectionsAndSNCsIt: 呼叫和顶级及关联的子网连接信息迭代查询接口
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

#### 4.3.5.5 指定终端点查询呼叫和顶级连接及子网连接（getAllCallsAndTopLevelConnectionsAndSNCsWithTP）(可选)

定 义	
<pre>void getAllCallsAndTopLevelConnectionsAndSNCsWithTP(     in globaldefs::NamingAttributes_T subnetName,     in globaldefs::NamingAttributes_T tPName,     out                                callSNC::CallAndTopLevelConnectionsAndSNCsList_T callAndTopLevelConnectionsAndSNCsList,     out                                callSNC::CallAndTopLevelConnectionsAndSNCsIterator_I callAndTopLevelConnectionsAndSNCsIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询由 EMS 管理的、在顶级的 MLRA 范围内的呼叫和顶级连接及关联的子网连接，其中至少有一个顶级连接的端点关联的终端点是指定的终端点
输入参数	subnetName: MLRA 名称 tPName: 终端点名称 how_many: 第一批返回数据数目
输入/输出参数	无
输出参数	callAndTopLevelConnectionsAndSNCsList: 表示首次查询返回的呼叫和顶级连接列表及关联的子网连接列表 callAndTopLevelConnectionsAndSNCsIt: 呼叫和顶级及关联的子网连接信息迭代查询接口
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

4.3.5.2 查询连接 (getSNC)

本操作使用TMF814中定义的“getSNC”。

4.3.5.3 指定终端点查询呼叫标识符 (getAllCallIdsWithTP)

定 义	
<pre>void getAllCallIdsWithTP(     in globaldefs::NamingAttributes_T tPName,     out callSNC::CallIdList_T callIdList)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定终端点查询相应的呼叫标识符列表, 这些呼叫相关的连接的端点使用了指定终端点支持的子网点
输入参数	tPName: 终端点名称
输入/输出参数	无
输出参数	callIdList: 呼叫标识符列表
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

广东省网络安全协会受控资料

## 4.3.5.4 指定 TNA 或 SNPP 查询呼叫标识符 (getAllCallIdsWithSNPPOrTNAName)

定 义	
<pre>void getAllCallIdsWithSNPPOrTNAName(     in globaldefs::NameAndStringValue_T SNPPOrTNAName,     out callSNC::CallIdList_T callIdList)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定 TNA 名称或 SNPP 名称查询相应的呼叫标识符列表
输入参数	SNPPOrTNAName: TNA 名称或 SNPP 名称
输入/输出参数	无
输出参数	callIdList: 呼叫标识符列表
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

广东省网络空间安全协会受控资料

4.3.5.5 查询指定呼叫和顶级连接及子网连接 (getCallAndTopLevelConnectionsAndSNCs)

定 义	
<pre>void getCallAndTopLevelConnectionsAndSNCs(     in globaldefs::NamingAttributes_T callName, in string callId,     out callSNC::CallAndTopLevelConnectionsAndSNCs_T callAndTopLevelConnectionsAndSNCs)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定呼叫名称及呼叫 ID 查询相关的呼叫和顶级连接及子网连接
输入参数	callName: 呼叫名称 callId: 呼叫标识
输入/输出参数	无
输出参数	callAndTopLevelConnectionsAndSNCs: 呼叫和顶级连接及子网连接信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NETWORK_COMM_LOSS

广东省网络安全协会受控资料

## 4.3.5.6 建立呼叫 (establishCall)

定 义	
<pre>void establishCall(     in callSNC::CallCreateData_T callCreateData,     in subnetworkConnection::SNCCreateDataList_T connectionCreateDataList,     in string routeGroupsNumber,     inout subnetworkConnection::TPDataList_T tpsToModify,     out callSNC::CallAndTopLevelConnections_T callAndTopLevelConnections,     out subnetworkConnection::SNCCreateDataList_T connectionsNotCreated,     out subnetworkConnection::SubnetworkConnectionList_T partialSNCs,     out string callErrorReason)     raises (globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	创建呼叫
输入参数	<p>callCreateData: 呼叫创建数据</p> <p>connectionCreateDataList: 连接创建数据列表</p> <p>routeGroupsNumber: 路由组数目, 当连接创建数据不提供此信息时有效, “0”表示 NMS 不关心, “NA”表示此参数可被忽略</p>
输入/输出参数	tpsToModify: (输入) 待修改的终端点列表; (输出) 修改后的终端点列表; 具体的修改数据不作要求
输出参数	<p>callAndTopLevelConnections: 呼叫和顶级连接信息</p> <p>connectionsNotCreated: 未创建的连接信息</p> <p>partialSNCs: 部分激活的 SNC 信息</p> <p>callErrorReason: 呼叫创建失败原因</p>
操作异常	<p>EXCPT_INTERNAL_ERROR</p> <p>EXCPT_NOT_IMPLEMENTED</p> <p>EXCPT_INVALID_INPUT</p> <p>EXCPT_ENTITY_NOT_FOUND</p> <p>EXCPT_UNABLE_TO_COMPLY</p> <p>EXCPT_OBJECT_IN_USE</p> <p>EXCPT_USER_LABEL_IN_USE</p> <p>EXCPT_UNSUPPORTED_ROUTING_CONSTRAINTS</p> <p>EXCPT_NE_COMM_LOSS</p>



4.3.5.7 修改呼叫 (modifyCall)

定 义	
<pre>void modifyCall(     in globaldefs::NamingAttributes_T callName,     in callSNC::CallModifyData_T callModifyData,     out callSNC::Call_T modifiedCall)     raises (globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	修改呼叫，能修改的部分属性包括 userLabel,owner,networkAccessDomain, additionalInfo 等
输入参数	callName: 呼叫名称 callModifyData: 待修改的呼叫数据
输入/输出参数	无
输出参数	modifiedCall: 修改后的呼叫数据
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_USERLABEL_IN_USE EXCPT_NE_COMM_LOSS

## 4.3.5.8 释放呼叫 (releaseCall)

定 义	
<pre>void releaseCall(     in globaldefs::NamingAttributes_T callName,     inout subnetworkConnection::TPDataList_T tpsToModify,     out subnetworkConnection::SubnetworkConnectionList_T remainingSNCs,     out string errorReason) raises (globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	释放呼叫
输入参数	callName: 呼叫名称
输入/输出参数	tpsToModify: (输入) 待修改的终端点列表; (输出) 修改后的终端点列表; 具体的修改数据不作要求
输出参数	remainingSNCs: 不能被删除的残余子网连接 errorReason: 错误原因
操作异常	EXCPT_INTERNAL_ERROR EXCPT_NOT_IMPLEMENTED EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

## 4.3.5.9 指定呼叫名称查询呼叫 (getCall)

定 义	
<pre>void getCall(     in globaldefs::NamingAttributes_T callName,     out callSNC::Call_T theCall)     raises (globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定呼叫名称查询呼叫
输入参数	callName: 呼叫名称
输入/输出参数	无
输出参数	theCall: 查询到的呼叫信息
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

## 4.3.5.10 增加呼叫的连接 (addConnections)

定 义	
<pre>void addConnections(     in globaldefs::NamingAttributes_T callName,     in subnetworkConnection::SNCCreateDataList_T connectionsToAdd,     in boolean connectionRouteReArrangementAllowed,     inout subnetworkConnection::TPDataList_T tpsToModify,     out subnetworkConnection::SubnetworkConnectionList_T connectionList,     out subnetworkConnection::SubnetworkConnectionList_T partialSNCs,     out string errorReason) raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	增加呼叫的连接
输入参数	<p>callName: 呼叫名称</p> <p>connectionsToAdd: 待增加的连接列表</p> <p>connectionRouteReArrangementAllowed: 是否允许重新配置连接的路由</p>
输入/输出参数	tpsToModify: (输入) 待修改的终端点列表; (输出) 修改后的终端点列表; 具体的修改数据不作要求
输出参数	<p>connectionList: 增加成功的连接列表</p> <p>partialSNCs: 部分激活的 SNC 列表</p> <p>errorReason: 错误原因</p>
操作异常	<p>EXCPT_NOT_IMPLEMENTED</p> <p>EXCPT_INTERNAL_ERROR</p> <p>EXCPT_INVALID_INPUT</p> <p>EXCPT_ENTITY_NOT_FOUND</p> <p>EXCPT_UNABLE_TO_COMPLY</p> <p>EXCPT_OBJECT_IN_USE</p> <p>EXCPT_USER_LABEL_IN_USE</p> <p>EXCPT_UNSUPPORTED_ROUTING_CONSTRAINTS</p> <p>EXCPT_CAPACITY_EXCEEDED</p> <p>EXCPT_NE_COMM_LOSS</p>

4.3.5.11 减少呼叫的连接 (removeConnections)

定 义	
<pre>void removeConnections (     in globaldefs::NamingAttributes_T callName,     in globaldefs::NamingAttributesList_T connectionNamesList,     inout subnetworkConnection::TPDataList_T tpsToModify,     out subnetworkConnection::SubnetworkConnectionList_T sNCsNotDeleted,     out string errorReason)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	减少呼叫的连接
输入参数	callName: 呼叫名称 connectionNamesList: 连接名称列表
输入/输出参数	tpsToModify: (输入) 待修改的终端点列表; (输出) 修改后的终端点列表; 具体的修改数据不作要求
输出参数	sNCsNotDeleted: 无法删除的 SNC 列表 errorReason: 错误原因
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

## 4.3.5.12 查询指定呼叫的连接和路由信息 (getConnectionsAndRouteDetails) (可选)

定 义	
<pre>void getConnectionsAndRouteDetails(     in string callID,     in globaldefs::NamingAttributes_T mLRAName,     in string sNPOrSNPPID,     in boolean mLSNPPLinkRequested,     in callSNC::RouteType_T routeType,     out callSNC::SNCAndRouteList_T connectionAndRouteList)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定呼叫标识符，查询呼叫相关的顶级连接和路由的详细信息
输入参数	<p>callID: 呼叫标识符</p> <p>mLRAName: MLRA 名称</p> <p>sNPOrSNPPID: 子网点标识符或子网点池标识符</p> <p>mLSNPPLinkRequested: 是否必须返回子网点池链路</p> <p>routeType: 路由类型</p>
输入/输出参数	无
输出参数	connectionAndRouteList: 连接和路由信息列表
操作异常	<p>EXCPT_INTERNAL_ERROR</p> <p>EXCPT_NOT_IMPLEMENTED</p> <p>EXCPT_INVALID_INPUT</p> <p>EXCPT_UNABLE_TO_COMPLY</p> <p>EXCPT_ENTITY_NOT_FOUND</p> <p>EXCPT_NE_COMM_LOSS</p>

4.3.5.13 修改分集和共路由信息 (modifyDiversityAndCorouting)

定 义	
<pre>void modifyDiversityAndCorouting(     in globaldefs::NamingAttributes_T callName,     in callSNC::Diversity_T callDiversity,     in callSNC::RouteGroupInfoList_T routeGroupInfoList,     in boolean connectionRouteReArrangementAllowed,     in string routeGroupsNumber,     inout globaldefs::NVSLList_T additionalInfo,     out callSNC::CallAndTopLevelConnections_T callAndTopLevelConnections)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	修改分集和共路由信息。本命令下发到控制平面后。EMS 需要将分集信息、路由关系组信息下发到控制平面负责路由计算的节点。如果呼叫的 A、Z 点与连接的 A、Z 点不一致，需要将分集信息、路由关系组信息分别下发到每条连接实际计算路由的 ASON 节点，这些节点之间在计算路由时需要相互确认预计算结果来避免冲突
输入参数	<p>callName: 呼叫名称。</p> <p>callDiversity: 呼叫分集信息。</p> <p>routeGroupInfoList: 路由关系组信息列表，可以包含与本呼叫不同源、宿的连接的路由关系组信息。当列表为空时表示删除路由关系组配置。</p> <p>connectionRouteReArrangementAllowed: 是否允许重新分配连接的路由。</p> <p>routeGroupsNumber: 路由关系组数目</p>
输入/输出参数	additionalInfo 附加参数。
输出参数	callAndTopLevelConnections: 呼叫和顶级连接信息
操作异常	<p>EXCPT_INTERNAL_ERROR</p> <p>EXCPT_NOT_IMPLEMENTED</p> <p>EXCPT_UNABLE_TO_COMPLY</p> <p>EXCPT_ENTITY_NOT_FOUND</p> <p>EXCPT_INVALID_INPUT</p> <p>EXCPT_UNSUPPORTED_ROUTING_CONSTRAINTS</p> <p>EXCPT_NE_COMM_LOSS</p>

## 4.3.5.14 查询连接的路由 (getRoute)

参考TMF814中定义的“getRoute”。

## 4.3.5.15 查询指定连接的备用路由 (getBackupRoutes)

定 义	
<pre>void getBackupRoutes(     in globaldefs::NamingAttributes_T sncName,     in string routeId,     in boolean includeHigherOrderCCs,     inout globaldefs::NVSLList_T additionalInfo,     out subnetworkConnection::RouteList_T routeList)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询连接的备用路由信息，如果其中的路由标识符没有指定（空字符串），则返回所有连接的路由，包括期望路由和备用路由
输入参数	sncName: 连接名称 routeId: 路由标识符 includeHigherOrderCCs: 返回数据是否包括服务层交叉连接
输入/输出参数	additionalInfo: 附加信息
输出参数	routeList: 路由列表
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY



4.3.5.16 切换当前路由 (switchRoute)

定 义	
<pre>void switchRoute(     in globaldefs::NamingAttributes_T sncName,     in string routeId,     in subnetworkConnection::GradesOfImpact_T tolerableImpact,     in EMSFreedomLevel_T emsFreedomLevel,     inout subnetworkConnection::TPDataList_T tpsToModify,     inout globaldefs::NVSLList_T additionalInfo,     out subnetworkConnection::SNState_T sncState,     out string errorReason)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	去激活当前的路由，激活输入路由
输入参数	sncName: 连接名称 routeId: 路由标识符 tolerableImpact: 最大可容忍的影响 emsFreedomLevel: EMS 执行操作的最大自由度
输入/输出参数	tpsToModify: 修改参数终端点列表，具体的修改数据不作要求 additionalInfo: 附加信息
输出参数	sncState: 连接状态 errorReason: 错误原因
操作异常	NOT_IMPLEMENTED INTERNAL_ERROR INVALID_INPUT OBJECT_IN_USE ENTITY_NOT_FOUND NE_COMM_LOSS NOT_IN_VALID_STATE UNABLE_TO_COMPLY

## 4.3.5.17 创建新路由 (addRoute)

定 义	
<pre>void addRoute(     in globaldefs::NamingAttributes_T sncName,     in subnetworkConnection::RouteCreateData_T createRoute,     in subnetworkConnection::GradesOfImpact_T tolerableImpact,     in EMSFreedomLevel_T emsFreedomLevel,     out subnetworkConnection::RouteDescriptor_T theRoute ,     out string errorReason)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	创建新路由并关联到指定的连接，创建后的路由处于锁定状态
输入参数	sncName: 连接名称 createRoute: 路由创建信息 tolerableImpact: 最大可容忍的影响 emsFreedomLevel: EMS 执行操作的最大自由度
输入/输出参数	
输出参数	theRoute: 创建成功的路由信息 errorReason: 错误原因
操作异常	NOT_IMPLEMENTED INTERNAL_ERROR INVALID_INPUT ENTITY_NOT_FOUND PROTECTION_EFFORT_NOT_MET UNABLE_TO_COMPLY OBJECT_IN_USE NE_COMM_LOSS UNSUPPORTED_ROUTING_CONSTRAINTS

4.3.5.18 删除路由 (removeRoute)

定 义	
<pre>void removeRoute(     in globaldefs::NamingAttributes_T sncName,     in string routeId,     in EMSFreedomLevel_T emsFreedomLevel,     inout globaldefs::NVSList_T additionalInfo)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	删除连接的路由，待删除的路由不能是期望路由
输入参数	sncName: 连接名称 routeId: 路由标识符 tolerableImpact: 最大可容忍的影响 emsFreedomLevel: EMS 执行操作的最大自由度
输入/输出参数	additionalInfo: 附加信息
输出参数	无
操作异常	NOT_IMPLEMENTED INTERNAL_ERROR INVALID_INPUT ENTITY_NOT_FOUND PROTECTION_EFFORT_NOT_MET UNABLE_TO_COMPLY OBJECT_IN_USE NE_COMM_LOSS UNSUPPORTED_ROUTING_CONSTRAINTS

## 4.3.5.19 设置指定路由为期望路由 (setIntendedRoute)

定 义	
<pre>void setIntendedRoute(     in globaldefs::NamingAttributes_T sncName,     in string routeId,     inout globaldefs::NVSList_T additionalInfo)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	设置指定的路由为期望的路由
输入参数	sncName: 连接名称 routeId: 路由标识符
输入/输出参数	additionalInfo: 附加信息
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

4.3.5.20 查询连接的期望路由 (getIntendedRoute)

定 义	
<pre>void getIntendedRoute(     in globaldefs::NamingAttributes_T sncName,     in boolean includeHigherOrderCCs,     inout globaldefs::NVList_T additionalInfo,     out subnetworkConnection::Route_T route)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询连接的期望路由
输入参数	sncName: 连接名称 includeHigherOrderCCs: 是否返回服务层交叉连接
输入/输出参数	additionalInfo: 附加信息
输出参数	route: 期望路由
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

广东省网络安全协会受控资料

## 4.3.5.21 查询连接经过的下级 RA(getAllSubordinateRAidsWithConnection)(可选)

定 义	
<pre>void getAllSubordinateRAidsWithConnection (     in globaldefs::NamingAttributes_T subnetName,     in globaldefs::NamingAttributes_T connection,     in callSNC::RouteType_T routeType,     out RoutePerRouteType_T routePerRouteType)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询经过指定连接的下级 RA
输入参数	subnetName: 指定的 MLRA 名称 connection: 指定的连接的名称 routeType: 输入的路由类型
输入/输出参数	无
输出参数	routePerRouteType: 返回的下级 RA 信息
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_EXCPT_NOT_IN_VALID_STATE EXCPT_NE_COMM_LOSS

4.3.5.22 查询在顶级的 MLRA 范围内的指定 ID 或名称的呼叫和该呼叫关联的顶级连接 (getCallAndTopLevelConnections)

定 义	
<pre>void getCallAndTopLevelConnections (     in globaldefs::NamingAttributes_T callName,     in string callId,     out callSNC::CallAndTopLevelConnections_T callAndTopLevelConnections)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询在顶级的 MLRA 范围内的指定 ID 或名称的 Call 和该 Call 的顶级连接
输入参数	callName: 指定的呼叫名称 callId: 指定的呼叫标识
输入/输出参数	无
输出参数	callsAndTopLevelConnections: 所有包含的呼叫对象及其相关的顶级连接对象
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

## 4.3.5.23 指定名称查询多层子网点池链路(getMLSNPPLink)

定 义	
<pre>void getMLSNPPLink(     in globaldefs::NamingAttributes_T mLSNPPLinkName,     in boolean sNPListRequested,     out mLSNPPLink::MultiLayerSNPPLink_T theMLSNPPLink)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定多层子网点池链路名称，查询多层子网点池链路
输入参数	mLSNPPLinkName: 指定的唯一多层子网点池链路名称 snpListRequested: 是否必须返回子网点
输入/输出参数	无
输出参数	theMLSNPPLink: 查询到的多层子网点池链路
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_NOT_IMPLEMENTED EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS



4.3.5.24 查询所有的 MLSNPP (getAllMLSNPPs)

定 义	
<pre>void getAllMLSNPPs(     in globaldefs::NamingAttributes_T subnetName,     in boolean sNPListRequested,     in unsigned long how_many,     out MLSNPP::MLSNPPList_T mLSNPPList,     out MLSNPP::MLSNPPIterator_I mLSNPPIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定的 MLRA，查询所有的 MLSNPP
输入参数	subnetName: 指定的 MLRA 名称（可选） sNPListRequested: 是否必须返回子网点 how_many: 第一批返回的多层子网点池数上限
输入/输出参数	无
输出参数	mLSNPPList: 第一批返回的多层子网点池列表 mLSNPPIt: 查询剩余多层子网点池信息的迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

## 4.3.5.25 查询指定 MLRA 内部所有的 MLSNPPLinks (getAllInternalMLSNPPLinks)

定 义	
<pre>void getAllInternalMLSNPPLinks(     in globaldefs::NamingAttributes_T subnetName,     in boolean sNPListRequested,     in unsigned long how_many,     out MLSNPPLink::MLSNPPLinkList_T mLSNPPLinkList,     out MLSNPPLink::MLSNPPLinkIterator_I mLSNPPLinkIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定 MLRA, 查询该 MLRA 内部所有的多层子网点池链路。如果 MLRA 是路由节点, 返回空
输入参数	subnetName: MLRA 名称 sNPListRequested: 是否必须返回子网点 how_many: 第一批返回的多层子网点池链路数上限
输入/输出参数	无
输出参数	mLSNPPLinkList: 第一批返回的多层子网点池链路列表 mLSNPPLinkIt: 查询剩余多层子网点池链路信息的迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

4.3.5.26 查询指定 MLRA 所有的边界 MLSNPPLink(getAllEdgeMLSNPPLinks)

定 义	
<pre>void getAllEdgeMLSNPPLinks(     in globaldefs::NamingAttributes_T subnetName,     in boolean sNPListRequested,     in unsigned long how_many,     out MLSNPPLink::MLSNPPLinkList_T mLSNPPLinkList,     out MLSNPPLink::MLSNPPLinkIterator_I mLSNPPLinkIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定的 MLRA，查询该 MLRA 所有的边界（外部）多层子网点池链路
输入参数	subnetName: MLRA 名称 sNPListRequested: 是否必须返回子网点 how_many: 第一批返回的多层子网点池链路数上限
输入/输出参数	无
输出参数	mLSNPPLinkList: 第一批返回的多层子网点池链路列表 mLSNPPLinkIt: 查询剩余多层子网点池链路信息的迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

## 4.3.5.27 查询所有的子 MLRA(getAllSubordinateMLSNs)

定 义	
<pre>void getAllSubordinateMLSNs(     in globaldefs::NamingAttributes_T subnetName,     in unsigned long how_many,     out SubnetworkList_T subordinateMLSNsList,     out SubnetworkIterator_I subnetIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定 MLRA 名称，查询所有的子 MLRA，包括路由节点
输入参数	subnetName: MLRA 名称 how_many: 第一批返回的子多层路由区数上限
输入/输出参数	无
输出参数	subordinateMLSNsList: 第一批返回的多层路由区列表 subnetIt: 查询剩余子多层路由区信息的迭代器
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

4.3.5.28 查询 EMS 中所有的多层子网点池链路(getAllMLSNPPLinks)

定 义	
<pre>void getAllMLSNPPLinks(     in globaldefs::NamingAttributes_T subnetName,     in boolean sNPListRequested,     in unsigned long how_many,     out mLSNPPLink::MLSNPPLinkList_T mLSNPPLinkList,     out mLSNPPLink::MLSNPPLinkIterator_I mLSNPPLinkIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询 EMS 中所有的多层子网点池链路
输入参数	subnetName: MLRA 名称, 可选 sNPListRequested: 是否必须返回子网点 how_many: 第一批返回的多层子网点池链路数上限
输入/输出参数	无
输出参数	mLSNPPLinkList: 第一批返回的多层子网点池链路列表 mLSNPPLinkIt: 查询剩余多层子网点池链路信息的迭代器
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_UNABLE_TO_COMPLY EXCPT_NOT_IMPLEMENTED EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

## 4.3.5.29 查询 MLRA 信息(getMultiLayerSubnetwork)

定 义	
<pre>void getMultiLayerSubnetwork(     in globaldefs::NamingAttributes_T subnetName,     out MultiLayerSubnetwork_T subnetwork)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	通过 MLRA 名称查询 MLRA 信息
输入参数	subnetName: MLRA 名称
输入/输出参数	无
输出参数	subnetwork: MLRA 信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

## 4.4 多层子网点池模块 (module mLSNPP)

## 4.4.1 LayeredSNPP\_T

定 义	
<pre>struct LayeredSNPP_T {     transmissionParameters::LayerRate_T layerRate;     SNPPList_T SNPPList; }</pre>	
说 明	
对象说明	一组层速率相同的子网点池
属性名	属性说明
layerRate	分层子网点池的速率
SNPPList	分层子网点池包含的子网点

## 4.4.2 LayeredSNPPList\_T

定 义	
<pre>typedef sequence&lt;LayeredSNPP_T&gt; LayeredSNPPList_T;</pre>	
说 明	
对象说明	分层子网点池列表

4.4.3 MultiLayerSNPP\_T

定 义	
<pre> struct MultiLayerSNPP_T {     globaldefs::NamingAttributes_T name;     string userLabel;     string nativeEMSName;     string owner;     terminationPoint::Directionality_T direction;     LayeredSNPPList_T layeredSNPPList;     globaldefs::NVSList_T additionalInfo; };                     </pre>	
说 明	
对象说明	一组分层的子网点池，每个层速率对应一个分层子网点池
属性名	属性说明
name	多层子网点池在 EMS 范围内的唯一名称
userLabel	多层子网点池的用户标签
nativeEMSName	多层子网点池在 EMS 系统上显示的名称
owner	多层子网点池的所有者
direction	子网点池包含的子网点的方向
layeredSNPPList	多层子网点池包含的分层子网点池列表
additionalInfo	附加信息

4.4.4 MLSNPPList\_T

定 义	
<pre> typedef sequence&lt;MultiLayerSNPP_T&gt; MLSNPPList_T;                     </pre>	
说 明	
对象说明	多层子网点池列表

## 4.4.5 SNP\_T

定 义	
<pre>struct SNP_T {     string sNPIId;     globaldefs::NamingAttributes_T tPName;     TNAName_T tNAName; };</pre>	
说 明	
对象说明	子网点，子网点是控制平面中对管理平面的终端点的描述
属性名	属性说明
sNPIId	子网点标识符
tPName	子网点关联的终端点名称
tNAName	子网点的 TNA 名称

## 4.4.6 SNPList\_T

定 义	
<pre>typedef sequence&lt;SNP_T&gt; SNPList_T;</pre>	
说 明	
对象说明	子网点列表

## 4.4.7 SNPP\_T

定 义	
<pre>struct SNPP_T {     string sNPPIId;     SNPList_T sNPList;     TNAName_T tNAName;     TNAName_T groupTNAName;     string rAId; };</pre>	
说 明	
对象说明	子网点池，是基于路由目的而组合在一起的一组子网点
属性名	属性说明
sNPPIId	子网点池的唯一标识符
sNPList	子网点池包含的全部子网点
tNAName	子网点池对象的传送网络分配名称
groupTNAName	传送网分配组的名称
rAId	子网点池对象的路由区的标识符



## 4.4.8 SNPPList\_T

定 义	
typedef sequence<SNPP_T> SNPPList_T;	
说 明	
对象说明	子网点池列表

## 4.4.9 SNPTNAPair\_T

定 义	
<pre>struct SNPTNAPair_T {     string SNPIId;     TNAName_T tNAName; };</pre>	
说 明	
对象说明	定义了子网点标识符和相应的传送网分配地址
属性名	属性说明
SNPIId	子网点标识符
tNAName	TNA 的名称

## 4.4.10 SNPTNAPairList\_T

定 义	
typedef sequence<SNPTNAPair_T> SNPTNAPairList_T;	
说 明	
对象说明	子网点标识符和相应的传送网分配地址对列表

## 4.4.11 SNPTNADData\_T

定 义	
<pre>struct SNPTNADData_T {     string sNPPIId;     transmissionParameters::LayerRate_T layerRate;     string rAId;     SNPTNAPairList_T sNPTNAPairList; };</pre>	
说 明	
对象说明	定义了和子网点标识符相关的信息
属性名	属性说明
sNPPIId	子网点池标识符
layerRate	速率
rAId	RA 的名称
sNPTNAPairList	子网点和相应的 TNA 组的名称列表

## 4.4.12 SNPTNADatList\_T

定 义	
typedef sequence<SNPTNADat_T> SNPTNADatList_T;	
说 明	
对象说明	和子网点标识符相关的信息列表

## 4.4.13 SNPPTNAPair\_T

定 义	
<pre> struct SNPPTNAPair_T {     string sNPPId;     transmissionParameters::LayerRate_T layerRate;     string rAId;     TNAName_T tNAName;     TNAName_T groupTNAName; }; </pre>	
说 明	
对象说明	定义了和子网点池标识符相关的信息
属性名	属性说明
sNPPId	子网点池标识符
layerRate	速率
rAId	RA 的名称
tNAName	TNA 的名称
groupTNAName	TNA 组的名称

## 4.4.14 SNPPTNAPairList\_T

定 义	
typedef sequence<SNPPTNAPair_T> SNPPTNAPairList_T;	
说 明	
对象说明	和子网点池标识符相关的信息列表

## 4.4.15 TNAName\_T

定 义	
typedef string TNAName_T;	
说 明	
对象说明	TNA 名称

## 4.4.16 TNAList\_T

定 义	
typedef sequence<TNAName_T> TNAList_T;	
说 明	
对象说明	TNA 名称列表

## 4.4.17 MLSNPPIterator\_I 接口

## 4.4.17.1 从迭代子中查询数据 (next\_n)

定 义	
boolean next_n(in unsigned long how_many, out MLSNPPList_T mLSNPPList) raises (globaldefs::ProcessingFailureException);	
说 明	
功能描述	从迭代子中查询 $n$ 条记录。当 $how\_many \geq$ 迭代子剩余记录数时，系统在数据取完后应该自动销毁该迭代子对象
输入参数	how_many: 表示首次迭代查询的数据数目
输入/输出参数	无
输出参数	mLSNPPList: 表示首次查询返回的多层子网点池信息
操作异常	EXCPT_INTERNAL_ERROR

## 4.4.17.2 查询迭代子记录条数(getLength)

定 义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说 明	
功能描述	查询迭代子中数据记录的总条数（注意这里指的是迭代子中数据记录的总条数，该值不随在迭代子生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long: 迭代子中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

## 4.4.17.3 销毁迭代子对象(destroy)

定 义	
<pre>void destroy()     raises (globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	销毁迭代子对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

## 4.4.18 MLSNPPMgr\_I 接口 (从 common::Common\_I 接口继承)

## 4.4.18.1 查询指定名称的 MLSNPP (getMLSNPP)

定 义	
<pre>void getMLSNPP(     in globaldefs::NamingAttributes_T mLSNPPName,     out MultiLayerSNPP_T theMLSNPP)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	指定的多层子网点池名称, 查询多层子网点池
输入参数	mLSNPPName: 多层子网点池名称
输入/输出参数	无
输出参数	theMLSNPP: 返回 MLSNPP 信息
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_NOT_IMPLEMENTED EXCPT_ENTITY_NOT_FOUND EXCPT_NE_COMM_LOSS

4.4.18.2 设置指定的 MLSNPP 中一个或者多个的对象的 TNA 名称 (setTNANameForMLSNPP)

定 义	
<pre>void setTNANameForMLSNPP(     in globaldefs::NamingAttributes_T mLSNPPName,     in SNPTNADataList_T sNPTNADataList,     in SNPPTNAPairList_T sNPPTNAPairList,     in string tNAName,     in string tNAGroupName,     out MultiLayerSNPP_T mLSNPP)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	设置指定的 MLSNPP 中一个或者多个的对象的 TNA 名称
输入参数	mLSNPPName : 指定的 MLSNP 名称 sNPTNADataList: 要设置的 SNP 标识符及相应的 TNA 名称的列表 sNPPTNAPairList: 要设置的 SNPP 标识符及相应的 TNA 名称的列表 tNAName: 分配给 MultiLayerSNPP 的 TNA 地址名称 tNAGroupName: 分配给 MultiLayerSNPP 的 TNA 组名称
输入/输出参数	无
输出参数	mLSNPP: 设置了新属性值的 MultiLayerSNPP
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

4.5 多层子网点池链路模块 (module mLSNPPLink)

4.5.1 AvailableCapacity\_T

定 义	
typedef sequence<LayeredCapacity_T> AvailableCapacity_T;	
说 明	
对象说明	连接容量的列表

4.5.2 Capacity\_T

定 义	
typedef string Capacity_T;	
说 明	
对象说明	连接容量信息

## 4.5.3 InterfaceType\_T

定 义	
typedef string InterfaceType_T;	
说 明	
对象说明	定义了接口类型
类型取值	取值说明
UNI	用户网络接口
I-NNI	内部网络节点接口
internal E-NNI	路由区内的外部网络节点接口
external E-NNI	路由区外的外部网络节点接口
UNSPECIFIED	未指明的类型

## 4.5.4 LayeredSNPPLink\_T

定 义	
<pre>struct LayeredSNPPLink_T {     transmissionParameters::LayerRate_T layerRate;     SNPPLinkList_T sNPPLinkList; };</pre>	
说 明	
对象说明	分层的 SNPP 链路
属性名	属性说明
layerRate	层速率
sNPPLinkList	SNPP 链路列表

## 4.5.5 LayeredSNPPLinkList\_T

定 义	
typedef sequence<LayeredSNPPLink_T> LayeredSNPPLinkList_T;	
说 明	
对象说明	分层的 SNPP 链路列表

## 4.5.6 LayeredCapacity\_T

定 义	
<pre>struct LayeredCapacity_T {     transmissionParameters::LayerRate_T layerRate;     Capacity_T capacity; };</pre>	
说 明	
对象说明	指定速率和链接，建立连接的个数容量
属性名	属性说明
layerRate	层速率
capacity	连接在指定速率层次上可以支持连接的能力，以整数形式表示当前可用的容量数目

4.5.7 MultiLayerSNPPLink\_T

定 义	
<pre> struct MultiLayerSNPPLink_T {     globaldefs::NamingAttributes_T name;     string userLabel;     string nativeEMSName;     string owner;     globaldefs::ConnectionDirection_T direction;     globaldefs::NamingAttributes_T aMLRAName;     globaldefs::NamingAttributes_T zMLRAName;     string aTNAName;     string zTNAName;     string aTNAGroupName;     string zTNAGroupName;     LayeredSNPPLinkList_T sNPPLinkList;     InterfaceType_T interfaceType;     globaldefs::NVList_T signallingParameters;     string signallingControllerIdentifier;     SignallingProtocol_T signallingProtocol;     boolean signallingEnabled;     globaldefs::NVList_T cost;     boolean discovered;     globaldefs::NVList_T availability;     string linkSRG;     globaldefs::NVList_T additionalInfo; };                     </pre>	
说 明	
对象说明	不同网络层的控制平面子网点池链路的集合抽象对象
属性名	属性说明
name	多层子网点池链路在 EMS 内的唯一名称
userLabel	多层子网点池链路的用户标签
nativeEMSName	多层子网点池链路在 EMS 系统上显示的名称
owner	多层子网点池链路的所有者
direction	连接的方向
aMLRAName	A 端点的多层路由区

说 明	
zMLRAName	Z 端点的多层路由区
aTNAName	A 端点的 TNA 名称
zTNAName	Z 端点的 TNA 名称
aTNAGroupName	A 端点的 TNA 组名称
zTNAGroupName	Z 端点的 TNA 组名称
sNPPLinkList	子网点池链路列表
interfaceType	连接的接口类型
signallingParameters	信令参数。该值的改变会发通知 AVC
signallingControllerIdentifier	信令控制器标识符。该值的改变会发通知 AVC
signallingProtocol	信令协议。该值的改变会发通知 AVC
signallingEnabled	描述信令使能的标记。该值的改变会发通知 AVC
cost	代表一个或多个矩阵的矢量，每个矩阵描述了一个连接通过路径的选择。该值的改变会发通知 AVC
discovered	该属性确定连接是被发现的或是被配置的
availability	描述多层子网点池链路的一个可用性因子。该值的改变会发通知 AVC
linkSRG	代表指定给多层子网点池链路的共享风险组。该值的改变会发通知 AVC
additionalInfo	附加信息

## 4.5.8 MLSNPPLinkList\_T

定 义	
typedef sequence<MultiLayerSNPPLink_T> MLSNPPLinkList_T;	
说 明	
对象说明	多层子网点池链路列表

## 4.5.9 SignallingProtocol\_T

定 义	
typedef string SignallingProtocol_T;	
说 明	
对象说明	定义了路由区的级别
类型取值	取值说明
"CRLDP"	基于约束路由的标签分发协议
"RSVP"	资源预留协议
"PNNI"	私有网络网络接口



4.5.10 SNPPLink\_T

定 义	
<pre>struct SNPPLink_T {     string SNPPLinkId;     mLSNPP::SNPP_T aEnd;     mLSNPP::SNPP_T zEnd; };</pre>	
说 明	
对象说明	子网点池链路
属性名	属性说明
SNPPLinkId	子网点池链路在控制平面的标识符
aEnd	A 端子网点池
zEnd	Z 端子网点池

4.5.11 SNPPLinkList\_T

定 义	
<pre>typedef sequence&lt;SNPPLink_T&gt; SNPPLinkList_T;</pre>	
说 明	
对象说明	子网点池链路的列表

4.5.12 MLSNPPLinkIterator\_I 接口

4.5.11.1 从迭代子中查询数据 (next\_n)

定 义	
<pre>boolean next_n(in unsigned long how_many,                out MLSNPPLinkList_T mLSNPPLink)     raises (globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	从迭代子中查询 $n$ 条记录。当 $how\_many \geq$ 迭代子剩余记录数时，系统在数据取完后应该自动销毁该迭代子对象
输入参数	$how\_many$ : 表示首次迭代查询的数据数目
输入/输出参数	无
输出参数	$mLSNPPLink$ : 表示首次查询返回的多层子网点池链路信息
操作异常	EXCPT_INTERNAL_ERROR

## 4.5.11.2 查询迭代子记录条数(getLength)

定 义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说 明	
功能描述	查询迭代子中数据记录的总条数（注意这里指的是迭代子中数据记录的总条数，该值不随在迭代子生命周期内是不变的）
输入参数	无
输入/输出参数	无
输出参数	无
返回值	unsigned long: 迭代子中包含的总的记录条数
操作异常	EXCPT_INTERNAL_ERROR

## 4.5.11.3 销毁迭代子对象(destroy)

定 义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说 明	
功能描述	销毁迭代子对象
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	EXCPT_INTERNAL_ERROR

## 4.5.13 MLSNPPLinkMgr\_I 接口（从 common::Common\_I 接口继承）

## 4.5.12.1 查询连接的指定层的可用容量(getAvailableCapacity)

定 义	
void getAvailableCapacity( in globaldefs::NamingAttributes_T mLSNPPLinkName, in transmissionParameters::LayerRate_T layerRate, out AvailableCapacity_T availableCapacity) raises(globaldefs::ProcessingFailureException);	
说 明	
功能描述	查询层可用的连接的容量。如果层速率没指定，则查询全部可用容量
输入参数	mLSNPPLinkName: 指定的多层子网点池链路的唯一名称 layerRate: 连接的层速率
输入/输出参数	无
输出参数	availableCapacity: 连接层速率和支持该层速率的连接
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

## 4.5.12.2 分配信令控制器(assignSignallingController)(可选)

定 义	
<pre>void assignSignallingController(     in globaldefs::NamingAttributes_T mLSNPPLinkName,     in string signallingControllerIdentifier)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	分配信令控制器
输入参数	mLSNPPLinkName: 指定的多层子网点池链路名称 signallingControllerIdentifier: 指定的信令控制器标识符
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

## 4.5.12.3 去分配信令控制器(deassignSignallingController)(可选)

定 义	
<pre>void deassignSignallingController(     in globaldefs::NamingAttributes_T mLSNPPLinkName)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	去分配信令控制器
输入参数	mLSNPPLinkName: 指定的多层子网点池链路名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

## 4.5.12.4 查询信令协议和参数(getSignallingProtocolAndParameters)

定 义	
<pre>void getSignallingProtocolAndParameters(     in globaldefs::NamingAttributes_T mLSNPPLinkName,     out SignallingProtocol_T signallingProtocol,     out globaldefs::NVList_T signallingParameters)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询信令协议和信令参数
输入参数	mLSNPPLinkName: 指定的多层子网点池链路名称
输入/输出参数	无
输出参数	signallingProtocol: 使用的协议名称, 除说明的协议名称外, 厂商可根据实现扩展 signallingParameters: 使用的信令控制参数
操作异常	NOT_IMPLEMENTED. INTERNAL_ERROR. INVALID_INPUT ENTITY_NOT_FOUND. UNABLE_TO_COMPLY NE_COMM_LOSS

## 4.5.12.5 设置信令协议和参数(setSignallingProtocolAndParameters)(可选)

定 义	
<pre>void setSignallingProtocolAndParameters(     in globaldefs::NamingAttributes_T mLSNPPLinkName,     in SignallingProtocol_T signallingProtocol,     in globaldefs::NVList_T signallingParameters)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	设置信令协议和信令参数
输入参数	mLSNPPLinkName: 指定的多层子网点池链路名称 signallingProtocol: 使用的协议名称, 厂商可根据实现扩展 signallingParameters: 使用的信令控制参数
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

## 4.5.12.6 设置 TNA 名称(setTNANameForMLSNPPLinkEnd)

定 义	
<pre>void setTNANameForMLSNPPLinkEnd(     in globaldefs::NamingAttributes_T mLSNPPLinkName,     in mLSNPP::SNPTNADataList_T sNPTNADataList,     in mLSNPP::SNPPTNAPairList_T sNPPTNAPairList,     in string aTNAName,     in string zTNAName,     in string aTNAGroupName,     in string zTNAGroupName,     out MultiLayerSNPPLink_T mLSNPPLink)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	设置 TNA 名称, 新的名称将改写原来存在的名称。当提供空值输入时, 表示取消 TNA 名称
输入参数	<p>mLSNPPLinkName: 指定的 MLSNPPLink 名称</p> <p>sNPTNADataList: 要设置的 SNP 标识符和相应的 TNA 名称</p> <p>sNPPTNAPairList: 要设置的 SNPP 标识符和相应的 TNA 名称</p> <p>aTNAName: 指定到 A 端点的 TNA 名称</p> <p>zTNAName: 指定到 Z 端点的 TNA 名称</p> <p>aTNAGroupName: 指定到 A 端点的 TNA 组名称</p> <p>zTNAGroupName: 指定到 Z 端点的 TNA 组名称</p>
输入/输出参数	无
输出参数	mLSNPPLink: 修改后的 MLSNPPLink
操作异常	<p>EXCPT_NOT_IMPLEMENTED</p> <p>EXCPT_INTERNAL_ERROR.</p> <p>EXCPT_INVALID_INPUT</p> <p>EXCPT_ENTITY_NOT_FOUND</p> <p>EXCPT_UNABLE_TO_COMPLY</p> <p>EXCPT_NE_COMM_LOSS</p>

## 4.5.12.7 修改信令协议参数(modifySignallingProtocolParameters)(可选)

定 义	
<pre>void modifySignallingProtocolParameters(     in globaldefs::NamingAttributes_T mLSNPPLinkName,     inout globaldefs::NVSList_T signallingParameters)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	修改信令协议参数。当信令使能或非使能时，均能操作
输入参数	mLSNPPLinkName: 指定的多层子网点池链路名称
输入/输出参数	signallingParameters: (输入) 需修改的信令参数; (输出) 修改后的信令参数
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

## 4.5.12.8 使能信令(enableSignalling)

定 义	
<pre>void enableSignalling(     in globaldefs::NamingAttributes_T mLSNPPLinkName)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	使能信令
输入参数	mLSNPPLinkName: 指定的多层子网点池链路名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

4.5.12.9 禁止信令(disableSignalling)

定 义	
<pre>void disableSignalling(     in globaldefs::NamingAttributes_T mLSNPPLinkName)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	禁止信令
输入参数	mLSNPPLinkName: 指定的多层子网点池链路名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

广东省网络空间安全协会受控资料

## 4.6 呼叫模块 (module callSNC)

## 4.6.1 Call\_T

定 义	
<pre> struct Call_T {     globaldefs::NamingAttributes_T callName;     string userLabel;     string owner;     string networkAccessDomain;     string nativeEMSName;     string callId;     CallState_T callState;     CallEnd_T aEnd;     CallEnd_T zEnd;     CallParameterProfile_T callParameters;     Diversity_T callDiversity;     string diversitySynthesis;     DiversityInfo_T linkDiversityViolations;     DiversityInfo_T nodeDiversityViolations;     DiversityInfoList_T linkPartialDiversityList;     DiversityInfoList_T nodePartialDiversityList;     globaldefs::NVList_T callAdditionalInfo; }; </pre>	
说 明	
对象说明	用于描述呼叫信息
属性名	属性说明
callName	呼叫名称
userLabel	用户标签
owner	呼叫对象拥有者标签
networkAccessDomain	网络接入域名称
nativeEMSName	EMS 本地名称
callId	呼叫的控制平面标识符
callState	呼叫的状态
aEnd	呼叫的 A 端点
zEnd	呼叫的 Z 端点
callParameters	呼叫的参数
callDiversity	呼叫的分集信息
diversitySynthesis	分集违反的类型值 "NotApplicable": 呼叫对象与分集违反无关 "Unknown": 呼叫对象与分集违反的相关未知 "NoViolations": 呼叫对象没有违反分集 "Violations": 呼叫对象有分集违反, 但是不知道是否与节点或链接相关 "LinkViolations": 呼叫对象有和链接相关的分集违反 "NodeViolations": 呼叫对象有和节点相关的分集违反 "LinkAndNodeViolations": 呼叫对象有和链接和节点相关的分集违反
linkDiversityViolations	链接的被违反的分集类型
nodeDiversityViolations	节点的被违反的分集类型
linkPartialDiversityList	当分集需要的未满足的情况下, 链接分集部分实现
nodePartialDiversityList	当分集需要的未满足的情况下, 节点分集部分实现
callAdditionalInfo	附加信息



#### 4.6.2 CallAndTopLevelConnections\_T

定 义	
<pre>struct CallAndTopLevelConnections_T {     Call_T theCall;     subnetworkConnection::SubnetworkConnectionList_T topLevelConnectionsList; };</pre>	
说 明	
对象说明	用于描述呼叫及顶级连接列表
属性名	属性说明
theCall	呼叫信息
topLevelConnectionsList	顶级连接列表

#### 4.6.3 CallAndTopLevelConnectionsAndSNCs\_T

定 义	
<pre>struct CallAndTopLevelConnectionsAndSNCs_T {     Call_T theCall;     ConnectionsAndSupportingSNCsList_T topLevelConnectionsAndSNCsList; };</pre>	
说 明	
对象说明	用于描述呼叫及连接和支持连接的子网连接信息
属性名	属性说明
theCall	呼叫信息
topLevelConnectionsAndSNCsList	连接和支持连接的子网连接信息的列表

#### 4.6.4 CallAndTopLevelConnectionsList\_T

定 义	
<pre>typedef sequence&lt;CallAndTopLevelConnections_T&gt; CallAndTopLevelConnectionsList_T;</pre>	
说 明	
对象说明	用于描述呼叫和顶级连接信息的列表

#### 4.6.5 CallAndTopLevelConnectionsAndSNCsList\_T

定 义	
<pre>typedef sequence&lt;CallAndTopLevelConnectionsAndSNCs_T&gt; CallAndTopLevelConnectionsAndSNCsList_T;</pre>	
说 明	
对象说明	用于描述呼叫和顶级连接及子网连接信息的列表

## 4.6.6 CallCreateData\_T

定 义	
<pre> struct CallCreateData_T {     globaldefs::NamingAttributes_T callName;     string userLabel;     boolean forceUniqueness;     string owner;     string networkAccessDomain;     CallEnd_T aEnd;     CallEnd_T zEnd;     CallParameterProfile_T callParameters;     Diversity_T callDiversity;     globaldefs::NVList_T additionalCreationInfo; }; </pre>	
说 明	
对象说明	用于描述呼叫的创建信息
属性名	属性说明
callName	呼叫名称
userLabel	用户标签
forceUniqueness	友好名称是否全局唯一
owner	呼叫对象拥有者标签
networkAccessDomain	网络接入域名称
aEnd	呼叫的 A 端点
zEnd	呼叫的 Z 端点
callParameters	呼叫的参数
callDiversity	呼叫的分集信息
additionalCreationInfo	附加信息

4.6.7 CallEnd\_T

定 义	
<pre>struct CallEnd_T {     mLSNPP::TNAName_T tNANameOrGroupTNAName;     string sNPPid;     string sNPid;     globaldefs::NamingAttributes_T tpName; };</pre>	
说 明	
对象说明	用于描述呼叫的端点
属性名	属性说明
tNANameOrGroupTNAName	TNA 名称或 TNA 组名称
sNPPid	子网点池标识符
sNPid	子网点标识符
tpName	呼叫 A/Z 端点的终端点名称

4.6.8 CallId\_T

定 义	
<pre>typedef string CallId_T;</pre>	
说 明	
对象说明	用于描述呼叫的控制平面标识符

4.6.9 CallIdList\_T

定 义	
<pre>typedef sequence&lt;CallId_T&gt; CallIdList_T;</pre>	
说 明	
对象说明	用于描述呼叫的控制平面标识符列表

## 4.6.10 CallModifyData\_T

定 义	
<pre>struct CallModifyData_T {     string userLabel;     boolean forceUniqueness;     string owner;     string networkAccessDomain;     globaldefs::NVSList_T additionalModificationInfo; };</pre>	
说 明	
对象说明	用于描述呼叫的修改信息
属性名	属性说明
userLabel	用户标签
forceUniqueness	友好名称是否全局唯一
owner	呼叫对象拥有者标签
networkAccessDomain	网络接入域名称
additionalModificationInfo	附加信息

## 4.6.11 CallParameterProfile\_T

定 义	
<pre>struct CallParameterProfile_T {     string severelyDegradedThreshold;     string degradedTreshold;     string classOfService;     globaldefs::NVSList_T classOfServiceParameters; };</pre>	
说 明	
对象说明	用于描述呼叫的参数
属性名	属性说明
severelyDegradedThreshold	呼叫严重劣化的门限
degradedTreshold	呼叫的劣化门限
classOfService	业务分类
classOfServiceParameters	业务分类参数

## 4.6.12 CallState\_T

定 义	
typedef string CallState_T;	
说 明	
对象说明	用于描述呼叫的状态
类型取值	取值说明
"IN_PROGRESS"	呼叫的连接已被创建，但对控制平面不可用
"ESTABLISHED_IN_SERVICE"	呼叫的所有连接已被成功创建
"ESTABLISHED_IN_SERVICE_SEARCHING"	呼叫已创建了所有的连接对象，但不是所有的连接对象都已经创建成功
"ESTABLISHED_OUT_OF_SERVICE"	呼叫的所有连接对象都创建失败，而且还没有还原
"ESTABLISHED_OUT_OF_SERVICE_SEARCHING"	呼叫的所有连接对象都创建失败，而且正在还原
"ESTABLISHED_IN_SERVICE_DEGRADED"	失败的连接对象个数已经达到或超过呼叫的劣化门限，而还没有达到或超过严重劣化门限
"ESTABLISHED_IN_SERVICE_SEVERELY_DEGRADED"	失败的连接对象个数已经达到或超过呼叫的严重劣化门限
"ESTABLISHED_IN_SERVICE_DEGRADED_SEARCHING"	至少一个连接对象失败，但并且还未达到或超过呼叫的严重劣化门限。失效的连接正在恢复
"ESTABLISHED_IN_SERVICE_SEVERELY_DEGRADED_SEARCHING"	失败的连接对象数量已经达到或者超过呼叫的严重劣化门限。失效的连接正在恢复

## 4.6.13 ConnectionAndSupportingSNCs\_T

定 义	
<pre>struct ConnectionAndSupportingSNCs_T {     subnetworkConnection::SubnetworkConnection_T topLevelConnection;     subnetworkConnection::SubnetworkConnectionList_T sNCList; };</pre>	
说 明	
对象说明	用于描述连接和支持连接的子网连接信息
属性名	属性说明
topLevelConnection	顶级连接
sNCList	支持连接的子网连接列表

## 4.6.14 ConnectionsAndSupportingSNCsList\_T

定 义	
typedef sequence<ConnectionAndSupportingSNCs_T> ConnectionsAndSupportingSNCsList_T;	
说 明	
对象说明	用于描述连接和支持连接的子网连接信息的列表

## 4.6.15 Diversity\_T

定 义	
<pre>struct Diversity_T {     LevelOfEffort_T coroutingLevelOfEffort;     LevelOfEffort_T nodeDiversityLevelOfEffort;     LevelOfEffort_T linkDiversityLevelOfEffort;     string nodeSRGType;     string linkSRGType; };</pre>	
说 明	
对象说明	分集数据
属性名	属性说明
coroutingLevelOfEffort	一个路由组中的顶级连接或子网连接是否需要共路由
linkDiversityLevelOfEffort	呼叫中的路由组是否有链路分集需求
nodeDiversityLevelOfEffort	呼叫中的路由组是否有节点分集需求
linkSRGType	链路分集 SRG 类型
nodeSRGType	节点分集 SRG 类型

## 4.6.16 DiversityInfo\_T

定 义	
<pre>struct DiversityInfo_T {     string sRGType;     SharedResourceList_T sharedResourceList; };</pre>	
说 明	
对象说明	分集信息
属性名	属性说明
srgType	SRG 类型
sharedResourceList	共享资源列表

## 4.6.17 DiversityInfoList\_T

定 义	
typedef sequence<DiversityInfo_T> DiversityInfoList_T;	
说 明	
对象说明	用于描述分集信息列表

## 4.6.18 LevelOfEffort\_T

定 义	
typedef string LevelOfEffort_T;	
说 明	
对象说明	用于完成请求的尽力级别
类型取值	取值说明
" NONE "	无级别
" BEST_EFFORT "	尽力完成
" MANDATORY "	强制要求完成

## 4.6.19 RouteGroupInfo\_T

定 义	
<pre>struct RouteGroupInfo_T {     globaldefs::NamingAttributes_T sNCname;     string routeGroupLabel; };</pre>	
说 明	
对象说明	用于描述与连接关联的路由信息
属性名	属性说明
sNCname	连接名称
routeGroupLabel	路由组用户标签

## 4.6.20 RouteGroupInfoList\_T

定 义	
typedef sequence<RouteGroupInfo_T> RouteGroupInfoList_T;	
说 明	
对象说明	用于描述与连接关联的路由信息列表

## 4.6.21 RouteType\_T

定 义	
typedef string RouteType_T;	
说 明	
对象说明	用于描述路由的类型
类型取值	取值说明
"HOME/INTENDED"	预期的路由
"ACTUAL/CURRENT"	实际的（当前的）路由
"BOTH"	既是预期的又是当前的路由
"BACKUP"	预置恢复路由

## 4.6.22 SharedResource\_T

定 义	
<pre>struct SharedResource_T {     globaldefs::NamingAttributes_T name;     globaldefs::NamingAttributesList_T connectionNameList; };</pre>	
说 明	
对象说明	用于描述可共享资源的连接名称列表
属性名	属性说明
name	被共享的节点或链接的名称
connectionNameList	共享节点或链接的连接对象的列表

## 4.6.23 SharedResourceList\_T

定 义	
typedef sequence<SharedResource_T> SharedResourceList_T;	
说 明	
对象说明	用于描述共享资源对象列表



## 4.6.24 SNCAndRoute\_T

定 义	
<pre>struct SNCAndRoute_T {     subnetworkConnection::SubnetworkConnection_T connection;     subnetworkConnection::RouteList_T connectionRoutes;     mLSNPPLink::MLSNPPLinkList_T edgeMLSNPPLinks;     mLSNPPLink::MLSNPPLinkList_T internalMLSNPPLinks;     RouteType_T routeType; };</pre>	
说 明	
对象说明	用于描述连接和路由信息
属性名	属性说明
connection	连接信息
connectionRoutes	连接的路由信息
edgeMLSNPPLinks	外部多层子网点池链路
internalMLSNPPLinks	内部多层子网点池链路
routeType	路由类型

## 4.6.25 SNCAndRouteList\_T

定 义	
<pre>typedef sequence &lt;SNCAndRoute_T&gt; SNCAndRouteList_T;</pre>	
说 明	
对象说明	用于描述连接和路由信息列表

## 4.6.26 CallAndTopLevelConnectionsIterator\_I 接口

## 4.6.26.1 从迭代器中查询数据 (next\_n)

定 义	
<pre>boolean next_n(in unsigned long how_many,                out CallAndTopLevelConnectionsList_T callAndTopLevelConnectionsList)                raises (globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	从迭代子中查询 $n$ 条记录
输入参数	how_many: 表示首次迭代查询的数据数目
输入/输出参数	无
输出参数	callAndTopLevelConnectionsList: 表示首次查询返回的呼叫和顶级连接信息
操作异常	INTERNAL_ERROR

## 4.6.26.2 查询迭代器保存的数据数目 (getLength)

定 义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说 明	
功能描述	查询迭代器保存的数据数目，这里的数目指首次查询返回后的数目，如总共有 100 条数据，第一批返回了 10 条，本操作返回 90，不论这 90 条中有多少条已经被 next_n 操作迭代查询过了，使用返回值返回
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	INTERNAL_ERROR

## 4.6.26.3 删除迭代器 (destroy)

定 义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说 明	
功能描述	删除迭代器
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	INTERNAL_ERROR

## 4.6.27 CallAndTopLevelConnectionsAndSNCsIterator\_I 接口

## 4.6.27.1 从迭代器中查询数据 (next\_n)

定 义	
boolean next_n(in unsigned long how_many, out CallAndTopLevelConnectionsAndSNCsList_T callAndTopLevelConnectionsAndSNCsList) raises (globaldefs::ProcessingFailureException);	
说 明	
功能描述	从迭代子中查询 $n$ 条记录
输入参数	how_many: 表示首次迭代查询的数据数目
输入/输出参数	无
输出参数	callAndTopLevelConnectionsAndSNCsList: 表示首次查询返回的呼叫和顶级连接以及关联的子网连接信息
操作异常	INTERNAL_ERROR

## 4.6.27.2 查询迭代器保存的数据数目 (getLength)

定 义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说 明	
功能描述	查询迭代器保存的数据数目，这里的数目指首次查询返回后的数目，如总共有 100 条数据，第一批返回了 10 条，本操作返回 90，不论这 90 条中有多少条已经被 next_n 操作迭代查询过了，使用返回值返回
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	INTERNAL_ERROR

## 4.6.27.3 删除迭代器 (destroy)

定 义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说 明	
功能描述	删除迭代器
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	INTERNAL_ERROR

## 4.7 ASON 连接模块 (module subnetworkConnection)

## 4.7.1 CrossConnect\_T

本部分采用TMF814中定义的CrossConnect\_T，在附加信息 (additionalInfo) 字段中，增加以下定义：

名 称	取 值	说 明
"ConnectionId"	字符串	控制平面的连接名称
"RouteExclusive"	"True", 独占; "False" 非独占	表示是否其他连接的路由可以共享交叉连接，即路由是否是独占的
"RouteId"	字符串	路由标识

## 4.7.2 RouteDescriptor\_T

定 义	
<pre>struct RouteDescriptor_T {     string id;     string intended;     string actualState;     string administrativeState;     string inUseBy;     string exclusive;     Route_T routeXCs;     globaldefs::NVSList_T additionalInfo; };</pre>	
说 明	
对象说明	路由描述符信息
属性名	属性说明
id	标识符
intended	是否为期望路由，取值：“y”，“n”
actualState	状态，取值：“inactive”表示包含的交叉连接未激活；“active”：表示包含的交叉连接已激活；“partial”：表示包含的交叉连接部分激活
administrativeState	管理状态，取值“locked”表示路由不能被激活；“unlocked”表示路由可以被激活
inUseBy	是否已被其他连接使用，取值：“y”，“n”
exclusive	是否独占，取值：“y”，“n”
routeXCs	包含的交叉连接
additionalInfo	附加信息

## 4.7.3 RouteList\_T

定 义	
<pre>typedef sequence&lt;RouteDescriptor_T&gt; RouteList_T;</pre>	
说 明	
对象说明	路由描述符列表

## 4.7.4 SNCCreateData\_T

本部分采用TMF814中定义的SubnetworkConnection\_T来表示ASON连接，采用SNCCreateData\_T来表示ASON连接创建数据（ConnectionCreateData），在SubnetworkConnection\_T及SNCCreateData\_T的附加信息（additionalCreationInfo）字段中，增加以下定义：

名 称	取 值	说 明
“Connection”	字符串	连接名称
“MaximumCost”	整数数字	选择路由时允许的最大的链路或节点开销
“NetworkAccessDomain”	字符串	网络接入域名称
“NetworkReroute”	字符串，取值为“Yes”，“No”	子网连接的重路由是否在网络层计算
“RouteGroupLabel”	字符串	路由组标签
“RoutingConstraintEffort”	“BEST_EFFORT”，表示尽力 “EXACT_MATCH”表示严格匹配	路由限制的尽力程度
“ConnectionSetUpType”	“PC”，表示永久连接 “SC”，表示交换连接 “SPC”，表示软永久连接	连接的建立类型
“AlarmReporting”	“On”，表示开启 “Off”，表示关闭	告警上报是否开启
“ASAPpointer”	ASAP 名值对的值部分	ASAP 名称
“SNC_INTENDED_ROUTE_EXCLUSIVE”	“True”，“False”	是否期望路由是独占的，如果是独占的，创建连接时需要判断期望路由是否与其他连接共享，如果共享，则创建不成功
“SNC_PRIORITY”	整型数字，从 0 开始，0 表示最高	子网连接优先级
“SNC_REVERTIVE”	“True”，“False”	表示连接是否恢复

## 4.7.5 SubnetworkConnection\_T

本部分采用TMF814中定义的SubnetworkConnection\_T来表示ASON连接（Connection），在SubnetworkConnection\_T的附加信息（additionalInfo）字段中，增加以下定义：

名称	取值	说明
“A<n>Role”	“CMEndPoint”表示 CP 点, “LCEndPoint”表示 TCP 点	其中的 $n$ 对应与 SubnetworkConnection_T 中 aEnd 中的一个端点, $n$ 为一个整数数字, 从 0 开始
“Z<n>Role”	同上	其中的 $n$ 对应与 SubnetworkConnection_T 中 zEnd 中的一个端点, $n$ 为一个整数数字, 从 0 开始
“NetworkReroute”	“Yes”, “No”, “NotSet”	表示网络层或 EMS 是可以重新计算路由
“ProtectionEffort”	“EFFORT_WHATEVER”, “EFFORT_SAME_OR_BETTER”, “EFFORT_SAME_OR_WORSE”, “EFFORT_SAME”	与 subnetworkConnection::Protection Effort_T 的定义相同, 为了保存创建连接时的保护尽力程度配置增加此参数
“RoutingConstraintEffort”	“BEST_EFFORT” 表示连接对象的路由尽力满足限制条件, “EXACT_MATCH”连接对象的路由必须严格满足限制条件	路由限制的尽力程度
“Connection”	字符串	连接名称
“MaximumCost”	数字形式字符串	连接的最大开销
“NetworkAccessDomain”	字符串	网络接入域(Network Access Domain)名称
“RouteGroupLabel”	字符串	路由组标签
“ConnectionSetUpType”	“PC”表示永久连接, “SPC”表示软永久连接, “SC”表示交换连接	连接的建立类型, 可以通过修改此参数进行连接建立类型的转换
“CallId”	字符串	呼叫名称, 用于控制平面
“ConnectionId”	字符串	连接名称, 用于控制平面
“ConnectionState”	“Complete”表示已全部建立, “Searching”表示正在搜索建立连接的资源 “NotApplicable”表示连接对象不能在控制平面应用	连接的状态
“UsingHomeRoute”	“True”, “False”	是否使用期望路由
“AlarmReporting”	“True”, “False”	是否上报告警
“SNC_PRIORITY”	“0”, “1”, “2”..., 其中 0 为最高优先级	连接的优先级
“SNC_REVERTIVE”	“True”, “False”	是否自动回复
“ASAPpointer”	ASAP 名值对中的值部分	ASAP 名称

4.8 控制通道管理模块 (module scnLink)

4.8.1 SCNLink\_T

定 义	
<pre> struct SCNLink_T {     globaldefs::NamingAttributes_T name;     string userLabel;     string owner;     string networkAccessDomain;     string nativeEMSName;     globaldefs::NamingAttributes_T aEnd ;     globaldefs::NamingAttributes_T zEnd ;     string scnType;     string scnState;     SCNMode_T scnMode;     globaldefs::NVSList_T additionalInfo; };                     </pre>	
说 明	
对象说明	控制通道
属性名	属性说明
name	控制通道名称
userLabel	用户标签
owner	对象所有者标签
networkAccessDomain	网络接入域名称
nativeEMSName	EMS 本地名称
aEnd	链路的 A 端点
zEnd	链路的 Z 端点
scnType	链路的类型, 取值为: "OSC", "FE", "GE", "DCC"(DCCR+DCCM), "DCCR", "DCCM"
scnState	链路的状态, 取值为: "Normal"表示连通, "Link Failure"表示断开
scnMode	链路模式
additionalInfo	附加信息

## 4.8.2 SCNLinkList\_T

定 义	
typedef sequence< SCNLink_T > SCNLinkList_T;	
说 明	
对象说明	控制通道列表

## 4.8.3 SCNMode\_T

定 义	
enum SCNMode_T { SCNM_INBAND, SCNM_OUTBAND };	
说 明	
对象说明	定义了信令通信网模式
类型取值	取值说明
SCNM_INBAND	带内方式 (包括 DCC/DCCR/DCCM/等)
SCNM_OUTBAND	带外方式(包括 OSC/FE/GE 等)

## 4.8.4 SCNLinkIterator\_I 接口

## 4.8.4.1 从迭代器中查询数据 (next\_n)

定 义	
boolean next_n(in unsigned long how_many, out SCNLinkList_T scnlinkList)  raises (globaldefs::ProcessingFailureException);	
说 明	
功能描述	从迭代子中查询 n 条记录
输入参数	how_many: 表示首次迭代查询的数据数目
输入/输出参数	无
输出参数	scnlinkList: 表示首次查询返回的控制通道
操作异常	INTERNAL_ERROR



4.8.4.2 查询迭代器保存的数据数目 (getLength)

定 义	
unsigned long getLength() raises (globaldefs::ProcessingFailureException);	
说 明	
功能描述	查询迭代器保存的数据数目，这里的数目指首次查询返回后的数目，如总共有 100 条数据，第一批返回了 10 条，本操作返回 90，不论这 90 条中有多少条已经被 next_n 操作迭代查询过了，使用返回值返回
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	INTERNAL_ERROR

4.8.4.3 删除迭代器 (destroy)

定 义	
void destroy() raises (globaldefs::ProcessingFailureException);	
说 明	
功能描述	删除迭代器
输入参数	无
输入/输出参数	无
输出参数	无
操作异常	INTERNAL_ERROR

## 4.8.5 SCNManager\_I 接口 (从 common::Common\_I 接口继承)

## 4.8.5.1 查询所有控制通道连接 (getAllSCNLinks)

定 义	
<pre>void getAllSCNLinks(     in unsigned long how_many,     out SCNLinkList_T scnlinkList,     out SCNLinkIterator_I scnlinksIt)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询所有控制通道连接
输入参数	how_many: 第一批返回数据数目
输入/输出参数	无
输出参数	scnlinkList: 首次迭代返回的控制通道列表 scnlinksIt: 迭代查询控制通道接口
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS EXCPT_TOO_MANY_OPEN_ITERATORS

## 4.8.5.2 查询指定的控制通道 (getSCNLink)

定 义	
<pre>void getSCNLink (     in globaldefs::NamingAttributes_T scnName,     out SCNLink_T scnlink)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询指定的控制通道连接
输入参数	scnName: 控制通道名称
输入/输出参数	无
输出参数	scnlink: 控制通道
操作异常	EXCPT_INTERNAL_ERROR EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

## 4.9 发现管理模块 (module discoveringMgr) (可选)

## 4.9.1 DiscoveringMgr\_I 接口 (从 common::Common\_I 接口继承)

## 4.9.1.1 分配发现控制器(assignDiscoveringController)

定 义	
<pre>void assignDiscoveringController(     in globaldefs::NamingAttributes_T mLSNPPOrTPName,     in string discoveringControllerIdentifier)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	分配发现控制器
输入参数	mLSNPPNameOrTPName: 指定的多层子网点池或终端点名称 discoveringControllerIdentifier: 指定的发现控制器标识符
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

## 4.9.1.2 去分配发现控制器(deassignDiscoveringController)

定 义	
<pre>void deassignDiscoveringController(     in globaldefs::NamingAttributes_T mLSNPPOrTPName)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	去分配发现控制器
输入参数	mLSNPPOrTPName: 指定的多层子网点池或终端点名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

## 4.9.1.3 查询发现协议和参数(getDiscoveringProtocolAndParameters)

定 义	
<pre>void getDiscoveringProtocolAndParameters(     in globaldefs::NamingAttributes_T mLSNPPOrTPName,     out DiscoveringProtocol_T discoveringProtocol,     out globaldefs::NVSList_T discoveringParameters)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	查询发现协议和信令参数
输入参数	mLSNPPOrTPName: 指定的多层子网点池或终端点名称
输入/输出参数	无
输出参数	discoveringProtocol: 使用的协议名称, 除说明的协议名称外, 厂商可根据实现扩展 discoveringParameters: 使用的发现控制参数
操作异常	NOT_IMPLEMENTED. INTERNAL_ERROR. INVALID_INPUT ENTITY_NOT_FOUND. UNABLE_TO_COMPLY NE_COMM_LOSS

## 4.9.1.4 设置发现协议和参数(setDiscoveringProtocolAndParameters)

定 义	
<pre>void setDiscoveringProtocolAndParameters(     in globaldefs::NamingAttributes_T mLSNPPOrTPName,     in string discoveringProtocol,     in globaldefs::NVSList_T discoveringParameters)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	设置发现协议和参数
输入参数	mLSNPPOrTPName: 指定的多层子网点池或终端点名称 discoveringProtocol: 使用的协议名称, 厂商可根据实现扩展 discoveringParameters: 使用的发现控制参数
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

4.9.1.5 修改发现协议参数(modifyDiscoveringProtocolParameters)

定 义	
<pre>void modifyDiscoveringProtocolParameters(     in globaldefs::NamingAttributes_T mLSNPPOrTPName,     inout globaldefs::NVSList_T discoveringParameters)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	修改发现协议参数。当信令使能或非使能时，均能操作
输入参数	mLSNPPOrTPName: 指定的多层子网点池或终端点名称
输入/输出参数	discoveringParameters: (输入) 需修改的信令参数; (输出) 修改后的信令参数
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

4.9.1.6 使能发现 (enableDiscovering)

定 义	
<pre>void enableDiscovering(     in globaldefs::NamingAttributes_T mLSNPPOrTPName)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	使能发现
输入参数	mLSNPPOrTPName: 指定的多层子网点池或终端点名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

## 4.9.1.7 禁止发现(disableDiscovering)

定 义	
<pre>void disableDiscovering(     in globaldefs::NamingAttributes_T mLSNPPOrTPName)     raises(globaldefs::ProcessingFailureException);</pre>	
说 明	
功能描述	禁止发现
输入参数	mLSNPPOrTPName: 指定的多层子网点池或终端点名称
输入/输出参数	无
输出参数	无
操作异常	EXCPT_NOT_IMPLEMENTED EXCPT_INTERNAL_ERROR. EXCPT_INVALID_INPUT EXCPT_ENTITY_NOT_FOUND EXCPT_UNABLE_TO_COMPLY EXCPT_NE_COMM_LOSS

## 4.10 故障管理模块

故障管理功能无ASON新增接口要求。

ASON新增的故障数据,应符合YD/T 1767.5-2009《自动交换光网络(ASON)网络管理技术要求 第3部分:EMS—NMS接口功能》中相应章节的要求。

## 4.11 性能管理模块

性能管理功无ASON新增接口要求。

ASON新增的性能数据,应符合YD/T 1767.2-2008《自动交换光网络(ASON)网络管理技术要求 第2部分:EMS—NMS接口功能》中相应章节的要求。

附录 A  
(规范性附录)  
对象管理相关定义

### A.1 对象命名规则

本部分定义的对象命名除表A.1中列出的之外可参考引用文件TMF 814 “Multi-Technology Network Management Solution Set Document NML-EML Interface Version 3.5”中的“SD1-25\_objectNaming”的定义，SCNLink\_T对象的命名规则见表A.1。

表A.1 控制通道对象的命名

对象	对象命名
控制通道 (SCNLink)	1. name="EMS";value="CompanyName/EMSname" 2. name="SCNLink";value="SCNLinkName" 其中 SCNLinkName 可由厂家指定，保证在 EMS 内唯一
控制通道节点 (SCNNode)	1. name="EMS" value="EMS 名称" 2. name="ManagedElement"= "网元名称" 3.name="SCNNode" value="SCN 节点名称"(可以为 IP 地址)
TNA	name="TNA",value="TNAName" 其中 TNAName 为整数
TNA 组(TNAGroup)	name="TNAGroup",value="TNAGroupName" 其中 TNAGroupName 为整数

附录 B  
(资料性附录)  
通知格式定义

### B.1 通知格式定义

本部分定义的对象创建、删除、属性改变、状态改变通知以及连接路由变化通知。请参考引用文件TMF 814 “Multi-Technology Network Management Solution Set Document NML-EML Interface Version 3.5 ” 中的 “SD1-26\_OMGServicesUsage” ; CosNotification::StructuredEvent的定义请参见 “CosNotification.idl” 。

本部分新增的对象创建、对象删除、对象属性改变、对象状态改变类通知见表B.1。

表B.1 新增对象的通知上报

对象类型	上报创建通知	上报删除通知	上报属性改变通知	上报状态改变通知
Call_T	Yes	Yes	Yes	Yes[可选]
SCNLink_T	Yes[可选]	Yes[可选]	Yes[可选]	Yes[可选]
MultiLayerSubnetwork_T (MLRA)	Yes[可选]	Yes[可选]	Yes[可选]	No
MultiLayerSNPPLink_T	Yes[可选]	Yes[可选]	Yes[可选]	Yes[可选]
MLSNPP_T	Yes[可选]	Yes[可选]	Yes[可选]	No
SubnetworkConnection_T (Connection)	Yes	Yes	Yes	Yes[可选]



广东省网络空间安全协会受控资料

中华人民共和国  
通信行业标准

自动交换光网络（ASON）网络管理技术要求  
第5部分：基于IDL/IIOP技术的EMS—NMS接口通用信息模型

YD/T 1767.5-2009

\*

人民邮电出版社出版发行  
北京市崇文区夕照寺街14号A座  
邮政编码：100061  
北京新瑞铭印刷有限公司印刷  
版权所有 不得翻印

\*

开本：880×1230 1/16 2010年1月第1版  
印张：5.75 2010年1月北京第1次印刷  
字数：158千字

ISBN 978 - 7 - 115 - 1943/10 - 5

定价：50元