**System Requirement Specifications**

**SPN Controller SRS**

**Author - Wei Yuemin**

**Revision No-0.97**

**Issue Date- Aug. 24, 2018**

|  |  |
| --- | --- |
| **DOCUMENT PDS ID:** | |
| **PRODUCT: SPN Controller** | **RELEASE:** |
| **FEATURE: SPN Domain Controller** | **SUBSYSTEM: System Engineering** |
| **DISTRIBUTE TO: SE, SW, ST,PM** | |

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Revision #** | **Author** | **Reviewed By** | **A. Description Of Change**  **B. Summary of Review** | **Issued by / Date** |
| 0.1 | Wei Yuemin |  | Initial draft. | July 22, 2018 |
| 0.8 | Wei Yuemin | HeQian, Yang Jianjun, Zhang Huafei, Long Jinpeng, Zhao Guo Quan, | 1. PathSID 范围定义 16..1048576 需排除SRGB，SRLB， 和PTN LSP标签  2. SR Tunnel的路径计算是否支持Binding SID，有两种方案：不将Binding作为基础拓扑计算，将Binding作为基础拓扑计算。第一阶段暂先不做。 | Aug. 24, 2018 |
| 0.9 | Wei Yuemin | Team | 增加需求：  Link SID  Tunnel管理更新，包括：  SR Tunnel的快捷创建  创建SR Tunnel时， 可以选择SRTP方式还是SRBE方式 | Aug. 29, 2018 |
| 0.91 | Wei Yuemin | Team | 增加需求：  FlexE要求  FlexE拓扑，及相关拓扑要求更新  SID部分更新  Slicing部分更新 | Sep. 5, 2018 |
| 0.92 | Wei Yuemin | Team | 增加界面要求 | Sep. 6, 2018 |
| 0.93 | Wei Yuemin | HeQian, Yang Jianjun, Zhang Huafei, Long Jinpeng, Zhao Guoquan, Lv Qiuming, Zhao Xiaofei | 增加需求：  1. 分出：DataLink Topology和IGP Topology  2. BGP-LS 的CMCC 测试要求  3. FlexE的CMCC 测试要求  4. 手工创建 SRTP Tunnel  5. 告警和性能的CMCC测试要求 | Sep. 10, 2018 |
| 0.94 | Wei Yuemin | HeQian, Yang Jianjun, Zhang Huafei, Long Jinpeng, Zhao Guoquan, Lv Qiuming, Zhao Xiaofei | 根据讨论，更新拓扑需求 | Sep. 14, 2018 |
| 0.95 | Wei Yuemin | Team | 增加需求：  MPLS TP Tunnel  去掉SID规划，SID划分就是SRGB和SRLB | Oct. 11, 2018 |
| 0.96 | Wei Yuemin | Team | 更新：  Tunnel部分需求 | Oct. 16, 2018 |
| 0.97 | Wei Yuemin | Team | 增加：  L3VPN要求 | Oct. 27, 2018 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

*UTStarcom* ***Proprietary***

Not for use or disclosure outside ***UTStarcom, Inc.*** or any of its subsidiaries

Except under prior written agreement.

Other Intended Audience: Quality Assurance Group & Internal Auditors.

This document is controlled electronically and any hard copy should be considered uncontrolled.

The latest version of this document may be obtained from the

This document is owned by **Wei Yuemin**

# Reference

中国移动5G传输实验室测试方案-PTN演进v1.2，以下简称测试方案

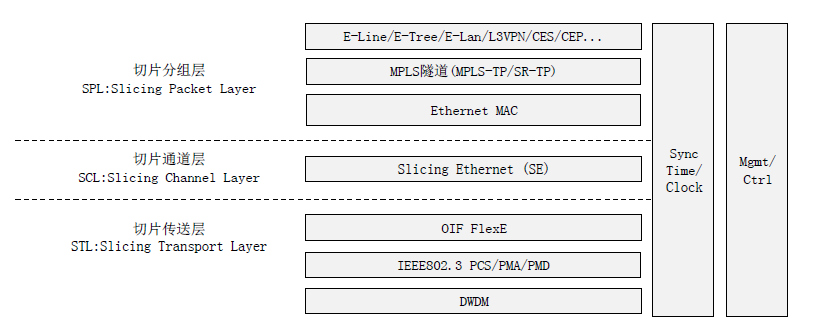
中国移动切片分组网（SPN）总体技术要求V0 005-0123，以下简称技术要求

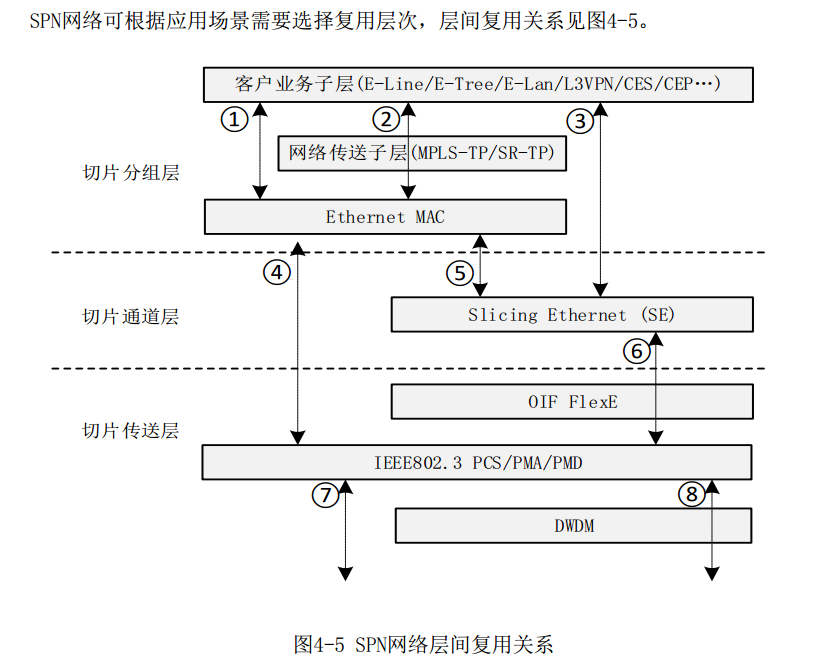
# 基本介绍

SPN: Slicing Packet Network

由中移动提出的，面向城域综合业务承载的传送网技术机制，对移动前/中/回传、企事业专线/专网、家庭宽带等高质量要求的业务进行综合承载的传输层网络。

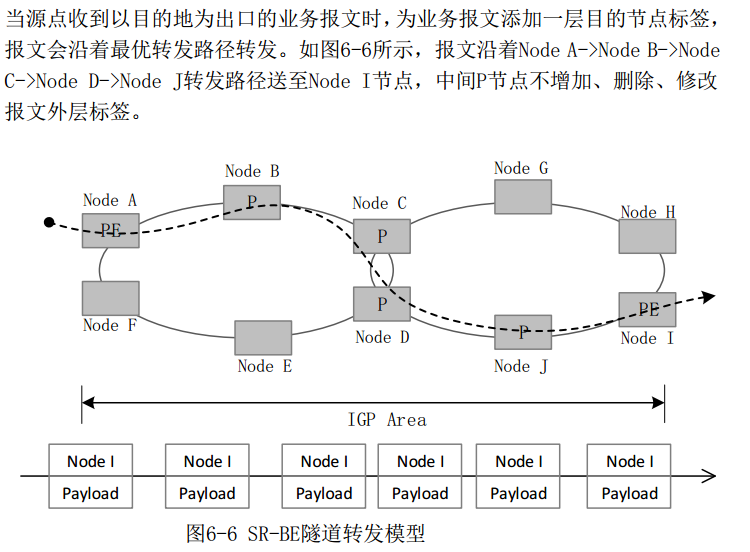
其层次架构如下图所示：

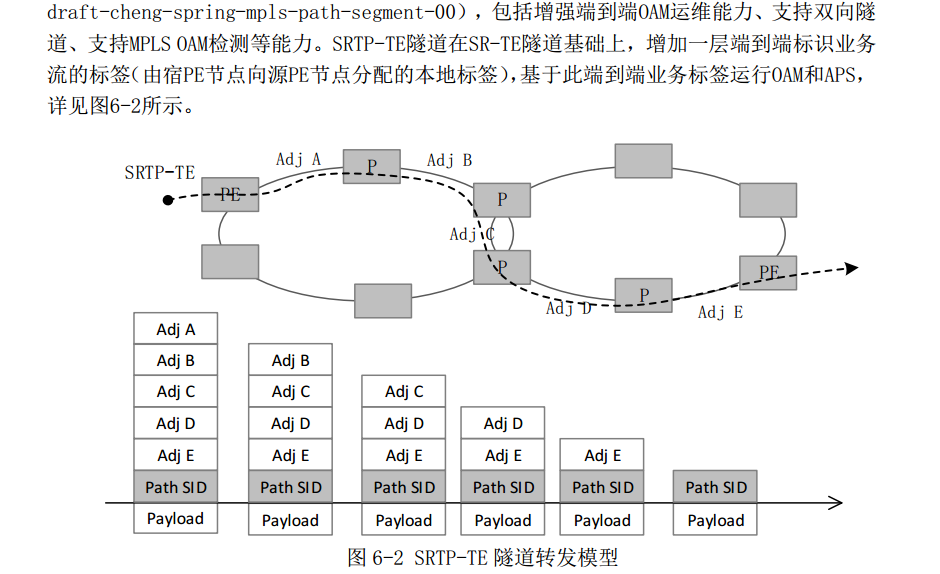




其中，切片分组层 - Slicing Packet Layer（SPL）基于MPLS和VPN技术，为分组业务提供具备统计复用能力的传送功能，以及业务传送过程中需要的OAM检测和保护功能。这一层包括了传统的MPLS-TP和新定义的SRTP 技术。

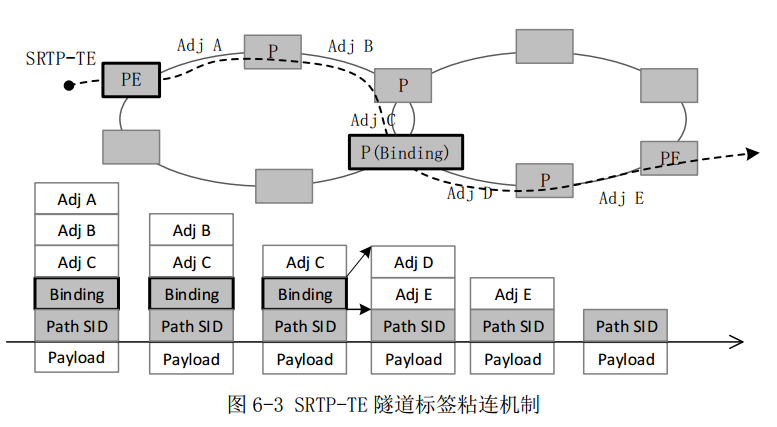
两种转发模型：





载自总体技术要求，现在把这种方式SRTP-TE叫成SRTP

Binding SID



# 版本计划

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Batch** | **Milestone** | **Features** | **HW** | **FPGA** | **SW** | **NMS** | **Controller** | **SIT** |
| batch1 | 9/15 | SPN805:MPLS-TP(L2VPN,syncE,1588) | SPN805（TN705E）now | now | now | now |  | 8/1~9/15 |
| batch2 | 10/15 | SPN805:SR-TP/BE, IS-IS E-LSP L3VPN | SPN805:OPCS：9/1 bring up 结束 | 9/15完成 | 9/15完成 | 9/15完成 |  | 9/14~10/15 |
| **batch3** | **10/30** | SPN803:FlexE | SPN803:91OPCL：8/15 bring up结束 | 10/7 完成 | 10/15 完成 | 10/15 完成 | 10/30 完成一轮ST | 10/16~10/30 |
| **batch4** | **12/30** | SPN803:FlexE+B1+B2+FlexE-APS |  | 11/15 完成 | 12/15 完成 | 12/15 完成 | 12/30 完成一轮ST | 10/31~12/30 |

网管部分版本计划

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Item** | **sub Item** | | **Description** | **Comments** | **NE** | **Roadmap** |
| DvM |  |  |  |  | SPN803/805 1.0.0 | 2.6.9 |
| E-LSP |  |  | QoS相关 |  | SPN803/805 1.0.0 | 2.6.9 |
| ISIS |  |  | ISIS拓扑发现 | 低优先级 | SPN803/805 1.0.0 | 2.6.9 |
| FlexE |  |  |  | 可选 | SPN803/805 1.0.0 | 2.6.9 |
| L3VPN |  |  |  | 可选 | SPN803/805 1.0.0 | 2.6.9 |

SPN Controller发布版本号3.2.0.x

同ODL的各个项目开发规则一样，不同Feature有不同bundle提供

UAR500 SRv6的功能模块bundle名字还是不变，比如sid-rest, ce-rest

SPN系列的功能模块bundle名字以spn打头，比如spn-sid-rest

公共功能Feature模块的bundle能继续部署，代码版本可以继续延续，比如aptn-alarm-service

(Qx Driver Alarm notification 用了ODL的websocket 专有框架接口）

初期的Controller版本主要提供端到端业务部署，网管Lite版本提供DvM

终极目标是提供管控一体的Controller版本

# TBD - Slicing 网络切片管理

虽然，切片不是前期第一阶段的要求，但后续要支持，架构设计上支持。

vNet = vNode + vLink

vNet的产生过程：

1. vNode的产生：物理网元资源池 分离 出虚拟网元

物理网元的资源池包括：物理端口，逻辑端口（vlan端口，FlexE Group，FlexE Client等）

2. vLink的产生

用户在vNode间的端口上配置Link

3. 用户将vNode和vLink组合成vNet

一个vNode只能属于一个vNet，一个vLink只能属于一个vNet，即vNet是一个独立的切片Slicing。

按上述流程，模型数据上Slicing的生成过程是：

1. 从物理Node中分离出资源，生成虚节点资源

2. 将虚节点规划入一个Net，形成一个拓扑，即使一个slicing子网

3. Slicing子网的物理拓扑的底层是物理节点所在的物理拓扑层

4. 在Slicing子网的物理拓扑层上可以继续创建Slicing子网的上层拓扑层

比如：一张FlexE网络，一张普通Ethernet网络

网络切片的示意图



在少量扩展后（资源部分Node type需支持vNode类型，vNode和Host Node的关联，资源所属SlicingID，等），现有的模型架构已经能支持以上vNet的构建，例如：

模型：network-topology

Physical Topology

Node

node-id=Node1

node-type=physical

Termination Port

port1

port2

node-id=Node2

node-type=physical

Termination Port

port1

port2

Link

Link1

Link2

Slicing1 Topology

underlay-topology

Physical Topology

Node

node-id=Node1-1

node-type=virtual

supporting-node

Node1

Termination Port

port1

node-id=Node2-1

node-type=virtual

supporting-node

Node2

Termination Port

port1

Link

Link1-1

supporting-link

Link1

# 拓扑管理

SPN管理面网络支持IPv4，IPv6。即网元的管理地址需支持IPv4, IPv6

SPN控制面网络支持IPv4，IPv6，路由协议是ISIS

SR Domain 和 SR Controller关系

因为关系到标签统一分配，所以一个SR Domain由且仅由一个Controller管辖。我们称为Domain Controller。

SR Domain和IGP域的关系

SR Domain：IGP可以是1:1，也可以是1：n

============载自技术总体要求==============

SR域(SR Domain)由一组支持SR功能的设备节点组成，SR Domain内Segment可以

是本地邻接标签或全局唯一的节点标签。 网络部署过程中，建议以IGP域为基础

规划SR域，以简化网络规划。

此外，若为网络中每个IGP域规划SR域，会导致多IGP域相交点标签空间规划繁

琐且标签利用率低； 若将整网所有IGP域规划为一个SR域，会导致接入设备节点

标签空间不足。综合考虑， SPN设备应支持将一组IGP域规划为一个SR域(如图6-5

所示，可将汇聚环1下所有接入环IGP1/2/3规划为一个SR Domain1，将汇聚环2

下所有接入环IGP4/5规划为一个SR Domain2)，不同SR域间可复用节点标签空间

================================================

域间的拓扑关系也通过ISIS 邻居表

拓扑YANG模型：继续沿用utstarcom-sdn-net-topology （以后BGP-LS模块生成的拓扑[draft-clemm-i2rs-yang-network-topo](https://datatracker.ietf.org/doc/draft-clemm-i2rs-yang-network-topo/)与此不同，到时再看如何归并）

Next UI 拓扑要求：

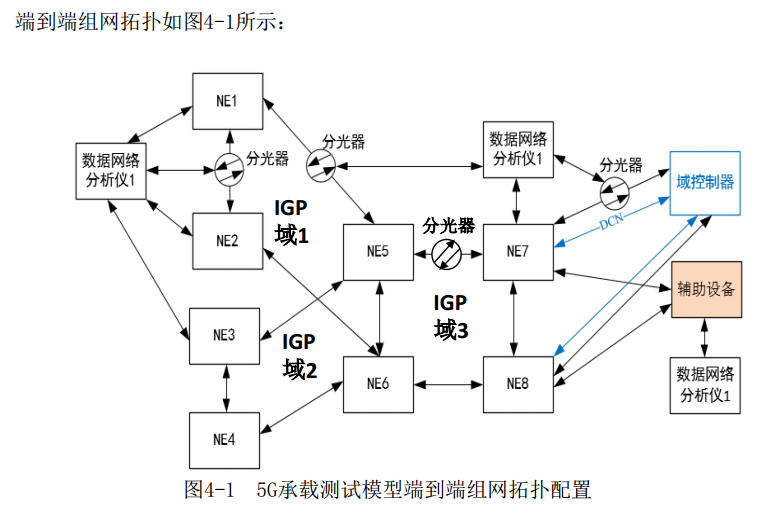
**TBD**【】Group： 可以一组网元以一块颜色区域表示，比如：

中移动测试规范里的拓扑，这里按IGP域划为3组。（当然，IGP域和组是两个概念，这里仅用来举例）

模型： 拓扑里的Node对象可以是group节点，其node-type为组，组下节点通过

/nt:network-topology/nt:topology/nt:node/nettopology:aptn-node-attributes/nettopology:parent-group-id

和组做关联



图表 1 CMCC SPN测试用例网络拓扑

**TBD**【】用户可以通过拓扑界面操作。选一个节点或多个节点后，做加入组或退出组操作。（这里的组就是指NextUI的组，一个网元可以属于多组）



【】Node可以加入一个NodeSet，或从一个Nodeset移出。一个Node只能加入一个NodeSet。

【】NodeSet展开和收缩操作：对上图的组为例，默认全展开， 下图对set1做收缩。

（一个节点只属于一个NodeSet，NodeSet支持嵌套）

NodeSet就是Subnet，Subnet和节点的关系在不同的拓扑层默认保持一致。 即物理层的Subnet-node和上层IGP层，LSP/SR-Tunnel层，PW层，CE层默认都保持一致。

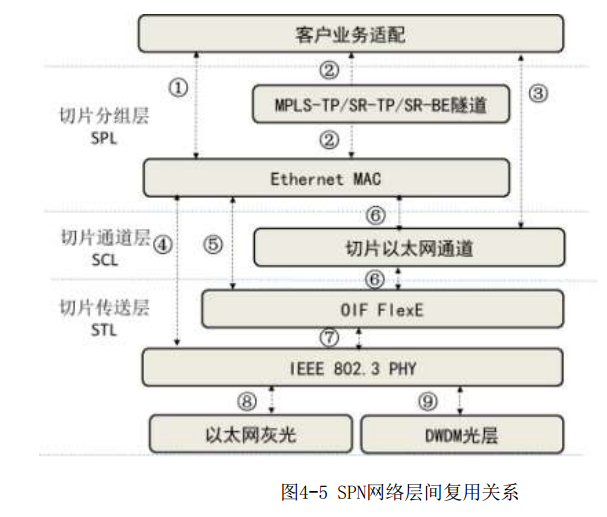




层次图



 SRTP业务建立在IP拓扑上。 FlexE业务可以直接建在MAC/PHY上。PW-Only业务可以直接建立在MAC层上。



在拓扑模型上需要分层，理由：

1. FlexE各层有承载关系

2. FlexE Channel需要底层拓扑（Client/Group层）做寻路支持

3. FlexE网络是硬切片网络的一种，分出单独拓扑层次比较清晰，便于今后的切片管理扩展

4. 移动的测试方案明确要求Controller创建配置Group/Client/Channel，分层有利于清晰管理（正如，移动在DL层分出LSP/PW/Ethnet一样的道理），Controller创建出来的Group，Client，Channel就是相关拓扑里的Link

层次是比较多，但目前没有单独拓扑界面去展示FlexGroup和FlexEClient的需要。 （其实层次的存在是大家都认同的，但层次的模型表示是用同层的LinkType来区分还是独立的拓扑层次来区分？我建议Client,Channel用独立拓扑层次，理由还是上面的。）

目前（9,10月），就只有底层物理拓扑是必须要界面展示和操作管理的。 上图看到，物理层底层包括Ethernet和FlexE Group（一条光纤要么使用普通以太模式要么使用FlexE方式传送，所以这里又需要用Linktype来分）

因为LSP寻路是基于底层，这里的底层可以是Ethernet层和FlexE Channel层。 即一条LSP可以Ethernet链路和FlexE Channel链路混杂的走。 （所以，我也在考虑是否加一层统一的Datalink底层，一个内存拓扑操作树，包容有Ethernet链路和FlexE Channel链路）

就9、10月份的工作优先级来说，先把FlexE Group/Client/Channel的管理界面先做起来（管理yang模型有单独定义），拓扑层次可以放后面。



拓扑形成（PTN LSP/PW拓扑，安排在二阶段，架构层次需考虑，但下面未列的应该不冲突。）

Step1. 物理拓扑形成

网元手工添加

链路手工添加

Step2. IGP拓扑形成 （第二阶段需支持通过BGP-LS收集拓扑）

读取网元ISIS相关表做发现 （节点和链路在IGP形成的前提条件是IP配置已完成）

Step3. SR SID在IGP拓扑上的信息补充，即

节点赋予Node SID，

接口赋予AdjSID，

多段赋予BindingSID （BindingSidLink可以不用显示，但需供SR Tunnel SID List算路用）

Step4. SR Tunnel的路径计算在带有SID信息的IGP拓扑上完成

FlexE模型对拓扑的影响，下面以例子说明：

Step1. 物理拓扑形成

AB间有两条100G以太链路，如

A(eth1.1,eth2.1)========(eth1.1,eth2.1)B(eth3.1)------------ (eth1.1)C

~~Step2. 将端口加入FlexE Group，并形成一条邻接网元FlexE Group Link（模型数据上：原有的以太链路删除，新加Group Link），物理拓扑展现如下~~ （Group层还是做了单独层）

~~A(FlexE Group1)-------------(FlexE Group1)B(eth3.1)------------ (eth1.1)C~~

~~Step3. 在FlexE Group上创建了2条FlexE Client(放置在FlexE Client Link拓扑层 )，物理拓扑展现不变，即还是~~

~~A(FlexE Group1)-------------(FlexE Group1)B(eth3.1)------------ (eth1.1)C~~

~~用户可以点选FlexE Group Link切到FlexE Group管理页面里看到Client信息~~

Step4. FlexE Channel端到端创建（过程就是选已有或新建FlexE Client Link，并做中间节点的Client交叉，这些都在FlexE Channel管理界面做）

选已有FlexE Client Link就是在FlexE Client Link拓扑层能寻到满足带宽条件的路（寻路拓扑界面就是FlexE Client Link拓扑层的展示）；若没有满足的路径存在，则可新建FlexE Client Link形成端到端的路。

Step5. FlexE Channel作为物理拓扑子层，和普通以太链路物理子层，可以作为LSP的底层选路拓扑。

Step6. 用户给Channel的端点接口配置IP，并发布到ISIS，那么在IGP拓扑层就会形成一条基于底层FlexE Channel Link的IGP Link。 这条IGP Link端点被赋予SID后，便可供SR Tunnel使用。

综上所述，现阶段，我们的实现优先级是：

1. 物理拓扑

2. FlexE管理

3. IGP拓扑和SID

4. SR Tunnel

5. 其他业务

\*注意：

FlexE Channel是Controller侧的FlexE Client端到端连接模型，是中移动要求的（SPN Channel/Slicing Ethernet Channel）。

对网元来讲，PW，LSP，OAM都是关联到FlexE Client上的。

移动要求必须有的拓扑：

1.物理拓扑

2. IP拓扑

物理拓扑就是节点光纤链路拓扑

IP拓扑是BGP-LS发现的拓扑

这两份拓扑不一定相同

现在我们的实现是只有Datalink拓扑（不论手工创建还是ISIS发现的拓扑）， 并且把SID信息也放置到该拓扑层，所以其实我们没有实现分层，就一层。 这个要改。

IP拓扑带SID信息，是SR Tunnel寻路的数据基础。

物理拓扑是手工建的。

IP拓扑一般是自动发现的，但因测试环境不具备，为方便测试也需支持手工创建Link。

IP拓扑如果是通过ISIS邻居表发现的link，且这样的link在物理拓扑中没有，则自动在物理拓扑中也建立一条。

## Physical Topology

网元间通过光纤相连而成的拓扑，我们叫为Media层，这里为减少拓扑层次，Media层不用做。我们做以太物理层，包括普通以太链路和FlexEGroup链路。

模型：

aptn-topology-attributes.layer-rate=physical

aptn-topology-attributes.sub-layer-rate= physical

aptn-link-attributes.link-type=1/2 /\*1-IEEE 802.3,2 FlexE Group\*/

【R-PhysicalTopo-000】网元添加：手工添加，输入属性有：Node Label, IP Address（v4，v6都支持）, NodeType=SPN803 or SPN805。 做类型检查（Version，NEStat ShelfType ~~BaseInventory~~，具体参考网管实现），如果类型不一致，则标志节点状态信息：/resource:resources/resource:nes/resource:ne-list/resource:ne/spn-ext:node-status-check = NODE-TYPE-MISMATCH

拓扑界面上需体现该状态

【R-PhysicalTopo-001】网元删除

以下情况不能删除：

1. 已配置业务（包括SID配置，SRTunnel配置，等任何Controller下发的配置）的网元不能被删除，删除有配置业务的网元需先删除Controller侧配置数据后，才能再删除网元。

2. 包含虚节点的物理网元

【R-PhysicalTopo-002】网元属性修改

可修改属性：IP，Label，NE Type

【R-PhysicalTopo-003】Ethernet Link手工添加

用户选择两端Ethernet端口，即可建立Link

aptn-link-attributes.link-type=1

【R-PhysicalTopo-004】Link手工删除

以下情况不能删除：

存在上层Link（比如FlexE Channel层，IGP层）的物理层Link不能被删除

【R-PhysicalTopo-005】网元添加后，若网元Netconf连接成功，则Controller需Mount网元mo文件里列出的表，出于性能考虑，目前load暂定以下表项（后续会加，如没有性能问题，将加载mo里的所有列出规格表）：

alarmcurrent

MO

IfCfg

IfIpAddrCfg

IfIpv6AddrCfg

IsisNbrStat

IsisNetCfg

EthSvcPWCfg

EthVplsPrep

EthVpwsPrep

PWCfg

EthAcCfg

LocalSRGBCfg

LocalSRLBCfg

SRLocalPrefixSIDCfg

SRRemotePrefixSIDCfg

L3VPNArpCfg

L3VPNNdCfg

L3VPNIpv4AcCfg

L3VPNIpv6AcCfg

L3VPNStaticIpv4RouteCfg

L3VPNStaticIpv6RouteCfg

VpnCmmCfg.V0

L3VPNGlobalCfg.V0

BgpRDCfg

BgpRTCfg

SubInterface

MplsXcCfg

FlexECalendarCfg

FlexEChannelIntfCfg

FlexEClientCfg

FlexEGroupCfg

FlexEGroupMemberCfg

FlexEPortCfg

FlexEXcCfg

MO表里的板卡和端口信息需同步到Controller资源模型，资源模型仍然沿用，即utstarcom-sdn-resources

【】链路TE属性

### FlexE Group Topology

flexe-group和 Ethernet链路同层，即为Physical以太底层

aptn-topology-attributes.layer-rate=physical

aptn-topology-attributes.sub-layer-rate= flexe-group

### FlexE Client Topology

flexe-client层

Supporting Link是底层FlexE Group层

aptn-topology-attributes.layer-rate=physical

aptn-topology-attributes.sub-layer-rate=flexe-client

本层不单独做Topology View展示

补充：

FlexE Group子层（可以看成相邻节点间链路的粗管子） 和 FlexE Client子层（可以看成粗管子里的细管子）

这两个子层的存在是为了FlexE Channel端到端寻路提供底层数据。即先找细管子，没有符合条件的细管子，则再找粗管子。

FlexE Group、FlexE Client是一段段的短管子；FlexE Channel是端到端的逻辑管道

### FlexE-Channel Topology

FlexE Channel是一条逻辑通道，由路径上的各网元节点SE-XC和各FlexE Group链路中若

干个时隙组成的端到端通路

底层是FlexE-Client 拓扑

Supporting Link是底层FlexE--Client链路

aptn-topology-attributes.layer-rate=physical

aptn-topology-attributes.sub-layer-rate=flexe-channel

本层不单独做Topology View展示

## DataLink Topology

DataLink Topology = FlexE Channel Topology + Ethernet Topology

aptn-topology-attributes.layer-rate=datalink

aptn-topology-attributes.sub-layer-rate=mac

【】DataLink拓扑产生：

节点：随物理节点产生而产生。（Node id = 物理层Node id）

Link：是一条普通Ethernet链路，或是一条FlexE Channel链路，或是一条FlexE Client（Link id = 物理层 Link id 或 FlexE Channel Link id）

/nt:network-topology/nt:topology/nt:link/nettopology:aptn-link-attributes/nettopology:link-type

1 - IEEE 802.3

2 - FlexE Channel

Datalink层麻烦的地方是Link转换，NNI链路初始是Ethernet Link，使能FlexE后，需删除原有ethnetlink，新建一条flexe link。这样Tunnel的寻路可以在Datalink层上做，用户可以输入走flexe，还是ethernet，或混跑。

## IGP Topology

底层是DataLink Topology拓扑

这层拓扑其实即包括了传统PTN的DL Link拓扑（因为我们PTN的DL也是基于IGP发现的），也包括了SR的IP拓扑

aptn-topology-attributes.layer-rate=ip

【R-IGPTopo-000】ISIS Link拓扑发现：同UAR500网络发现，即通过ISIS邻居表和接口地址表（UAR500控制面用IPv6，SPN控制面用IPv4，所以这里接口表用IfIpAddrCfg；当然，以前用在IPv6上的邻居发现也仍然有用的，因为网元支持双栈；所以代码要适配，即如果isisnbrstat告知的邻居接口地址是v4就查IfIpAddrCfg，v6就查IfIpv6AddrCfg）的数据生成拓扑

【R-IGPTopo-001】已发现的IGP拓扑层的Link若不存在对应的Physical拓扑Link，则自动创建对应的Physical拓扑Link。

【R-IGPTopo-002】手工创建Link (这功能主要是为前期测试用，实际部署应该是ISIS发现或BGP-LS发现)

用户在IGP Topology上选两端端口后，可手工创建Link。Link的方向默认是双向。

【R-IGPTopo-003】LinkId

IGP Link ID表示的是本层LinkID，不同于物理层的LinkID，需单独分配

【R-IGPTopo-004】Carried Link Type

表示该IGP Link的承载层链路类型：1-IEEE 802.3，2-FlexE Channel

模型：/nt:network-topology/nt:topology/nt:link/nettopology:aptn-link-attributes/nettopology:carried-link-type

这个信息在选路时，根据输入条件，可过滤出要求的link范围

【R-IGPTopo-005】Carried Link

IGP Link需记录承载层的关联Link id，比如该Link的底层Link是一条FlexE Channel Link

模型：/nt:network-topology/nt:topology/nt:link/nt:supporting-link/nt:link-ref

【】链路TE属性

|  |
| --- |
| 时延：/nt:network-topology/nt:topology/nt:link/nettopology:aptn-link-attributes/nettopology:link-te-attr-cfg/nettopology:latency |
|  |
| 最大可用带宽： /nt:network-topology/nt:topology/nt:link/nettopology:aptn-link-attributes/nettopology:link-te-attr-cfg/nettopology:max-reservable-bandwidth |

## Segment-Routing SID Topology

本层可以作为IGP拓扑层的SID信息扩展

Node-SID（参见“本地Prefix SID”需求）和Link SID（参见“Link SID”需求）的拓扑层，该层为SR Tunnel SRTP寻路使用。

TBD【R-SRTopo-000】拓扑显示

节点的显示label为 Node Label [Node-SID]，链路显示标示为Link SID

【R-SRTopo-001】拓扑模型数据

根据对节点的Node SID和Adj SID的配置，需记录到相关拓扑模型数据里。为SR Tunnel SRTP寻路使用。

节点：

/nt:network-topology/nt:topology/nt:node/spn-topology:sr/spn-topology:node-SID

端口：

/nt:network-topology/nt:topology/nt:node/nt:termination-point/spn-topology:Adj-SID/spn-topology:Adj-SID

Binding SID：

/nt:network-topology/nt:topology/nt:link/spn-topology:link-segment-SID/spn-topology:link-segment-SID

## LSP/SR-Tunnel Topology View

数据来自sptn-service-tunnel

LSP 拓扑层的底层是DL层

SR Tunnel的底层是IGP层

本层不单独做Topology View展示

## PW Topology View

本层不单独做Topology View展示

## CE Topology View

本层不单独做Topology View展示

# BGP-LS

Controller作为BGP-LS Speaker，和网元做协议交互，获取各个IGP Domain内的拓扑信息。

BGP-LS发现的拓扑数据需同步到IGP Topology层

拓扑图参见：图表 1 CMCC SPN测试用例网络拓扑

【】测试方案要求：8.1.1 验证基于管控融合系统实现 SPN 网络拓扑搜集

测试步骤

a） 按照上图搭建测试配置，核心设备 NE7 和 NE8 与域控制器相连，检查控制

器为初始状态，无网络拓扑信息；

b） 域控制器通过网关方式管理核心侧设备 NE7 和 NE8，通过非网关管理组网

中的其他设备；

c） 通过域控制器对全网设备按照规划部署 ISIS 协议，相邻设备间建立 ISIS

邻居，打通所有设备之间的公网 ISIS 路由；

d） 通过域控制器在核心设备和域控制器之间部署 BGP LS，域控制器通过 BGPLS 协议搜集网络中的拓扑；

e） 通过 FlexE 接口的以太网端口镜像来抓包检查 NE1-NE2、NE1-NE5 之间的

ISIS 协议报文、控制器和 NE7 或 NE8 之间的 BGP-LS 协议报文。

f） 检查控制器是否完成网络拓扑的搜集过程， 是否能正确呈现所有的节点端

口和链路信息。

预期结果

步骤 d） ： 域控制器通过 BGP-LS 协议搜集网络中的拓扑， 并在界面上展现拓扑；

步骤 e） ：设备之间的 ISIS 报文以及核心设备和域控制器之间的 BGP-LS 报文

满足 SPN 规范要求。

【】测试方案要求：8.1.2 验证基于管控融合系统实现 SPN 网络拓扑刷新

测试步骤

a） 按照图 39 搭建测试配置，核心设备 NE7 和 NE8 与域控制器相连；

b） 完成域控制器和设备之间 ISIS、BGP-LS 等控制平面协议部署；

c） 构造网络中的某条链路故障（关闭端口或拔纤） ，观测域控制器上的拓扑

能否正确刷新；

d） 恢复网络中的链路故障，观测域控制器上的拓扑能否正确刷新。

e） 构造网络中 NE1 或 NE5 的节点故障（节点掉电或两侧拔纤） ，观测域控制

器上的拓扑能否正确刷新；

f） 恢复网络中的节点故障，观测域控制器上的拓扑能否正确刷新

预期结果

步骤 b） ： 域控制器通过 BGP LS 协议搜集网络中的拓扑， 并在界面上展现拓扑；

步骤 c） ：域控制器界面上的拓扑不显示故障的链路；

步骤 d）：域控制器界面上的拓扑包含故障恢复后的链路。

步骤 e） ：域控制器界面上的拓扑不显示故障的节点；

步骤 f）：域控制器界面上的拓扑包含故障恢复后的节点。

# PCEP

测试方案：

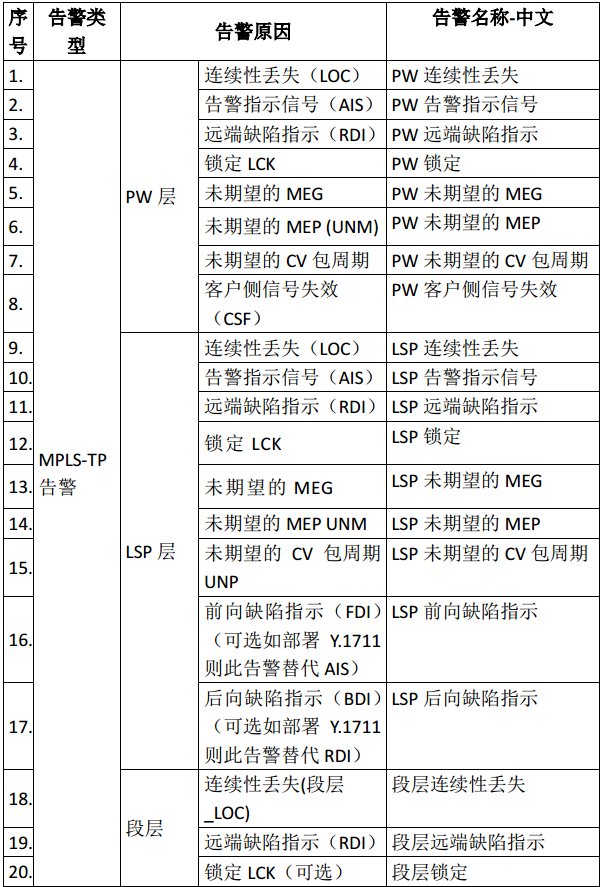
8.2.2 SR-TP 隧道的算路策略测试

要求Controller在完成隧道计算后，将隧道配置通过PCEP下发

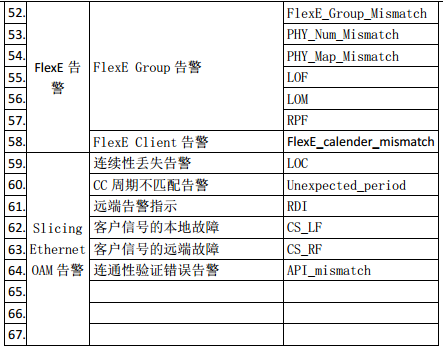
# 告警管理

当前告警：同UAR500 SR-Controller里的当前告警实现

8.3.1 业务和隧道的告警上报功能







c) 测试工具选择域内隧道和业务，查看告警是否上报成功；

d) 抓包分析北向接口告警上报报文。

e) 通过测试工具记录 D-Controller 的域内告警和 EMS 管理的域内告警的数量

及类型区别

预期结果：

步骤 c）告警能够正确上报；

步骤 d）抓包分析告警上报报文符合预期；

步骤 e）测试工具可正确查询告警信息。

# 性能管理

Port/SRTunnel/LSP/PW 各个性能指标

TWAMP

测试方案 8.3.2 业务和隧道的性能监测功能

测试步骤：

a) 按照测试配置连接网络；

b) 测试工具选择域内隧道和业务（测试前需要清空性能数据，重新记录，便

于对比），查询当前和历史性能；

c) 通过数据分析仪构造网络中一定的线路丢包





d) 查询业务监测点的时延、抖动、丢包和流量性能，并能按列表方式呈现；

e) 验证控制器查询的时延、抖动、丢包和流量性能与仪表测试结果基本一致，

记录误差范围。

预期结果：

步骤 d）SR-TP、LSP 和端口业务性能参数查询结果正确

步骤 e）收发报数、时延、抖动、丢包等和仪表测试结果基本一致

# 安全管理

登录认证

分权分域

# 日志管理

操作日志

系统日志

# SID管理

Prefix-SID，一个全网唯一标签，表示一个网络前缀

Node-SID，一种Prefix-SID，一般对应于网元loopback地址前缀

Adj-SID，一个网元本地标签，对应于网元本地的一个出接口

SRGB - Segment Routing Global Block， Prefix-SID标签池

SRLB - Segment Routing Local Block， Adj-SID标签池

ietf Yang仅供参考：draft-ietf-teas-yang-sr-te-topo-02

具体Yang文件会单独出

## SRGB Management

Controller 配置树YANG: utstarcom-sdn-spn-sid

网元规格表名：LocalSRGBCfg

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item Name | Data Type | Limit | Default Value | Show Name | Description |
| Index | Octet | 1…8 | 1 |  | Key |
| LowerBound | Long | 0,16..220 | 524288 |  |  |
| UpperBound | Long | 0,16..220 | 532479 |  |  |

初始化数据： （默认标签不动，避免和网管冲突）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **node-id** | **igp-instance-id** | **srgb-id** | **lower-bound** | **upper-bound** |
| 0 | 0 | 1 | 524288 | 532479 |
| 0 | 0 | 2 | 0 | 0 |
| 0 | 0 | 3 | 0 | 0 |
| 0 | 0 | 4 | 0 | 0 |
| 0 | 0 | 5 | 0 | 0 |
| 0 | 0 | 6 | 0 | 0 |
| 0 | 0 | 7 | 0 | 0 |
| 0 | 0 | 8 | 0 | 0 |

数据表项要求：每条记录的lower bound不大于upper bound；每条记录的[lowerbound, upperbound]范围不重叠。

【R-SRGB-000】操作要求：不可增删，只能读（Refresh）和修改（Update）。注意，标签范围不能和PTN标签重叠：LSP：[102400, 204799] PW: [204800, 307300]

【R-SRGB-001】全网应用：即把配置树上的8条SRGB数据下发（Set to Network）到所有网元上。

**TBD** 单一网元应用：即把配置树上的8条SRGB数据下发（Set to Node）到指定网元上。

【R-SRGB-002】对已经下发的SRGB条目做修改（status=Activated），则修改后需同步到网元；对已使用过的SRGB条目（status=Used），则不允许修改

Used表示标签已在Prefix-SID里有分配过。

【R-SRGB-003】Controller需维护已分配的SRGB SID列表：使用比例...

【R-SRGB-004】节点和Controller间的SRBG数据校对和同步。在以下情况需做自动校对：

1. 新网元添加后

2. 网元重连后

其实就是发生在任何导致网元Mount点上的SRBG表数据变更后，Controller需核对Inventory网元级配置数据和Controller SRBG模型数据的一致性。如果发现不一致，需置网元不一致状态，并在拓扑上对该网元做可识别表示，在Controller系统级的Event库里记录这不一致事件。

用户可以做手工校对，即重新Load网元SRGB数据，和Controller数据做对比，不同的情况下，需列出详细的不同条目。

不论是自动校对还是手工校对，发生不一致情况后，都是有手工来执行从controller复制到网元的同步。

### 界面参考





## SRLB Management

Controller 配置树YANG: utstarcom-sdn-spn-sid

网元规格表名：LocalSRLBCfg

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item Name | Data Type | Limit | Default Value | Show Name | Description |
| Index | Octet | 1…8 | 1 |  | Key |
| LowerBound | Long | 0,16..220 | 520192 |  |  |
| UpperBound | Long | 0,16..220 | 524287 |  |  |

初始化数据：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **node-id** | **igp-instance-id** | **srlb-id** | **lower-bound** | **upper-bound** |
| 0 | 0 | 1 | 520192 | 524287 |
| 0 | 0 | 2 | 0 | 0 |
| 0 | 0 | 3 | 0 | 0 |
| 0 | 0 | 4 | 0 | 0 |
| 0 | 0 | 5 | 0 | 0 |
| 0 | 0 | 6 | 0 | 0 |
| 0 | 0 | 7 | 0 | 0 |
| 0 | 0 | 8 | 0 | 0 |

数据表项要求：每条记录的lower bound不大于upper bound；每条记录的[lowerbound, upperbound]范围不重叠。

【R-SRLB-000】操作要求：不可增删，只能读（Refresh）和修改（Update）

【R-SRLB-001】全网应用：即把配置树上的8条SRLB数据下发（Set to Network）到所有网元上。

**TBD**单一网元应用：即把配置树上的8条SRLB数据下发（Set to Node）到指定网元上。

SRLB里的范围不能和SRGB里的范围相重

为运维简单，SRLB在全网节点上都一样

【R-SRLB-002】Controller和网元的SRLB数据校对和同步

类同SRGB校对和同步要求

【R-SRLB-003】Controller需记录每节点的可分配SRLB里的最小标签值，以便分配时使用。

### 界面部分



## Prefix-SID标签分配

网元规格表：

SRLocalPrefixSIDCfg

SRRemotePrefixSIDCfg

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item Name | Data Type | Limit | Default Value | Show Name | Description |
| Name | String |  |  |  |  |
| AddressFamily | Octet | Ipv4, ipv6 | Ipv4 | Address family |  |
| Prefix | String |  |  | Prefix | Key |
| PrefixMask | Octet | [1…….128] |  | PrefixMask |  |
| PrefixType | Octet | Node,Anycast (TBD),Other | Node | Prefix Type |  |
| StartSID | Long | 0..220-16OR16..220 |  |  | Value associated with the prefix  1.If prefix-SID type is absolute, the possible value is 16..220  2.If prefix-SDI type is index, the possible value is 0..220-16 |
| Range | Long | 1..8191 |  |  |  |
| PrefixSIDType | Octet | Absolute, index | Absolute |  |  |
| Algorithm | Octet | SPF,Strict SPF | SPF |  | Hidden(Key) |
| LastHopBehavior | Octet | explicit-null, no-php, php | no-php |  | Hidden |

//\*注意：网元规格的取值和Controller yang定义的取值不同：

ietf-segment-routing-common sid-value-type prefix type

index 0

absolute 1

LPrefixType

index 1

absolute 0

ietf-inet-types

ip-version

0-unknown

1-ipv4

2-ipv6

LSRAddressType

0-ipv4

1-ipv6

sid类型， Controller用 uint32，网元用long

/////////////////////

Prefix SID是全局SID，其标签可定位一个SR网络里的节点。

简单地说，Controller上维护的Prefix-SID整表数据就是网元上的Local表和Remote表数据之和。为运维便利，目前版本可以仅支持全网网元相同的一份Prefix-SID表数据。

【R-PrefixSID-000】用户给网元分配本地Prefix SID

需支持的操作：

添加：默认的SID Value Type是Index，默认的Node SID 填否。

删除：只在planning状态下允许

修改：

Node SID Boolean值可以修改

Prefix/Length是索引，不允许修改

其他值的修改需要在planning状态下

注意：Prefix-SID取值范围来自SRGB池，一个Prefix-SID值只能分配一次，即不会出现一个Prefix-SID值被分配到不同网元。（模型里有used-SRGB，新分配的SID就不能再用到）

| Node Label | Prefix/Length | SID Value Type | SID Value | Node SID |
| --- | --- | --- | --- | --- |
| SPN-hangzhou1 | 1.1.1.1/32 | Index | 1 | Yes |
| SPN-hangzhou1 | 10.10.10.0/30 | Index | 100 | No |
| ... |  |  |  |  |
| SPN-hangzhou2 | 2.2.2.2/32 | Index | 2 | Yes |
| SPN-hangzhou2 | 10.10.20.0/30 | Index | 200 | No |
| ... |  |  |  |  |

Yang模型： utstarcom-sdn-spn-sid prefix-SIDs

【R-PrefixSID-001】Node-SID需同步到拓扑数据：

/nt:network-topology/nt:topology/nt:node/spn-topology:sr/spn-topology:node-SID

【R-PrefixSID-002】用户可以选择对Prefix SID mapping表的任一列做单列排序

【R-PrefixSID-003】用户可以模糊查找，比如输入1.1.1，可查询出相关内容记录

一般，loopback的Prefix SID就是Node SID

【R-PrefixSID-004】Prefix SID需支持一下操作：激活，去激活，发布 (因为网元支持了通过ISIS做PrefixSID发布，所以Controller发布功能可以不做)

Prefix SID的配置状态：Planning, Activated, Advertised

新建的SID状态是Planning

激活下发（即写本节点的SRLocalPrefixSIDCfg）的SID状态是Activated

发布到网内所有其他节点（即写其他节点的SRRemotePrefixSIDCfg）的是Advertised （在后期网元IGP支持SID发布后，就不需要该功能了）

Activated状态通过去激活回到Planning状态

Advertised状态通过去激活也回到Planning状态（先做本网元的删除，再做其他网元的删除）

操作和状态图如下：



【R-PrefixSID-005】Activated或Advertised Prefix-SID 数据在Controller和网元间的手工校对

对Controller上所有处在Activated或Advertised的Prefix-SID数据条目和全网内所有节点上的SRLocalPrefixSIDCfg和SRRemotePrefixSIDCfg配置数据做校对。如有不一致的，则列出不匹配的网元和Prefix-SID列表。

（因为网元ISIS支持自动发布PrefixSID，所以这条功能可以暂不用做）~~对不一致的网元，用户可以将Controller的数据再次同步到网元。即将Controller的Prefix-SID下发或覆盖到网元上~~。



【R-PrefixSID-006】节点和Controller间的PrefixSID数据自动校对和手工同步。在以下情况需做自动校对：

1. 新网元添加后

2. 网元重连后

其实就是发生在任何导致网元Mount点上的Remote/Local PrefixSID表数据变更后，Controller需核对Inventory网元级配置数据和Controller 模型数据的一致性。如果发现不一致，需置网元不一致状态，并在拓扑上对该网元做可识别表示，在Controller系统级的Event库里记录这不一致事件。

用户可以做手工校对，即重新Load网元PrefixSID数据，和Controller数据做对比，不同的情况下，需列出详细的不同条目。 界面参考前条需求描述。

### 界面参考



需求R-PrefixSID-005，R-PrefixSID-006的界面参考需求描述

## 分配Adj SID

Adj SID表示一条本地L3链路（后续，会支持用一个Adj SID表示多个邻接节点，用在load balance上；也可以用多个Adj SID表示同一个邻接节点，用在保护；）

网元规格：SRLocalAdjSIDCfg

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item Name | Data Type | Limit | Default Value | Show Name | Description |
| Phyid | Octet | 0----254 | 0 | Phyid | hidden |
| Name | String |  |  |  |  |
| AddressFamily | Octet | Ipv4, ipv6 | Ipv4 | Address family |  |
| LocalPrefix | String |  |  | LocalPrefix | Key |
| LocalPrefixMask | Octet | [1….128] |  | LocalPrefixMask |  |
| RemotePrefix | String |  |  | RemotePrefix | Key |
| RemotePrefixMask | Octet | [1….128] |  | RemotePrefixMask |  |
| AdjSIDType | Octet | Absolute, index | Absolute | AdjSIDType |  |
| AdjSID | Long | 0..220-16OR16..220 |  |  | alue associated with the adjacency  1. If prefix-SID type is absolute, the possible value is 16..220  2. If prefix-SDI type is index, the possible value is 0..220-16 |
| BackUp | Boolean | True, false | false |  | Key  If set, the Adj-SID refers to an  adjacency being protected |
| Local | Boolean | True, false | True |  |  |
| Weight | Octet | 1..255 | 1 |  | Hidden |

默认以绝对值方式分配Adj SID, 且Local=true

~~如果网元支持AdjSID的自动生成，则R-AdjSID-000功能不需要~~

【R-AdjSID-000】用户可以选择物理拓扑上的链路后，可以查看和设置该链路的标签。标签的取值范围就是SRLB的标签池范围。（因为存在loadbalance，不同链路的标签值可能相同。）标签设置需下发到网元。

下表的接口地址和掩码根据链路端口信息从网上读取IfIpAddrCfg/IfIpV6AddrCfg，若该端口不是L3口，则无法配置Adj SID。 用户可以为同一条链路设置保护的SID （以后，还可以添加Parallel Link SID）。

用户可以对链路做保护配置，就是对指定链路设置受保护的Adj-SID。

|  |
| --- |
| Controller模型： /spn-sid:SR-Domain-SID/spn-sid:SR-Domains/spn-sid:Adj-SID |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Local  Node | Local  Link Interface | Local  IP Address/  MaskLength | Remote Node | Remote Interface | Remote  IP Address/  MaskLength | Adj SID | Protection |
| Hangzhou1 | eth1.1 | 10.10.10.1/30 | Hangzhou2 | eth1.1 | 10.10.10.2/30 | 10001 | No |
| Hangzhou2 | eth1.1 | 10.10.10.2/30 | Hangzhou1 | eth1.1 | 10.10.10.1/30 | 20001 | No |
| Hangzhou1 | eth2.1 | 10.10.11.1/30 | Hangzhou2 | eth2.1 | 10.10.11.2/30 | 30001 | No |
| Hangzhou3 | eth2.1 | 10.10.11.2/30 | Hangzhou1 | eth2.1 | 10.10.11.1/30 | 40001 | No |
| Hangzhou1 | eth2.1 | 10.10.11.1/30 | Hangzhou2 | eth2.1 | 10.10.11.2/30 | 50001 | Yes |
|  |  |  |  |  |  |  |  |

Adj SID是可以在不同网元分配相同值的，如下图：



因连接两端的AdjSID是可以不同的。 所以上图改为下图更合适：



【R-AdjSID-001】用户可以浏览和查询Adj SID总表。支持模糊查询，比如输入10001，则列出：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Node | Link Interface | IP Address/  MaskLength | Adjacent Node | Adjacent Interface | Adjacent  IP Address/  MaskLength | Adj SID | Protection |
| Hangzhou1 | eth1.1 | 10.10.10.1/30 | Hangzhou2 | eth1.1 | 10.10.10.2/30 | 10001 | No |
| Hangzhou2 | eth1.1 | 10.10.10.2/30 | Hangzhou1 | eth1.1 | 10.10.10.1/30 | 10001 | No |

【R-AdjSID-002】上述非保护的Adj SID 值需同步到拓扑数据  
/nt:network-topology/nt:topology/nt:node/nt:termination-point/spn-topology:Adj-SID/spn-topology:Adj-SID

【R-AdjSID-003】Link SID - 应用在SR-Topology里，为SRTP SIDList寻路服务

在network-topology模型中对link属性做扩展，用link-SID来表示一条边的关联SID，Adj SID表示一条相邻两节点间的边（这时LinkSID=Link.SouceTP.Adj-SID），Binding SID表示一条跨节点相接的边。

模型：

/nt:network-topology/nt:topology/nt:link/spn-topology:link-segment-SID

【R-AdjSID-003】Controller和网元的AdjSID数据校对和同步

如果网元支持AdjSID的自动生成，则同步方向是网元到Controller

如果网元不使能AdjSID的自动生成，则同步方向是Controller到网元，类同PrefixSID校对和同步要求

### 界面参考





Controller分配Label原则：

SR的label 和 PTN的label 不相交

Node SID和Prefix SID在SRBG里分配

Path SID, Binding SID, AdjSID默认在SRLB里分配

SRTP里的PW标签还是用PTN里的PW 标签范围。

# FlexE 管理

FlexE+B1+B2+FlexE-APS

## 介绍

FlexE服务客户 - FlexE Client

FlexE网络承载的服务客户， 目前OIF标准定义为以太分组业务流，可扩展支持TDM、 CPRI

等业务类型。

FlexE组 - FlexE Group

多个具备相同组编号的PHY组成的一个FlexE协议组， 为FlexE Shim提供统一的底层服务

FlexE Shim

FlexE Client进行映射或解映射的服务层。

FlexE Client通过FlexE Shim承载，FlexE Shim通过FlexE Group进行传送。

FlexE时隙 - FlexE Slot

66比特的数据块码流。 在物理PHY为100G时， FlexE Shim中有n\*20个时隙， n是成员数量，每个成员有20个时隙，每个时隙是一个66比特块的数据码流。

FlexE 则是将100GE/200GE/400GE的标准以太网时隙化，每个时隙的颗粒度为5G，20/40/80个时隙组成100G/200G/400G FlexE物理口。

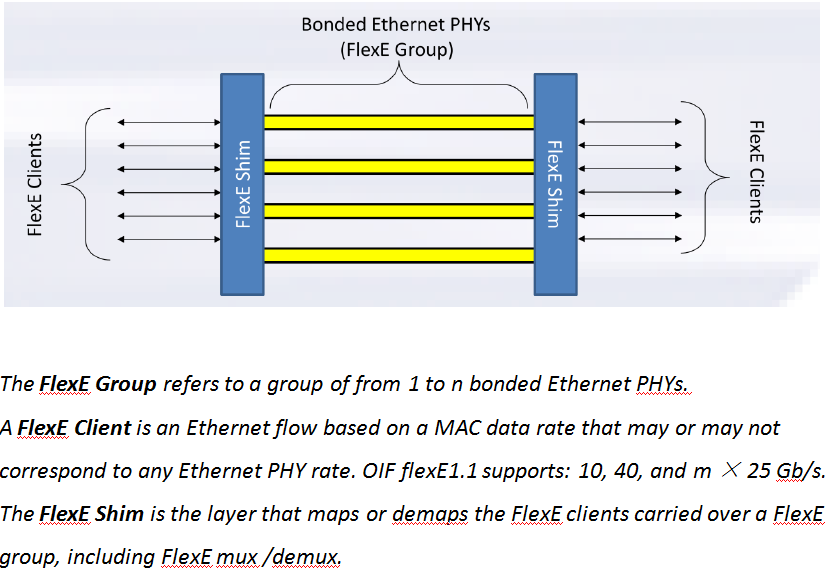
SPN Channel

在网络中传递的一条逻辑通道，由路径上的各网元节点SE-XC和各FlexE Group链路中若

干个时隙组成的端到端通路

SE-XC – Slicing Ethernet Cross Connect

SPN Channel通过设备交叉矩阵进行以太网码流交叉连接的行为



总体技术要求提到的要求：

a） 应支持 SPN Channel 的端到端建立、拆除和维护操作；应支持指定 SPN Channel 的

头节点和尾节点自动算路建立一条 SPN Channel；应支持指定 SPN Channel 建立必

需经过和必需不经过节点、接口的约束路径 SPN Channel；

b） 应支持 SPN Channel 与承载时隙的绑定,一条 SPN Channel 可绑定一个或多个 FlexE

时隙；应支持 SPN Channel 时隙或带宽配置调整；

c） 应支持 SPN Channel 最小 5G 带宽粒度配置；

d） 应支持 N\*5G（N≥1）为颗粒的任意 FlexE client；

e） 应支持 MPLS-TP 隧道和 SRTP 隧道承载于 SPN Channel 功能；

f） 应支持 SPN Channel 的 OAM 机制，详见第 9.2 节要求；

g） 应支持 SPN Channel 的保护机制，详见第 10.2 节要求

总体技术要求提到的SPN Channel（就是SPN设备支持的FlexE Channel ）

SPN切片通道层（SCL）采用Slicing Ethernet技术，基于IEEE 802.3(50GE及以上的接口)对PCS（物理编码子层）进行时隙化处理，实现绑定、子速率和通道化等基础功能，进一步为多业务承载提供基于L1的低时延、硬隔离切片通道，

SPN切片通道是网络中的一条源宿节点之间的传输路径，用于在网络中提供端到端的以

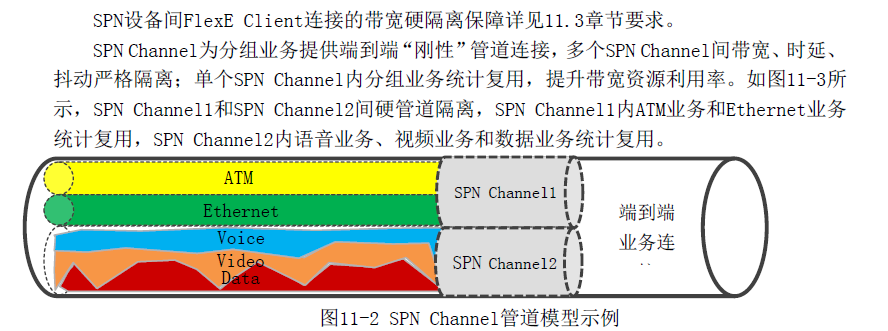
太切片连接，具有低时延、透明传输、硬隔离等特征。采用Slicing Ethernet技术，兼容FlexE

接口，客户层业务在源节点映射到FlexE Client，网络中间节点基于FlexE Client进行交换，

在目的节点从FlexE Client中解映射客户层业务，可实现客户数据的接入/恢复、增加/删除

OAM信息、数据流的交叉连接，以及通道的监控和保护等功能。

SPN网络切片通道层即端到端的SPN Channel，用于在网络中提供端到端的Slicing Ethernet连接，具有低时延、透明传输、带宽硬隔离等特征。端到端SPN Channel带宽硬隔离由设备内SE-XC和设备间FlexE Client通路实现。



SPN网络切片传送层分为FlexE Group链路层和IEEE 802.3以太网(PMA和PMD)层，以及光

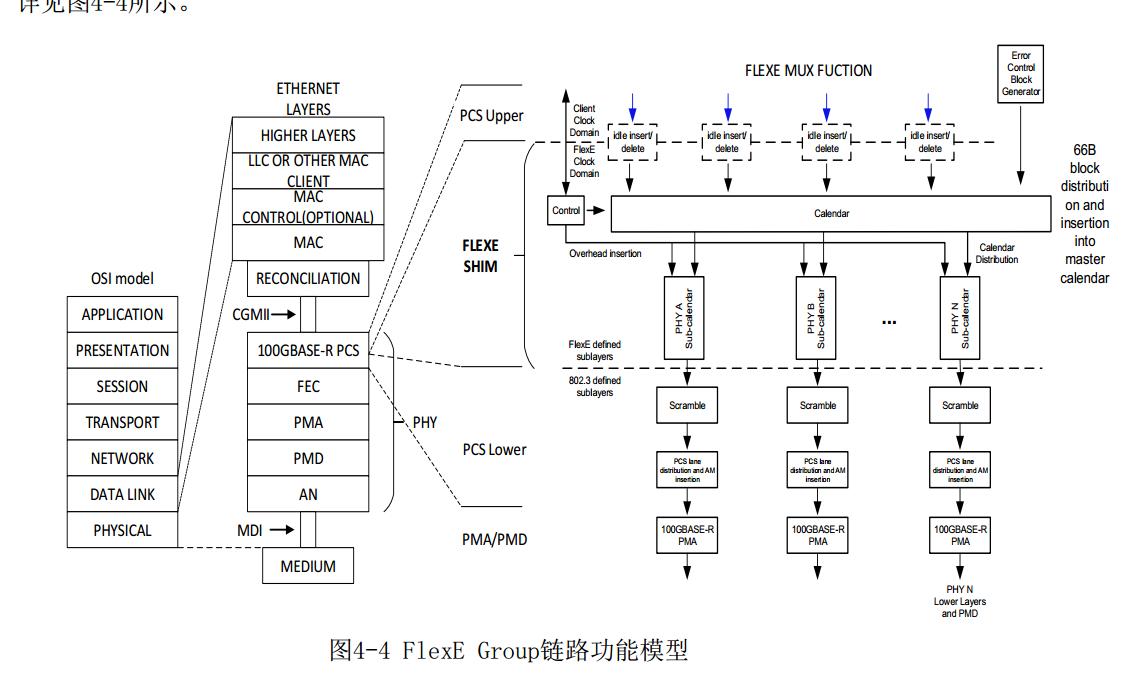
层的WDM波分技术。

FlexE Group链路层基于FlexE Shim提供监控物理媒介层的点到点的连接能力，位于基

于FlexE Client的切片通道层和IEEE 802.3以太网PHY之间。在FlexE Group链路层，实现接入数据流的频率和速率适配、数据流在FlexE Shim的映射与解映射、FlexE Group开销的插

入与提取等功能。一个FlexE Group支持多个FlexE Client

FlexE工作在物理层



## 要求

基本配置：

1. Enable FlexE端口，设置PHY ID，Flex Shim两端的PHY ID配置需一致

2. FlexE Instance设置，目前不需要，给200G和400端口设置多个Instance/100G单位

3. FlexE Group配置，Flex Shim两端的group id需一致

4. FlexE Group 成员配置，目前（SPN803）一个组就只一个100G口成员，以后支持多成员口

FlexE Channel邻接节点间的配置都需要一样，Phyid，Instance，Groupid，Phy Map, Calendar type，Client id, slot no

可以把时隙理解成标签，本端出标签 即是 对端入标签

这句话对吗？？？一个具有FlexE功能的物理口，要么工作在FlexE模式，要么工作在NativeEthernet模式。也就是没有这种情况：100G接口Eth1/1可以工作在两种模式，分割出5G带宽走FlexE Channel，剩下95G工作在普通NNI口模式。

是的，只能二选一，不能同时。

【】FlexE Group管理

创建步骤：

1. 设置Group名称，Group id（表示一条Shim 编号，不同相邻节点间的编号可以重复，比如下例；Group index是节点唯一，比如下例。 Group id可以手工输入，也可以自动生成）

A-Node ====group id 1=====B-Node=====group id 1=====C-Node

B-Node: group index = 100, group id =1, group member = port1

B-Node: group index = 101, group id =1, group member = port2

2. 设置Calendar类型：A或B

3. 选择相邻节点和对接端口

4. 创建

FlexE Group激活，去激活

所涉及的表项：

FlexEGroupCfg

FlexEGroupMemberCfg

FlexECalendarCfg

FlexEPortCfg

FlexE Group的修改，去激活后，可以修改包括Group id在内的所有内容

FlexE Group删除

FlexE Group内的时隙占用情况：已使用，未使用

Group id的校验：

取值范围：1..1048574

不同的Group Index记录里的Group Id可以相同

一个NNI接口只能加入一个Group

Group Name：

长度：128

不支持字符：/ \ \* : <> " | () {}（）

界面参考





【】FlexE Client管理

创建步骤：

1. 输入Client名称，Client id（选择自动生成，或手工输入），带宽（5G粒度，5G一个时隙），时隙位置，

2. 选择FlexE Group

3. 创建

FlexE Client 激活，去激活

FlexE Client修改：能支持包括Client id，带宽，时隙位置内所有字段修改

FlexE Client删除

Client id的校验：

取值范围：1..65534

同一个FlexE Group里的Client id不能相同

不同FlexE Group里的Client id可以相同

Slot id: 0-159

界面参考



【R-FlexE-003】FlexE Channel管理

创建：

输入名称，带宽（5G粒度），

选择AZ节点，并可输入路径必过节点或必绕节点，

Controller依据FlexShimGroup/Client拓扑层做寻路并创建Channel。

首先根据FlexShimClient拓扑列出可选路径（带宽严格匹配，即Channel Bandwidth=Client Bandwidth，如果Channel要求5G，但现存没有5G Client路径，而有10G路径，那么也不行。）（最短前3条，默认则选最短）

如果找不到Client路径，则根据FlexShimGroup拓扑列出可选路径（最短前3条，默认则选最短），确认后，controller自动创建满足条件Client，并在此client上创建Channel。

创建完成的FlexE Channel需同步到拓扑层数据。

~~FlexE Client也可以看成是最简单的Channel，即只有首尾节点没有中间节点的Channel。~~

~~Client可以为不同Channel共享~~

一条client只能给一条channel用

并且，Client也不当成一条默认Channel了，要建Channel后，再承载业务FlexE业务。这样对Controller层次模型上做一致。 Channel是逻辑概念，网元上没有Channel。

激活：将创建成功的Channel下发到网元，涉及到的网元有：

首尾端设置~~FlexEClientCfg~~，FlexEChannelIntCfg

中间节点设置FlexEXcCfg

去激活：将Channel配置从网元删除

修改：Channel的时隙和带宽调整（即路径重算和重建）

OAM：

1+1 APS保护：

删除：将Channel从Controller删除

查看：包括图形化显示 Slicing Ethernet Channel 占用的时隙和端口、承载的业务（MPLS-TP 隧道和 SR-TP 隧道）、状态监控（OAM 告警、保护倒换的状态）

对应于

测试方案8.2.1 FlexE 和 Slicing Ethernet 的配置管理能力测试

测试步骤

a） 按照图 39 搭建测试配置，核心设备 NE7 和 NE8 与域控制器相连；

b） 通过管控系统使能所有 NE 的线路端口 FlexE 模式；

c） 管控系统应支持 FlexE Group 的创建：自动生成 FlexE Group ID，并支持用

户配置和修改 FlexE Group ID；

d） 管控系统应支持 FlexE Client 的配置： 基于 FlexE Group （FG） 支持创建多个

FlexE Client， 自动生成 FlexE Client ID，并支持用户配置和修改 FlexEClientID；

支持配置 FlexEClient 的带宽（占用的时隙个数），并支持带宽调整（增加

或减少时隙） ，支持人工配置和自动配置（Calenda A/B switch）两种模式；

e） 管控系统应支持显示 FlexE Group 中已使用和未使用的时隙；

f） 管控系统应支持创建和配置 Slicing Ethernet Channel，满足如下功能要求：

（1） 应支持 Slicing Ethernet Channel 的端到端创建、删除、查看和状态监

控；

（2） 应支持指定 Slicing Ethernet Channel 的头节点和尾节点自动算路建立

一条隧道；

（3） 应支持指定 Slicing Ethernet Channel 建立必需经过和必需不经过节

点；

（4） 应支持 Slicing Ethernet Channel 与承载时隙的绑定, 一条 Slicing

Ethernet Channel 邦定一个或多个时隙；

（5） 应支持 Slicing Ethernet Channel 时隙或带宽配置调整；

（6） 应支持 Slicing Ethernet Channel 的相关 OAM 配置；

（7） 应支持 Slicing Ethernet Channel 的 1+1 APS 保护配置；

（8） 应支持 MPLS-TP 隧道和 SR-TP 隧道承载于 Slicing Ethernet Channel 上

的配置功能；

（9） 应支持 Slicing Ethernet Channel 的信息查看，包括图形化显示 Slicing

Ethernet Channel 占用的时隙和端口、承载的业务、状态监控（OAM 告警、保

护倒换的状态）；

（10） 应支持 Slicing Ethernet Channel 的端到端删除

预期结果：

步骤 b） ：管控系统支持将所有 NE 的线路端口使能为 FlexE 模式；

步骤 c） ：管控系统支持 FlexE Group 的创建和相关参数配置；

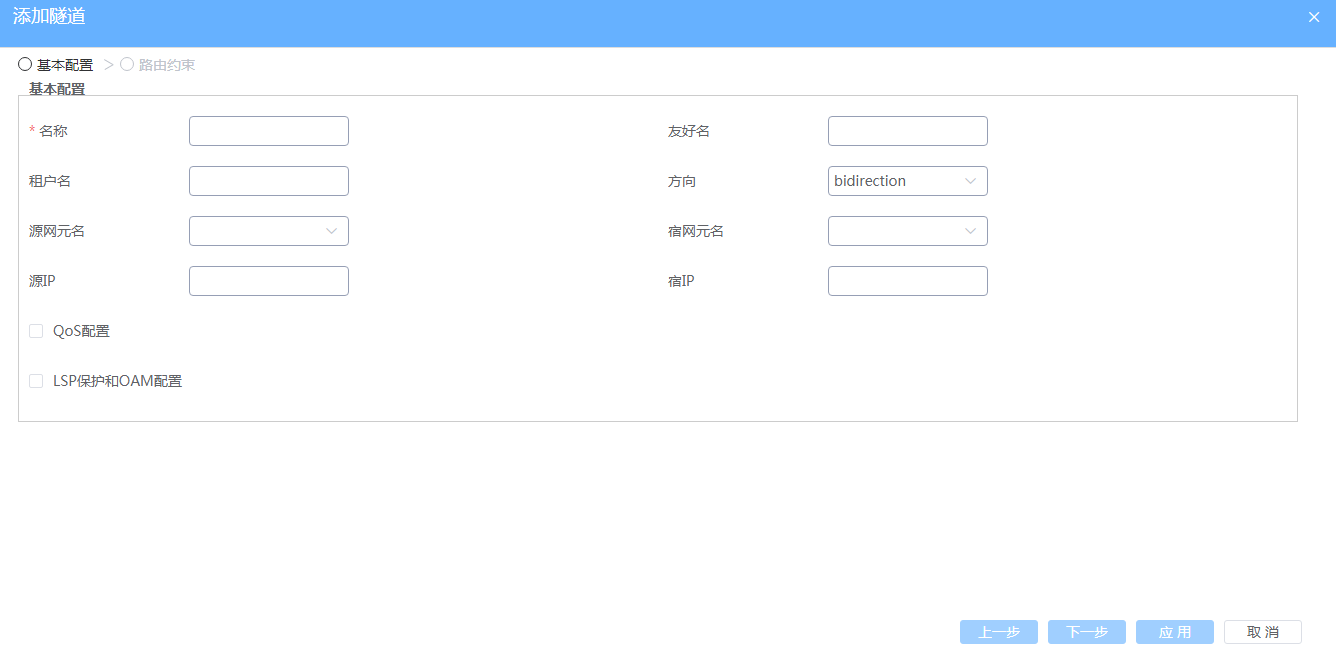
步骤 d） ：管控系统支持 FlexE Client 的创建和相关参数配置；

步骤 e） ：管控系统应支持显示 FlexE Group 中已使用和未使用的时隙；

步骤 f） ：管控系统应支持创建和配置 Slicing Ethernet Channel。

界面参考：类同Tunnel界面

创建（其中QoS，保护和OAM，目前不提供，灰掉处理）

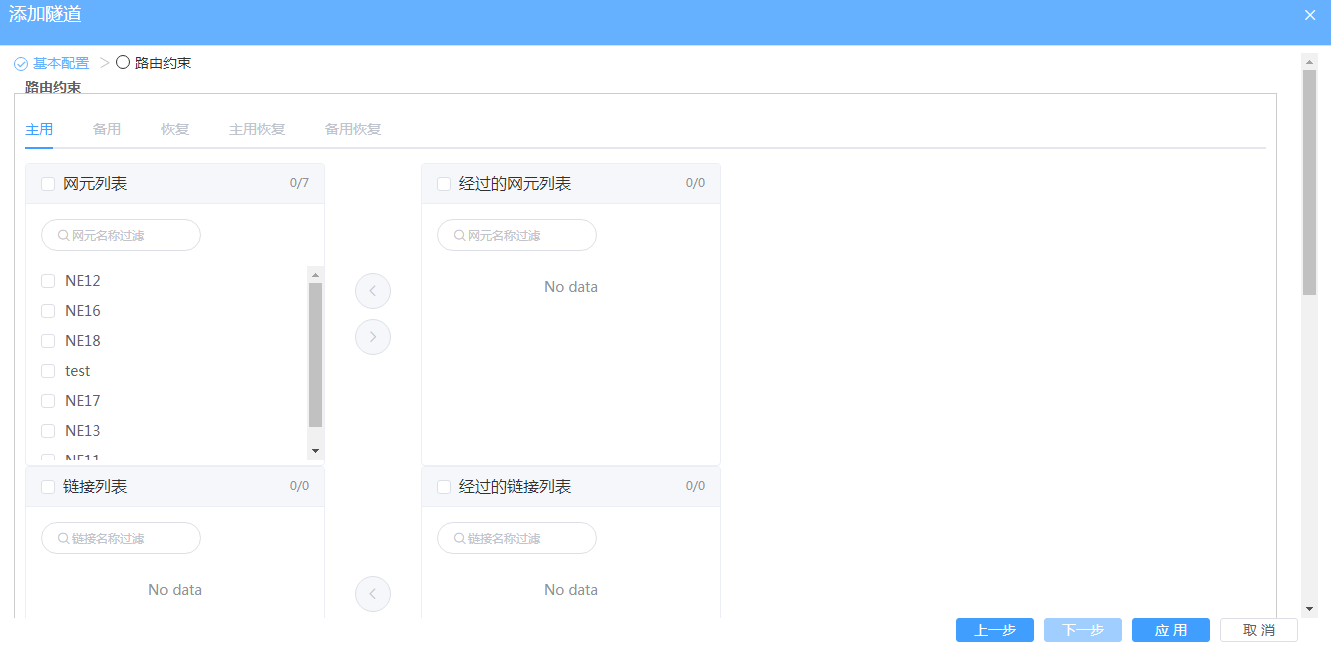


需增加一输入项：带宽，以5G为粒度

源IP和宿IP，对应到模型Channel Interface 的ip-prefix

网元规格变更：ip prefix不写入FlexEChannelIntfCfg表的IpPrefix，

路由约束界面也和Tunnel相同：



添加最后综合展示的页面：



【R-FlexE-004】FlexE Channel可作为LSP、SR-Tunnel的底层链路，为LSP/SR-Tunnel的寻路路径的部分。可以根据用户输入条件：链路类型或低时延(MinimumDelayPath)，来选择走FlexChannel路径。

LSP，SR-Tunnel底层的选路拓扑范围需支持两种方式：

LSP的底层是Datalink Topo，而FlexE Channel默认就应以一条Link加入到DL Topo

SR-Tunnel的底层是IGP Topo，FlexE Channel Link加入IGP Topology的前提是：FlexE Channel 接口作为三层口加入ISIS域

Controller需知道这条DL/IGP Link是有FlexE Channel承载的，这样选路时可根据条件做过滤。

FlexE保护 ？？？

# Tunnel 管理

Tunnel类别和下发表 示意图

【】用户自动或手工创建Tunnel时，可以选择MPLS Tunnel或SR Tunnel类型（默认SR Tunnel），

在SR Tunnel方式下，还可以选择是SRTP或SRBE或Binding的方式。

模型：

/tunnel:service/tunnel:snc-tunnels/tunnel:snc-tunnel/tunnel:type

leaf type {

description 'type of snc tunnel.

1 is line MPLS,

2 is ring MPLS.

97 SRTP tunnel

98 SRBE tunnel

99 SR Binding tunnel' ;

type int8;

}

【】提供统一管理界面对MPLS Tunnel和SR Tunnel的管理操作：创建，激活，去激活，删除，

【】用户可以选择Tunnel的承载方式：IEEE 802.3 或 FlexE 或 Mix

IEEE 802.3表示Tunnel的选路只走普通以太链路，即普通Eth NNI口，

FlexE表示Tunnel的选路只走FlexE链路，FlexE链路指FlexE Channel 链路

Mix表示Tunnel的选路可以走FlexE链路或以太链路，即No Limit

注：如前面Datalink Topology所述，当一条以太链路使能FlexE后（即加入FlexEGroup），该以太链路就不存在了，而在FlexE Channel配置后，将生成FlexE链路

## PTN MPLS Tunnel管理要求

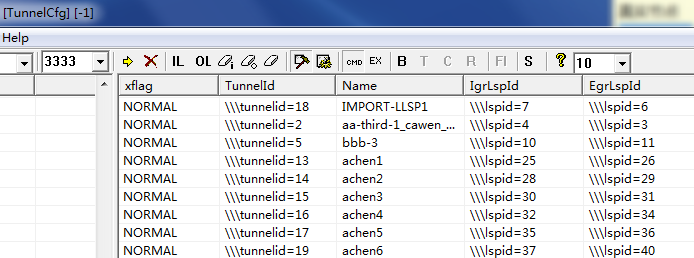
参见《LSP-PW模型设计》

Controller模型：

utstarcom-spn-service-tunnel  
utstarcom-spn-service-route

网元规格：

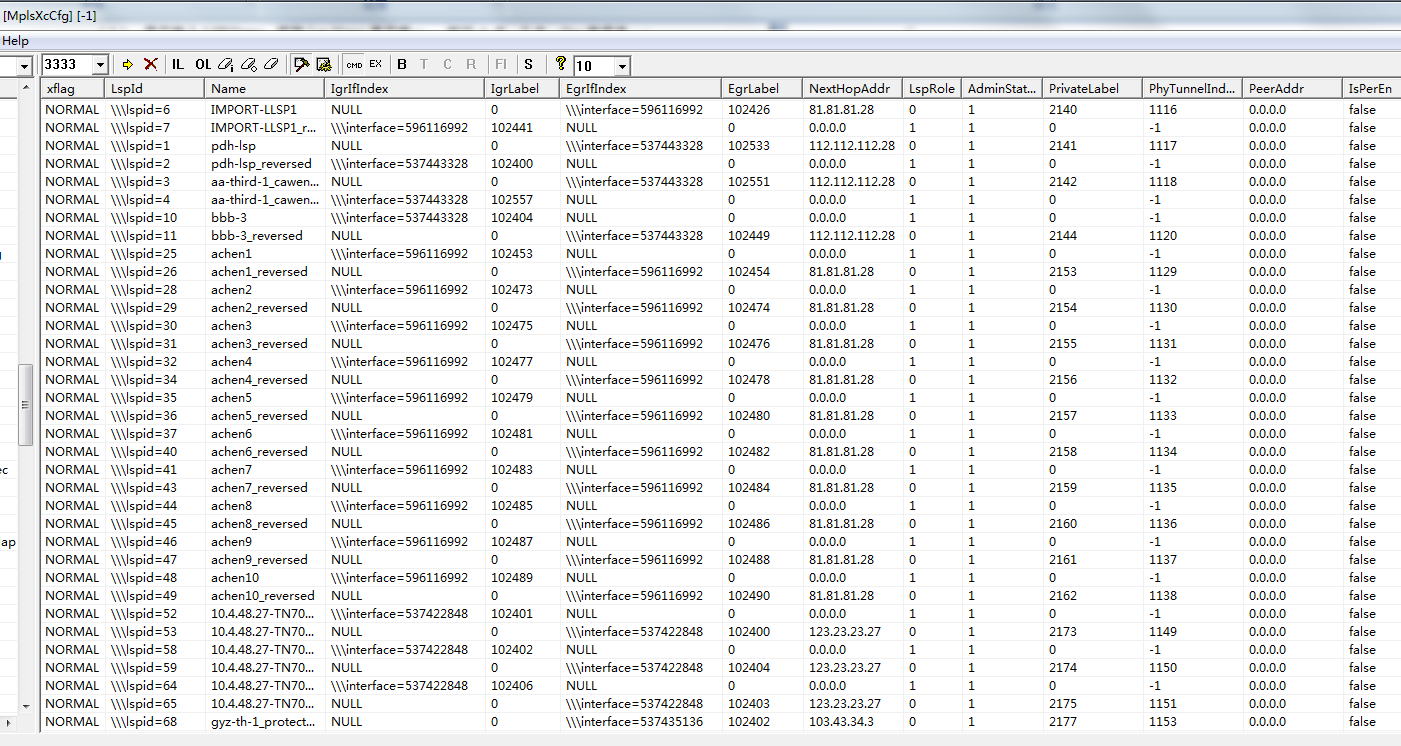
TunnelCfg



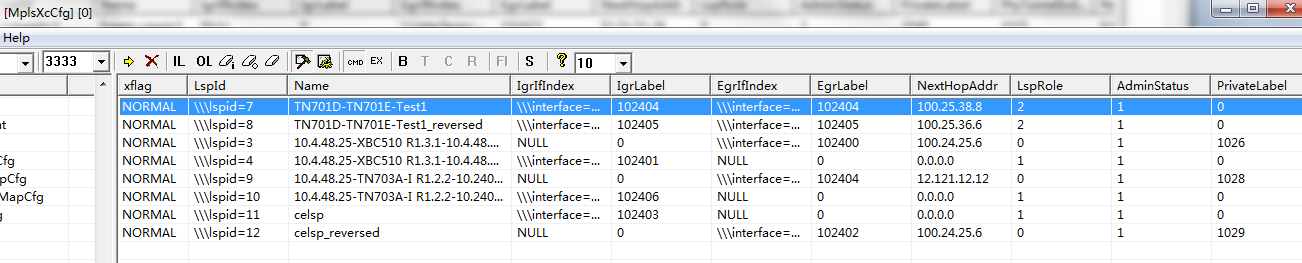
MplsXcCfg （AZ点Sample）

IfType=Normal， IfIndex 填interface=\\\xxxx， ClientIndex填空

IfType=FlexE， IfIndex填空，clientIndex填数值



MplsXcCfg （P点Sample）





由入节点分配入向标签

系统会按照~~出Label端口唯一（TBD）~~、进Label网元唯一的要求产生可用的Label

【】MPLS-TP Tunnel管理要求

创建 （具体表项参见SRTunnel对照表）

激活

去激活

删除

查询

【】MPLS-TP Tunnel LSP标签分配规则

LSP标签池范围：同PTN LSP标签

LSP标签池属性：属SRLB

下面是一个方向的LSP标签分配伪代码：

//从末节点开始做标签分配 （同LDP协议，下游节点决定入标签，告知上游节点做出标签）

node = thisLSP.getEndNode();

headNode = thisLSP.getHeadNode();

previousNode = thisLSP.getEndNode().getPreviousNode();

newLabel = node.applyLocalSID("LSP");

while (previousNode != headNode) {

//网管里有中间节点的出标签在出端口内唯一，即一个出端口没有两个重的出标签

//但和网元讨论后，认为没有这个必要，所以这里没做这个限制

previousNode.getLSP(thisLSP).setEgressLabel(newLabel);

node.getLSP(thisLSP).setIngressLabel(newLable);

node = previousNode;

prevousNode = node.getPreviousNode();

newLabel = node.applyLocalSID("LSP");

}

//首节点的出标签不能重复 （这条逻辑可不做）

while ( !headNode.isUniqueLabeLInSystem(newLabel) ) {

excludeSet.add(newLabel);

newLabel = node.applyLocalSID("LSP", excludeSet);

}

headNode.getLSP(thisLSP).setEgressLabel(newLabel);

node.getLSP(thisLSP).setIngressLabel(newLable);

【】MPLS LSP寻路拓扑

以下选项需支持

基于FlexE Channel拓扑寻路 （即MplsXcCfg表里的iftype为FlexE）

基于Physical Topology寻路（即MplsXcCfg表里的iftype为Normal）

基于Datalink Topology寻路（即包括了Physical Topology和FlexE Channel）拓扑寻路

TBD 【】MPLS LSP/PW 标签的手工输入

因互通对接要求

## SRTunnel管理要求

SR Tunnel的几种类型

单向 SRTP/SRBE Tunnel

双向 SRTP/SRBE Tunnel

Binding SRTP Tunnel （仅用来做binding sidlist处理，不表示端到端tunnel）



图表 2 SRTunnel示例

网元规格：SRTunnelCfg

LSP/Path的支持数据量能达100K

PathSID 在SRLB内

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item Name | Data Type | Limit | Default Value | Yang Item Mapping | Description |
| SRTunnelid | String | 1…16K | 1 | 需扩展  tunnel:snc-tunnel/tunnel:id用uuid，在Controller系统用， | Key  \\\srtunnelid=x |
| Name | String |  |  | tunnel:snc-tunnel/tunnel:name |  |
| BindingSID | Long | 0, 16..1048576 | 0 | 需扩展 | 0 means no binding SID is configured |
| IngressPathSID | Long | 0, 16..1048576 | 0 | 需扩展  tunnel:snc-lsp/tunnel:id 表示lsp/path的ID， | 0 means no ingress path SID is configured |
| EgressPathSID | Long | 0, 16..1048576 | 0 | 需扩展  tunnel:snc-lsp/tunnel:id 表示lsp/path的ID，  EgressPathSID 找到tunnel:snc-lsp/tunnel:id 即path id， 然后path id对应到的route id /route:service/route:snc-routes/route:snc-route/route:snc-id | 0 means no egress path SID is configured |
| SIDListType | Octet | MPLS label,  IPv4 prefix,  IPv6 prefix(TBD), Interface(R1.2) | MPLS label |  |  |
| SIDList | String | {xx,xx,xx} |  | 取自route:snc-route/route:xcs/route:xc-list |  |
| Algorithm | Octet | SPF, Strict SPF | SPF |  | Hidden |
| AdminStatus | Octet | up, down,testing |  |  | Hidden |

网元最多支持N层标签栈，包括最底层的Path-id和PW-id。所以SIDList的长度是N-2

（第一阶段，N可能只有4层）

PTN网管里的tunnel模型：双向的两条共路LSP，规格表tunnelCfg

SR Controller里的tunnel模型：SRTunnelCfg，只需在首尾节点设置（binding除外）

SRTunnel模型里的pathSID表示一条SRTP路径。首节点的egressPathSID就是尾节点的ingressPathSID，中间节点不管这个PathSID。（PathSID由宿PE节点向源PE节点分配的本地标签）

一条SRTunnel里的Path是一条双向lsp模型。

获取PathSID接口的要求

~~applyGlobalPathSID(appliedPathId)，由 Controller系统 从Path SID池分配一个未被使用最小SID值。（为节省全局标签资源，一条双向Path的两个方向pathSID相同）~~

applyLocalPathSID(ingress-node-id,appliedPathId)，由 目的节点(即入节点) 从SRLB中分配 一个未被使用最小的本地SID值。

SRTunnel支持双向共路

YANG模型：sptn-service-tunnel.yang

【】MSD - Maximum Segment/SID Depth，即Tunnel SID List里的最大SID个数，目前网元仅支持4，在SRTP场景下，还要减去PathSID, PWSID，所以首节点仅支持2层SID，首节点样例：HeadNodeAdjSID, NextHopAdjSID, PathSID, PWSID；中间节点支持4层，样例：LocalNodeAdjSID, NextHopAdjSID, NextHopAdjSID, NextHopAdjSID, PathSID, PWSID。 因为PGA的处理能力就是新打的出标签就3层。

【】SR Tunnel的快捷创建并激活

用户直接在物理层拓扑上选源宿节点（参考SR的快捷业务创建方式）做Tunnel创建。

【】手工创建SRTP Tunnel （包括Binding），需支持单双向SR Tunnel和Bing Tunnel的创建。

测试方案8.2.4，

d. NE5 上配置 Binding 标签 BSID5：{NE5-NE6-NE8-NE7}

e. 部署 NE2 和 NE7 之间的 SRTP-TE 隧道 1: {NE2-NE1-BISD5}；

从测试要求看，先建好Binding，然后SRTP Tunnel选走Binding

要求支持手工创建SRTP Tunnel，因为要求的路不是自动寻路结果：最短路径 （或提供ERO的路径计算）

【】当以SRBE方式创建SR Tunnel时，根据单双向选择， 在单端或两端网元上下发SR Tunnel配置，SIDList填对端网元的NodeSID。

【】自动创建最短路径的SRTP Tunnel，需支持单双向SR Tunnel和Bing Tunnel的创建。

基于SR topology计算

用户输入首尾节点。

Controller根据当前的IGP Topology，计算一条最短路径（不支持ECMP）。并列出根据节点和接口AdjSID串接而形成的SIDList。

用户输入首尾节点，和必经过的节点顺序，比如[Node1 -> Node2]，则Controller将计算NodeA->Node1, Node1->Node2, Node2->NodeZ的最短路径，并列出拼接后的SIDList。

用户输入首尾节点，和必不经过的节点集合，比如（Node1, Node2），则Controller将计算NodeA->NodeZ的最短路径，并避绕Node1和Node2，并列出途径的SIDList。

中移动Case：

8.2.2 SR-TP 隧道的算路策略测试

图表 1 CMCC SPN测试用例网络拓扑

...

d） 部署 NE1 和 NE7 之间的 SRTP-TE 隧道 1；

e） 部署 NE1 和 NE7 之间的 SR-TP 隧道 2，指定路径约束不经过 NE5；

f） 部署 NE1 和 NE7 之间的 SR-TP 隧道 3，指定路径约束经过 NE6，且不经过

NE2；

...

【TBD】Binding SID在SR Tunnel路径计算的支持

如果说Adj SID是一条到邻居节点的链路，那么Binding SID可以认为是一条到远端节点的逻辑链路。可以把Binding SID放置到SR Topology这层，这样和Node SID（代表点），Adj SID（代表节点间直接相接的边），Binding SID（代表节点间间接相接的边）。 参见*Link SID*。

【】隧道方向：

默认双向，即有A->Z的Path，也有Z->A的Path。

单向的Tunnel也需支持。

【】用户可以将Controller计算得出的隧道路径保存在Controller，数据模型是 sptn-service-tunnel

/tunnel:service/tunnel:snc-tunnels/tunnel:snc-tunnel，

【】用户可以浏览隧道信息，包括：

隧道基本信息，包括隧道名称，创建日期，创建者，激活状态，激活时间，方向（单，双向）

路径视图，包括首尾节点，经过节点和链路，链路需标识链路端点所在端口

标签列表

【】用户可以将Tunnel下发到网元，即激活Tunnel，模型的snc-tunnel就是网元的SRTunnel，snc-tunnel里的LSP就是网元SRTunnel内的Path。数据样例参考 图表 2 SRTunnel示例 。网元下发涉及到头端和末端节点，需用网络级配置事务保证。(Binding的隧道就只需下发到所在配置网元节点即可。)

（网元SRS里要求：一条Path只能和一条Tunnel相关联；网管的实现是Tunnel就是两条LSP组成；建议Controller里实现的Tunnel和Path关系，可参考网管里的Tunnel和LSP关系。）

【】对已处在激活状态的SRTunnel（admin-status=up），用户不能修改该tunnel数据；对未激活的SRTunnel（admin-status=down），用户可以修改该tunnel数据内容，包括：首尾节点，SIDList等。

【】用户可以将已处在激活状态的SRTunnel去激活。

【TBD】LSP层的拓扑数据需和tunnel配置数据模型同步

LSP层拓扑

/nt:network-topology/nt:topology/nettopology:aptn-topology-attributes/nettopology:layer-rate=datalink

/nt:network-topology/nt:topology/nettopology:aptn-topology-attributes/nettopology:sub-layer-rate=lsp

sptn-service-tunnel模型到拓扑模型映射：

/tunnel:service/tunnel:snc-tunnels/tunnel:snc-tunnel/tunnel:id

=   
/nt:network-topology/nt:topology/nt:link/nt:link-id

/tunnel:service/tunnel:snc-tunnels/tunnel:snc-tunnel/tunnel:source-ne-id

=

/nt:network-topology/nt:topology/nt:node/nt:node-id

/nt:network-topology/nt:topology/nt:link/nt:source/nt:source-node

/tunnel:service/tunnel:snc-tunnels/tunnel:snc-tunnel/tunnel:destination-ne-id

=

/nt:network-topology/nt:topology/nt:node/nt:node-id

/nt:network-topology/nt:topology/nt:link/nt:destination/nt:dest-node

LSP拓扑层的Subnet-node关系 默认继承 底层拓扑的Subnet-node关系。即如果tunnel端点的node在IGP层属于一个有subnet关联，那么这个subnet也存在于LSP拓扑层

SID Label List 可以通过 /route:service/route:snc-routes/route:snc-route/route:snc-id 即tunnel-id 将路由关联到path，path关联到SRTunnel。

TBD 【】Binding Tunnel 创建的Binding SID 将同步到SR Topology层，作为一个特殊的Adj SID。这样的边可以做为Tunnel SID List路径途径的一个SID。

**TBD** Tunnel/LSP 保护

## OAM

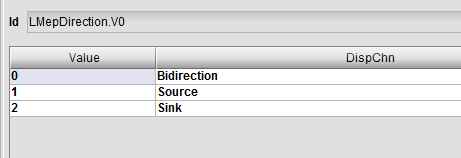
对MPLSTP Tunnel， SRTP Tunnel里的每条LSP/Path要支持OAM配置

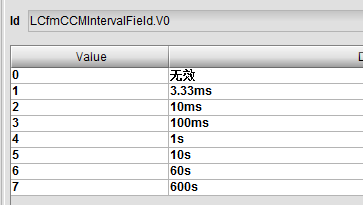
MplsTpOam1MegCfg

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item Name | Data Type | Limit | Default Value | Yang Item Mapping | Description |
| MegIndex | Long | 1-16K |  | yang item不加了，如果要去激活通过ne表的megname来查 | 自增 |
| ICC | String | 6 chars |  | meg-id 前6个字符 | cncmcc |
| UMC | String | 7 chars |  | meg-id 后7个字符 | 一个7位数，首尾两端一致  比如：  cncmcc0000001 |
| MegName | String | 128 varchar |  | name（需唯一） |  |  | 0, 16..1048576 | 0 | 需扩展  tunnel:snc-lsp/tunnel:id 表示lsp/path的ID，  EgressPathSID 找到tunnel:snc-lsp/tunnel:id 即path id， 然后path id对应到的route id /route:service/route:snc-routes/route:snc-route/route:snc-id |
| MegType | Octet |  |  | 6 - LSP MEG, 2 - PW MEG, 7 - SR Tunnel MEG  根据tunnel类型：如果是SRTP Tunnel，则SR Tunnel MEG；如果是MPLSTP Tunnel，则LSP MEG |  |
| Fid | String |  |  | SRTP: \\\srtunnelid=xxx（其实表示的是一条Path）  MPLSTP LSP:\\\tunnelid=xxx（其实表示的是一条双向lsp） |  |
| ReverseLSP | String |  |  | 不下发 |  |
|  |  |  |  | belonged-id：相关LSP（snc-lsp id）或PW id |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

MplsTpOam1MepConfig

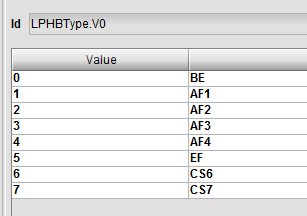
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Item Name | Data Type | Limit | Default Value | Yang Item Mapping | Description |
| MepId |  |  |  | ... tunnel:meps/tunnel:mep/tunnel:id | 自增 |
| MegIndex |  |  |  | 关联上述megindex |  |
| MepName |  |  |  | ... tunnel:meps/tunnel:mep/tunnel:name |  |
| MepDirection |  |  |  | 0 |  |
| PeerMepId |  |  |  | 对端节点配置的mepid |  |
| CcmFeqType |  |  |  | cc-interval  看LCfmCCMIntervalField枚举值 |  |
| CcmTxEn |  |  |  | cc-allow |  |
| CcmRxEn |  |  |  | cc-allow |  |
| CcmEnLM |  |  |  | disable |  |
| CcmPHB |  |  |  |  | Hidden |
|  |  |  |  | 其他字段值不设置 |  |
|  |  |  |  |  |  |





|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| cc-interval | decimal64 | CRU | 否 | 3.3ms | 3.3、10、100、1000ms | CC报文的发送间隔 |

cc-exp不支持，即CcmPHB，默认用CcmPHB(7)，Hidden字段



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| cc-exp | enum | CRU | 否 | CS7 | BE：0  AF1：1  AF2：2  AF3：3  AF4：4  EF：5  CS6：6  CS7：7 | CC报文的发送优先级 |

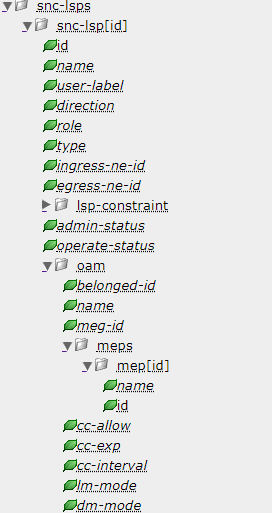
CcmTxEn(1),CcmRxEn(1)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| cc-allow | boolean | CRU | 否 | 0 ：不允许 | 0 ：不允许  1 ：允许  用于保护时，固定为“允许”。  仅当cvAllow为允许的情况下才支持该字段。 | 连接确认（CC）允许 |

LM，DM默认Disable

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| lm-mode | enum | CRU | 否 | disable | disable：0  preactive：1  on-demand：2 | Disable  Preactive  On-demand |
| dm-mode | enum | CRU | 否 | disable | disable：0  preactive：1  on-demand：2 | Disable  Preactive  On-demand |

Yang：



## APS

## 网元命令脚本

SRTP:

47

$.dbgcli.0.Cmd "addasyn valid 1 SRTunnelCfg {SRTunnelid(\\\srtunnelid=2),BindingSID(0),IngressPathSID(1247),EgressPathSID(1259),SIDListType(0),SIDList(2047/,2258)}"

$.dbgcli.0.Cmd "addasyn valid 1 PwCfg {PwId(\\\pwid=65542),Name(pw24759),PwType(0),PwOutLabel(204802),PwInLabel(204802),PwMplsType(2),PwAttachedId(NULL),TunnelId(\\\srtunnelid=2),InBoundIfIndex(NULL),OutBoundIfIndex(NULL)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRTunnelCfg {SRTunnelid(\\\srtunnelid=6),BindingSID(0),IngressPathSID(1647),EgressPathSID(1659),SIDListType(0),SIDList(2147/,148)}"

interface ethernet 1/1

set etherac ingress sp-vid 20 mapto pwid 65542

exit

48

$.dbgcli.0.Cmd "addasyn valid 1 SRTunnelCfg {SRTunnelid(\\\srtunnelid=2),BindingSID(148),IngressPathSID(0),EgressPathSID(0),SIDListType(0),SIDList(2248/,2046)}"

46

$.dbgcli.0.Cmd "addasyn valid 1 SRTunnelCfg {SRTunnelid(\\\srtunnelid=2),BindingSID(146),IngressPathSID(0),EgressPathSID(0),SIDListType(0),SIDList(2146/,2148)}"

59

$.dbgcli.0.Cmd "addasyn valid 1 SRTunnelCfg {SRTunnelid(\\\srtunnelid=2),BindingSID(0),IngressPathSID(1259),EgressPathSID(1247),SIDListType(0),SIDList(2059/,2058)}"

$.dbgcli.0.Cmd "addasyn valid 1 PwCfg {PwId(\\\pwid=65542),Name(pw24759),PwType(0),PwOutLabel(204802),PwInLabel(204802),PwMplsType(2),PwAttachedId(NULL),TunnelId(\\\srtunnelid=2),InBoundIfIndex(NULL),OutBoundIfIndex(NULL)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRTunnelCfg {SRTunnelid(\\\srtunnelid=6),BindingSID(0),IngressPathSID(1659),EgressPathSID(1647),SIDListType(0),SIDList(2159/,146)}"

interface ethernet 1/1

set etherac ingress sp-vid 20 mapto pwid 65542

exit

### OAM & APS

47

$.dbgcli.0.Cmd "addasyn valid 1 MplsTpOam1MegCfg {MegIndex(2),ICC(123456),UMC(abcdefh),MegName(meg472),MegType(7),Fid(\\\srtunnelid=2),ReverseLSP(Empty)}"

$.dbgcli.0.Cmd "addasyn valid 1 MplsTpOam1MepConfig {MepId(472),MegIndex(2),MepName(mep472),MepDirection(0),PeerMepId(592),CcmFeqType(1),CcmTxEn(1),CcmRxEn(1),CcmEnLM(0),CcmPHB(7),SDEn(0),SDThreshold(0),RdiTxEn(1),AisTxEn(1),AisFeqType(10),AisPHB(7),LckEn(0),LckFeqType(10),LckPHB(7),CsfEn(0),CsfFeqType(10),CsfPHB(7),LMEn(0),LMPHB(7),LMFeqType(1),LMDuration(1),DMType(0),DMEn(0),DMPHB(7),DMFeqType(1),DMDuration(1),LMTxEn(0),LMRxEn(0),LMSDEn(0),LMSDThr(30),LMSDBs(10),LMSDGs(10),ONEWAYDMSDEn(0),ONEWAYDMSDThr(10),ONEWAYDMSDBs(10),ONEWAYDMSDGs(10),TWOWAYDMSDEn(0),TWOWAYDMSDThr(10),TWOWAYDMSDBs(10),TWOWAYDMSDGs(10)}"

$.dbgcli.0.Cmd "addasyn valid 1 MplsTpOam1MegCfg {MegIndex(6),ICC(123456),UMC(abcdefj),MegName(meg476),MegType(7),Fid(\\\srtunnelid=6),ReverseLSP(Empty)}"

$.dbgcli.0.Cmd "addasyn valid 1 MplsTpOam1MepConfig {MepId(476),MegIndex(6),MepName(mep476),MepDirection(0),PeerMepId(596),CcmFeqType(1),CcmTxEn(1),CcmRxEn(1),CcmEnLM(0),CcmPHB(7),SDEn(0),SDThreshold(0),RdiTxEn(1),AisTxEn(1),AisFeqType(10),AisPHB(7),LckEn(0),LckFeqType(10),LckPHB(7),CsfEn(0),CsfFeqType(10),CsfPHB(7),LMEn(0),LMPHB(7),LMFeqType(1),LMDuration(1),DMType(0),DMEn(0),DMPHB(7),DMFeqType(1),DMDuration(1),LMTxEn(0),LMRxEn(0),LMSDEn(0),LMSDThr(30),LMSDBs(10),LMSDGs(10),ONEWAYDMSDEn(0),ONEWAYDMSDThr(10),ONEWAYDMSDBs(10),ONEWAYDMSDGs(10),TWOWAYDMSDEn(0),TWOWAYDMSDThr(10),TWOWAYDMSDBs(10),TWOWAYDMSDGs(10)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRTunnelApsCfg {ApsId(\\\apsid=1),ApsType(0),ApsMode(1),ApsKind(0),HoldOffTime(0),WtrTime(5),ExtApsCmd(0),ProtectId(1)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRBiApsTunnelCfg {ApsId(\\\apsid=1),TunnelId(\\\srtunnelid=2),Type(0),Priority(1),Candidate(\\\srtunnelid=2),Id(1)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRBiApsTunnelCfg {ApsId(\\\apsid=1),TunnelId(\\\srtunnelid=6),Type(1),Priority(1),Candidate(\\\srtunnelid=2),Id(1)}"

59

$.dbgcli.0.Cmd "addasyn valid 1 MplsTpOam1MegCfg {MegIndex(2),ICC(123456),UMC(abcdefh),MegName(meg592),MegType(7),Fid(\\\srtunnelid=2),ReverseLSP(Empty)}"

$.dbgcli.0.Cmd "addasyn valid 1 MplsTpOam1MepConfig {MepId(592),MegIndex(2),MepName(mep592),MepDirection(0),PeerMepId(472),CcmFeqType(1),CcmTxEn(1),CcmRxEn(1),CcmEnLM(0),CcmPHB(7),SDEn(0),SDThreshold(0),RdiTxEn(1),AisTxEn(1),AisFeqType(10),AisPHB(7),LckEn(0),LckFeqType(10),LckPHB(7),CsfEn(0),CsfFeqType(10),CsfPHB(7),LMEn(0),LMPHB(7),LMFeqType(1),LMDuration(1),DMType(0),DMEn(0),DMPHB(7),DMFeqType(1),DMDuration(1),LMTxEn(0),LMRxEn(0),LMSDEn(0),LMSDThr(30),LMSDBs(10),LMSDGs(10),ONEWAYDMSDEn(0),ONEWAYDMSDThr(10),ONEWAYDMSDBs(10),ONEWAYDMSDGs(10),TWOWAYDMSDEn(0),TWOWAYDMSDThr(10),TWOWAYDMSDBs(10),TWOWAYDMSDGs(10)}"

$.dbgcli.0.Cmd "addasyn valid 1 MplsTpOam1MegCfg {MegIndex(6),ICC(123456),UMC(abcdefj),MegName(meg596),MegType(7),Fid(\\\srtunnelid=6),ReverseLSP(Empty)}"

$.dbgcli.0.Cmd "addasyn valid 1 MplsTpOam1MepConfig {MepId(596),MegIndex(6),MepName(mep596),MepDirection(0),PeerMepId(476),CcmFeqType(1),CcmTxEn(1),CcmRxEn(1),CcmEnLM(0),CcmPHB(7),SDEn(0),SDThreshold(0),RdiTxEn(1),AisTxEn(1),AisFeqType(10),AisPHB(7),LckEn(0),LckFeqType(10),LckPHB(7),CsfEn(0),CsfFeqType(10),CsfPHB(7),LMEn(0),LMPHB(7),LMFeqType(1),LMDuration(1),DMType(0),DMEn(0),DMPHB(7),DMFeqType(1),DMDuration(1),LMTxEn(0),LMRxEn(0),LMSDEn(0),LMSDThr(30),LMSDBs(10),LMSDGs(10),ONEWAYDMSDEn(0),ONEWAYDMSDThr(10),ONEWAYDMSDBs(10),ONEWAYDMSDGs(10),TWOWAYDMSDEn(0),TWOWAYDMSDThr(10),TWOWAYDMSDBs(10),TWOWAYDMSDGs(10)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRTunnelApsCfg {ApsId(\\\apsid=1),ApsType(0),ApsMode(1),ApsKind(0),HoldOffTime(0),WtrTime(5),ExtApsCmd(0),ProtectId(1)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRBiApsTunnelCfg {ApsId(\\\apsid=1),TunnelId(\\\srtunnelid=2),Type(0),Priority(1),Candidate(\\\srtunnelid=2),Id(1)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRBiApsTunnelCfg {ApsId(\\\apsid=1),TunnelId(\\\srtunnelid=6),Type(1),Priority(1),Candidate(\\\srtunnelid=2),Id(1)}"

## 界面参考

SRBE/SRTP Tunnel的SID List 显示



Tunnel拓扑展示



# PW管理

PTN网管的PW创建 和 Controller里PW创建 规则一样

目前没有多段PW

PW，伪纤PseudoWire，专供给某一业务使用的虚电路。VPWS业务就一条PW。VPLS/L3VPN可以有多条PW组成。PW封装进LSP，网元实现将PW绑定到tunnel（PWCfg）。

PW 标签和VPN id相关联。

PW 标签分配规则：

下游分配入标签，上游出标签同下游入标签

入标签节点系统内唯一

~~出标签节点系统内唯一~~

一个节点上的入标签和出标签可以相同

YANG模型：网管LSP模型？ SPTN模型？

中移动的PW模型跟着业务走，如eline模型里的PW：

/eline:service/eline:snc-elines/eline:snc-eline/eline:snc-pws

为业务创建便捷考虑，PW随业务创建而创建。

~~扩展移动模型，将PW做一个container，和sptn-service-tunnel同层。~~

~~PW下的项目加route id，关联到路径~~

# 业务管理

PW关联Tunnel的模型如下：



注：EthPwCfg应改为EthSvcPWCfg

L3VPNIPv4AC/ L3VPNIPv6AC

L3VPNEthPW关联到指定PW

需支持L3VPN，VPLS，VPWS （先支持VPWS，然后是L3VPN）

注：

SPN系列设备兼容PTN业务和SR业务，网元的业务配置数据模式还是PTN模式，即还是使用网管的配置数据模型，仅仅以SRTunnel表加入了一种LSP Tunnel方式而已。

【】PW-Only 的支持 即PW直接建立在DL上，这里的DL指：AZ间直接物理连接相邻，这种连接方式可以是PTN的DataLink，或FlexE的Client，或FlexE的Channel。 这种情况下，就不用LSP/Path上的标签交换了，即使AZ路径上没有Tunnel创建，依然可以在DL上直接建PW。

多段PW（支持？？ ）

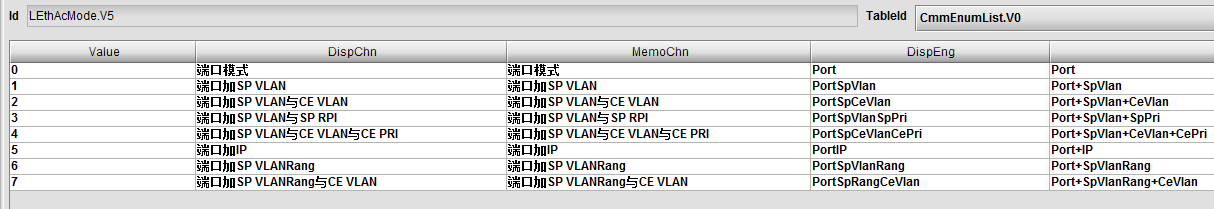
## VPWS

### 业务模型

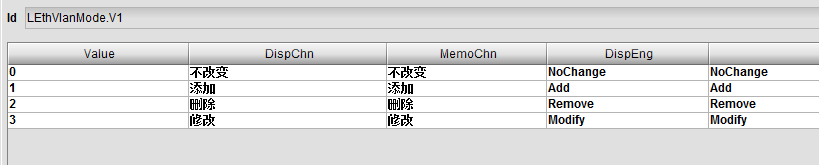
/eline:service/eline:snc-elines/eline:snc-eline

|  |  |  |
| --- | --- | --- |
| Field Item | Description |  |
| id | 业务id，uuid表示 |  |
| name |  |  |
| user-label |  |  |
| parent-ncd-id | 所在domain id |  |
| admin-status |  |  |
| operator-status |  |  |
| snc-type | 上下业务类型，我们是双向业务，类型值simple |  |
| ingress-end-points | 即A->Z的A点，Z->A的Z点，具体见下面的EthAcCfg表， |  |
| egress-end-points | 即A->Z的Z点，Z->A的A点，具体见下面的EthAcCfg表 |  |
| snc-switch | 保护倒换，... |  |
| snc-pws | 见下表PWCfg |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

LEthAcMode



LEthVlanMode



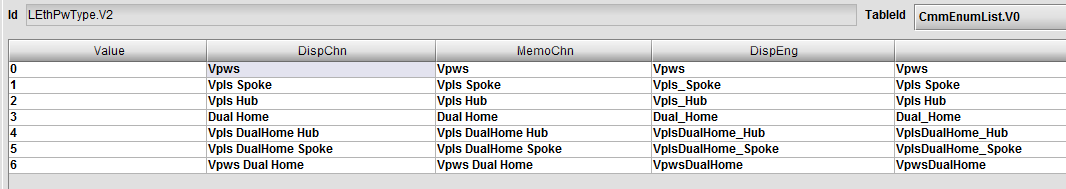
#### 业务端口 EthACCfg

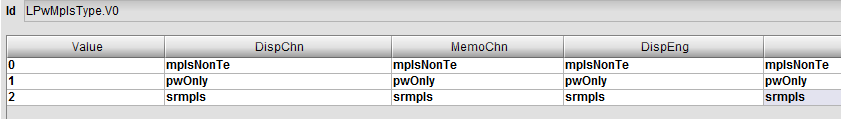
|  |  |  |  |
| --- | --- | --- | --- |
| Field Item | Sample | Description | Yang Item |
| EthAcId | \\\ethacid=x | 本网元当前最大值加1，下发时生成 | 即ep.id， |
| IfIndex | \\\interface=x | eth1.1的值  参考网管的AC口IfIndex生成算法 | 能和相映射  源端口：  /eline:ingress-end-point/eline:ne-id  +/eline:ingress-end-point/eline:ltp-id  宿端口：  egress |
| InSpVlan | 1 | 1-4094 | eline:dot1q-vlan-bitmap or  eline:qinq-svlan-bitmap |
| InCeVlan | 1 | 1-4094 | eline:qinq-cvlan-bitmap |
| EthAcMode | 0 | LEthAcMode  : 0,1,2 | /eline:xxx-end-point/eline:access-type  Port, Dot1Q, QinQ |
| OutSpVlan | 1 |  | eline:xxxx-end-point/eline:action-vlan-id,表示本端点出Vlan |
| OutCeVlan | 1 |  | 忽略，默认填1 |
| OutSpVlanMode | 0 | LEthVlanMode  :0,1,2,3 | eline:xxx-end-point/eline:access-action，表示本端点出Vlan操作 |
| EthPrepId | \\\vpwsprepid=20 or \\\vplsprepid=x | Vpn id | node-vpwsprep-id (新增) |
| Pri | 0 | 0-7 | eline:xxx-end-point/eline:qos/eline:traffic-class |
| TrustCeQos | True | Boolean |  |
| TrafficParamDescr | 空窜 |  |  |
| IsETreeLeaf | False |  |  |
| SpVlanRang | 空窜 |  |  |
| IveSpVlan | 1 | 1-4094 |  |
| IveSpVlanMode | 0 | LEthVlanMode  :0,1,2,3 |  |
|  |  |  |  |
|  |  |  | eline:ingress-end-point/eline:role master or slave |
|  |  |  | /eline:ingress-end-point/eline:id 端口编号，uuid |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

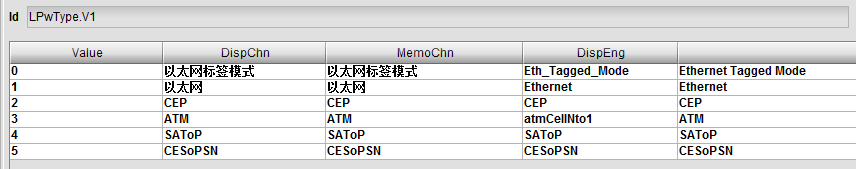
#### SRTunnelCfg

【】创建业务时，若业务端点间没有SR Tunnel，则用户可以选择SR Tunnel可以随业务创建而自动创建；若已有SR Tunnel，也可以作为PW的选路路径。

#### PWCfg







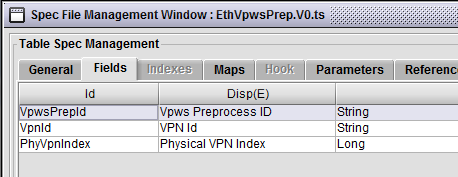
【】随业务产生的PW需同步到PW Topo配置树

|  |  |  |  |
| --- | --- | --- | --- |
| Field Item | Sample | Description | Yang Item |
| PwId | \\\pwid=65542 | 网元当前最大值加1 | node-pw-id(新增) |
| Name | pw24748 |  | /eline:snc-pw/eline:name |
| PwType | 0 | LPwType, 0,1 | eline:snc-pw/eline:encaplate-type  取值：ethernet-vlan 4，ethernet 5 |
| PwOutLabel | 204802 |  | /route:service/route:snc-routes/route:snc-route/route:snc-id = eline:snc-pw/eline:id，通过VPWS的pw id找到route，route里的forward-out-label |
| PwInLabel | 204802 |  | route里的forward-in-label  入标签有本地分配，本端的入就是对端的出标签 |
| PwMplsType | 2 | 值2，表示srmpls，即tunnel id关联到SRTunnel | pw-mpls-type(新增) |
| PwAttachedId | 可以不下该项目 |  |  |
| TunnelId | \\\srtunnelid=2 |  | /eline:tunnel-ids/eline:tunnel-id/eline:tunnel-id |
| InBoundClientIndex | 可以不下该项目 |  | 当所关联Tunnel是FlexE Channel或FlexE Client时，下业务点需填 |
| OutBoundClientIndex | 可以不下该项目 |  | 当所关联Tunnel是FlexE Channel或FlexE Client时，上业务点需填 |
|  |  |  | eline:snc-pw/eline:admin-status |
|  |  |  | /eline:snc-pw/eline:operate-status |
|  |  |  | eline:snc-pw/eline:user-label |
|  |  |  | /eline:snc-pw/eline:sn-support |
|  |  |  | /eline:snc-pw/eline:conn-ack-type |
|  |  |  | /eline:snc-pw/eline:vccv-type |
|  |  |  | eline:snc-pw/eline:ctrl-word-support |
|  |  |  | /eline:snc-pw/eline:oam... |
|  |  |  | /eline:snc-pw/eline:qos... |
|  |  |  | eline:snc-eline/eline:snc-switch... |
|  |  |  |  |
|  |  |  | /eline:snc-eline/eline:parent-ncd-id |
|  |  | simple,... | /eline:snc-eline/eline:snc-type |
|  |  |  | eline:snc-pw/eline:id uuid，pw系统id |

#### PW绑业务 EthSvcPWCfg

|  |  |  |  |
| --- | --- | --- | --- |
| Field Item | Sample | Description |  |
| id | \\\id=xxx |  | node-svcpw-id(新增) |
| PwId | \\\pwid=65542 |  | node-pw-id(新增) 同前 |
| PwVlan | 1 | 1-4094 | 忽略，填1 |
| PwVlanMode | 0 | LEthVlanMode | 忽略，填0 |
| EthPwType | 0 | 0-vpws | 暂忽略，填0 |
| VpnId | \\\vpnid=20 |  | node-vpn-id(新增) |
|  |  |  |  |
|  |  |  | /eline:service/eline:snc-elines/eline:snc-eline/eline:id  Uuid表示Controller存储的系统业务id |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

#### VPWS业务表 EthVpwsPrep



|  |  |  |  |
| --- | --- | --- | --- |
| Field Item |  |  |  |
| VpwsPrepId | \\\vpwsprepid=20 | node-vpwsprep-id |  |
| VpnId | \\\vpnid=20 | node-vpn-id(新增) |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### 测试样例

#### 脚本参考

##### SID

47

$.dbgcli.0.Cmd "addasyn valid 1 SRLocalAdjSIDCfg {LocalPrefix(10.47.58.47),LocalPrefixMask(24),RemotePrefix(10.47.58.58),RemotePrefixMask(24),AdjSIDType(0),AddressFamily(0),AdjSID(2047),BackUp(0),Local(0),Weight(1)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRLocalAdjSIDCfg {LocalPrefix(10.47.48.47),LocalPrefixMask(24),RemotePrefix(10.47.48.48),RemotePrefixMask(24),AdjSIDType(0),AddressFamily(0),AdjSID(2147),BackUp(0),Local(0),Weight(1)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRLocalPrefixSIDCfg {Prefix(1.1.1.47),PrefixMask(24),PrefixType(0),Name(index47),AddressFamily(0),StartSID(47),Range(1),PrefixSIDType(1),Algorithm(0),LastHopBehavior(0),PhyId(0)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRRemotePrefixSIDCfg {Prefix(1.1.1.58),PrefixMask(24),PrefixType(0),Name(index58),AddressFamily(0),StartSID(58),Range(1),PrefixSIDType(1),Algorithm(0),LastHopBehavior(0),PhyId(0)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRRemotePrefixSIDCfg {Prefix(1.1.1.48),PrefixMask(24),PrefixType(0),Name(index48),AddressFamily(0),StartSID(48),Range(1),PrefixSIDType(1),Algorithm(0),LastHopBehavior(0),PhyId(0)}"

58

$.dbgcli.0.Cmd "addasyn valid 1 SRLocalAdjSIDCfg {LocalPrefix(10.47.58.58),LocalPrefixMask(24),RemotePrefix(10.47.58.47),RemotePrefixMask(24),AdjSIDType(0),AddressFamily(0),AdjSID(2058),BackUp(0),Local(0),Weight(1)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRLocalAdjSIDCfg {LocalPrefix(10.58.48.58),LocalPrefixMask(24),RemotePrefix(10.58.48.48),RemotePrefixMask(24),AdjSIDType(0),AddressFamily(0),AdjSID(2158),BackUp(0),Local(0),Weight(1)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRLocalPrefixSIDCfg {Prefix(1.1.1.58),PrefixMask(24),PrefixType(0),Name(index58),AddressFamily(0),StartSID(58),Range(1),PrefixSIDType(1),Algorithm(0),LastHopBehavior(0),PhyId(0)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRRemotePrefixSIDCfg {Prefix(1.1.1.47),PrefixMask(24),PrefixType(0),Name(index47),AddressFamily(0),StartSID(47),Range(1),PrefixSIDType(1),Algorithm(0),LastHopBehavior(0),PhyId(0)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRRemotePrefixSIDCfg {Prefix(1.1.1.48),PrefixMask(24),PrefixType(0),Name(index48),AddressFamily(0),StartSID(48),Range(1),PrefixSIDType(1),Algorithm(0),LastHopBehavior(0),PhyId(0)}"

48

$.dbgcli.0.Cmd "addasyn valid 1 SRLocalAdjSIDCfg {LocalPrefix(10.58.48.48),LocalPrefixMask(24),RemotePrefix(10.58.48.58),RemotePrefixMask(24),AdjSIDType(0),AddressFamily(0),AdjSID(2048),BackUp(0),Local(0),Weight(1)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRLocalAdjSIDCfg {LocalPrefix(10.47.48.48),LocalPrefixMask(24),RemotePrefix(10.47.48.47),RemotePrefixMask(24),AdjSIDType(0),AddressFamily(0),AdjSID(2148),BackUp(0),Local(0),Weight(1)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRLocalPrefixSIDCfg {Prefix(1.1.1.48),PrefixMask(24),PrefixType(0),Name(index48),AddressFamily(0),StartSID(48),Range(1),PrefixSIDType(1),Algorithm(0),LastHopBehavior(0),PhyId(0)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRRemotePrefixSIDCfg {Prefix(1.1.1.58),PrefixMask(24),PrefixType(0),Name(index58),AddressFamily(0),StartSID(58),Range(1),PrefixSIDType(1),Algorithm(0),LastHopBehavior(0),PhyId(0)}"

$.dbgcli.0.Cmd "addasyn valid 1 SRRemotePrefixSIDCfg {Prefix(1.1.1.47),PrefixMask(24),PrefixType(0),Name(index47),AddressFamily(0),StartSID(47),Range(1),PrefixSIDType(1),Algorithm(0),LastHopBehavior(0),PhyId(0)}"

##### SRTP case

47

$.dbgcli.0.Cmd "addasyn valid 1 SRTunnelCfg {SRTunnelid(\\\srtunnelid=2),BindingSID(0),IngressPathSID(0),EgressPathSID(1247),SIDListType(0),SIDList(2047/,2158)}"

$.dbgcli.0.Cmd "addasyn valid 1 PwCfg {PwId(\\\pwid=65542),Name(pw24748),PwType(0),PwOutLabel(204802),PwInLabel(204802),PwMplsType(2),PwAttachedId(NULL),TunnelId(\\\srtunnelid=2),InBoundIfIndex(NULL),OutBoundIfIndex(NULL)}"

interface ethernet 1/1

set etherac ingress sp-vid 20 mapto pwid 65542

exit

48

$.dbgcli.0.Cmd "addasyn valid 1 SRTunnelCfg {SRTunnelid(\\\srtunnelid=2),BindingSID(0),IngressPathSID(1247),EgressPathSID(0),SIDListType(0),SIDList()}"

$.dbgcli.0.Cmd "addasyn valid 1 PwCfg {PwId(\\\pwid=65542),Name(pw24748),PwType(0),PwOutLabel(204802),PwInLabel(204802),PwMplsType(2),PwAttachedId(NULL),TunnelId(\\\srtunnelid=2),InBoundIfIndex(NULL),OutBoundIfIndex(NULL)}"

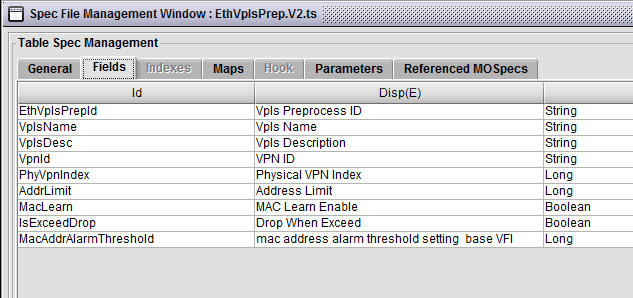
interface ethernet 1/2

set etherac ingress sp-vid 20 mapto pwid 65542

exit

## VPLS

VPLS业务表



详见样例

## L3VPN

### 基于PTN MPLS的L3VPN模型

可以参考以前L3VPN项目（kernel）里的模型

#### 界面参考











#### IPv4 L3VPN

支持Trunk口，

**PE1配置：**

interface ethernet 1/5

no switchport

//trunk

set aggregate-port 1 mode static

set aggregate-port 1 hash-rule dstmac

interface ethernet 1/1

bind aggregate-port 1

exit

interface ethernet 1/2

bind aggregate-port 1

exit

show interface trunk

//VRF

set srv6 vpncommon vpnid 1 servicetype l3vpn vpnname aaa vrfname bbb vrfid 1

**VpnCmmCfg**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Item |  |  |  |
| VpnId |  |  |  |
| VpnName |  |  |  |
| ServiceType | 1=L3VPN |  |  |
| VrfName |  |  |  |
| VrfId |  |  |  |
| Sip |  |  |  |
|  |  |  |  |
|  |  |  |  |

//port AC

interface trunk 1

subinterface id 1 type l3 encapsmode port

exit

subinterface trunk 1/1

set srv6 ipv4ac vpnid 1 ipaddress 2.2.2.10 mask 255.255.255.0

exit

**subinterface**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Item |  |  |  |
| IfIndex |  |  |  |
| IfName |  |  |  |
| IfAlias |  |  |  |
| EncapsMode |  |  |  |
| SpVlanId |  |  |  |
| CeVlanId |  |  |  |
|  |  |  |  |
|  |  |  |  |

//tunnel

mpls

lsp

set ingress-lsp lspid 1 servicetype l3vpn outgoing 1/5 outlabel 102501 nexthop 1.1.1.1 peer 1.1.1.1

set egress-lsp lspid 2 servicetype l3vpn incoming 1/5 inlabel 102502

set mplstunnel tunnel1\_2 ingress lspid 1 egress lspid 2

exit

**TunnelCfg**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Item |  |  |  |
| TunnelId |  |  |  |
| Name |  |  |  |
| IgrLspId |  |  |  |
| EgrLspId |  |  |  |
|  |  |  |  |
|  |  |  |  |

MplsXcCfg

|  |  |  |  |
| --- | --- | --- | --- |
| Field Item |  |  |  |
| LspId |  |  |  |
| Name |  |  |  |
| IgrIfType |  |  |  |
| IgrIfIndex |  |  |  |
| IgrClientIndex |  |  |  |
| IgrLabel |  |  |  |
| EgrIfType |  |  |  |
| EgrIfIndex |  |  |  |
| EgrClientIndex |  |  |  |
| EgrLabel |  |  |  |
| NextHopAddr |  |  |  |
| LspRole |  |  |  |
| AdminStatus |  |  |  |
| PrivateLabel |  |  |  |
| PhyTunnelIndex |  |  |  |
| PeerAddr |  |  |  |
| IsPerEn |  |  |  |
| IngressVlanId |  |  |  |
| EgressVlanId |  |  |  |
|  |  |  |  |
|  |  |  |  |

承载PW的Tunnel可以手工配置，测试方案8.2.4，

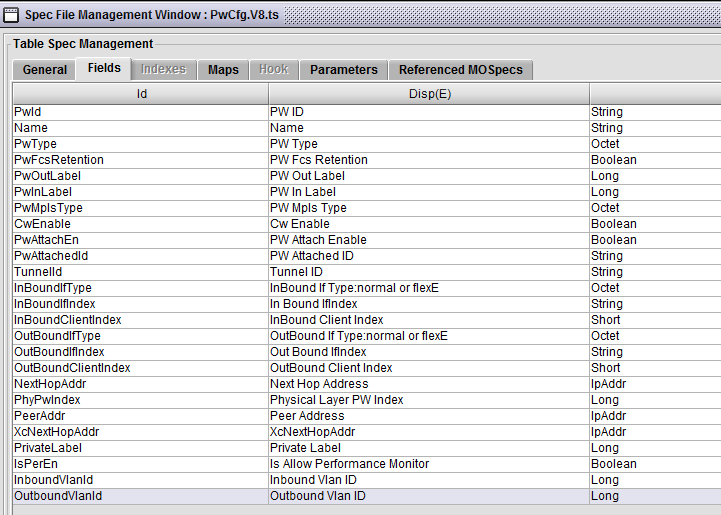
g. 在 NE2 和 NE7 上部署 L3VPN 业务 1，在 SR-TP 隧道 1 上承载；

pw

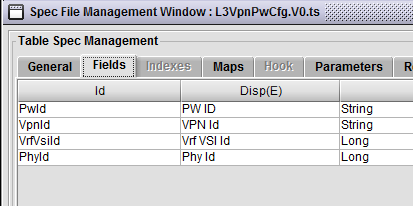
set pwid 65537 pwname pw1\_2 pwtype ethernet-tag peer 1.1.1.1 static outbound-label 204800 inbound-label 204801 mapto tunnel tunnel1\_2

exit

**PwCfg**



set l3vpnpw pwid 65537 vpnid 1

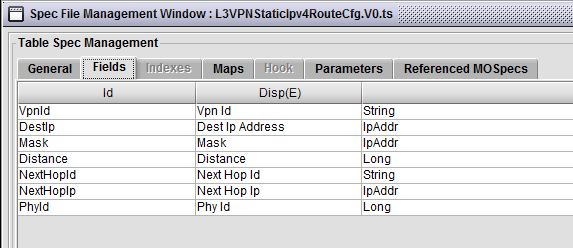


//route

set srv6 staticipv4route vpnid 1 destip 12.1.1.1 mask 255.255.255.0 distance 2 nexthopid \\\pwid=65537

set srv6 staticipv4route vpnid 1 destip 2.2.2.20 mask 255.255.255.0 distance 2 nexthopip 2.2.2.20

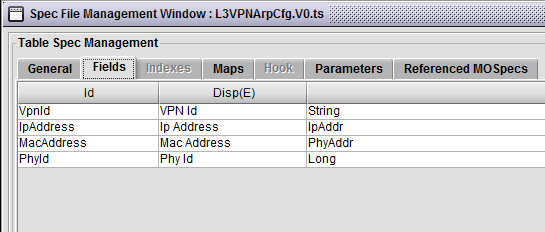
**L3VPNStaticIpv4RouteCfg**



//arp

set srv6 arp vpnid 1 ipaddress 2.2.2.20 mac 00:00:00:00:00:0a

**L3VPNArpCfg**



##### 私网路由引入和扩散功能测试

即测试方案8.2.3

测试步骤：

a） 按照图 39 搭建测试配置，核心设备 NE7 和 NE8 与域控制器相连；

b） 通过域控制器部署 NE1 到 NE8 网元的邻接标签和 SID；

c） 完成域控制器和设备之间 ISIS、SR、PCEP 等控制平面协议部署；

d） 部署 NE1、NE2、NE3 和 NE4 与 NE7 和 NE8 之间的 SR-TP 隧道 1-8；

e） 在 NE1、NE2、NE3、NE4 和 NE7、NE8 部署 L3VPN 业务 1，配置接入侧接口

和 IP；

f） 部署 NE1、 NE2、 NE3、 NE4 和 NE7、 NE8 的 peer，在 SR-TP 隧道 1-8 上承载；

g） 部署 NE1、NE2、NE3、NE4 互为 peer，在 SR-BE 隧道上承载；

h） 域控制器集中引入和扩散路由，并完成部署；

i） 修改 NE7 设备接入侧接口 IP，重新扩散路由并完成部署；

j） 修改 NE1 设备接入侧接口 IP，重新扩散路由并完成部署；

k） 删除 NE1 设备接入侧接口，重新扩散路由并完成部署；

预期结果

步骤 h） ：NE7 设备查询到与 NE1、NE2、NE3、NE4 接入侧接口直连路由相同的

静态路由，NE1 设备查询到与 NE2、NE3、NE4、NE7 和 NE8 接入侧接口直连路

由相同的静态路由；

步骤 i） ：NE1、NE2、NE3、NE4 设备上 NE7 接入侧接口原 IP 静态路由撤销，查

询到新 IP 的静态路由；

步骤 j） ： NE2、NE3、NE4、 NE7 和 NE8 设备上 NE1 接入侧接口原 IP 静态路由撤

销，查询到新 IP 的静态路由；

步骤 k） ： NE2、 NE3、 NE4、 NE7 和 NE8 设备上 NE1 接入侧接口 IP 静态路由撤销

##### 连接模型

###### 全连通模型



###### 分层模型



5.3.3. L3VPN 业务部署要求

SPN网络L3VPN应支持部署到边缘接入能力，即城域核心、汇聚、接入设备均具备按需部

署L3VPN的能力，应具备支持分层L3VPN（HoVPN）业务部署的能力。

分层L3VPN模型见图5-2所示，通过层次化部署增强VPN扩展能力。分层L3VPN

应符合以下功能要求：

a） UPE、SPE、NPE 设备间通过公网隧道互连，公网隧道可以是MPLS-TP、SR-TP 或者

SR-BE 隧道；

b） 分层部署模型要求网络内有且仅有一个核心L3VPN域，核心L3VPN 与多个接入L3VPN

域邻接，且接入L3VPN 域间不直接互通；

c） 接入L3VPN 域、核心L3VPN 域内私网路由扩散均遵循水平分割原则，实现业务就近

转发；

d） 同一对SPE 设备下挂的接入域设备IP 地址在一个或多个网段内分配，便于接入

域路由聚合；

e） UPE 发布明细路由给接入L3VPN 域的其他节点，包括SPE；

f） 允许用户手动聚合SPE 下挂接入域内明细路由，并将聚合路由发布给其它SPE

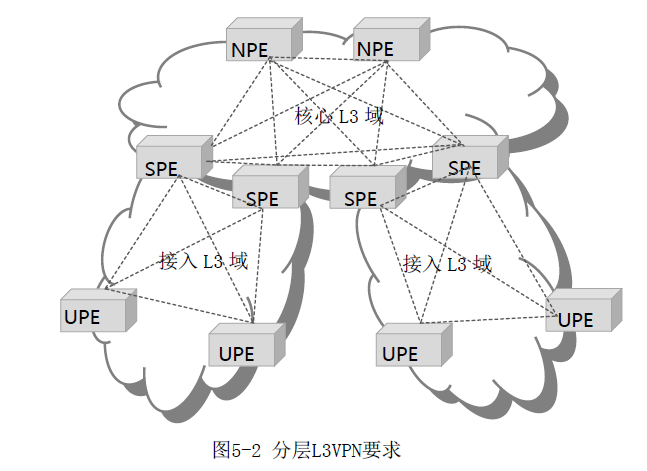
和NPE，其他SPE 和NPE 不再重发布；不向核心域发布UPE 明细路由；

g） NPE 将核心网侧明细路由发布给L3VPN 域内其他SPE 和NPE，其他SPE 和NPE 不再

重发布；

h） SPE 向接入L3VPN 域的UPE 发布默认认路由；

i） UPE 至SPE 或NPE 设备、SPE/NPE 设备间可通过VPN FRR 实现节点故障保护；



###### 用户化模型



见 私网路由引入和扩散功能测试

#### IPv6 L3VPN

/\*\*

trunk + port + ipv6

\*\*/

subinterface trunk 1/1

set srv6 ipv6ac vpnid 1 ipaddress 2222::10 mask 100

exit

set srv6 staticipv6route vpnid 1 destip 1002::1 mask 100 distance 2 nexthopid \\\pwid=65537

set srv6 staticipv6route vpnid 1 destip 2222::2 mask 100 distance 2 nexthopip 2222::2 outintf trunk1.1

set srv6 nd vpnid 1 ipaddress 2222::2 mac 00:00:00:00:00:0a outintf trunk1.1

#### Local Service

#### 端口封装模式

/\*\*

trunk + spVlan25 + ipv4

\*\*/

//port AC

interface trunk 1

subinterface id 2 type l3 encapsmode dot1q spvlanid 25

exit

subinterface trunk 1/2

set srv6 ipv4ac vpnid 1 ipaddress 3.3.3.10 mask 255.255.255.0

set srv6 staticipv4route vpnid 1 destip 13.1.1.1 mask 255.255.255.0 distance 2 nexthopid \\\pwid=65537

set srv6 staticipv4route vpnid 1 destip 3.3.3.20 mask 255.255.255.0 distance 2 nexthopip 3.3.3.20

set srv6 arp vpnid 1 ipaddress 3.3.3.20 mac 00:00:00:00:00:0a

/\*\*

trunk + spVlan25 + ipv6

\*\*/

subinterface trunk 1/2

set srv6 ipv6ac vpnid 1 ipaddress 3333::10 mask 100

exit

set srv6 staticipv6route vpnid 1 destip 1003::1 mask 100 distance 2 nexthopid \\\pwid=65537

set srv6 staticipv6route vpnid 1 destip 3333::2 mask 100 distance 2 nexthopip 3333::2 outintf trunk1.1

set srv6 nd vpnid 1 ipaddress 3333::2 mac 00:00:00:00:00:0a outintf trunk1.1

/\*\*

trunk + spVlan50 + ceVlan100 + ipv4

\*\*/

interface trunk 1

subinterface id 3 type l3 encapsmode qinq spvlanid 50 cevlanid 100

exit

subinterface trunk 1/3

set srv6 ipv4ac vpnid 1 ipaddress 4.4.4.10 mask 255.255.255.0

set srv6 staticipv4route vpnid 1 destip 14.1.1.1 mask 255.255.255.0 distance 2 nexthopid \\\pwid=65537

set srv6 staticipv4route vpnid 1 destip 4.4.4.20 mask 255.255.255.0 distance 2 nexthopip 4.4.4.20

set srv6 arp vpnid 1 ipaddress 4.4.4.20 mac 00:00:00:00:00:0a

/\*\*

trunk + spVlan50 + ceVlan100 + ipv6

\*\*/

subinterface trunk 1/3

set srv6 ipv6ac vpnid 1 ipaddress 4444::10 mask 100

set srv6 staticipv6route vpnid 1 destip 1004::1 mask 100 distance 2 nexthopid \\\pwid=65537

set srv6 staticipv6route vpnid 1 destip 4444::2 mask 100 distance 2 nexthopip 4444::2 outintf trunk1.1

set srv6 nd vpnid 1 ipaddress 4444::2 mac 00:00:00:00:00:0a outintf trunk1.1

#### 多VPN测试

/\*\*

VPN2

\*\*/

set srv6 vpncommon vpnid 2 servicetype l3vpn vpnname ccc vrfname ddd vrfid 2

//port AC

interface trunk 1

subinterface id 4 type l3 encapsmode dot1q spvlanid 111

exit

subinterface trunk 1/4

set srv6 ipv4ac vpnid 2 ipaddress 2.2.2.10 mask 255.255.255.0

exit

mpls

lsp

set ingress-lsp lspid 3 servicetype l3vpn outgoing 1/5 outlabel 102503 nexthop 1.1.1.1 peer 1.1.1.1

set egress-lsp lspid 4 servicetype l3vpn incoming 1/5 inlabel 102504

set mplstunnel tunnel3\_4 ingress lspid 3 egress lspid 4

exit

pw

set pwid 65538 pwname pw3\_4 pwtype ethernet-tag peer 1.1.1.1 static outbound-label 204802 inbound-label 204803 mapto tunnel tunnel3\_4

exit

exit

set l3vpnpw pwid 65538 vpnid 2

//route

set srv6 staticipv4route vpnid 2 destip 12.1.1.1 mask 255.255.255.0 distance 2 nexthopid \\\pwid=65538

set srv6 staticipv4route vpnid 2 destip 2.2.2.20 mask 255.255.255.0 distance 2 nexthopip 2.2.2.20

//arp

set srv6 arp vpnid 2 ipaddress 2.2.2.20 mac 00:00:00:00:00:0a

# OAM

## TroubleShoot

Ping

TraceRoute

## tilfa

## FlexE 保护

## Service 保护

# QoS

# 时钟同步

同步以太：以太物理层频率同步信号传送

CES/CEP业务时钟：业务端到端频率时钟同步

时间同步

# 数据同步

SID同步：PathSID，AdjSID，PrefixSID （在SID管理章节已由提到）

PW/LSP标签同步

业务同步：

业务发现：

# 互通测试 - 第三方网元

# 界面要求

【】界面要支持中文

# 网元容量规格

Tunnel数量

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product** | **SR Tunnel** | **SR-TP Tunnel** | **SR-BE Tunnel** | **Notes** |
| SPN803 and varieties | 2k | 2k | 2k | Shared |
| SPN804 and varieties | 2k | 2k | 2k | Shared |
| SPN806 | 8k | 8k | 8k | Shared |
| SPN808 | 16k | 16k | 16k | Shared |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product** | **Local Prefix-SID Mapping Entries** | **Non-local Prefix-SID Mapping Entries** | **Adj-SID Mapping Entries** | **Notes** |
| SPN803 and varieties | 2k | 2k | 1k | Shared.  Each column item includes static configured entries and dynamically learned entries. |
| SPN804 and varieties | 2k | 2k | 1k | Shared.  Each column item includes static configured entries and dynamically learned entries. |
| SPN806 | 8k | 8k | 1k | Shared.  Each column item includes static configured entries and dynamically learned entries. |
| SPN808 | 16k | 16k | 1k | Shared.  Each column item includes static configured entries and dynamically learned entries. |

# 网络规模

【】一个Controller最大需支持

节点数：5K （最小需支持PRD提出的1000节点）

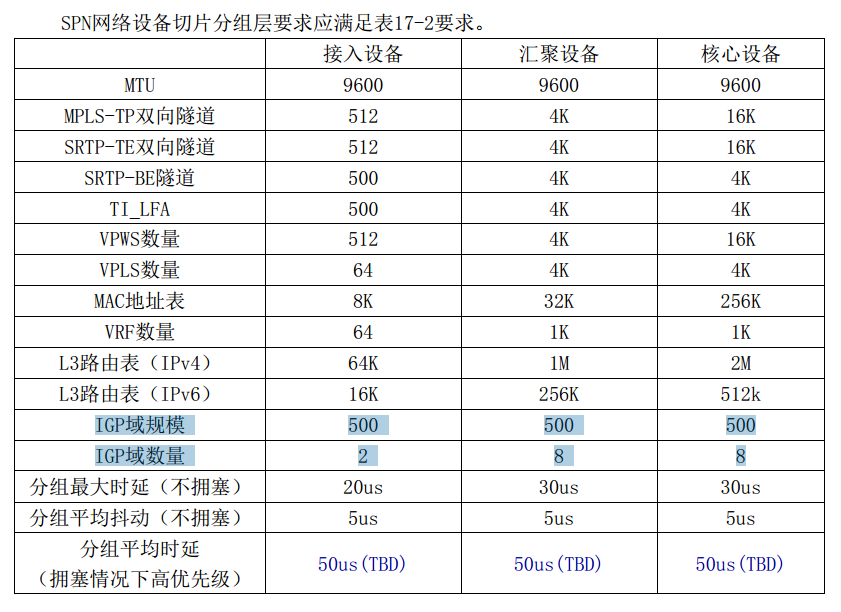
LSP/SRTunnel数：50K

PW数：100K

并发Web 客户端个数：20

PCE计算路径条目性能：2k/s

## 其他参考



Agreement between UTStarcom & Ericsson (These are original requirements from CMCC) –  2016初~ 2017/5  from DoBong

* D-Controller支持设备节点数量不少于1k，支持LSP数量不少于10k；
* D-Controller的路径计算能力大于500/s；
* D-Controller处理消息大于15k/s，并且消息的RTT时延低于100ms；

CMCC 《4.2-16-附件3-5 面向集中化运维的OMC能力提升要求-网络部16年12月更新》  2016  from Zhengkai

单套OMC系统是指目标实现方式下的全省一套OMC系统，系统容量要求如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 专业 | 等效网元 | 容量要求（个） | 系统负荷要求 |
| 无线网 | 无线小区 | ≥50000 | ≤70% |
| 核心网 | HLR | ≥1000 | ≤70% |
| 传输网 | PTN：等效155M带宽管理能力 | ≥200000 | ≤70% |
| OTN、小型化PTN：等效155M带宽管理能力 | ≥50000 | ≤70% |
| PON：端口数/线 | ≥4,000,000 | ≤70% |
| IP网 | 路由器 | ≥1000 | ≤70% |

应支持多客户端接入，操作终端接入的数量不少于16（不包含北向接口）；

CMCC  4.2-16-附件3-5 面向集中化运维的OMC能力提升要求-网络部16年12月更新  from Zhengkai

1. OMC系统应面向省级集中化运维的需求，单套应用软件系统具备30万等效网元的管理能力
2. 支持全省统一部署，统一拓扑、统一北向
3. 系统达到30万等效网元管理能力（含与周边系统对接等）要求时，资源占用率不能超过预分配资源的70%；
4. 总用户数≥2000个，同时在线用户数≥1000个，并发用户数≥500个，告警吞吐量≥10000条/秒

CMCC  《中国移动软件定义分组传送网（SPTN）总体技术要求》 2017  from Zhengkai

* S-Controller应支持直接控制至少64个下层控制器，D-controller应支持直接控制至少10k台传送设备。
* 业务数量：

A等级控制器LSP数量不少于1000k，PW数量不少于2000k；

B等级控制器LSP数量不少于200k，PW数量不少于400k；

C等级控制器LSP数量不少于50k，PW数量不少于100k；

D等级控制器LSP数量不少于10k，PW数量不少于20k；

* 路径计算能力：单位时间完成业务路径计算的数量

A等级控制器不少于50k/s；

B等级控制器不少于10k/s；

C等级控制器不少于2.5k/s；

D等级控制器不少于0.5k/s；

* 控制器每秒处理消息能力：控制器单位时间内与下层控制器或传送节点之间消息交互的能力，其中

A等级控制器不少于300k Message/s；

B等级控制器不少于100k Message/s；

C等级控制器不少于50k Message/s；

D等级控制器不少于15k Message/s。

CMCC  《4.2-10-附件1.3-中国移动PTN设备测试规范-16年12月更新》对节点数的要求，遵循《中国移动软件定义分组传送网（SPTN）总体技术要求》；但是测试网络拓扑很简单

CMCC 《小型化PTN管控一体化控制器功能测试规范》 2018/8  from HeQian

协议分析仪仿真客户设备，向厂商控制器发起Openflow会话数量50000个

CMCC 《小型化PTN管控一体化控制器功能技术规范》2018/8   from HeQian

控制器应具有以下性能指标：

1. 节点数量：

A等级控制器节点数量不少于100k；

B等级控制器节点数量不少于20k；

C等级控制器节点数量不少于5k；

D等级控制器节点数量不少于1k；

1. 业务数量：

A等级控制器LSP数量不少于1000k，PW数量不少于2000k；

B等级控制器LSP数量不少于200k，PW数量不少于400k；

C等级控制器LSP数量不少于50k，PW数量不少于100k；

D等级控制器LSP数量不少于10k，PW数量不少于20k；

1. 路径计算能力：单位时间完成业务路径计算的数量

A等级控制器不少于50k/s；

B等级控制器不少于10k/s；

C等级控制器不少于2.5k/s；

D等级控制器不少于0.5k/s；

1. 控制器每秒处理消息能力：控制器单位时间内与下层控制器或传送节点之间消息交互的能力，其中：

A等级控制器不少于300k Message/s；

B等级控制器不少于100k Message/s；

C等级控制器不少于50k Message/s；

D等级控制器不少于15k Message/s。

采用控制器集群指标至少满足B，单控制器系统满足指标C。

# 附录

## PTN标签

LSP：[102400, 204799]

PW: [204800, 307300]

OMCO PTN PW LSP标签分配原则： 一条单向路径上的标签一样

705 出入标签 节点级 唯一

## Admin状态和Operate状态

新创建业务

admin状态：Down

operate状态：Down

成功激活业务后

admin状态：Up

operate状态：Up

激活业务失败 （包括全部失败或部分失败，我们给Operate状态加了Partial表示部分失败。对这种状态下的业务，可以做去激活再激活，来达到最终成功激活目的）

admin状态：Up

operate状态：Down/Partial

成功去激活业务后

admin状态：Down

operate状态：Down

去激活业务失败（对这种状态下的业务，可以重复做去激活，来达到最终成功去激活目的）

admin状态：Down

operate状态：Up/Partial

## 10/30 测试基本实现要求

FlexE 测试

|  |  |  |  |
| --- | --- | --- | --- |
| **功能** | **后台开发进度** | **前端开发进度** | **备注** |
| FlexE Group创建 | 完成 | 预计25日完成 | Member格式问题 |
| FlexE Group激活 | 完成 | 预计25日完成 | 激活问题 |
| FlexE Group去激活 | 完成 | 预计25日完成 |  |
| FlexE Group删除 | 完成 | 预计25日完成 |  |
| FlexE Client创建 | 完成 | 预计25日完成 |  |
| FlexE Client激活 | 完成 | 预计25日完成 | 激活问题 |
| FlexE Client去激活 | 完成 | 预计25日完成 |  |
| FlexE Client删除 | 完成 | 预计25日完成 |  |
| FlexE Channel创建 | 预计26日完成 | 预计28日完成 | 包括FlexE Channel拓扑展示，PCE模块（为测试可以先用模拟数据代替） |
| FlexE Channel激活 | 预计26日完成 | 预计28日完成 |  |
| FlexE Channel去激活 | 预计26日完成 | 预计28日完成 |  |
| FlexE Channel删除 | 预计26日完成 | 预计28日完成 |  |
| MPLS-TP Tunnel创建 | 预计29日完成 | 预计30日完成 | 包括MPLS LSP Tunnel拓扑，PCE模块（为测试可以先用模拟数据代替） |
| MPLS-TP Tunnel激活 | 预计29日完成 | 预计30日完成 | 基于FlexE Client/Channel |
| MPLS-TP Tunnel去激活 | 预计29日完成 | 预计30日完成 |  |
| MPLS-TP Tunnel删除 | 预计29日完成 | 预计30日完成 |  |
| VPWS创建 | 29日完成 | 预计30日完成 | 基于MPLS-TP Tunnel |
| VPWS激活 | 预计29日完成 | 预计30日完成 |  |
| VPWS去激活 | 预计29日完成 | 预计30日完成 |  |
| VPWS删除 | 预计29日完成 | 预计30日完成 |  |
|  |  |  |  |
|  |  |  |  |

SR-TP 测试

|  |  |  |  |
| --- | --- | --- | --- |
| **功能** | **后台开发进度** | **前端开发进度** | **备注** |
| Prefix SID创建 | 25日完成 | 预计26日完成 | 包括SID拓扑模型 |
| Prefix SID激活 | 25日完成 | 预计26日完成 | 激活问题 |
| Prefix SID去激活 | 25日完成 | 预计26日完成 |  |
| Prefix SID删除 | 25日完成 | 预计26日完成 |  |
| Adj SID创建 | 25日完成 | 预计26日完成 | 包括SID拓扑模型 |
| Adj SID激活 | 25日完成 | 预计26日完成 | 激活问题 |
| Adj SID去激活 | 25日完成 | 预计26日完成 |  |
| Adj SID删除 | 25日完成 | 预计26日完成 |  |
| SR Tunnel创建 | 完成 | 预计27日完成 | 包括SR Tunnel拓扑，PCE模块（为测试可以先用模拟数据代替） |
| SR Tunnel激活 | 完成 | 预计27日完成 |  |
| SR Tunnel去激活 | 完成 | 预计27日完成 |  |
| SR Tunnel删除 | 完成 | 预计27日完成 |  |
| VPWS创建 | 25日完成 | 预计28日完成 | 基于SR Tunnel |
| VPWS激活 | 预计27日完成 | 预计28日完成 |  |
| VPWS去激活 | 预计27日完成 | 预计28日完成 |  |
| VPWS删除 | 预计27日完成 | 预计28日完成 |  |
|  |  |  |  |
|  |  |  |  |

FlexE Case1: 两个节点间建立基于FlexE承载的VPWS

FlexE Physical Topology Overview

(AC eth1.1)NE1(eth1.39)--------(eth1.40)NE2(AC eth1.1)

步骤：

1.手工建好NE1和NE2的物理连接

两端的eth1.39端口需是三层口

2.在物理连接上建立FlexE Group

3.在FlexE Group建立FlexE Client

4.创建NE1和NE2间的MPLS-TP LSP Tunnel

5.创建VPWS业务 (包含创建PW)

注意：

要使能FlexE端口，配置文件/tffs/sys/netring.ini，最末段如下：[Customer]

CustomerID=4

[ZebosXP]

NsmThread=1

RibThread=1

IsisThread=1

BgpThread=1

LacpThread=1

[FEATURE]

FEATURE\_MPLS=1

FEATURE\_MPLS\_PERPORT=0

FEATURE\_SRTP=1

FEATURE\_FLEXE=1

FEATURE\_SRV6=0

FEATURE\_L3VPN=1

## WangFei Review后建议

SID校验和同步

1. 新网元加入，对SRGB，SRLB的自动同步

2. 如何和网元上已有LocalAdjSID，Tunnel/PathSID的同步

3. 支持手工选路

## 时间点

11月底要进ST

12月15要北京Trial

## 10/26 Demo

**SRTP Demo**



**FlexE Demo**