

YD

中华人民共和国通信行业标准

YD/T 1084—2000

基于 ATM 的网络接口 (PNNI) 信令规范

PNNI Signalling Specification Based on ATM Network Interface

2000-11-28 发布

2001-05-01 实施

中华人民共和国信息产业部 发布

目 次

前言	III
1 范围	1
2 引用标准	1
3 缩略语和定义	2
3.1 缩略语	2
3.2 定义	3
4 PNNI 选路和信令的一般描述	6
4.1 PNNI 的选路	6
4.2 PNNI 信令描述	22
5 PNNI 路由规范	26
5.1 通用的操作程序	27
5.2 寻址	27
5.3 标识符和表示语	28
5.4 逻辑链路	30
5.5 PNNI 路由控制通路	30
5.6 Hello 协议	37
5.7 数据库同步	52
5.8 拓扑的描述和分配	60
5.9 通告和归纳可达地址	73
5.10 层次	74
5.11 对等组的分区	87
5.12 当拓扑数据库处于过负荷时节点的操作	88
5.13 通道选择	89
5.14 分组包格式	92
6 PNNI 的信令规范	111
6.1 协议模型	111
6.2 呼叫/连接控制概述	112
6.3 消息功能定义和内容	112
6.4 消息的一般格式和信息单元编码	127

6.5 ATM 点一点呼叫/连接控制过程	139
6.6 点一点的呼叫/连接控制过程	151
附录 A(标准的附录) 指定转接序列	153
附录 B(标准的附录) 折回过程	158
附录 C(标准的附录) 软 PVC 程序	166
附录 D(标准的附录) 结构化常量	169
附录 E(标准的附录) 结构变量	170
附录 F(标准的附录) PNNI 层次配置	172
附录 G(标准的附录) PNNI 最小子集	173
附录 H(标准的附录) PNNI 管理信息库	175

前 言

本标准参考了 ATM 论坛 1996 年 PNNI 1.0 技术规范、UNI3.1 规范和 UNI4.0 规范以及 ITU Q.2931 建议,并根据我国的实际情况,对我国 ATM 网络中 PNNI 协议需要使用的地址方式、路由选择、信令消息格式和程序等,特别是对 PNNI 协议中应用的动态路由选择程序、折回过程以及 DTL 的处理等程序进行了详细的规定。

本标准的附录 A、B、C、D、E、F、G、H 都是标准的附录。

本标准由信息产业部电信研究院提出并归口。

本标准起草单位:信息产业部电信传输研究所

本标准的主要起草人:续合元 吕 军 王立言 盛 蕾 吴立贞 魏晨光

中华人民共和国通信行业标准

基于 ATM 的网络接口(PNNI)信令规范

**PNNI Interface Signalling Specification Based
on ATM Network Interface**

YD/T 1084—2000

1 范围

本标准规定了在我国多媒体通信网中 ATM 交换局之间采用的 PNNI 协议。PNNI 包括两类协议：

- 第一类是规定在交换局和交换局之间分布式的拓扑信息；
- 第二种协议是规定信令消息和程序。

PNNI.0 版本具有下列特点：

- 支持 UNI3.1 全部的和 UNI4.0 部分的能力；
- 可应用于很大的网络；
- 支持等级选路；
- 支持 QoS；
- 支持路由的量和属性；
- 支持起源选路的连接建立；
- 支持资源可用性变化后的动态选路；
- 对等组间和对等组内的选路协议；
- 与外部路由域的互操作性；
- 支持物理链路和 VPC 的隧道；
- 支持软的 PVPC/PVCC；
- 支持任播(anycast)能力。

PNNI 适用于网络节点接口间或网络间的信令协议标准，其协议主要用在网内的交换局间，也可应用于不同的网络之间。

2 引用标准

下列标准包含的条文，通过在本标准中引用而构成本标准的条文。在标准出版时，所示版本均为有效。所有标准都会被修订，使用本标准的各方应探讨使用下列标准最新版本的可能性。

YDN 083—1998	B-ISDN 的 DSS2 信令方式技术规范—适配层
YDN 053.4—1997	B-ISDN ATM 适配层 (AAL) 类型 5 技术规范
ATM Forum (1996)	UNI4.0 用户—网络信令接口
ATM Forum (1996)	ILMI4.0 暂时的本地管理接口
ITU-T I.150 (1996)	B-ISDN 异步转移模式的功能特性
ITU-T I.361 (1996)	B-ISDN ATM 层
ITU-T Q2931(1995)	B-ISDN DSS2 用户网络接口 (UNI) 基本呼叫/连接控制的第 3 层规范
ITU-T Q2971 (1996)	B-ISDN DSS2 UNI 点到多点呼叫/连接控制的第 3 层规范
IEEE802—1990	本地和大城市网络的 IEEE 标准：概况和配置

3 缩略语和定义

3.1 缩略语

AAL	ATM 适配层
ABR	可用比特率
AFI	授权和格式标识符
ATM	异步转移模式
AVCR	可用的信元率
AW	管理加权
BGP	边界网关协议
B-ICI	B-ISDN 承载者之间的接口
CAC	连接接纳控制
CBR	恒定比特率
CDV	信元时延变化
CLP	信元丢失优先级
CLR	信元丢失率
CLR0	CLP=0 业务量的信元丢失率指标
CRM	信元速率边际
CTD	信元转移时延
DSP	域特定部分
DTL	指定的转接一览表
ES	端系统
ESI	端系统标识符
FSM	有限状态机
GCAC	通用连接接纳控制
ICR	初始信元速率
IDI	初始域标识符
IDP	初始域部分
IDRP	域间选路协议
ILMI	过渡期本地管理接口
IE	信息单元
ID	标识符
IG	信息组
LGN	逻辑组节点
LSB	最低有效位
maxCR	最大的信元率
maxCTD	最大的转接时延
MBS	最大的突发尺寸
MCR	最小的信元率
MIB	管理信息库
MSB	最高有效位
NNI	网络至网络接口
NSAP	网络业务接入点
OSPF	优选开放的最短通道

PCR	峰值信元率
PG	对等组
PGL	对等组领导
PGLE	对等组领导选择
PTSE	PNNI 拓扑状态单元
PTSP	PNNI 拓扑状态分组包
PNNI	专用的网络至网络接口
PVC	永久虚电路
PVCC	永久虚通路连接
PVPC	永久虚通道连接
QoS	服务质量
RAIG	资源可用性信息组
RCC	路由控制通路
RDF	速率减少系数
RIF	速率增加系数
SAAL	信令 ATM 适配层
SCR	持续的信元率
SSCOP	业务指定面向连接协议
SSCS	业务指定会聚子层
SVC	交换虚电路
SVCC	交换虚通路连接
TBE	转接缓存器显露
TLV	未规定的比特率
UBR	可变比特率
ULIA	上行链路信息属性
UNI	用户网络接口
VBR	可变比特率
VCC	虚通路连接
VCI	虚通路标识符
VF	变化系数
VP	虚通道
VPC	虚通道连接
VPI	虚通道标识符

3.2 定义

地址前缀：0 个或多个比特串，最大可达 152 比特。它是一个或多个 ATM 地址的开始部分。

相邻性：两个通信的邻近同级节点之间的关系。

汇集令牌：外部链路的边界节点分配给外部链路的号码。这个号码与所有的上行链路和外部链路推导出的上行链路有关。在父和所有较高等级的对等组中，具有同样汇集令牌的所有上行链路进行汇集。

替换选路：在连接建立失败后，使用新通道进行连接建立。

祖先节点：对某给定节点有直接的本源关系的逻辑组节点。

ATM 任播能力：到 ATM 组中一个 ATM 端系统建立点到点连接的能力。

边界节点：位于某指定对等组中且至少有一条链路穿过该对等组边界的逻辑节点。

旁路：旁路是复杂节点表示中两个端口之间的连通性。

子节点：在等级结构中的较低级节点，它包含在逻辑组节点所表示的对等组中。它可以是一个逻辑

组节点，或者是一个物理节点。

子对等组：对等组的子对等组是包含对等组内逻辑组节点的子节点的任何一个对等组；逻辑组节点的子对等组是包含逻辑组节点的子节点的对等组。

公共对等组：表示一组节点的最低等级的对等组。对等组可以直接或通过其祖先来表示节点。

复杂节点表示：提供与逻辑节点有关的详细状态信息的节点状态参数集合。

连接范围：选路层次等级，在这个等级中限制对给定的组地址的连接请求。

折回：在连接建立过程中遇到失败时进行部分释放，它允许 PNNI 完成迂回选路。

缺省节点表示：每个节点状态参数的单值，它给出了到逻辑节点和核的任何入或出之间的假定值。

指定转接序列：通过一个 PNNI 对等组通道的节点和链路 ID 的序列。

Dijkstra's 算法：用来计算路由的一种算法。

DTL 始发节点：在整个 PNNI 选路域中对某给定连接建立初始 DTL 堆栈的第一个最低级节点。

DTL 终接节点：在整个 PNNI 选路域中处理连接（即连接的 DTL）的最后一个最低级节点。

端系统：连接到终接节点的系统。

入口边界节点：在外部链路上接收呼叫的节点。它是对等组中的第一个接收到该呼叫的节点。

例外：在 PNNI 复杂节点表示中的连接通告，它表示除了缺省节点表示以外的某些表示。

出口边界节点：将使呼叫在外部链路上进行的节点。它是对等组中处理这个呼叫的最后一个节点。

域外：说明某一项（例如：链路、节点或可达地址）超出了 PNNI 路由域。

域外链路：穿过 PNNI 路由域边界的链路。PNNI 协议不在域外链路上运行。

域外可达地址：通过 PNNI 路由域可能到达的地址，但该地址不在 PNNI 路由域中。

域外路由：穿过域外链路的路由。

外来地址：与节点的任何归纳地址都不匹配的地址或地址字冠。

Hello 分组包：在相邻逻辑节点之间进行交换的一种 PNNI 选路分组包。

完整源路由：表示穿过 PNNI 路由域的 DTL 堆栈，该堆栈中包括对源、目的地都可达的当前的层次等级和最低可见级。

逐跳路由：由每个交换机根据自身的路由情况决定呼叫进展的下一路由，PNNI 不使用逐跳路由。

水平链路：在属于同一个对等组的两个逻辑节点之间的链路。

导出上行链路：上行链路“A”的产生是由于上行链路“B”的存在，上行链路“B”位于由生成上行链路“A”的节点表示的子对等组中。“A”和“B”两者共享同一个上行节点，它在 PNNI 分级中高于见到上行链路“A”的对等组。

内部链路：与上文水平链路同义。

实例 ID：唯一标识 MIB 实例的对象属性的子集。

域内：说明某项（例如：链路、节点或可达的地址）位于 PNNI 选路域的内部。

内部可达的地址：直接接到逻辑节点通告地址的目的地节点的地址。

领导权优先级：逻辑节点使用领导优先级选定其对等组领导。

等级：节点在 PNNI 等级结构中所处的位置。

链路：与下文逻辑链路同义。

链路汇集令牌：见汇集令牌。

链路属性：链路状态参数，它用来确定某给定链路是否可用来承载某给定连接。

链路限制：为特定连接通道选择时链路使用的限制。

链路量度：链路参数，它要求将沿着给定通道的所有链路的参数值综合起来以便确定该通道是否可用来承载给定连接。

链路状态参数：链路特性的信息。

逻辑组节点：较低级对等组作为一个点的抽象表示，可用于工作在 PNNI 选路等级结构中的一个级别。

逻辑链路：在两个逻辑节点之间连接的抽象表示。这包括各种物理链路，各种虚通道连接，并行的物理链路和/或虚通道连接。

逻辑节点：最低级节点或逻辑组节点。

逻辑节点 ID：明确标识路由域中的逻辑节点的比特串。

最低级节点：在 PNNI 选路等级结构中的叶；抽象表示 PNNI 选路协议的一个事例。最低级节点是在交换系统中经过配置得到的，它们不是动态建立的。

成员范围：选路等级结构的级，在该级中限制通告某给定地址。

本土地址：匹配一个给定节点的归纳地址的一个地址或地址前缀。

邻近节点：经由一个逻辑链路直接连到一个特定节点的节点。

节点属性：节点状态参数，用以确定一给定的节点是否能承载一个给定的连接。

节点限制：对于给定连接，通道在选择节点时的一种限制。

节点量度：节点参数，它要求将沿着给定通道的所有节点的参数值综合起来以便确定该通道是否可用来承载给定连接。

节点状态参数：节点特性的信息。

节点：与逻辑节点同义。

非一支节节点：目前不能够支持点到多点呼叫附加分支点的一种节点。

核：在 PNNI 复杂节点表示中一个逻辑节点的内部参考点。

空：一个全 0 的值。

分离节点 (Outlier)：如果将该节点从对等组中分离出去，将明显改善对等组拓扑汇集的准确性和简单性。

外部链路：到最低级外部节点的一条链路。

外部节点：参与 PNNI 选路但又不属于特定对等组的一个节点。

父节点：在上一级表示包含特定节点的对等组的一个逻辑组节点。

父对等组：一个对等组的父对等组表示包含该对等组的一个逻辑组节点的对等组。一个节点的父对等组是包含那个节点的一个父节点。

通道限制：对沿着一个特定连接的通道中一个拓扑矩阵的极限值。

通道范围：通道使用的 PNNI 分级的最高级。

对等组：为创建选路层次划分的一组逻辑节点。PTSE 在组内的全部成员中是可交换的。

对等组标识符：用来唯一标识一个对等组使用的一个比特串。

对等组领导者：为把整个对等组单表示成一个单独的节点，对等组内的一个节点要完成采集、汇集、组建数据等额外工作，这种表示可用在父节点。

对等组级：一个特定对等组的对等组标识符的有效的比特数。

对等节点：它是作为给定节点的对等组内的一个节点。

物理链路：接在两个交换系统间的一条实际链路。

PNNI 协议实体：在交换系统中执行 PNNI 协议和提供路由业务的软件实体。

PNNI 路由控制通路：为交换 PNNI 路由协议消息使用的 VCC。

PNNI 选路层次：用于 PNNI 选路的对等组层次。

PNNI 拓扑状态分组包：用在对等组内逻辑节点间泛播 PTSE 的 PNNI 选路分组包类型。

端口：一条链路到一个节点的连接点。

端口标识符：由一个逻辑节点分配的标识符，表示一条链路到一个节点的连接点。

ATM 地址：用来标识一个 ATM 终接点的 20 个八位位组的地址。

可达地址前缀：在 20 个八位位组 ATM 地址中的一个前缀，表示以这个字冠开始的所有的地址都是可达的。

受限的转接节点：在受限的环境中仅能够由转接呼叫采用的一种节点。如果它用于始发或终端呼

叫时，则可以不受这种限制。

路由计算：对一个拓扑数据库利用数学算式计算路由的进程。

路由限制：指拓扑限制或通道限制的通用术语。

范围：范围规定通告一个地址的等级。等级是在 PNNI 路由分级中的对等组的等级。

轮幅 (Spoke)：在复杂的节点表始中，这表示在核心和一个特定端口之间的连通性。

归纳地址：一个地址的前缀，表示一个节点归纳的可达性信息。

交换系统：一组或多组的物理设备以作为单个的 PNNI 的网络管理实体。当作为一个 PGL、一个或多个 LGN 时，一个交换系统包含一个或多个最低级的节点。

拓扑状态参数：是链路参数或节点参数的通用术语。

拓扑汇集：归纳和压缩在向上通告的分级拓扑信息的一种过程。

拓扑属性：是链路属性或是节点属性的通用术语。

拓扑限制：是链路限制或节点受限的通用术语。

拓扑数据库：该数据库描述由一个节点见到的完整的 PNNI 路由域。

拓扑量度：是链路量度或节点量度的通用术语。

上行链路：表示边界节点到上行节点的连通性。

上行节点：该节点表示在公共对等组中的一个边界节点的外部邻近点。上行节点必须是一个相邻对等的边界节点的祖先。

4 PNNI 选路和信令的一般描述

本节对 PNNI 选路和信令进行一般性的描述，有关选路协议的详细定义在第 5 章给出，有关信令的详细定义由第 6 章给出。

4.1 PNNI 的选路

PNNI 选路协议的功能包括：

- 查明相邻节点和链路状态；
- 拓扑数据库的同步；
- PTSE 的泛播；
- PGL 的选举；
- 拓扑状态信息的归纳；
- 选路层次的构造。

4.1.1 物理网络

PNNI 选路协议用于含最低级节点的网络。图 1 表示了一个由 26 个互连的最低级网络节点构成的网络。

数据从最低级节点传至其他最低级节点并传至端系统。端系统是指连接的起源点和终接点，端系统并未在图 1 中标出。为了确定路由，端系统由 ATM 端系统地址的最高 19 个八位位组来标识。最后一个八位位组 (SEL) 不用于 PNNI 路由选择，但可被端系统使用。

图 1 中的每条链路均表示两交换系统间的物理链路。在一个交换系统中，端口是链路到最低级节点的连接点。物理链路是双向的（业务可在两个方向上传输），但是每个方向上的物理链路特性可能不同，例如容量、业务负荷不同。因此每条物理链路由两组参数来标识，每个方向一组。这样一组参数包括传输端口标识符，加上包含那个端口的最低级节点的节点 ID。PNNI 端口 ID（如 5.3.4 节中所述）可能与设备特有的端口标识符不同。

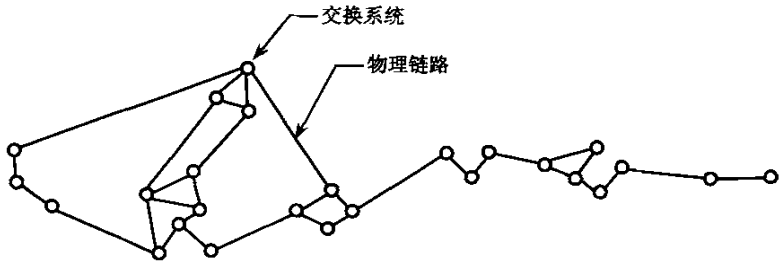


图 1 26个ATM交换系统、33条双向链路的ATM网络

4.1.2 最低等级

如果 PNNI 协议只支持图 1 表示的平面网络，那么每个最低级节点都将不得不保留网络的整个拓扑结构，包括网络中的每条物理链路信息和网络中每个节点的可到达信息。这种情况对小网络可行，但对大网络则会产生大量开销。PNNI 选路层次结构设计的要求是在提供高效选路的同时减小这种开销。

4.1.2.1 对等组和逻辑节点

PNNI 选路层次从最低级开始，最低级节点组成对等组。在最低等级的情况下，“逻辑节点”就是最低级节点。为简便起见，逻辑节点常常表示成“节点”。对等组是逻辑节点的集合，每个节点都与组中其他节点交换信息，因此所有节点都在同一组内。逻辑节点是唯一的，并由“逻辑节点 ID”明确地标识。

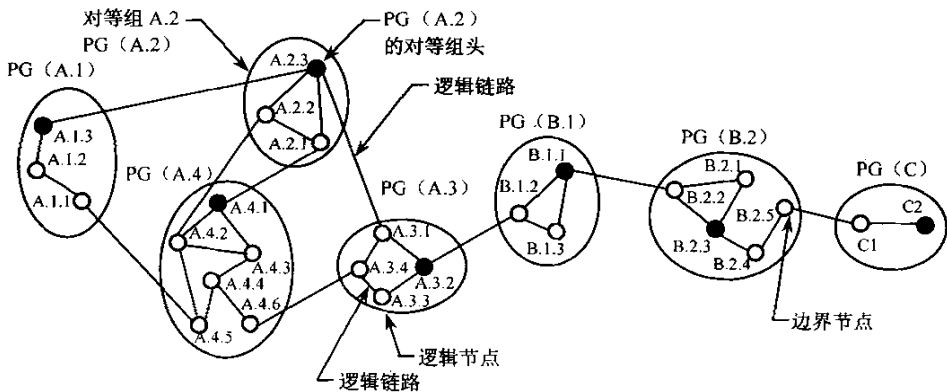


图 2 最低级节点的局部PNNI层次配置

图 2 中，网络被分成 7 个对等组：A.1、A.2、A.3、A.4、B.1、B.2 和 C。为避免节点和对等组间的混淆，对等组在图 2 中表示成 PG ()。例如，PG (A.3) 表示对等组 A.3。节点和对等组的编号(如 A.3.2 和 A.3)用于在读文件时进行区分。这种抽象表示反映了描述的层次结构，如由 A.3.2 表示的节点位于 PG (A.3) 中，即在对等组 A.3 中。

对等组用对等组标识符来标识，对等组标识符在配置期间指定。相邻节点用“Hello 分组包”来交换对等组 ID (见 5.6 节关于 Hello 协议的详细信息)。如果它们的对等组标识符相同则它们属于同一对等组，否则不属于同一对等组。

“边界节点”至少有一条链路经过边界线，因此与对等组标识符不同的相邻节点是它们各自对等组的边界节点。当出现错误或故障时，对等组可以分区，形成具有相同 PG 标识的多个 PG。

对等组 ID 被定义为 ATM 端系统地址的最高 13 个有效八位位组的前缀。因此对等组 ID 可缺省为这个 PG 的一个或多个节点地址的前缀。

4.1.2.2 逻辑链路及其初始化

逻辑节点通过“逻辑链路”实现连接。在最低级节点间，逻辑链路是物理链路（如图 1 所示）或者两最低级节点间的 VPC。同一对等组中最低级节点间的链路不被汇集在一起。例如，如果在图 1 中两条物理链路连接的是同一对节点，那么它们在图 2 中将由两条分开的逻辑链路表示。PG 内部的逻辑链路是“水平链路”而连接两个 PG 的是“外部链路”。

当一条逻辑链路开始工作，所连接的节点通过一条众所周知的用作“PNNI 选路控制通路”（RCC）的 VCC 启动信息交换。在这条链路上的节点周期性地发送 Hello 分组包以指定 ATM 端系统的地址、节点 ID 和链路的端口 ID。如前所述，PNNI 的 Hello 协议也支持对等组 ID 的交换，使相邻节点能判断是否属于同一对等组。

只要链路工作，Hello 协议就会运行，因此当其他机制失效时，Hello 协议就可以作为链路故障检测器。

4.1.2.3 PNNI 信息交换

每个节点与它的邻节点交换 Hello 分组包，确定它的本地状态信息。这种状态信息包括相邻节点的身份和对等组成员以及与相邻节点连接的链路本身的状态。每个节点将自身的状态信息放入“PNNI 拓扑状态单元”（PTSE）中，而 PTSE 在对等组中被可靠地泛播。

PTSE 是最小的 PNNI 选路信息集合，它在 PG 内的所有节点间作为一个单元泛播。节点的拓扑数据库由所有收到的 PTSE 组成，它反映了 PNNI 选路域节点当前状态。特别情况下，拓扑数据库提供了用于计算从给定节点到选路域内任何可达或可通过地址的路由的全部信息。

1) 节点信息

每个节点产生一个 PTSE 用来描述自己的身份和能力，选举 PG 领导（见 4.1.2.4 节）的信息以及建立 PNNI 等级的信息，这些就是节点信息。

2) 拓扑状态信息

PTSE 包含“拓扑状态参数”（即“链路状态参数”——它描述了逻辑链路的特性和“节点状态参数”——它描述了节点的特性）。

拓扑状态参数按属性（attribute）和量度（metric）参数分类。属性在路由决策时单独考虑。例如，出于某种安全考虑的“节点属性”会引起给定通道被拒绝。在另一方面量度参数是沿通道累积影响的参数。例如延迟在给定通道上不断增长，参见 5.8 节。

一定的拓扑状态信息，尤其是与带宽有关的信息，是相当动态化的。另一方面，其他拓扑状态信息如管理加权（administrative weight）信息，是相对静态的。PNNI 拓扑分布的动/静态拓扑状态参数在传输机制上没有区别。

3) 可达性信息

可达性信息由该呼叫要到达的目的地地址和地址前缀组成。这种信息由节点在 PNNI 选路域中用 PTSE 通告。

根据信息源逻辑地分为内部和外部可达性信息。PNNI 选路协议不是用于 ATM 网络选路的唯一协议。域外的可达性是从 PNNI 选路域外的其他协议交换获得的。内部的可达性表示 PNNI 选路域内的可达性。两者主要差别处在于不把域外的可达性信息通告到其他选路协议或选路域（以免跨越选路域形成环路）。人工配置能用于创建内部或外部的可达性信息，它们对被通告的其他选路协议或选路域具有相应的影响。

域外可达地址可用于向其他独立的 PNNI 选路域通告其连通性。

4) 初始的拓扑数据库交换

逻辑链路两端的相邻节点通过交换 Hello 分组包进行初始化时，断定它们位于同一对等组后，就开始同步它们的拓扑数据库。数据库同步（即相邻节点间的信息交换），使两个节点有相同的拓扑数据库。拓扑数据库包括逻辑节点所在对等组的详细拓扑信息以及表示 PNNI 选路域的其余部分的更详细的拓扑信息。较高级信息流入对等组的方式将在后续的 4.1.3~4.1.5 节里描述。

在拓扑数据库同步期间，节点首先请求交换 PTSE 信头信息，即它们通告在各自的数据库中 PTSE 的存在。当节点收到更新的或原来所没有的 PTSE 信头信息时，它就请求被通告 (advertise) PTSE 并用后来收到的 PTSE 更新它的拓扑数据库。如果一个最近初始化了的节点和对等组连接，那么随之发生的数据库同步会变成单向的拓扑数据库拷贝。

只有在相邻节点的数据库同步完成后，链路才通过 PTSE 发送而通告。链路状态参数以这种方式被分配给包含这条链路的对等组里的所有拓扑数据库。

5) 泛播

泛播是在整个对等组内可靠地逐个节点的传播 PTSE。它可确保一个对等组里的每个节点都有同样的拓扑数据库 (见 5.8.3 节)。泛播是 PNNI 的通告机制。

泛播基本过程如下。PTSE 被封装在“PNNI 拓扑状态分组包”(PTSP)内传送。当收到 PTSP 后，就对其组成部分进行检测。通过把 PTSE 信头的信息封装在“证实分组包”中，并把它发送到发送信息的邻近节点，来证实每个 PTSE。如果 PTSE 是新的或比节点当前的拷贝更新，则将其存入拓扑数据库，并泛播到除发出此 PTSE 的节点外的其他相邻节点。一个 PTSE 被周期性传送给邻近节点，直到被证实为止。

泛播是一个动态的活动，即每个节点发出 PTSP 都包含已更新信息的 PTSE。在经过一段时间后，拓扑数据库里的 PTSE 若未被新的 PTSE 刷新，则这些 PTSE 由于老化而被删除。只有最初产生某个特殊 PTSE 的节点可以重发 PTSE。重发 PTSE 是周期性的并且是基于事件驱动的。

4.1.2.4 对等组领导

4.1.3 节描述了如何由一个被称为“逻辑组节点”的单一节点将一个对等组体现在上一个等级层里。完成这一任务的功能由一个被称为“对等组领导”(PGL)的节点来执行，它是代表这个对等组的一个节点。每个对等组内最多有一个激活的对等组领导，更确切地说在一个分区的对等组里，每个分区最多有一个对等组领导。

除了为保持 PNNI 层次结构而收集与转发信息外，PGL 在对等组内并没有任何特殊作用。对于所有其他功能(比如建立连接)其作用像其他节点一样。

“对等组领导选择”过程决定那一个节点具有领导节点的功能。选择的标准是节点的“领导优先级”(leadership priority)。对等组内具有最高领导优先级的节点就成为该对等组的领导节点。注意：这个选举过程是一个持续的运行协议。当一激活的节点比现有对等组领导具有更高的优先级时，选择过程就把对等组领导转移到新激活的节点上。当 PGL 被删除或故障时，次最高优先级的节点就成为 PGL。最终，如果在多个具有相同优先级的节点间产生竞争的情况下，那么具有最高节点 ID 的节点将成为 PGL。当一个节点被选为 PGL 后，它的领导优先级将提高到可以确保稳定性的级别。

对等组的内部操作并不要求具有对等组领导。即使没有对等组领导，对等组内的全连接(full connectivity)也能获得。一个 PNNI 选路域以单个对等组配置时，即使没有对等组领导也能获得全连接。

一个对等组的简化形式是只含有一个节点。单节点对等组的领导就是该节点本身。这可以通过配置来得到，或由某些故障引起。

4.1.3 上一等级(level)

4.1.3.1 逻辑组节点

“逻辑组节点”是对一个对等组的抽象，以便在上一个 PNNI 选路等级中代表该对等组。例如在图 3 中，逻辑组节点 A.2 在更高级对等组 A 中表示对等组 A.2。图 3 表示了如何把图 2 中的对等组组织到上一个等级中。

逻辑组节点的功能和它的子对等组领导节点的功能紧密相关。在本标准中这两个节点的功能被认为在同一系统中执行，因此这两个节点间的接口没有指定。

逻辑组节点的功能包括汇集(aggregate)和归纳(summarize)它的子对等组的信息，并把这些信息泛播到它自己的对等组中。逻辑组节点也把从它的对等组中收到的信息送给其子对等组的 PGL 以供泛播。逻辑组节点不参与 PNNI 信令。

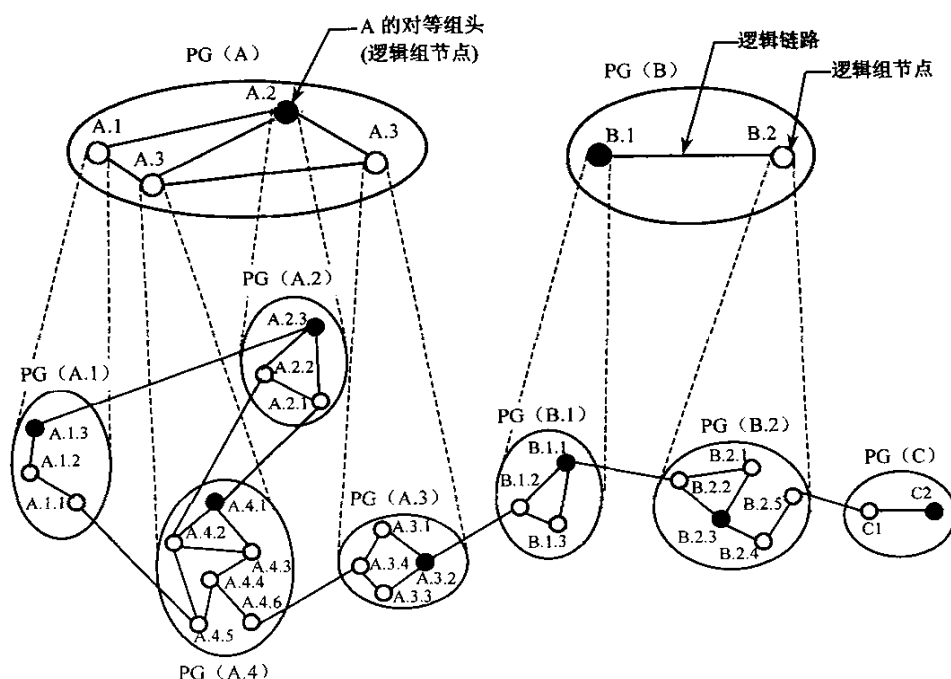


图 3 表示最低等级层次的部分PNNI选路结构

逻辑组节点由节点标识符来标识，该节点标识符是节点所表示的对等组的对等组标识符的缺省值。

逻辑组节点可以通过唯一的 ATM 端系统地址来寻址，例如，它可以与同一交换系统中使用不同 SEL 取值的最低级节点地址相对应。

表示对等组的方式取决于对等组领导的策略和算法，因此在采取相同的策略和算法对等组中指定的两个潜在的对等组领导，对等组的表示并不依赖于选择了两个对等组领导中的哪一个。

在图 3 中，逻辑组节点组成了对等组。例如，逻辑组节点 A.1、A.2、A.3 和 A.4 组成了对等组 A。这个更高级的对等组是 4.1.2 节意义上的对等组，唯一的不同是它内部的节点都代表了一个独立的更低级的对等组。因此，对等组 A 有一个有领导选举过程产生的对等组领导（逻辑组节点 A.2）。定义 A 的对等组领导的功能在 A.2 节点内实现，相应地在包含最低级节点 A.2.3 的交换系统中执行。

对等组 A 称为对等组 A.1、A.2、A.3 和 A.4 的父对等组。反过来，对等组 A.1、A.2、A.3 和 A.4 称为对等组 A 的子对等组。

父对等组由对等组 ID 标识，此 ID 在长度上比子对等组的更短。任何可以成为对等组领导的节点必须配置它的父对等组 ID。

对等组 ID 长度表明了该对等组在 PNNI 层次中的等级。ID 长度就像“等级指示语”一样。PNNI 的等级不是连续的，就是说不是所有的级别都应用于任一特定的拓扑结构中。例如，对等组 ID 长度为 N 比特，它的父对等组的标识长度就可以为 $1 \sim N-1$ 个比特。类似地，若对等组 ID 长度为 M 比特，它的子对等组的标识符长度为 $M+1 \sim 104$ 个比特（104 是对等组标识的最大长度，等于 13 个八位位组）。

4.1.3.2 向上供给的信息

一个逻辑组节点表示了整个下属（underling）对等组。作为下属对等组的成员之一的对等组领导接收来自此对等组内所有节点的拓扑状态信息。这样就使对等组领导具有了生成逻辑组节点所需的全部信息。这可以理解为对等组领导给它生成的逻辑组节点提供信息。这种上行信息流包括两类信息：可达性和拓扑汇集。可达性指的是归纳的地址信息（见 4.1.5 节），它用于确定通过较低级对等组可到达哪个地址。拓扑汇集则是归纳的拓扑信息（见 4.1.3.8 节），它用于选路进入或经过这个对等组。

在归纳过程中有一固有的过滤功能，它只传输更高级所需的信息。PTSE 从不向上层流动。相反地，归纳信息会在逻辑组节点产生的 PTSE 内通告，并被泛播到它的对等组中。

4.1.3.3 向下供给的信息

在 PNNI 选路层次中向上提供信息用于创建等级层次和分配子对等组的选路信息。反之，向下层提供信息是为了使较低级的对等组内的节点通过 PNNI 选路域选择到所有可达目标节点的路由。路由计算利用这种信息来选择到目的地的路由。

每个逻辑组节点给它的下属对等组提供信息，这些下发的信息包括它自己产生所有的 PTSE 和接收到对等组内其他 LGN 成员泛播的所有 PTSE。向下传送到对等组领导的 PTSE 在该对等组内泛播。这就使对等组内每个节点对它所汇集到更高级的状况有所了解。PTSE 在对等组内水平传送并向下传送到子对等组。

4.1.3.4 上行链路

当邻节点从 Hello 协议得知它们属于不同的对等组时，它们就成为边界节点。例如图 4 中，节点 A.3.4 和 A.4.6 就是边界节点。在不同对等组内的边界节点之间的链路称为外部链路（outside link）。在外部链路上无数据库交换，Hello 协议是唯一的 PNNI 协议流。

边界节点在外部链路上发送的 Hello 协议还包含了关于各自较高级对等组和表示它们的逻辑组节点信息（节点层次表），这种信息边界节点可以确定出两个边界节点公共的最低级对等组。如图 4 中，可判断出边界节点 A.3.4 和 A.4.6 共属于对等组 A。

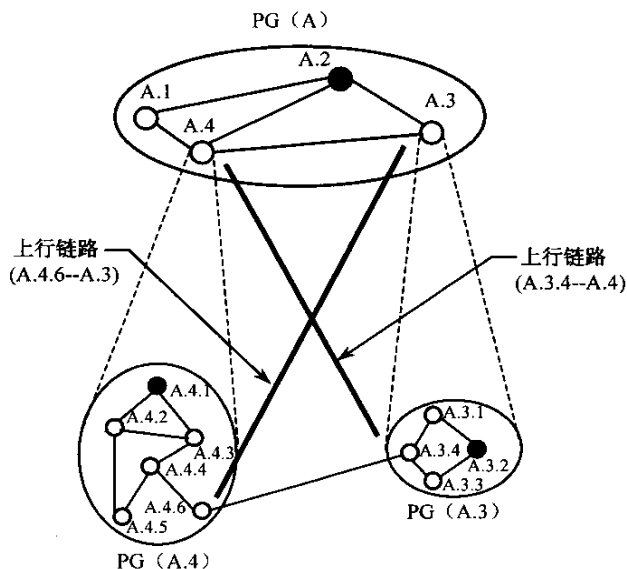


图 4 上行链路

上述机制使每个节点都能了解到它所在的对等组的全部拓扑结构（包括节点和链路）、较高级父对等组完全的拓扑结构（归纳的）和祖先对等组的拓扑结构等。为使节点知道哪个（边界）节点同更高级节点有连通性，边界节点必须向较更高级节点通告这种链路，这种链路就称为上行链路（uplink）。在上行链路另一端的节点，也就是上行节点（upnode），始终是它的一个祖先节点的邻节点（在图 4 表示的两级的例子里，是它的父节点的邻节点）。

对于某些 Hello 状态而言，边界节点必须在外部链路上发送包含一个上行链路信息属性（ULIA）信息组的 Hello 分组包。发往外部邻节点的 ULIA 包括一整套关于反向资源的信息，这些信息将在接收端外部邻节点生成的上行链路通告中通告。

节点层次表提供给边界节点必要的信息，用来确定它们公共的较高级对等组，并且用来标识边界节

点声明的到哪个较高级节点有连通性。例如，在图 4 中，边界节点 A.3.4 发现它的邻节点 A.4.6 由同一父对等组 A 内的逻辑组节点 A.4 表示。因此 A.3.4 就在自身和上行节点 A.4 之间通告一条上行链路。这条上行链路定义为 (A.3.4--A.4)。类似地，节点 A.4.6 有一条上行链路 (A.4.6--A.3) 可到上级节点（或上行节点）A.3。

边界节点在各自对等组内泛播的 PTSE 来通告上行链路，这使得对等组内的所有节点能够把上行链路加入到它们的拓扑数据库。由于上行链路帮助建立上级对等组，对等组领导必须把可达性信息提供给上级。

由于上行节点不通告上行链路下行方向的 PTSE，所以上行链路双向的拓扑状态参数包含于上行链路的 PTSE 内。上行链路反向的拓扑状态参数在外部链路上的 Hello 分组包内进行交换。

4.1.3.5 PNNI 选路控制通路

相邻 PNNI 节点有一条用于 PNNI 选路信息交换的选路控制通路。邻节点在它们的 PNNI 选路层最低级使用了一条预留的 VCC 用于它们的选路控制通路。位于逻辑组节点之间的选路控制通路是 SVCC。从逻辑组节点表示的对等组内的上行链路通告中，可以获得建立这条 SVCC 所需要的信息。

例如，要在 A.3 和 A.4 之间建立一条 RCC 的话，应采取以下步骤。对等组领导 A.3.2 收到了从 A.3.4 泛播来的描述上行链路 (A.3.4--A.4) 的 PTSE。A.3.2 就从该 PTSE 中提取出两条重要的信息：一条信息是其所在的公共对等组的 ID，它在前面已通过 Hello 协议在 (A.3.4--A.4.6) 上的执行确认为对等组 A。另一条是上行节点 A.4 的 ATM 端系统地址。节点 A.3.2 再把这些信息传送给它的逻辑组节点 A.3。从这些信息中，A.3 推断出它和 A.4 同属于同一个对等组，同时也推断出 A.4 是它的邻节点并因此需要有一条到 A.4 的 RCC。

既然 A.3 可以到达 A.4 的 ATM 端系统地址，它就有充足的信息建立 SVCC。类似地，逻辑组节点 A.4 也有充足的信息建立到 A.3 的 SVCC。

当用于 RCC 的 SVCC 建立时，两个相邻对等组内的节点具有选择 SVCC 所需的全部信息。特别的情况下，每个对等组有它自己的内部拓扑信息，并知道上行链路和上行节点的身份与存在，其中上行链路和上行节点是同基于 SVCC 的 RCC 选路相关的。

4.1.3.6 上一级拓扑数据库交换初始化

较高级的逻辑组节点和较低级的节点工作方式很相似。如图 3 中，当 A.3 和 A.4 间基于 SVCC 的 RCC 建立时，Hello 协议就在该 RCC 上被激活。A.3 和 A.4 确认它们在同一对等组内，就会在 RCC 上执行一个初始的拓扑数据库交换。

注：A.3 和 A.4 已经知道了它们属于同一对等组，否则它们就不会建立 RCC；Hello 协议仅仅是确认二者关系。

在 A.3 和 A.4 之间的初始化拓扑数据库交换会使两节点的数据库同步。由于 A.3 和 A.4 的子对等组特有的信息并不属于两者的数据库。因而在 A.3 和 A.4 之间的初始化拓扑数据库交换只包括对等组 A 内泛播的 PTSE。由于 PTSE 水平传送和向下传送（见 4.1.3.3 节），数据库交换也将包含由 A.2（A 的对等组领导）传送给对等组 A 的 PTSE。我们假设在链路 (A.3--A.4) 出现前 A 的对等组领导就已经存在了。否则，此时对等组中将没有可用的上级信息。

4.1.3.7 水平链路

水平链路是同一对等组内的节点之间的逻辑链路。因此不仅 (A.3.4--A.3.3) 是水平链路，(A.3--A.4) 也是水平链路（见图 4）。(A.3--A.4) 在上级对等组 A 中并作为 4.1.3.4 节里导出的上行链路的结果被建立。Hello 分组包通过基于 SVCC 的 RCC 送到相邻 LGN，交换端口 ID 和水平链路的状态信息。

在邻节点之间的 RCC 上完成 Hello 分组包交换和数据库同步之后，水平链路才能建立。这时描述新链路的 PTSE 就可以在包含此链路的对等组内（例如：对等组 A 内）泛播，并向下传送到子对等组。

一对 LGN 之间的水平链路表明了该两节点间的连通性。逻辑组节点负责分配水平链路和其他链路端口 ID。

4.1.3.8 拓扑汇集 (topology aggregation)

拓扑汇集是一种通过减少节点和链路的信息来获得扩展网络的能力。这样不仅降低了复杂度，也可

以出于安全考虑隐藏对等组的内部拓扑结构。

1) 链路汇集

链路汇集是指两个对等组之间的某组链路由单独一条逻辑链路表示(见 5.10.3.1 节)。如图 4-3 中, 连接 A.2 和 A.4 的链路表示了两条链路 (A.2.1--A.4.1) 和 (A.2.2--A.4.2) 的汇集。

逻辑组节点负责链路汇集, 逻辑组节点检查所有从它的子对等组到特定的上行节点的上行链路的通告。所有到同一上行节点的上行链路且具有相同的汇集令牌, 就被汇集成为单独一条链路。如果上行节点是一个对等的逻辑组节点时, 这条链路可以是水平链路, 否则就是导出的上行链路(见 4.1.4.2 节)。

2) 节点汇集

节点汇集是用父对等组内的逻辑组节点表示子对等组的过程。

“复杂节点表示”(complex node representation)用于在父对等组内表示这种汇集的结果。复杂节点表示也允许用来规定最低级节点。

最简单的复杂节点表示是具有相同半径的对称星型结构。星型的中心是逻辑节点的内部参考点, 称为核(nucleus)。核与逻辑节点的一端之间的连通性称为轮辐(spoke)。两轮辐的串联(concatenation)表示一个对称对等组的穿越。对称星型拓扑用作“缺省节点表示”, 它由每个节点状态参数的单个值组成, 这些参数值是两个方向上逻辑节点与核心的出、入口之间的假设值。

通常对等组是非对称的。例如它们可能含有“分离节点”(outlier), 即被删除后能使对等组的对称性大大提高的节点。这种非对称性可以用一些例外(exception)来规定。例外可以表示其与核的连通性与缺省值明显不同的特殊端口。另外, 例外能用来表示两端口之间的连通性, 即旁路(bypass), 旁路比穿越核要好得多。

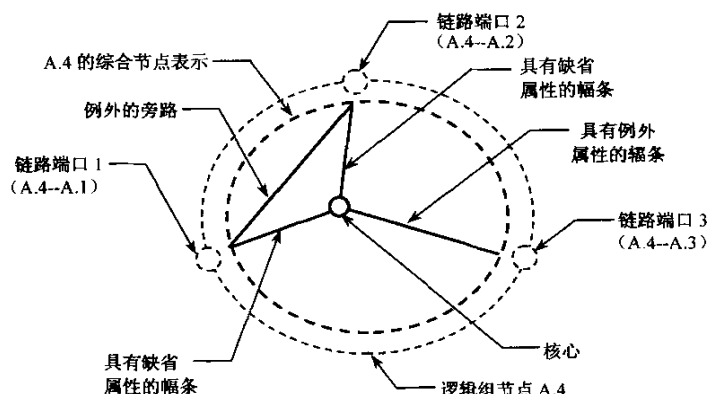


图 5 图3的逻辑组节点A.4的复杂节点表示

复杂节点表示在下例中说明。在图 3 中的对等组 A.4 由在图 5 中表示的综合节点归纳。核心表示逻辑组节点 A.4 的内部。从核发出的轮辐与逻辑组节点 A.4 的一端连在一起。图 5 含有 3 条轮辐, 每端口一条。这 3 个端口与图 3 的关系如下:

- 端口 1 表示 (A.4--A.1) 链路上的端口;
- 端口 2 表示 (A.4--A.2) 链路上的端口;
- 端口 3 表示 (A.4--A.3) 链路上的端口。

端口 1 与 2 的轮辐使用了缺省属性, 而到端口 3 的是例外。出现例外的可能原因是节点 A.4.1 到 A.4.5 紧密地簇集在一起, 而 A.4.6 却是距离远的分离节点。

节点穿越(traversal)也能用旁路了核的例外来建模。例如图 5 中, 在 A.1 和 A.2 之间传递时, 连接端口 1 与 2 的例外旁路对应穿越 A.4 节点, 这个例外可以由节点 A.4.2 和 A.4.5 之间的高速旁路引起。

节点 A.4 通过把 PTSE 泛播到 A.1、A.2 和 A.3, 来分配它的复杂节点表示。接着 PTSE 被向下传送到它们各自的对等组领导(见 4.1.3.3 节), 对等组领导又把相应的 PTSE 泛播到自己所在的对等组内,

结果对等组 A.1、A.2、A.3 和 A.4 内的所有节点的拓扑数据库均包含有 A.4 的复杂节点表示信息。

通过一个逻辑组节点选路对应于选择串联轮辐和/或旁路。在这样一个串联中不可能有两条以上的轮辐，否则将形成一个环路。选择的这个串联必须满足呼叫的资源需求。例如图 3 和图 5 中，假设逻辑链路 (A.1--A.2) 拥塞，那么节点 A.3.4 选择路由到对等组 A.1 中的一个节点，要通过逻辑链路 (A.3--A.4)、(A.4--A.1) 或 (A.3--A.2)、(A.2--A.4)、(A.4--A.1)。进一步假设，从选路标准的角度看两条路由是等效的（而不是因为两者都经过相同数量的链路）。那么路由选择就选择 3 条复杂节点穿越中最好的一条（见图 5）。

- a) 采用缺省属性的两轮辐的串联；
- b) 采用缺省属性的轮辐同例外轮辐的串联；
- c) 例外旁路。

如果最好的是例外旁路，那么 (A.3--A.2)、(A.2--A.4)、(A.4--A.1) 就是优选的路由。因而节点 A.3.4 在选择路由到对等组 A.1 中的一个节点时，使用链路 (A.4.2--A.4.5)。

选择到逻辑组节点内一可达地址，对应于选择轮辐或者是经过核心带有可接受属性旁路与轮辐的串联。

4.1.4 完整 PNNI 选路层次

4.1.4.1 形成最高级对等组

前面 PNNI 选路层次的例子仍是不完整的。较高级的对等组（见图 3）没有显示出彼此之间的连通性。图 6 显示出了一个可能的完整 PNNI 层次。不断创建更高级的对等组，直到整个网络都包含在唯一的最高级对等组内，就形成了完整的 PNNI 层次。在图 6 的例子里，完整性是通过配置另一个包含逻辑组节点 A、B 和 C 的对等组而获得的。节点 A 表示对等组 A，而后者又表示了对等组 A.1、A.2、A.3、A.4 等等。另一种可能的配置是对等组 B 和 C 汇集成一个对等组 BC，然后 BC 又同 A 汇集成一个最高级对等组。网络设计者通过配置定义逻辑节点和对等组的参数，进而控制网络的层次结构。

4.1.4.2 上一级的水平链路和上行链路

PNNI 选路层次允许非对称。对于一个给定的较低级对等组，它的父节点能同时是其他较低级对等组的各级祖先。例如在图 6 中，较低级对等组 C 是由最高级对等组内的逻辑组节点 C 直接表示的，而低级对等组 B.1 和 B.2 先是汇集到父对等组 B 内，再由最高层对等组内的逻辑组节点 B 表示。

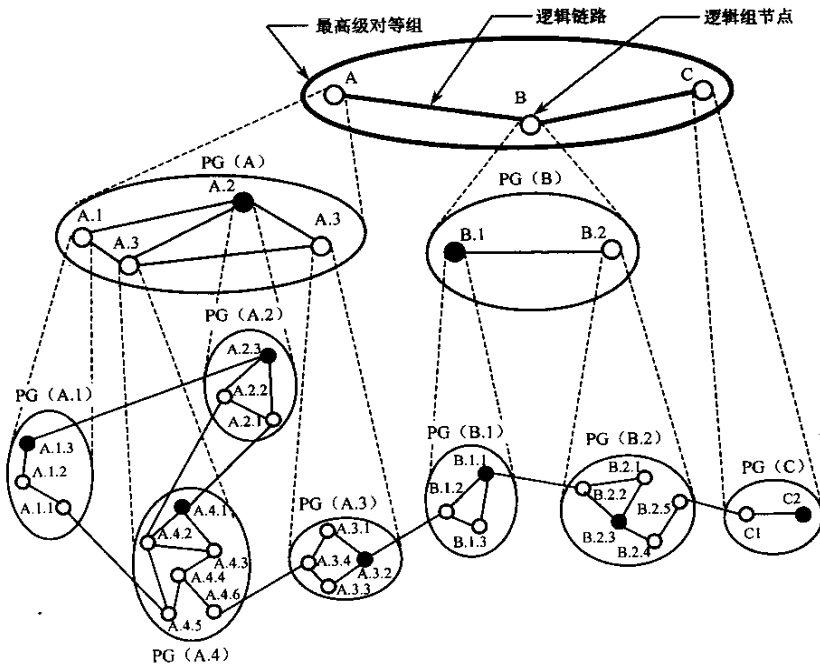


图 6 完整 PNNI 层次网络

如图 7 所示, 给出了 4 条上行链路 (B.2.5--C)、(C.1--B)、(B.1.1--B.2) 和 (B.2.2--B.1), 然而, 上行链路 (B.2--C) 与其他 3 条上行链路有所不同, 它是从上行链路 (B.2.5--C) 得到的。这称为“推导的上行链路”。

当对等组领导 B.2.3 收到 B.2.5 泛播的描述上行链路 (B.2.5--C) 的 PTSE 时, 它就把公共的对等组 ID (在此例中是最高级对等组) 和上行节点 C 的 ATM 端系统地址传送给 LGN B.2。由这些信息, B.2 标识出节点 C 不是对等组 B 的成员。因此它推导出 1 条新的上行链路 (B.2--C), 该链路反映了从 B.2 至 C 的可达性。由于 B.2 代表了 B.2.5, 那么 (B.2--C) 就是 (B.2.5--C) 的高层表示。

如果有另一条链路连接 B.2.1 和 C.1, 则上行链路 (B.2--C) 就可以表示两条低级上行链路的汇集, 即 (B.2.5--C) 和 (B.2.1--C)。这将由两条上行链路的汇集令牌决定。

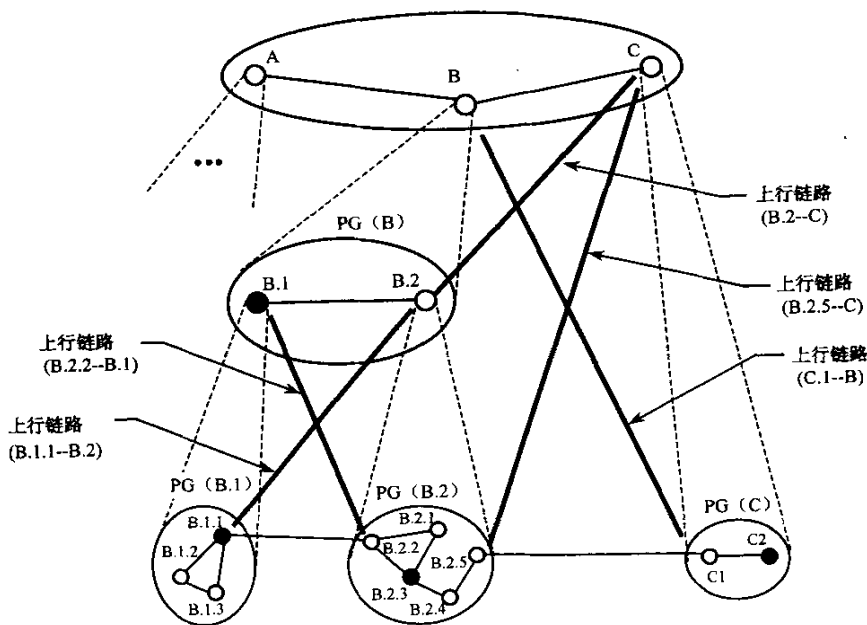


图 7 上一级的上行链路

上行链路也可以从低一级的推导得到。在本例中, 如果在对等组 B 和最高级对等组之间有附加的对等组, 将有到对等组 C 的附加上行链路。

一条基于 SVCC 的 RCC 在 LGN B 和 C 之间建立, 步骤如下。创建完上行链路 (B.2--C) 后, B.2 在它的对等组 B 泛播一个 PTSE 来说明这条上行链路。一接收到上行链路的信息, 对等组领导 B.1 就把公共对等组 ID (在此例中是最高级对等组) 和上行节点 C 的 ATM 端系统地址传送给它的 LGN B。节点 B 由此标识出节点 C 位于它的对等组内并因此具有足够的信息建立一条 LGN B 与 C 之间的基于 SVCC 的 RCC。

节点 B 经过通告了一条到 C 的上行链路的边界节点, 来选择一条到节点 C 的路径。此例中 B.2 是与上行节点建立上行链路的唯一的节点, 因此 SVCC 将通过边界节点 B.2。

4.1.4.3 层次的递归

PNNI 选路层次的创建被看作是对等组递归生成, 它由最低级节点开始到包含整个 PNNI 路由域的唯一最高级对等组结束。层次结构由配置时对等组 ID 和逻辑组节点联系的方式决定。

4.1.5 地址归纳和可达性

地址归纳减少了需要在 PNNI 网络中分配的寻址信息的数量, 可以扩展网络。它用“可达地址前缀”表示端系统和/或节点地址的集合。可达地址前缀可以是归纳地址或是外来地址。

和节点相关的“归纳地址”是地址前缀, 它可以在该节点明确配置, 也可以采用一些缺省值。和节

点相关的“外来地址”是与该节点的任何归纳地址不匹配的地址。与其相反，“本土地址”是匹配于该节点的某归纳地址的地址。

这些概念在图 8 中说明，图 8 是从图 6 推导出的。节点 A.2.1、A.2.2 和 A.2.3 的附属端表示端系统。与每个端系统相关的数据表示了该端系统的 ATM 地址。例如 <A.2.3.2> 表示了一个 ATM 地址，P<A.2.3>，P<A.2> 和 P<A> 相继表示相同的 ATM 地址的更短的前缀。

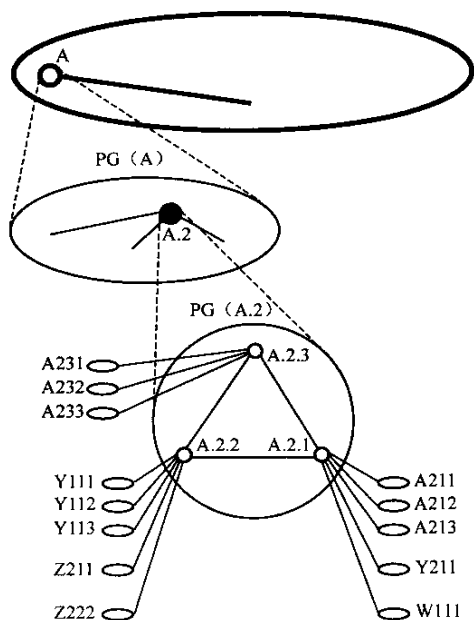


图 8 PNNI 层次中的地址归纳

对等组 A.2 里，每个节点配置的归纳地址的样表如表 1 所示：

表 1 配置的归纳地址

A.2.1 处的归纳地址	A.2.2 处的归纳地址	A.2.3 处的归纳地址
P<A.2.1>(配置的)	P<Y.1>(配置的)	P<A.2.3>(配置的)
P<Y.2>(配置的)	P<Z.2>(配置的)	

其他的归纳地址应被选择，例如在节点 A.2.2 中是 P<Y.1.1> 而非 P<Y.1> 或是节点 A.2.1 中的 P<W>。但是 P<A.2>（非 P<A.2.1> 或 P<A.2.3>）不能选择为节点 A.2.1 和 A.2.3 中的缺省的归纳地址，因为选择了一条路由的远端节点不能区分它是属于 A.2.3 的端系统还是属于 A.2.1 的端系统。

注：对于 A.2.1 内选定的归纳地址表，P<W.1.1.1> 是一个外部地址，它不与任一节点的归纳地址匹配。

LGN A.2 需要它自己的归纳地址表如表 2 所示。由于 PG(A.2) 是对等组 A.2 的 ID，它应当把 P<A.2> 当作缺省的归纳地址。同时由于归纳地址 P<Y.1> 和 P<Y.2> 能被 P<Y> 更进一步地归纳，这也就要求把 P<Y> 配置为归纳地址。

表 2 归纳地址表

LGN A.2 归纳地址表
P<A.2>(缺省的)
P<Y>(配置的)

而 P〈A.2〉则未被包含在 LGN A.2 的归纳地址表内。由于对等组 A.2 内的每个节点都有可能成为对等组领导，所以每个节点在它的 LGN 配置都包含相同的归纳地址表。

表 3 给出对等组 A.2 里每个节点通告的可达地址前缀。节点 A.2.1 泛播它的归纳地址和外来地址，节点 A.2.2 和 A.2.3 则只发泛播归纳地址，这是因为没有外部寻址端系统。

表 3 通告的可达地址前缀

节点 A.2.1 泛播的可达地址前缀	节点 A.2.2 泛播的可达地址前缀	节点 A.2.3 泛播的可达地址前缀
P<A.2.1>	P<A.2.2>	P<A.2.3>
P<Y.2>	P<Y.1>	
P<W.1.1.1>	P<Z.2>	

可达性信息(例如：可达地址前缀)在整个 PNNI 选路层次中传送，这样所有节点能根据这些归纳前缀地址到达端系统。与这种信息流有关的过滤功能可以获得更进一步的归纳。例如：LGN A.2 通过与表 2 中的所有归纳地址进行匹配，来归纳在对等组 A.2 中通告的每个可达地址前缀。例如：当 LGN A.2 收到由节点 A.2.2 发出的(见表 1)可达地址前缀 P〈Y.1〉(通过 PGL A.2.3)，并发现它与已配置的归纳地址前缀 P〈Y〉相匹配，LGN A.2 就通过通告它的归纳地址 P〈Y〉而不是较长的地址前缀 P〈Y.1〉来获得更进一步的归纳。

还有另外一种过滤功能与可达性信息的通告相关，它用来限制分配可达地址前缀。通过禁止把与端系统的地址有关“抑制归纳信息”在 LGN 内通告。这种选项允许低层对等组内的某些地址对高层或其他对等组隐含。这种特性出于安全因素而实施的，从而保证某些特殊的端系统地址不为对等组外所知。

不能被 LGN A.2 进一步归纳的可达地址前缀在通告时则不能修改的。例如当 LGN A.2 收到由 A.2.2 发出的可达地址前缀 P〈Z.2〉，而它同表 2 的归纳地址都不匹配，因而 LGN A.2 在通告时就不能修改 P〈Z.2〉。尽管 P〈Z.2〉是来自节点 A.2.2 的归纳地址，但由于它与所有归纳地址都不匹配，LGN A.2 就把 P〈Z.2〉看成是外来的。由 LGN A.2 通告的结果可达性信息在表 4 中列出。

表 4 已通告的可达性信息

LGN A.2 通告的可达性信息
P<A.2>
P<Y>
P<Z.2>
P<W.1.1.1>

尽管节点 A.2.3 是对等组 A.2 的 PGL，但由 A.2.3 通告的可达性信息与由 LGN A.2 通告的信息不同，如表 3 和表 4 中所示，由 LGN A.2 通告的可达性信息只是在该对等组外关于它的可达性信息。

LGN A 和对等组领导 A.2 的关系类似于 LGN A.2 和对等组领导 A.2.3 的关系。如果 LGN A 没有配置归纳地址，那么它将通告所有的可达地址前缀，并将经对等组 A 泛播到最高级对等组(包含表 4 的全部内容)。另一方面，如果 LGN A 配置了缺省的归纳地址 P〈A〉[这个值的设置是因为对等组 A 的标识是 PG(A)]，那么 LGN A 将在通告前进一步归纳所有由 P〈A〉开头的可达地址前缀。例如它将通告归纳地址 P〈A〉而不是由 LGN A.2 泛播的地址前缀 P〈A.2〉(见表 4)。

逻辑节点的 ATM 地址和端系统地址一样，都使用相同的归纳准则。

由特定的 PNNI 节点发出的可达性信息(可达地址前缀)泛播并向上送到相应的父对等组，接着向下又送到相应(子)对等组并进行泛播，从而保证到达位于除此节点以外的所有 PNNI 节点。

4.1.5.1 地址范围

由逻辑节点通告的可达性信息有一个与它相关的范围。这个范围定义了 PNNI 选路层次的一个等级，

而且是此地址能被通告或归纳的最高级。如果某地址的范围指出比该节点所在等级更低，则该节点不会通告这个地址。如果范围指出某级与该节点所在级是同级或是更高级的话，此地址将在该节点的对等组内被通告。

当归纳地址时，要被归纳的地址具有的最高范围决定了归纳地址的范围。同一原则也应用于组地址。例如：如果在对等组内两个或多个节点通告了到相同组地址的可达性，而它们又具有不同的范围，它们的父节点将通告其到具有最高范围的组地址的可达性。

该原则用于地址抑制时比用于范围归纳时优先权高。

4.1.5.2 具有地址范围的地址归纳地址举例

本节给出一个示例，但这个示例大部分是缺省地址归纳时情况。假设有一个如图 9 所示的网络，它包含 3 个等级层次和两个节点，并假设不能被抑制的地址数为 1。

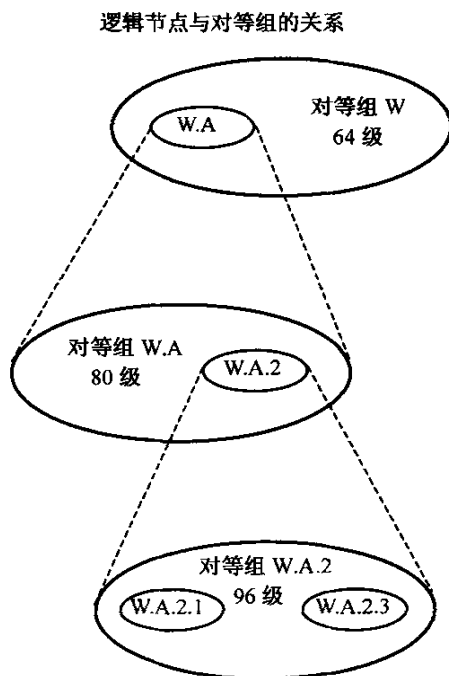


图 9 简单的网络

假设在 96 级的节点 W.A.2.1 具有以下的可达地址：

W.A.2.1.ddd

W.A.2.1.eee

W.A.2.3.fff

W.F.2.7.zzz

A.N.Y.C.ast1 范围 80

A.N.Y.C.ast2 范围 104

A.N.Y.C.ast3 范围 96

A.N.Y.C.ast4 范围 72

该节点将通告如下：

W.A.2.1

W.A.2.3.fff

W.F.2.7.zzz

A.N.Y.C.ast1 范围 80

A.N.Y.C.ast3 范围 96

A.N.Y.C.ast4 范围 72

W.A.2.1 是缺省的节点归纳地址。与它相匹配的地址不需要通告。地址 A.N.Y.C.ast2 没有被通告，这是因为它的范围(104)比节点的级别低。

节点 W.A.2.3 也在 96 级并有如下附加地址：

W.A.2.3.aaa

W.A.2.3.bbb

W.A.2.3.ccc

W.F.1.5.xxx

W.F.2.7.yyy

A.N.Y.C.ast3 范围 80

节点将通告如下：

W.A.2.3

W.F.1.5.xxx

W.F.2.7.yyy

A.N.Y.C.ast1 范围 80

节点 W.A.2 在 80 级。对等组 W.A.2 的对等组领导（是相同的物理节点但工作在 96 级上）收到了由两个节点通告的地址。这些信息又被提供给逻辑组节点 W.A.2，而后者把它们归纳如下：

W.A.2

W.F.1.5.xxx

W.F.2.7.yyy

W.F.2.7.zzz

A.N.Y.C.ast1 范围 80

A.N.Y.C.ast3 范围 80

A.N.Y.C.ast4 范围 72

W.A.2 是节点的缺省归纳地址。所有与它匹配的地址都被抑制。地址 A.N.Y.C.ast3 是由于有足够的范围而被通告的。逻辑组节点 W.A 在 64 级。它再次归纳地址如下：

W.A

W.F.1.5.xxx

W.F.2.7.yyy

W.F.2.7.zzz

它的缺省归纳地址是 W.A。由于组地址的范围均比 64 级大，故而它们没有被通告。

4.1.6 单节点视图

本节得出一个最低级节点构成的网络视图。由于对应的网络描述包含在节点的拓扑数据库中，该视图提供了不少信息。图 10 显示了在图 6 里描述的 PNNI 选路层最低级对等组 A.3 内的所有节点的视图。对于组内全部的 4 个节点 A.3.1~A.3.4 而言这个视图是相同的，这是由于在对等组 A.3 内的泛播保证了它的所有成员的拓扑数据库是一致的。

图 10 所示的视图包括了对等组 A.3 的所有祖先对等组，即对等组 A 和最高级对等组。本例重点在节点 A.3.3 上并且探究它的视图是如何形成的。为简化步骤，假设 A.3.3 的邻节点在 A.3.3 出现前就已经激活。当 A.3.3 初始化时，出现两条新链路（A.3.3--A.3.2）和（A.3.3--A.3.4）。新链路上的 Hello 协议告知 A.3.3 它的邻节点是 A.3.2 和 A.3.4。

通过初始化拓扑数据库交换和泛播，节点 A.3.3 这时可以从它的邻节点处获得整个拓扑数据库。

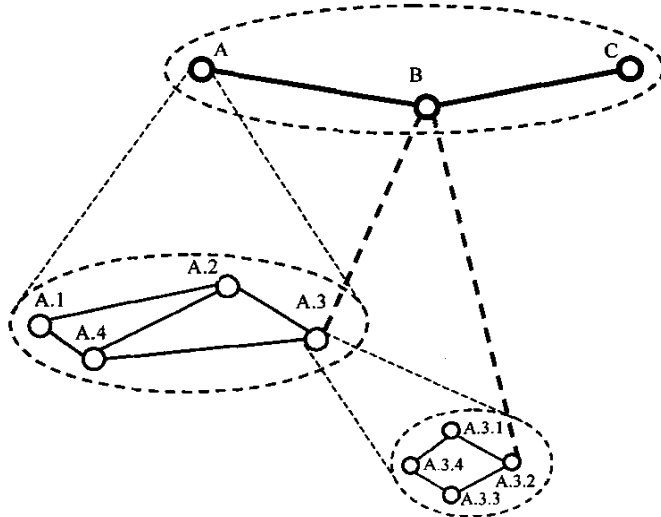


图 10 节点 A.3.3、A.3.2、A.3.1 与 A.3.4 的网络视图

A.3.3 的单节点视图是由拓扑数据库的 PTSE 按如下步骤而形成的。节点 A.3.3 由 PTSE（由对等组 A.3 内的其他节点产生的 PTSE）的信息获知对等组 A.3 的拓扑结构。到上行节点 A.4（在对等组 A 内，见图 6）的连通性是通过由边界节点 A.3.4 创建并泛播的上行链路（A.3.4--A.4）获知的。同样，到上行节点 A.2 的连通性是通过边界节点 A.3.1 创建并泛播上行链路（A.3.1--A.2）而得知的。从对等组 A.3 内的节点到 A.1 之间没有上行链路。上行链路只提供关于低级对等组邻近上行节点（如 A.4 和 A.2）之间连通性的信息。节点 A.3.3 对节点 A.1、A.2 和 A.4 以及它们之间的连通性的完整视图是通过这些节点产生的 PTSE 获知的。

到上行节点 B（见图 6）的连通性是从节点 A.3.2（A.3.2--B）创建的上行链路得到的。A.3.3 节点对节点 B 和 C 以及它们之间的连通性的完整视图是通过这些节点产生的 PTSE 获知的。

拓扑汇集暗示了图 10 中所有不属于对等组 A.3 的节点，以及 A.3 所属的对等组。图 5 所示的复杂节点表示就与 LGNA.4 相关。

拓扑汇集暗示了图 10 中所有不属于对等组 A.3 的链路中。对于链路（A.2--A.4）来说，它就表示了图 6 中的链路（A.2.1--A.4.1）和（A.2.2--A.4.2）的汇集。

在它的拓扑数据库中，A.3.3 具有描述经每个节点的可达地址前缀的通告。例如，与连接到节点 A.2.1, A.2.2, A.2.3（见图 6 和 8）的端系统相关的可达地址前缀，被归纳到来自逻辑组节点 A.2 的通告。

4.1.1.7 通道选择

ATM 是一种面向连接的连网技术。这意味着 PNNI 为 VC 或 VP 的建立选择的通道将持续使用一个时期，也就是说低效率路由选择策略的结果将影响连接。因此 PNNI 谨慎选择路由是非常关键的。

ATM 允许一个用户指定建立呼叫时的服务质量（QoS）和带宽参数值，ATM 网络就必须能保证为该呼叫实现这些参数。呼叫建立包括两个过程：（1）通道选择；（2）在沿该通道每个端点的连接状态的设置。通道选择基于当前有效的信息，使得选择的通道能支持所要求的 QoS 和带宽。在沿着通道的各个节点上，呼叫建立过程保证所要求的资源能真正满足。如果不满足，那么将发生折回，并导致重新计算一条新的通道，因此最终结果要么是满足要求的路径建立，要么是呼叫的拒绝。

对 QoS 敏感的高效的通道选择仍在研究之中。当前一些用于 QoS 敏感通路选择的算法要考虑多种限制，例如：对多个单独的链路参数和绝对的计算开销。因此在选择对 QoS 敏感的通道选择算法时，PNNI 应当具备灵活性。

在网络技术中有两种最基本的选路技术用：起源（source）路由选择和逐跳(hop-by-hop)路由选择。对于起源路由选择方式，起源系统选择到的目的地的路由，在该路由上的其他系统服从起源路由选择

的指令。对于逐跳路由选择，每个系统独立选择下一个段的路由，这样在每段所作的决策都一致的前提下，就使得呼叫逐步向目的地接近。

从原理上讲，这两种选路技术都能应用于面向连接的网络(如 ATM)。然而，逐跳路由选择应用于在这种网络时则存在一些弊端。

首先就是可能出现环路。以下是出现环路的几个可能原因：

- 在采用选路策略上的前后不一致，交换局使用了不同的选路算法；
- 在交换局中选路数据库的不一致(由于拓扑信息的变化还未传输完全的)。

在选路策略上的不一致是影响逐跳路由选择方式的最主要限制，这就使得路由选择必须完全指定，所有系统必须准确地按所指定的选择执行。

由于选路数据库的不一致，各自决定路由会使得虽没有产生环路，但却远不是最优化的路由。这种低效路由建立的任何连接将在连接存活期间一直使用这条路由。

另外，逐跳路由选择的另一个缺点是每个系统中都要重复路由计算开销。这对于面向连接的网络而言还不是非常重要的(在此网络中，开销只在连接建立时产生)，而 PNNI 内对 QoS 敏感的路由选择则会产生更多开销。

在起源路由选择里，起源点负责选择到目的地的路由。它在了解局部网络的拓扑结构的基础上完成这一任务。由于只有一个数据库被涉及到，就不可能形成环路，也不可能因为数据库的不一致而引起路由效率不高。而且由于是起源点选择路由，所以不要求每个系统内使用的算法都相同，并且路由计算的开销只产生一次。另外，也更易于基于特定的考虑进行路由选择，因为对算法而言没有在所有节点都保持一致性的要求。

为了避免在提供面向连接的业务时使用逐跳路由选择带来的问题，并能得到起源路由选择的优点，PNNI 对所有的连接建立都要求使用起源路由选择。这就意味着对等组内的第一个节点选择通过此对等组的整个路由。路由被编码成指定转接序列(DTL)，它直接包含于连接建立请求中。DTL 指定了连接通过对等组时经过的节点，也可以指定在这些节点之间使用的逻辑链路。如果沿此路由的节点由于缺少资源而不能按照 DTL 建立请求的连接。那么该节点必须拒绝此请求并把呼叫请求折回至创建 DTL 的节点。

由于 PNNI 允许多等级层次的选路，起源交换局选择一条包含所有详细的已知等级的到目的地的通道，这称为完整层次的源路由。这样的路由不是详细的源路由，因为它未包含起点的源对等组外部的通道的详细情况，而是把通道抽象成将被转接的逻辑组节点序列。当呼叫建立到达对等组的入口交换系统时，交换系统负责选择一条(较低级的)源路由，用来描述经过该对等组的转接情况。通过低级对等组的通道必须同高一级规定的通道相一致(例如：必须到达有高一级通道指定的“下一段”的目的地)。

起源路由选择意味着对于特定的连接建立请求只有一个节点涉及到通过任一对等组时确切的路由选择。也就是说，不需要每个对等组内的节点都使用相同的路由选择算法。每个节点可以根据节点的能力和呼叫的限制选择一条它认为最合适的路由。出于多种原因，不同的节点可以使用不同的通道选择算法，例如更换新的软件，节点的处理能力有明显不同的网络，或用在有多厂商设备的网络中。同样，用于某一 PNNI 选路层次的通道选择算法会和用于不同级别的算法有所不同(在任一的较低级别中，通路选择必须与上一级选择的通路一致)。

因此，PNNI 不为路由选择指定任何必要的算法，每次执行都可以自由使用任何适当的路由选择算法。

4.1.7.1 通用连接接纳控制

因为 CAC 没有被标准化，通用连接接纳控制(GCAC)用来确定交换系统是否允许一个新的连接。

对于每个连接请求，通道计算都必须基于节点拓扑数据库中的信息，这些信息包括连接业务类别、业务量特性和 QoS 要求。选择的通道应当满足连接的端到端业务量和 QoS 要求，同样也要满足网络有效性标准。连接建立期间，每个交换系统沿着选择的通道执行 CAC 算法以确保每个连接都可以被接纳而不会影响到其他现存的连接。如果一个交换系统接受了连接，它的可接受连接的能力就发生变化。如

果这些变化是显著的,那么将激发新的描述节点更新 (updated) 的资源可用性的 PTSE 产生 (见 5.8.5.2 节)。“显著”变化由配置参数控制定义的算法判断。通过选择配置参数的值来平衡 PTSE 的更新频率和由于信息陈旧引起的呼叫失效可能性。

用于通路选择过程的通用 CAC 判断一条链路或一个节点是否可能有足够的资源来支持新连接。如果交换系统可以接受新连接,则通用 CAC 应包括链路或节点。相反的,如果交换系统拒绝新连接,则通用 CAC 应不包括链路或节点。通用 CAC 描述了一交换系统中实际 CAC 执行的结果。

使用通用 CAC 时,每个交换系统任意使用的 CAC,但交换系统需要通告一系列的拓扑状态参数,用来携带能被通用 CAC 使用的信息。通用 CAC 和一组拓扑状态参数必须足够灵活,允许任何的 CAC 使用时都有效。

4.2 PNNI 信令描述

4.2.1 概述

PNNI 信令基于 UNI4.0 信令的一个子集。它不支持一些 UNI4.0 的信令特性,如代理信令、叶启动加入能力或用户到用户的补充业务等,但它加入了一些新特性,适于动态呼叫建立的 PNNI 路由功能。另外,PNNI 信令不同于 UNI4.0 的信令之处在于它是对称的。

PNNI 信令利用了 PNNI 选路收集来的信息。特别是它使用 PNNI 选路获得的动态维持的可达性信息、连通性信息和资源信息进行路由计算。这些路由是根据当前拓扑结构的节点视图的需要计算的。

为满足要求,除了那些为 UNI 信令而定义的特性之外,PNNI 信令使用了 4 个附加的特性:指定转接序列 (DTL) 用于携带等级层次的完整源路由。折回和替换选路则是用于处理不准确信息在替换通道上重新试呼。随路信令用于 VPC 上的 PNNI 操作。并支持软 PVPC 和 PVCC。

4.2.2 随路信令

在缺省方式下 (没有配置 VPC 用作接口上的逻辑链路时),VPI=0 的信令通路控制物理接口上的所有 VP。PNNI 也支持经单个物理接口的到多目的地的多 VPC 上的信令和选路。在这种情况下,配置的 VPC 作为逻辑链路有一个与之相关的信令通路,在这些 VPC 中的 VC 只由该特定 VPC 的随路信令通路来控制,即在同一个物理接口上的缺省的信令通路 (VPI=0 中的信令通路) 不控制用作逻辑链路的 VPC 上的 VC,但控制物理链路上所有其他的 VC 和 VP。

4.2.3 指定转接序列 (DTL)

在处理一个呼叫时,PNNI 信令会向 PNNI 选路协议请求一条路由。PNNI 使用 DTL 规定路由。DTL 是经过对等组的完整通道,它包含穿过对等组的节点 ID 和端口 ID (任选的) 序列,它由起源节点 (即 DTL 始发者) 或对等组的入口边界节点供给。等级层次的完整源路由表示了经 PNNI 选路域的一条路由,选路域包含了在当前等级和可见的最低级之间的每一个层次结构的选路等级,在这些等级上,源和目的地都是可达的。它表示成按从最低级对等组到最高级对等组排列的 DTL 序列,并组成栈结构 (即后入先出)。栈顶部的 DTL 是和最低级对等组相应的 DTL。

4.2.4 折回

当创建一个 DTL 时,节点使用当前资源和连通性的可用信息。由于某些原因,信息有可能不准确,这些原因可能是产生新的呼叫所引起层次汇集和资源可用性的变化。因此沿着指定的路由根据 DTL 来处理的呼叫可能会被阻塞。折回和替换选路是一种减少清除呼叫返回源的情况的机制。当呼叫不能根据 DTL 处理时,就折回至 DTL 的生成者并带有故障的指示。此节点可以选择一条替换路由处理呼叫或进一步折回呼叫。替换路由必须服从所有收到的高级 DTL,也必须回避已阻塞节点或链路。

4.2.5 软 PVPC 和 PVCC

PVPC 或 PVCC 是永久的虚通道连接或虚通路的连接。“永久”一词表示连接是在管理下 (即在网管下) 建立的而不是按需 (即使用经 UNI 的信令下) 建立的。软 PVPC 或 PVCC 在网络中是由信令完成其建立的。通过配置交换系统在软 PVPC 或 PVCC 一端起动软 PVPC 或 PVCC 的信令建立。这些过程将在附录 C 中详细说明。

网管系统提供地址给软 PVPC 或 PVCC 的一端,用于标识网络的出口。主叫端负责建立、释放和 (随

机的)重建呼叫。

4.2.6 折回的连接建立举例

本节里, 举出一个详细的例子, 描述包括折回和替换选路的连接建立。例中的网络如图 11 所示。假设 A.3.3 和 B.1.2 之间的链路拥塞(尽管 PNNI 选路信息提示说是正常的), 且 A.3.1 和 B.2.3 之间的链路不能支持呼叫要求的峰值信元速率。

考虑一下图 11 中的对等组 A.[1~3]和 B.[1~3]的详细视图。假设要在属于交换系统 A.1.2 的端系统 A.1.2.X 到属于交换系统 B.3.3 的端系统 B.3.3.Y 之间建立一个连接。A.1.2.X 向 A.1.2 发出作为一个正常的 UNI SETUP 的建立请求。A.1.2 检查它的视图, 发现目的地地址 B.3.3.Y 通过节点 B 可达。如在对等组的名字中出现的那样, B 可以不是目的地 ATM 端点地址的一部分的另一个字符串。这就是为什么从 B 来的通告包含可达地址前缀的原因。如上述, 从节点 B 而来的 PTSE 实际上通告了它能到达一个描述多端系统(它包含端系统 B.3.3.Y)的地址前缀。

经检查拓扑结构, A.1.2 发现路由是 (A.1.2, A.1.1, A.2, A.3, B)、(A.1.2, A.1.1, A.2, B) 或 (A.1.2, A.3, B)。假设在度量和策略的基础上, A.1.2 选择了 (A.1.2, A.1.1, A.2, B)。那么在栈里将构造 3 个 DTL:

DTL: [A.1.2, A.1.1], 指针-2

DTL: [A.1, A.2, A.3], 指针-1

DTL: [A, B], 指针-1

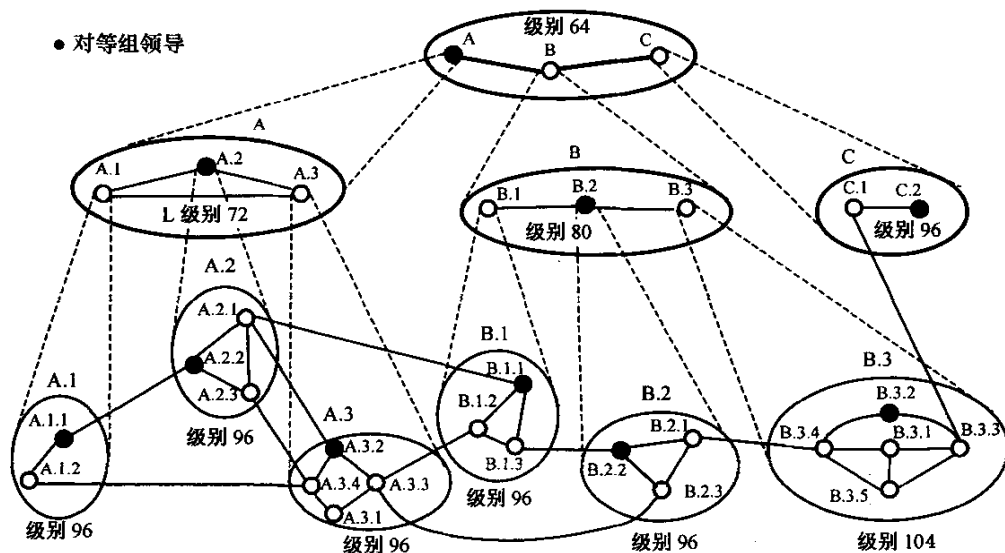


图 11 简单的层次网络

每个 DTL 列出一个给定的层次等级处呼叫建立需要访问(包括当前节点)的一些节点。

除了顶部 DTL 由于处理进展指示了哪个节点将接着被访问, 其他每个 DTL 的当前转接指针指定转接序列中的哪一个成员是当前级中正被访问。转接指针并未真正编码成一条索引, 而更像是一个偏移八位位组。为了实现透明性, 本节里它被简化成索引。DTL 表被构造成 DTL 堆栈的形式(即顶部 DTL 持续工作; 当顶部 DTL 到达末尾时, 就从转接序列中移去该 DTL 并开始检查下一个 DTL)。

在继续建立呼叫前, A.1.2 保存 SETUP 消息的内容以备使用替代路由。然后 A.1.2 把呼叫建立到它的邻节点 A.1.1。A.1.1 检查顶部 DTL 后发现 DTL 正指向它自己的节点 ID。A.1.1 就在顶部 DTL 中查找下一个入口, 但是发现它已经是末尾, 则继续查找下一个 DTL, 下一目的地是 A.2。因为 A.1.1 不是 A.2

(即 A.1.1 未被归纳到逻辑节点 A.2 的对等组内)，它就开始考虑如何到达 A.2。A.1.1 发现有一个到 A.2 的直接邻节点，所以它移去顶部 DTL 并把当前转接指针指向下一个 DTL：

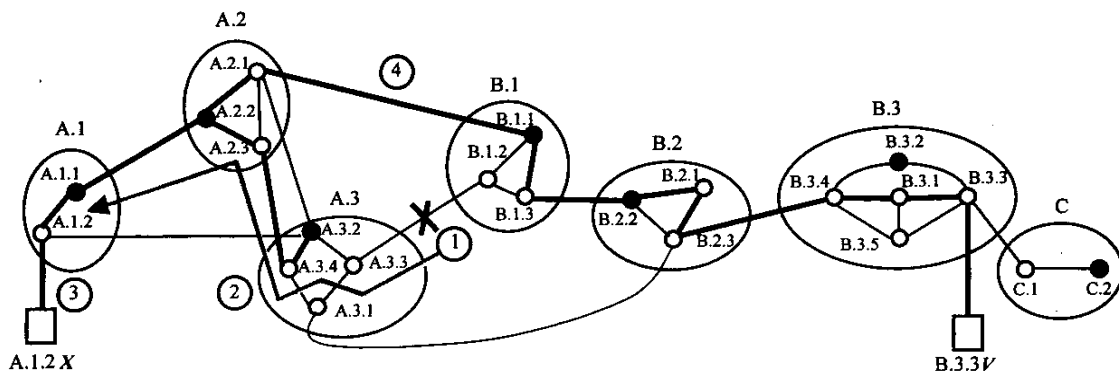


图 12 折回及替换路由

DTL: [A.1, A.2, A.3], 指针-2

DTL: [A, B], 指针-1

A.1.1 并没有添加任何 DTL，也没有做过任何选路决策，所以 A.1.1 不会涉及到替换选路，而且在传送呼叫建立到 A.2 之前不必做 SETUP 消息的拷贝。

A.2.2 检查顶部 DTL，发现当前目标是 A.2。由于 A.2.2 在 A.2 中，它就检查 DTL 中的下一个入口，然后开始选路到 A.3。分析了拓扑结构后 A.2.2 得知路由是通过 A.2.3 的，所以它把该 DTL 加入表内。

DTL: [A.2.2, A.2.3], 指针-2

DTL: [A.1, A.2, A.3], 指针-2

DTL: [A, B], 指针-1

由于 A.2.2 加入了一个 DTL 到栈里并有可能参与替换选路，所以在向 A.2.3 发送分组包之前，A.2.2 保存收到的 SETUP 消息的内容和它加入栈内的 DTL 的内容。

A.2.3 起初判断已经到达了顶部 DTL 目标且 DTL 已经用完。最终，A.2.3 注意到下一个 DTL 的目标是邻节点，所以 A.2.3 删除顶部 DTL 并传送当前传送指针给下一个 DTL：

DTL: [A.1, A.2, A.3], 指针-3

DTL: [A, B], 指针-1

呼叫建立到达 A.3.4。由于 A.3.4 在 A.3 内，目标已经到达了。剩下的目标就是 B 了。通过 A.3.2 和 A.3.3，A.3.4 建立一条到 B 的路由，并加入一个新 DTL 到表中：

DTL: [A.3.4, A.3.2, A.3.3], 指针-2

DTL: [A.1, A.2, A.3], 指针-3

DTL: [A, B], 指针-1

由于 A.3.4 加入了一个 DTL 到栈里并有可能参与替换选路，所以在向 A.3.2 传送呼叫之前，A.3.4 保存收到的 SETUP 消息的内容。

A.3.2 收到呼叫建立，传送当前呼叫指针和建立请求给 A.3.3。A.3.2 不参与替换选路，因此不必保留 SETUP 消息的任何拷贝。A.3.3 收到呼叫建立并决定传送 SETUP 消息给它在 B 内的邻节点。在传送 SETUP 消息之前，A.3.3 删除顶部的两个 DTL，这是因为两者都已被占用而且 SETUP 消息是 A.3.3 和 A.3 发出的。

DTL: [A, B], 指针-2

在这一点处，呼叫建立被阻塞，这是 A.3.3（前向节点）根据 A.3.3 内部的 CAC，或是根据包含阻塞传输类型为 #2 的，“在接口的后向端呼叫被阻塞”（遇忙返回 IE 的 RELEASE 或 RELEASE COMPLETE 消息）而发现的，如同 1 号标记环指示的那样。遇忙返回过程此时开始，A.3.3 发回 RELEASE 消息给前

向接口。RELEASE 消息中的遇忙返回 IE 指出，在 A.3.3 到 B 的链路上呼叫被阻塞，这是根据来自收到的 SETUP 消息的 DTL 而判断的，并指出遇忙返回级别是 96（对等组 A.3 所在级别）。

RELEASE 消息在 A.3.2 处收到，A.3.2 不为此呼叫创建任何 DTL，所以遇忙返回可以进一步进行下去。当后向 VC 拆除时或拆除之后，RELEASE 消息被发送到 A.3.4 方向的前向接口。

RELEASE 消息在 A.3.4 处收到，A.3.4 在 96 级为这个呼叫指定了一个 DTL，因而 A.3.4 试图进行替换选路。不考虑这条阻塞链路，在 A.3 到 B 之间找到两条不同的路由。一条通过 A.3.1 到 B，另一条通过 A.3.2、A.3.3、A.3.1 再到 B。然而，A.3 到 B 之间的链路不支持最初 SETUP 消息要求的峰值信元速率，所以 A.3.4 决定继续遇忙返回过程而不是进行其他的选择（沿着最初的路由回溯可能不是好的路由）。通过前向接口 A.3.4 发送另一个 RELEASE 消息，但改变了阻塞传送标识符来表示从 A.3 到 B 的链路被阻塞了，同时也如 2 号标记环表示的那样改变遇忙返回级为 72（A.1 的对等组所在级别）。

RELEASE 消息在 A.2.3 处收到，A.2.3 不为此呼叫创建 DTL，所以遇忙返回可以更进一步进行下去（向 A.2.2 的方向）。

RELEASE 消息在 A.2.2 处收到，A.2.2 不为此呼叫创建 DTL。A.2.2 检查遇忙返回 IE 中的遇忙返回级并判断出该级别比它自己创建的 DTL 的级别更高，所以遇忙返回可以更进一步进行下去（向 A.1 的方向）。

RELEASE 消息在 A.1.1 处收到，A.1.1 不为此呼叫创建 DTL，所以遇忙返回可以更进一步进行下去（向 A.1.2 的方向）。

RELEASE 消息在 A.1.2 收到，A.1.2 得知遇忙返回级和它为这个呼叫指定的 DTL 的级别相同，并发现有相应 SETUP 消息内容的拷贝。它将试图进行替换选路，如 3 号标记环指示的那样，而且 A.1.2 决定试选路由（A.1.2，A.1.1，A.2，B），因而产生如下的 DTL 栈：

DTL: [A.1.2, A.1.1], 指针-2

DTL: [A.1, A.2], 指针-1

DTL: [A, B], 指针-1

A.1.2 接着就传输呼叫建立到 A.1.1。A.1.1 从栈里删除顶部 DTL，传送当前呼叫指针给下一个 DTL，并传送呼叫给 A.2.2。A.2.2 找到通过 A.2.1 到 B 的一条路由，就加入一个新的 DTL 到栈内，并传输呼叫建立给 A.2.1：

DTL: [A.2.2, A.2.1], 指针-2

DTL: [A.1, A.2], 指针-2

DTL: [A, B], 指针-1

A.2.1 收到呼叫建立而且得知 B 是邻节点。它就从栈里删除顶部的两个 DTL，传送当前呼叫指针给下一个 DTL，并传送呼叫建立给 B.1.1，如图 12 里 4 号标记环指示的那样：

DTL: [A, B], 指针-2

B.1.1 收到呼叫建立，发觉当前 DTL 目标已经到达。此时 B.1.1 必须建立一条新的源路由到最终的目的地。它注意到该路由将途经 B.2 到 B.3，所以它就加入相应的 DTL 到栈里。接着就计算出通过 B.1 的路由只是到 B.1.3，故而它也在栈里继续加入相应的 DTL，给出如下：

DTL: [B.1.1, B.1.3], 指针-2

DTL: [B.1, B.2, B.3], 指针-1

DTL: [A, B], 指针-2

B.1.1 传送呼叫建立给 B.1.3。在通常情况下经核实，B.1.3 删除顶部 DTL，传送当前 DTL 指针给 B.2 中它的邻节点。B.2.2 发现它就是当前目标，并找到途经 B.2 到 B.3 的一条路由，就在表内加入 DTL 如下：

DTL: [B.2.2, B.2.1, B.2.3], 指针-2

DTL: [B.1, B.2, B.3], 指针-2

DTL: [A, B], 指针-2

接着 B.2.2 就传输呼叫建立给 B.2.1。B.2.1 查看顶部 DTL 并传输它，又把呼叫建立传给 B.2.3。B.2.3 删除该 DTL，传输指针，接着把呼叫建立传给 B.3 里它的邻节点。B.3.4 建立一个新的 DTL 如下：

DTL: [B.3.4, B.3.1, B.3.3], 指针-2

DTL: [B.1, B.2, B.3], 指针-3

DTL: [A, B], 指针-2

B.3.1 查看顶部 DTL 并传送当前传送指针，又把呼叫建立传给 B.3.3。B.3.3 判断 DTL 栈最终在这里结束，理由如下：

- 所有的 DTL 都已结束；
- B.3.3 是最低级的节点；
- B.3.3 具有到目标的可达性。

B.3.3 就从建立消息里除去 DTL 栈，并经 UNI 传送此消息到目标。

4.2.7 支持带作用域的任播

由逻辑节点通告的可达性信息包括组地址的信息，它有一个与其相联系的范围（见 4.1.5.1 节）。范围服务于两个目的。首先它定义最高级，在这个等级里一个给定的可达地址（或前缀）可以被通告。其次，它为用于任播目的的给定的组地址定义了业务的范围，即最高级对等组包含了主叫方不能与之建立连接的外部组地址。

对于每个任播连接的要求，用户在 UNI 内使用连接字段选择 IE 来指示连接选路域，即它允许网络可以向外多远寻找任播地址。利用局部视图，头一个节点把连接范围选择 IE 的值转化为一个 PNNI 选路级，并利用它计算到任播地址的一条路由。PNNI 选路级没有在 PNNI 信令中被携带（但连接范围，如主叫用户所指定的那样是被携带的，因而当呼叫进入其他的选路域时它是可用的）。然而如下例所示，在多个具有不同范围的任播组地址在它们各自的对等组内被通告的条件下，DTL IE 携带了足够的信息用于入口边界节点确定适当的目的地。

参考图 11 所示的网络配置并做如下假设：

B.2.1 通告到组地址 G1 的可达性，它的范围为 80。

B.2.3 也通告到组地址 G1 的可达性，但它的范围为 64。

利用 4.1.5.1 节里讨论的地址范围规则，B.2 和 B 也将通告到 G1 的可达性，通告的范围为 64。

假设在 A.1.1 的一个用户要求建立一个到 G1 的任播连接，连接范围选择局部转变为 PNNI 选路级 64。再假设从 A.1.1 一端来看，对等组 A 内没有节点通告了到 G1 的可达性（在对等组 A.2 或 A.3 内的某些节点可能通告了到 G1 的可达性，但它们的范围不比它们各自对等组的范围大）。结果，A.1.1 计算了到 B 的一条路由并产生下面的 DTL：

DTL: [A.1.1, A.1.2], 指针-2

DTL: [A.1, A.3], 指针-1

DTL: [A, B], 指针-1

这些 DTL 将选路连接 B.1.2，在 B.1.2 上被叫地址（即 G1）是用于第一次计算较低级的路由的。在这种情况下，B.1.2 得到从 B.2 而来又去往 G1 的可达性通告并计算出下面的 DTL：

DTL: [B.1.2, B.1.3], 指针-2

DTL: [B.1, B.2], 指针-1

DTL: [A, B], 指针-2

当连接建立到达 B.2.2 时，被叫方地址重新用于计算较低级的路由。在这种情况下，也就是在从 B.2.1 到 B.2.3 的方向上，B.2.2 得知了对 G1 的多个通告。B.2.2 从最高级 DTL（即[A, B]）推断出主叫方是在包含 A、B 的对等组内，但是在对等组 B 外。因此 B.2.2 选择到 B.2.3，而不是到 B.2.1。

5 PNNI 路由规范

本章提供了 PNNI 路由协议的详细规范，本章从用于整个协议的可操作的程序开始，5.2 节~5.5 节

是系统单元的规范，5.5节~5.13节是执行 PNNI 的程序，5.14节包括所有分组包格式。

5.1 通用的操作程序

5.1.1 定时器

跟踪消息传输的定时器可以是抖动的，这样做是为了防止不希望的传输同步，这可能导致网络的不稳定。

每次重置定时器时，对停止时间值采用新的随机的百分比变化来实现抖动，百分比变化的范围是正常值的 $\pm 25\%$ 。

5.1.2 分组包传输

除 Hello 分组包（见 5.6.2.2 节）之外的其他所有分组包都根据 Hello 数据结构中的版本域所给出的协议版本进行编码。

5.1.3 分组包的有效性

对任一收到的分组包，如果 PNNI 的分组包头中的分组包长度超过了收到的数据长度，就直接丢弃分组包而无需进一步处理。

如果 PNNI 分组包头中的分组包类型不可识别则丢弃整个分组包。

任何其他类型或长度域分析的错误（包括长度不是 4 的倍数的情况）可通过忽略错误的单元处理、封装单元或忽略整个分组包的方式进行处理。

如果收到的分组包是不支持的版本（见 5.6.1 节），分组包必须被丢弃，是否产生管理报告取决于具体实施。对于非 Hello 分组包，收到的分组包如果其支持的版本与期望的版本值不符都将被丢弃。

5.2 寻址

5.2.1 ATM 寻址协定

PNNI 选路层次的寻址和标识是以 ATM 端系统地址和/或用于 ATM 端系统地址的前缀为基础。ATM 端系统地址是基于图 13 中所示的 NSAP 地址。

ATM 端系统地址有 20 个八位位组长。PNNI 仅根据 ATM 地址的前 19 个八位位组进行选路。选择器（第 20 个）八位位组对端系统只具有本地意义，因此 PNNI 的选路将不考虑这一分组包。PNNI 不使用、检查或解释出现的附加的子结构。但有一个的例外，PNNI 使用第一个八位位组，即鉴权和格式标识符（AFI），来区分单个地址，这种单个地址是用来标识单一的 ATM 端系统，或标识一个或多个的 ATM 子系统的组地址。AFI 有效的单一值和组地址值在 UNI4.0 的表 A5-1 中给出。在这里为了完整性的原因，包括了 NSAP 地址的子结构（IDP, DSP, ESI 和 SEL），但它对于选路操作并不重要。有关 ATM 端系统地址的子结构的更多信息可参见 UNI4.0 的 3.1 节，分配 ATM 组地址的原则可见 UNI4.0 的附录 5。

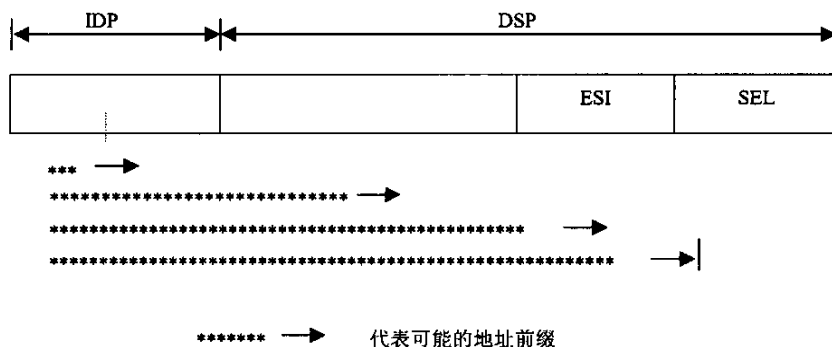


图 13 NSAP地址结构和地址前缀

ATM 端系统地址的前缀是地址的前“ p ”个比特。 p 的值可取由 0~152 之间不同的值。长度为 0 的前缀将所有 ATM 端系统地址归纳进一个通告里；PNNI 将把没有更具体匹配的呼叫直接选路到通告前缀长度为 0 的系统（即“缺省”路由）。长度大于 0 小于 152 的前缀用于归纳寻址域的某个部分，较短

的前缀比较长的前缀能归纳更多的部分（即较长的前缀更具体）。

在图 13 中通过星和箭头的组合提供了一组可能的地址前缀，注意箭头像前缀一样总是用于地址的左边的大多数比特，前缀长度决定了这些箭头能向右边伸多远。

PNNI 运行在可拓扑的层次环境中，层次的结构由在路由域中使用的对等组 ID（见 5.3.2 节）来定义，地址分配的层次通常在适当的规模上与拓扑层次相对应。这使得地址归纳（即地址前缀），可代表到以规定的前缀开始的所有地址的可达性。在通用的层次中，特殊的情况是地址用较长的前缀来描述。

5.2.2 节点地址

主动参加 PNNI 选路的节点用 ATM 端系统地址来寻址。

一个交换系统可能包括多个节点。每个这样的节点都要求有一个独一无二的 ATM 端系统地址。这个地址用于建立在 PNNI 中使用的 SVCC。交换系统为它示例的这些节点产生唯一地址的一个方法就是通过使用选择器的不同值。

5.2.3 端系统地址

ATM 端系统地址有 20 个八位位组。如前所说，PNNI 选路是在前 19 个八位位组的基础上进行的。选择器八位位组仅用来区别在同一接口上可到达的目的地。

通常，PNNI 将使用 ATM 端系统地址前缀形成的归纳向终端系统通告可达性。如果附属于节点的端系统不符合配置的节点归纳，节点有必要对那个端系统进行明确的通告，通告中携带全部 152bit（19 个八位位组）的前缀长度。端系统可达性归纳的前缀长度 $< 152\text{bit}$ 。当寻址层次与拓扑层次一致时，就有可能用一个前缀向许多端系统通告可达性。较短的前缀（即较小的前缀长度值）归纳较多的地址，反之亦然。

或许需要计算到仅支持 E.164 地址的网络内的端系统的路由。由于这个原因，PNNI 允许端系统可达性可以通过使用嵌入的 E.164 NSAP 格式的 E.164 地址的前缀进行通告。虽然 PNNI 是为专用网的使用设计的，但能 PNNI 也可以在公众 ATM 网络内部使用。

既然归纳可包含其他归纳，有可能有许多处于不同等级的归纳与给定的目的地地址匹配。PNNI 的路由计算总是将呼叫直接选路到通告与给定目的地最佳匹配的逻辑节点。最佳匹配的定义是与（归纳）通告匹配时前缀最长。

5.3 标识符和表示语

5.3.1 等级表示语

PNNI 实体（节点、链路和对等组）发生在不同层次的等级上。等级定义了比特字符串长度，范围从 0~104。

祖先是两个实体中具有较高等级的实体，并且比另一个实体的等级表示语要小。

从等级表示语定义对等组 ID 有效比特的确切数目的意义上来看，它是一个绝对值。

从任意特定的拓扑中不是所有的等级都会被用到的意义上看，PNNI 的等级不是“稠密的”。ID 长度为“ n ”比特的对等组可以有 ID 长度是 $0 \sim n-1$ 比特范围内任意值的父对等组。类似的，ID 长度为 m 的对等组可以有一个子对等组，其 ID 长度是 $m+1 \sim 104\text{bit}$ 范围内的任意值。

5.3.2 对等组标识符

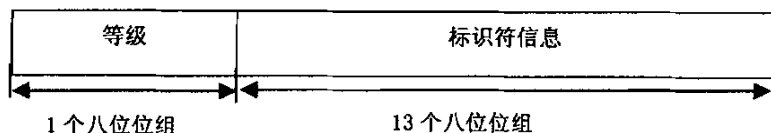


图 14 对等组标识符的 14 个八位位组的编码

对等组的标识符是长度在 0~104bit（13 个八位位组）之间的比特串。对等组的标识符必须是 ATM 端系统地址的前缀以使管理对等组的组织在前缀上有分配的权威性。例如，如果一个组织分配了 n bit 的前缀，它可以分配给对等组长度为 n 或大于 n 的标识符，但不能小于 n 。

对等组标识符使用 14 个八位位组进行编码；1 个八位位组 (8bit) 等级表示语跟着 13 个八位位组 (104 比特) 的标识符信息。等级表示语的值必须介于 0 和 104 (bit) 之间。在标识符信息域中发送的值的编码，其右端的第(104- n) 位必须置成 0，其中 n 是等级。

5.3.3 节点标识符

节点标识符有 22 个八位位组长，包括 1 个八位位组长的等级表示语，后跟一个 21 个八位位组长的不透明的值，即这 21 个八位位组没有包含内部结构。在整个路由域中，这个不透明的值必须是唯一的。节点的等级与它包含的对等组的等级是相同的。收到 PNNI 分组包中的节点标识符的节点绝对不能对内部结构做其它假定。目前已定义了产生节点 ID 的两种方法，但也不排除产生节点 ID 的其他方法。

对于不代表对等组的节点：

——等级表示语定义了节点包含的对等组的等级；

——第二个八位位组取值 160；这有助于将这种情况与下面的其他情况区分开来，因为对等组 ID 的编码不能以这个值开始；

——节点 ID 的剩余部分包含由节点代表的系统的 20 个八位位组的 ATM 端系统地址。

对于在父对等组 A 中代表子对等组 A.2 的逻辑组节点：

——等级表示语定义了包含 LGN 的对等组的等级（即对等组 A 的等级）；

——后 14 个八位位组是子对等组 A.2 的对等组 ID 的编码；

——后 6 个八位位组包含执行逻辑组节点功能的物理系统的终端系统标识符 (ESI)。

注：节点标识符域中包含的 ESI 使同一等级的多个节点能够被区分，以防较低等级对等组被分区。

——节点 ID 的最后一个八位位组是 0。

在 PNNI 路由协议运行的过程中，运行在特定等级上的节点将收到包括来自位于同一等级的对等组中节点的 PTSE 和更高等级对等组中节点的 PTSE。每个 PTSP 的源点通过在每个 PTSP 中包括起源节点 ID 来定义。要求执行 PNNI 选路的系统能够接受这样的入局分组包而不考虑节点 ID 的不透明值部分的格式。

两个 PNNI 路由分组包当且仅当它们包含相同的 22 个八位位组的起源节点 ID 时，才可以确定是来自同一节点。节点 ID 的第一个八位位组是最高有效八位位组。

当节点在运行时，即当节点有任意邻近点，hello FSM 处于非 Down 的其他任何状态，或者任一拓扑数据库的入口时，它的节点 ID 是不允许改变的。

5.3.4 端口标识符和逻辑链路

端口 ID 是由节点分配的 32bit 号码，以标识到这个节点的逻辑链路的附着点。由节点 ID 标识的端口 ID 仅在分配节点 ID 后才有意义。

值 0 和 0xFFFFFFFF 保留且不能用来标识特定的端口。

逻辑链路由链路的任一端点的节点 ID 和由那个节点分配的端口 ID 标识。

ATM 逻辑链路是双向的，且在每个方向上有着不同的特征。对等组内链路的每个端点都通告端口 ID 和通往外部链路的特征。正在通告链路的每个端点的节点互相交换它们的端口 ID。这是在 PNNI 的 Hello 分组包交换的过程中实现的。

边界节点为通告的上行链路分配本地端口 ID，且在两个方向上通告链路特征和上行节点的节点 ID (见 5.10.2 节)。

在节点方面，来自节点的被通告的每一条链路必须有唯一的端口 ID。

5.3.5 汇集令牌

汇集令牌是 4 个八位位组的标识符，它与远端节点 ID 一起用来标识在层次的下一级被汇集到一起的上行链路。汇集令牌和远端节点 ID 也可以用于将较高等级的链路绑定到相邻的较低等级的链路上。

汇集令牌的意义的范围限于成对彼此知道的逻辑组节点。在一对逻辑组节点之间的所有有着相同汇集令牌值的链路必须作为一条逻辑链路进行通告，同一个令牌值可以用于其他成对的节点间而不会引起语义冲突。然而，为了管理的简单和灵活，范围被选择的足够大使得在全球范围内可以分配独一无二的

令牌值。

外部链路的每个终端的节点同意在使用 5.10.3.1 节中的算法的有关上行链路中通告汇集令牌。

汇集令牌包括在把上行链路描述成汇集和捆绑信息的 PTSE 中。汇集令牌也包括在把水平链路描述成捆绑信息的 PTSE 中。

5.3.6 地址范围

PNNI 使用的地址和地址前缀与范围有关。个别和组地址都有成员范围，以用来决定可到达地址（见 5.9.1 节）的通告范围。对于组地址，范围也可用来进行连接范围控制，这使得主叫用户可以在路由层次的特定等级上与同组的成员通过使用 ATM 的任播能力约束点到点连接请求（见 5.13 节，6.4.5.23 节，6.5.2.1 节和附录 B）。

对于 PNNI，可达地址范围由等级表示语来定义。在 UNI，包括在使用 ILMI 地址登记时（在 ILMI 对象 `atmfAddressOrgMemberScope` 中），范围由 UNI 信令 4.0 的 A5.2 节中定义的组织范围的 15 个等级中的一个来指示。网络具体的匹配用在 UNI 和 PNNI 指出的路由等级表示语之间组织范围的翻译。匹配用于两个成员范围，一个经 ILMI 地址登记传递通过 UNI，然后和 PTSE 中的相应的可达地址一起作为 PNNI 的路由等级通告给连接范围，一个在 SETUP 消息中传递通过 UNI，匹配是网络可配置的对象，匹配的缺省值如下表 5 所示。

表 5 缺省的 UNI 范围与 PNNI 等级之间的匹配

UNI 范围	PNNI 路由等级表示语
1~3	96
4~5	80
6~7	72
8~10	64
11~12	48
13~14	32
15 (全局)	0

5.4 逻辑链路

逻辑链路是两个逻辑节点间的连通性的抽象表示。逻辑链路包括单独的物理链路、单独的虚通道连接、上行链路和逻辑链路的汇集，它们也可以是两个逻辑节点间的平行的逻辑链路。如果两个最低等级节点由 PNNI 的虚通道连接 (VPC) 连接，每个节点都必须知道（通过配置/管理）VPC 存在且用于 PNNI。

5.5 PNNI 路由控制通路

PNNI 路由控制通路 (RCC) 是用在逻辑上或物理上相邻的逻辑节点间交换 PNNI 路由协议分组包 (Hello, PTSP, PTSE ack 等) 的 VCC，有 3 种情况：

1) 对于 PNNI 路由信息在物理链路上的交换，保留 VPI=0 的 VCC，使用 VCI=18 的 VCC。

2) 对于 PNNI 路由协议在 VPI=V 的虚通道连接上的交换，PNNI 路由协议的交换将在 VPC 内的 PNNI VCC 上发生，即 VPI=V，VCI=18。

3) 对于 PNNI 路由协议消息在逻辑组节点间的交换，为了这个目的建立一个 SVCC，且为 SVCC 在正常通道中通过信令分配 VPI 和 VCI。

5.5.1 PNNI 路由控制通路使用的 AAL 规范

本节定义了 PNNI 路由控制通路使用的 ATM 适配层，PNNI 协议分组封装在 AAL 业务数据单元中。

由 RCC 使用的 AAL 是 AAL 类型 5 (AAL5)。AAL5 有两个子层，即由公共部分和业务特定部分子层组成的。公共部分的 AAL 协议提供了不确定的信息传输和 PNNI 路由协议使用的 AAL-SDU 错误检

测的机制(错误的或丢失的 AAL SDU 的恢复由 AAL 之上的 PNNI 路由协议处理)。AAL5 用于支持 PNNI RCC。AAL5 公共部分的协议在 ITU-T 建议 I.363 中定义。RCC 使用空的业务特定会聚子层 (SSCS)，空的 SSCS 在 ITU-T 建议 I.363 的 AAL 类型 5 部分进行了描述。

在 ITU-T 建议 I.363 中描述的消息模式中使用了 PNNI RCC。这意味着只有一个完整 PNNI 分组包能被封装到一个 AAL-SDU 中。此外每个 AAL-SDU 都只包括一个完整的 PNNI 分组包。

5.5.2 物理链路上的 RCC 的业务量合约

两个通过物理链路连接的最低级节点间的 RCC 将使用下列缺省的业务量描述符:

- 业务类别是 nrt_VBR
- PCR(CLP=0+1)=RCCPeakCellRate
- SCR(CLP=0)=RCCSustainableCellRate
- MBS(CLP=0)=RCCMaximumBurstSize
- 采用标注
- 允许丢弃帧

5.5.3 VPC 上 RCC 的业务量合约

通过 VPC 连接的两个最低等级节点间的 RCC，在缺省的情况下与 VPC 有相同的业务类别，且使用下列缺省的业务描述符:

CBR:

- PCR=RCCPeakCellRate

VBR (rt 和 nrt):

- PCR=RCCPeakCellRate
- SCR=RCCSustainableCellRate (注 1)
- MBS=RCCMaximumBurstSize (注 1)
- 标注 (注 2)
- 允许帧丢弃

ABR:

- PCR = VPC 的 PCR
- MCR = 0.005*VPC 的 MCR

UBR:

- PCR = VPC 的 PCR

注 1: SCR 和 MBS 值用于同样的 CLP 子流作为 VPC 的 SCR 参数。

注 2: 当 SCR 参数应用于 CLP=0 的子流时, 缺省值是采用标注。否则, 缺省值是不采用标注。

5.5.4 基于 SVCC 的 RCC 连接建立概述

当建立的 SVCC 用作 PNNI 的路由控制通路时, LGN 使用本地配置信息, 提供建立 SVCC 所有必需的呼叫建立参数值。当收到建立 RCC 的呼叫时, 如果这个节点不支持 SETUP 消息中的呼叫建立参数值, 那么这个参数可能被拒绝。但在判断接受或拒绝这个呼叫时, 不考虑本地 PNNI 配置的参数值。要求所有的 PNNI 的实施支持下文指出的缺省值。

LGN 请求非实时的 VBR 业务。如果非实时的 VBR 业务不可用而实时的 VBR 业务可用, 那么 LGN 将在新的连接请求中请求这一业务; 如果实时的 VBR 业务不可用而 CBR 可用, 就请求 CBR; 如果 CBR 不可用而 ABR 可用, 就请求 ABR; 如果没有其他的业务类别可用, 就请求 UBR; 如果发现没有路由有那个业务类别, 或由于业务类别、业务量参数、QoS 参数的问题, 这个业务类别的呼叫建立失败时, 则认为该业务类别也不可用。

SVCC 建立的呼叫和连接程序在 6.5 节定义。下面详细叙述了 SETUP、CONNECT、RELEASE 和 RELEASE COMPLETE 消息的内容, 特别在 RCC 的情况下, 如果所请求的信息单元没有出现, 呼叫就必须被拒绝。注意在本标准的 6.4 节中描述的其他信息单元也可能出现, 使用的编码标准在本文件的

6.4.5.1 节中讨论。

5.5.4.1 SETUP 消息的内容

1) AAL 参数

AAL 参数信息单元（见 6.4.5.8 节）必须用在 SETUP 消息中，表 6 列出了使用的参数。

表 6 AAL 参数

字段	值
AAL 类型	5（用于 AAL5）
前向最大的 CPCS-SDU 大小	8192 个八位位组
后向最大的 CPCS-SDU 大小	8192 个八位位组
SSCS 类型	0（空 SSSCS）

2) ATM 业务量描述符

ATM 业务量描述符信息单元（见 6.4.5.9 节）必须出现。在下表中包括了建立 RCC 时所需要的参数。主叫端系统有可能希望以不同于表 7 中给出的缺省值的信元率来建立连接。在这种情况下，被叫方可以拒绝这个呼叫，因为它不能支持所请求的连接。

当被叫方拒绝了以用户信元率参数为基础的呼叫时，主叫用户可采取如下操作。如果响应 SETUP 消息，主叫 LGN 收到 RELEASE COMPLETE 消息或后面跟有 RELEASE 消息的 CALL PROCEEDING 消息，并且 RELEASE 消息带有原因编码（原因#37，用户信元率不可得），主叫可以检查原因 I.E. 的诊断字段并为指定参数选择了较小的值后重试呼叫。如果收到的 RELEASE 和 RELEASE COMPLETE 消息带有原因编码（原因#73，不支持的业务参数组合），它可以尝试 6.4.5.9 节允许的其他组合。

表 7 ATM 用户信元率/ATM 业务描述符

字段	值
前向峰值信元率（CLP=0+1）	RCCPeakCellRate(nrt-VBR,rt-VBR,CBR 情况下）（注）
前向可维持信元率（CLP=0）	RCCSustainableCellRate(nrt-VBR,rt-VBR 情况下）
前向最大突发容量（CLP=0）	RCCMaximumBurstSize(nrt-VBR,rt-VBR 情况下）
后向峰值信元率（CLP=0+1）	RCCPeakCellRate(nrt-VBR,rt-VBR,CBR 情况下）（注）
后向可维持信元率（CLP=0）	RCCSustainableCellRate(nrt-VBR,rt-VBR 情况下）
后向最大突发长度（CLP=0）	RCCMaximumBurstSize(nrt-VBR,rt-VBR 情况下）
最佳效果表示语	（仅在 UBR 时包括）
前向帧丢弃	允许帧丢弃（SETUP 消息中）
后向帧丢弃	允许帧丢弃（CONNECT 消息中）
前向标志	请求标志（SETUP 消息中，nrt-VBR,rt-VBR 情况下）
后向标志	请求标志（CONNECT 消息中，如果在 SETUP 消息中收到了“支持标志”，nrt-VBR,rt-VBR 情况下）

注：对于 ABR 和 UBR 业务类别，缺省的前向和后向峰值信元率（CLP=0+1）等于线速率。

3) 宽带承载能力

这个信息单元（见 6.4.5.10 节）必须在主叫用户发送的 SETUP 消息中使用，它用来指出需要的是哪种网络连接，字段值如表 8 所示。本标准使用非实时 VBR（nrt-VBR）的业务类别。当主叫用户从网络收到 RELEASE、RELEASE COMPLETE 消息，其中包含原因值#57（承载能力未授权）、#58（承载能

力目前不可得)或#65(承载能力未实现),表明不支持业务类别 nrt-VBR 的指示时,必须尝试业务类别 rt-VBR;如果业务类别 rt-VBR 不可用,则必须尝试业务类别 CBR;如果 CBR 也不可用,那么必须尝试 ABR;如果 ABR 不可用,则尝试 UBR。

在信令消息中,业务类别用 BCOB-X 对承载类进行编码,并用这个值对这个信息单元中的 ATM 传输能力字段的业务类别进行编码。

表 8 宽带承载能力

字段	值
承载类	BCOB-X
ATM 传输能力	非实时 VBR 实时 VBR 固定比特率 可获得的比特率 未定义的比特率
滑码敏感度	0(对滑码不敏感)
用户平面连接配置	点到点时是 0

4) 宽带低层信息

在 SETUP 消息中必须使用这个信息单元(见 6.4.5.12 节),字段值如表 9 所示。它用于指出在连接中携带的协议类型。这个编码使用 ATM Forum 分配的 24bitOUI,其 PID 指出 PNNI 路由控制通路。

表 9 宽带低层信息

字段	值
用户信息三层协议	11(ISO/IEC TR 9577)
ISO/IEC TR 9577 初始协议表示语	64(SNAP 标识符-0x80,长度超过两个八位位组,左面已被验证)
来自前向的持续的八位位组	持续的(Ext 比特置为 1)
SNAP ID	0x80(指出 SNAP 和后面有 PID)
SNAP 可组织的单元标识符	0x 00 A0 3E(ATM Forum OUI)
PID	0x000A

5) QoS 参数

这个信息单元(见 6.4.5.28)必须用在主叫用户发送的 SETUP 消息中,字段值如表 10 所示。

表 10 QoS 参数

字段	值
QoSClassForward	RCCQoSClass
QoSClassBackward	RCCQoSClass

6) 扩展的 QoS 参数

使用业务类别非实时 VBR、实时 VBR 或 CBR 建立 PNNI 路由控制通路时,主叫用户发送的 SETUP 消息中就必须包括扩展的 QoS 参数信息单元,字段值如表 11 所示。缺省情况下没有任何字段包括信元丢失率或信元延迟变化。

表 11 扩展 QoS 参数

字段	值
起源者	0 (对于发端用户) (对于 nrt-VBR, rt-VBR, CBR)

7) 端到端传递迟延

缺省情况下, SETUP 消息中不包括端到端迟延信息单元。

8) 被叫用户号码

这个信息单元 (见 6.4.5.15 节) 必须用在主叫用户发送的 SETUP 消息, 字段值如表 12 所示。用于建立 RCC 的所有 ATM 地址使用了 20 个八位位组 ATM Forum UNI ATM 端系统地址格式。

表 12 被叫用户号码 (ATM 端系统地址)

字段	值
编码标准	0
I.E.指令字段	0
被叫用户长度	20 个八位位组
号码类型	0 (未知情况)
寻址/编号计划	2 (0010B) (ATM 端系统地址情况)

9) 主叫用户号码

这个信息单元 (见 6.4.5.18) 必须用在主叫用户 PNNI 发送的 SETUP 消息。

10) 连接标识符

这个信息单元必须包含在后续返回的第一个消息中 (CALL PROCEEDING 或者 CONNECT 消息中的任一个), 且指出了分配的 VPI/VCI 的值。前侧可以在 SETUP 消息中任意包括这个信息单元来指出对于本连接优选的 VPI/VCI 值 (见 6.4.5.22 节)。

11) DTL

这个信息单元必须包含在由主叫用户发送的 SETUP 消息中 (见 6.4.6.4 节)。

5.5.4.2 CONNECT 消息内容

CONNECT 消息由被叫用户确定格式, 由主叫用户接收。它基本上用于确认连接。

1) AAL 参数

被叫用户可以在 CONNECT 消息中包括 AAL 参数信息单元。

如果这个信息单元由网络返回, 将对它进行校验以确保参数不变或值可以被接受。反之, 将释放呼叫。

2) 宽带低层信息

被叫用户可以在 CONNECT 消息中包括 B-LLI 信息单元。

如果这个信息单元从被叫用户返回, 它必须与 SETUP 消息中的编码请求相同。反之, 释放呼叫。

3) 连接标识符

这个信息单元指出按照 6.5.2.2 节定义分配了什么 VPI/VCI 值。

5.5.4.3 RELEASE 和 RELEASE COMPLETE 消息

PNNI 定义使用下列原因编码以指明为什么 PNNI 端点要释放 RCC。在明确释放 RCC 时使用表 13 所示的编码。

表 13 原因 I.E.

编码标准	0	(ITU-TS), 用于原因值 16 和 31
	3	ATM Forum, 用于原因值 53
I.E.指令字段	0	(未指定)
位置	0	(用户)
原因值	16	正常呼叫清除
	31	正常, 未指定
	53	用于对等组领导的改变清除呼叫 (这是 PNNI 定义的原因码; 见 5.5.6 节)

5.5.5 当一个端点不是 LGN 时的 SVCC

在首次交换了 Hello 分组包后, 最低等级节点可能发现与它的相邻节点不在同一对等组中, 即相邻节点是一个外部相邻节点。通过检查相邻对等节点的层次列表 (见 5.6.2.3 节), 最低级节点发现它在相邻节点的祖先的对等组中, 即最低等级节点与该列表确定的适当 LGN 是对等的。

在这种情况下, 最低等级节点必须准备与相邻对等 LGN 在 SVCC 的基础上进行通信, 它必须遵循 SVCC 发起者的规则来接受或发起 SVCC。然后它必须使用为 SVCC 的操作规定的程序, 如为逻辑组节点间的 SVCC 描述的程序。在这种情况下, 要求最低级节点把外部链路汇集成水平链路。

这种情况的例子可见图 15。节点 A 是最低一级节点, 它到 B.1、B.2、B.3 的链路是外部链路, B.1 等节点都不在节点 A 的对等组内。然而, 从这些外部链路上交换的节点层次列表中, 这些节点发现节点 A 是 B.1、B.2、B.3 所在的对等组的 LGN 的对等节点, 因此就必须在节点 A 和节点 B 之间建立基于 SVCC 的 RCC。

此外, 要求节点 A 依赖与这些外部链路有关的汇集令牌值将外部链路汇集成一条或多条水平链路。例如, 假定在图 15 中使用了两个不同的汇集令牌值: 1) 对于从 B.1 到 A 和 B.2 到 A 的链路; 2) 从 B.3 到 A 的链路。给定这些汇集令牌值, 在节点 A 和 B 之间将产生两条水平链路: 1) 汇集从 B.1 到 A 和 B.2 到 A 的链路; 2) 汇集从 B.3 到 A 的链路。

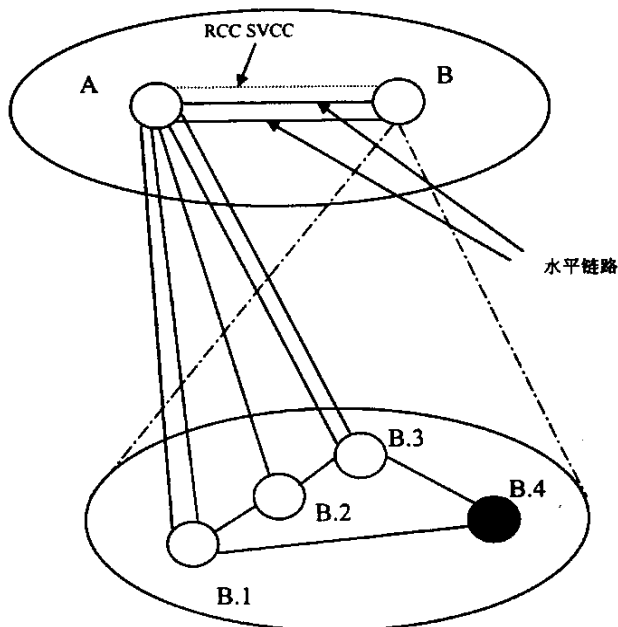


图 15 最低等级节点和LGN间的基于SVCC的RCC和水平链路

5.5.6 建立和保持在逻辑组节点间的 SVCC

5.5.6.1 建立 SVCC

相邻 LGN 间基于 SVCC 的 RCC 是在 PNNI 收集信息的基础上建立的。

SVCC 将穿越到合适的逻辑组节点有上行链路的边界节点。当几个边界节点正在通告到相同的逻辑组节点的上行链路时，SVCC 可以选择任意一个边界节点。

每个 LGN 可以从它表示的对等组内的边界节点通告的上行链路得知相邻 LGN 的地址，特别地，所用的被叫方地址必须是边界节点通告，且位于 SVCC 要通过的上行链路的另一端的相邻的 LGN 的 ATM 端系统地址。为了将来的灵活性，注意对不同的边界节点报告相邻 LGN 的相同的 ATM 端系统地址不作要求。

为了便于 SVCC Setup 消息在 LGN 间的水平链路的通告之前进行选路，对 LGN 产生的 DTL 提出了特殊要求。堆栈底部的 DTL 必须只包含两个逻辑节点 ID，即发起 SVCC 的逻辑节点 ID 和作为 SVCC 目的地的 LGN ID。发起 LGN 的逻辑端口 ID 与此无关。另外，在所有较低的 DTL 中的最后一个节点（即边界节点）的逻辑端口 ID 必须指定一个已经由边界节点在到正确的上行节点的上行链路 IG 中通告的逻辑端口 ID，这个逻辑端口 ID 必须不能为 0。

5.5.6.2 不一致的信息和分区的相邻对等组

正常情况下，在一对相邻的对等组间只有一条 SVCC。然而在某些情况下，在特定的对等组内与同一个相邻对等组有链路的一些边界节点，可能会通告不一致的信息。这样，一些边界节点可能正在通告到一个逻辑组节点的上行链路（LGN 表示了特定的相邻对等组），而另一些边界节点可能正在通告到不同的逻辑组节点的上行链路（LGN 被宣称表示了同一个相邻对等组）。

这种情况的发生至少有两个原因：1) 相邻对等组可能被分区，例如由于物理链路和/或节点故障，2) 相邻对等组的 PGL 可能已经改变，且边界节点在发现和通告这一变化的速度上存在差异。

通常，不可能判断发生了哪一种情况。因此有必要在相邻对等组被分区的假定下进行操作，且打开到每一个被宣称相邻的 LGN 的 SVCC。事实上，如果相邻的 PGL 发生了变化（一个通告已经过期），那么就不可能成功打开到相应 LGN 的 SVCC。

5.5.6.3 建立和保持 SVCC 的具体机制

下文描述了由特定逻辑组节点(叫做 ThisLGN)执行到 ThisLGN 的相邻 LGN SVCC 建立的机制。

当包含上行节点 X 的上行链路通告到达 ThisPGL， ThisLGN 进行处理（ThisPGL 表示 ThisLGN 的子对等组的对等组领导）：

A.1 如果节点 X 比 ThisLGN 的对等组的等级高，那么就产生 ThisLGN 对等组中适当的 PTSE 来公布到 X 的上行链路。

A.2 否则：如果 ThisLGN 到节点 X 有开放的 SVCC，就无动作。

A.3 否则：如果 ThisLGN 的节点 ID 比节点 X 的节点 ID 小，就无动作。

A.4 否则：如果节点 X 比 ThisLGN 对等组的等级低，或者在同一等级但是在不同的对等组，那么产生错误的条件，无操作并且指出了在通告上行链路的边界节点发生了错误。注意如果 ThisLGN 的对等组刚发生了变化， ThisPGL 刚被选出或者 ThisPGL 对等组刚刚愈合了一个分区，则这个错误的条件是有可能发生的；否则这种条件不会发生。

A.5 否则（X 与 ThisLGN 在同一对等组；目前对 X 没有开放 SVCC 且 ThisLGN 有较大的节点 ID）：用值 InitialLGNSVCTimeout 启动定时器 InitialLGNSVCTimer。注意这个定时器可能会抖动。当这个定时器超时，开放到 X 的 LGN 间的 SVCC。

A.6 如果 SVCC 建立尝试成功，那么使用 SVCC 作为 ThisLGN 对等组内的 PNNI RCC（开始交换 Hello 和其他 PNNI 协议）。

当 ThisLGN 建立了基于 SVCC 的 RCC，它是 DTL 的发起者并且产生第一个 DTL：

DTL: [ThisLGN, 节点 X], 指针-1

SETUP (逻辑地) 传递到 ThisPGL 且产生到边界节点的 DTL:

DTL: [ThisPGL, ...,边界节点 1], 指针-1

ThisPGL 负责折回发生时选替换路由。给出 SVCC 的目的是所有实际的路由都要被尝试。

如果 SVCC 建立尝试失败, 那么以值 RetryLGNSVCTimeout 开始 RetryLGNSVCTimer 定时器。注意这个定时器可能有抖动。如果从相同的对等节点 LGN (X) 收到成功的 SVCC 而定时器正在工作, 则取消定时器。

如果 ThisLGN 检测到到相同的邻近 LGN 有两个或更多个 SVCC, 则:

B.1 如果 ThisLGN 的节点 ID 比相邻的 LGN 的节点 ID 小, 那么 ThisLGN 将继续使用所有的 SVCC。Hello 分组包必须在所有 SVCC 上传送。数据库交换分组包, PTSP 可以在任一 SVCC 上传送。在任一 SVCC 上收到的分组包独立于收到它们的 SVCC, 即对于多个 SVCC 来说只有一个 LGN Hello FSM。

B.2 如果 ThisLGN 的节点 ID 比相邻的 LGN 的节点 ID 大, 那么选择一个 SVCC 保持开放状态, 以原因值 16(正常呼叫清除)关闭其他 SVCC。

如果 ThisPGL 不再是 PGL, 则:

C.1 刷新 ThisLGN 发起的所有 PTSE。

C.2 ThisLGN 发送 RELEASE 消息清除到所有相邻 LGN 的 SVCC, 该消息中的原因 IE. 指出原因为 53(由于 PGL 的变化清除呼叫)。

如果到相邻 LGN 存在的 SVCC 关闭, 则:

D.1 如果 ThisLGN 收到带有原因编码值为 53(由于 PGL 的变化清除呼叫)的 RELEASE 消息, 这个消息与到相邻节点 X 的特定 SVCC 有关, 那么将执行下列动作删除各自较高等级的链路。事件 LinkDown 将在到上行节点 X 的基于 SVCC 的 RCC 的 Hello FSM 中触发(见 5.6.3.1 节), 且在所有与 LGN 有关的水平链路 Hello FSM 中触发事件 LinkDown(见 5.6.3.2 节), 以值 RetryLGNSVCTimeout 启动定时器 RetryLGNSVCTimer。

D.2 否则: 如果原因码指出呼叫是由于错误清除的, 且上行节点 X 仍被通告为由一个或多个边界节点发起的上行链路的目的地, 且另一 SVCC 没有向 X 开放, ThisLGN 比上行节点 X 的节点 ID 大, 那么立即尝试重新建立到上行节点 X 的 SVCC 且跳到步骤 A.6; 否则无动作。

如果 RetryLGNSVCTimer 超时, 则:

E.1 如果上行节点 X 仍被通告为由一个或多个边界节点发起的上行链路的目的地, 且如果 X 与 ThisLGN 在同一对等组中, 目前没有开放到 X 的 SVCC, 同时 ThisLGN 有较大的节点 ID, 那么重试建立到 X 的 SVCC 并跳到步骤 A.6。

E.2 否则: 无动作。

注: 在相邻 LGN 间的 SVCC 的故障可能是由两个中的任一相邻对等组内的一个单元的故障引起的。这并不意味着对等组间的连通性的任何损失。这样, 如果 SVCC 失败, 在逻辑组节点间的相应的较高等级的链路的通告上不会产生立即的效果。而是必须在改变被通告的链路状态之前尝试重新建立 SVCC。

详情可见 5.6.3 节。

5.6 Hello 协议

PNNI Hello 分组包协议在相邻的节点之间交换, 该交换在 RCC 上实现。

为了发现和证实相邻节点的标识和确定到这些节点的链路, Hello 在全部物理链路和 VPC 上传送。

此外, Hello 也在作为 RCC 的所有 SVCC 上传送, 这些 Hello 程序用于证实相邻节点的标识、至该相邻节点的水平链路的状态和确定 RCC 的状态。

有关 Hello 的数据在 Hello FSM 的相应章节中, 其描述见本标准的 5.6.2.1 节的 1)条、5.6.3.3 节的 1)条和 5.6.3.2 节的 1)条。

5.6.1 PNNI 版本的协商

通常 PNNI 支持多种协议的版本。在协议消息中, 版本字段是未规定的整数字段。节点由支持的“最新”和“最旧”版本表示可以支持版本的范围, 在该范围内的全部版本由该通告者支持。一个附加的字段“协议版本”包括产生分组包的版本。一经发生了协商, 就将反映出协商的结果。该协商由本地支持

的最高版本和相邻节点支持的最高版本中的较低版本组成。这种协商仅作为 Hello FSM 实施的一部分。如果该版本本地不支持，应不启动相邻行，并通告管理系统。版本字段支持的“最旧”版本可以由另一侧采用以检出不兼容性，因此，也可以向管理系统通告这种问题。

5.6.2 Hello 在物理链路和 VPC 中的传送

5.6.2.1 Hello 的状态机

在两个最低级相邻节点之间，每个物理链路或 VPC 具有它自身的 Hello 协议实例。实例由 Hello 数据结构 Hello 状态机组成。

对于具有并行物理链路和/或 VPC 的最低级的相邻节点来说，它们之间存在 Hello 协议的多重实例。然而，为了数据库的同步和 PTSE 的泛播，仅存在一个相邻对等数据结构和相关的相邻对等状态机实例。为了描述在多重 Hello 对话和单个相邻对等对话之间的交互（对于数据库同步和泛播程序），对相邻对等状态机和它的事件 AddPort 和 DropPort 提供参考。事件 AddPort 表示到相邻的对等节点新的组内链路已经启动（即已经进入 Hello 状态 2-Way Inside）。事件 DropPort 表示到相邻对等的链路已经停止（即已经离开 Hello 状态 2-Way Inside）。Hello 状态机的描述包括何时必须触发事件 AddPort 和 DropPort 的指示，这样描述了多重 Hello 对话和相应的单个相邻对等对话之间的相互作用。

1) Hello 的数据结构

对每个节点的物理端口和逻辑接口存在一个 Hello 数据结构。对每个 VP 连接来说，该节点是一个端点）每个 Hello 数据结构由下列项目组成：

状态：一条链路的操作状态，更详细的说明在第 5.6.2.1 节的 2) 条。

端口 ID：由节点指定分配的一个号码，标识物理端口和虚拟通道的连接。端口 ID 由 Hello 数据结构描述。

远端节点 ID：链路另一端相邻节点的对等组(peer group)ID，远端对等组 ID 是当 Hello 由相邻节点收到时获得。

远端对等组 ID：链路另一端相邻节点的远端对等组 ID，是从相邻节点收到的 Hello 中得到。

远端口 ID：该链路的相邻端口 ID 是从相邻节点收到的 Hello 中得到。当远端口 ID 未知时，该值为 0。

HelloInterval：在没有事件触发 Hello 时，节点在链路上发送两个 Hello 之间的时间间隔，单位为秒。

Hello 定时器：每间隔 HelloInterval 启动定时器一次，每当定时器启动时，节点在链路上发送一个 Hello。

InactivityFactor：该系数乘以 HelloInterval 的时间内，如果相邻节点的 Hello 未被收到，则节点认为链路退出工作。

不工作定时器：该定时器的初始值为 InactivityFactor 乘以 HelloInterval，该定时器超时表示未收到从相邻节点来的最新的 Hello。

版本：表示目前正在与相邻节点通信使用 Hello 协议的版本。如果无法得到可认可的版本号，该字段置为 0。

对于外部链路（即至其他对等组节点的链路）应包括下列附加的项目：

——发送的 ULIA 的序号：最新发送的 ULIA 的序号

——接收的 ULIA 序号：最新收到的 ULIA 的序号，或最后收到的 Hello 未包括 ULIA 时清除 ULIA 的值。

——接收的节点层次序号：最新收到的节点层次表的序号，如果最后收到的 Hello 中未包括节点层次表，则清除序号。

——上行节点 ID：上行节点的节点 ID，上行节点 ID 是从相邻节点收到的 Hello 的足够完整节点层次表中取得的。

——公共对等组 ID：边界节点和它的外部相邻节点的公共对等组的对等组 ID，公共对等组 ID 是从相邻节点收到的 Hello 的足够完整节点层次表中取得的。

——上行节点 ATM 端系统地址：上行节点 ATM 端系统的地址，上行节点 ATM 端系统的地址是从相邻节点收到的 Hello 的足够完整节点层次表中取得的。

——派生的链路汇集令牌：该链路的链路汇集令牌的运算值。

——配置的链路汇集令牌：该节点的对该链路的配置链路汇集令牌。

——远端链路汇集令牌：收到在链路另一端相邻节点来的 Hello 中的链路汇集令牌。

2) Hello 状态

图 16 给出了 Hello 协议的状态转移图。

方框是表示状态，箭头线上的文字表示导致状态变化的事件。这些事件由 5.6.2.1 节的 3) 条说明。有关更详细的状态改变说明和伴随的动作见 5.6.2.1 节的 4) 条。

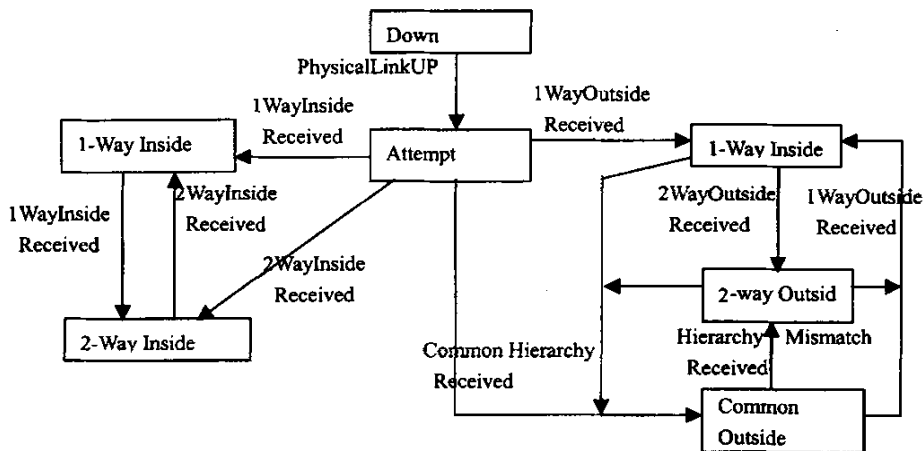


图 16 Hello 状态的变化

除了图上画出的状态转移，还有如下状态转移：

事件 **LinkDown** 总是强制进入 **Down** 状态；

事件 **InactivityTimer** 总是强制进入 **Attempt** 状态；

事件 **HelloMismatchReceived** 总是强制进入 **Attempt** 状态。

Down:

Hello FSM 的初始状态，低层协议表示链路不可用时进入该状态，链路上将不会发送和接收 PNNI 路由分组包。

Attempt:

该状态表示从相邻节点未能收到 Hello 或收到不匹配信息的 Hello。在该状态下，周期性发送 Hello 与相邻节点联系。

1-Way inside:

从相邻节点接收最新的 Hello 表明两个节点是同一对等组的成员，并且相邻节点 Hello 中的远端节点 ID 和远端端口 ID 是置 0。

2-Way Inside:

从相邻节点接收最新的 Hello 表明两个节点是同一对等组的成员，并且包括了正确的远端节点 ID 和远端端口 ID 字段。当处于本状态时，它表示两个节点在该链路上的双向通信已经实现。数据库归纳分组包、PTSE 请求分组包、PTSP 和 PTSE 证实分组包在 **2-Way Inside** 状态时才能传递。对于物理链路和 VPC，只有链路处于 **2-Way Inside** 状态时，这个节点才能够用 PTSE 通告该链路作为水平链路的 PTSE 中。

1-Way Outside:

该状态表示最新的 Hello 已经从相邻的节点收到，并且相邻节点属于不同对等组，但在相邻的 Hello 中的远端节点 ID 和远端端口 ID 置 0。在该状态和在 **2WayOutside** 状态，该节点寻找一包含本节点和相

邻节点的公共对等组。

2-WayOutside:

该状态表示最新的 Hello 已经从相邻的节点收到，表示相邻节点属于不同的对等组，并且包括了正确的远端节点 ID 和远端口 ID 字段，节点层次表未包括任何公共对等组。在该状态和在 **1WayOutside** 状态，该节点寻找一包含本节点和相邻节点的公共对等组。

Common Outside:

该状态表示已经找到了选路层次中的公共等级。除此以外，两节点间已进行了双向通信。处于该状态的链路能够作为上行链路在 PTSE 中向上行节点通告。

3) 引起 Hello 状态变化的事件

有几类与 Hello 协议操作有关的事件可能引起 Hello 协议的状态变化，这些事件在图 16 中的箭头上表示，对于状态变化和在产生一个事件后采取的动作的详细说明见 5.6.2.1 节的 4)条。

Link Up:

低层协议已经表示出链路在运行。

1WayInsidReceived:

从相邻节点收到的 Hello 中，相邻节点的对等组 ID 和本节点的对等组 ID 相匹配，并且远端节点 ID 和远端口 ID 置 0。此外，如果在 Hello 数据结构中的版本、远端节点 ID、远端口 ID 和远端对等组 ID 字段不为 0，它们必须分别与收到的 Hello 中的协议版本、源节点 ID、端口 ID、对等组 ID 相匹配。

2WayInside Received:

从相邻节点收到的 Hello 中，节点 ID 和远端口 ID 能正确地标识本节点的节点 ID 和端口 ID。并且相邻节点的对等组 ID 与本节点的对等组 ID 相匹配。此外，如果在 Hello 数据结构中的版本、远端节点 ID、远端口 ID 和远端对等组 ID 字段不为 0，它们必须分别与收到的 Hello 中协议版本、源节点 ID、端口 ID、对等组 ID 相匹配。

1Way OutsideReceived:

从相邻节点收到的 Hello 中，相邻节点的对等组 ID 与本节点的对等组 ID 不匹配，且所具有的远端节点 ID 和远端口 ID 字段都置为 0。此外，如果在 Hello 数据结构中的版本、远端节点 ID、远端口 ID 和远端对等组 ID 字段不为 0，它们必须分别与收到的 Hello 中协议版本、源节点 ID、端口 ID、对等组 ID 相匹配。

2WayOutside Received:

从相邻节点收到的 Hello 中包括：

——远端节点 ID 和远端口 ID 字段，正确反映出该节点的节点 ID 和该链路的端口 ID。

——对等组 ID，与本节点对等组 ID 不匹配。

——公共对等组 ID、上行节点 ID 和上行节点 ATM 端系统地址，且 Hello 的数据结构中设置为 0；且下列条件之一为真。

a) 节点分组包表、汇集令牌或 ULIA 不存在；

b) 节点层次表存在，但不包含公共对等组。

此外，如果版本、远端节点 ID、远端口 ID、远端对等组 ID 的字段都不为 0，它们必须与收到的 Hello 中的协议版本、源节点 ID、端口 ID 和对等组 ID 相匹配。

Common Hierarchy Received:

从相邻节点收到的 Hello 中包括：

——远端节点 ID 和远端口 ID，正确地反映该节点的节点 ID 和该链路的端口 ID。

——对等组 ID，与本节点的对等组 ID 不匹配。

——ULIA。

——汇集令牌。

——节点层次表，包括以下内容：

a) 序号，与收到节点层次序号相匹配，且公共对等组 ID 已经在 Hello 数据结构中设置；

b) 公共对等组；且 Hello 数据结构中的公共对等组 ID、上行节点 ID 和上行节点的 ATM 端系统地址均设置为 0，或者是这些字段分别与收到的节点层次表中的最低公共对等组、与公共对等组 ID 有关的节点 ID 和 ATM 端系统地址相匹配。

此外，如果在 Hello 数据结构中的版本、远端节点 ID、远端口 ID 和远端对等组 ID 的字段都不为 0，它们必须分别与收到的 Hello 中的协议版本、源节点 ID、端口 ID 和对等组 ID 相匹配。

HelloMismatchReceived:

从相邻节点收到的 Hello 中，至少版本、始发节点 ID、对等组 ID 或端口 ID 之一与 Hello 数据结构中的协议版本、远端节点 ID、远端对等组 ID 或者远端口 ID 不同。

另一种情况是收到的 Hello 中，远端节点 ID 和/或远端口 ID 与本节点的节点 ID 或与接收链路的端口 ID 不同，并且都不为 0。在 Attempt 状态，仅考虑第二种情况。优先考虑 HelloMismatchReceived 的事件。

HierarchyMismatchReceived:

从相邻节点收到的 Hello 中包括：

——远端节点 ID 和远端口 ID 字段，正确地反映这个节点 ID 和这条链路的端口 ID；

——对等组 ID 与本节点的对等组 ID 不匹配；

——不满足 HelloMismatchReceived 的标准；

——或者满足下列条件之一

a) 无节点层次表；

b) 未找到 ULIA；

c) 未找到汇集令牌；

d) 已经收到新的节点层次表序号，且满足下列条件之一：

● 对等组级顺序值不是严格的递减；

● 未找到公共对等组；

● 找到的最低的公共对等组 ID 与 Hello 数据结构中的公共对等组 ID 不同；

● 包括公共对等组 ID，但较高级的节点 ID 和/或较高级的 ATM 端系统地址与 Hello 数据结构中的上行节点 ID 或上行节点 ATM 端系统地址匹配。

HelloTimerExpired:

Hello 定时器已经到时。

InactivityTimerExpired:

不工作定时器已经超时。这意味着目前从相邻节点未收到最新的 Hello。

LinkDown:

较低层的协议表示这个链路是不工作。

4) Hello 状态机的描述

有限状态机是一个两维表，如表 14 所示。表的顶端表示状态，左侧为事件，交叉部分表示状态的转换和待采取的动作。例如对于“上行链路”和“退出”的交叉部分“Hp1，尝试”。“尝试”是新的状态，“Hp1”是待采取的动作。动作的定义在表的后面说明。

表 14 Hello 协议的有限状态机

Event	Down	Attempt	1-Way Inside	2-Way Inside	1-Way Outside	2-Way Outside	Common Outside
Link Up	Hp1 Attempt	Hp0 Attempt	Hp0 1-WayIn	Hp0 2-WayIn	Hp0 1-WayOut	Hp0 2-WayOut	Hp0 Common
1-Way Inside Received	Hp0 Down	Hp2 1WayIn	Hp12 1-WayIn	Hp10 1-WayIn	FSM-Err	FSM-Err	FSM-Err

续表14

2-Way Inside Received	FSM-Err	Hp3 2WayIn	Hp4 2WayIn	Hp12 2WayIn	FSM-Err	FSM-Err	FSM-Err
1-Way Outside Received	Hp0 Down	Hp5 1WayOut (注 1)	FSM-Err	FSM-Err	Hp12 1-WayOut	Hp13 1-WayOut	Hp14 1-WayOut
12Way Outside Received	FSM-Err	Hp5 2WayOut (注 1)	FSM-Err	FSM-Err	Hp12 2-WayOut	Hp12 2-WayOut	FSM-Err
Common Hierarchy Received	FSM-Err	Hp6 Common (注 1)	FSM-Err	FSM-Err	Hp7 Common	Hp7 Common	Hp20 Common
Hello Mismatch Received	FSM-Err	Hp0 Attempt	Hp8 Attempt	Hp16 Attempt	Hp8 Attempt	Hp8 Attempt	Hp17 Attempt
Hierarchy Mismatch Received	FSM-Err	FSM-Err	FSM-Err	FSM-Err	FSM-Err	FSM-Err	Hp11 2-WayOut
Hello Timer Expired	FSM-Err	Hp15 Attempt	Hp15 1WayIn	Hp15 2WayIn	Hp15 1WayOut	Hp15 2WayOut	Hp15 Common
Inactivity TimerExpired	FSM-Err	FSM-Err	Hp8 Attempt	Hp16 Attempt	Hp8 Attempt	Hp8 Attempt	Hp17 Attempt
Link Down	Hp0 Down	Hp9 Down	Hp9 Down	Hp18 Down	Hp9 Down	Hp9 Down	Hp19 Down

注 1: 不能成为边界节点的节点必须采用 Hp0 并维持在尝试状态。

FSM-Err: 表示一个内部的实施差错。

Hp0

动作: 无动作。

Hp1

动作: 在链路上发送一个 Hello 并启动 Hello 定时器, 使之能周期性发送 Hello。

Hp2

动作: 启动链路的不工作定时器。将 Hello 数据结构中的远端节点 ID、远端对等组 ID 和远端端口 ID 设置为收到的 Hello 中所列的节点 ID、对等组 ID 和端口 ID。计算收到的支持的最新版本和本地支持的最新版本中的低者, 并记录此版本的版本号, 向相邻节点发送一个 Hello 并重新启动 Hello 定时器。

Hp3

动作: 启动链路的不工作定时器。将 Hello 数据结构中的远端节点 ID、远端对等组 ID 和远端端口 ID 设置为收到的 Hello 中所列的节点 ID、对等组 ID 和端口 ID。计算收到的

支持的最新版本和本地支持的最新版本中的低者，并记录此版本的版本号码，向相邻节点发送一个 Hello，并重新启动 Hello 定时器。调用具有 AddPort 事件的相邻对等状态机。

Hp4

动作：再启动不工作定时器。调用具有 AddPort 事件的相邻对等状态机。

Hp5

动作：启动链路的不工作定时器。将 Hello 数据结构中的远端节点 ID、远端对等组 ID 和远端端口 ID 设置为收到的 Hello 中所列的节点 ID、对等组 ID 和端口 ID。计算收到的支持的最新版本和本地支持的最新版本中的低者，并记录此版本的版本号码。向相邻节点发送 Hello，它包括节点层次表和描述链路所有出局业务类别量度信息组，并启动 Hello 定时器。

Hp6

动作：启动链路的不工作定时器。将 Hello 数据结构中的远端节点 ID、远端对等组 ID 和远端端口 ID 设置为收到的 Hello 中所列的节点 ID、对等组 ID 和端口 ID。从相邻点收到的 Hello 中找出节点层次表中的最低公共对等组，并且用接收的节点层次表中的对应信息设置上行节点 ID、公共对等组 ID 和上行节点的 ATM 端系统地址。计算收到的支持的最新版本和本地支持的最新版本中的低者，并记录此版本的版本号码。向相邻节点发送一个 Hello，它包括节点层次表和描述链路所有出局 ULIA，并启动 Hello 定时器。该节点可以在 PTSE 中把该链路作为上行链路通告上行节点。

Hp7

动作：启动链路的不工作定时器，从相邻点收到的 Hello 中找出节点层次表中的最低公共对等组，并且用接收的节点层次表中的对应信息设置上行节点 ID、公共对等组 ID 和上行节点的 ATM 端系统地址。该节点可以在 PTSE 中把该链路作为上行链路通告上行节点。

Hp8

动作：停止不工作定时器，清除在 Hello 数据结构中的版本、远端节点 ID、远端对等组 ID、远端端口 ID、上行节点 ID、公共对等组 ID、上行节点 ATM 端系统地址、收到的 ULIA 序号和节点层次表序号。向相邻点发送一个 Hello，并启动 Hello 定时器。

Hp9

动作：停止不工作定时器，清除在 Hello 数据结构中的版本、远端节点 ID、远端对等组 ID、远端端口 ID、上行节点 ID、公共对等组 ID、上行节点 ATM 端系统地址、收到的 ULIA 序号和节点层次表序号。

Hp10

动作：启动不工作定时器。向相邻点发送一个 Hello，并启动 Hello 定时器。调用具有 DropPort 事件的相邻对等状态机。

Hp11

动作：启动链路的不工作定时器。清除在 Hello 数据结构中的上行节点 ID、公共对等组 ID 和上行节点 ATM 端系统地址。必须从由该节点生成作为上行链路通告的任何 PTSE 中移去该链路。

Hp12

动作：启动该链路的不工作定时器。

Hp13

动作：启动链路的不工作定时器。清除在 Hello 数据结构中收到的 ULIA 序号和节点层次表序号。向相邻节点发送一个 Hello，它包括节点层次表和描述链路所有出局 ULIA，并

启动 Hello 定时器。

Hp14

动作：停止不工作定时器，清除在 Hello 数据结构中的上行节点 ID、公共对等组 ID、上行节点 ATM 端系统地址字段、收到的 ULIA 序号和节点层次表序号。向相邻节点发送一个 Hello，它包括节点层次表和描述链路所有出局 ULIA，并启动 Hello 定时器。必须从由该节点生成作为上行链路通告的任何 PTSE 中移去该链路。

Hp15

动作：启动 Hello 定时器。

Hp16

动作：停止不工作定时器，清除在 Hello 数据结构中的版本、远端节点 ID、远端对等组 ID、远端端口 ID、上行节点 ID、公共对等组 ID、上行节点 ATM 端系统地址、收到的 ULIA 序号和节点层次表序号。向相邻点发送一个 Hello，并启动 Hello 定时器。调用具有 DropPort 事件的相邻对等状态机。

Hp17

动作：停止不工作定时器，清除在 Hello 数据结构中的版本、远端节点 ID、远端对等组 ID、远端端口 ID、上行节点 ID、公共对等组 ID、上行节点 ATM 端系统地址、收到的 ULIA 序号和节点层次表序号。向相邻点发送一个 Hello，并启动 Hello 定时器。必须从由该节点生成作为上行链路通告的任何 PTSE 中移去该链路。

Hp18

动作：停止 Hello 和不工作定时器，清除在 Hello 数据结构中的版本、远端节点 ID、远端对等组 ID、远端端口 ID、上行节点 ID、公共对等组 ID、上行节点 ATM 端系统地址、收到的 ULIA 序号和节点层次表序号。调用具有 DropPort 事件的相邻对等状态机。

Hp19

动作：停止 Hello 和不工作定时器，清除在 Hello 数据结构中的版本、远端节点 ID、远端对等组 ID、远端端口 ID、上行节点 ID、公共对等组 ID、上行节点 ATM 端系统地址、收到的 ULIA 序号和节点层次表序号。必须从由该节点生成作为上行链路通告的任何 PTSE 中移去该链路。

Hp20

动作：启动不工作定时器。如果 ULIA 序号与在 Hello 数据结构中收到的 ULIA 序号不匹配，则对应于该链路上行链路的通告必须用收到的 ULIA 更新，并重新发送通告。

5.6.2.2 发送 Hello

为了建立已经获得的双向通信，在每条链路上发送 Hello。当 Hello 数据结构的版本字段是 0，Hello 按照本实施支持的最新版本的编码发送。反之则采用记录的版本。在任何一种情况下，所采用版本是在 Hello 分组包的协议版本字段。

在不是 Down 状态的所有情况下，Hello 每隔 HelloInterval 周期发送一次。此外，在下列情况下发送事件触发的 Hello：

— 不包括下列的状态变化：

1-Way Inside 到 2-Way Inside;

1-Way Outside 到 2-Way Outside;

1-Way Outside 到 Common Outside;

2-Way Outside 到 Common Outside;

Common Outside 到 2-Way Outside。

——在外部链路上 ULIA 的重大变化。

——外部链路上节点层次表的变化。

——外部链路汇集令牌的变化。

在连续发送 Hello 之间的周期不能小于 MinHelloInterval。Hold-Down 定时器是防止用过高速率发送 Hello。由于 Hold-Down 定时器存在，就延缓了多重事件触发 Hello 的发送，当 Hold-Down 定时器超时后，只有一个包含最新信息的 Hello 被发送。每当事件触发的 Hello 发送时，则启动 Hello 定时器，保证下一个 Hello 在 HelloInterval 时间后按程序发送。

在 **Attempt** 状态时，Hello 必须把远端节点 ID 和远端端口 ID 设为 0 并存储在 Hello 数据结构中。当 Hello 状态在 **1-Way Inside**、**2-Way Inside**、**1Way Outside**、**2-Way Outside**、**Common Outside**，Hello 必须把从 Hello 中收到的相邻节点 ID 和端口 ID 作为本节点的远端节点 ID 和端口 ID 并存储在 Hello 数据结构中。

在状态 **1-Way Outside**、**2-Way Outside**、**Common Outside** 下，从 PGL 中得到的高级绑定信息中的较高级节点 ID、对等组 ID 和 LGN 的 ATM 端系统地址作为节点层次表包含在 Hello 中。如果有多个节点要成为对等组领导，则从中选择一个作为对等组领导进行通告。对于较高等级的对等组，如果有多个节点要成为 PGL，则由那个对等组的祖先 LGN 选择一个 PGL 进行通告信息。

较高等级的数量和内容发生变化时，节点层次表的序号必须加 1，并且必须发送事件触发的 Hello。此外，当较低等级的节点 ID、对等组 ID 或 ATM 地址发生改变时，节点层次表的序号必须加 1，并必须发送事件触发的 Hello。向相邻节点发送的节点层次表的第一个事例时，序号值必须大于 0。

每次发送包括节点层次表的一个 Hello 时，该表必须包含全部已知的较高级情况，如果不知道层次中的较高级，则需要包含一个空的节点层次表。

在状态 **1-Way Outside**、**2-Way Outside**、**Common Outside** 下，Hello 中应当包含具有出局资源可用信息组的 ULIA IG。

在状态 **1-Way Outside**、**2-Way Outside**、**Common Outside** 下，Hello 中应当包含用于标识上行链路的配置的链路汇集令牌。

——在外部链路上生成并发送的 ULIA。

对于某些 Hello 状态，边界节点在外部链路上发送 Hello 分组包中必须包括 ULIA IG。ULIA 本身不规定任何的链路状态信息，但用于捆绑其他包含链路状态信息的 IG。这种机制的灵活性允许一个边界节点指示由相邻点通告的准确的链路状态信息 (IG)，而相邻点要求理解或翻译单个的 IG。使用这种方式，可以使部分网络更新，并在全网内有正确的链路状态信息。

ULIA 中内容的重大变化是通过改变 ULIA 序号来表示的。当 ULIA 的内容发生重大变化时，始发者只增加 ULIA 序号。

因为边界节点不解释收到的 Hello 分组包中的 ULIA 内的单独的 IG。任何对 ULIA 序号的改变表明是一个重大的变化，因此必须触发一个对相应上行链路 PTSE 的更新。为响应这个触发的更新，重新产生的上行链路 PTSE 实例应当受 PTSE Hold-Down 定时器的控制。

除非包含在 ULIA 中的一个或多个 IG 已经发生重大变化，否则边界节点决不在发送 Hello 中产生 ULIA 的新的序号。如果上次发送的 ULIA 未发生重大变化，则 ULIA 仍使用先前的序号。定义的一组 IG 的非重大的变化，不会造成处理开销的变化，因此不必携带一个新的序号，如果新的 ULIA 是为了响应重大变化而产生的，则序号必须增加。

捆绑在 ULIA 中的 IG 发生的重大的变化事件触发生成一组新的 ULIA，并发送到相应的外部相邻节点。当新的 Hello 分组包发送时，它将包括一个新的 ULIA 和新序号。为响应对一个或多个捆绑在 ULIA 中的 IG 重大变化而生成新的 ULIA，无论是否发生重大的变化，边界节点对所有捆绑的 IG 插入最近的和精确链路状态信息。用这种方式，任何一个 IG 的重大的变化，都将造成 ULIA 中捆绑的 IG 的最新信息的通告。

5.6.2.3 接收 Hello

如果收到一个不支持的协议版本的 Hello，则舍弃该 Hello 分组包。如果在 Hello 分组包中表示的版本范围与这个节点支持的版本范围不重叠，则记录一个管理通告，参见 5.1.3 节。

如果收到的 Hello 分组包的必备标签中带有高级组未知 TLV，则舍弃该 Hello 分组包。此外，如果在收到的 Hello 分组包中的 HelloInterval 或端口 ID 置 0，则该分组包被认为是无效并被舍弃。

相邻节点是通过包含在 Hello 头中的节点 ID 来标识的。如果在 Hello 数据结构中的远端节点的 ID、远端对等组 ID 和远端口 ID 未规定，则它们必须分别设置成为收到的 Hello 中的源节点 ID、对等组 ID 和端口 ID。如果在 Hello 数据结构中的版本字段为 0，则计算收到的支持的最新版本和本地支持的最新版本中的低者，并记录此版本的版本号码。

如果收到的 Hello 中包含一个新的节点层次表的事例，则在节点层次表中查所有节点的公共的最低级对等组。尽管节点层次表中没有明确的指示出相邻节点 ID、对等组 ID 和 ATM 端系统地址，但它们必须从逻辑上是节点层次表的最低级成份。

如果一个节点发现与其所在的对等组 ID 匹配，则不存在上行链路（见 5.5.5 节）。这个节点就成为 LGN 的临近边界节点，LGN 是在公共对等组中表示相邻边界节点。将建立一个 SVCC，对一个和多个水平链路的通告详细规定在 5.5.6 节。

收到的汇集令牌是按 5.10.3.1 节处理。

每个收到的 Hello 根据下列事件执行 Hello 状态机，

1-WayInsideReceived;

2-WayInsideReceived;

1-WayOutsideReceived;

2-WayOutsideReceived;

CommonHierarchyReceived;

HelloMismatchReceived;

HierarchyMismatchReceived.

——外部链路上收到 Hello 中 ULIA 的处理。

当边界节点从外部链路上收到 Hello 分组包时，它必须确定是否（再）产生相应的上行链路 PTSE。如果最新收到 ULIA 的序号与先前的 ULIA 的序号相同，则上行链路的 PSTE 不需要进行更新。在接收端，序号比较用来确定包含链路状态信息的 IG 是否发生重大变化。当与先前收到的 ULIA 序号比较时，ULIA 序号字段的改变则构成重大的变化，将引起接收者重新生成相应的上行链路 PTSE。

5.6.3 LGN 的 Hello 协议

在 LGN 之间的 Hello 协议有两个作用。一个是用于监视用作 PNNI RCC 的 SVCC 的状态，第二个用途是用于水平链路的通信和协商。

SVCC 可以穿过多个链路。这些链路之一的故障，就会导致 SVCC 故障。这种故障是两个 LGN 之间连通性的变化。对等组内链路故障，不会造成连通性的变化。因此，在 LGN 之间的 RCC 的状态维护和 LGN 之间链路的状态维护必须是独立的功能，以保证层次的稳定性。当两个功能放在同一个 Hello 消息时，应当暂时把这两个功能分开，并分别设置其定时器。RCC 监视功能的故障将引起 SVCC 重新建立。定时器到时后，这个事件才会引起与逻辑链路有关的状态机的改变。

5.6.3.1 基于 SVCC 的 RCC Hello 协议

用于验证两个 LGN 之间通信的协议与在最低级相邻节点之间的协议很类似，并使用相同的分组包类型。然而与最低级相邻节点不同，在它们之间只有一个单个 PNNI RCC。这个基于 SVCC 的 RCC 是用于在 LGN 相邻节点之间交换的所有 PNNI 的选路分组包，包括 PTSP 和用以维持数据库同步和 Hello 同步的分组包。用于监视 SVCC 状态的 Hello 协议触发了相邻对等状态机中 AddPort 和 DropPort 事件，用来控制在 LGN 之间的数据库同步。当 SVCC 的 Hello 状态机到达 2-Way Inside 状态时，触发相邻对等状态机中的 AddPort 事件；当 SVCC 的 Hello 状态机离开到 2-Way Inside 状态时，触发相邻对等状态机中的 DropPort 事件。

一旦 SVCC 由信令协议建立，就启动一个 Hello 协议实例。该协议基本上与在最低级相邻节点运行的协议相同，只有如下的修改。

——Hello 消息的端口 ID 字段总使用 0xFFFFFFFF。在 LGN 之间的 SVCC 不能给它们分配端口 ID。如果收到的 Hello 中端口 ID 值不是 0xFFFFFFFF，则触发事件 **HelloMismatchReceived**。

——基于 SVCC 的 RCC 总是在一个对等组内。LGN 之间的 SVCC 不会触发事件 **1-WayOutsideReceived**、**2-WayOutsideReceived**、**CommonHierarchyReceived** 和 **HierarchyMismatchReceived**。如果 Hello 中收到的对等组 ID 与该节点的对等组 ID 不同，则触发 **HelloMismatchReceived**。

——收到的 Hello 中的节点 ID 必须等于对应的上行链路 PTSE 的值。如果不相等，则触发事件 **HelloMismatchReceived**。

对于基于 SVCC 的 RCC 的主叫 LGN，在启动 SVCC 之前必须接收到上行链路 PSTE。对于被叫 LGN，在从相邻节点来的呼叫建立到达和上行链路 PSTE 之间存在竞争条件，如果被叫 LGN 接收一个来自还未被标识为相邻节点的 SETUP，被叫用户 LGN 必须接受此呼叫，但忽略任何 Hello 直到接收到表明该节点为相邻点的上行链路。上行链路 PTSE 和基于 SVCC 的 RCC 的绑定是基于 SVCC 的 RCC 上接收的 Hello 的节点 ID 和上行链路 PTSE 中上行节点 ID。

——在状态 **Down**、**Attempt**、**1-way Inside** 和 **1-way Outside** 下，设置定时器 **SVCIntegrityTimer**。如果定时器超时，基于 SVCC 的 RCC 被认为不工作并释放 SVCC。

——对于被叫 LGN，**HelloMismatchReceived** 导致返回 **Attempt** 状态；对于主叫 LGN，**HelloMismatchReceived** 导致释放并重新建立 SVCC，这种情况也将记录至网管。

——低层指示（ATM、PHY、信令）的 SVCC 故障，当作 **LinkDown** 事件，并重新建立 SVCC。

1) 基于 SVCC 的 RCC Hello 数据结构。

到相邻点的基于 SVCC 的 RCC 只有一个单独的 Hello 数据结构。每个数据结构的实例由以下项目组成。

状态：链路的工作状态。

远端节点 ID：链路另一端相邻节点的节点 ID。

远端端口 ID：链路的相邻节点的端口 ID。当远端端口 ID 还未收到时，它的值为 0。

HelloInterval：在没有事件触发的 Hello 的情况下，节点发送到链路上的 Hello 之间的时间。

Hello 定时器：该定时器每隔 HelloInterval 时间超时，超时后节点发送 Hello 到链路上。

InactivityFactor：在该系数乘以 HelloInterval 的时间内，如果相邻节点的 Hello 未被收到，则节点认为链路退出工作。

不工作定时器：该定时器的初始值为 InactivityFactor 乘以 HelloInterval，该定时器超时表示未收到从相邻节点来的最新的 Hello。

SVCIntegrity 定时器：该定时器用于确定 SVCC 不可用并释放之。

SVCIntegrityTime：在释放基于 SVCC 的 RCC 之前，节点等待 RCC 进入 2-Way Inside 的时间。

HorizontalLinkInactivityTime：如果节点没有收到 LGN 水平链路 IG，就继续通告水平链路的时间。

HorizontalLinkInactivity 定时器：表示没有从相邻节点收到最新的 LGN 水平链路 IG。

版本：与相邻节点通信所采用的 Hello 协议的版本。如果没有可接受的版本号，这个字段置 0。

2) SVCIntegrity 定时器。

LGN 间基于 SVCC 的 RCC 是由较高节点 ID 的 LGN 建立。在主叫节点如下情况时设置该定时器。

——SVCC 启始建立时。

——每当状态机进入 **Attempt** 或 **1-Way Inside**、**1-Way Outside** 状态且定时器还未工作时。

该定时器在 **2-Way Inside**、**2-Way Outside** 和 **Down** 状态下停止。定时器到时使得返回到 **Down** 状态。在进入 **Down** 状态时，SVCC 释放并进入重新建立 SVCC 的正常程序。SVCIntegrity 定时器的值等于 SVCIntegrityTime 的取值。

在被叫节点，SVCIntegrity 定时器当收到一个来自 LGN 相邻节点的建立消息或每当状态机进入 **Attempt** 状态或处于 **1-Way Inside**、**1-Way Outside** 状态且定时器未工作时被设置。定时器在 **2-Way Inside**、**2-Way Outside** 状态时释放。定时器到时使得返回到 **Down** 状态并释放 SVCC。SVCIntegrity 定

时器的值等于 **SVCIntegrityTime** 的取值。

3) 相邻节点关系的丢失。

如果相邻节点关系终止,即描述到 **LGN** 相邻节点的上行链路的全部 **PTSE** 都被删除,则调用下列的程序。如果相邻节点的关系确实被终止,则 **SVCC** 将由较低层释放。由于较低层对等组分区(未引起在这一层的分区)或一个新的节点是被选作 **PGL**,上行链路可能消失。

在这个事件中,该节点返回到 **Attempt** 状态,这将引起 **SVCIntegrity** 定时器启动。

5.6.3.2 LGN 水平链路 Hello 协议

确定在逻辑组节点之间水平链路状态的协议也是基于 **Hello** 协议。所有到一个相邻 **LGN** 的水平链路的状态是根据在基于 **SVCC** 的 **RCC** 上发送的 **Hello** 中的单个 **LGN** 水平链路扩展信息组来确定的。**LGN** 水平链路扩展信息组出现在所有发送到相邻对等 **LGN** 的 **Hello** 中。每次发送一个 **Hello** 到相邻对等 **LGN**,**LGN** 水平链路扩展信息组将包含所有到相邻对等节点的任何非 **Down** 状态的水平链路。对于每一个水平链路来说都应包括汇集令牌、本地端口 **ID** 和远端端口 **ID**。每个不同的汇集令牌的值表示具有独立状态机的不同水平链路,该信息组在所有的发送的 **Hello** 中存在。

只有当基于 **SVCC** 的 **RCC** 的 **Hello** 状态机处于 **2-Way Inside** 状态且相应相邻对等状态机处于 **Full** 状态时,才处理在包含来自相邻对等 **LGN** 的 **Hello** 中的 **LGN** 水平链路扩展信息组。

当且仅当 **LGN** 水平链路的 **Hello** 状态机处于 **2-Way Inside** 和 **2-Way Outside** 状态时,这个节点才产生关于 **LGN** 水平链路的通告 **PTSE**。

当包含新的汇集令牌值的上行链路 **PTSE** 到达时,就分配一个逻辑端口 **ID** 并创建一个 **Down** 状态的状态机。触发 **AddInducingLink** 事件且状态机转移到 **Attempt** 状态。后续的 **LGN** 水平链路扩展信息组中包含该令牌、分配的本地端口 **ID** 和远端端口 **ID** 为 0。当用现有的汇集令牌值作为一个新的导出上行链路,则触发 **AddInducingLink** 事件。

当具有特定汇集令牌值的上行链路从上行链路 **PTSE** 中取消,或者是上行链路 **PTSE** 被删除时,触发 **DropInducingLink** 事件或 **DropLastInducingLink** 事件。对于 **DropLastInducingLink** 事件,从信息组中取消汇集令牌值和相关的端口 **ID**,在收到具有汇集令牌的新 **PTSE** 之前状态机维持在 **Down** 状态。

忽略收到的状态机不存在的 **LGN** 水平链路扩展 **IG** 中的汇集令牌值。

如果收到在一个 **LGN** 水平链路扩展信息组中缺少汇集令牌时,则强制与汇集令牌有关的状态机进入 **Attempt** 状态,为此必须检测信息组中所有期望的令牌值。

在丢失基于 **SVCC** 的 **RCC** 时,除原因值 53(由于 **PGL** 改变的呼叫拆线)的释放情况外,与水平链路相关的状态是由它们自身的状态机控制。这表明在水平链路不工作定时器超时之前,或者在包含这个汇集令牌的最后的上行链路被取消(即发生 **DropLastInducingLink** 事件),每个水平链路要保持工作状态。用 **HorizontalLinkInactivityTime** 启动水平链路的不工作定时器。由于每个 **Hello** 分组包中都报告所有的水平链路,到相邻节点的所有的链路只使用一个 **LGN** 水平链路不工作定时器。

SVCC 由于原因值 # 53(由于 **PGL** 中改变的呼叫拆线)的原因而释放,则水平链路 **FSM** 触发 **BadNeighbor**。

1) **LGN** 水平链路 **Hello** 数据结构。

对每条到相邻节点的水平链路都有一个 **LGN** 水平链路 **Hello** 数据结构。每个 **LGN** 水平链路 **Hello** 数据结构都包括下列项目。

汇集令牌:与 **LGN** 水平链路有关的汇集令牌。

状态:水平链路的工作状态。

端口 **ID**:节点分配的号码以标识这个水平链路 **Hello** 数据结构描述的是哪一条链路。

远端节点 **ID**:水平链路的另一端的相邻节点的节点 **ID**。

远端端口 **ID**:这个链路的相邻节点的端口 **ID**。当从相邻点受到 **LGN** 水平链路的扩展 **IG** 时就可以知道远端端口 **ID**。当不知道远端端口 **ID** 时,它的值必须置为 0。

导出上行链路表:这个水平链路的导出上行链路的表。每个上行链路由子对等组中的边界节点的节

点 ID 和导出上行链路的端口 ID 来标识。

2) LGN 水平链路状态。

本节描述了 LGN 水平链路 Hello FSM 可以达到的状态。

Down:

这个状态指出没有收到包括到相邻对等 LGN 的上行链路、汇集令牌值与 LGN 水平链路 Hello 数据结构指出的汇集令牌值相同的上行链路 PTSE。

Attempt:

这个状态指出虽然至少收到一个包括到相邻对等 LGN 的上行链路、汇集令牌值与 LGN 水平链路 Hello 数据结构指出的汇集令牌值相同的上行链路 PTSE，但不存在来自相邻对等点的适当的证实。

1-Way:

在这个状态，刚从相邻对等节点收到了包括 LGN 水平链路扩展 IG 中的水平链路的入口的 Hello，但远端口 ID 置为 0。

2-Way:

在这个状态，刚从相邻对等节点收到了包括 LGN 水平链路扩展 IG 中的水平链路的入口且包含正确的远端口 ID 的 Hello。到达这个状态就意味着两个相邻对等 LGN 都知道了水平链路的两侧的端口 ID。在这种状态，水平链路在 PTSE 中通告。

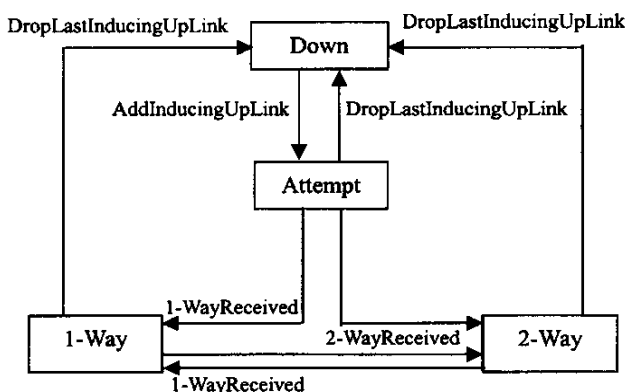


图 17 LGN水平链路Hello FSM

3) LGN 水平链路事件

AddInducingLink:

在下列三种情况中的任意一种情况下，尝试找到给定导出链路的汇集令牌的 LGN 水平链路 Hello 数据结构。如果没有发现，就初始化一个，内容如下：状态置成 Down；汇集令牌置成与导出链路有关的值；选择一个唯一的端口 ID 值且把它的值放进端口 ID 字段中；远端节点 ID 置成对等的节点 ID；远端口 ID 置成空；导出链路表置成空。

a) 包括到相邻对等节点的新的导出上行链路的上行链路 PTSE 已到达。

b) 通告导出上行链路重新建立到边界节点的连通性。注意这样的连通性恢复可能影响几条上行链路且会依次影响几个水平链路的 FSM。

c) 直接连接相邻对等 LGN 的后代节点的外部链路已达到 2-WayOutside 状态，且带有与 LGN 水平链路 Hello 数据结构中的汇集令牌值相同的汇集令牌值。

1-WayReceived:

包括 LGN 水平链路扩展 IG 中的汇集令牌的入口的 Hello 已经收到，Hello 中的远端口 ID 为 0。此外，如果在 LGN 水平链路 Hello 数据结构中的远端口 ID 字段不是 0，它必须与收到的 Hello 中的端口 ID 匹配。

2-WayReceived:

包括 LGN 水平链路扩展 IG 中的汇集令牌的入口的 Hello 已经收到，Hello 中的远端端口 ID 能够正确标识 LGN 水平链路 Hello 数据结构中的端口 ID。除此之外，如果在 LGN 水平链路 Hello 数据结构中的远端端口 ID 字段不是 0，它必须与收到的 Hello 中的端口 ID 匹配。

HelloMismatchRecieved:

已经收到了 Hello，因为以下原因之一：

——在相应的基于 SVCC 的 RCC Hello 协议中 HelloMismatch 事件已产生。

——LGN 水平链路 Hello 数据结构中的汇集令牌没有出现在 LGN 水平链路扩展 IG 中。

——LGN 水平链路 Hello 扩展 IG 中的汇集令牌的入口的远端端口 ID 与 LGN 水平链路 Hello 数据结构中的不同，它没有置成 0。

——LGN 水平链路 Hello 数据结构的远端端口 ID 与收到的在 LGN 水平链路 Hello 扩展 IG 中的汇集令牌的入口的端口 ID 不同。

HorizontalLinkInactivityTimerExpired:

HorizontalLinkInactivity 定时器超时。这意味着当前没有从相邻节点收到 Hello，其中的 LGN 水平链路 Hello 扩展 IG 已进行了处理。

BadNeighbor:

相应的基于 SVCC 的 RCC 已被释放，原因值为 53“由于 PGL 的改变呼叫清除”，或者在相应的相邻对等 FSM 中产生了 DSMismatch 或者 BadPTSERequest。

DropInducingLink:

——有着特定汇集令牌值的导出上行链路已经从它出现的上行链路 PTSE 中删去，且它不是有着特定汇集令牌值的最后一条导出上行链路。

——包含有着特定汇集令牌值的导出上行链路的上行链路 PTSE 已被删除（即超时或刷新），仍然保留至少一个有着特定汇集令牌值的导出上行链路。

——到正在通告导出上行链路的子对等组中的边界节点的连通性已经丢失，且它不是最后一个有着特定汇集令牌值的导出上行链路。注意这样的连通性的丢失可能会影响几条上行链路和水平链路 FSM。

——直接附属于带有与 LGN 水平链路 Hello 数据结构中的相同的汇集令牌值的相邻对等 LGN 的后代节点的外部链路落在了 **2-WayOutside** 状态之外，它不是最后一条有着特定汇集令牌值的导出外部链路。

DropLastInducingLink:

——最后一条有着特定汇集令牌值的导出上行链路已被从它出现的上行链路 PTSE 中删除。

——包含有最后一条有着特定汇集令牌值的导出上行链路的上行链路 PTSE 已被删除（即超时或刷新）。

——到正在通告最后一个导出链路的子对等组的边界节点的连通性丢失。这样的连通性的丢失可能会影响几条上行链路和水平链路 FSM。

——直接附属于带有与 LGN 水平链路 Hello 数据结构中的相同的汇集令牌值的相邻对等 LGN 的后代节点的外部链路落在了 **2-WayOutside** 状态之外。

4) 水平链路的 Hello 状态机的描述

有限状态机采用的是一个二维表，如表 15 所示。状态表的最上一行反映了经过的状态，表的左侧是出现的事件。每个状态与每个事件一起构成表的单元，这些单元反映了将转移到哪个状态和采取什么动作。例如对应于 **AddInduceingLink** 事件和 Down 状态的单元是“Hlhp1, Attempt”；Attempt 是下一个要转移到的状态，“Hlhp1”则是需要采取的动作。在这个状态表后定义了这些动作。

表15 水平链路Hello状态机

事 件	状 态			
	Down	Attempt	1-Way	2-Way
AddInducing-Uplink	Hlhp10, Attempt	Hlhp11, Attempt	Hlhp11, 1Way	Hlhp12, 2Way
1-WayReceived	Hlhp0, Down	Hlhp1, 1Way	Hlhp0, 1Way	Hlhp6, 1Way
2-WayReceived	Hlhp0, Down	Hlhp2, 2Way	Hlhp3, 2Way	Hlhp0, 2Way
HelloMismatchReceived	Hlhp0, Down	Hlhp0, Attempt	Hlhp4, Attempt	Hlhp5, Attempt
HorizontalLink-Inactivity-TimerExpired	Hlhp0, Down	Hlhp0, Attempt	Hlhp4, Attempt	Hlhp5, Attempt
BadNeighbor	Hlhp0, Down	Hlhp0, Attempt	Hlhp4, Attempt	Hlhp5, Attempt
DropInducing-Uplink	FSM_Err	Hlhp13, Attempt	Hlhp13, 1Way	Hlhp14, 2Way
DropLast-InducingUplink	FSM_Err	Hlhp13, Down	Hlhp16, Down	Hlhp15, Down

FSM_Err 不会发生。

Hlhp0

动作：无动作。

Hlhp1

动作：把 LGN 水平链路 Hello 数据结构的远端端口 ID 设置为接收 LGN 水平链路扩展 IG 中用于合并令牌的端口 ID。

Hlhp2

动作：把 LGN 水平链路 Hello 数据结构的远端端口 ID 设置为接收 LGN 水平链路扩展 IG 中用于合并令牌的端口 ID。触发由该节点产生的水平链路 PTSE 新实例中对水平链路的通告。提供处于 **Full** 状态的相邻对等点的状态机。

Hlhp3

动作：触发由该节点产生的水平链路 PTSE 新实例中对水平链路的通告。提供处于 **FULL** 状态的相邻对等点的状态机。

Hlhp4

动作：清除 LGN 水平链路 Hello 数据结构的远端端口 ID。

Hlhp5

动作：清除 LGN 水平链路 Hello 数据结构的远端端口 ID。水平链路必须从由该节点产生并且已经从通告的 PTSE 中删除。

Hlhp6

动作：水平链路必须从由该节点产生并且从已经通告的 PTSE 中删除。

Hlhp10

动作：在导出的上行链路表中增加导出的上行链路（导出的上行链路通过在子对等组中的边界节点的节点 ID 和端口 ID 来标识）。

Hlhp11

动作：在 LGN 的水平链路 Hello 数据结构的导出的上行链路表中增加导出的上行链路（导出的上行链路通过在子对等组中的边界节点的节点 ID 和端口 ID 来标识）。

Hlhp12

动作：在 LGN 的水平链路 Hello 数据结构的导出的上行链路表中增加导出的上行链路（导出的上行链路通过在子对等组中的边界节点的节点 ID 和端口 ID 来标识）。如果导出的上行链路的增加导致了对于合并水平链路的拓扑状态参数的有效变更，还应产生一个关于水平链路 PTSE 的实例。

Hlhp13

动作：删除在 LGN 的水平链路 Hello 数据结构的导出的上行链路表中的导出上行链路（导出的上行链路通过在子对等组中的边界节点的节点 ID 和端口 ID 来标识）。

Hlhp14

动作：删除在 LGN 的水平链路 Hello 数据结构的导出的上行链路表中的导出上行链路（导出的上行链路通过在子对等组中的边界节点的节点 ID 和端口 ID 来标识）。如果删除导出上行链路导致了对于合并水平链路的拓扑状态参数的有效变更，还应产生一个关于水平链路 PTSE 的实例。

Hlhp15

动作：清除 LGN 水平链路 Hello 数据结构的远端端口 ID。删除在 LGN 的水平链路 Hello 数据结构导出的上行链路表中的导出上行链路（导出的上行链路通过在子对等组中的边界节点的节点 ID 和端口 ID 来标识）。生成一个未包含关于该水平链路任何内容的水平链路 PTSE 的新实例。

Hlhp16

动作：清除 LGN 水平链路 Hello 数据结构的远端端口 ID。删除在 LGN 的水平链路 Hello 数据结构的导出的上行链路表中的导出上行链路（导出的上行链路通过在子对等组中的边界节点的节点 ID 和端口 ID 来标识）。

5.6.3.3 全部的过程

当 Hello 发送后，每个基于 SVCC 的 RCC 都由一个单独的 Hello 定时器控制，任何要求发送 Hello 的事件都会使该定时器复位。此外还有两个未激活定时器，在正常 Hello 协议中，未激活定时器与 SVCC 操作相关。每次处理 LGN 的水平链路扩展 IG 时就要对水平链路未激活定时器进行复位，因为这个 IG 描述了到该相邻点的所有水平链路。

注：只有当基于 SVCC 的 RCC Hello 协议处于 **2-WayInside** 状态且相应的相邻对等点的状态机处于 Full 状态时才进行这些处理。

5.7 数据库同步

当节点得知相邻对等节点（位于同一对等组内）的存在时，为了与相邻对等节点的拓扑数据库的内容同步，节点启动数据库交换程序。数据库交换程序包括一系列数据库归纳分组包的交换。数据库归纳分组包包含节点拓扑数据库中所有 PTSE 的标识信息，数据库归纳分组包使用互控(lock-step)机制进行交换，一侧发送数据库归纳分组包而另一侧用它自己的数据库归纳分组包来响应（对收到分组包的隐含确认）。在任意一次交换中两个相邻对等节点间至多只允许有一个显著的分包。

节点收到来自相邻对等节点的数据库归纳分组包，它就为分组包中描述的每个 PTSE 的出现检查它的拓扑数据库。如果拓扑数据库中没有这个 PTSE，或者相邻的对等节点有更新的 PTSE 的版本，那么节点必须向这个相邻节点或任选地从有最新版本 PTSE 的相邻节点请求 PTSE。

对于最低级的相邻对等节点，它们之间可能有多个平行的物理链路和/或 VPC 存在。正如 5.6 节中描述的，两个相邻的对等节点间的每个物理链路和/或 VPC 必须运行独立的 Hello 状态机。然而为了数据库同步和泛播的目的，在相邻的对等节点间只能进行一次对话，这个对话由相邻的对等节点状态机和

相邻的对等节点数据结构(它包括与相邻对等节点保持数据库同步和泛播到相邻对等节点的信息)来描述。无论何时链路到达 Hello 状态 **2-WayInside**, 相应的相邻对等节点的状态机就触发事件 **AddPort**; 类似地, 当链路离开 Hello 状态 **2-WayInside** 时, 相应的相邻对等节点的状态机就触发事件 **DropPort**。当事件 **AddPort** 被首次触发时, 两个相邻对等节点间的第一条链路建立后, 数据库交换程序就开始了。当两个相邻对等节点间的最后一条链路上发生事件 **DropPort** 时, 相邻的对等节点的状态机就在内部产生一个 **DropPortLast** 事件导致相邻的对等节点的所有状态信息都被清除掉。

当 PTSP、PTSE 确认分组包、数据库归纳分组包或者 PTSE 请求分组包传送时, 相邻对等节点间任意一条处于 Hello 状态 **2-WayInside** 的链路都有可能被用到。后续分组包可能在不同的链路上发送, 但不会对 PNNI 的路由信息的发布和维护造成不良后果。当相邻对等节点的状态机处于 **Full** 状态时, 最低级相邻的对等节点间的链路只能在 PTSE 中通告。在相邻的最低一级的对等节点是由物理链路和 VPC 连接的情况下, 进入或离开 **Full** 状态的变化将导致这个节点的 PTSE 的一个或多个新实例的发生或刷新。

在相邻的对等的逻辑组的节点间, 只有基于 SVCC 的 RCC 用于 PNNI 路由分组包的交换。类似的, 对最低一级相邻的对等节点的情况, 相邻的对等节点的状态机被连接至 RCC 的 Hello 状态机。注意 LGN 间的水平链路的 Hello 状态不会影响相邻的对等节点的状态。当 RCC 的 Hello 状态到达状态 **2-WayInside** 时, 相邻的对等节点的状态机就触发事件 **AddPort** 并开始数据库交换程序。当 RCC 的 Hello 状态处在 **2-WayInside** 状态之外时, 相邻的对等节点的状态机就触发事件 **DropPort**, 使它从 **Full** 状态转移至 **NPDown** 状态。

在相邻节点通过基于 SVCC 的 RCC 进行通信的情况下, 相邻的对等节点的状态机不会直接影响水平链路的 PTSE 的发生。相当于它不直接通过水平链路的 Hello 协议影响水平链路的 PTSE 的产生(见 5.6.3.2 节)。另外, 当给定链路首次产生 PTSE 时, 相关的 LGN Hello 状态机必须是 **2-WayInside** 状态而且对等数据结构必须是 **Full** 状态。

5.7.1 相邻的对等数据结构

每一个节点无论节点间的链路数有多少, 对应每一相邻的对等节点只有一个相邻对等数据结构。相邻的对等对话处于非 **NPDown** 状态时叫相邻。相邻对等数据结构包含所有在两个相邻节点间正在形成或已形成相邻的所有相关信息。属于不同对等组的相邻节点不会形成相邻关系。

状态: 相邻对等 FSM 的状态, 在 5.7.2 节有详细描述。

远端节点 ID: 用于标识相邻对等节点的节点 ID。

端口 ID 表: 端口 ID 表仅在最低一级相邻对等节点的情况下使用, 相邻对等节点由物理节点和/或 VPC 连接。端口 ID 表是连接至处于状态 **2-WayInside** 的相邻对等节点的链路的表。当 PTSP、PTSE 确认分组包、数据库归纳分组包或者 PTSE 请求分组包传送或重传至相邻对等节点, 在这个表中定义的任意链路都可以用。

主/从: 当两个相邻的对等节点正在交换数据库时, 它们形成了主/从的关系。这种关系仅与启动拓扑数据库交换有关。主节点发送第一个数据库归纳分组包并选择初始的 DS 序号。数据库归纳分组包的重发由主节点控制。从节点只能响应主节点的数据库归纳分组包。主/从关系是在协商阶段决定的。

DS 序号: DS 序号是用来标识各个数据库归纳分组包的无符 32bit 号码。当开始进入协商阶段时, DS 序号就应置为事先协商双方都不知道的值, 但这个值不能太大而不能安全地防止序号的 wrapping。一个可能的方案是使用机器的日期计数器的低 24bit。DS 序号就能由主节点随着每一个新的数据库归纳分组包的发送而逐渐增加。从节点的 DS 序号指出了它从主节点收到的最后一个分组包。

对等重发表: 对等重发表是已被泛播但没有得到相邻的对等节点确认的 PTSE 表。这些 PTSE 将要周期性地重发直到它们被确认, 或直到相邻的对等节点的状态机停止。与这个表中每个入口有关的是 PTSE 的重发定时器。这是在 PTSE 重发时间间隔秒后工作的时间间隔定时器。这个定时器在与那个 PTSE 有关的确认收到后才停止。

PTSE 重发时间间隔: 每个未被确认的 PTSE 按每个 PTSE 重发时间间隔秒重发。

对等延迟确认表: PTSE 的表, 对这些 PTSE 的延迟的确认被发送到相邻的对等节点。每个对等延

迟确认间隔秒确认分组包都被传送到包含有对等延迟确认表中所有入口的 PTSE 标识信息的相邻的对等节点后，表将被清除掉。

对等延迟确认间隔：这是对等延迟确认表的连续校验之间的时间间隔。

对等延迟确认定时器：当这个定时器超时，在对等延迟确认表中的任意未被确认的 PTSE 都要被绑定在一个确认分组包中并发送到相邻的对等节点。

PTSE 请求表：为了与两个相邻的对等节点的拓扑数据库同步需要请求的 PTSE 的表。这个表在收到了数据库归纳分组包后生成。PTSE 请求分组包用于向相邻对等节点请求表中的每个 PTSE，或者向任意其他知道拥有丢失的 PTSE 的相邻对等节点请求。这个表在合适的 PTSE 收到后清空。

DS Rxmt 时间间隔：在很短时间内，节点再次发送前一个数据库归纳分组包前等待的时间。

DS Rxmt 定时器：在 DS RxmtInterval 秒之后工作的时间间隔定时器。这个定时器在节点收到正确的数据库归纳分组包后停止。

请求 RxmtInterval：在很短时间内，节点发送新的 PTSE 请求分组包前的时间，这个分组包用于请求最后一个 PTSE 请求分组包中请求的但还没有收到的那些 PTSE。

请求 Rxmt 定时器：在请求 RxmtInterval 秒之后工作的时间间隔定时器。这个定时器在最后一个 PTSE 请求分组包中请求的 PTSE 已经全部收到时停止。

5.7.2 相邻对等状态

相邻对等状态机描述的是与相邻对等节点正在进行的数据库同步和泛播的状态。图 18 提供了可能的状态变化图。图中的箭头用导致每个状态变化的事件做标记。这些事件在 5.7.3 节中进行描述。相邻对等状态和每个状态变化包括的操作的详细描述可见 5.7.4 节。

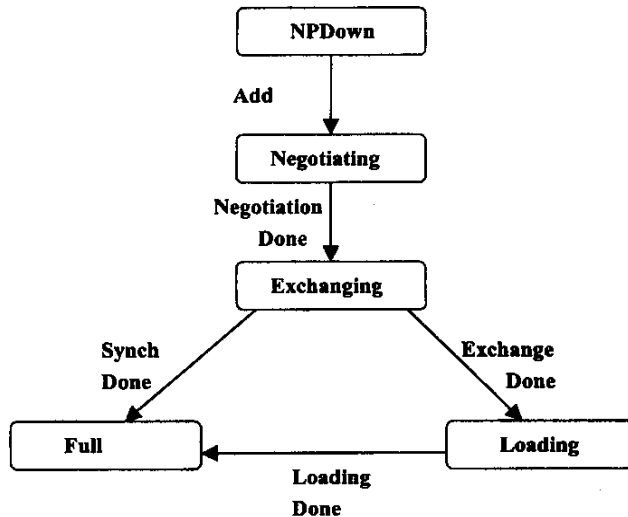


图 18 相邻对等状态的变化（数据库同步）

除了图中所示的状态转移，事件 **DSMismatch** 强制相邻对等状态进入 **Negotiating** 状态；事件 **BadPTSERequest** 强制相邻对等状态进入 **Negotiating** 状态；事件 **DropPort** 不引起状态变化；事件 **DropPortLast** 强制相邻对等状态进入 **NPDwn** 状态。

NPDwn：相邻对等 FSM 的初始状态。这一状态指出到相邻的对等节点没有激活的链路（例如在 Hello 状态 **2-WayInside**）。在这个状态，没有与相邻对等节点相连的邻近点。

Negotiating：这是在两个相邻的对等节点间产生相邻关系的第一步，这一步的目的是决定哪个节点是主节点，并且依赖 DS 序号做出这一判断。

Exchanging：这个状态下节点通过向相邻对等节点发送数据库归纳分组包来描述它的拓扑数据库。处理数据库归纳分组包的结果就是可以请求需要的 PTSE。

Loading: 在这个状态, 节点与相邻对等节点交换数据库归纳分组包的完全序列, 所请求的需要的分组包且至少有一个还没有收到。

Full: 在这个状态下, 节点已经从相邻的对等节点收到了所有已知可获得的 PTSE, 到相邻对等节点的链路现在可以在 PTSE 中通告了。

5.7.3 导致相邻对等节点状态变化的事件

状态变化是由许多事件引起的。这些事件是由两个相邻对等节点间与数据库同步有关的程序, 或者相关链路的 Hello 状态机的操作触发的。这些事件如图 18 的标注中所示。有关状态变化和事件发生后采取的动作的详细解释可见 5.7.4 节。事件定义如下:

AddPort: 连接至相邻对等节点的链路的 Hello 状态机已到达 **2-WayInside** 状态。

NegotiationDone: 主/从的关系已经协商完成, 且确定了初始 DS 序号。事件产生的更多信息可参见 5.7.6 节。

ExchangeDone: 已经收到了相邻对等节点的最后一个数据库归纳分组包, 节点的最后一个数据库归纳分组包已经发送, PTSE 请求表还未空, 节点现在知道需要请求哪个 PTSE。事件产生的更多的信息参见 5.7.6 节。

SynchDone: 已经收到了相邻对等节点的最后一个数据库归纳分组包, 节点的最后一个数据库归纳分组包已经发送, PTSE 请求表已空。

LoadingDone: PTSE 请求表的最后一个 PTSE 已收到。

DSMismatch: 已经收到有下列内容的数据库归纳分组包:

- 有不期望的 DS 序号。
- 有不期望的初始比特组。
- 有不期望的主节点比特设置。

任意这些条件都指出在数据库同步中发生了错误。

BadPTSERequest: 收到了对数据库中不存在的 PTSE 的 PTSE 请求, 或者收到的 PTSE 比 PTSE 请求表中的实例要旧。这指出在数据库同步中发生了错误。

DropPort: 到相邻对等节点的链路的 Hello 状态机脱离了 **2-WayInside** 状态。

DropPortLast: 在处理 **DropPort** 事件的过程中, 到这个相邻节点的所有端口已停机。

5.7.4 相邻对等状态机

有限的状态机是一个二维的表格, 如表 16 所示。表格的顶端是状态, 左列是事件。每对事件和状态交叉于表的一个单元。单元显示的是事件转移会发生些什么以及要采取的操作。例如, **AddPort** 和 **NPDown** 的事件和状态对交叉的单元是“Ds1, **Negotiating**”。**Negotiating** 是新状态, “Ds1”是要采取的操作。这些操作都能在下表找到。

表 16 相邻对等 FSM

事件	状态				
	NPDown	Negotiating	Exchanging	Loading	Full
Add Port	Ds1 Negotiating	Ds7 Negotiating	Ds7 Exchanging	Ds7 Loading	Ds8 Full
Negotiation Done	FSM_Err	Ds2 Exchanging	FSM_Err	FSM_Err	FSM_Err
Exchange Done	FSM_Err	FSM_Err	Ds3 Loading	FSM_Err	FSM_Err
SynchDone	FSM_Err	FSM_Err	Ds4 Full	FSM_Err	FSM_Err
Loading Done	FSM_Err	FSM_Err	FSM_Err	Ds4 Full	FSM_Err

续表 16

DS Mismatch	FSM_Err	FSM_Err	Ds5 Negotiating	Ds5 Negotiating	Ds6 Negotiating
BadPTSE Request	FSM_Err	FSM_Err	Ds5 Negotiating	Ds5 Negotiating	Ds6 Negotiating
DropPort	FSM_Err	Ds9 Negotiating	Ds9 Exchanging	Ds9 Loading	Ds9 Full
DropPort Last	FSM_Err	Ds10 NPDown	Ds10 NPDown	Ds10 NPDown	Ds10 NPDown

FSM_Err

动作：协议错误，不会发生。

Ds0

动作：无动作。

Ds1

动作：对通过物理链路和/或 VPC 相连的最低一级节点的情况，在相邻对等数据结构的端口 ID 表中要加上端口 ID。一进入这种状态，节点就会逐渐增加相邻对等节点的 DS 序列号码。如果是第一次尝试建立相邻关系，DS 序列号应被分配一个唯一的值（如日期中的时间）。然后该节点通告自己是主节点（主节点比特置 1），且开始发送带有初始、更多和主节点比特组的数据库归纳分组包。这个分组包不包括 PTSE 归纳。这个数据库归纳分组包将以 DSRxmtInterval 的时间间隔进行重发直到进入下一状态（见 5.7.5 节）。

Ds2

动作：节点必须用数据库归纳分组包向相邻对等节点发送它的拓扑数据库的内容的归纳。拓扑数据库包括这个节点的在它的对等组或是更高级对等组启动的或收到的 PTSE，每个数据库归纳分组包都有 DS 序号，这是一种隐含的确认。在任一次中只允许有一个显著的数据数据库归纳分组包。发送和接收数据库归纳分组包的细节可见 5.7.5 节和 5.7.6 节。

Ds3

动作：如果先前没有停止 DS Rxmt 定时器，就停止 DS Rxmt 定时器。开始（或继续）向相邻的对等节点和/或任意其他的对等节点发送 PTSE 请求分组包（见 5.7.7 节），每个 PTSE 请求分组包请求相邻对等节点的较新的 PTSE（在 Exchanging 状态已发现但还没有收到的 PTSE）。在相邻对等节点的数据结构的 PTSE 表中列出了这些 PTSE。

Ds4

动作：如果先前没有停止 DS Rxmt 定时器，就停止 DS Rxmt 定时器。现在数据库已同步。对最低级相邻节点的情况，所有到相邻节点的链路都在 PTSE 中进行通告。

Ds5

动作：如果在此之前未被终止，对等延迟确认定时器，DS Rxmt 定时器和请求 Rxmt 定时器将被终止。对等重传表、对等延迟确认表、PTSE 请求表和所有相关的定时器都被清除。数据库归纳的交换必须重新开始。节点为相邻的对等节点逐步增加 DS 序号，声称自己是主节点（主节点比特置为 1），开始发送带有初始、更多和主节点比特组的数据库归纳分组包。这个分组包不包括 PTSE 归纳。启动 DS Rxmt 定时器且数据库归纳分组包将以 DSRxmtInterval 的时间间隔进行重发。

Ds6

动作：与 Ds5 相同，除了如果有 PTSE 通告到相邻节点有链路存在，那些 PTSE 将被修改以删掉那些链路。这些 PTSE 可以重新启动，或者如果需要将它刷新。

DS7

动作：对于由物理链路和/或 VPC 连接的最低一级的相邻对等节点，在相邻对等节点数据的端口 ID 表中增加这个端口的 ID。

DS8

动作：与 DS7 的一样，特殊的要求是这个动作会增加到相邻对等节点的链路，导致新的 PTSE 实例的启动。

DS9

动作：在相应的相邻对等数据结构的端口 ID 表中撤消这条链路。这个动作使到相邻对等节点的链路被撤消，如果有通告那条链路的 PTSE，就必须启动受影响的 PTSE 的新的实例。如果这是到相邻对等节点的最后一条工作的链路，就产生 DropPort 事件。

DS10

动作：如果之前未终止，对等延迟确认定时器、DS Dxmt 定时器和请求 Rxmt 定时器被终止。对等重传表、对等延迟确认表、PTSE 请求表和所有相关的定时器都被清除。

5.7.5 发送数据库归纳分组包

一次只允许有一个显著的数据库归纳分组包。数据库归纳分组包的发送依赖于相邻对等节点的状态。

在 **Negotiating** 状态，节点发送带有初始化、更多和主节点比特组的空的数据库归纳分组包。当发送这样的数据库归纳分组包时，启动 DS Rxmt 定时器。当 DS Rxmt 定时器超时，这些分组包必须以每个 DSRxmtInterval 的间隔重发。

在 **Exchanging** 状态，包括发送数据库归纳分组包响应事件 **NegotiationDone** 时，数据库归纳分组包包括节点数据库中包含的拓扑状态信息的归纳。在逻辑组节点的情况下，那些在逻辑组节点级或更高级启动或接收的拓扑数据库部分包括在数据库归纳中（较低级的 PTSE 属于作为较低一级节点的交换系统的一个或多个化身的拓扑数据库，但不属于逻辑组节点的拓扑数据库）。每个这样的 PTSE 的 PTSP 和 PTSE 的头信息都列在节点的一个数据库归纳分组包中。在进入 **Exchanging** 状态后的新实例的 PTSE 无需包括在每个数据库归纳分组包中，因为它们已由正常的泛播程序处理过了。建议但不要求每个 PTSE 在发送到相邻对等节点的数据库归纳分组包的整个表中只能出现一次。

在 **Exchanging** 状态，何时发送数据库归纳分组包的判断依赖于节点是主节点还是从节点。当发送新的数据库归纳分组包时，分组包的 DS 序号按 5.7.6 节中的描述设置，并描述来自于节点的拓扑数据库的新的 PTSE 组。当单元所在的数据库归纳分组包被确认时，该单元才被相邻的对等节点认为是收到了。注意更多 (More) 比特被置为异步的，使用的原则依赖于节点是主节点还是从节点。

主节点：

数据库归纳分组包在以下情况发送：（1）从节点通过返回 DS 序号对前一数据库归纳分组包进行确认；（2）DSRxmtInterval 未被确认就过去了，这种情况下，要重传前一数据库归纳分组包。无论何时数据库归纳分组包要重传，都必须重新启动 DS Rxmt 定时器。如果节点已经发送了它的全部的数据库归纳分组包的表，或者这个分组包包括发送给从节点的数据库归纳的最后部分，那么更多 (More) 比特必须置为 0。

从节点：

仅在响应收到来自于主节点的数据库归纳分组包时发送数据库归纳分组包。如果从主节点收到的分组包是新的，就发送新的数据库归纳分组包；否则重传前一个数据库归纳分组包。如果节点已经发送了它的完整的数据库归纳分组包的表（例如这个数据库归纳分组包的内容为空），那么更多 (More) 比特必须置为 0。在 **Loading** 和 **Full** 的状态下，从节点必须重发它的最后一个数据库归纳分组包响应从主节点收到的复制的数据库归纳分组包。注意在 **Loading** 和 **Full** 状态，从节点发送的最后一个分组包必须是空的，且初始化 (Initialize)、更多 (More) 和主节点 (Master) 比特置为 0，DS 序号与目前相邻的对等节点数据结构中的相同。

5.7.6 接收数据库归纳分组包

本节详细解释对收到的数据库归纳分组包的处理过程。到来的数据库归纳分组包与相邻的对等节点通过它被接收的端口发生关系。每个数据库归纳分组包都有一个 DS 序号，且是隐含确认的。数据库归纳分组包的进一步处理依赖于与远端节点 ID 有关的相邻对等数据结构的状况。如果相邻对等状态是 **NPDown**，就必须忽略该分组包。

否则，如果状态是：

Negotiating：如果收到的分组包匹配下列一种情况，那么相邻对等状态机与事件 **NegotiationDone**（导致状态向 **Exchanging** 转移）一起执行，且分组包必须与表中的下一个一样被接受和进一步处理。否则，忽略分组包。

- 初始化 (**Initialize**)、更多 (**More**) 和主节点 (**Master**) 比特是 1，分组包内容为空，且相邻对等节点的节点 ID 大于节点本身的节点 ID。这种情况下，节点是从节点。一旦产生事件 **NegotiationDone**，从节点就必须采取下列动作：

- 停止 DS Rxmt 定时器；

- 将主节点 (**Master**) 比特置为 0（指出是从节点），设置初始化 (**Initialize**) 比特为 0，按主节点定义设置 DS 序号，向主节点发送包括节点数据库归纳的第一部分的数据库归纳分组包（见 5.7.5 节）。

- 初始化 (**Initialize**) 和主节点 (**Master**) 比特置为 0，分组包的 DS 序号等于节点自己的 DS 序号（指示确认），相邻对等节点的 ID 小于节点自己的节点 ID。这种情况下，节点是主节点。一旦产生事件 **NegotiationDone**，主节点必须采取下列动作（后面的两个动作不用按这里的顺序进行）：

- 停止 DS Rxmt 定时器；

- 处理收到的数据库归纳分组包的内容（见下段中在 **Exchanging** 状态下采取的动作的描述）；

- DS 序号加 1，向从节点发送包括节点数据库归纳分组包的第一部分的数据库归纳分组包（见 5.7.5 节），且重新启动 DS Rxmt 定时器。

Exchanging：如果主节点 (**Master**) 比特的状态与连接中的主/从状态一致，产生事件 **DSMismatch** 且停止处理分组包，否则针对如下情况作处理。

- 如果初始化 (**Initialize**) 比特已设置，产生事件 **DSMismatch** 且停止处理分组包；

- 如果节点是主节点且分组包的 DS 序号等于节点自己的 DS 序号（这个分组包是序列中的下一个），则必须接受该分组包且按如下处理（后两个动作不需要按下列顺序进行）：

- 停止 DS Rxmt 定时器；

- 处理收到的数据库归纳分组包的内容：

- a) DS 序号加 1；

- b) 如果节点已经发送了它的数据库归纳分组包的全部表，即节点发送的前一数据库归纳分组包的更多 (**More**) 比特置为 0，收到的分组包也将更多 (**More**) 比特置为 0。如果 PTSE 请求表非空则产生事件 **ExchangeDone**，或者 PTSE 请求表是空的则产生事件 **SynchDone**。否则，向从节点发送一个新的数据库归纳分组包且重新启动 DS Rxmt 定时器（见 5.7.5 节）。

- 如果节点是主节点且分组包的 DS 序列号码比节点的 DS 序号小 1，分组包是重复的。重复的分组包必须由主节点丢弃。

- 如果节点是从节点，且分组包的 DS 序列号码比节点的 DS 序号大 1（该分组包是表中的下一分组包），分组包必须被接受且如下处理（无特定顺序）。

处理收到的数据库归纳分组包的内容：

- a) 将 DS 序号设成在收到的分组包中出现的 DS 序号；

- b) 向主节点发送数据库归纳分组包（见 5.7.5 节）；

c) 如果收到的分组包的更多 (More) 比特置为 0, 刚刚传送的数据库归纳分组包也将它自己的更多 (More) 比特置为 0 (即刚传送的数据库归纳分组包为空), 那么如果 PTSE 请求表非空, 就产生事件 **ExchangeDone**, 或者如果 PTSE 请求表为空, 就产生事件 **SynchDone**。

- 如果节点是从节点, 分组包的 DS 序号等于节点的 DS 序号, 分组包是重复的。从节点必须重发上一个它已发送的数据库归纳分组包以响应重复的分组包。

- 否则, 产生事件 **DSMismatch** 且停止处理分组包。

处理收到的数据库归纳分组包的内容:

当节点收到表中下一个的数据库归纳分组包时, 最近传送的数据库归纳分组包的内容被确认为已经收到, 而且收到的数据库归纳分组包进行如下处理。

对于每一个列出的 PTSE, 节点在它自己的数据库中寻找是否已有了 PTSE 的实例。如果它没有, 或者数据库的备份比较旧 (见 5.8.2.2.3 节), 采取下列一个动作:

- 如果列出的 PTSE 是一个节点自己启动的 PTSE, 节点必须:

- 如果节点有 PTSE 的有效的实例 (见 5.8.3.7 节), 则它以较大的序号重新启动 PTSE 的新实例。

- 否则在将 PTSE 安装在拓扑数据库后将它从路由字段中删除, 并将保留生存时间设为 **ExpiredAge** (见 5.8.3.8 节)。

- 另外, 如果列出的 PTSE 有 PTSE 保留生存时间 **ExpiredAge**, PTSE 归纳分组包中的 PTSP 和 PTSE 头的内容必须被接受为内容为空的新的或更新过的 PTSE。在将 PTSE 安装到节点的拓扑数据库之后, 就执行 5.8.3.3 节中接收 PTSE 之后的程序来决定 PTSE 是否要泛播给其他的相邻对等节点。

- 另外, PTSE 是放在 PTSE 请求表中的, 使得它可以由 PTSE 请求分组包的相邻对等节点进行请求 (立即或稍后)。

Loading 或 Full:

处于这两个状态中的任意一个时, 节点已经发送和接收到了数据库归纳分组包的完整表。收到的唯一分组包应该是备份的。收到的任意其他数据库归纳分组包必须产生事件 **DSMismatch**, 导致相邻关系回复到协商状态且两个相邻的对等节点要重新同步它们的数据库。

在 **Loading** 和 **Full** 状态下收到数据库归纳分组包的后续程序与在 **Exchanging** 状态下的一样, 除了接受的表中的下一个分组包必须产生事件 **DSMismatch**, 且必须停止对这个分组包的进一步的处理。注意带有不兼容的主节点 (Master) 比特或初始化 (Initialize) 比特为 1 的分组包的接收也必须产生事件 **DSMismatch**。

5.7.7 发送 PTSE 请求分组包

在 **Exchanging** 和 **Loading** 状态, PTSE 请求表包括为了使节点的拓扑数据库与相邻对等节点的拓扑数据库同步而必须得到的 PTSE。为了请求这些 PTSE, 节点发送包含 PTSE 请求表一个或多个入口的 PTSE 请求分组包。PTSE 请求分组包仅在 **Exchanging** 和 **Loading** 状态下发送。节点可以向相邻对等节点发送 PTSE 请求分组包, 和/或任意其他处于 **Exchanging** 和 **Loading** 状态且数据库归纳指出它们有丢失的 PTSE 的节点。

对于每对相邻的对等节点每次至多只有一个显著的 PTSE 请求分组包。注意节点可以通过选择合适的在每个 PTSE 请求分组包中包括的 PTSE 的数量来控制应答流。

无论何时向相邻的对等节点发送 PTSE 请求分组包时, 相应的相邻对等节点的数据结构的请求 **Rxmt** 定时器必须重新启动。当相邻对等节点用正确的 PTSE 响应这些请求时, 即所请求的 PTSE 的子集, 收到的 PTSE 就从 PTSE 请求表中删掉。当包含在最后一个 PTSE 请求分组包中的所有 PTSE 都收到时, 新的 PTSE 请求分组包就可以发送给同一个相邻对等节点。如果在 **RequestRxmtInterval** 时间间隔内没有收到所有请求的 PTSE, 那么就发送包含丢失的 PTSE 和/或相应的相邻对等数据结构中 PTSE 请求表中其他的 PTSE 的新 PTSE 请求。丢失的 PTSE 可以由处于 **Exchanging** 和 **Loading** 状态且数据库归纳指出丢失的 PTSE 的其他相邻对等节点代为请求。这个过程持续到 PTSE 请求表为空。

当 PTSE 请求表为空时, 相邻对等状态是 **Loading** (如相邻对等节点已经发送和接收了数据库归纳

分组包的完整表)，产生了事件 **LoadingDone**。

5.7.8 接收 PTSE 请求分组包

接收的 PTSE 请求分组包定义了相邻对等节点希望收到的 PTSE 表。在相应的相邻对等节点状态处于交 **Exchanging**、**Loading** 或 **Full** 状态时，PTSE 请求分组包必须被接受。在所有其他状态，PTSE 请求分组包必须被丢弃。

对于在 PTSE 请求分组包中定义每个 PTSE，必须在节点的拓扑数据库中寻找它的实例。请求的 PTSE 被绑定进 PTSP（以节点选择的方式）且适时地被传送到相邻对等节点。这些 PTSE 千万不能放在相应的相邻对等节点的数据结构的重传表中。如果在数据库找不到 PTSE，数据库交换程序就出了错误且要产生事件 **BadPTSERequest**。

5.8 拓扑的描述和分配

5.8.1 拓扑信息

拓扑信息分组包括拓扑状态参数和节点信息。

为了允许今后的延展性，应按照如下的方法规定拓扑信息。拓扑状态信息明确地按照类型/长度/取值进行编码。这样就可以保证其信息的延展性，例如为了满足第二阶段信令要求，这种方式可以支持资源优先选择以及与策略相关的功能、QoS 和业务量合约协商等。

5.8.1.1 拓扑状态参数

拓扑状态参数是一个通用的术语，它既可以是链路状态参数，也可以是节点状态参数，如表 17 所示。拓扑状态参数可以分为两类：拓扑量度和拓扑属性。拓扑量度是一个拓扑状态参数，它要求把在一个给定通道上所有链路和节点的状态参数取值综合后，用来确定该通道是否可以接受和/或值得承载给定的连接。拓扑属性是一个被单独考虑的拓扑状态参数，以确定一个给定的链路或节点是否可以接受和/或值得承载一个给定的连接。

拓扑属性可以被进一步分为与性能/资源有关的拓扑属性与与策略有关的拓扑属性。与资源/性能有关的拓扑属性用来提供对信元传送性能的测量以及与拓扑成分有关的资源限制。与策略有关的拓扑属性用来划分拓扑成分对特定的策略限制的顺从等级。例如：可用信元率是一个与性能/资源有关的拓扑属性，而限制传递则是与策略有关的拓扑属性。

表 17 拓扑状态参数

拓扑状态参数		
拓扑量度	拓扑属性	
	与性能/资源有关的拓扑属性	与策略有关的拓扑属性
信元适用变化 (CDV)	对于 $CLP = 0$ 的信元丢失率 (CLR_0)	限制传递标志
最大信元传送时延 ($maxCTD$)	对于 $CLP = 0+1$ 的信元丢失率 (CLR_{0+1})	
管理加权	最大信元率 ($maxCR$)	
	可用信元率 ($AvCR$)	
	信元率边界 (CRM)	
	变化系数 (VF)	
	限制分支标志	

1) 链路状态参数

链路状态参数用来提供反映链路特性的信息。因为在 ATM 网络中链路是双向的，所以链路状态参数隐含地指出了其描述的方向。在 PNNI 的层次结构中，连接到一个节点上的链路可以是一条水平链路，

也可以是一条上行链路。水平链路是在同一个对等组中节点间的一条链路，而上行链路则表示了在对等组的边界节点同对等组外的高一级上行节点之间的连通性。对于水平链路，链路出局方向的状态参数由该节点通告，并使用本地节点 ID 和本地端口 ID 进行标识。对于链路入局方向的状态参数则由相邻节点进行通告，并通过与该链路相关的相邻对等节点 ID 和端口 ID 进行标识。对于上行链路，则双向的链路状态参数都由该节点通告，并且使用本地节点 ID 和本地端口 ID 进行标识。

2) 节点状态参数

节点状态参数用来提供反映节点特性的信息。如果节点信息中的节点表示标志设置为 1，则节点状态参数用构造 PNNI 复杂节点来表示。节点状态参数是与方向有关的，其方向通过逻辑节点的一对输入和输出端口 ID 来标识。复杂节点表示中的核或者是内部参考点的端口 ID 赋值为 0，PNNI 复杂节点表示中的缺省“半径”通过其设置为 0 的输入和输出端口和定义的对称性来标识。对于每个节点状态参数，复杂节点表示中只要求缺省“半径”，对于其他与一对端口或一个端口，或者是与核有关的非缺省取值则是任选的。核可以解释为一个目的地对等组的中心，或者是转接对等组一对边界节点间的“中间点”。

3) 资源可用性信息组 (RAIG)

资源可用性信息组分组包含的信息用来把拓扑状态参数的值同节点、链路和可达地址相联系（参见 5.14.5 节）。

——资源可用信息组标志。

这些标志对于解释 PNNI 协议消息中的 RAIG 产生影响。

RAIG 标志的高 5 比特用来指示 RAIG 中包含拓扑状态参数采用的是哪种业务类，对每种定义的业务类使用一个比特，这样就允许一组拓扑状态参数可以由一个或多个业务类使用。如果对于不同的业务类其拓扑状态参数也不同，则需要通告多个包含一组拓扑状态参数的 RAIG，每个 RAIG 中都包含在一个用来描述实体的信息组中。对于每个业务类至多只需要通告一组拓扑状态参数。如果对于一个业务类没有通告拓扑状态参数，则表明该实体不支持该业务类。

这些标志的最低有效比特定义了了解释包含在 RAIG 的拓扑状态参数时对 CLP 比特的应用性。可以明确指出，当该比特为 0，则表 16 将用来把信令中的 PCR 和 SCR 直接对应到 GCAC 算法使用的 PCR 和 SCR 中，当该比特为 1，则使用表 17。

——信元时延变化 (CDV)。

对于峰值到峰值的 CDV 是 $(1-\alpha)$ CTD 的分位点) 减去在整个连接保持期间连接上任何递交的信元可能经历的固定的 CTD，该取值的单位为 ms。

对应 CBR 和实时 VBR 业务来讲，CDV 是一个要求的拓扑量度，对于非实时 VBR、ABR 和 UBR 业务类，则 CDV 不适用。

——最大信元转移时延 (maxCTD)。

maxCTD 是通过链路或节点固定时延成分与 CDV 的总和。

对于 CBR、实时 VBR、非实时 VBR 业务类来讲，maxCTD 是要求的拓扑量度。对于 ABR 和 UBR 业务类，则 maxCTD 不适用。该取值的单位为 ms，该拓扑量度在通道上使用加法进行计算。

在本标准的此版本中，CDV 的累计可以使用 TM4.0 规范中定义的简单方法，例如可以使用加法。

4) 管理加权 (AW)

管理加权是由网络运营者设定的，它用来指示根据网络运营者考虑的原因重要性决定在使用链路和节点时的相对期望值。但该值不应在其他拓扑状态参数或参数组合中表达的信息里误用（例如：AvCR 或 maxCR）。

对于所有业务类而言，管理加权都是要求的拓扑量度，它是一个无没有方向的取值，其缺省值为 DefaultAdminWeight。管理加权的高低反映了对于该链路或节点进行选择的希望性。

通道的管理加权定义为通道上包含的节点和链路的管理加权的总和。

5) 对于 CLP=0 的信元丢失率。

CLR₀ 是对于 CLP=0 的业务量的最大信元丢失率指标。CLR 定义为未能通过链路或节点传递的信

元数目和通过链路或节点传递总的信元数目的比率。

对于 CBR、实时 VBR 和非实时 VBR 业务类， CLR_0 是要求的拓扑属性，对于 ABR 和 UBR 业务类， CLR_0 不适用。

6) 对于 $CLP=0+1$ 的信元丢失率 (CLR_{0+1})。

CLR_{0+1} 是对于 $CLP=0+1$ 的业务量的最大的信元丢失率指标。CLR 定义为未能通过链路或节点传递的信元数目和通过链路或节点传递总的信元数目的比率。

对于 CBR、实时 VBR 和非实时 VBR 业务类， CLR_{0+1} 是要求的拓扑属性，对于 ABR 和 UBR 业务类， CLR_{0+1} 不适用。

7) 最大信元率 (MaxCR)

MaxCR 是特定业务类型的连接可以使用的最大容量。

对于 ABR 和 UBR 业务类，MaxCR 是要求的拓扑属性，对于 CBR、实时 VBR 和非实时 VBR 业务类，MaxCR 是一个任选项。MaxCR 用信元个数/s 来表示。

MaxCR=0 这个值用来指示没有能力接受新的 ABR 或 UBR 业务类型的连接。

8) 可用信元率 (AvCR)

AvCR 是对 CBR、实时 VBR 和非实时 VBR 业务类有效可用能力的测量值，对于 ABR 业务类，AvCR 是对预留的最小信元率 (MCR) 可用容量的测量值。

对于 CBR、实时 VBR、非实时 VBR 和 ABR 业务类，AvCR 是要求的拓扑属性，对于 UBR 业务类，AvCR 不适用。AvCR 用信元个数/s 来表示。

9) 信元率边界 (CRM)

信元率边界是对有效带宽分配同为持续信元率分配的带宽差值的一个测量值，用信元个数/s 来表示，CRM 是一个关于上述分配的合并持续信元率的安全边界的指示。

对于实时 VBR 和非实时 VBR 业务类，CRM 是一个任选的拓扑属性，对于 CBR、ABR 和 UBR 业务类，CRM 不适用，CRM 用信元个数/s 来表示。

10) 变化系数 (VF)

VF 是一个对信元率边界的平方与现存所有连接的信元率的总和的变化比例的相对测量值。

对于实时 VBR 和非实时 VBR 业务类，VF 是一个任选的拓扑属性，对于 CBR、ABR 和 UBR 业务类，VF 不适用，VF 是一个无方向的比率。

5.8.1.2 节点信息

节点信息描述了节点的情况（不包含资源可用性信息和可达性信息），它应当包含如下信息：

- 节点的 ATM 端系统地址；
- 领导优先；
- 节点信息标志；
- 优选的对等组领导节点 ID；
- 上一级绑定的信息（如果该节点是对等组领导则包含该信息）。

1) 节点的 ATM 端系统地址。

该字段提供了该节点可以到达的 ATM 地址。

注：它与节点的节点 ID 不一样。

2) 领导权优先级。

该字段指示了该节点可以作为对等组领导的希望性，它主要用于 PGL 选择。

3) 节点信息标志。

目前定义了 5 个标志：“我是领导”标志，限制转接标志，节点表示标志，限制分支标志和用于 PGL 选择的非转接标志。该字段的其他比特为今后版本的 PNNI 备用。

“我是领导”标志由对等组节点设置为 1，用来说明该节点是对等组领导，它用于 PGL 选择。

在某些情况下，PNNI 节点对限制的业务量不允许进行转接。限制转接标志在节点状态参数中提供，

用来指示节点是否被限制。如果被限制，则该节点不能进行转接，除非是这个信息在节点状态参数中明确解释或解除了限制，限制转接标志是一个与策略有关的拓扑属性。

如果一个节点的限制传递设置为 1，该节点还可以作为 DTL 中的第一个或最后一个节点（例如：该节点可以产生呼叫，对通过该节点可以直达的通告的内部 ATM 地址不加限制）。对于一个逻辑组节点，任何低层的对等组都可以被归纳成 LGN，并且认为是逻辑组节点的一部分，尽管该节点可能被限制，但仍旧可以通过该节点到达。

限制转接标志也可以用来标识一个节点无法进行转接，同时也能够同其他信息一起，标识符在某些限制条件下可以转接。例如：在今后的 PNNI 的规范中定义了一个新的 PNNI 信息组，则该信息组就可以用来确定该节点是否可以转接。在这种情况下，该节点就应标记为限制转接。包含的信息组中可能会标记为“非必备”（表示该节点对于不能标识的信息组可以安全的忽略）。在这种情况下，旧版的节点（不能标识新定义的信息组的节点）只使用那些能够直接到达系统的节点。新节点可以解释信息组，并且可以使用信息组中的取值来确定是否可以解除限制转接标志。

节点使用节点表示比特来说明该节点使用的是简单节点表示还是复杂节点表示。如果该标志设置为 1，表示使用简单节点表示，在计算通道时，不需要考虑该节点的节点状态参数 PSTE。更确切地说，该节点表示为一个单点，所有的相关拓扑状态参数信息都在对应的水平链路、上行链路、域外可达地址和连接到该节点的转接网中给出。

限制分支标志用来指示该节点能否对点到多点的业务进行分支，如果该比特设置为 0，则该节点是能够支持对点到多点呼叫的附加分支点。如果设置为 1，则该节点目前不是能够支持对点到多点呼叫的附加分支点。限制分支标志是一个与资源有关的拓扑属性。

当节点在 PGL 选择算法中计算连通性时需要被忽略，则由该节点设置用于 PGL 选择的非转接标志。对于正常工作的节点，该标志必须设置为 0；如果该节点处于拓扑数据库过负荷状态时，则该标志必须被设置为 1（例如：当前已经不能存储和传播完整的激活的 PSTE 集合）。

4) 优选的对等领导 ID。

该字段规定了可以考虑被选择为 PGL 的节点 ID，它用于 PGL 选择。

5) 上一高层绑定的信息。

该字段由一个信息组构成，PGL 发送该信息组用来通知上一个高层中的所有节点，把信息绑定到父对等组中代表该对等组的父 LGN，该信息用来计算通过一个具有多等级层次时的路由，也由边界节点在层次生成时使用。

——父逻辑组节点 ID。

该字段包含了父 LGN 的节点 ID。

——父逻辑组节点的 ATM 端系统地址。

该字段中包含了父 LGN 的 ATM 端系统地址。

——父对等组 ID。

该字段包含了父 LGN 代表的对等组 ID。

——父对等组的 PGL 节点 ID。

该字段包含了作为父对等组领导的节点 ID。如果父 LGN 不知道该父对等组中选出的领导，则该字段设置为 0。当父对等组中的优选 PGL 通过设定节点信息组中的“我是领导比特”进行通告时，父 LGN 确定应当通告父对等组中 PGL 的节点 ID。

5.8.1.3 可达性信息

可达性信息是一个在 PNNI 层次中用来把可达地址同节点绑定的拓扑信息。

内部和域外可达信息是根据其起源来逻辑划分的，域外可达性是通过 PNNI 路由域外的其他协议交换导出的，内部可达性表示了对 PNNI 路由域内的本地可达性的了解。这两者区别的主要意义是域外可达信息不应在其他路由协议和路由域中通告（担心造成路由域内出现选路环路）。人工配置可以用来创建内部和域外可达性信息，并对在其他路由协议和路由域中通告的内容产生影响。

1) 内部可达地址。

该信息组用来描述内部可达的 ATM 目的地。在最低层，内部可达地址通常表示使用 ILMI 方式或者是人工配置方式登记的系统的归纳。在层次的高层中，则内部可达地址归纳了对等组成员提供信息。

当内部可达性的归纳地址是通过其配置导出的，它们由存在于归纳的可达地址中的目的地来驱动。在 5.9 节中给出了正确使用这些配置归纳地址的方针。4.1 节中也给出了相应的示例。

内部可达的 ATM 地址信息组的信息包括：

端口 ID：端口 ID 为 0 表示内部可达的 ATM 地址直接附属于 PNNI 用于选路的缺省节点表示的核。如果该端口已确定（例如：非 0），则它可以任选地用于 DTL 中；同样的，如果使用复杂节点表示，则包含在内部可达地址中的端口 ID 允许由与节点通告的地址相关的节点参数进行更精确的定义，任何通告的内部可达地址的非 0 端口都不需要在节点 PTSE 中包含的水平链路或上行链路信息组中进行通告。

通告范围：相关的可达性在层次结构中被通告的最高级。

地址信息长度：每个可达归纳地址的八位位组的数目，包括前缀长度字段。

前缀长度和前缀：前缀长度是前缀中的有效比特数，前缀本身又是对可达 ATM 地址的归纳。前缀中使用比特 0 进行右填充（右填充算法规定在 5.14.9.1 节）。地址信息长度包含前缀长度字段。

资源管理信息的任选信息组包括入局和出局两个方向。

注：对于给定的内部可达 ATM 地址信息组中的前缀，不要求它们具有相同的前缀长度（尽管它们使用相同的地址信息长度进行编码）。唯一的要求是每个地址信息长度中规定的八位位组的长度都包含单独前缀长度字段。

资源可用性信息可以任选地同内部可达地址相关联，如果存在则适用于信息组中所有的可达地址信息。如果未规定，则假定在内部地址中通告的到达的节点（节点或端口）完全可以到达规定的目的地。资源可用信息可以按照入局和出局方向来规定。如果规定了资源可用信息，则必须使用与在 PNNI 路由域中相同的类型集合，规定链路两个方向上支持所有业务类。

2) 域外可达 ATM 地址。

域外可达地址的通告包括域外地址和转接网的接入性信息。

同内部可达归纳地址一样，它描述了对一组 ATM 目的地的可达性。使用域外地址通告隐含说明了关于可达性的信息来自别处。它可以包含的情况有其他完整 PNNI 实例或者是关于通过一条特定链路可达性的配置。

在具有转接网信息的外部可达地址通告中，提供的域外可达性信息和 RAIG 与特定转接网相关。对于给定的多个相同转接网的通告，则相关的信息用于选择到达转接网的入口。

域外可达 ATM 地址信息组包括的信息如下：

端口 ID：端口 ID 为 0 表示域外可达的 ATM 地址直接附属于 PNNI 用于选路的缺省节点表示的核。如果该端口已确定（例如：非 0），则它可以任选地用于 DTL 中；同样的，如果使用复杂节点表示，则包含在域外可达地址中的端口 ID 允许由与节点通告的地址相关的节点参数进行更精确的定义，任何域外可达地址通告中的非 0 端口都不需要在节点 PTSE 中包含的水平链路或上行链路的信息组中进行通告。

通告范围：相关的可达性在层次结构中被通告的最高级。

地址信息长度：每个可达归纳地址的八位位组的数目，包括前缀长度字段。

前缀长度和前缀：前缀长度是前缀中的有效比特，前缀本身又是对可达 ATM 地址的归纳。前缀中使用比特 0 进行右填充（右填充算法规定在 5.14.9.1 节）。地址信息长度包含前缀长度字段。

资源管理信息的任选信息组包括入局和出局两个方向。

任选转接网 ID 信息组。

注：对于给定的域外可达 ATM 地址信息组中的前缀，不要求它们具有相同的前缀长度（尽管它们使用相同的地址信息长度进行编码）。唯一的要求是每个地址信息长度中规定的八位位组的长度都包含单独前缀长度字段。

资源可用性信息可以任选地同域外可达地址相关联，如果存在则适用于信息组中所有的可达地址和转接网。如果未规定，则假定在域外地址中通告的到达的节点（节点或端口）完全可以到达规定的目的地。资源可用信息可以按照入局和出局方向来规定。如果规定了资源可用信息，则必须使用与 PNNI 路

由域中相同的类型集合，规定链路两个方向上支持所有业务类。

5.8.2 PTSP 和 PTSE

拓扑信息被可靠地发布到对等组的所有节点，这些信息描述了对等组的拓扑，包括对等组内的所有节点和链路，以及每个链路可以到达的目的地。

PTSP 和 PTSE 的详细格式和编码参见 5.14 节。

5.8.2.1 PTSP

PTSP 是用来向相邻的对等点发送 PNNI 拓扑信息的分组包。PTSP 包含一个或多个用于描述独立拓扑特性的 PTSE。每个 PTSP 都携带从单独起源点发出的单一范围的信息。这些信息包含在下述两个 PTSP 头中的字段中。

起源节点 ID：生成包含在 PTSP 中所有 PTSE 的节点标识符。

起源节点的对等组 ID：生成包含在 PTSP 中所有 PTSE 的节点的对等组标识符。它定义了 PTSP 中包含的 PTSE 需要被泛播的跨度。

注：该跨度逻辑上包含了标识的对等组的所有子对等组。

注：PTSP 中的 PTSE 组可以不同，发送方不需要完全按照接收方式把从相邻对等点收到的 PTSP 中的 PTSE 转发。

同样的，发送方也不必按照特定的方式把 PTSE 组织到 PTSP 中。

5.8.2.2 PTSE

PTSE 是进行泛播和重发的单元。一个节点产生一个或多个用来描述其当前工作状态的 PTSE（例如：激活的链路、可用的带宽等），PTSE 通过泛播发布到 PNNI 的整个对等组中。

1) PTSE 头中的内容。

每个 PTSE 头中都包含如下项目：

PTSE 标识符：当一个节点产生了多个 PTSE，每个 PTSE 描述了节点环境的不同部分（例如：每个 PTSE 可以描述不同的链路组），这些 PTSE 通过 PTSE 标识符来区分，它是一个 32bit 的无符号数字，该标识符由起源节点自行分配。

PTSE 序号：对于有多个相同的 PTSE 实例同时存在的情况，PTSE 序号用来确定哪个实例是“最近”的，PTSE 序号是一个 32bit 的无符号数字。

PTSE 校验和：PTSE 校验和有如下作用：

——用来指示 PTSE 在泛播过程中或者是在节点的拓扑状态数据库中保留时是否出现了差错；

——在一些例外的情况下，它用来确定两个 PTSE 中的哪一个作为最新的进行处理。

PTSE 剩余生存时间：PTSE 被认为无效前的秒数，PTSE 剩余生存时间在如下情况中递减：

——当 PTSE 在节点的拓扑数据库中保留时；

——泛播期间。

如果 PTSE 剩余生存时间触发了“超龄”（ExpiredAge）标志，则该 PTSE 将被重新泛播，并从对等组的拓扑数据库中删除。该字段的初始值等于 PTSE 刷新间隔乘以 PTSE 生存时间系数。

PTSE 类型：PTSE 类型字段用来指示哪些受限的信息组允许出现在 PTSE 中（详见 5.14.9 节）。

2) PTSE 校验和算法。

PTSE 校验和字段是把 PTSE 中除生存时间外的所有 16bit 字的 1 的补码和再用 1 取补码。此外还包括 PTSP 中的起源节点 ID 和起源节点对等组 ID 字段，为了计算校验和，校验和字段初始值为 0。

3) PTSE 标识符和 PTSE 实例标识符。

为了实施泛播程序，每个 PTSE 和每个给定的 PTSE 实例，都必须唯一的被标识。用来标识 PTSE 的信息包括：

——起源节点 ID；

——PTSE 标识符。

用来从相同 PTSE 的实例中标识出一个 PTSE 实例，还需要如下一些附加信息：

——PTSE 序号；

——PTSE 剩余生存时间；

——PTSE 校验和。

除了起源节点 ID 是在包含 PTSE 的 PTSP 头中携带的，这些用于标识 PTSE 的数据大多包含在 PTSE 头中。

4) PTSE 实例比较。

相同 PTSE 的不同实例是指具有相同起源节点 ID 和 PTSE 标识符但 PTSE 序号不同的两个 PTSE，例如一个 PTSE 是另一个 PTSE 的更新。

当相同 PTSE 的两个实例同时存在时，必须要比较这两个实例，来确定它们是否是不同的实例。如果是，哪个实例是“最近”的，最近的 PTSE 总是比其他 PTSE 具有更高的优先权。这种比较包含如下步骤：

- a) 比较 32bit 的无符号整数，具有更大的 PTSE 序号的 PTSE 实例是最近的实例；
- b) 否则（例如两个版本具有相同的 PTSE 序号）如果一个 PTSE 版本的 PTSE 剩余存活时间等于“超龄”；而其他的 PTSE 不等，则剩余存活时间等于超龄的 PTSE 认为是最近的；
- c) 否则（两个版本都超时或都没超时）当 PTSE 的校验和作为无符号整数比较时比其他的大，则认为是最近的 PTSE（注当两个具有不同的 PTSE 的校验和，它们可以有不同的内容，当然也是不同的实例，但却不能确定哪个是最近的实例，如果选择 PTSE 版本不是最近的，则 PTSE 生成者需要结束重新产生 PTSE）。否则两个版本的 PTSE 具有相同的校验和并且认为这两个 PTSE 是相同的实例。

5.8.3 泛播

5.8.3.1 概述

PNNI 泛播算法使得 PTSE 能够可靠地分配在 PNNI 的对等组中，它保证了对等组中的每个节点都有一个同步的拓扑数据库。

所有处于 **Exchanging**、**Loading** 和 **Full** 状态的相邻点用于泛播程序，并且必须能够完全发送和接收所有类型的 PNNI 路由分组包。

实际的泛播程序如下所述。PTSE 被封装在 PTSP 中，收到 PTSP 后对其包含的 PTSE 进行检查。通过把 PTSE 的标识信息封装到“确认分组包”中对每个 PTSE 进行证实，然后把“确认分组包”发回相邻的对等点；此外，如果 PTSE 比节点中当前的版本更近，则把该 PTSE 安装到拓扑数据库中，并且向其相邻对等点进行泛播。同时把向相邻对等节点发送 PTSE 的情况记录下来，如果没有收到证实则重新进行发送。

泛播过程将在以下内容中详细描述。

5.8.3.2 发送 PTSP

PNNI 节点使用 PNNI 拓扑状态分组包（PTSP）来交换链路状态信息，5.14 节给出了 PTSP 的格式。

节点为了响应如下事件，生成并发送 PTSP：

- （自身）生成一个新的 PTSE（参见 5.8.3.7 节）。
- 重新生成一个新的 PTSE（自身生成的）实例（参见 5.8.3.7 节）。
- 把收到的非自身生成新的 PTSE 实例安装到数据库中，并且要把该 PTSE 向其他对等点泛播（参见 5.7.6 节和 5.8.3.3 节）。
- 如果在对等重发表中有未证实的 PTSE，且 PTSE 重发定时器超时（参见 5.8.3.4 节）。
- 响应 PTSE 请求（参见 5.7.8 节）。
- PTSE 剩余生存时间超时（参见 5.8.3.8 节）。

无论造成节点生成和发送 PTSP 的事件是否相同，但采取的动作都是相同的。首先要捆绑 PTSE，PNNI 协议要求一个节点产生的所有 PTSE 必须捆绑到一个 PTSP 中传送。

通常建议把尽可能多的 PTSE 封装到一个 PTSP 中发送，但是一个节点不能使生成的 PTSP 长度大于链路上允许传送的分组包最大长度。同样一个节点也不可能违背给定链路使用的业务量合约。这暗示

了对于一个给定的相邻点，所有 PTSP 的数量和容量需要在给定的链路上排队，并且需要由发送节点进行监控。如果发送节点到达的点因为需要保证在链路和业务量约定的范围而不能够立即发送 PTSP，则必须保存状态，这样在条件满足时，需要的 PTSP 更新就可以发送了。

在捆绑和封装完 PTSP 中的 PTSE 后，就需要初始化 PTSP 头，PNNI 分组包然后传递到 AAL5 的业务接口进行传送。

不需要捆绑在一起传送的 PTSE 保存其状态，任何后续重发的 PTSE 或者是发送新的 PTSE 实例都可以捆绑到不同的 PTSP。节点允许重新安排其他节点的 PTSE 到不同的 PTSP 中，与之相对应的限制是节点不允许对由其他节点产生的 PTSE 内的信息进行重新整理。

5.8.3.3 接收到 PTSP

从相邻的对等点收到 PTSP 后，每个包含在其中的 PTSE 都要使用如下所述的步骤进行检查。

1) 如果 PTSE 剩余生存时间与“超龄”标志不同，且 PTSE 的校验和是有效的；如果 PTSE 确定为无效的，则 PTSE 处理完毕；

2) 否则如果接收链路具有相邻对等节点 **Exchanging** 和 **Loading** 的状态，并且在接收链路的 PTSE 请求表中已经有 PTSE 的实例，则比较收到 PTSE 和请求表中的实例。

a) 如果接收的 PTSE 没有请求表中的实例新，则表明在数据库交换过程中出现差错。在这种情况下，通过产生 **BadPTSERequest** 事件与相邻对等点启动数据库交换过程并停止处理 PTSP。

b) 否则从接收链路 PTSE 请求表中删除 PTSE，并继续处理该 PTSE。

3) 如果节点在拓扑数据库中具有的 PTSE 实例比接收到的 PTSE 要新。

a) 该 PTSE 在接收链路的对等重发表中，则立即对接收的 PTSE 进行证实，该 PTSE 处理完毕。

b) 该 PTSE 不在接收链路的对等重发表中，则把数据库的副本封装到 PTSP 中并对相邻对等点进行泛播。PTSP 中的 PTSE 将把 PTSE 剩余生存时间减 1（除非 PTSE 的剩余时间已经等于“超龄”），这样就避免了由于实施差错带来的泛播环路。同样的，PTSE 要添加到相邻对等点的对等重发表中，这样在没收到证实的情况下就可以进行重发。

4) 如果数据库中已有的实例同接收的 PTSE 的实例相同。

a) 如果 PTSE 包含在接收链路的对等重发表中，则泛播的 PTSE 被看作是隐含的证实。在这种情况下，从接收链路的对等重发表中把 PTSE 清除，该 PTSE 处理完毕；

b) 否则立即发送 PTSE 证实分组包对接收的 PTSE 进行证实，该 PTSE 处理完毕。

5) 否则如果没有数据库的副本，且接收的 PTSE 具有“超龄”的 PTSE 剩余生存时间，则立即发送 PTSE 证实分组包对接收的 PTSE 进行证实。

a) 如果接收的链路有相邻对等的 **Exchanging** 和 **Loading** 状态，超时的 PTSE 安装到接收节点的拓扑数据库，该 PTSE 处理完毕。这种做法避免了在特定的竞争条件下错误的 PTSE 请求事件；

b) 否则该 PTSE 的处理完毕。

6) 若非第 5) 步的结果，或者没有数据库副本且接收的 PTSE 不是超龄的，或者是接收的 PTSE 实例比数据库中的副本中更新，在这种情况下，必须要执行下述步骤。

a) 如果起源节点 ID 作为接收节点自身（例如：起源节点 ID 等于接收节点的节点 ID）。

i) 如果接收节点有一个有效的 PTSE 实例，则必须重新产生一个 PTSE，其序号比接收的 PTSE 的序号大 1（参见 5.8.3.7 节），该 PTSE 的处理完毕。

ii) 否则接收的 PTSE 必须通过设置 PTSE 剩余生存时间为超龄由 PNNI 路由域清除，在拓扑数据库中安装超时的 PTSE 并且启动泛播 PTSE，所有这些完成之后，该 PTSE 的处理完毕。

b) 否则将 PTSE 添加到接收链路的对等延迟证实表。

c) 此外 PTSE 安装到接收节点的数据库，在处理中，以前形成的数据库副本（如果有）从所有

相邻对等点的对等重发表和对等延迟证实表中删除（包括那些相应的接收链路）。

d) 此外 PTSE 向所有相邻对等点的子集进行泛播，并对每个相邻对等点执行下述步骤：

i) 如果从相邻的对等点收到新的 PTSE，则检查下一个相邻对等点。

ii) 否则如果相邻对等点处于比 **Exchanging** 次要的状态，它则不参加泛播，并且必须检查下一个相邻对等点。

iii) 否则如果链路对等节点处于 **Exchanging** 和 **Loading** 状态，检查与该相邻点相关的 PTSE 请求表，如果在表中有新 PTSE 的实例，则比较新的 PTSE 和相邻对等点的副本。

——如果新的 PTSE 不是最近的，则检查下一个相邻对等点。

——如果两个副本是相同的实例，则从 PTSE 请求表中删除 PTSE，并检查下一个相邻对等点。

——否则新的 PTSE 是最近的，从 PTSE 的请求表中删除 PTSE，并且向相邻的对等点泛播 PTSE，如步骤(iv)所述：

iv) 否则把 PTSE 封装到 PTSP 中，并向相邻的对等点泛播。并 PTSP 中的 PTSE 副本中的 PTSE 剩余生存时间减 1（除非 PTSE 剩余生存时间等于超龄）。同样的，PTSE 要添加到相邻对等点的对等重发表中，这样在没收到证实的情况下就可以进行重发。

5.8.3.4 对等重发表的处理

一个节点必须周期地检查对每个相邻对等点的对等重发表以确定需要重发的 PTSE。这些 PTSE 在上次封装到 PTSP 中发送到相邻对等点后的时间已经超过 $PTSERetransmissionInterval$ ，在节点拓扑数据库中的当前版本的 PTSE 就用于这个目的。PTSP 中的 PTSE 的副本的 PTSE 生存时间是把当前数据库内的生存时间值减 1。

5.8.3.5 发送 PTSE 证实

根据 5.8.3.3 说明的程序，一些 PTSE 需要立即证实，一些则可以延迟证实。对等延迟证实表将保留 PTSE 证实，将在一个特定的时延（称为 $PeerDelaydAckInterval$ ）后发送。每隔 $PeerDelaydAckInterval$ 后对该表进行扫描，如果该表不为空，则产生并发送包含一定数量的 PTSE 标识信息项目的 PTSE 证实分组包。被证实的 PTSE 从对等延迟证实表中删除。

注：对等延迟证实表中的 PTSE 证实可以和立即响应的证实共同发送。附录 G 中给出了如何选择 $PeerDelaydAckInterval$ 的原则。

5.8.3.6 接收证实分组包

一个证实分组包中包含一组 PTSE 标识信息单元，这些证实分组包的发送是为了响应 PTSP，参见 5.8.3.3 节中的步骤 3 的 a)、4)的 b)、5)和 6)的 b)。

当从相邻对等点收到一个证实分组包，则检查该分组包中封装的 PTSE 标识信息单元。对于每个 PTSE 标识信息单元，如果相邻对等点的对等重发表中包含了完全相同的 PTSE 实例，则把对等重发表中 PTSE 删除。

5.8.3.7 产生一个新的 PTSE 或者是新的 PTSE 实例

当一个节点产生或更新了其一个或多个自身生成的 PTSE 时，启动 PTSE 的泛播。

当一个节点产生了一个 PTSE，或者是先前生成的 PTSE 被更新，节点则按照下述步骤进行：

1) 确定 PTSE 的 PTSE 序号的取值。

a) 如果该 PTSE 是第一次产生，则 PTSE 序号根据初始序号（InitialSequence Number）进行设置。

b) 如果这是对应于在泛播期间接收到一个自我产生的 PTSE（见 5.7.6 节和 5.8.3.3 节），则选择接收到的 PTSE 后的下一个 PTSE 序号。

c) 如果这是对于已经存在的 PTSE 的更新，则选择当前数据库副本中 PTSE 的下一个的序号。

“下一个” PTSE 序号表示把参考的 PTSE 序号加 1。

为了保证 PTSE 序号的空间不至于过快地耗尽，在每个 $MinPTSEInterval$ 内限制节点只能更新任何特定的 PTSE 一次。 $MinPTSEInterval$ 的缺省值为 1s，这意味着在没有差错的情况下 PTSE 序号的空间每 136 年才会耗尽一次。

2) 节点生成的 PTSE 的内容, 说明了节点环境的相应部分。无论在上次 PTSE 产生之后是否在信息组中产生了重大的变化, 只要重新生成了一个 PTSE, 则 PTSE 中所有信息中的信息必须反映出当前节点的状态。

3) 节点计算并安装 PTSE 校验和。

4) 节点通过计算 $PTSERefreshInterval$ 与 $PTSELifeTimeFactor$ 的乘积得到 PTSE 的剩余生存时间, 并把结果赋给 PTSE 剩余生存时间。

5) PTSE 安装到节点拓扑数据库后, 从所有相邻对等点的对等重发表和对等延迟证实表中把以前生成的 PTSE 实例 (如果存在) 删除。

6) 对处于 **Exchanging** 状态或更高级别状态的每个相邻对等点, 把 PTSE 封装到 PTSP 中, 并把它向相邻对等点泛播。这些 PTSE 也同时添加到相邻对等点的对等重发表中。

5.8.3.8 PTSE 的生存时间超时

PTSE 生存时间超时有下面两种方式:

1) 一个节点提早使其自身的生产的 PTSE 老化, 这种情况的经常发生是由于 PTSE 中的信息变为无效造成的。

2) 在节点的拓扑数据库中的 PTSE 自然老化。

有些情况下, 节点可能希望在自身生成的 PTSE 未老化之前, 把该 PTSE 从拓扑数据库中删除, 节点把该 PTSE 的 PTSE 剩余生存时间设置为超龄, 这种方式称为“提早老化”, 提前老化方式只允许节点把自身生成的 PTSE 老化 (例如: 那些起源节点 ID 等于节点自身标识符的 PTSE)。

当 PTSE 的剩余生存时间变为超龄 (例如: 由于提早老化或者是自然老化) 则该 PTSE 被清除。清除一个 PTSE 需要执行如下步骤:

1) 从所有的相邻对等点的对等重发表和对等延迟证实表中删除该 PTSE。

2) 设置该 PTSE 的剩余生存时间为超龄。

3) 向处于 **Exchanging** 状态或更高级别状态的每个相邻对等点泛播不带任何内容的 PTSE。把超时的 PTSE 添加到相邻对等点的对等重发表中。

4) 如果满足了 5.8.3.9 节描述的条件, 则从拓扑数据库中把超时的 PTSE 删除。

5.8.3.9 从拓扑数据库中删除 PTSE

当且仅当下列条件满足时, 必须把 PTSE 从节点的拓扑数据库中删除。

1) PTSE 的剩余生存时间等于超龄;

2) 该 PTSE 已经不再包含在任何节点的对等重发表或者是节点的对等延迟证实表中。

3) 节点的相邻对等点没有一个处于 **Exchanging** 和 **Loading** 状态。

5.8.4 拓扑数据库中的 PTSE 的老化

PNNI 节点必须监视收集在拓扑数据库中 PTSE 的老化情况。老化功能通过调整 PTSE 的剩余生存时间, 反映 PTSE 进入到拓扑数据库后的历时情况。老化功能也用来检测自身生成的 PTSE 需要被刷新, 或者非自身生成的 PTSE 需要被清除 (例如: 因为它们已经到达超龄门限)。

在进行选路计算时不应当考虑已经到达超龄门限的 PTSE。

5.8.4.1 PTSE 剩余生存时间的计算

PTSE 剩余生存时间是根据 PTSE 被装入拓扑数据库后和经历时间的剩余生存时间来确定的。如果初始的剩余生存时间小于或等于经历时间, 则剩余生存时间必须被设置为超龄, 并且执行 5.8.3.8 节规定的程序。否则剩余生存时间的计算是把 PTSE 初始安装的剩余时间减去经历时间。

5.8.4.2 刷新自身生成的 PTSE

当自身生成的 PTSE 的剩余生存时间低于初始生存时间减去 $PTSERefreshInterval$ 的门限时, 暗示了该 PTSE 至少在一个 $PTSERefreshInterval$ 内未被刷新, 因此需要重新生成 PTSE (参见 5.3.8.7 节)。

5.8.5 拓扑信息的触发更新

为了避免网络信息的老化, 未变化和未更新的 PNNI 拓扑信息需要定期地重新生成。这些已经在

5.8.4.2 节中讨论过了。

PNNI 实体只有在响应重大变化事件时，才在其他时间重新生成 PTSE。对于任何信息组中的任何成分的重大变化，都会生成一个包含 PNNI 拓扑状态信息 (PTSE) 的重大变化事件。这种为了响应重大变化事件的更新称之为触发更新。

在两个成功地重新生成的 PTSE 之间的周期不应小于 MinPTSEInterval。MinPTSEInterval 定时器的作用是为了防止节点以过高的速率向网络中注入 PTSE。PTSE 的重新生成使用 5.8.3.7 节定义的程序。

5.8.5.1 PNNI 拓扑状态单元的重大变化事件

PTSE 中包含了不同数量的信息组 (参见 5.14)，向 PTSE 中增加一个新的 IG 就构成了对包含该信息组的 PTSE 的重大变化事件。删除 PTSE 中最后的 IG 也构成了重大变化事件，也可能导致 PTSE 的提前老化，对 PTSE 包含的 IG 中的任何重大变化都构成了对 PTSE 的重大变化。

对 IG 内容重大变化的定义取决与 IG 中的信息类型，这里定义了 6 种 IG：

- 节点信息；
- 内部可达的 ATM 地址；
- 域外可达的 ATM 地址；
- 节点状态参数；
- 水平链路；
- 上行链路。

IG 中包含了资源可用性信息组 (有时是任选的)，任何对资源可用性信息组内容的修改都被看作是对 IG 内容的修改，因此对于 RAIG 的任何重大变化构成了对包含该信息组的 PTSE 的重大变化

5.8.5.2 PTSE 重大变化的处理

当发生了重大变化事件，如果最后生成的 PTSE 是在 MinPTSEInterval 时间间隔外，则应立即重新生成；如果最后生成的 PTSE 是在 MinPTSEInterval 时间间隔内，则不必立即重新生成 PTSE，起源节点必须等待 MinPTSEInterval 超时后再重新产生 PTSE。如果从上次生成已经超过 MinPTSEInterval 的时间，则必须重新产生 PTSE 并按 5.8.3.7 节规定的程序进行泛播。

以下子节定义了对 PTSE 中的每种高级信息组和 RAIG 类型构成的重大变化。

1) 节点信息组内的变化。

对节点信息的变化是重大变化。

2) 内部可达的 ATM 地址 IG 的变化。

对内部可达的 ATM 地址 IG 的变化是重大的变化。这种情况通常发生在内部地址由可达变为不可达的情况，反之亦然。

内部可达的 ATM 地址信息组针对一种或多种业务类的每个方向，任选的包含了资源可用性信息组。对资源可用性信息组的变化关系到内部可达的 ATM 地址时，按照 5.8.5.2 节的 5) 条中对资源可用性信息组描述的相同原则认为是重大变化。

3) 域外可达的 ATM 地址 IG 的变化。

对域外可达的 ATM 地址 IG 的变化是重大的变化。这种情况通常发生在域外地址由可达变为不可达的情况，反之亦然。

域外可达的 ATM 地址信息组针对一种或多种业务类的每个方向，任选地包含了资源可用性信息组。对资源可用性信息组的变化关系到域外可达的 ATM 地址时，按照 5.8.5.2 节 5) 条中对资源可用性信息组描述的相同原则认为这种变化是重大的。

4) 在其他信息组中的变化。

本节指出了在其余信息组中的变化。

- 节点状态参数；
- 水平链路 IG；
- 上行链路 IG。

对这些 IG 的增加和删除都是重大的变化。

在这些 IG 中, 每个都包含了资源可用性信息组, 任何对资源可用性信息的重大变化都被看作是对 IG 的重大变化。

这些 IG 中包含了节点、端口和对等组的标识符, 任何对这些字段的变化都被看作是重大的, 水平链路和上行链路 IG 也包括汇集令牌, 对于汇集令牌的变化 (由于重新配置) 也认为是重大的变化。

上行链路信息属性 (ULIA) 中包含了一个序号, 按照 ULIA 生成的定义, 这个号码只有在其他内容已进行了重大的变化后才变化。因此包括对上行链路的通告, ULIA 序号的改变也认为是重大的, 尽管 ULIA 没有其他变化, 也认为该变化是重大的。

5) 对资源可用性信息的变化。

下面的子节定义了对构成 RAIG 重大变化的 RAIG 的成分变化。

a) 管理加权

AW 是一个要求的量度, 任何变化都是重大的。

b) 信元丢失率 (CLR₀)

CLR₀ 是一个要求的属性, 任何变化都是重大的。

c) 信元丢失率 (CLR₀₊₁)

CLR₀₊₁ 是一个要求的属性, 任何变化都是重大的。

d) 可用信元率 (AvCR)

一般来讲, 当 AvCR 接近 0 时, AvCR 的变化是重大的。例如: AvCR 从 0 变为非 0 时表示可以承载一些刚才还不能接纳的呼叫, 这也是重大的变化。同样的, 当 AvCR 从非 0 变为 0 后, 这表示已经不能再接收其他的呼叫了, 这也是重大的变化。

AvCR 的变化是根据对上一次通告的值的差的比来测量的。比例乘积参数 (AvCR_PM) 用百分比方式来表示, 提供了对 AvCR 重大变化定义的灵活控制。最小门限参数 (AvCR_mT), 是用 maxCR 的百分比来表示的。它用来保证非 0 的无效变化范围取值。

根据给出的 AvCR 的前一个取值的算法给出了无效 AvCR 变化范围的上下边界, 任何根据前一值计算出的 AvCR 变化在上下边界范围内, 这种变化则是无效的, 如果计算出新的 AvCR 在边界以外, 则这种变化则是重大的。

无效变化的边界范围使用下述算法进行计算:

```
compute_AvCR_bounds ( PREV_AvCR, maxCR, AvCR_PM, AvCR_mT )
```

```
{
  /*
```

```
PREV_AvCR = 对业务类而言, 当前或前一个通告的 AvCR 取值 (用信元/s 表示)
```

```
maxCR = 业务类的最大信元率 (用信元/s 表示)
```

```
AvCR_PM = 比例乘积参数 (用百分比表示)
```

```
( 1 <= AvCR_PM <= 99 )
```

```
AvCR_mT = 最低门限 (用百分比表示)
```

```
( 1 <= AvCR_mT <= 99 )
```

```
*/
```

```
delta = PREV_AvCR * ( AvCR_PM/100);
```

```
min_delta = maxCR * ( AvCR_mT/100 );
```

```
if ( delta < min_delta ) { delta = min_delta; }
```

```
upper_AvCR_bound = PREV_AvCR + delta;
```

```
if ( upper_AvCR_bound > maxCR )
```

```
{ upper_AvCR_bound = maxCR; } /* set upper bound to maxCR */
```

```
if ( delta > PREV_AvCR )
```

```

{ lower_AvCR_bound = 0; }      /* set lower bound to zero */
else
{ lower_AvCR_bound = PREV_AvCR - delta; }
} /* end compute_AvCR_bounds() */

```

当 AvCR 发生变化，则使用下述算法来确定这种变化是否有效。

```

/* NEW_AvCR = new value for AvCR */
if (NEW_AvCR <= lower_AvCR_bound ||
    NEW_AvCR >= upper_AvCR_bound)
{ /* change in AvCR is significant */ }
else
{ /* change AvCR is NOT significant */ }

```

e) 最大信元传递时延(maxCTD)

maxCTD 要求度量，用于决定 maxCTD 重大变化的算法，除没有时延上限外，与上述用于 AvCR 的算法非常相似。

maxCTD 变化的测量是根据与通告的上一值差的比例。用百分比表达的比例系数 (maxCTD_PM) 参数，对 maxCTD 重大变化的定义提供灵活的控制。

给定 maxCTD 先前的值，算法建立 maxCTD 的上限和下限，这样定义了不重大的范围。计算出 maxCTD 的新值，如果在范围之内就是从先前值不重大的变化，如果超出范围就是重大的变化。

不重大范围的极限按照下列算法计算：

```

compute_maxCTD_bounds(PREV_maxCTD,maxCTD_PM)

```

```

{
/*
. PREV_maxCTD 是先前/现在为 maxCTD 通告的值
. maxCTD_PM 是百分比 maxCTD 的比例系数
(1<= maxCTD_PM<=99)
*/

```

```

*/

```

```

delta= PREV_maxCTD *( maxCTD_PM /100);

```

```

upper_maxCTD_bound=PREV_maxCTD+delta;

```

```

if (delta>PREV_maxCTD)

```

```

{lower_maxCTD_bound=0;}/* 设置下限为 0 */

```

```

else

```

```

{lower_maxCTD_bound=PREV_maxCTD-delta;}

```

```

} /* 结束 compute_AvCR_bounds */

```

当 maxCTD 变化时，使用下列算法决定变化是否重大：

```

/* NEW_MaxCTD 是 MaxCTD 的新值 */

```

```

if(NEW_MaxCTD<=lower_MaxCTD_bound || NEW_MaxCTD>=
upper_MaxCTD_bound)

```

```

{ /*MaxCTD 的变化重大 */}

```

```

else { /*MaxCTD 的变化不重大 */}

```

f) 信元时延变化 (CDV)

CDV 要求度量，用于决定 CDV 重大变化的算法，与上述用于 maxCTD 的算法非常相似。

CDV 变化的测量是根据与通告的上一值差的比例。用百分比表达的比例系数 (CDV_PM) 参数，对 CDV 重大变化的定义提供灵活的控制。

给定 CDV 先前的值，算法建立 CDV 的上限和下限，这样定义了不重大的范围。计算出 CDV 的新值，如果在范围之内就是从先前值不重大的变化，如果超出范围就是重大的变化。

不重大范围的极限按照下列算法计算：

```
compute_CDV_bounds(PREV_CDV,CDV_PM)
{
  /*
    .PREV_CDV 是先前/现在为 CDV 通告的值
    .CDV_PM 是百分比 CDV 的比例系数
    (1<= CDV_PM<=99)
  */
  delta= PREV_CDV *( CDV_PM /100);
  upper_CDV_bound=PREV_CDV+delta;
  if (delta>PREV_CDV)
    {lower_CDV_bound=0;}/* 设置下限为 0 */
  else
    {lower_CDV_bound=PREV_CDV-delta;}
  } /* 结束 compute_AvCR_bounds */
  当 CDV 变化时，使用下列算法决定变化是否重大：
  /* NEW_CDV 是 CDV 的新值 */
  if (NEW_CDV<=lower_CDV_bound || NEW_CDV>=
    upper_CDV_bound)
    { /*CDV 的变化重大 */}
  else { /*CDV 的变化不重大 */}
```

g) 最大信元率 (maxCR)

maxCR 要求属性，maxCR 任何的变化都不重大。

注：maxCR 用于计算 AvCR 不重大的范围。

h) 信元率边界 (CRM) 和变化因素 (VF)

CRM 和 VF 是任选属性，CRM 和 VF 的变化不认为是重大。

5.9 通告和归纳可达地址

本节描述逻辑节点在层次较高级表示的对等组归纳地址的缺省行为，同样的行为应用于最低节点的附着系统的归纳地址。

5.9.1 地址通告的范围

可达地址通告范围由等级表示语规定，它规定地址将被通告一直到这级，但不能到 PNNI 的任何较高级。如果等级表示语设为 0，那么通告范围就不被限制，这意味着地址可以被通告到全 PNNI 路由域。

只有在地址的通告范围高于或等于对等组的范围，地址才有资格通告或归纳进对等组。如果归纳的地址具有不同的通告范围，用归纳的最高通告范围通告。

PNNI 地址通告范围在 UNI 中是用组织范围登录，它决定于网络的具体对应关系，参见 5.3.6 节。

5.9.2 归纳地址和抑制的归纳地址

归纳地址是一组地址的缩写，它由归纳地址所共有的地址前缀表示。当节点通告归纳地址时，匹配归纳地址的任何目的地的呼叫（除非目的地地址匹配已通告的更长地址前缀）可以选取路由到那个节点。这意味着当匹配这个归纳的所有可达地址要到达这个节点时应该使用地址归纳。节点能够有任意个归纳地址。

抑制的归纳地址用于匹配这个前缀的地址的通告，而不管范围。如果为给定的可达性通告配置抑制，将不管地址的范围而抑制地址的通告。范围足够宽而在较高级通告的通告（即通告范围值小于这个节点

的等级)将被这种配置抑制。节点能够有任意个抑制的归纳地址。

内部和域外可达地址分别有归纳地址和抑制的归纳地址。域外归纳信息不影响内部可达地址的通告,内部归纳信息不影响域外可达地址的通告。

逻辑组节点有一个与它表示的对等组 ID 等同缺省的内部归纳地址,没有抑制的归纳地址。除非节点在 104 等级(在这种情况下节点没有缺省的归纳地址),否则最低级节点有一个缺省的内部归纳地址(节点地址的 13 个八位位组的前缀)。节点缺省没有内部抑制的归纳地址、域外归纳地址或域外抑制的归纳地址。5.8.1.3 节讲述了内部和域外地址的区别。

当归纳地址和/或抑制的归纳地址之间出现重叠时,将使用最长匹配归纳地址或抑制的归纳地址来确定地址是否明确地被通告,或通过使用归纳地址或抑制的归纳地址间接地被通告。归纳地址和抑制的归纳地址一样的配置是错误的。

5.9.3 本土地址

如果地址与节点的归纳地址之一匹配,这个地址被认为是本土地址。归纳地址被通告;正在被归纳的地址(或更详细的归纳地址)被抑制。

如果没有匹配合适的范围,就不通告归纳地址。

归纳地址范围内的地址能够用于网络的其他地方。如果网络的不同部分正通告的地址前缀具有不同的长度,那么较长的地址前缀将用于匹配它的目的地;如果它们具有相同的长度,那么呼叫发起者可以选择任何一个。在后一种情况,除非所有的目的地可达两个地方,否则这可能导致经常的回转。

5.9.4 外来地址

不匹配节点的任何一个归纳地址的地址,被认为是节点的外来地址。如果相关的通告范围高于或等于节点的等级,地址和通告范围没有进一步归纳而传递。如果地址与一种的归纳地址之一匹配,那么地址就不再通告。

5.9.5 组地址

当存在多个相同的组地址时,只通告一个地址。在这种情况下,使用最高通告范围。没有缺省的归纳组地址。

5.9.6 域外可达地址

PNNI 路由协议允许域外可达地址通告入 PNNI 路由域。

当归纳域外可达地址时,与正进行归纳的地址有关的 RAIG 汇集的实现决定于具体实施。

5.10 层次

5.10.1 对等组领导(PGL)

PNNI 层次需要对等组选择一个节点完成 LGN 的功能,被选择的这个节点称为对等组领导,成为对等组领导的可能性是通过配置而建立,这种可能性由每一节点通告的 PGL 优先级指出。节点 ID 用于具有相同 PGL 优先级节点之间的平局打破者。

5.10.1.1 对等组领导选举算法

PGL 选举算法用于动态选择适当的节点来承担领导职责或替代出局 PGL。与某节点具有连通性的所有节点,必须为具有最高且非零 PGL 优先级的节点投票,PGL 优先级使用节点 ID 完成平局的打破。对等组内所有节点用 PTSE 通告领导权优先级。如果没有节点通告非零 PGL 优先级,那么就没有节点可选。如果节点确定在充足的时间内不能得到意见一致的投票之后,节点将为 PGL 选举考虑 2/3 的大多数投票,所以在对等组内有少量的节点的情况下,错误实施不可能引起悬而不决的选举。出于类似的原因,在正常操作的情况下对等组中的所有节点必须参与 PGL 选举,然而为 PGL 选举在节点 IG 设置非转接标志的节点。不参与 PGL 选举,即它不运行 PGL 选举的 FSM,且通告领导权优先级(Leadership Priority)和优选的 PGL(Preferred PGL)为 0。在决定对等组的连通性时,对等组的其他节点也不考虑这种节点,其他节点将忽略这种过载节点通告的 PGL 优先级和优选的 PGL。

下面描述的节点参考范围、连通性和可达性在一个单独的对等组内。在发生分割的对等组,这意味着单独的分割区。

1) PGL 选举数据结构。

每个节点由一个单独的对等组领导选举数据结构，它由下列项目构成：

State:

对等组领导的选举 FSM 的操作状况，细节在 5.10.1.1 2) 节描述。

PreferredPeerGroupLeader:

这个节点认为应该是对等组领导的节点的 ID。为了选择优选的对等组领导，节点比较领导权优先级和 PTSE 内通告的节点 ID(在这个节点的可达节点拓补数据库中)。见 5.10.1.1 4) 节的动作 PGL4。

PreferredPGLLeadershipPriority:

优选的对等组领导的领导权优先级。

SearchPeer 定时器:

间隔定时器，在 InactivityFactor 乘 HelloInterval 之后超时。在定时器启动之后，节点认为没有对等实体，且立即启动选举过程。

PGLInit 定时器:

在 PGLInitTime 之后启动的间隔定时器。当定时器超时之后，启动选举过程，节点选择优选的对等组领导，通过产生一个新的节点信息 PTSE 实例而通告它的选择。PGLInit 定时器用于保证：为拓扑信息传播给整个对等组，在等待充分的时间后，每个节点只投一票。

OverrideUnanimity 定时器:

在 OverrideDelay 之后超时启动的间隔定时器，它用于防止节点长时间地等待意见一致的投票。

ReElection 定时器:

当到 PGL 的连接丢失时，启动的间隔定时器，且在 ReElectionInterval 之后超时。超时后，选择过程必须重新启动，节点选择一个新优选的对等组领导，用节点信息 PTSE 的新实例通告它的选择。ReElection 定时器用于对等组的每一节点在丢失连接时，一直到对等组的节点有机会接收被更新的拓扑信息才为 PGL 投票，这样改善了系统的稳定性。

2) PGL 选举状态。

在这节里描述对等组领导选举 FSM 可以得到的状态。图 19 是状态变化的示意图，箭头线标识引起状态变化的事件，这些事件在 5.10.1.1 节的 3) 条中描述。对于状态变化和每一状态变化涉及的动作的详细描述，见 5.10.1.1 节的 4) 条。

Starting:

状态机的初始状态。

Awaiting:

节点至少在一条链路上已经启动 Hello FSM，还没有找到对等实体。

AwaitingFull:

至少已经找到一个相邻对等实体，还没有完成数据库同步过程。

InitialDelay:

至少与一个相邻对等实体已经完成数据库同步，且已经启动 PGLInit 定时器。节点在选择和通告它优选的 PGL 之前必须等 PGLInitTime。

Calculating:

节点正处于计算选择哪个 PreferredPGL，这是一个瞬间状态；一旦节点对 PGL 作出选择，就退出这个状态，选择是通过执行 ChoosePreferredPGL 算法而完成，细节见下面。

calculating 状态的下一状态转换依赖于的节点选择自身作为 PGL 还是选择其他节点作为 PGL。产生下列事件之一将会影响状态的转换：

——如果选择其他一些节点，产生 PreferredPglNotSelf 事件。

——如果节点选择了自身，产生 PreferredPglSelf 事件。

OperNotPGL:

这个节点不是对等组领导，它继续检查其他节点发出的 PTSE 以确定哪个节点具有最高优先级作为 PGL。

OperPGL:

这个节点是对等组领导，它继续检查其他节点发出的 PTSE 以确定是否另一节点的优先级比它高。

AwaitUnanimity:

节点已经选择自身作为对等组领导。一旦进入这个状态节点必须立即查证它是否已经被一致地选为 PGL，如果是这样，产生 Unanimity 事件。在宣布它是对等组之前，要等待 Unanimity 事件或 OverrideUnanimity 定时器超时。

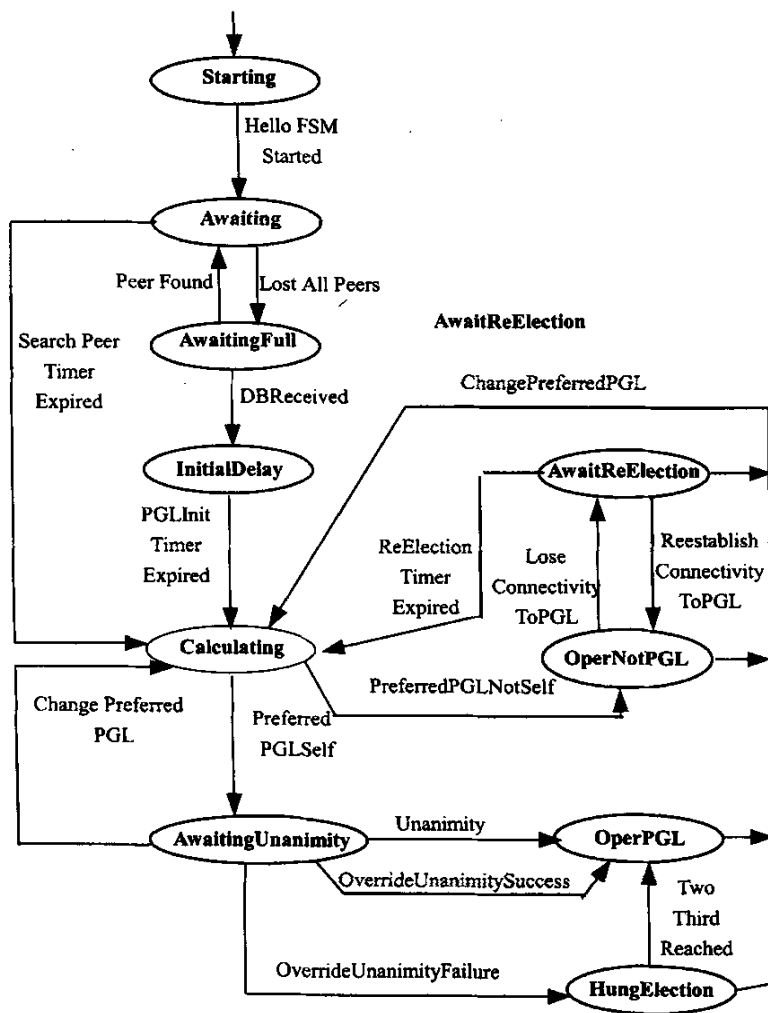


图 19 PGL选举FSM

HungElection:

节点已经选择它作为对等组领导，但是在 OverrideUnanimity 定时器超时之后，少于 2/3 的其他节点通告它作为它们优选的 PGL，这可能来自拓扑或网络参数的变化。在这种情况下，要由自身恢复，即这个节点改变优选的对等组领导或其他节点接受它作为 PGL。这种情况也可以是发生故障交换机或链路的结果。

AwaitReElection:

节点已经丢失到现行 PGL 的连通性，ReElection 定时器已经启动。在定时器超时之前如果还没有建

立连通性，重新进行选举。

3) 引起相邻对等状态变化的事件。

几个事件可能引起状态变化，这些事件可能由与对等组领导选举有关的程序引起，尤其是接收包含节点信息组的 PTSE，它们也可以被 Hello 状态机和相邻对等状态机内的变化所触发，事件如图 19 所示，事件发生后的详细状态机的变化和采取的动作，见 5.10.1.1 节的 4)条。

HelloFSMStarted:

在链路上已经启动第一个 Hello 状态机。

PeerFound:

Hello 状态机已经到达状态 2-way Inside。

LostAllPeers:

处于状态 2-way Inside 的最后余留的 Hello 状态机已经离开那个状态。

SearchPeerTimerExpired:

SearchPeer 定时器已经超时，没有找到对等实体，对等组领导选举立即启动。

DBReceived:

相邻对等实体状态机已经到达 Full 状态。

PGLInitTimerExpired:

PGLInit 定时器已经超时，节点现在能够选择和通告它优选的对等组领导。

PreferredPGLNotSelf:

节点进入 calculating 状态后，确定它优选的 PGL 不是它自己。这发生在领导权优先级等于 0 或对等实体具有较高优先级的情况下。

Unanimity:

节点优选的 PGL 是它自己，对等组内的其他节点在它们的 PTSE 内指出它们优选的 PGL 是这个节点。

OverrideUnanimitySuccess:

OverrideUnanimity 定时器已经超时，节点优选的 PGL 是它自己，对等组内 2/3 或更多的节点在它们的 PTSE 内指出它们优选的 PGL 是这个节点。

OverrideUnanimityFailure:

OverrideUnanimity 定时器已经超时，节点优选的 PGL 是它自己，对等组内少于 2/3 的节点在它们的 PTSE 内指出它们优选的 PGL 是这个节点。

PreferredPGLSelf:

节点进入状态 calculating 后，确定它应该是 PGL。

ChangePreferredPGL:

接收到新的信息改变了节点优选的 PGL(例如：一个新的高优先级节点出现；现行的 PGL 比它的优先级低；或管理提出这个节点的优先级高于现行 PGL 的优先级)。

LoseConnectivityToPGL:

已经丢失到对等组领导的连接。

注：确定节点到 PGL(运行 PGL 选举的其他任意节点)是否有连通性需要特殊的路由计算。特殊情况下，它必须忽略对等组内对链路资源和策略限制，以确定到 PGL 是否有连通性。除为 PGL 选举在节点 IG 设置非转接标志的节点之外，评价 PGL 的连通性，必须考虑链路两端节点通告的所有链路和对等组内被通告的所有节点。

ReestablishConnectivityToPGL:

已经重新建立到对等组领导的连通性。

ReElectionTimerExpired:

ReElection 定时器超时，必须重新选举。

4) PGL 选举状态机。

有限状态机是一个二维表，如表 18 所示。表的顶部是状态，左侧是事件。每对状态和事件交叉在表的表格，表格表明什么状态转换应该发生和采取什么动作。例如，对于 **HelloFSMStarted** 和 **Starting** 的事件状态对，表格中为“PGLE1”，“Awaiting”。**Awaiting** 是新的状态，“PGLE1”为采取的动作。在表 19 中可以找到动作。

表 18 PGL 选举 FSM 第一部分

状态 动作 状态 事件	Starting (注 1)	Awaiting	AwaitingFull	InitialDelay	Calculating
HelloFSMStartrd	PGLE1 Awaiting	FSM Err	FSM Err	FSM Err	FSM Err
PeerFound	FSM Err	PGLE2 AwaitingFull	PGLE0 AwaitingFull	PGLE0 InitialDelay	PGLE0 Calculating
LostAllPeers	FSM Err	FSM Err	PGLE3 Awaiting	PGLE0 InitialDelay	PGLE0 Calculating
SearchPeerTimerExpired	FSM Err	PGLE4 Calculating	FSM Err	FSM Err	FSM Err
DBReceived	FSM Err	FSM Err	PGLE5 InitialDelay	PGLE0 InitialDelay	PGLE0 Calculating
PGLInitTimerExpired	FSM Err	FSM Err	FSM Err	PGLE4 Calculating	FSM Err
PreferredPGLNotSelf	FSM Err	FSM Err	FSM Err	FSM Err	PGLE7 OperNotPGL
Unanimity	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err
OverrideUnanimitySuccess	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err
OverrideUnanimityFailure	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err
TwoThirdReached	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err
PreferredPGLSelf	FSM Err	FSM Err	FSM Err	FSM Err	PGLE9 AwtUnanimity
ChangePreferredPGL	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err
LoseConnectivityToPGL	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err
ReestablishConnectivityToPGL	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err
ReElectionTimerExpired	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err

注 1: 为了 PGL 选举 FSM 的描述，假定在第一个 Hello FSM 启动之前，就生成 PGL 选举 FSM 的实例。

表 19 PGL 选举 FSM 第二部分

事件	状态		OperNotPGL	OperPGL	AwtUnanimity	HungElection	AwtReElection
	动作	事件					
HelloFSMStartrd	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err
PeerFound	PGLE0	PGLE0	PGLE0	PGLE0	PGLE0	PGLE0	PGLE0
	OperNotPGL	OperPGL	AwtUnanimity	HungElection	AwtReElection		
LostAllPeers	PGLE0	PGLE0	PGLE0	PGLE0	PGLE0	PGLE0	PGLE0
	OperNotPGL	OperPGL	AwtUnanimity	HungElection	AwtReElection		
SearchPeerTimerExpired	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err
DBReceived	PGLE0	PGLE0	PGLE0	PGLE0	PGLE0	PGLE0	PGLE0
	OperNotPGL	OperPGL	AwtUnanimity	HungElection	AwtReElection		
PGLInitTimerExpired	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err
PreferredPGLNotSelf	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err
Unanimity	FSM Err	PGLE0	PGLE8	PGLE8	PGLE8	PGLE8	FSM Err
		OperPGL	OperPGL	OperPGL			
OverrideUnanimitySuccess	FSM Err	FSM Err	PGLE8	PGLE8	PGLE8	FSM Err	FSM Err
OverrideUnanimityFailure	FSM Err	FSM Err	PGLE0	PGLE8	PGLE8	FSM Err	FSM Err
			HungElection				
TwoThirdReached	FSM Err	PGLE0	PGLE0	PGLE0	PGLE8	PGLE8	FSM Err
		OperPGL	AwtUnanimity	OperPGL			
PreferredPGLSelf	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err
ChangePreferredPGL	PGLE4	PGLE6	PGLE4	PGLE4	PGLE4	PGLE4	PGLE4
	Calculating	Calculating	Calculating	Calculating	Calculating	Calculating	Calculating
LoseConnectivityToPGL	PGLE10	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err
	AwtReElection						
ReestablishConnectivityToPGL	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err	PGLE11
							OperNotPGL
ReElectionTimerExpired	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err	FSM Err	PGLE4
							Calculating

FSM Err

动作：协议错误，不应该发生。

PGLE0

动作：什么也不作。

PGLE1

动作：启动了初始值为 InactivityFactor 乘 HelloInterval 的 SearchPeer 定时器。

PGLE2

动作：停止 SearchPeer 定时器。

PGLE3

动作：用初始值 InactivityFactor 乘 HelloInterval 重新启动 SearchPeer 定时器。

PGLE4

动作：如果 OverrideUnanimity 定时器在运行就停止它；如果 ReElection 定时器在运行，就停止它。节点按照 5.10.1.1 节的 6) 条描述，对 $\text{PreferredPeerGroupLeader}$ 的值重新评价。

PGLE5

动作：启动了初始值为 PGLInitTime 的 PGLInit 定时器。节点为了通告领导权优先级，产生一个包含节点信息组的 PTSE ，“我是领导”比特设为 0，优选的对等组领导节点 ID 设为 0。

PGLE6

动作：通过完成下列动作，节点停止 PGL 的行为，去活父 LGN 。

- 必须产生节点信息 PTSE 的新实例，根据动作 PGLE4 给出的程序，更新优选的对等组领导 ID 区域。节点必须通告起先领导权优先级，关闭“我是领导”比特。
- 启动去活 LGN 的过程，过程涉及下列动作：
 - 1) 如果父 LGN 是对等组的 PGL ，通过在本等级完成动作 PGLE6 而停止作为 PGL 的行为。
 - 2) 完成 5.5.6.3 节项目 c) 的程序，使节点停止了作为 PGL 。
 - 3) 父 LGN 停止供给 PTSE 到子对等组。

PGLE7

动作：如果优选的对等组领导不同于先前通告的对等组领导，必须产生节点信息 PTSE 的新实例，其中带有优选的对等组领导 ID，“我是领导”比特设为 1。

PGLE8

动作：通过采取下列动作节点开始 PGL 的行为。

- 节点把领导权优先级加 $\text{GroupLeaderIncrement}$ ，这个较高的优先级用于将来计算谁是 PGL 。
- 如果 OverrideUnanimity 定时器仍在运行，则停止之。
- 产生节点信息 PTSE 的新实例，其中带有更新以后的领导权优先级，“我是领导”比特设为 1，以指出节点要求作为对等组领导，较高级的对等组也包括在 PTSE 内。
- 父 LGN 被示例，在适当的时间引起下列动作的发生：
 - 1) 在 5.5 节描述的父对等组内，父 LGN 向相邻对等节点建立基于 SVCC 的 RCC 。
 - 2) 一旦在较高级对等组完成对等组选举，结果就包括在这个节点的节点信息 PTSE 内的较高一级节点信息中。
 - 3) 从较高级对等组接收的 PTSE 在这个对等组泛播。

注：“我是领导”比特允许边界节点知道什么时候 PGL 稳定，能够用 Hello 公布发送到域外相邻点。与成为对等组领导相关的领导权优先级，它的增加提供了关于谁是 PGL 的一些稳定性。

PGLE9

动作：必须产生节点信息 PTSE 的新实例，其中带有优选的对等组领导 ID，“我是领导”比特设为 0。启动初始值为 OverrideDelay 的 OverrideUnanimity 定时器。

注：在这个动作的时候，如果所有节点对于这个节点作为 PGL 没有异议，由于立即将转换为状态 OperPGL 必然将产生另一节点信息 PTSE，所以产生新的节点信息 PTSE 是任选。

PGLE10

动作：用初始值 ReElectionInterval 启动 ReElection 定时器。

PGLE11

动作：停止 ReElection 定时器。

5) 发送节点信息 PTSE。

在对等组领导选举开始阶段，节点必须产生包含节点信息组的 PTSE。当节点发生变化或定期地防止节点老化，必须重新产生 PTSE。

在状态 Awaiting、AwaitingFull 和 InitialDelay，当节点具有为 PGL 选举设置的非转接标志，优选的对等组领导节点 ID 区域必须设置为 0。进一步，如果节点具有为 PGL 选举设置的非转接标志，它必须设置领导权优先级为 0；在状态 Calculating、AwaitingUnanimity、AwaitReElection、OperNotPGL、OperPGL 和 HungElection 必须用数据结构区的 PreferredPeerGroupLeader(见 5.10.1.1 节的 4)条的动作 PGLE4 设置这个字段。“我是领导”比特只有在状态 PGL 必须设置。

6) 选择优选的 PGL

每次节点接收和接受包含节点信息组的 PTSE，它必须重新评价 PreferredPeerGroupLeader 的值，当对等组内另一节点的连通性被建立或丢失时，节点也重新评价 PreferredPeerGroupLeader。

节点比较对等组内所有可达节点的领导权优先级(包括自己)。应该注意到，这种比较不考虑出现在数据库中但没有到这个节点的连通性 (LoseConnectivityToPGL 事件的描述中阐述了确定连通性的规则) 的节点，以及最高领导权优先级为 0 的节点。具有最高领导权优先级的节点被选为优选的 PGL。如果对等组内没有领导权优先级为非 0 的可达节点，则优选的 PGL ID 设为 0，节点 ID 用作平局打破者。出于这个目的，22 个八位位组节点 ID 被当作无符号整数，节点 ID 的第一个八位位组是最高有效八位位组。在平局的情况下，在最高领导权节点组中节点 ID 值最大的节点被选为优选的 PGL。

应该注意到，节点必须只考虑直接从其他节点接收的信息，特别是从其他节点通告优选的 PGL 必须不影响这个节点选择优选的 PGL。

5.10.1.2 领导权优先级

对等组领导权优先级的值在 $0 \sim \text{MaxLeaderShip}$ 之间，它的分配用来指出节点成为 PGL 相对的优先级。

1) 组领导的增加。

当节点推断它将作为对等组领导时，为了灵活性和稳定性，允许节点通告的对等组领导权优先级增加长量 GroupLeaderIncrement。

2) 可标识值和范围限制。

优先级值 0 为节点备用，表明节点不愿意或不能成为对等组领导。

节点不选择正在通告的优先级为 0 的节点作为对等组领导。如果对等组内所有节点正在通告的优先级为 0，则遵循 5.10.1.4 节描述的行为。

没有配置领导权优先级大于 $\text{MaxLeaderShip} - \text{GroupLeaderIncrement}$ 的节点。注意激活的 PGL 通告被配置的优先级加 GroupLeaderIncrement。

3) 范围。

在没有节点必须作为对等组领导的对等组中，能够作为对等组领导的节点的对等组领导权优先级应该在 $1 \sim \text{GroupLeaderIncrement} - 1$ 之间。

某些节点对第一次平局中的节点(即使它们中之一已经是 PGL)应该优选的情况下，应该分配优选节点的领导权优先级在 $2 \times \text{GroupLeaderIncrement} \sim 3 \times \text{GroupLeaderIncrement}$ 之间。在节点的这次平局中，为了稳定优选要服从已存在的对等组领导。

最后，如果需要一个严格的主控领导，节点的领导权优先级应分配在 $4 \times$

GroupLeaderIncrement~MaxLeadership- GroupLeaderIncrement 之间。这将产生一个对领导的第二次平局有优先的值，且仍然可增加。

4) 层次隐含。

在同一交换系统而在不同层次等级示例的两个节点，它们的对等组领导权优先级之间没有隐含关系。如果一个交换系统作为一个对等组的节点，它成为 PGL 有要求，这意味着它必须是它所属的所有较低级对等组的领导。因此，当一个节点成为某层次等级的 PGL 时，在同一交换系统示例的节点的所有祖先节点就停止存在。如果这些祖先节点中的任何一个 PGL，在对应的对等组中要强迫新的 PGL 选举。注意交换系统在每级的 PGL 领导权优先级可能不同。

作为以上叙述的特例，并不是所有对等组领导愿意成为较高级对等组的领导。如果节点不知道较高级对等组实体，它不能成为领导；出于本地原因，它也可以选择不愿意成为较高级的领导。在任何一种情况，节点在那个层次通告领导权优先级为 0。

5.10.1.3 缺省配置的一致性

节点缺省配置领导权优先级为 0。

注：这与最低级节点可以自动配置兼容，但是较高级节点需要主动配置，这是由于对等组领导必须知道对等组层次。

5.10.1.4 没有对等组领导的网络运行

本节规定的行为也应用在对等组领导选举之前。没有对等组领导的运行是网络足够小而作为一个单独的对等组不需要层次时，网络正常的运行方式。它也应用于层次的最高级。

对等组内部的运行不依赖于对等组领导的存在。这样对等组领导不存在时，对等组内部继续正常运行。节点必须传送 PNNI Hello，且确定本地的拓扑结构包括与之处于同一对等组相邻节点的标识。在对等组内必须正常地发送和接收 PTSP，计算对等组内部的路由，可以建立对等组的连接。因为对等组没有对等组领导，较高级的信息就不能传送到对等组内，对等组不参加相同相邻 PNNI 路由域内 PNNI 路由由较高级的运行。无领导对等组的节点允许通告域外可达信息，这种信息可用于到达路由域外的目的地。

无对等组领导的边界节点继续在外部链路上交换 Hello，由于没有较高级的信息，就传送空的节点层次列表，这阻止 Hello 状态机到达 CommonOutsideState。这样这种链路将不用于建立连接。

5.10.1.5 在外部链路上交换节点层次信息

一旦选出了对等组领导，对等组领导通过设置 PTSE 中节点信息“我是领导”比特为 1，而通告自己为对等组领导，边界节点与外部邻居交换 Hello 而在节点层次列表中接收到较高级的绑定信息。

如果多个节点要求成为这个对等组的对等组领导，边界节点通告来自被本地选为对等组领导的信息。对于较高级的对等组，如果多个节点要求成为 PGL，边界节点通告这个节点所在对等组的祖先 LGN 选出节点的信息。

5.10.2 通告的上行链路

当产生上行链路的通告并泛播对等组时，必须包含前后向拓扑状态参数，以便 PNNI 在通道选择期间能够考虑这条链路。下列是得到的拓补状态参数。

5.10.2.1 从较低级节点通告的上行链路

当外部链路的端节点发现另一端的节点与之处于不同的对等组时，在外部链路上的 Hello 分组包必须包含出对等组方向的资源可用性信息，ULIA 包含在 Hello 分组包中传送这种信息。

这里交换的拓扑状态参数与对等组内链路的通告中的拓扑状态参数基本一样，唯一不同之处在于这里的拓扑状态参数是在 Hello 分组包中携带而不是在 PTSE 中携带。

由于 Hello 只能在一条链路上传播，因此接收者把拓扑状态参数绑定到接收 Hello 端口对应的外部链路上，不要求额外的绑定信息。在这个端口接收 Hello 中的拓扑状态参数是外部链路入对等组方向的参数。

当层次启动时，外部链路成为上行链路的基础，上行链路的通告进入外部链路两端节点的对等组，边界节点使用外部链路入对等组方向的拓扑状态参数，作为计算上行链路入对等组方向(即向下方向)拓扑状态参数的基础。这样把上行链路通告注入对等组的边界节点，边界节点就规定了上行链路双向的拓

扑状态参数。

每一条外部链路可以产生一条上行链路，这种情况下在 Hello 中接收到的拓扑状态参数不进行任何改变而为对应的上行链路通告。作为选择，单独的上行链路可以是几条外部链路的汇集，这种汇集的程序与链路汇集的程序完全一致。

5.10.2.2 上行链路的反向信息

上行链路通告派生于外部链路或其他上行链路。

对于从外部链路派生的上行链路，反向信息(包括拓扑状态参数)由外部相邻节点规定，边界节点标识外部相邻节点包含的信息而把它作为上行链路通告的反向信息，这些信息包含在外部链路上发送 Hello 的上行链路信息属性(ULIA)中。Hello 中的 ULIA 封装了完整的上行链路通告的信息，这些信息由接收的外部相邻节点产生。

上行链路也可能从其他上行链路派生而来。当子对等组内的一个或几个边界节点正在向上行节点通告上行链路，而这个上行节点所处的等级要比这个 LGN 的对等组等级较高时，在这种情况下，这个 LGN 向同一个上行节点通告产生的上行链路。LGN 如何确定是否将多条较低级的上行链路汇集为一条较高级的上行链路的细节可参见 5.10.3.1 节。

单独外部链路派生的上行链路(derived uplink)或导出上行链路(inducing uplink)的情况下，必须为导出的上行链路的通告分别拷贝外部链路上接收到 Hello 中包含的 ULIA 或为派生的上行链路通告的 ULIA(独立于包含在 ULIA 内信息组的标签)。

在多条外部链路或导出上行链路存在于同一较高级节点的情况下，这多条链路可以汇集成一条来自这个节点公共的上行链路(见 5.10.3.1 节)。在这种情况下，为汇集上行链路产生的 ULIA 中包含的信息组是来自 Hello 消息中 ULIA 的个别信息组的集合。这些信息组分别是封装在外部链路接收 Hello 的 ULIA 或为导出的上行链路通告的 ULIA 中。包含在汇集 ULIA 中的任何未定义的信息组是否包含在产生的上行链路的通告中，是由与这些信息组相关的信息组标签控制。由于这个节点正在构造汇集的上行链路通告的拓扑状态参数，它也负责指出已构造的值什么时候发生重大变化。这意味着必须为汇集的上行链路的序列号产生适当的值。

如果节点理解上行链路的所有固定的信息组，那么节点只能在路由计算中使用上行链路，且在 DTL 中规定它们。此外，作为标签范围的标准准则的例外，节点也必须理解上行链路的 ULIA 内的所有固定信息组，例如，如果 ULIA 中包装的信息组不能被标识且标记为固定，路由计算时决不能使用对应的链路。

5.10.3 拓扑汇集

拓扑汇集是为了减少父对等组通告的信息量，而归纳子对等组拓补信息的过程。PNNI 支持大的网络需要有效的汇集。

链路汇集是把几条并行的链路表示为一条单独的较高级链路，这条链路的拓扑状态参数来源于被汇集的个别链路。

PNNI 复杂节点表示是描述逻辑节点内连通性的灵活方案。当逻辑节点产生复杂节点表示时，在表示的准确性和它的大小方面要作权衡。作为替换，也可以使用简单的节点表示，即整个 LGN 当作一个点，没有资源限制。

5.10.3.1 链路汇集和绑定信息

同样的两个对等组之间的一组外部链路的汇集是通过外部链路终结的边界节点的汇集令牌值配置。为了保证对等组领导发生变化时前后一致的行为，链路汇集的信息配置在边界节点而不是逻辑节点。具有相同汇集令牌值的一对逻辑组节点之间的所有链路，必须被逻辑节点通告为一条逻辑链路。汇集令牌的范围限制在一对逻辑组节点之间的链路，没有二意性的情况下同样的令牌值可以用于其他对节点之间的链路。

汇集令牌包含在外部链路连接的两个最低级相邻节点之间的 Hello 协议中传送。除非重新配置否则两个最低级相邻节点之间 Hello 中的令牌值不改变，这个值被认为是后续内容中的

ConfiguredAggregationToken (配置汇集令牌)。

一旦交换了配置汇集令牌, 每一个最低级相邻节点对一样的信息运行一样的算法, 来确定 DerivedAggregationToken (派生的汇集令牌) 的共同值, DerivedAggregationToken 是为对应的上行链路在 PTSE 中通告的值。因此最低级节点之间 Hello 包含配置汇集令牌的值, 逻辑组节点之间的 PTSE 和 Hello 包含派生的汇集令牌的值。

派生过程要设计成最小的配置以及对误配置的可接受行为。一条链路的配置汇集令牌的缺省值是 0, 它具有特殊的意义。任何其他值是通过配置建立, 0 意味着链路不关心汇集行为。在那种情况下节点将使用远端具有的任何值作为派生的汇集令牌值。如果没有作配置, 那么所有的链路用值为 0 的派生汇集令牌结束。

计算派生汇集令牌值的算法如下:

- 1) 如果相邻节点交换了等同的配置汇集令牌值, 那么这个值就是派生的汇集令牌值。
- 2) 如果一个相邻节点的配置汇集令牌值为 0, 而其他相邻节点的为非 0, 那么非 0 值就是派生的汇集令牌值。
- 3) 如果相邻节点的配置汇集令牌值不同且非 0, 那么派生的汇集令牌值是 0。

为了实现不汇集, 至少链路的一端需要配置。如果所有的链路具有 0 值的配置汇集令牌, 那么只有链路的一端需要重新配置使这条链路不汇集。

以上算法是连续的运行, 当配置汇集令牌发生变化, 立即在外部 Hello 中向最低级相邻节点传递新的值。如果这引起派生的汇集令牌值的任何变化, 必须发出所有 PTSE 新的实例, 它包含派生的汇集令牌值变化的链路。

对于最低级节点, 到同一较高级相邻节点且具有相同派生的汇集令牌值的上行链路可以按照节点的处理汇集或不汇集; 然而具有不同派生的汇集令牌值的上行链路不能汇集。如果最低级节点有相同级的相邻逻辑节点, 那么到那个具有相同派生的汇集令牌值的所有水平链路, 必须被汇集且被通告为一条单独的链路。

对于逻辑节点, 到相同相邻节点且具有相同派生的汇集令牌值的所有链路必须被汇集且被通告为一条单独的链路。即如果对等组有多条上行链路到具有相同派生的汇集令牌值的相同上行节点, 那么表示对等组的逻辑组节点必须汇集这些上行链路成一条单独的上行链路或水平链路。由于配置汇集令牌值的缺省值为 0, 所以缺省的行为是汇集两个逻辑组节点之间的所有链路成一条派生汇集令牌值为 0 的单独链路。

派生汇集令牌值与远端节点 ID 一起作为等级层次之间的绑定信息, 这些值包含在 PTSE 内传递, 用来描述上行链路或水平链路。在上行链路 PTSE 中通告的值用于这个节点的父节点, 在上一较高级作进一步的汇集判决。此外, 为上行链路或水平链路通告的值用于边界节点的 DTL 处理期间。当入口边界节点处理接收到的 DTL 堆栈, 而最顶端 DTL 堆栈的现行入口为非 0 端口 ID 时, 穿过这个对等组被选择的通道, 必须结束在被端口 ID 指出的汇集的上行链路处 (即对等组出口的上行链路具有相同汇集令牌值作为 DTL 中规定的端口 ID 值)。

为了格式的一致性, 物理链路或 VPC 的水平链路的 PTSE 也携带派生汇集令牌值区域, 在传送时这个令牌必须设为 0。由于这条水平链路之下没有导出的上行链路, 这条水平链路通告的汇集令牌没有作用。

5.10.3.2 简单的节点表示

连接的端到端 QoS 和业务量参数受连接所穿越的每一节点和链路影响。当穿越节点不太影响连接的端到端参数时, 把这种节点构成一个单点是最简单的方法。这种表示简化了用最高顶点表示节点, 用弧形表示链路而表示出的网络拓扑, 如图 20 所示。这种表示特别对构造单独交换机上实施的最低级节点的模型有用, 这里对连在节点上链路的限制比对交换机本身的限制要重要。简单节点表示的编码见 5.8.1.2 节的 3) 条和 5.14.9.1 节。

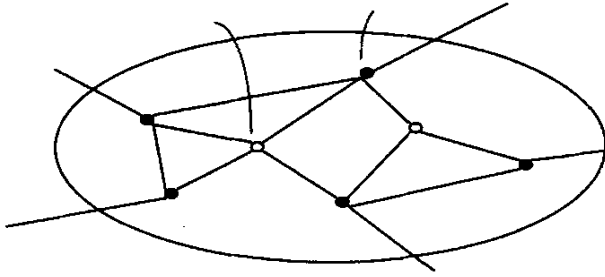


图 20 5个边界节点的对等组

5.10.3.3 复杂的节点表示

复杂的节点表示是提供有关逻辑节点状态信息的节点状态参数的集合，它用于表达穿越节点对连接的端到端参数有显著影响的情况。

为了适应穿越逻辑节点以及节点“内”的选路，使用具有统一“半径”的对称星型拓补。星的中心是逻辑节点的内部参考点，被认为是核；核和逻辑节点之间的逻辑连接被认为是轮幅。缺省的节点表示由每一节点状态参数单独的值构成，它给出逻辑节点入口或出口与核之间的任一方向的假设值。PNNI选路支持缺省的节点表示，如图 21 所示。

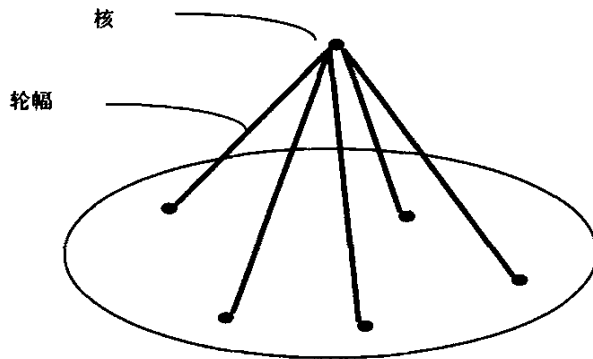


图 21 缺省的节点表示

对于逻辑节点的每一节点状态参数，“半径”来自逻辑节点的“直径”。对于节点的度量，“半径”是“直径”的一半；对于节点属性，“半径”的属性与“直径”的一致。PNNI选路没有精确地规定如何汇集来确定“直径”，保守的通告者可以采用最差的值，好进的通告者可以考虑平均值或更好的情况。

缺省节点表示最大限度地减少了通告的信息，然而不可能充分捕捉典型逻辑节点多连接或反映非对称的拓扑信息。

通常逻辑节点并不是理想的圆，PNNI选路允许与给定轮幅有关的拓扑状态参数，不同于缺省的“半径”。此外，也可以通告直接的端口到端口的连通性（称为“旁路”）。

用这种灵活性，节点可以通告实际的从对称星型结构到网状结构任何汇集拓补，表示非缺省节点的节点表示的连通性通告，称为“例外”，如图 22 所示。

PNNI选路的复杂节点表示能够按照下面描述的方式构造。

- 1) 用核表示对应节点“内部”轮幅连接逻辑节点的端口和核的星型拓补，覆盖了每个逻辑节点。每一个端口 ID 必须与标识链路或端口可达地址的端口 ID 相同。
- 2) 对于每一个节点状态参数，通告“半径”作为轮幅的缺省值。
- 3) 任何轮幅或一对端口之间的任何逻辑连接可以设计成“例外”。
- 4) 对于每一个“例外”，通告与它有关的全部节点状态参数；对于旁路，必须规定双向的节点状态

参数。

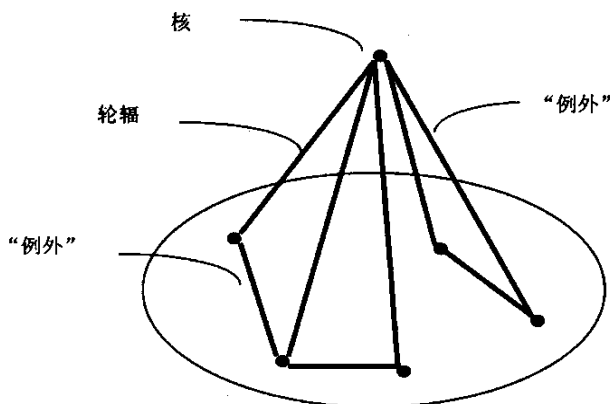


图 22 具有“例外”的复杂节点表示

5) 对于每一个作为“例外”而通告的轮辐，“例外”节点状态参数替换规定“例外”的方向上的缺省信息。

6) 从复杂节点表示中相联的任何几个旁路和最多两个轮辐（缺省或“例外”）得到穿越逻辑节点的通道。

用以上复杂节点表示，依据分配给“半径”和“例外”的参数值，节点可以选择保守地通告或好进的通告。

PNNI 选路不规定如何选择轮辐和旁路作为通告的“例外”。

5.10.4 向下的层次供给信息

以上描述的层次归纳允许层次最高级的节点计算到最高级对等组中任何目的地的路由，包括通过较低级（通过归纳地址前缀而公布）的系统可达。然而，需要 PNNI 网中所有的节点能够使呼叫选路到任何目的地，而不仅仅是到层次最高级的节点。这意味着较低级的节点必须知道描述层次较高级的拓扑信息。

这要求参加 PNNI 选路的所有节点不仅维护自身对等组拓扑数据库（和计算路由的能力）的信息，而且维护所有祖先对等组的拓扑数据库的信息。

要想所有节点直接计算合适的路由（即使是在归纳的较高级拓补中没有明确地表示的节点），较高级的 PTSE 不仅要泛播到它们对等组的所有节点，而且要泛播到后代对等组的所有节点。

向所有的后代对等组（即，向这个对等组中节点表示的较低级对等组中的所有较低级节点）泛播 PTSE 如下：

——当发起（originate）PTSE 或更新先前产生的 PTSE 时，较高级的节点泛播 PTSE 到较高级节点表示的对等组的 PGL，以及泛播到这级的所有相邻点。PGL 将在子对等组内泛播 PTSE。

——当接收到新或比拓扑数据库新的（而不是拓扑数据库的拷贝）PTSE 而泛播时，较高级节点泛播 PTSE 到较高级节点表示的对等组的 PGL，以及泛播到这级的所有相邻点而不包括起初发送这个 PTSE 的节点。PGL 将在子对等组内泛播 PTSE。

在给定对等组产生的 PTSE 决不泛播到较高级的对等组。作为代替，对等组的领导根据对等组内产生的 PTSE 归纳对等组的拓扑，然后父对等组级的 LGN 发起（originate）新的 PTSE 而泛播归纳结果。

在正常情况下，对等组领导有较高级产生的 PTSE 最新的实例；非正常情况下，对等组存在的较高级产生的 PTSE 的序号大于现行的序号，可能发生的经过是这样：当较低级对等组的节点与对等组隔绝，而被隔绝节点所知的 PTSE 已老化时，较高级对等组节点刷新和重新发起了 PTSE。当隔绝的较低级对等组与较低级对等组的其他部分重新得到连接时，它仍然可以具有 PTSE 旧的拷贝但可能是不合理的信息和大于现行序号的号码。

在这种情况下，对等组领导将最终得到不合理的 PTSE 且认识到 PTSE 不正确。在下面的情况时对等组领导检测到这种 PTSE：

- PTSE 封装在 PTSP，其中 PTSE 发起者的逻辑节点 ID 的等级比 PGL 所属对等组的等级高；
- 在 PGL 的 PTSE 数据库中没有实例或确定这个 PTSE 比 PGL 的数据库拷贝新。

当检测到这种 PTSE 时，对等组领导通过设置 PTSELifetime 为 ExpiredAge、设置 PTSE 序号为接收的 PTSE 的序号、向全对等组泛播 PTSE 来刷新了对等组的 PTSE。如果 PGL 在数据库中没有 PTSE 的拷贝，PGL 不需要采取进一步的动作。否则处理按照下面继续进行。

对等组领导需要保存那个拷贝和正在刷新的 PTSE 一直到被刷新的 PTSE 能够被取消（见 5.8.3.9 节）。起动刷新后，在试图泛播合理的实例之前，PGL 通过把对等组节点数乘 PTSERetransmission，得出的数分配给称为 TimeToFlush 的变量来确定合适的等待初始时间量。然后它起动时间为 TimeToFlush 的定时器，等待定时器的超时。如果 PGL 再一次接收到 PTSE，而这个 PTSE 是在定时器超时前它就试图通过用非 ExpiredAge 的泛播刷新的 PTSE 时，它又一次起动刷新用新值重新起动定时器，使用的新值是先前值的两倍与 8 倍的 PTSERetransmissionInterval 再乘以对等组的节点数之间较小的值。如果先前的值已经是 8 倍的 PTSERetransmissionInterval 再乘以对等组的节点数，应该把这个错误记录在网络管理。对等组的节点数意味着为 PGL 选举定义而连接的节点数（见 5.10.1.1 节）。如果定时器超时，那么合理的较高等级的 PTSE 重新泛播入对等组。

5.10.5 多等级层次的刷新 PTSE

正如 5.8.3.8 节的描述，由于 PTSE 中信息不合法发起者决定 PTSE 过早老化，或 PTSE 的 RemainingLifetime 到达 ExpiredAge 而 PTSE 自然地老化时，刷新 PTSE。

当节点刷新其中一个自身的 PTSE 时，它按照 5.8.3.8 节描述的程序向它所属等级中相邻的节点泛播超时的 PTSE。此外，如果节点不是层次的最低等级，这个节点向它表示的对等组的 PGL 泛播超时的 PTSE。按照 5.10.4 节的描述 PGL 在子对等组中泛播超时的 PTSE。

当 PTSE 老化（这个 PTSE 需属于另一节点），起初接收到 PTSE 的交换系统内的逻辑节点必须刷新 PTSE，这样保证完成最大范围的刷新。节点的刷新程序按照前面章节描述的刷新程序。

注：PNNI 规范不规定实施中如何组织 PTSE 入数据库。在实施中，交换系统内的逻辑节点可能使 PTSE 老化。以上所述的要求不排除逻辑节点（不包括起初从刷新超时 PTSE 接收的 PTSE）。当发生 PGL 接收对等组内起初从较高级发起的超时 PTSE，而这个 PTSE 的数据库拷贝还没有超时的情况时，按照 5.10.4 节描述的非正常情况处理。不同之处是当 PGL 接收刷新时，它不重新刷新 PTSE，但按惯例前转刷新的 PTSE，保持超时的拷贝和自身数据库的拷贝，设置 TimeToFlush 定时器，遵循 5.10.4 节描述的剩余程序。

5.11 对等组的分区

分区的对等组是指一个对等组由于偶然的原因被划分为两个或两个以上的不连续的部分。这种情况是由设备的偶然故障造成的（例如链路或交换局故障）。每个分区在这种情况下都作为一个对等组独立的工作。例如每个分区会选择出一个 PGL，这种情况暗示在高一级上出现了多个逻辑组节点（假设每个分区都选择了一个 PGL）。每个分区都必须在层次结构中被明确的表示，路由计算应能在一个包含多个分区的对等组情况下正确工作。

5.11.1 层次结构中分区的表示

每个分区选择一个对等组领导（PGL）。如果对等组出现分区，则 PGL 选择算法应保证每个分区选择其 PGL（假设总存在一个可以担当 PGL 的节点）。同样的，如果对等组的两个分区又重新合并，则 PGL 选择算法应保证只剩下一个节点可以作为 PGL。PGL 用来确定其父对等组的标识，并且表示在父对等组中的分区。

在某些情况下，分区可以没有对等组领导（PGL）（例如：一个小的分区可能没有包含具有担当 PGL 能力的节点）。在这种情况下，该分区应按照“无对等组领导”方式进行工作，并同其他的 PNNI 的路由域有效的隔离。具体描述参见 5.10.1.4 节。

为了保证父对等组正确的工作，父对等组中的每个节点必须有一个唯一的节点标识符（ID）。这个

ID 可以通过把对等组领导 (PGL) 中的 48bit 的 ESI 包含到节点 ID 部分中的“FLAT ID”获得 (参见 5.3.3 节)。这就保证了如果发生分区, 则每个分区将在父对等组中使用带有唯一的节点 ID 的逻辑组节点来自动的表示。这就要求可以作为 PGL 的节点必须具有 48bit 的 ESI, 且该值在父对等组中应当是唯一的。如果使用了其他算法生成节点 ID, 则必须在现存的分区中提供唯一的 ID。

注: 使用 ESI 可能出现 ID 不唯一的情况, 特别是在跨国的高级对等组间。

5.11.2 出现分区后的选路

对等组中的每个给定的分区都应该按照单独的对等组进行工作, 呼叫应通过父对等组正常选路并通过不同的分区。同样的, 起源于对等组分区至对等组外的呼叫应能正确的选路。但在呼叫至分区的对等组中的选路会存在一些问题。

在许多情况下, 对等组中端系统可达性的设置是使用一个地址前缀归纳来实现的 (或者是地址前缀的一个小号码)。在分区的对等组中存在的问题是每个分区通告的可以到达的地址前缀都相同, 而实际上只有一些 (不是全部) 系统的地址与某些特定分区中的前缀相匹配。这时就可能出现选路到一个分区的对等组的系统时, 却出现呼叫截止在一个错误的分区中。分区对等组的进入边界节点应确定呼叫的目的地地址是否在该分区中可以实际到达。如果未在该分区中, 且在网络中通告的归纳地址都是相同的, 则该呼叫回选, 并由 DTL 起源节点重新构造一个新的 DTL 至另外一个分区, 具体内容详见附录 B。

5.11.3 如何汇集成一个 PGL

对等组内产生的 PTSE 均标记有对等组 ID, 所有对等组的分区都带有相同的对等组 ID。因此当一个对等组产生分区时, 则节点的拓扑数据库可能包含了在分区中节点的 PTSE, 以及其他分区内节点的 PTSE (也可能过期)。

当一个 LGN 汇集且/或归纳了子对等组 (分区) 的拓扑状态信息, 它必须只使用属于该分区的子对等组的拓扑状态信息。同样的, 如果运行 PGL 选择算法, 则该算法应当只考虑属于其分区的节点。用来确定哪些 PTSE 子集属于其分区内节点的算法规定见 5.10.1 节中 (例如节点产生的到与其有连通关系的节点的 PSTE)。

5.12 当拓扑数据库处于过负荷时节点的操作

当一个节点不能存储完整的拓扑数据库时, 则该节点处于拓扑数据库过载状态, 并且需要正常的 PNNI 算法来进行维护。这种情况发生在节点内没有足够的内存可以用来存储当前对等组内激活的 PTSE 数据 (也包括用来说明祖先对等组拓扑的 PTSE) 时。

当一个节点处于拓扑数据库过载状态, 就意味着它已经不能按照一个正常的 PNNI 节点进行工作, 然而它在许多方面仍期望继续正常操作, 并且符合同网络中其他节点正常操作的要求, 不必同网络彻底分开。在本节中描述当处于拓扑数据库过载状态时的要求和建议。

5.12.1 处于拓扑数据库过载状态的操作要求

5.12.1.1 最小状态要求

按照如下指出的原则, 一个节点只要能构保存一个拓扑数据库的子集, 就允许它继续工作。但是它必须始终能够存储并通告如下内容:

- 1) 用于描述节点自身的所有要求的状态。
- 2) 用于描述到相邻节点的链路的所有拓扑状态。

如果该节点不能满足要求 1), 则该节点必须被停止。如果不能满足要求 2), 该节点不能提供链路, 因为它已不能存储相关的状态。

5.12.1.2 不允许的功能

当处于拓扑数据库过载状态, 节点的正常操作中应限制如下功能:

- 1) 该节点应不支持外部链路 (OUTSIDE LINK), 而其动作应与不能作为边界节点的情况相同。参见 5.6.2.1 节的 4) 条中状态机表的注 1。
- 2) 该节点不能作为 DTL 的起源节点。例如: 不允许产生新的去话呼叫。

5.12.1.3 PGL 生成

在该状态下，节点必须通过设置节点标志中 PGL 选举标志为不能转接来通告该节点所处的状态。也必须停止 PGL 生成的有限状态机（FSM），并且通告其领导权优先级和优选的 PGL 为 0。

5.12.1.4 PTSP 的接收和传送

收到 PTSP 后，必须按照 5.8.3.3 节规定的正常原则发送 PTSE 证实。证实的应用取决于该节点是否能存储收到的 PTSE。这个要求保证了相邻节点的 PTSP 的发送处理不会比在正常操作下要求更多的资源（链路带宽、重发表空间）。

5.12.1.5 周期的重新同步

为了从造成拓扑数据库过载的暂时的网络故障中恢复，该节点必须周期地去尝试与其相邻节点重新同步。这种重新同步是通过重新启动其相邻同等有限状态机（好像发生了 DS 不匹配事件），这样会使其与相邻的每个节点的数据库取得同步。

如果数据库同步处理完成后并且该节点可以存储所有的 PTSE，它通过清除节点标志中的“不能成为 PGL 生成标志”，返回一个正常操作指示，并返回到正常 PNNI 算法操作状态。否则该节点继续保持拓扑数据库过载状态。

周期的重新同步过程在结构化常量 OVERLOADRETRYTIME 规定的时间内进行。

5.12.2 处于拓扑数据库过载状态的操作建议

本节给出了当一个节点处于拓扑数据库过载状态时进行其他操作的建议。在实施中采用这些建议可以使得数据库过载情况对网络其他部分的不利影响最小。

5.12.2.1 呼叫处理

假设拓扑数据库所要求的内存与呼叫和连接控制所使用的内存是相互独立的。如果不是这种情况，为了支持输入网络管理接入，则实施至少应预留最小数量的呼叫状态内存。

当节点进入拓扑数据库过载状态时，现存的呼叫应当在一定范围内尽可能的被保证。

一个新的呼叫应当在一定范围内尽可能的被允许，尤其是应当接收在该节点终接的呼叫，因为远端带内网管接入可能需要建立一条到过载节点的 SVCC。

转接呼叫应当被正常的处理，因为它只需要按照 SETUP 消息中的 DTL 进行处理。

因为拓扑数据库不完整，因此 DTL 的生成必然不准确（对于呼叫产生节点或进入边界节点处理）。但是允许 DTL 生成仍旧是有用的，因为尽管用于选路的拓扑数据库不完整，但该信息仍旧是有效的（尽管它们可能不是最优的）。

5.12.2.2 拓扑数据库的处理

处于拓扑数据过载状态的节点应当保存被隔开的当前激活的 PTSE 的子集，虽然保留最小的拓扑数据库是合法的，但为了降低对网络的其他节点的分裂性影响应当尽可能保留激活的 PTSE。任何实际保存在拓扑数据库中的 PTSE 必须应按照发送 PTSP 的正常原则传送到相邻点。

上面的原则就可能造成当一个节点处于拓扑数据库过载时导致对等组分区，某种意义上可能会出现两个对等组领导 PGL，甚至可能从一个分区到另一个分区建立呼叫。这种情况比较少见但却是可能出现的，因为一方可能会认为过负荷的节点是“限制转接”，如果出现这种情况，那么从一个分区到另一个分区的呼叫可能需要从另外一个对等组进行选路，参见 5.12.2 节。但保留的拓扑可能不允许这么做。

5.13 通道选择

在选择一条到目的地 ATM 地址的通道时，始发节点将选路到一个与目的地地址前缀最大匹配的节点。如果只有祖先节点与目的地前缀是最大匹配的，则目的地不可达。只有当几个节点通告的等长匹配前缀都大于其他的通告前缀，这些节点可以作为到目的地的本地计算节点。在通告的等长的匹配节点中应忽略任何祖先节点，并且在剩余的节点中选择一个。

PNNI 允许到一个组地址建立一个点到点的连接（例如：ATM 的任播能力 ANYCAST）。在这种情况下，应当选择一个到组中的一个成员的路由。这是组地址通告的最大匹配的应用结果。该组成员在指示的连接范围中必须是可达的。

所有的地址（单独的或组地址）具有一个通告范围。在构造通道时，一个节点必须确保目的地地址具有的通告范围应等于通过 PNNI 路由域的通道范围。地址通告的处理和归纳应保证地址不被通告到该范围外。因此这种检查对 DTL 起源节点并无影响，但对入口边界节点则会造成影响。这说明在当前 DTL 的最后对等组中选择一个特定目的地时，进入边界节点必须进行范围检查。特别应当说明的是范围检查的优先级应当高于最长前缀匹配检查。

如果 SETUP 消息中包含了转接网选择信息单元，则该呼叫的选路应向与转接网信息单元中规定的转接网有可达性的节点进行。在这种情况下通道选择根据规定的转接网选择进行，也可以任性地考虑目的地地址。同包含在外部可达地址信息组内的转接网 ID 信息组一样，接入到特定转接网的节点应当被通告。对于同一个转接网要进行多级通告，RAIG 和/或与转接网相关的可达地址将用来选择到转接网的入口。

如果相同的转接网对于多个节点可达，则这种情况对于要在最近的节点离开 PNNI 网络时就不利（代价过大）。然而做为特定的地理区域，它可能被不同的网络所覆盖，因此对于要在更远的节点离开 PNNI 网络时则更不利。关于这个问题，目前 PNNI 1.0 尚无法解决。

对于点到多点连接时的路由选择，节点应当选择一条通道从而构成一个树形连接。一个点到多点连接的任意两个分支都不可以共用同一条链路，除非该节点是分支上的节点，否则他们也不可以共用一个节点。

注 1：该原则造成的影响是一个节点要保持一个点到多点的树形表示，而不只是保持它叶结构。

注 2：如果存在并行链路，则上面提到的限制避免了多于一条的链路被用于一个点到多点的呼叫，这是因为如果多于一条的并行链路被使用，则呼叫肯定早就被进行了分支，并且在水平链路或上行链路上重新合并。

PNNI 路由提供了一个避免点到多点的连接树在一个不能支持附加分支的节点上被分支。这些节点都被通告为限制分支节点，点到多点的连接必须避免在这样的节点上进行分支。

注 3：DTL 中最大的单元数量（当前为 20）和 SETUP 消息中包含的 DTL 的数量（当前为 10）限制了一个对等组的直径和体系的等级。

5.13.1 通道选择的合格实体

当计算通过一个 PNNI 路由域的通道时，选择一组串联的 PNNI 实体作为通道。这些实体包括节点、端口、水平链路、上行链路、轮辐或旁路通过相连的复杂节点、和/或从节点端口到内部或外部可达地址和转接网的路由。对于水平链路、轮辐或旁路通过的复杂节点，如果该实体被两个端点通告，且具有一组由端点对其出局方向通告的请求的业务类别的拓扑状态参数，则包含在通道中的实体均是合格实体。对于内部或外部可达地址和转接网，在对可达地址和转接网通告中出现的 PTSE 完全可以认为合格（拓扑状态参数对于外部可达地址和转接网任选）。如果内部或外部可达地址或转接网选择信息中包含了拓扑状态信息，如果它可以支持请求的业务类，则只对包含的内容有效。当拓扑状态信息参数对所有业务类都不存在时或者是对特定的业务类别在双向上都存在时，一个业务类别才可以被支持。

当从计算一个边界节点到其他节点的通道时，需要考虑上行链路，上行链路必须使用包含在上行链路的 PTSE 中的对各个方向的一组拓扑状态参数和支持的请求业务类进行通告。DTL 生成节点在计算通过 PNNI 路由域的通道时应当考虑上行链路，但应有如下附加限制：对应的水平链路必须被一个计算到上行节点路由的节点祖先通告，这个要求能够保证经过上行链计算出路的路由是根据当前通过或越过上行节点的拓扑信息。此外为了解在哪个端口进入上行节点，在计算时还需要水平链路的信息。例如：在两个相邻的对等 LGN 之间没有基于 SVCC 的 RCC，在收到这个拓扑的最后的 PTSE 后，通过和越过上行节点的拓扑可能已经有效地修改，这样会导致大量的呼叫建立消息在通过上行节点至目的地时失败。在两个 LGN 间建立基于 SVCC 的 RCC 后，在计算两个 LGN 的路由时，对于 DTL 的起源者来说在挑选路由时对相关水平链路的要求就相对容易了。

5.13.2 链路限制、节点限制和通道限制

PNNI 选路用来确定可以满足性能限制的通道，但不需要对于任何预确定的性能标准进行优化。性能限制可以通过以下 3 种之一方法实现：链路限制、节点限制和通道限制。对于链路限制，没有附加的

链路参数需要使用。对于节点限制，没有附加的节点状态参数需要使用。对于通道限制，则需要使用附加的链路状态参数。链路限制和节点限制在进行通道选择剪除网络图时使用。PNNI 选路支持的链路限制和节点限制实施包括信元丢失率和通用 CAC 参数（例如：可用信元率、信元率幅度和变化系数等）。PNNI 选路支持的节点限制包括限制通过和限制分支。PNNI 选路支持的通道限制实施包括管理权重、最大信元传递时延和信元时延变化等。

5.13.3 CBR 和 VBR 业务的 CLR 选择

连接请求 CBR, rt-VBR, nrt-VBR 业务类是需要规定 CLR 指标，根据承载业务类、业务量参数和 QoS 参数的组合，CLR 的指标与 CLP_0 和 CLP_{0+1} 的业务流有关。用来确定哪种 CLR 指标适用的规定见 UNI 4.0 信令，对于给定连接请求的 CLR 指标按照这种原则与相关链路和节点通告的 CLR 进行比较。

5.13.4 CBR 和 VBR 业务的通用 CAC 算法

5.13.4.1 GCAC 算法的参数选择

在下文中描述通用 CAC 算法。所有的连接都使用两种业务量参数进行划分：PCR 和 SCR。这两个参数从 ATM 业务量描述字信息单元中定义的业务量参数中得到。对 CBR 和 VBR 业务类，UNI 3.1 中的表 5~表 7 定义了以下在 ATM 用户信元率信息单元中的 6 种允许的业务量参数组合（PCR：峰值信元率、SCR：持续信元率，MBS：最大突发量）。

- 1) PCR(CLP=0) ; PCR(CLP=0+1)
- 2) PCR(CLP=0) ; PCR(CLP=0+1) ; 请求标注
- 3) PCR(CLP=0+1) ; SCR(CLP=0) ; MBS(CLP=0)
- 4) PCR(CLP=0+1) ; SCR(CLP=0) ; MBS(CLP=0) ; 请求标注
- 5) PCR(CLP=0+1)
- 6) PCR(CLP=0+1) ; SCR(CLP=0+1) ; MBS(CLP=0+1)

1)和 3)组合产生的业务量特征与 2)和 4)组合产生的业务量特征的相同。所以，GCAC 必须忽略“标注”子字段。

对于候选通道上的任何拓扑单元，如果 RAIG CLP 比特设置为 0，则表示可以对 $CLP=0$ 的业务量分配资源。在这种情况下，GCAC 中使用的连接的 PCR 和 SCR 取值应按照表 20 进行设置。

表 20 对于 $CLP=0$ 的 PCR 和 SCR 取值

参数组合	PCR	SCR
1, 2	PCR(CLP=0)	PCR(CLP=0)
3, 4	PCR(CLP=0+1)	SCR(CLP=0)
5	PCR(CLP=0+1)	PCR(CLP=0+1)
6	PCR(CLP=0+1)	SCR(CLP=0+1)

对于候选通道上的任何拓扑单元，如果 RAIG CLP 比特设置为 1，则表示可以对 $CLP=0+1$ 的业务量分配资源。在这种情况下，GCAC 中使用的连接的 PCR 和 SCR 取值应按照表 21 进行设置。

表 21 对于 $CLP=1$ 的 PCR 和 SCR 取值

参数组合	PCR	SCR
1, 2, 3, 4	PCR(CLP=0+1)	PCR(CLP=0+1)
5	PCR(CLP=0+1)	PCR(CLP=0+1)
6	PCR(CLP=0+1)	SCR(CLP=0+1)

5.13.4.2 复杂的 GCAC

选择完参数值后 GCAC 算法用来执行比较，以评估呼叫选择的链路是否有足够的带宽，如果对于给定的链路通告了 CRM 和 VF，则建议使用复杂的 GCAC 算法。

1) 算法。

注：附录 B 中给出了算法的来历。

步骤 1：如果 $AvCR(i) \geq PCR$ ，包括链路，停止。

步骤 2：如果 $AvCR(i) < SCR$ ，不包括链路，停止。

步骤 3：如果 $[AvCR(i) - SCR] \times [AvCR(i) - SCR + 2 \times CRM(i)] \geq VF(i) \times SCR(PCR - SCR)$

则：包括链路；

否则：不包括链路。

步骤 4：停止。

2) 特殊情况。

值得注意的是，如果 $PCR = SCR$ （如果在业务量描述语 IE 中未规定 SCR），只执行上述的步骤 1 和步骤 2。例如：步骤 1 和步骤 2 总能给出确定的结构，所以步骤 3 在这种情况下则不需要执行。

同时在一种极端的情况下，通告的 CRM 和 VF 均为 0，则步骤 3 的结果总是“包括”，在另一种极端的情况下，通告的 $VF(i) = \infty$ ，则步骤 3 的结果总是“不包括”。

3) 算法的改进选项。

当在两个 PTSE 之间有多个连接建立请求， $CRM(i)$ 和 $AvCR(i)$ 可以由本地选择修改，用来反映新建立连接，当这些连接在一个新的 PTSE 到达前释放，则 $CRM(i)$ 和 $AvCR(i)$ 同样需要修改。这个选项在使用时会有极端的告警，因为它会造成不同节点认为网络情况不一致。

5.13.4.3 简单的 GCAC

如果只通告了 $AvCR$ ，则建议使用简单的 GCAC，S-GCAC 算法如下。

步骤 1：如果 $AvCR > C$ ，则包括链路，停止。

否则不包括链路，停止。

此处 C 可以通过如下方法得到：

令 $X = PCR/SCR$

如果 $X > 39$ ，则 $C = SCR \times (0.0145 \times X + 4.22)$

否则 如果 $C > 5$ ，则 $C = SCR \times (0.042 \times X + 3.14)$

否则 $C = SCR \times (0.48 \times X + 0.52)$

5.13.5 Best-effort 业务的 GCAC 算法

对于 UBR 连接，当且仅当支持 UBR 业务类且最大信元率不为 0 的情况下，包括链路/节点。

对于 ABR 连接，当且仅当支持 ABR 业务类，最大信元率不为 0，且对 ABR 业务类通告的可用信元率大于或等于连接规定的最小信元率时，包括链路和节点。

如果呼叫中的最小可接受的 ATM 业务量描述字中包含一个峰值信元率，则峰值信元率同通告的最大信元率相比较，以确定呼叫是否能接受。

5.14 分组包格式

在本节中所规定的分组包格式可为 PNNI 路由分组包提供灵活且可扩充的编码。尤其是，它们依靠使用嵌入式长度值 (TLV) 编码。每一个 TLV 实体称为一个信息组。在所有的信息组中，类型字段和长度字段的长度每一个均为两个八位位组。类型字段的 4 个最高有效位用作编码信息组标签。在所有可能出现在同一位置的类型值中类型值必须只是唯一的。注意：对于第一阶段 PNNI，按照全局唯一的方式分配了类型值，以便使我们更易于确定每个 TLV 信息组的身份。长度字段的值包括类型字段和长度字段的长度（各为两个八位位组）以及值（信息组的剩余部分）的长度。

请注意每个 TLV 信息组对于 PNNI 选路的未来版本可任意扩充。尤其是在每个信息组数值部分开始的固定字段以后，可以出现几个子 TLV 信息组。只了解包括信息组文本中的类型值。准确地说，甚至在目前全局唯一分配类型值时，在包括 IG 中出现的未定义的某一类型应按照未知来处理，而不是采用全局定义。如果不了解这些子信息组中的任何一个，则它对父信息组译码的影响由按子信息组类型字段编码的信息组标签值来确定。

当需要时,可对所有的 LTV 进行填充,以致于它们的长度将总是 4 个八位位组的整倍数。如果需要填充,则所讨论的 TLV 的格式将说明填充这些八位位组的方案。所有预留的字段和任何填充的字段必须由始发者置成 0,且在接收时必须忽略。

除非另有说明,否则所有的字段都是无符号的整数。多个八位位组字段按“网络次序”发送,即先发送在八位位组流中第一个出现的最高的八位位组。

5.14.1 注释

在某个八位位组中的比特位置用号码 1~8 说明,其中 1 表示最低有效位,而 8 表示最高有效位。在几个八位位组字段中的比特位置用号码 1~ $8 \times n$ 来说明,其中 1 表示最后一个八位位组的最低有效位,而 $8 \times n$ 表示第一个八位位组的最高有效位。

5.14.2 信息组标签

如上所述,PNNI 分组包使用信息组构成。在本标准中规定了这些信息组中的许多信息组。但是,不可能规定所有可能的信息组。为了提供未来的扩充,重要的是要按这样一种方法来标识信息组,即不能标识这些信息组的系统应能正确地处理它们。这将有助于逐渐增加新的性能。

其他的选路协议发现涉及信息组未来发展的可行技术是要给它们作出标记。例如,BGP 和 IDRP 有一个必备对任选的标签,和一个转接对非转接的标签。当我们不能正确使用同样的标签时,我们可以利用它们的经验。为了提供协调处理,所有的信息组都有标签。这些标签只有在处理不能标识的信息组时才使用。

5.14.2.1 必备标签

必备标签可防止不了解该信息组的系统立即使用所包括的信息组。例如:如果未知的必备信息组在水平链路信息组中立即出现,则所述的链路在路由计算时一定不能使用。

如果系统收到的 PTSE 有一个高级必备标签的信息组,则它必须接受该 PTSE。换句话说该信息组是未知的。必须按正常方式检查校验以及序号,且必须确认存储和视需要按正常方式传送该 PTSE。然而,由该 PTSE 所描述的网络实体(节点或链路)必须不用于任何路由计算,在由路由计算所引起的 DTL 中不包括。

然而,收到包括所规定的 DTL 呼叫请求的节点必须根据规定接受该 DTL,与在该 DTL 中所包括的任何单元是否可使用未知的必备信息组进行描述无关。收到该呼叫请求的节点必须处理该呼叫请求,且假设:准备好相关 DTL 的系统知道它已作什么(即了解任何必备信息组),因此能为该呼叫选择一个正确的 DTL。同样的,如果是上行链路,则选择相应低级 DTL 的节点必须选择其 DTL 的相应上行链路,与该上行链路上的任何未知必备信息组无关。在第 6 章中有更详细的描述。

对于第一阶段,必须标识本技术规范中所规定的所有信息组。

5.14.2.2 归纳标签

PGL 准备归纳在下一级往上使用的对等组信息。如果该信息的某些部分包括未知的信息组,则 PGL 必须决定使用该信息做什么。请注意这个决定与必备标签无关。

未知的信息组可以作上标记,以指明 PGL 必须不把包括的实体归纳入对等组中。在基于对等组转接特性计算缺省节点表示时将不认为是包括未知非归纳的信息组的内部链路和节点。高级逻辑组节点将不通告包括未知可归纳的信息组的可达性信息。在构成高层上行链路和水平链路时不计包括未知非可归纳的信息组。这个规划的一个影响是:如果具有某给定混合标记的所有上行链路都有未知的非可归纳的信息组,则将不会建立受影响的上行链路或水平链路。

5.14.2.3 转接标签

假设可以归纳具有未知信息组的某一项,在各个信息组方面还有问题要去解决。如果信息组标签为非转接的,则必须在归纳之前去掉该信息组。如果标为转接标签,则必须保留该信息组。

在分别陈述所有的项时,如有一条外部链路,则保留一个转接信息组不是一个问题。在混合的实体中更有问题。

如果某一信息组标为转接,且在父对等组中汇集成一个通告,且对等组领导不认识该信息组,则导

致的通告必须包括该信息组（仍标为转接和可归纳）。

注：这意味着如果在某对等组中的一个单节点有可归纳、转接、未知的量度，则这个信息组将适用于整个对等组。

如果正被汇集的两个通告载送同一转接类信息组但有不同值，则 PGL 将保留两个信息组，这就意味着同一个信息组可以出现多次。

5.14.2.4 必备非转接信息组

信息组是必备，指有关的节点或链路不能在较高级通告，除非包括必备的信息组。信息组是非转接指如果相关的网络实体在较高级汇集，则在归纳实体通告中不能包括该信息组。

接到网络实体通告的必备非转接信息组指：在较高级不能为通告归纳该实体。因此，如果上行链路标为必备且非转接，则在相应的上行链路通告中不能将外部链路与其他链路进行汇集。如果某对等组中的节点或内部链路标为必备非转接信息组，则这个节点或内部链路不能用于计算相应逻辑组节点的特性。

5.14.2.5 举例

为了扩大这些信息组标签的情况，给出了几个可增加的信息组示例及其标签。

1) 建议的量度和属性

随着技术的发展，我们可以建立新的量度、属性或与业务有关的特性。当对某些呼叫可能需要这些时，即使这些节点不了解这些能力，不需要它们的透明呼叫仍可以使用这些链路。

因此，这些将是任选、可归纳的信息组。这种信息组是否是转接的可能取决于它们汇集行为的详细情况，因此从一个信息组到另一个信息组将有变化。

2) 本地信息组

某特定客户可相信一个或多个设备的提供者可以提供新的信息，这种信息可能被忽略，但由于它是试验性的，所以它一定不能归纳到该实验的域外。有标签的单元在外部仍可使用，这些信息组通常将是任选、可归纳、非转接的信息组。

另一方面，我们可有一条特殊的链路对某类用户限制其用法。为了保证一些对等组领导不发送通告到域外，这条链路有一个标识所允许用户类别的信息组。这个通告将是必备和非归纳的。

3) 用户列表

我们可以在将来规定载送哪些用户能使用资源（链路或节点）表的信息组，然后我们将按照必备和非转接来规定这些信息组。完成这个非转接信息组将保证在使用期间该资源从外部看是明显可见的，而不是通过没有限制的汇集方法。

4) 安全性标记

在没有深入研究安全性的全部情况时，我们可以预计在将来的某个时间建立一个安全性标记。这些可能是必备、可归纳的转接信息组。

5.14.2.6 信息组标签比特定义

所有的 PNNI 类型按 16bit 字段进行编码，这个字段最高的 4 个有效位预留给信息组标签。

这 4 个最高有效位如下：

Mandat	D-Sum	Trans	Reserv
--------	-------	-------	--------

Mandat: 必备比特（比特 16）。对信息组这个比特置成 1，如果不明白这个信息组，则在通道计算时一定不能使用包括的 IG 或 PTSE。

D-Sum: 非归纳比特（比特 15）。对信息组这个比特置成 1，如果不明白这个信息组，则一定不能包括的 IG 或 PTSE 归纳成包括的对等组。

Trans: 转接比特（比特 14）。这个比特置成 1，如果汇集包括的 IG 或 PTSE 且对等组领导不明白该信息组，则要保留这个信息组。

Reserv: 备用比特（比特 13）。当整个字段备用时，如果发送，则这个比特必须为 0 且在接收时忽略。

所有的 PNNI 1.0 信息组在始发时，应将它们的信息组标签置成任选的、可归纳、非转接。只有一个例外，即转接网 ID 信息组应有信息组标签值为任选的、可归纳的、转接。

5.14.3 信息组归纳

信息组归纳见表 22。

表 22 信息组归纳

类型	IG 名称	包括以下一级 IG	
32	汇集令牌		
33	节点层次列表		
34	上行链路信息属性	出局资源可用性 (128)	
35	LGN 水平链路扩展		
64	PTSE	节点状态参数 (96)，节点信息组 (97)，内部可达的 ATM 地址 (224)，外部可达的 ATM 地址 (256)，水平链路 (288)，上行链路 (289)，系统能力 (640)	
96	节点状态参数	出局资源可用性 (128)	
97	节点信息组	下一个较高级绑定信息 (192)	
128	出局资源可用性	任选的 GCAC 参数 (160)	
129	入局资源可用性	任选的 GCAC 参数 (160)	
160	任选的 GCAC 参数		
192	下一个较高级绑定信息		
224	内部可达的 ATM 地址	出局资源可用性 (128) 入局资源可用性 (129)	
256	外部可达的 ATM 地址	出局资源可用性 (128) 入局资源可用性 (129)，转接网 ID (304)	
288	水平链路	出局资源可用性 (128)	
289	上行链路	上行链路信息属性 (34)，出局资源可用性 (128)	
304	转接网 ID		
384	节点 PTSE 证实		
512	节点 PTSE 归纳		
513	所请求的 PTSE 头		
640	系统能力		
32	汇集令牌	Hello (1)	
33	节点层次列表	Hello (1)	
34	上行链路信息属性	上行链路 (289)	Hello (1)
35	LGN 水平链路扩展	Hello (1)，对于 LGN 水平 Hello	
64	PTSE	PTSP (2)	
96	节点状态参数	PTSE—受限的 IG	PTSP (2)
97	节点信息组	PTSE—受限的 IG	PTSP (2)

续表 22

类型	IG 名称	包含在以上一级 IG	包含在分组包中
128	出局资源可用性	上行链路信息属性 (34), 节点状态参数 (96), 内部可达的 ATM 地址 (224), 外部可达的 ATM 地址 (256), 水平链路 (288), 上行链路 (289)	Hello (1), PTSP (2)
129	入局资源可用性	内部可达的 ATM 地址 (224), 外部可达的 ATM 地址 (256)	PTSP (2)
160	任选的 GCAC 参数	出局资源可用性 (128), 入局资源可用性 (129)	Hello (1), PTSP (2)
192	下一个较高级绑定信息	节点信息组 (97)	PTSP (2)
224	内部可达的 ATM 地址	PTSE—受限的 IG	PTSP (2)
256	外部可达的 ATM 地址	PTSE—受限的 IG	PTSP (2)
288	水平链路	PTSE—受限的 IG	PTSP (2)
289	上行链路	PTSE—受限的 IG	PTSP (2)
304	转接网 ID	外部可达的 ATM 地址 (256)	PTSP (2)
384	节点 PTSE 证实		PTSE 证实 (3)
512	节点 PTSE 归纳		DB 归纳 (4)
513	所请求的 PTSE 头		PTSE 请求 (5)
640	系统能力	PTSE	所有的分组包

在 PNNI 分组包中的信息组如表 23 所示。

表 23 在 PNNI 分组包中的信息组

类型	分组包名称	包含 IG
1	Hello	汇集令牌 (32), 节点层次列表 (33), 上行链路信息属性 (34), LGN 水平链路扩展 (35), 出局资源可用性 (128), 任选的 GCAC 参数 (160), 系统能力 (640)
2	PTSP	PTSE (64), 节点状态参数 (96), 节点信息组 (97), 出局资源可用性 (128), 入局资源可用性 (129), 下一个较高级绑定 (192), 任选的 GCAC 参数 (160), 内部可达的 ATM 地址 (224), 外部可达的 ATM 地址 (256), 水平链路 (288), 上行链路 (289), 转接网 ID (304), 系统能力 (640)
3	PTSE ACK	节点 PTSE 证实 (384), 系统能力 (640)
4	DB 归纳	节点 PTSE 归纳 (512), 系统能力 (640)
5	PTSE 请求	所请求的 PTSE 头 (513), 系统能力 (640)

5.14.4 PNNI 分组包头

所有的 PNNI 选路分组包都是从公共的 PNNI 分组包头开始, 示于表 24 和表 25。

表 24 PNNI 分组包头

偏差	大小（八位位组）	名称	功能/描述
0	2	分组包类型	见表 25
2	2	分组包长度	
4	1	协议版本	根据构成这个分组包的情况来说明该版本。这个技术规范规定了 PNNI 选路协议分组包格式的版本。
5	1	所支持的最新版本	包括所支持的最新版本和所支持的最老版本字段，以便为交换特定类型分组包的两个节点就都能理解的最近协议版本进行协商。
6	1	所支持的最老版本	请见上述。
7	1	备用	

表 25 PNNI 分组包类型

分组包类型	分组包名称
1	Hello
2	PTSP
3	PTSE 证实
4	数据库归纳
5	PTSE 请求

5.14.5 资源可用性信息组

对于节点，只使用出局资源可用性信息组。对于链路和可达地址，出局资源可用性信息组（类型=128）用来规定从这个节点至相邻节点或可达地址的路由和属性。入局资源可用性信息组（类型=129）用来规定从相邻节点或可达地址至这个节点的路由和属性。除了始发类型编码以外，两个资源可用性信息组是相同的。

每一个资源可用性信息组适用于一组业务类别，它们使用比特掩码来描述，如表 26 和表 27 所示。当所有共享的路由和属性值对所有规定的业务类别都相同时，这个编码允许使用一个信息组来规定几个业务类别的路由和属性。不可用于某些业务类别的度量或属性，当认为那些业务类别可用时必须忽略。为了对不同组的业务类别通告不同的路由和属性，必须使用独立的资源可用性信息组。某一资源必须不能用来支持某一业务类别，除非该类别在为该资源所通告的资源可用性信息组的比特掩码中规定。

表 26 资源可用性信息组

偏差	大小（八位位组）	名称	功能/描述
0	2	类型	类型=128，对于出局资源可用性信息 类型=129，对于入局资源可用性信息
2	2	长度	
4	2	RAIG 标志	对于比特定义见表 27

续表 26

偏差	大小 (八位位组)	名称	功能/描述
6	2	备用	
8	4	管理加权	缺省值=缺省管理加权, 加法
12	4	最大的信元速率	单位: 信元/s
16	4	可用的信元速率	单位: 信元/s
20	4	信元转接时延	单位: ms
24	4	信元时延变化	单位: ms
28	2	信元丢失率 (CLP=0)	按照该值的负算法进行编码, 即在消息中的值 n 表明 CLR 为 10^{-n}
30	2	信元丢失率 (CLP=0+1)	按照该值的负算法进行编码, 即在消息中的值 n 表明 CLR 为 10^{-n}
任选的 GCAC 有关的信息:			
32	2	类型	类型=160 (任选的 GCAC 参数)
34	2	长度	
36	4	信元率边际	单位: 信元/s
40	4	变化系数	单位为 2^{-8} 。请注意: 变化系数的值 0xFFFFFFFF 用来表明无穷大

表 27 RAIG 标志

比特 ID:	比特 16 (MSB)	比特 15	比特 14	比特 13	比特 12	比特 11~2	比特 1 (LSB)
含义:	CBR	nt-VBR	nr1-VBR	ABR	UBR	备用	GCAC CLP 属性

5.14.6 上行链路信息属性

由边界节点和 PGL 所完成的上行链路通告以及在外链路上所发送的 Hello 必须包括上行链路信息属性 (ULIA), 如表 28 所示。ULIA 基本上是一个 TLV 编码的存储器或打包器, 用来包装和标识用于上行链路上相反方向的其他信息组。

对所支持的每个业务类别都必须包括资源可用性信息组。作为一种任选, 在这里将包括为描述上行链路上任何其他方面所需要的任何其他的信息组。

表 28 上行链路信息属性

偏差	大小 (八位位组)	名称	功能/描述
0	2	类型	类型=34 (上行链路信息属性)
2	2	长度	
4	4	序号	
<ul style="list-style-type: none"> 用来描述上行链路相反方向的所有的出局资源可用性信息组 (类型=128) 为了描述上行链路相反方向所需要的任何附加的任选 IG 			

5.14.7 转接网 ID

在 PNNI 中需要转接网标识信息，如表 29 所示。这个信息组用来载送这种信息。这个信息组可以出现在外部可达地址 IG 中。

表 29 转接网 ID

偏差	大小（八位位组）	名称	功能/描述
0	2	类型	类型=304（转接网 ID）
2	2	长度	
4	2	TNS 的长度	在网络标识串+网络标识数据中的八位位组数
6	1	网络标识数据	见表 30
7	N	网络标识	IRA 字符。每个字符 1 个八位位组。请见 ITU-T T. 50（1992）国际基准 Alphabet (IRA)，或以前的国际 Alphabet No.5 或 IA5
$7+N$	填充		0~3 个八位位组填充是使用公式： $(4 - [(3+N) \text{ 模 } 4]) \text{ 模 } 4$

表 30 网络标识数据

比特 ID:	比特 8 (MSB)	比特 7~5	比特 4~1
描述:	备用	网络标识的类型	网络标识规划
注：网络标识数据字段的值同 ITU 的规定。			

5.14.8 PNNI Hello

每个节点在所有的物理链路上，这个节点是一个端点的所有虚通道连接上和对于这个节点是一个端点为了交换 PNNI 路由信息所建立的所有 SVCC 上向最相邻的节点发送 PNNI Hello 如表 31 所示。在相邻节点之间交换 Hello 的目的是为了建立它们之间的链路状态。

表 31 PNNI Hello

偏差	大小（八位位组）	名称	功能/描述
0	8	PNNI 头	PNNI 分组包头结构中分组包类型=1（Hello）。见表 24
8	2	标志	备用
10	22	节点 ID	始发节点的节点 ID
32	20	ATM 端系统地址	实现 PNNI 选路的系统对它表示的每个节点必须有一个唯一的 ATM 端系统地址，且必须能接受到那个 ATM 端系统地址的呼叫（例如，这可用于网络管理）。对于分开的系统和对于对等组领导，通过使用区分相应于不同节点的 ATM 端系统地址的选择符八位位组最多可以实现 256 个唯一的 ATM 端系统地址
52	14	对等组 ID	始发节点的对等组 ID
66	22	远端节点 ID	根据在同一条链路上所收到的 Hello 中节点 ID 的值来设置，如果还不知道则置成全零
88	4	端口 ID	<ul style="list-style-type: none"> 有效的端口 ID 不允许有零值。 对于任一端或两端作为对等组领导的情况（o 是在 SVCC 上而不是在 VCI 18 上进行交换），端口 ID 取值为 0xFFFFFFFF

续表 31

偏差	大小(八位位组)	名称	功能/描述
92	4	远端端口 ID	根据在同一条链路上所收到的 Hello 中端口 ID 的值来设置。如果还不知道则置成零
96	2	HelloInterval	HelloInterval 表明 Hello 分组包是如何频繁的发送。如果在某一时间内未收到 Hello (不活动系数乘 HelloInterval), 则认为该链路有故障
98	2	备用	

注: 在这个节点的对等组以外指定的链路上发送的 Hello 包括汇集令牌、节点层次列表和上行链路信息属性信息组 (请见 5.6.2.2 节), 上行链路信息属性在表 28 中描述。

在 LGN 之间作为 LGN 水平链路 Hello 协议的一部分所发送的 Hello (请见 5.6.3.2 节) 包括 LGN 水平链路扩展信息组。

5.14.8.1 汇集令牌 IG

汇集令牌用来控制外部链路如何在该分级结构中的高级进行汇集, 如表 32 所示。链路汇集的完整描述请见 5.10.3.1 节。

表 32 汇集令牌 IG

偏差	大小(八位位组)	名称	功能/描述
0	2	类型	类型=32 (汇集令牌)
2	2	长度	
4	4	汇集令牌	

5.14.8.2 节点层次列表 IG

为了允许低级相邻节点确定它们最低级公共的对等组, 且还要交换节点 ID 和该公共对等组中相应的高级相邻节点的 ATM 端系统地址包括了节点层次列表, 如表 33 所示。相邻对等组之间同步动作的工作详情和实现高级选路的正确操作在 5.6 节中描述。

表 33 节点层次列表

偏差	大小(八位位组)	名称	功能/描述
0	2	类型	类型=33 (节点层次列表)
2	2	长度	
4	4	序号	
8	2	备用	
10	2	级数	已知道的较高级对等组数
重复下面的结构级别计数次数, 从父级和前一级开始, 直到在最高已知级列出了该节点的身份:			
	22	下一个较高级逻辑节点 ID	
	20	下一个较高级 ATM 端系统地址	下一个较高级逻辑节点的 ATM 端系统地址
	14	下一个较高级对等组 ID	下一个较高级逻辑节点的对等组 ID

5.14.8.3 LGN 水平链路扩展 IG

下面的 TLV 称为 LGN 水平链路扩展，它载送两个逻辑节点之间有关所有水平链路的状态信息，如表 34 所示。它作为附加的 TLV 在用来保持 LGN 之间 SVCC 状态的同一消息中载送。

表 34 LGN 水平链路扩展 IG

偏差	大小 (八位位组)	名称	功能/描述
0	2	类型	类型=35 (LGN 水平链路扩展)
2	2	长度	
4	2	备用	
6	2	H 链路数	所描述的水平链路数
重复下面的结构 H 链路计数次数			
	4	汇集令牌	
	4	本地 LGN 端口	
	4	远端 LGN 端口	

5.14.9 PNNI 拓扑状态分组包 (PTSP)

PNNI 拓扑状态分组包 (PTSP) 用来将信息分散到整个对等组。

所有的 PTSP 包括在表 35 中所描述的 PTSP 头中。

表 35 PTSP 头

偏差	大小 (八位位组)	名称	功能/描述
0	8	PNNI 头	PNNI 分组包头结构中分组包类型=2 (PNNI 拓扑状态分组包)，见表 24
8	22	始发节点 ID	确定该节点是否在这个 PTSP 中始发 PTSE
30	14	始发节点的对等组 ID	

每一个 PTSP 都由多个 PNNI 拓扑状态单元 (PTSE) 组成，所有的单元都来自同一个始发节点。每个 PTSE 包括其自己的校验和与鉴权字段 (在 PNNI 第一阶段不存在)，于是允许来自同一个始发节点的 PTSE 按始发时的不同组合以及在它们穿过某一对等组进行传播时要进行分组包。PTSE 提供用于重发目的、应用序号、校验和、鉴权的信息单元。

每一个 PTSE 都包括在表 36 所描述的 PTSE 头中。

表 36 PTSE 头

偏差	大小 (八位位组)	名称	功能/描述
0	2	类型	类型=64 (PTSE)
2	2	长度	
4	2	PTSE 类型	表明允许哪一个受限的信息组出现在 PTSE 的内，只可以包括那些带有匹配 TLV 类型的受限信息组。 注：这不是为了影响嵌入在受限信息组内部的 TLV 类型，只有 PTSE 中‘最高级’的受限信息组才必须符合这个原则
6	2	备用	

续表 36

偏差	大小 (八位位组)	名称	功能/描述
8	4	PTSE 标识符	标识来自某个节点的多个不同的 PTSE 中的一个。带有 PTSE 标识符值为 '1' 的 PTSE 打算用于分配节点信息组。它的 PTSE 类型字段的值与节点信息组类型值一致
12	4	PTSE 序号	
16	2	PTSE 校验和	注: PTSE 校验和包括逻辑节点 ID、来自 PTSP 头的始发节点的对等组 ID 以及 PTSE 的全部内容 (包括 PTSE 头), 不包括 PTSE 剩余生存时间。用于计算 PTSE 校验和的校验和算法在 5.8.2.2 节的 2) 条中描述
18	2	PTSE 剩余生存时间	

除了受限的信息组以外, 下述任何数目的不受限的信息组可出现在所有的 PTSE 类型中。

5.14.9.1 受限的信息组

1) 节点状态参数 IG

节点状态参数信息组用来通告这个复杂节点表示的组成部分如表 37 和表 38 所示。在同一个节点状态参数信息组中必须包括每对输入—输出端口的所有量度和属性。这个信息组可以出现多次且每次都出现在多个不同的 PTSE 中, 使用不同的输入和输出端口。

表 37 节点状态参数 IG

偏差	大小 (八位位组)	名称	功能/描述
0	2	类型	类型=96 (节点状态参数)
2	2	长度	
4	2	标志	见表 38
6	2	备用	
8	4	输入端口 ID	
12	4	输出端口 ID	
对每种 (组) 业务类别重复: 出局资源可用性信息组 (类型=128)			

表 38 标志

比特 ID:	比特 16 (MSB)	比特 15~1
比特名称:	VP 能力标志	备用

如果相应的实体能载送 VPC, 则 VP 能力标志码置成 1。如果不支持 VPC, 则 VP 能力标志码置成 0。例如: 如果水平链路或上行链路全部由 VPC 或汇集 VPC 组成, 其中没有真正物理链路, 则它将在其中载送 VPC 且 VP 能力标志码必须置成 0。

2) 节点信息组

节点信息组用来传送有关节点的一般信息, 如表 39 和 40 所示。例如: 其对等组 ID 和其 ATM 端系统地址, 还传送所有 PGL 选择信息。这个信息组在由这个节点通告的所有 PTSE 中只出现一次。

表 39 节点 IG

偏差	大小 (八位位组)	名称	功能/描述
0	2	类型	类型=97 (节点信息组)
2	2	长度	
4	20	ATM 地址	始发节点的 ATM 端系统地址。
24	1	领导优先权	0 值预留给不愿或不能成为对等组领导的节点
25	1	节点标志	见表 40
26	22	优选的对等组领导节点 ID	
下一个较高级绑定信息 (如果有较高级, 则由对等组领导发送):			
48	2	类型	类型=192 (下一个较高级绑定信息)
50	2	长度	
52	22	父逻辑组节点 ID	父逻辑组节点 ID (在父对等组中)
74	20	父 LGN 的 ATM 端系统地址	
94	14	父对等组 ID	
108	22	父对等组 PGL 的节点 ID	如果不知道就无
130	2	备用	

表 40 节点标志

比特 ID:	比特 8 (MSB)	比特 7	比特 6	比特 5	比特 4	比特 3~1
比特名称:	“我是领导” 比特	“受限转接” “比特”	“节点表示比 特”	“受限分支” “比特”	为 PGL 选择 的非转接	备用
描述:	值 0: “我不是 PGL”; 值 1: “我是 PGL”	值 0: “我是转接节点”; 值 1: “我是受限转接节点”	值 0: “简单节点表示”; 值 1: “复杂节点表示”	值 0: “可以支持附加的分支点”; 值 1: “不能支持附加的分支点”	值 0: “正常操作”; 值 1: “为了 PGL 选择不能通过这个节点连接”	

3) 内部可达的 ATM 地址信息组

内部可达的 ATM 地址信息组用于通告直接相连的 ATM 端点的地址, 如表 41 所示。这个信息组可以出现多次且是在多个不同的 PTSE 中, 每次都列出了内部可达的 ATM 地址的不同设置。

表 41 内部可达的 ATM 地址 IG

偏差	大小 (八位位组)	名称	功能/描述
0	2	类型	类型=224 (内部可达的 ATM 地址)
2	2	长度	
4	2	标志	见表 38

续表 41

偏差	大小 (八位位组)	名称	功能/描述
6	2	备用	
8	4	端口 ID	
12	1	通告的范围	PNNI 选路级别。通告的范围适用于在信息组中所包括的所有地址前缀, 不管它们是单个的地址还是组地址或两者的混合。
13	1	地址信息长度	ail, 以八位位组为单位
14	2	地址信息计数	aic
重复 (aic) 次数:			
	1	前缀长度	单位: bit
	ail-1	可达的地址前缀	在可达地址前缀最后八位位组的右边最末尾 (最低的比特) 用几个 0 的 $[8 \times (\text{ail}-1) - \text{前缀长度}]$ 比特进行填充
		填充	0~3 八位位组 请注意: 填充字段的大小使用下式进行计算: $(4 - ((\text{aic} \times \text{ail}) \bmod 4)) \bmod 4$
资源可用性信息的任选 TLV 组, 对每种 (组) 业务重复:			
<ul style="list-style-type: none"> • 出局资源可用性信息组 (类型=128); • 入局资源可用性信息组 (类型=129); 如果存在, 则资源可用性信息组适用于所有存在的可达地址前缀, 且直接与 PNNI 内部业务类别参数进行组合。			

4) 外部可达的 ATM 地址信息组

外部可达的 ATM 地址信息组用来通告可通过外部网到达不参加 PNNI 选路的 ATM 端点的地址, 如表 42 所示。这个信息组可以出现多次且在多个不同的 PTSE 中, 每次可列出外部可达 ATM 地址的不同组。与每个外部可达 ATM 地址有关的所有量度和属性必须包括在出现可达地址的同一个外部可达 ATM 地址信息组中。

表 42 外部可达的 ATM 地址 IG

偏差	大小 (八位位组)	名称	功能/描述
0	2	类型	类型=256 (外部可达的 ATM 地址)
2	2	长度	
4	2	标志	见表 38
6	2	备用	
8	4	端口 ID	
12	1	通告的范围	PNNI 选路级别。通告的范围适用于在信息组中所包括的所有地址前缀, 不管它们是单个的地址还是组地址或两者的混合
13	1	地址信息长度	ail, 以八位位组为单位

续表 42

偏差	大小 (八位位组)	名称	功能/描述
14	2	地址信息计数	aic
重复 (aic) 次数:			
	1	前缀长度	单位: bit
	ail-1	可达的地址前缀	在可达地址前缀最后八位位组的右边最末尾 (最低的比特) 用几个 0 的 $[8 \times (\text{ail}-1) - \text{前缀长度}]$ 比特进行填充
	0~3	填充	0~3 八位位组 请注意: 填充字段的大小使用下式进行计算: $(4 - ((\text{aic} \times \text{ail}) \bmod 4)) \bmod 4$
<ul style="list-style-type: none"> 资源可用性信息的任选 TLV 组, 对每种 (组) 业务重复。如果存在, 则资源可用性信息组适用于所有存在的可达地址前缀, 且直接与 PNNI 内部业务类别参数进行组合。 出局资源可用性信息组 (类型=128); 入局资源可用性信息组 (类型=129); 附加的任选 TLV 组; 转接网 ID (类型=304)。 			

5) 水平链路信息组

水平链路信息组用来通告至同一个对等组中其他节点的链路, 如表 43 所示。这个信息组可以出现多次且在多个不同的 PTSE 中, 每次都列出不同的水平链路。与每条水平链路有关的所有量度和属性都包括在同一个水平链路信息组中。

表 43 水平链路 IG

偏差	大小 (八位位组)	名称	功能/描述
0	2	类型	类型=288 (水平链路)
2	2	长度	
4	2	标志	见表 38
6	22	远端节点 ID	本地节点 ID 和端口 ID 的组合明确地确定该链路。远端节点 ID 和远端口 ID 的组合明确地确定相反的链路—为了在有关的两个方向进行量度需要这样做
28	4	远端口 ID	同上
32	4	本地端口 ID	
36	4	汇集令牌	
<ul style="list-style-type: none"> 对每种 (组) 业务类别重复; 出局资源可用性信息组 (类型=128)。 			

6) 上行链路信息组

上行链路信息组用来通告至节点的链路且位于这个对等组的外部。这个信息组可以出现多次且在多个不同的 PTSE 中, 每次列出不同的上行链路。与每条上行链路有关的所有量度和属性必须包括在同一个上行链路信息组中。

表 44 上行链路 IG

偏差	大小 (八位位组)	名称	功能/描述
0	2	类型	类型=289 (上行链路)
2	2	长度	
4	2	标志	见表 38。
6	2	备用	
8	22	远端高级节点 ID	
30	14	公共对等组 ID	
44	4	本地端口 ID	
48	4	汇集令牌	
52	20	上行节点的 ATM 端系统地址	
<ul style="list-style-type: none"> • 对每种 (组) 业务类别重复; • 出局资源可用性信息组 (类别=128); • 上行链路信息属性 (类型=34)。 			

5.14.9.2 不受限的信息组

系统能力信息组只是用于 PNNI 第一阶段不受限的信息组。其他不受限的信息组可能在 PNNI 的以后版本中规定。哪些预见有待证实和接入控制。

5.14.9.3 未知的信息组

未知的信息组是出现在某类 PTSE 中的那些信息组, 而且与所指出的类型不一致和不知道是否是受限的 TLV。

当由某节点发送的 PTSE 运行 PNNI 的较新版本且由运行旧版本的节点接收时, 可以遇到这种信息组。

5.14.9.4 处理违反限制的 PTSE

由于 PTSE 文本中所述的限制, 所以不能排除某些 PTSE 将溢出, 它们将违反这些规定。可能的差错情况和采取的动作如下:

- 1) 如果 PTSE 包括一个受限的信息组, 其类型与 PTSE 类型不一致, 则在这种组中所包括的信息对于路由计算可以忽略。
- 2) 未知的信息组和受限的信息组一样处理, 根据所规定的信息组标签处理它们。
- 3) 如果在某 PTSE 中丢失了一个必备的受限信息组 (实例将是一个空的 PTSE, 其类型等于“节点信息”和 PTSE 标识符=1), 在路由计算时必须忽略该 PTSE。
- 4) 如果收到了一个带有未知 PTSE 类型的 PTSE, 则所有受限的信息组内容根据规则 1) 和 2) 进行处理。

TLV 或 IG 不仅对于通道计算应忽略。这些计算类别因为它们影响协议的内部状态故称为状态有效计算且现规定为:

- i) 链路组合;
- ii) 上行链路产生;
- iii) 对等组复杂节点表示的计算;
- iv) 对等组领导选出, 除了计数选举的投票赞成数 (如果存在);
- v) 计算外部呼叫时要通告的级数;
- vi) 可达性计算。

通道计算不认为是一个状态有效计算，这是由于以下事实。即如果两个节点由同组信息计算不同的路由，则它可能对整个网和呼叫拒绝率产生有害的影响且不保证协议状态机自身正常工作。因此，当涉及下述情况随实现不同变化时，虽然建议使用所规定的规则，但通道计算是任意的以便选择不同的策略。当引起忽略的条件改变时，根据以下几节忽略的信息必须重新估算。

5.14.9.5 PTSE 差错情况

在 PTSE 中所支持的 TLV 构成句法上不防止分组包语义中的多义性，例如以下情况：

- a) 不支持某 TLV 出现，至少出现一次。
- b) 支持某 TLV 在嵌入的 TLV 内正好出现一次，出现多次。
- c) 支持某 TLV 至少出现一次，完全不出现。
- d) 不支持这种 TLV 按嵌入类型出现嵌入。
- e) 同一个 TLV 按不同的 PTSE（带有不同 ID 的 PTSE）中出现多次的情况 d) 统一处理。在嵌入 TLV 内出现的不希望的 TLV（例如出现在水平链路 TLV 内的节点信息 TLV）基于 5.14.9.4 节进行处理（RIG 出现在 PTSE 内，表明不同的类型）。

注：a) 和 d) 之间以及 a) 和未知的 TLV 情况（后者由属性标签处理）之间稍有差别。

1) 在节点信息 TLV 中的节点信息组。

- a) 不应用。
- b) 如果节点信息 TLV 在指定的 PTSE 中出现多次，则说明这些 TLV 的不同是有效的两种实现方法可能导致 PGL 选择不会聚。所提出的解决办法是只考虑第一个出现当对状态有效的计算是有效的和有意义的节点信息 TLV 并忽略后面所有的 TLV。
- c) 如果节点信息 TLV 完全不出现在指定 PTSE 中，则建议由这个节点发送的 PTSE 和所有另外的 PTSE 对于状态有效计算除 v_i 以外都应忽略。于是，带有无效节点信息的节点可用用来计算主网中其他节点的可达性，以便开始对等组领导选择，但其 PTSE（例如水平链路）不能用于通道选择。
- d) n/a。
- e) 由于这个 IG 只能出现在具有 ID1 的 PTSE 中，所以不可能，否则就不理睬它。

2) 在节点信息 TLV 中的下一个高级绑定 TLV。

- a) 如果这种 TLV 在带有透明的“我是领导”-比特的节点信息 TLV 内出现一次或多次，则这种 TLV 必须忽略。以下情况只适用于设置“我是领导”-比特的 TLV 且另外该节点确定为对等组领导：
- b) 如果绑定出现多次，则必须把第一个用于状态有效计算且忽略所有后续的。
- c) 正常情况。
- d) n/a。
- e) 等同于 5.14.8.5 节的 1) 条的 e)。

3) 节点状态参数 TLV。

以下情况只适用于设置“复杂表示”比特为 1 的 TLV。

- a) 不适用。
- b) 如果节点状态 TLV 只在一个 PTSE 内出现多次，则第一个可用于任何状态有效的计算（即使所规定的业务类别组不交叉）。这包括缺省的级别。
- c) 缺省级不出现。为了状态有效计算起见，除了 v_i 以外忽略节点的任何 PTSE。这种节点可以用来计算至网中其他节点的可达性，以便像对等组领导选择之类的情况，但其 PTSE 像水平链路不能用于通道计算一样。
- d) n/a。
- e) 如果节点状态在不同的 PTSE 中出现多次（即使所规定的业务类别组不交叉）则只有具有最低 ID 的 PTSE 中出现的一个才能用于状态有效计算。

如果节点未设置“复杂表示”比特，则

- a) 如果节点状态 TLV 出现，则对于任何状态有效计算可忽略它们。
- b) 不可用。
- c) 不可用。
- d) n/a。
- e) 类似 a)。

4) 水平链路 TLV

- a) 不可用。
- b) 如果这种 TLV 在一个 PTSE 内出现多次，则只有第一个可用于任何状态有效计算，即使所规定的业务类别组未交叉。
- c) 不可用。
- d) n/a。
- e) 不可用。

5) 上行链路 TLV

等同于 5.14.8.5 节的 4) 条。

6) ULIA

- a) 不可用。
- b) 如果多个 ULIA 出现在上行链路 TLV 中，则只有第一个可用于状态有效计算。
- c) 不可用。
- d) n/a。
- e) 不可用。

7) RAIG

如果业务类别出现在水平链路、上行链路、节点状态参数或 ULIA IG 内的多个 RAIG 中，则只有在第一个 RAIG 中出现的一个才可用于状态有效计算。

5.14.9.6 节点信息 PTSE 及其与由该节点所发送的其他 PTSE 的连接

由于节点信息载送几个重要的标志码和节点发布的要求，所以重要的是要规定在没有合适的节点信息 PTSE 存在时收到的一组 PTSE 的语义。为了改善协议的稳定性，所有的状态有效计算除了 vi) 以外都应忽略对其不存在有效节点信息 PTSE 的 PTSE。通道计算可以忘记这个限制，但是不应忘记：例如通过节点的复杂表示所计算的通道在节点信息 PTSE 收到该节点没有复杂表示的说明以后是无效的。

5.14.10 PTSE 证实分组包

PTSE 证实分组包用来证实从相邻节点收到了 PTSE，如表 45 所示。每个 PTSE 证实分组包由多个节点 PTSE 证实信息组组成。每个信息组通过对每个被证实的 PTSE 包括 PTSE 标识信息用来证实收到了由同一始发节点且由同一相邻节点发送的多个 PTSE。

表 45 PTSE 证实分组包

偏差	大小 (八位位组)	名称	功能/描述
0	8	PNNI 头	PNNI 分组包头结构中的分组包类型=3 (PTSE 证实分组包)，见表 24
对有关一个节点的每个 PTSE 证实组重复：			
	2	类型	类型=384 (节点的 PTSE 证实)
	2	长度	
	22	节点 ID	
	2	证实计数	这个节点证实 的数

续表 45

偏差	大小 (八位位组)	名称	功能/描述
重复以下结构证实计数的次数:			
	4	PTSE 标识符	
	4	PTSE 序号	
	2	PTSE 校验和	
	2	PTSE 剩余生存时间	

5.14.11 数据库归纳分组包

数据库归纳分组包在两个相邻的对等组之间开始数据库交换过程时使用, 如表 46 和表 47 所示。在数据库交换过程中, 发送数据库归纳分组包, 同时包括在节点拓扑数据库中所有 PTSE 的 PTSP 和 PTSE 头信息。数据库归纳分组包包括一个序号和用于协商所用的主从关系, 以便保证锁定步骤协议正常工作。

表 46 数据库归纳分组包

偏差	大小 (八位位组)	名称	功能/描述
0	8	PNNI 头	PNNI 分组包头结构中的分组包类型=4 (据库归纳分组包), 见表 24
8	2	标志	见表 47
10	2	备用	
12	4	DS 序号	
对拓扑数据库中的每组 PTSE 重复:			
	2	类型	类型=512 (节点的 PTSE 归纳)
	2	长度	
	22	始发节点 ID	
	14	始发节点的对等组 ID	
	2	备用	
	2	PTSE 归纳计数	对这个始发节点 ID 的 PTSE 归纳数
重复以下结构 PTSE 归纳计数次数:			
	2	PTSE 类型	
	2	备用	
	4	PTSE 标识符	
	4	PTSE 序号	
	2	PTSE 校验和	
	2	PTSE 剩余生存时间	

表 47 数据库归纳分组包标志

比特 ID:	比特 16 (MSB)	比特 15	比特 14	比特 13~1 (LSB)
比特名称:	‘启动’ (I) 比特	‘多’ (M) 比特	‘主’ (MS) 比特	备用
描述:	在数据库同步过程启动期间置成 1, 否则就置成 0	如果发送的节点有附加的 PTSE 要归纳, 则置成 1, 或如果所有的 PTSE 已被归纳则置成 0 (请见 5.7.5 节)	开始由两个节点置成 1。然后由在数据库同步过程中起从属作用的节点置成 0	

5.14.12 PTSE 请求分组包

PTSE 请求分组包在数据库同步时使用, 以便由相邻同级的那些 PTSE 请求, 这些 PTSE 是新发现或已发现是过时的, 如表 48 所示。

表 48 PTSE 请求分组包

偏差	大小 (八位位组)	名称	功能/描述
0	8	PNNI 头	PNNI 分组包头构中的类型=5 (PTSE 请求分组包), 见表 24
对每组所请求的 PTSE 重复:			
	2	类型	类型=513 (节点 PTSE 请求列表)
	2	长度	
	22	始发节点 ID	
	2	PTSE 请求计数	对这个始发节点 ID 所请求的 PTSE 数
重复 PTSE 请求计数次数:			
	4	PTSE 标识符	

5.14.13 系统能力

系统通过在 PNNI 分组包中包括“系统能力”IG 可表明支持任选的 PNNI 能力, 如表 49 所示。这可以用来表明支持标准能力 (由 ATM 论坛规定) 和专有能 (它可由各个设备生产厂家规定)。

系统能力 IG 的说明取决于它出现在何处。如果它出现在某分组包的高级, 则它描述那个分组包发送者, 即相邻节点的系统能力。如果它出现在某个 PTSE 内, 则它描述 PTSE 始发者的系统能力。

作为任选, 系统能力 IG 可包括在任何 PNNI 分组包 (特别包括呼叫、数据库归纳分组包和 PTSP) 中, 还包括在 PTSE 的高级, 但不在 PTSE 内的某信息组中。同样的, 这个字段可以出现多次 (例如: 可能希望区分扩充的标准能力以及专有能)。

表 49 系统能力 IG

偏差	大小 (八位位组)	名称	功能/描述
0	2	类型	类型=640 (系统能力)
2	2	长度	
4	2	系统能力内容的长度	IEEE OUI+系统能力信息的长度
6	3	IEEE OUI	IEEE 组织唯一的标识符, 参考 IEEE 标准 802-1990。

续表 49

偏差	大小 (八位位组)	名称	功能/描述
9	n	系统能力信息	这个字段的语义由 OUI 确定的组织进行控制。
9+n	0~3	填充	填充字段的大小使用下式进行计算： $(4 - [(5+n) \text{ 模 } 4]) \text{ 模 } 4$

标准能力可由 ATM 论坛使用分配给论坛的 OUI 规定。目前，没有规定的系统能力，它可使用 ATM 论坛的 OUI。

由 LGN 通告的系统能力表示其子对等组和/或 LGN 实现的能力。

6 PNNI 的信令规范

本节规定了在网络节点接口上,动态建立、维护和清除 ATM 连接的程序。

6.1 协议模型

图 23 表示在控制平面内，信令程序与业务之间的关系。

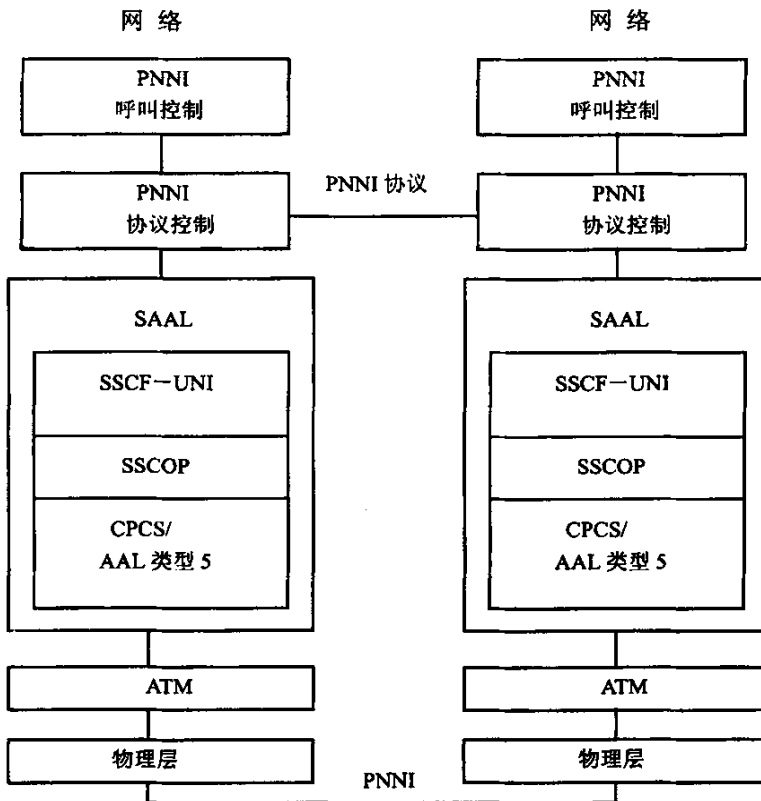


图 23 PNNI控制平面

6.1.1 信令层

信令层包含两个不同的实体：PNNI 呼叫控制和 PNNI 协议控制。PNNI 呼叫控制为上层功能服务，如资源分配和选路信息。PNNI 协议控制实体提供至 PNNI 呼叫控制的服务。

PNNI 协议控制采用对称程序处理实际的信令有限状态机。

6.1.2 信令适配层

PNNI 信令对称程序使用 UNI 信令 ATM 适配层 (SAAL) 的业务。SAAL 层包含以下内容:

1) 业务特定协调功能 (SSCF), 见邮电部技术规范 YDN 083—1998 《B-ISDN 的 DSS2 信令方式技术规范—适配层》。该功能是将业务特定面向连接协议 (SSCOP) 映射为信令程序所需的要求。

2) 业务特定面向连接协议 (SSCOP), 见邮电部技术规范 YDN 083—1998 《B-ISDN 的 DSS2 信令方式技术规范—适配层》。该业务为传送信令信息, 在一对 SSCOP 实体间提供对等的协议。

3) ATM 适配层类型 5 (AAL) CPCS 和 SAR 子层, 见邮电部技术规范 YDN 053. 4—1997 《ATM 适配层 (AAL) 类型 5 技术规范》。该业务提供了每个 ATM 层信元结构要求的信令数据单元的分段与重装。

6.1.3 ATM 层

ATM 层的要求参见 ITU-T 建议 I.150。ATM 层是在通信的 AAL 实体间透明传送固定尺寸的 ATM 层业务数据单元 (ATM-SDU)。PNNI 中使用的信元标记结构与 NNI 侧信元标记的格式和编码相同(见 I.361 标准的图 3), VPI 或 VCI 子字段中没有设置的比特应置为“0”。

6.2 呼叫/连接控制概述

本节定义了一个呼叫可能存在呼叫控制状态。这些定义并不用于接口本身或所连接的设备或信令虚通路的状态。因为在 PNNI 处可同时存在几个呼叫, 每个呼叫可能处于不同状态, 所以, 接口的状态是不能明确定义的。

6.2.1 ATM 点到点呼叫状态

本节定义了前方侧和后续侧之间的 PNNI 两端, ATM 点到点呼叫的控制状态。

6.2.1.1 零状态 (NN0)

无呼叫存在。

6.2.1.2 呼叫起始 (NN1)

这个状态在后续侧存在, 这时从前方网络节收到了呼叫建立请求, 但还没有收到响应。

6.2.1.3 呼叫进展发送 (NN3)

当后续侧已经知道收到建立呼叫所需的信息时, 这个状态存在。

6.2.1.4 提醒递交 (NN4)

当后续侧已经向前方侧发送了 ALERTING 消息时, 这个状态存在。

6.2.1.5 呼叫呈现 (NN6)

这个状态在前方侧存在, 这时已经向后续侧发送呼叫建立请求但尚未收到响应。

6.2.1.6 提醒接收 (NN7)

当前方侧已经从 PNNI 接口的后续侧收到了 ALERTING 消息后, 这个状态存在。

6.2.1.7 呼叫进展接收 (NN9)

当前方侧收到了后续侧对已收到呼叫建立请求的证实后, 这个状态存在。

6.2.1.8 运行 (NN10)

当 ATM 连接已经建立时, 这个状态存在。

6.2.1.9 释放请求 (NN11)

当网络节点请求 PNNI 接口另一侧的网络节点释放 ATM 连接请求并等待响应时, 这个状态存在。

6.2.1.10 释放指示 (NN12)

当网络节点已收到 PNNI 接口另一侧的网络节点对 ATM 连接的释放请求但尚未响应时, 这个状态存在。

6.2.2 全局呼叫参考的状态

同 Q.2931 标准的 2.3 节。

6.3 消息功能定义和内容

本节提供了消息结构的概述, 着重说明了每个消息的功能定义和信息内容 (即: 语义)。每个定义

包括：

1) 消息方向和使用的简述，包括消息的有效范围：

——局部有效，即仅与本地 PNNI 有关；

——接入有效，即与发端和终端接入有关，但与网络无关；

——全网有效，即与相关呼叫的本地 PNNI、其它 PNNI 和 UNI 有关。

2) 码组 0 信息单元表。对每个信息单元，该表表明：

——描述该信息单元实现协议的章节；

——该信息单元是必选（“M”）还是任选（“O”）的，若是任选，给出参考注释，说明包含该信息单元的情况；

——用八比特组表示的信息单元长度（或允许的长度范围）。

3) 必要时，作出进一步解释。

6.3.1 ATM 点到点呼叫和连接控制消息

表 50 概括了 ATM 点到点呼叫和连接控制消息。

表 50 ATM 呼叫和连接控制消息

消息	参考（节号）
呼叫建立消息：	
ALERTING	6.3.1.1
CALL PROCEEDING	6.3.1.2
CONNECT	6.3.1.3
SETUP	6.3.1.6
呼叫清除消息：	
RELEASE	6.3.1.4
RELEASE COMPLETE	6.3.1.5
其它消息：	
NOTIFY	6.3.1.9
STATUS	6.3.1.7
STATUS ENQUIRY	6.3.1.8

6.3.1.1 ALERTING

这个消息由后续侧发送，表示被叫用户已开始提醒，见表 51。

表 51 ALERTING 消息内容

消息类型：ALERTING			
方向：后续侧至前方侧			
有效范围：全网			
信息单元	参考（节号）	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4
消息类型	6.4.4.1	M	2

续表 51

消息类型: ALERTING			
方向: 后续侧至前方侧			
有效范围: 全网			
信息单元	参考 (节号)	类型	长度
消息长度	6.4.4.2	M	2
端点参考	6.4.8.1	O ⁽¹⁾	4~7
通知表示语	6.4.5.27	O ⁽²⁾	4~*
通用标识传送	6.4.5.31	O ^(2,3)	4~33
注:			
1 如果 SETUP 消息中包含端点参考, 则是必选信息单元。			
2 当收到的提醒指示中含有此信息时, 包含这个信息单元。			
3 这个信息单元可以在该消息中出现 3 次。			

6.3.1.2 CALL PROCEEDING

这个消息由后续侧发送, 表示所请求的呼叫/连接已开始建立, 并且不再接受任何呼叫建立信息, 见表 52。

表 52 CALL PROCEEDING 消息内容

消息类型: CALL PROCEEDING			
方向: 后续侧至前方侧			
有效范围: 本地			
信息单元	参考 (节号)	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
连接标识	6.4.5.22	M	9
端点参考	6.4.8.1	O ⁽¹⁾	4~7
注: 1 如果 SETUP 消息中包含端点参考, 则是必选信息单元。			

6.3.1.3 CONNECT

这个消息由后续侧发送并传递到前方侧, 表示被叫用户接受了呼叫/连接, 见表 53。

表 53 CONNECT 消息内容

消息类型: CONNECT			
方向: 后续侧至前方侧			
有效范围: 全网			
信息单元	参考 (节号)	类型	长度
协议鉴别语	6.4.2	M	1

续表 53

消息类型: CONNECT			
方向: 后续侧至前方侧			
有效范围: 全网			
信息单元	参考(节号)	类型	长度
呼叫参考	6.4.3	M	4
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
ABR 附加参数	6.4.5.5	O ⁽¹⁾	4~14
ABR 建立参数	6.4.5.6	O ⁽¹⁾	4~36
AAL 参数	6.4.5.8	O ⁽¹⁾	4~11
ATM 业务量描述语	6.4.5.9	O ⁽⁴⁾	4~30
宽带低层信息	6.4.5.12	O ⁽¹⁾	4~17
被叫用户软 PVPC 或 PVCC	6.4.6.2	O ⁽²⁾	4~11
被连接号码	6.4.5.20	O ⁽¹⁾	4~26
被连接子地址	6.4.5.21	O ⁽¹⁾	4~25
端点参考	6.4.8.1	O ⁽³⁾	4~7
端到端转接时延	6.4.5.24	O ⁽¹⁾	4~7
扩展的 QoS 参数	6.4.5.25	O ⁽¹⁾	4~13
通用标识传送	6.4.5.31	O ^(1,5)	4~33
通知表示语	6.4.5.27	O ⁽¹⁾	4~*
注:			
1 当收到的连接指示中含有此信息时, 包含这个信息单元。			
2 在软 PVPC 或 PVCC 建立时, 包含这个信息单元。			
3 如果 SETUP 消息中包含端点参考, 则是必选信息单元。			
4 如果主叫用户请求 ABR 业务量类别的连接, 则是必选信息单元。			
5 这个信息单元可以在该消息中出现 3 次。			

6.3.1.4 RELEASE

这个消息由一个网络节点向相邻网络节点发送, 表示连接已被清除并等待清除呼叫参考, 见表 54。

表 54 RELEASE 消息内容

消息类型: RELEASE			
方向: 双向			
有效范围: 全网			
信息单元	参考(节号)	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4

续表 54

消息类型: RELEASE			
方向: 双向			
有效范围: 全网			
信息单元	参考(节号)	类型	长度
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
原因	6.4.5.19	M ⁽¹⁾	6~34
折回	6.4.6.3	O ⁽²⁾	4~72
通知表示语	6.4.5.27	O ⁽³⁾	4~*
通用标识传送	6.4.5.31	O ^(3,4)	4~33
注:			
1 这个信息单元可以在该消息中出现两次。			
2 当折回时包含这个信息单元。			
3 当收到的释放指示中含有此信息时, 包含这个信息单元。			
4 这个信息单元可以在该消息中出现 3 次。			

6.3.1.5 RELEASE COMPLETE

这个消息由一个网络节点向相邻网络节点发送, 表示连接已内部清除并释放了呼叫参考, 见表 55。

表 55 RELEASE COMPLETE 消息内容

消息类型: RELEASE COMPLETE			
方向: 双向			
有效范围: 本地(注 1)			
信息单元	参考(节号)	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
原因	6.4.5.19	M ⁽¹⁾	4~34
折回	6.4.6.3	O ⁽²⁾	4~72
注:			
1 这个消息具有局部意义, 但当用作第一个呼叫清除消息时, 可以传送全网意义的信息。			
2 在第一个呼叫清除消息中, 这个信息单元为必备的。当因差错处理而发送这个消息时, 应包含这个信息单元。这个信息单元在这个消息中可以出现两次。			
3 当折回时包含这个信息单元。			

6.3.1.6 SETUP

这个消息由前方侧向后续侧发送, 表示启动呼叫/连接的建立, 见表 56。

表 56 SETUP 消息内容

消息类型: SETUP			
方向: 前方侧至后续侧			
有效范围: 全网			
信息单元	参考 (节号)	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
AAL 参数	6.4.5.8	O ⁽¹⁾	4~21
ABR 附加参数	6.4.5.5	O ⁽¹⁾	4~14
ABR 建立参数	6.4.5.6	O ⁽¹¹⁾	4~36
选择 ATM 业务量描述语	6.4.5.7	O ⁽¹²⁾	4~36
ATM 业务量描述语	6.4.5.9	M	12~30
宽带承载能力	6.4.5.10	M	6~7
宽带高层信息	6.4.5.11	O ⁽¹⁾	4~13
宽带重复表示语	6.4.5.13	O ⁽⁶⁾	4~5
宽带低层信息	6.4.5.12	O ⁽¹⁾	4~17
被叫用户号码	6.4.5.15	M	最大为 25
被叫用户软 PVPC 或 PVCC	6.4.6.2	O ⁽⁴⁾	4~11
被叫用户子地址	6.4.5.16	O ⁽¹⁾	4~25
主叫用户号码	6.4.5.17	O ⁽¹⁾	4~26
主叫用户软 PVPC 或 PVCC	6.4.6.2	O ⁽³⁾	4~10
主叫用户子地址	6.4.5.18	O ⁽¹⁾	4~25
连接标识	6.4.5.22	O ⁽⁵⁾	4~9
连接范围选择	6.4.5.23	O ⁽¹⁾	4~6
指定转接序列	6.4.6.4	M ⁽⁷⁾	33~546
端点参考	6.4.8.1	O ⁽¹⁾	4~7
端到端转接时延	6.4.5.24	O ⁽⁸⁾	4~13
扩展的 QoS 参数	6.4.5.25	O ⁽¹⁰⁾	4~25
通用标识传送	6.4.5.31	O ^(1,9)	4~33
最小可接受的 ATM 业务量描述语	6.4.5.26	O ⁽¹²⁾	4~20
通知表示语	6.4.5.27	O ⁽¹⁾	4~*
QoS 参数	6.4.5.28	O ⁽¹⁾	4~6
转接网选择	6.4.5.30	O ⁽¹⁾	4~9

续表 56

消息类型: SETUP			
方向: 前方侧至后续侧			
有效范围: 全网			
信息单元	参考(节号)	类型	长度
注:			
1 如果收到的建立指示含有这个信息时, 包含这个信息单元。			
2 最小长度由编号计划决定。最大长度为 25 个八位位组。			
3 当主叫端点需通知目的网络接口主叫端所使用的 PVPC 或 PVCC 段的值时, 在软 PVPC 或 PVCC 建立时, 应包含这个信息单元。			
4 在软 PVPC 或 PVCC 建立时, 应包含这个信息单元。			
5 当前方侧需指出特定的 VP 或 VC 时, 应包含这个信息单元。如果不包含, 意味着可以接受任何 VP 或 VC。仅在使用非随路信令程序时, 可以不包含这个信息单元。			
6 当宽带重复表示语信息单元后面是 DTL 信息单元时, 它表示 DTL 堆栈中 DTL 信息单元的顺序。当只有一个 DTL 信息单元时, 这个信息单元是必备的。当宽带重复表示语信息单元后面是任何其他信息单元时, 如果收到的建立指示含有这个信息单元时, 包含这个信息单元。			
7 源节点包含这个信息单元, 表示呼叫的等级源路由。等级层的入口节点包含这个信息单元, 表示通过等级层的路径。这个信息单元可以重复 10 次。			
8 当规定要求端到端转接时延时, 包含这个信息单元。			
9 这个信息单元可以在该消息中出现 3 次。			
10 当规定呼叫要求的个别 QoS 参数时, 包含这个信息单元。			
11 当主叫用户请求一个 ABR 业务量类别连接时, 这个信息单元是必备的。			
12 这个信息单元仅在收到的建立指示中出现。			

6.3.1.7 STATUS

这个消息由前方侧或后续侧发送, 作为对 STATUS ENQUIRY 消息的响应, 或者在任何时候用来报告一些差错情况, 见表 57。

表 57 STATUS 消息内容

消息类型: STATUS			
方向: 双向			
有效范围: 本地			
信息单元	参考(节号)	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
呼叫状态	6.4.5.14	M	5
原因	6.4.5.19	M	6~34

续表 57

消息类型: STATUS			
方向: 双向			
有效范围: 本地			
信息单元	参考(节号)	类型	长度
端点参考	6.4.8.1	O ⁽¹⁾	4~7
端点状态	6.4.8.2	O ⁽²⁾	4~5
注:			
1 当响应一方用户状态的状态询问时或在任何时候报告点到多点程序的特定差错情况时, 包含这个信息单元。			
2 当包含端点参考信息单元时, 包含该信息单元。			

6.3.1.8 STATUS ENQUIRY

这个消息由前方侧或后续侧在任何时刻发送, 用于向对等实体请求 STATUS 消息。对于一个 STATUS ENQUIRY 消息, 必须发送一个 STATUS 消息作为响应, 见表 58。

表 58 STATUS ENQUIRY 消息内容

消息类型: STATUS			
方向: 双向			
有效范围: 本地			
信息单元	参考(节号)	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
呼叫状态	6.4.5.14	M	5
端点参考	6.4.8.1	O ⁽¹⁾	4~7
注: 当询问点到多点程序中一方用户状态时, 包含这个信息单元。			

6.3.1.9 NOTIFY

这个消息由前方侧或后续侧发送, 指示与呼叫/连接有关的信息, 见表 59。

表 59 NOTIFY 消息内容

消息类型: NOTIFY			
方向: 双向			
有效范围: 接入			
信息单元	参考(节号)	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4

续表 59

消息类型: NOTIFY			
方向: 双向			
有效范围: 接入			
信息单元	参考(节号)	类型	长度
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
端点参考	6.4.8.1	O ⁽¹⁾	4~7
通知表示语	6.4.5.27	M	5~*
注: 当收到的通知指示中带有这个信息时, 包含这个单元。			

6.3.2 与支持基于 64kbit/s ISDN 电路方式业务相关的附加消息和修改的消息

表 60 概括了用于基于 64kbit/s 的 ISDN 电路方式业务的 ATM 呼叫和连接的控制消息。

除非在程序中明确指出, 本节中附加的信息单元透明通过 PNN1, 不需要修改。

表 60 基于 64kbit/s ISDN 电路方式业务的 ATM 呼叫/连接的控制消息

消息	参考(节号)
呼叫建立消息:	
ALERTING	6.3.2.1
CONNECT	6.3.2.2
PROGRESS	6.3.2.3
SETUP	6.3.2.5
呼叫清除消息:	
RELEASE	6.3.2.4

6.3.2.1 ALERTING

这个消息由后续侧发送, 表示如果收到 ALERTING, 被叫用户已开始提醒。ALERTING 消息的内容见表 61。

表 61 ALERTING 消息内容

消息类型: ALERTING			
方向: 后续侧至前方侧			
有效范围: 全网			
信息单元	参考(节号)	类型	长度
窄带承载能力	6.4.7.1	O ⁽¹⁾	4~14
窄带高层兼容性	6.4.7.2	O ⁽¹⁾	4~7
进展表示语	6.4.7.4	O ^(1, 2)	4~6
其他信息单元同 6.3.1.1 节。			
注:			
1 当收到的提醒指示中含有此信息时, 包含这个信息单元。			
2 这个信息单元可以在该消息中出现两次。			

6.3.2.2 CONNECT

这个消息由后续侧发送并传递到前方侧，表示被叫用户接受了呼叫/连接。CONNECT 消息的内容见表 62。

表 62 CONNECT 消息内容

消息类型: CONNECT			
方向: 后续侧至前方侧			
有效范围: 全网			
信息单元	参考(节号)	类型	长度
窄带承载能力	6.4.7.1	O ⁽¹⁾	4~14
窄带高层兼容性	6.4.7.2	O ⁽¹⁾	4~7
窄带低层兼容性	6.4.7.3	O ⁽¹⁾	4~20
进展表示语	6.4.7.4	O ^(1, 2)	4~6
其他信息单元同 6.3.1.3 节。			
注:			
1 当收到的连接指示中含有此信息时, 包含这个信息单元。			
2 这个信息单元可以在该消息中出现两次。			

6.3.2.3 PROGRESS

这个消息透明通过, PNNI 不进行任何修改, 表示在互通事件中的呼叫进展情况。PROGRESS 消息的内容见表 63。

表 63 PROGRESS 消息内容

消息类型: PROGRESS			
方向: 后续侧至前方侧			
有效范围: 全网			
信息单元	参考(节号)	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
窄带承载能力	6.4.7.1	O ⁽¹⁾	4~14
窄带高层兼容性	6.4.7.2	O ⁽¹⁾	4~7
通知表示语	6.4.5.27	O ^(1, 2)	4~*
进展表示语	6.4.7.4	O ⁽¹⁾	4~6
注:			
1 当收到的进展指示中含有此信息时, 包含这个信息单元。			
2 这个信息单元可以在该消息中出现两次。			

6.3.2.4 RELEASE

这个消息由一个网络节点向相邻网络节点发送, 表示连接已被清除并等待清除呼叫参考。RELEASE

消息的内容见表 64。

表 64 RELEASE 消息内容

消息类型: RELEASE			
方向: 双向			
有效范围: 全网			
信息单元	参考(节号)	类型	长度
进展表示语	6.4.7.4	O ^(1, 2)	4~6
其他信息单元同 6.3.1.4 节。			
注:			
1 当收到的释放指示中含有此信息时, 包含这个信息单元。			
2 这个信息单元可以在该消息中出现两次。			

6.3.2.5 SETUP

这个消息由前方侧向后续侧发送, 表示启动呼叫/连接的建立。SETUP 消息的内容见表 65。

表 65 SETUP 消息内容

消息类型: SETUP			
方向: 前方侧至后续侧			
有效范围: 全网			
信息单元	参考(节号)	类型	长度
窄带承载能力	6.4.7.1	O ^(1, 2)	4~14
窄带高层兼容性	6.4.7.2	O ^(1, 3)	4~7
宽带重复表示语	6.4.5.13	O ^(1, 4)	4~5
窄带低层兼容性	6.4.7.3	O ^(1, 4)	4~20
进展表示语	6.4.7.4	O ^(1, 3)	4~6
其他信息单元同 6.3.1.6 节。			
注:			
1 当收到的建立指示中含有此信息时, 包含这个信息单元。			
2 这个信息单元可以在该消息中出现 3 次。			
3 这个信息单元可以在该消息中可重复两次。			
4 当含有两个窄带低层兼容性信息单元时, 包含宽带重复表示语。			

6.3.3 使用全局呼叫参考的消息

表 66 为使用全局呼叫参考的消息。

表 66 使用全局呼叫参考的消息

消息	参考(节号)
RESTART	6.3.3.1
RESTART ACKNOWLEDAGE	6.3.3.2
STATUS	6.3.1.7

6.3.3.1 RESTART

这个消息由前方侧或后续侧发送，以请求接受重新启动（即释放所有相关资源）所指示的虚通路/通道，或者是由信令虚通路控制的所有虚通路/通道。RESTART 消息的内容见表 67。

表 67 RESTART 消息内容

消息类型：RESTART			
方向：双向			
有效范围：本地			
信息单元	参考（节号）	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
连接标识	6.4.5.22	O ⁽²⁾	4~9
重新启动表示语	6.4.5.29	M	5
注：			
1 这个消息发送时带有全局呼叫参考。			
2 当需要对指示重新启动的特定虚通路或虚通道时，包含该信息单元。			

6.3.3.2 RESTART ACKNOWLEDGE

这个消息是在收到 RESTART 消息后发送的，并表示所请求的重新启动已完成，RESTART ACKNOWLEDGE 消息的内容见表 68。

表 68 RESTART ACKNOWLEDGE 消息内容

消息类型：RESTART ACKNOWLEDGE			
方向：双向			
有效范围：本地			
信息单元	参考（节号）	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M ⁽¹⁾	4
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
连接标识	6.4.5.22	O ⁽²⁾	4~9
重新启动表示语	6.4.5.29	M	5
注：			
1 这个消息发送时带有全局呼叫参考。			
2 当需要对指示重新启动的特定虚通路或虚通道时，包含该信息单元。			

6.3.4 ATM 点到多点呼叫和连接控制消息

表 69 概括了 ATM 点到多点呼叫和连接控制消息。

表 69 ATM 呼叫和连接控制消息

消息	参考 (节号)
ADD PARTY	6.3.4.1
ADD PARTY ACKNOWLEDGAGE	6.3.4.2
PARTY ALERTING	6.3.4.3
ADD PARTY REJECT	6.3.4.4
DROP PARTY	6.3.4.5
DROP PARTY ACKNOWLEDGAGE	6.3.4.6

6.3.4.1 ADD PARTY

这个消息是在对现有连接增加一方用户时发送的, ADD PARTY 消息的内容见表 70。

表 70 ADD PARTY 消息内容

消息类型: ADD PARTY			
方向: 前方侧至后续侧			
有效范围: 全网			
信息单元	参考 (节号)	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
AAL 参数	6.4.5.8	O ⁽¹⁾	4~21
宽带高层信息	6.4.5.11	O ⁽¹⁾	4~13
宽带低层信息	6.4.5.12	O ⁽¹⁾	4~17
被叫用户号码	6.4.5.15	M	Max=25
主叫用户软 PVPC 或 PVCC	6.4.6.1	O ⁽³⁾	4~10
被叫用户软 PVPC 或 PVCC	6.4.6.2	O ⁽⁴⁾	4~11
被叫用户子地址	6.4.5.16	O ⁽¹⁾	4~25
主叫用户号码	6.4.5.17	O ⁽¹⁾	4~26
主叫用户子地址	6.4.5.18	O ⁽¹⁾	4~25
宽带重复表示语	6.4.5.13	M ⁽⁵⁾	5
指定转接序列	6.4.6.4	M ⁽⁶⁾	33~546
端点参考	6.4.8.1	O ⁽³⁾	5
端到端转接时延	6.4.5.24	O ⁽¹⁾	4~12
通用标识传送	6.4.5.31	O ^(1,8)	4~33
通知表示语	6.4.5.27	O ⁽¹⁾	4~*
转接网络选择	6.4.5.30	O ⁽¹⁾	4~8

续表 70

消息类型: ADD PARTY			
方向: 前方侧至后续侧			
有效范围: 全网			
信息单元	参考(节号)	类型	长度
注:			
1 当收到的增加一方用户指示中含有这个信息时, 则包含这个信息单元。			
3 当主叫端点需通知目的网络接口主叫端所使用的 PVPC 或 PVCC 段的数值时, 在软 PVPC 或 PVCC 建立时, 应包含这个信息单元。			
4 在 PVCC 建立时, 应包含这个信息单元。			
5 表示在 DTL 堆栈中 DTL 信息单元的顺序。当只有一个 DTL 信息单元时, 这个信息单元也是存在的。			
6 源节点可以包含这个信息单元, 表示一方用户连接的等级源路由。在等级层的入口节点中包含这个信息单元表示通过这个等级的路径。这个信息单元可以重复 10 次。			
7 在指定的链路上的指定呼叫参考中, 端点参考必须是唯一的。			
8 这个信息单元可以出现 3 次。			

6.3.4.2 ADD PARTY ACKNOWLEDGE

当 ADD PARTY 请求成功时发送这个消息, ADD PARTY ACKNOWLEDGE 消息的内容见表 71。

表 71 ADD PARTY ACKNOWLEDGE 消息内容

消息类型: ADD PARTY ACKNOWLEDGE			
方向: 后续侧至前方侧			
有效范围: 全网			
信息单元	参考(节号)	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
宽带低层信息	6.4.5.12	O ⁽¹⁾	4~17
AAL 参数	6.4.5.8	O ⁽¹⁾	4~11
被叫用户软 PVPC 或 PVCC	6.4.6.2	O ⁽¹⁾	4~11
被连接号码	6.4.5.20	O ⁽¹⁾	4~26
被连接子地址	6.4.5.21	O ⁽¹⁾	4~25
端点参考	6.4.8.1	M ⁽²⁾	7
端到端转接时延	6.4.5.24	O ⁽¹⁾	4~7
通用标识传送	6.4.5.31	O ^(1,3)	4~33
通知表示语	6.4.5.27	O ⁽¹⁾	4~*
注:			
1 当收到的增加一方用户证实指示中含有这个信息时, 则包含这个信息单元。			
2 端点参考必须与相应的 ADD PARTY 消息中的值相同。			
3 这个信息单元可以出现 3 次。			

6.3.4.3 PARTY ALERTING

这个消息由后续侧发送，表示在收到 ALERTING 时，被叫用户的提醒已经开始，PARTY ALERTING 消息的内容见表 72。

表 72 PARTY ALERTING 消息内容

消息类型: PARTY ALERTING			
方向: 后续侧至前方侧			
有效范围: 全网			
信息单元	参考(节号)	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
端点参考	6.4.8.1	M	7
通知表示语	6.4.5.27	O ⁽¹⁾	4~*
通用标识传送	6.4.5.31	O ^(1,2)	4~33
注:			
1 当收到一方用户提醒指示中含有这个信息时, 则包含这个信息单元。			
2 这个信息单元可以出现 3 次。			

6.3.4.4 ADD PARTY REJECT

当证实 ADD PARTY 请求不成功时发送这个消息，ADD PARTY REJECT 消息的内容见表 73。

表 73 ADD PARTY REJECT 消息内容

消息类型: ADD PARTY REJECT			
方向: 后续侧至前方侧			
有效范围: 全网			
信息单元	参考(节号)	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
原因	6.4.5.19	M	4~34
折回	6.4.6.3	O ⁽¹⁾	4~72
端点参考	6.4.8.1	M ⁽²⁾	7
通用标识传送	6.4.5.31	O ^(3,4)	4~33
注:			
1 指示折回时, 包含这个信息单元。			
2 端点参考必须与相应的 ADD PARTY 消息中的值相同。			
3 当收到的增加一方用户拒绝指示中含有这个信息时, 则包含这个信息单元。			
4 这个信息单元可以出现 3 次。			

6.3.4.5 DROP PARTY

当从现有的点到多点连接中拆除一方用户时发送这个消息。DROP PARTY 消息的内容见表 74。

表 74 DROP PARTY 消息内容

消息类型: DROP PARTY			
方向: 双向			
有效范围: 全网			
信息单元	参考(节号)	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
原因	6.4.5.19	M	4~34
端点参考	6.4.8.1	M	7
通知表示语	6.4.5.27	O ⁽¹⁾	4~*
通用标识传送	6.4.5.31	O ^(1,2)	4~33
注:			
1 当收到一方用户拆除指示中含有这个信息时, 则包含这个信息单元。			
2 这个信息单元可以出现 3 次。			

6.3.4.6 DROP PARTY ACKNOWLEDGE

证实 DROP PARTY 消息时发送这个消息, 表示一方用户已从连接上拆除。DROP PARTY ACKNOWLEDGE 消息的内容见表 75。

表 75 DROP PARTY ACKNOWLEDGE 消息内容

消息类型: DROP PARTY ACKNOWLEDGE			
方向: 双向			
有效范围: 本地			
信息单元	参考(节号)	类型	长度
协议鉴别语	6.4.2	M	1
呼叫参考	6.4.3	M	4
消息类型	6.4.4.1	M	2
消息长度	6.4.4.2	M	2
原因	6.4.5.19	O ⁽¹⁾	4~34
端点参考	6.4.8.1	M	7
注: 当 DROP PARTY ACKNOWLEDGE 作为差错情况的结果发送时, 这个消息单元是必备的。这个消息单元可以出现两次。			

6.4 消息的一般格式和信息单元编码

本章的图和正文描述了消息的内容。

6.4.1 概述

同 Q.2931 标准的 4.1 节。其中，紧接 Q.2931 标准的图 4-1 后的段落中的最后一句不用。

6.4.2 协议鉴别语

同 Q.2931 标准的 4.2 节。修改部分如下。

第一段第一句中“用户—网络”改为“PNNI”。

协议鉴别语的编码见图 24。

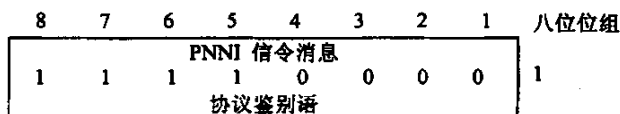


图 24 协议鉴别语

6.4.3 呼叫参考

同 Q.2931 标准的 4.3 节。修改部分如下。

第一段第一句中“用户—网络”改为“PNNI”。

6.4.4 消息类型和消息长度

6.4.4.1 消息类型

同 Q.2931 标准的 4.3 节。修改部分如下。

在 Q.2931 中的图 4-6，消息类型的格式中第二个八位位组中第 4 个比特由“备用”改为“传递请求”。

Q.2931 标准中表 4-2 的续表中的修改如下：

——传递请求（第二个八位位组）

比特 4：

1 传递请求

0 无传递请求

下列消息类型不支持：

CONNECT ACKNOWLEDGE；

SETUP ACKNOWLEDGE；

INFORMATION。

用于国内特定消息类型不支持。

6.4.4.2 消息长度

同 Q.2931 标准的 4.4.2 节。

6.4.5 可变长信息单元

6.4.5.1 编码原则

同 Q.2931 标准的 4.5.1 节。修改部分如下：

Q.2931 标准的图 4-8 的一般信息单元格式中，第二个八位位组的第 4 比特由“备用”改为“传递请求”。表 4-3（续表）修改为：

——传递请求（第二个八位位组）

比特 4

1 传递请求

0 无传递请求

表 76 列出了 PNNI 支持的信息单元的最大长度和出现的最大次数。

表 76 PNNI 中使用的信息单元

比特		信息单元	最大长度	出现的最大次数
8 7 6 5	4 3 2 1			
0000	0100	窄带承载能力 ^(1, 2)	14	3
0000	1000	原因 ⁽¹⁾	34	2
0001	0100	呼叫状态	5	1
0001	1110	进展表示语 ⁽¹⁾	6	2
0010	0111	通知表示语	(3)	(3)
0100	0010	端到端转接时延	13	1
0100	1100	被连接号码	26	1
0100	1101	被连接子地址	25	1
0101	0100	端点参考	7	1
0101	0101	端点状态	5	1
0101	1000	AAL 参数	21	1
0101	1001	ATM 业务量描述语	30	1
0101	1010	连接标识	9	1
0101	1100	QoS 参数	6	1
0101	1101	宽带高层信息	13	1
0101	1110	宽带承载能力	7	1
0101	1111	宽带低层信息 ⁽²⁾	17	3
0110	0000	宽带锁定转换程序	5	(4)
0110	0001	宽带非锁定转换程序	5	(4)
0110	0011	宽带重复表示语 ⁽¹⁾	5	3
0110	1100	主叫用户号码	26	1
0110	1101	主叫用户子地址 ⁽¹⁾	25	2
0111	0000	被叫用户号码	25	1
0111	0001	被叫用户子地址 ⁽¹⁾	25	2
0111	1000	转接网选择	9	1
0111	1001	重新启动表示语	5	1
0111	1100	窄带低层兼容性 ⁽²⁾	20	2
0111	1101	窄带高层兼容性 ⁽¹⁾	7	2
0111	1111	通用标识传送	33	3
1000	0001	最小可接受的 ATM 业务量描述语	20	1
1000	0010	选择 ATM 业务量描述语	30	1
1000	0100	ABR 建立参数	36	1

续表 76

比特		信息单元	最大长度	出现的最大次数
1110	0000	被叫用户软 PVPC 或 PVCC	11	1
1110	0001	折回	72	1
1110	0010	指定转接序列	546	1
1110	0011	主叫用户软 PVPC 或 PVCC	10	1
1110	0100	ABR 附加参数	14	1
1110	1011	连接范围选择	6	1
1110	1100	扩展的 QoS 参数	25	1
<p>注 1: 该信息单元重复时可以不包含宽带重复表示语信息单元。</p> <p>注 2: 该信息单元重复时可以包含宽带重复表示语信息单元。</p> <p>注 3: 该信息单元的最大长度和重复次数由网络确定。</p> <p>注 4: 详见 6.5.6.8 节的 1) 条。</p> <p>保留的用于信息单元标识的值 '1111 1111' 不支持。</p>				

6.4.5.2 码组扩展

本规范仅使用码组 0 中的信息单元。

6.4.5.3 宽带锁定转换程序

宽带锁定转换信息单元的标识是必备的。所有的转换信息单元按不认识的信息单元处理。见 Q.2931 的 5.6.6 节。

6.4.5.4 宽带非锁定转换程序

宽带非锁定转换信息单元的标识是必备的。所有的转换信息单元按不认识的信息单元处理。见 Q.2931 的 5.6.6 节。

6.4.5.5 ABR 附加参数

同 UNI4.0 信令规范 10.1.2.1 节。

6.4.5.6 ABR 建立参数

同 UNI4.0 信令规范 10.1.2.2 节。

6.4.5.7 选择 ATM 业务量描述语

同 UNI4.0 信令规范 8.1.2.1 节。

6.4.5.8 AAL 参数

同 UNI4.0 信令规范的 4.5.5 节。

6.4.5.9 ATM 业务量描述语

同 UNI4.0 信令规范的 4.5.6 节。并对 UNI4.0 信令规范 10.1.2.3 节的内容做如下修改:

Tb(后向标注) (第 17.1 个八位位组)

比特 2	在 CONNECT 消息中(注 1)	在 SETUP 消息中(注 2)
0	不允许标注	不支持标注
1	已请求标注	支持标注

注 1: 在目的侧,若省略第 17.1 个八位位组,应使用缺省值 '不允许标注'。

注 2: 在目的侧,若省略第 17.1 个八位位组,应使用缺省值 '不支持标注'。

Tf(前向标注) (第 17.1 个八位位组)

比特 1	在 SETUP 消息中(注 3)	在 CONNECT 消息中(注 4)
------	------------------	--------------------

0 不允许标注 不使用标注
 1 已请求标注 使用标注

注 3: 在目的侧, 若省略第 17.1 个八位位组, 应使用缺省值 ‘不允许标注’。

注 4: 在目的地侧, 若省略第 17.1 个八位位组, 应使用缺省值 ‘不使用标注’。

Tb= ‘不允许标注’: 来自后续侧交换机的后向业务量将被标注;

Tb= ‘已请求标注’: 来自后续侧交换机的后向业务量可能未被 ‘标注’ 并仅需与 CLP=0+1 业务量参数一致;

Tb= ‘不支持标注’: 来自后续侧交换机的后向业务量必须已被标注;

Tb= ‘支持标注’: 前方侧交换机将支持来自后续侧交换机的未被标注’ 的后向业务量并仅需与 CLP=0+1 业务量参数一致;

Tf= ‘不允许标注’: 来自前方侧交换机的前向业务量将被标注;

Tf= ‘已请求标注’: 前方侧交换机请求允许向后续侧交换机发送未被 ‘标注’ 的后向业务量并仅需与 CLP=0+1 业务量参数一致;

Tf= ‘不使用标注’: 来自前方侧交换机的前向业务量必须已被标注;

Tf= ‘已使用标注’: 后续侧交换机将允许来自后续侧交换机的未被 ‘标注’ 的后向业务量并仅需与 CLP=0+1 业务量参数一致;

6.4.5.10 宽带承载能力

同 UNI4.0 信令规范的 4.5.7 节。

6.4.5.11 宽带高层信息

同 UNI4.0 信令规范的 4.5.8 节。

6.4.5.12 宽带低层信息

同 UNI4.0 信令规范的 4.5.9 节。

6.4.5.13 宽带重复表示语

同 Q.2931 的 4.5.9 节。修改部分为第 5 个八位位组。

宽带重复指示 (第 5 个八位位组)

比特 4 3 2 1 含义

0 0 1 0 选择某一种可能性的优先级

1 0 1 0 堆栈进出方式为: 后进先出 (注)

注: 最后出现的重复信息单元放在堆栈顶部, 倒数第 2 个、倒数第 3 个出现的重复信息单元分别在堆栈的第 2 和第 3 位。

6.4.5.14 呼叫状态

呼叫状态信息单元的用途是描述 PNNI 或在全局接口的呼叫的现行状态, 其信息单元的格式见图 25。

8	7	6	5	4	3	2	1	八位位组
呼叫状态信息单元								1
0	0	0	1	0	1	0	0	
1 ext	编码标准		IE 指令字段					2
呼叫状态内容的长度								3
呼叫状态内容的长度 (继续)								4
0	0	PNNI呼叫状态值/全局接口状态值						5
备用								

图 25 呼叫状态信息单元

PNNI 呼叫状态值 (第五个八位位组) 见表 77。

表 77

比特	含义
65 4321	
00 0000	NN0—零态
00 0001	NN1—呼叫起始
00 0011	NN3—呼叫进展发送
00 0100	NN4—提醒递交
00 0110	NN6—呼叫呈现
00 0111	NN6—提醒接收
00 1001	NN9—呼叫进展接收
00 1010	NN10—运行
00 1011	NN11—释放请求
00 1100	NN12—释放指示

注：PNNI 呼叫状态值的编码采用的编码标准为“11”的 ATM 论坛标准。

全局接口状态值（第五个八位位组）见表 78。

表 78

比特	含义
65 4321	
00 0000	REST 0—零态
11 1101	REST 1—重新启动请求
11 1110	REST 2—重新启动

6.4.5.15 被叫用户号码

同 UNI4.0 信令规范的 4.5.11 节。

6.4.5.16 被叫用户子地址

同 UNI4.0 信令规范的 4.5.12 节。

6.4.5.17 主叫用户号码

同 UNI4.0 信令规范的 4.5.13 节。

6.4.5.18 主叫用户子地址

同 UNI4.0 信令规范的 4.5.14 节。

6.4.5.19 原因

同 UNI4.0 信令规范的 4.5.15 节。PNNI 特定的原因值如表 79，编码标准为 11。

表 79

比特	编号	含义	诊断
7654321			
0110101	53	由于在 PGL 中改变而清除呼叫	
0100010	34	请求的被叫用户软 PVPC 或 PVCC 不可用	

6.4.5.20 被连接号码

同 UNI4.0 信令规范的附件 A4.5。

6.4.5.21 被连接子地址

同 UNI4.0 信令规范的附件 A4.5。

6.4.5.22 连接标识

同 UNI4.0 信令规范的 4.5.16 节。

修改如下：支持随路信令。

6.4.5.23 连接范围选择

同 UNI4.0 信令规范的 7.1.3.1 节。

6.4.5.24 端到端转接时延

端到端转接时延信息单元的用途是指在连接中基于每个呼叫的前向最大的可接受的信元传送时延和连接中期望的前向积累最大信元传送时延。后向积累最大信元传送时延可以从前向积累最大信元传送时延和前向积累与后向积累信元延迟变化中获得。

端到端转接时延的编码见图 26。

这个信息单元的最大长度为 13 个八位位组。

8	7	6	5	4	3	2	1	八位位组
端到端转接时延信息单元标识								1
0	1	0	0	0	0	1	0	
1	1	1	IE 指示字段					2
ext	ATM 论坛标准							
端到端转接时延内容的长度								3
端到端转接时延内容的长度 (继续)								4
前向积累最大信元传送时延标识								5*
0	0	0	0	0	0	0	1	(注2)
前向积累最大信元传送时延								5.1*
前向积累最大信元传送时延 (继续)								5.2*
PNNI 可接受的前向最大信元传送时延标识								6*
0	0	0	0	1	0	1	1	
PNNI 可接受的前向最大信元传送时延								6.1*
PNNI 可接受的前向最大信元传送时延 (继续)								6.2*
PNNI 可接受的前向最大信元传送时延 (继续)								6.3*
PNNI 前向积累最大信元传送时延标识								7*
0	0	0	1	0	0	0	1	(注2)
PNNI 前向积累最大信元传送时延								7.1*
PNNI 前向积累最大信元传送时延 (继续)								7.2*
PNNI 前向积累最大信元传送时延 (继续)								7.3*
网络产生的表示语								8*
0	0	0	0	1	0	1	0	(注1)

图 26 端到端转接时延

注 1：只有当这个信息单元起源不是发端用户时包含这个信息单元。

注 2：八位位组 5 和 7 是相互独立的。CONNECTION 和 ADD PARTY ACKNOWLEDGE 消息中使用八位位组 5，SETUP 和 ADD PARTY 消息中使用八位位组 7。

——前向积累最大信元传送时延（八位位组 5.1~5.2）。

前向积累最大信元传送时延值以 ms 为单位。编码采用 16bit 的二进制整数，第一个八位位组的比特 8 为最高有效位，第二个八位位组的比特 1 为最低有效位。

——PNNI 可接受的前向最大信元传送时延（八位位组 6.1~6.3）。

PNNI 可接受的前向最大信元传送时延参数表示主叫用户最大可接受的前向最大信元传送时延值。编码采用 24bit 的二进制整数，第一个八位位组的比特 8 为最高有效位，第三个八位位组的比特 1 为最低有效位。然而，值“1111 1111 1111 1111 1111 1111”不应解释为最大可接受的信元传送时延值，该编码表示“任何前向最大信元传送时延值可接受”。

——PNNI 前向积累最大信元传送时延（八位位组 7.1~7.3）。

PNNI 前向积累最大信元传送时延值以 ms 为单位。编码采用 24bit 的二进制整数，第一个八位位

组的比特 8 为最高有效位，第三个八位位组的比特 1 为最低有效位。

——网络产生表示语（八位位组 8）。

如果这个字段不存在，则这个信息单元是由发端用户启动的（所以被叫用户收到的积累值为端到端的值）。如果含有这个字段，则表示这个信息单元不是由发端用户发送的。

6.4.5.25 扩展的 QoS 参数

同 UNI4.0 信令规范的 9.1.2.2 节。

6.4.5.26 最小可接受的 ATM 业务量描述语

同 UNI4.0 信令规范的 8.1.2.2 节。

6.4.5.27 通知表示语

见 UNI4.0 信令规范的附件 A7.1.2.1 和 Q.2931 的 4.5.12 节。

6.4.5.28 QoS 参数

同 UNI4.0 信令规范的 4.5.18 节。

6.4.5.29 重新启动表示语

同 UNI4.0 信令规范的 4.5.20 节。

6.4.5.30 转接网络选择

同 UNI4.0 信令规范的 4.5.22 节。

6.4.5.31 通用标识传送

同 UNI4.0 信令规范的 2.1.1 节。

6.4.6 PNNI 特有的信息单元

6.4.6.1 主叫用户软 PVPC 或 PVCC

主叫用户软 PVPC 或 PVCC 信息单元的用途是指示主叫连接点使用的 PVC 段的 VPI 或 VPI/VCI 值。这些值透明传送到被叫连接点。主叫用户软 PVPC 或 PVCC 信息单元的编码见图 27。

8	7	6	5	4	3	2	1	八位位组
主叫用户软PVPC或PVCC								1
1	1	1	0	0	0	1	1	
1 ext	1 1 ATM论坛标 准		IE 指令字段					2
主叫用户软PVPC或PVCC内容的长度								3
主叫用户软PVPC或PVCC内容的长度(继续)								4
VPI标识								5*
1	0	0	0	0	0	0	1	
VPI值								5.1* 5.2*
VCI标识								6*
1	0	0	0	0	0	1	0	
VPI值								6.1* 6.2*

图 27 主叫用户软PVPC或PVCC信息单元

——VPI 值（八位位组 5.1 和 5.2）。

ATM 连接用两个八位位组的二进制码表示 VPI 的标识。VPI 值用低 12bit 编码，4 个高比特位（八位位组 5.1 中比特 8~5）编码为 0。

——VCI 值（八位位组 6.1 和 6.2）。

ATM 连接用两个八位位组的二进制码表示虚通路连接的标识。

6.4.6.2 被叫用户软 PVPC 或 PVCC

被叫用户软 PVPC 或 PVCC 信息单元的用途是指示被叫连接点和 PVPC 或 PVCC 用户间 PVC 段的 VPI 或 VPI/VCI 值。这些值透明传送到被叫连接点。被叫用户软 PVPC 或 PVCC 信息单元的编码见图 28。

							7	6	5	4	3	2	1	八位位组
被叫用户软PVPC或PVCC														
1 1 1 0 0 0 0 0														
1 1 1 IE 指令字段														
1 ext ATM论坛标准														
被叫用户软PVPC或PVCC内容的长度														
被叫用户软PVPC或PVCC内容的长度(继续)														
选择类型														
VPI标识														
1 0 0 0 0 0 0 1														
VPI值														
VCI标识														
1 0 0 0 0 0 1 01														
VCI值														
1														
2														
3														
4														
5														
6														
6.1														
6.2														
7														
(注1, 2)														
7.1														
7.2														

注1: 当选择类型为“任何值”时, 不包含这组八位位组。如果存在, 则忽略这组八位位组。

注2: 这组八位位组仅在软PVCC时存在。

图 28 被叫用户软PVPC或PVCC信息单元

选择类型 (八位位组 5) 见表 80。

表 80

比特	含义
8 7 6 5 4 3 2 1	
0000 0000	任何值
0000 0010	所需值
0000 0100	分配值

——VPI 值 (八位位组 6.1 和 6.2)。

ATM 连接用两个八位位组的二进制码表示 VPI 的标识。VPI 值用低 12bit 编码, 4 个高比特位 (八位位组 6.1 中比特 8~5) 编码为 0。

——VCI 值 (八位位组 7.1 和 7.2)。

ATM 连接用两个八位位组的二进制码表示虚通路连接的标识。

6.4.6.3 折回

折回信息单元的用途是表示折回程序已经启动。它也表示不能在节点或链路上接受呼叫/连接或增加一方用户, 且折回过程已经 PNN1 层次的等级中执行。折回信息单元的编码见图 29。

								8	7	6	5	4	3	2	1	八位位组
折回信息单元标识																
1 1 1 0 0 0 0 1																
1 1 1 IE 指令字段																
1 ext ATM论坛标准																
折回内容的长度																
折回内容的长度(继续)																
折回等级																
阻塞转接类型																
阻塞转接标识																
(格式和长度取决于阻塞转接类型的值)																
折回原因																
折回原因诊断																
1																
2																
3																
4																
5																
6																
6.1等																
7(注)																
7.1																

图 29 折回信息单元

——折回等级（八位位组 5）。

折回等级表示进行呼叫/连接折回的 PNNI 等级。折回等级采用一个八位位组的二进制编码，

——阻塞转接类型（八位位组 6）。

见表 81。

表 81

比特	含义	阻塞转接标识的长度
8 7 6 5 4 3 2 1		
0 0 0 0 0 0 1 0	在这个接口的后续端，呼叫或一方用户已阻塞	0
0 0 0 0 0 0 1 1	阻塞的节点	22
0 0 0 0 0 1 0 0	阻塞的链路	48

——阻塞转接标识（八位位组 6.1 等）。

——阻塞转接标识的格式取决于阻塞转接类型。

当阻塞转接类型 = “阻塞节点标识”，八位位组 6.1~6.22 如下：

阻塞节点标识 (格式和长度取决于阻塞转接类型的值)	6.1~6.22
------------------------------	----------

——阻塞节点标识（八位位组 6.1~6.22）

阻塞节点标识表示呼叫/连接或一方用户已阻塞的逻辑节点。它采用 22 个八位位组的二进制编码。

当阻塞转接类型 = “阻塞链路标识”

阻塞链路前方节点标识	6.1 到 6.22
阻塞链路的端口标识	6.23 到 6.26
阻塞链路后续节点标识	6.27 到 6.48

阻塞链路前方节点标识（八位位组 6.1~6.22）

阻塞链路前方节点标识表示呼叫/连接或一方用户发生阻塞的链路的前方逻辑节点。它采用 22 个八位位组的二进制编码。

阻塞链路的端口标识（八位位组 6.23~6.26）。

阻塞链路的端口标识表示阻塞链路前方节点标识的逻辑端口。阻塞链路前方节点标识和阻塞链路端口标识共同明确地（但不是唯一地）标识了呼叫/连接或一方用户发生阻塞的链路。端口标识 0 表示阻塞链路前方节点至阻塞链路的后续节点间的所有链路均阻塞。逻辑端口标识采用 4 个八位位组的二进制编码。

阻塞链路后续节点标识（八位位组 6.27~6.48）。

阻塞链路后续节点标识表示呼叫/连接或一方用户发生阻塞的链路的后续逻辑节点。它采用 22 个八位位组的二进制编码。

表 82

比特	号码	含义	诊断
8 7 6 5 4 3 2 1			
0 0 0 0 0 0 1 0	2	转接网不可达	
0 0 0 0 0 0 1 1	3	目的地不可达	
0 0 1 0 0 0 0 0	32	过多悬置增加一方用户请求	
0 0 1 0 0 0 1 1	35	请求的 VPCI/VCI 不可用	

续表 82

比特	号码	含义	诊断
8765 4321			
0010 0101	37	用户信元率不可用	注 1
0010 0110	38	网络故障	
0010 1001	41	临时故障	
0010 1101	45	无 VPCI/VCI 可用	
0010 1111	47	资源不可用, 未指定	
0011 0001	49	业务质量不可用	注 2
0011 1001	57	承载能力未规定	
0011 1010	58	承载能力目前不可用	
0011 1111	63	业务或选择不可用, 未指定	
0100 0001	65	承载业务未实现	
0100 1001	73	业务参数的组合不支持	
1000 0000	128	下个节点不可达	
1010 0000	160	DTL 转接不本身节点的 ID	

注 1: CAC 的更新拓扑状态参数可作为任选包含在折回原因诊断中。这些拓扑状态参数的使用可以替代收到的有关阻塞节点或链路的最新 GCAC。从 PTSE 或其他呼叫清除消息中收到的下一组拓扑状态参数将替代当前的拓扑状态参数。如果包含 AvCR, CRM 和 VF 时, 将使用复杂 GCAC。如果仅包含 AvCR, 则使用简单的 GCAC。图 30 中的编码仅能包含一个方向的更新拓扑状态参数。

折回原因诊断 (7.1 八位位组等)。

折回原因诊断和诊断编码格式的使用根据不同的原因值而变化。表 82 中诊断一栏表示如何使用诊断以及诊断字段编码格式。

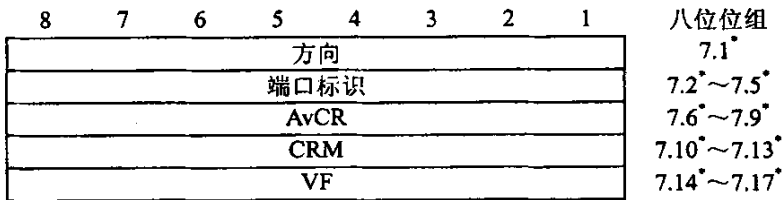


图 30

——方向 (八位位组 7.1) 见表 83。

表 83

比特	含义
8765 4321	
0000 0000	前向
0000 0001	后向

——端口标识 (八位位组 7.2~7.5)。

端口标识是在使用更改拓扑状态参数的链路前方节点上。该端口标识由阻塞链路前方节点确定。如果阻塞的转接是一个节点或由阻塞链路的后续侧发送时, 端口标识设置为 0。

如果方向 (八位位组 7.1) 是前向的, 阻塞链路前方节点端口的出口方向使用更新的拓扑状态参数。

如果方向为后向且阻塞链路是一条水平链路，则阻塞链路的后续节点的相应端口的出口方向使用更新的拓扑状态参数。如果方向为后向且阻塞链路是垂直链路，则阻塞链路的前方节点端口的入口方向使用拓扑状态参数。

注 2：下列编码将被采用：

8	7	6	5	4	3	2	1	八位位组
备用			CTD		CDV	CLR	其他 QoS	7.1

CTD (八位位组 7.1)

比特	含义
4	
0	CTD 可用
1	CTD 不可用

CDV (八位位组 7.1)

比特	含义
3	
0	CDV 可用
1	CDV 不可用

CLR (八位位组 7.1)

比特	含义
2	
0	CLR 可用
1	CLR 不可用

其他 QoS 参数 (八位位组 7.1)见表 84。

表 84

比特	含义
1	
0	其他 QoS 参数可用
1	其他 QoS 参数不可用

6.4.6.4 指定转接序列

指定转接序列目的指出当连接穿过在某一等级的对等组所经过的逻辑节点和逻辑链路。指定转接序列的编码见图 31。

8	7	6	5	4	3	2	1	八位位组
指定转接序列信息单元标识								1
1	1	1	0	0	0	1	0	1
1 ext	编码标准	IE 操作指示						2
指定转接序列内容的长度								3
指定转接序列内容的长度 (继续)								4
当前转接指针								5
当前转接指针 (继续)								6
逻辑节点/逻辑端口表示语								7 (注3)
0	0	0	0	0	0	0	1	7.1~7.22
逻辑节点标识								7.1~7.22
逻辑端口标识								7.23~7.26

注 1：指定转接序列是一个顺序表。指定转接序列信息单元中八位位组的解释与位置有关。

注 2：指定转接序列信息单元的结构应使每个传送能够将目前传送指针指到指定转接序列中的下一个节点。

注 3：八位位组 7 可出现 20 次。

图31 指定转接序列信息单元

——编码标准 (第 2 个八位位组)见表 85。

表 85

比特	含义
7 6	
1 1	ATM 论坛标准

——当前转接指针 (第 5 个八位位组到第 6 个八位位组)。

当前转接指针编码为指向按照指定转接序列进行的当前呼叫/连接所用的传送节点/连接端口的八位位组偏移量。当一个节点收到 DTL 时, 这个指针指向这个节点的 ID 或这个节点父辈的 ID。它采用 16bit 无符号整数编码。值 0 表示信息单元中的第一个传送 (即产生该信息单元的节点)。值 27 表示第二个传送 (即产生该信息单元的节点的最近相邻点), 值 54 表示第三个传送, 依此类推。

——逻辑节点标识 (第 7.1~7.22 个八位位组)。

逻辑节点标识唯一地标识了呼叫/连接传送的逻辑节点 (即一个真正的交换系统或在某些等级的一个等级组)。它采用 22 个八位位组的二进制编码。

——逻辑端口标识 (第 7.23~7.26 个八位位组)。

逻辑端口标识唯一地标识了呼叫/连接传送的逻辑节点的逻辑端口。逻辑节点标识和逻辑端口标识的组合明确地 (但不是唯一地) 标识了一条逻辑链路。逻辑端口标识采用 4 个八位位组的二进制编码。值 H'00000000 为保留值, 表示没有定义逻辑端口 (即这个传送仅定义了逻辑节点)。

6.4.7 支持基于 64kbit/s 电路方式业务的信息单元

本节中描述的信息单元将透明通过 PNNI, 除非在程序中有特殊说明。

6.4.7.1 窄带承载能力

同 Q.2931 的 4.6.2 节。

6.4.7.2 窄带高层兼容性

同 Q.2931 的 4.6.3 节。

6.4.7.3 窄带低层兼容性

同 Q.2931 的 4.6.4 节。

6.4.7.4 进展表示语

同 Q.2931 的 4.6.5 节。

6.4.8 支持点到多点呼叫/连接控制的信息单元

6.4.8.1 端点参考值

同 Q.2971 的 8.2.1 节。

6.4.8.2 端点状态

同 Q.2971 的 8.2.2 节。

6.5 ATM 点一点呼叫/连接控制过程

本节仅描述点一点呼叫过程。6.6 节描述点一多点呼叫的附加过程, 以及为支持点一多点呼叫而对点一点过程所作的修改。

点一点呼叫过程是用来建立 ATM 交换虚连接。它是基于 ATM 论坛用户—网络接口(UNI)和 ITU-T 建议 Q.2931。对于非随路信令, 信令虚通路用 VPI=0, VCI=5; 对于随路信令, VCI=5 用作 VPC 中的信令通路。

本标准允许任选支持交换虚通道(SVP)。对 SVC 的参考除非注明也同样用于 SVP。

6.5.1 信令 AAL 的建立

在调用这些过程前, 必须在两个专用网络节点间建立确保方式的信令 AAL 连接。所有第三层的消息用 AALDATAREQUEST 原语发送给信令 AAL。

6.5.2 呼叫/连接建立

6.5.2.1 呼叫/连接请求

呼叫建立由前侧发送 SETUP 消息启动。前侧应启动定时器 T303 并进入呼叫出现状态。消息中包括一个呼叫参考,它是按照 6.4.3 节中给出的过程选择。这个消息的发送仅当呼叫资源可用,否则,呼叫向着主叫用户清除。

SETUP 消息应包括处理呼叫所需的所有信息。其中被叫地址信息包括在被叫号码信息单元中(可包含被叫子地址信息单元)。在 SETUP 消息中,ATM 业务量描述语、宽带承载能力和服务质量参数信息单元是必备的。

当 SETUP 消息的被叫号码信息单元中包括 ATM 组地址时,这表明呼叫进一步应转给在连接范围选择信息单元(即这个呼叫用了 ATM 任播能力)中指定的连接范围中的组的成员之一。这样,若 SETUP 消息中无连接范围选择信息单元,则节点应假定一个本地网的缺省的连接范围选择。

若在第一个定时器 T303 超时之前,前侧未收到对 SETUP 消息的响应,则可重发 SETUP 消息且再启动定时器 T303。

若前侧在最后一个定时器 T303 超时之后未收到对 SETUP 消息的响应,则前侧进入空(Null)状态并向后侧发送带原因值为#102(定时器超时时恢复)的启动向主叫清除(无折回)的 RELEASE COMPLETE 消息。呼叫控制应被通知呼叫失败。

后侧收到 SETUP 消息后应进入呼叫起始状态。

6.5.2.2 连接标识分配/选择

有两种情况:

——随路信令。

第三层信令实体仅控制承载信令 VC 的 VPC 中的 VC。

——非随路信令。

第三层信令实体控制承载信令 VC 的 VPC 中的 VC,也可控制其他 VPC 中 VCC。

在 PNNI 中, VPI=0/VCI =5 的虚通路仅用于非随路信令。这个信令通路不控制配置给随路信令使用的 VPC 中的虚通路,但控制在物理链路上的所有剩余的虚通路和虚通道。

PNNI 接口应支持非随路信令过程,并作为任选,支持随路信令过程。随路信令过程仅用在当两个 PNNI 网络节点是由一个用作逻辑链路的虚通道连接时。

当一个网络节点收到一个连接标识信息单元,它的 VP 随路信令字段(见本标准 6.4.5.22 节)的编码包括这个网络节点不支持的值,呼叫应以原因#36(VPCI/VCI 分配失败)拒绝。

1) 随路信令。

对于随路信令,前侧在承载信令 VC 的 VPC 中请求一个虚通路。承载信令 VC 的 VPC 是被隐含指示的。

在连接标识信息单元中,VP 随路信令字段编码为“VP 随路信令”。下列值之一在优选/指定字段中指示:

- a) 指定的 VPCI; 任何 VCI;
- b) 指定的 VPCI; 指定的 VCI。

在 a)中,后侧在承载信令 VC 的 VPC 中选择任何一个可用的 VCI。

在 b)中,若所指示的 VCI 在承载信令 VC 的 VPC 中是可用的,则后侧选择它。

选择的 VCI 值在后侧响应(SETUP 消息)的第一个消息(即 CALL PROCEEDING 消息)的连接标识信息单元中指示。VP 随路信令字段编码为“VP 随路信令”。优选/指定的字段编码为“指定的 VPCI; 指定的 VCI”。

在 a)中,若没有可用的 VCI,后侧发送 RELEASE COMPLETE 消息,原因为#45(无可用的 VPCI/VCI)。此外,折回信息单元包括折回原因#45。

在 b)中,若所指示的 VCI 不可用,后侧发送 RELEASE COMPLETE 消息,原因为#35(请求的 VPCI

/VCI 不可用)。此外, 折回信息单元包括折回原因#35。

当一个接口的两侧同时传送指示同一 VPCI /VCI 的 SETUP 消息时, 可能发生呼叫冲突。对于 PNNI 接口, 为避免呼叫冲突, 有较高节点标识侧(见 5.3.3 节)应分配连接标识(VPCI ,VCI)值。有较高节点标识的前侧应在 SETUP 消息中包括连接标识为选项 b)的连接标识信息单元。有较低节点标识的前侧应在 SETUP 消息中用选项 a)。

2) 非随路信令。

a) 交换虚通路的分配。

当前侧在 SETUP 消息中请求一个虚通路, 前侧应指示下列之一:

i) 指定的 VPCI; 任何 VCI;

ii) 指定的 VPCI; 指定的 VCI;

iii) 无指示(即 SETUP 消息中不包括连接标识信息单元)。

在 i)和 ii)中, 连接标识信息单元中的 VP 随路信令字段编码为“VPCI 的明确指示”。

在 i)和 ii)中, 若所指示的 VPCI 是可用的, 则后侧选择它。在 i)中, 后侧选择 VPCI 中的任意一个可用的 VCI。在 ii)中, 若所指示的 VCI 在 VPCI 中是可用的, 则后侧选择它。

在 iii)中, 后侧选择任何可用的 VPCI 和 VCI。

选择的 VCI 值在后侧响应(SETUP 消息)的第一个消息(即: CALL PROCEEDING 消息)的连接标识信息单元中指示。VP 随路信令字段编码为“VPCI 的明确指示”。优选/指定的字段编码为“指定的 VPCI; 指定的 VCI”。

在 i)和 ii)中, 若所规定的 VPCI 不可用, 后侧发送 RELEASE COMPLETE 消息, 原因为#35(请求的 VPCI/VCI 不可用)。此外, 折回信息单元包括折回原因#35。

在 i)中, 若没有可用的 VCI, 后侧发送 RELEASE COMPLETE 消息, 原因为#45(无可用的 VPCI /VCI)。此外, 折回信息单元包括折回原因#45。

在 ii)中, 若所指示的 VPCI 中的 VCI 不可用, 后侧发送 RELEASE COMPLETE 消息, 原因为#35(请求的 VPCI/VCI 不可用)。此外, 折回信息单元包括折回原因#35。

在 iii)中, 若在任何 VPCI 中, 后侧不能分配一个 VCI, 则后侧发送 RELEASE COMPLETE 消息, 原因为#45(无可用的 VPCI/VCI)。此外, 折回信息单元包括折回原因#45。

当一个接口的两侧同时传送指示同一 VPCI /VCI 的 SETUP 消息时, 可能发生呼叫冲突。对于 PNNI 接口, 为避免呼叫冲突, 有较高节点标识侧(见 6.5.3.3 节)应分配连接标识(VPCI ,VCI)值。有较高节点标识的前侧应在 SETUP 消息中包括连接标识为选项 ii)的连接标识信息单元。有较低节点标识的前侧应在 SETUP 消息中用选项 i)或 iii)。

b) 交换虚通道的分配

当请求建立一个交换虚通道(SVP)(即 SETUP 消息中的宽带承载能力信息单元的承载类别字段指示“VP 业务”)时, 前侧应指示下列之一:

i) 无指示(即 SETUP 消息中不包括连接标识信息单元);

ii) 指定的 VPCI; 无 VCI。

在 i)中, 后侧选择任何可用的 VPCI 和 VCI。

在 ii)中, 若所指示的 VPCI 是可用的, 则后侧选择它。选择的 VPCI 值在后侧响应(SETUP 消息)的第一个消息(即 CALL PROCEEDING 消息)的连接标识信息单元中指示。VP 随路信令字段编码为“VPCI 的明确指示”。优选/指定的字段编码为“指定的 VPCI; 无 VCI”。

在 i)中, 若后侧不能分配一个 VPCI, 则发送 RELEASE COMPLETE 消息, 原因为#45(无可用的 VPCI/VCI)。此外, 折回信息单元包括折回原因#45。

在 ii)中, 若所指示的 VPCI 不可用, 后侧发送 RELEASE COMPLETE 消息, 原因为#35(请求的 VPCI /VCI 不可用)。此外, 折回信息单元包括折回原因#35。

当一个接口的两侧同时传送指示同一 VPCI /VCI 的 SETUP 消息时, 可发生呼叫冲突。对于 PNNI

接口，为避免呼叫冲突，有较高节点标识侧（见 5.3.3 节）应分配连接标识(VPCI)值。有较高节点标识的前侧应在 SETUP 消息中包括连接标识为选项 ii)的连接标识信息单元。有较低节点标识的前侧应在 SETUP 消息中用选项 i)。

3) VPCI 的使用。

信令消息中的连接标识信息单元是用来标识相应的用户信息流。连接标识信息单元包括 VPCI 和 VCI。用 VPCI 而不是 VPI 的原因是有可能使用 VP 交叉连接并且信令虚通路可能控制多个接口。

前侧和后侧都应了解信令协议中用的 VPCI 和用户信息流中用的 VPI 之间的关系。VPCI 仅对一个给定的信令虚通路有意义。

对于这个实施约定，每一个非随路信令虚通路仅控制 PNNI 的单个接口，并且 VPCI 和 VCI 有相同的数值。

4) VPCI 和 VCI 的范围。

有效 VCI 值的范围指示如下：

0~31：不用于即时用户平面连接；

32~65535：VC 的标识（注）。

注：该范围中的一些值暂不能使用（例：有一些可能用于永久连接）。VCI 值的上限（即：65535）将可能被激活的 VCI 比特的数目限制。此外，可能需要保留多于 32 个 VCI 值。

有效 VPCI 值的范围指示如下：

0~4095：VP 的标识（注）。

注：该范围中的一些值暂不能使用（例：有一些可能用于永久虚通道连接）。VPCI 值的上限（即 4095）将可能被激活的 VPI 比特的数目限制。

6.5.2.3 业务类别、业务量参数和 QoS 的选择过程

1) 业务类别的决定。

见 UNI 4.0 信令规范的附件 9 的 A9.2 节。

此外，若网络节点判定请求的业务类别不可用，则应启动符合 UNI 4.0 信令规范的 6.5.3.3 和附件 B 的折回。采用下列原因和折回原因码之一：

#57：承载能力未授权；

#58：承载能力目前不可用；

#59：承载业务未实现。

2) 业务类别、业务量参数和 QoS 的允许组合。

UNI 4.0 信令规范的附件 9 的 A9.3 节用下述例外。

若后侧检查出宽带承载能力、ATM 业务量描述语、端到端转接时延和扩展的 QoS 参数信息单元包括一系列不支持的参数，后侧应返回 RELEASE COMPLETE 消息，原因为#73(不支持的业务量参数组合)。此外，折回信息单元包括折回原因#73。

3) 业务量参数的选择过程。

若后侧不能提供所指示的 ATM 业务量参数，则应折回呼叫并返回 RELEASE COMPLETE 消息，原因和折回原因为#37(用户信元速率不可用)。

若收到的建立指示规定 Tf 字段编码为“已请求标注”，则前侧进行如下动作：

若这个节点对这个连接不执行标注，则可在发送的 SETUP 消息中将 Tf 字段编码为“已请求标注”。

若这个节点对这个连接执行标注，则应在发送的 SETUP 消息中规定一个“不允许标注”的 Tf 指示。

若收到的建立指示规定了“不允许标注”的 Tf 指示，则前侧应在发送的 SETUP 消息中规定一个“不允许标注”的 Tf 指示。

若收到的建立指示规定了“不支持标注”的 Tb 指示，则前侧进行如下动作：

若这个节点对这个连接不执行标注，则应在发送的 SETUP 消息中规定一个“不支持标注”的 Tb 指示。

若这个节点对这个连接执行标注, 则可在发送的 SETUP 消息中将 Tb 字段编码为“支持标注”。

若收到的建立指示规定 Tb 字段编码为“支持标注”, 则前侧可发送 Tb 字段编码为“支持标注”的 SETUP 消息。这指示给后侧对后向业务量前侧有支持标注的能力并且要接受未标注的业务量。

当来话呼叫请求包含这个指示时, 后侧可在 SETUP 消息中包括编码为“已请求标注”Tf 字段。当 ATM 业务量描述语信息单元中有 CLP=0 的业务量参数时, 后侧可在 SETUP 消息中包括编码为“支持标注”Tb 字段。

当收到带有帧丢弃指示的 SETUP 消息时, PNNI 节点遵从 UNI 4.0 信令规范的 2.2.1 节描述的过程。

4) 呼叫/连接建立期间业务量参数的协商过程。

当 SETUP 消息中最小可接受的 ATM 业务量描述语和选择的 ATM 业务量描述语信息单元同时出现时, 网络节点应按 6.5.3 节规定的原因#73(不支持的业务量参数组合)(无折回)拒绝连接建立请求。若选择的 ATM 业务量描述语信息单元或最小可接受的 ATM 业务量描述语信息单元的参数不是按照 6.4.5.7 节和 6.4.5.26 节中分别规定的所允许的组合, 网络应把它们当作 Q.2931 中的 5.6.8 节规定的有内容差错的非必备信息单元。

当 SETUP 消息中有最小可接受的 ATM 业务量描述语并且网络节点能提供 ATM 业务量描述语信息单元中规定的业务量参数值时, 网络节点应用 ATM 业务量描述语信息单元和最小可接受的 ATM 业务量描述语信息单元进展连接建立请求。

当 SETUP 消息中有最小可接受的 ATM 业务量描述语, 但网络节点不能提供 ATM 业务量描述语信息单元中指示的一些信元速率, 而能提供至少对应最小可接受的 ATM 业务量描述语信息单元中的信元速率, 网络节点应在将信元速率调整减少至在最小可接受的 ATM 业务量描述语信息单元中规定的值之上后, 进展连接建立请求。若最小可接受的 ATM 业务量描述语信息单元中的一些参数仍小于修改的 ATM 业务量描述语信息单元中的相应参数, 呼叫应用最小可接受的 ATM 业务量描述语信息单元进行, 其中仅包括除修改的 ATM 业务量描述语信息单元中的参数。否则, 呼叫应用修改的 ATM 业务量描述语信息单元而不用最小可接受的 ATM 业务量描述语信息单元进行。

当 SETUP 消息中有选择的 ATM 业务量描述语信息单元并且网络节点能提供 ATM 业务量描述语信息单元和选择的 ATM 业务量描述语信息单元中规定的业务量参数值时, 网络节点应用 ATM 业务量描述语信息单元和选择的 ATM 业务量描述语信息单元进行连接建立请求。

当 SETUP 消息中有选择的 ATM 业务量描述语信息单元并且网络节点能提供 ATM 业务量描述语信息单元中规定的业务量参数值但不能提供选择的 ATM 业务量描述语信息单元中规定的业务量参数值时, 网络节点应用 ATM 业务量描述语信息单元而不用选择的 ATM 业务量描述语信息单元进展连接建立请求。

当 SETUP 消息中有选择的 ATM 业务量描述语信息单元并且网络节点不能提供 ATM 业务量描述语信息单元中规定的业务量参数值但能提供选择的 ATM 业务量描述语信息单元中规定的业务量参数值时, 网络节点应按 ATM 业务量描述语用选择的 ATM 业务量描述语信息单元进展连接建立请求, 并且不向前传送选择的 ATM 业务量描述语信息单元。

若网络节点不支持 ATM 业务量描述语信息单元中规定的业务量参数值, 也不支持选择的 ATM 业务量描述语信息单元或最小可接受的 ATM 业务量描述语信息单元中规定的业务量参数值时, 网络节点应按 6.5.3 节规定的原因#37(用户信元速率不可用)拒绝连接建立请求。此外, 折回信息单元包括折回原因#37。

5) QoS 参数的选择过程。

QoS 要求是在 PNNI 网内通过个别的服务质量参数表示。服务质量参数是包括在扩展的 QoS 参数信息单元和/或端到端转接时延信息单元中的。SETUP 消息中 QoS 参数的允许集合由呼叫的 ATM 业务类别决定。

对于每个 QoS 参数, 若已包括参数的可接受值并且参数的端到端值是由累计决定, 则应包括相应参数的累计值。SETUP 消息中发送的 QoS 参数的前向和后向的累计值是沿着呼叫路由按序修正以决定呼

叫所期望的值。本节规定的过程支持 ATM 论坛业务量管理规范 4.0 中描述的时延参数的累计的（即加法的）简单方法。

当前侧收到包括 QoS 参数信息单元的建立指示时，前侧应在相应的 SETUP 消息中包括 QoS 参数信息单元。

当前侧收到来自前一接口(不是 PNNI 接口)的建立指示时，呼叫的 ATM 业务类别是 CBR、实时 VBR 或非实时 VBR，并且扩展的 QoS 参数信息单元是包括在收到的 SETUP 消息中，前侧应用 QoS 参数信息单元中的业务类别和前向与后向 QoS 级别字段的本地映射产生扩展的 QoS 参数信息单元。此外，若收到的建立指示中未包括端到端转接时延信息单元，则作为上述映射的一部分产生。当使用上述映射时，所有 QoS 参数的隐含值（自包括在 QoS 参数信息单元中的 QoS 级别和呼叫的 ATM 业务类别）必须被规定，并且每个包括一个或多个新产生的 QoS 参数的信息单元的起源字段应标记为“中间网”（即在扩展的 QoS 参数信息单元中起源字段是置为“中间网”，端到端转接时延信息单元中包括“网络—产生 指示语”）。自映射产生的累计参数值应在开始处理单个 QoS 参数前初始化为 0。

对于包括在扩展的 QoS 参数信息单元和/或端到端转接时延信息单元中的每个参数，前侧应采取如下行动（不论信息单元是包括在收到的 SETUP 消息中还是用 QoS 级别的本地映射产生）：

——若前一接口不是 PNNI 接口并且参数是累计的。

a) 前侧应增加参数的累计前向值来计算通过从网络边界到这个交换系统的前一链路进行用户数据传送而预期地增长。

b) 此外，前侧应增加参数的累计后向值来计算通过从这个交换系统到网络边界的前一链路进行用户数据传送而预期地增长，以及计算这个交换系统内部由于用户数据传送而预期地增长。

注：在这个步骤中描述的过程不必与这个 PNNI 接口相联系，它们也可被认为是前一接口（非 PNNI）的后侧所承担的过程。在这里提供的原因是为了完成 UNI 4.0 信令规范中要求支持网络侧过程的规范。

——若参数是累计的，前侧应增加参数的累计前向值来计算在这个交换系统内通过从这个交换系统到后侧的交换系统的链路进行用户数据传送而预期地增长。

——前侧应决定是否能支持参数的最高/最低可接受值。若不支持参数的最高/最低可接受值（包括小于/大于或等于），则前侧应按附录 B 8.3.1.2 节中规定的折回过程，并用折回原因#49(服务质量不可用)。

若对相应的 ATM 业务类别（在扩展的 QoS 参数信息单元或端到端转接时延信息单元中）未规定单个允许的 QoS 参数的可接受的前向值，则缺省值为单个 QoS 参数的任意值，并且前侧应继续处理呼叫。

若前侧能提供所有规定的单个 QoS 参数的可接受值，前侧应将呼叫向后侧进展。

后侧不使用 QoS 参数信息单元，但若它出现且若呼叫进展，则传递它。

对于包括在扩展的 QoS 参数信息单元和/或端到端转接时延信息单元中的每一个参数，后侧应进行如下动作：

——若参数是累计的，前侧应增加参数的累计后向值来计算在这个交换系统内和通过从这个交换系统到前侧的交换系统的链路进行用户数据传送而预期地增长。

——若呼叫选路的下一接口不是 PNNI 接口并且参数是累计的则进行如下动作：

a) 后侧网络节点应增加参数的累计前向值来计算在这个交换系统内和通过这个交换系统到网络边界的链路进行用户数据传送而预期地增长。

b) 此外，应增加参数的累计后向值来计算通过网络边界到这个交换系统的链路进行用户数据传送而预期地增长。

注：在这个步骤中描述的过程不必与这个 PNNI 接口相联系，它们也可被认为是下一接口（非 PNNI）的前侧所承担的过程。在这里提供的原因是为了完成 UNI 4.0 信令规范中要求支持网络侧过程的规范。

上述执行累计过程的结果是网络决定从网络边界到网络边界的累计前向和后向参数。注意这还用在呼叫路径未穿过任何 PNNI 接口的情况。

应决定是否能支持参数的最高/最低可接受值。若不支持参数的最高/最低可接受值（包括小于/大于

或等于)，则后侧应按附录 B 8.3 节中规定的折回过程，并返回一个带原因的 RELEASE 或 RELEASE COMPLETE 消息（取决 CALL PROCEEDING 消息是否已发送），若可用，折回原因为#49(服务质量不可用)。

若对相应的 ATM 业务类别（在扩展的 QoS 参数信息单元或端到端转接时延信息单元中）未规定单个允许的 QoS 参数的可接受的前向值，则缺省值为单个 QoS 参数的任意值，并且后侧应继续处理呼叫。

若后侧能提供所有规定的单个 QoS 参数的可接受值，后侧应继续进展呼叫。

6) ABR 连接的业务量参数选择过程。

在 ABR 连接情况下，下列附加和修改的过程用于业务量参数的处理。

ATM 论坛 UNI 4.0 信令规范中定义的 ABR 参数的缺省是在主叫用户 UNI 的网络侧执行。ABR 建立参数在 SETUP 消息中都是必备的（注）。在软 PVPC 或 PVCC 建立时，起源交换系统负责发送包括所有出现参数的 SETUP 消息。

注：当主叫用户请求 ABR 连接时

协商过程允许每个交换系统按需调整呼叫参数以保护交换系统中的资源和对其他连接的服务质量保证。上述资源不必与一个特别的 PNNI 相联系；它们可与 UNI 相关或为几个接口共用。

ATM 业务量描述语和 ABR 建立参数信息单元中的参数协商如下。

对于给定方向的 PCR、ICR、TBE、RIF 和 RDF 的参数值可由任一侧协商。若 SETUP 消息中的最小可接受 ATM 业务量描述语信息单元中包括 MCR 参数，则 MCR 用 6.5.2.3 节的 4) 条的过程协商。

若能提供指示的 PCR 和 ABR 建立参数值，PNNI 网应用起源参数继续呼叫。

若不能提供指示的 PCR 但至少能提供所协商的 MCR 值，PNNI 网应在调整 PCR 值后继续呼叫。调整的 PCR 值将大于或等于 MCR 值。

当进展呼叫时，网络若有必要也应调整前向和/或后向参数、ABR 建立参数、ABR 初始信元速率、ABR 瞬时缓冲器、速率增加和减少因素。

表 86 归纳了可能做的修改。

表 86

给定方向的参数	网络的修改
PCR	仅减少
ICR	仅减少
TBE	仅减少
RIF	仅减少
RDF	仅增加

参数协商保持下列不等式：

$$MCR \leq ICR \leq PCR$$

若网络不能提供等于 MCR 的峰值信元速率，则应按附录 B 的折回过程，原因和折回原因为#37(用户信元速率不可用)。

ATM 附加参数信息单元中的参数协商如下。

ATM 附加参数中的参数值可由任一侧协商，但仅当参数是出现在 SETUP 消息中时（即：由主叫用户提供）。若参数缺席，则用缺省值并且在这种情况下不可能有参数协商。若 SETUP 消息中不包括 ATM 附加参数信息单元，则缺省值可用于上述所有参数并且无协商。附加参数的缺省值在业务量管理规范 4.0 中规定。

ATM 论坛业务量管理规范 4.0 提供了有关协商的更详细考虑。表 87 归纳了可能做的修改

表 87

给定方向的参数	网络的修改
Nrm	无协商（注）
Trm	无协商（注）
CDF	仅增加
ADTF	仅减少
注：若网络不支持所指示的值，则可通过改变表示语出现比特将其变为缺省。	

参数是否能协商取决于它的编码所规定的固定的范围。

7) ABR 连接的累计 RM 固定往返时间参数的处理。

后侧应调整 ABR 建立参数信息单元中的累计 RM 固定往返时间参数。调整的量是 PNNI 上 RM 信元时延的前向和反向的固定部分的总和。它包括前向和反向链路广播时延和任何在 PNNI 和交换系统内的固定处理时延。以微秒表示且编码为整数的调整值是加到 RM 固定往返时间参数中。

在呼叫路径中的第一个和最后一个交换系统中，UNI 的网络侧应调整 ABR 建立参数信息单元中的累计 RM 固定往返时间参数。调整的量是 UNI 上 RM 信元时延的前向和反向的固定部分的总和。它包括至网络边界前向和反向链路广播时延和任何在 UNI 和交换系统内的固定处理时延。

上述规则的结果是：每一个交换系统的累计 RM 固定往返时间参数是由前一跳（无论这跳是 UNI 还是 PNNI）的往返时间来调整的。最后的交换系统例外，它包括调整的前一跳和下一跳（至被叫用户）。

注：累计 RM 固定往返时间参数仅在 SETUP 消息而不在 CONNECT 消息处理中调整。

6.5.2.4 呼叫/连接进展

后侧应向前侧发送一个 CALL PROCEEDING 消息来响应 SETUP 消息并指明呼叫被处理并进入呼叫进展发送状态。当前侧收到 CALL PROCEEDING 消息，则前侧应停止定时器 T303，启动 T310 并进入呼叫进展接收状态。

注：CALL PROCEEDING 消息的发送是必备的。

若收到 SETUP 消息后，后侧判断由于某种理由不能支持呼叫，后侧应启动 6.5.3 节定义的呼叫清除。可用到的一些原因在 6.5.2.10 中给出。

若前侧已收到 CALL PROCEEDING 消息但在停止定时器 T310 超时前未收到 CONNECT、ALERTING 或 RELEASE 消息，则前侧应向起源接口启动清除过程（无折回）和向被叫清除，原因值为#102(定时器超时时恢复)。

6.5.2.5 呼叫/连接提醒

收到被叫方提醒的指示后，后侧应向前侧发送 ALERTING 消息并进入提醒递交状态。

当收到来自后侧的 ALERTING 消息时，前侧应停止定时器 T310，启动 T301，进入提醒接收状态并向主叫用户发送一个提醒指示。

6.5.2.6 呼叫/连接接受

当从呼叫控制收到呼叫已被接受的指示时，后侧应向前侧发送 CONNECT 消息并进入运行状态。

这个消息指示给前侧：从接口至被叫用户的连接已建立。

收到 CONNECT 消息后，前侧应停止定时器 T310 或 T301 并进入运行状态。

1) 呼叫/连接接受期间业务量参数的选择过程。

若收到的 CONNECT 消息中包括 ATM 业务量描述语信息单元（带有可能的 Tf 和 Tb 字段的例外情况），后侧应立即向前侧传送它。

若 SETUP 消息包括 ATM 业务量描述语信息单元且 Tf 参数编码为“已请求标注”，则后侧交换机应在 CONNECT 消息中指示。

“已使用标注”，若这个节点对这个连接执行标注，或与来话连接指示相同的标注值，若这个节点对

这个连接不执行标注。

若 SETUP 消息包括 ATM 业务量描述语信息单元且 Tf 参数编码为“不允许标注”，则后侧交换机应在 CONNECT 消息中指示“不使用标注”。

如果前侧发送包括 ATM 业务量描述语信息单元且 Tf 字段编码为“已请求标注”的 SETUP 消息，且前侧收到包括 ATM 业务量描述语信息单元且 Tf 字段指示“不使用标注”的 CONNECT 消息。则为前向业务量连接分配的资源可被修改，因为前向业务量必须符合 CLP=0 业务量参数和 CLP=0+1 业务量参数。

若 SETUP 消息包括 ATM 业务量描述语信息单元且 Tb 参数编码为“支持标注”，则后侧交换机应在 CONNECT 消息中指示“不允许标注”，若这个节点对这个连接执行标注，或与来话连接指示相同的标注值，若这个节点对这个连接不执行标注。

如果前侧发送包括 ATM 业务量描述语信息单元且 Tb 字段编码为“支持标注”的 SETUP 消息，且前侧收到包括 ATM 业务量描述语信息单元且 Tb 字段指示“不允许标注”的 CONNECT 消息。

则为后向业务量连接分配的资源可被修改，因为后向业务量必须符合 CLP=0 业务量参数和 CLP=0+1 业务量参数。

当收到一个带有帧丢弃指示的 CONNECT 消息，PNNI 节点应按照 UNI 4.0 信令规范 2.2.1 节中的过程。

2) 呼叫/连接接受期间业务量参数的协商过程。

若对方在 CONNECT 消息中返回 ATM 业务量描述语信息单元，网络节点应向向前传送它。若 CONNECT 消息中的业务量参数值与已在 SETUP 消息中传送的值不同，已给连接分配的资源应被修改。

若对方在 CONNECT 消息中未返回 ATM 业务量描述语信息单元且当网络节点处理相应的 SETUP 消息时已协商业务量参数，节点应在向前传送它之前，在 CONNECT 消息中包括 ATM 业务量描述语信息单元。

ATM 业务量描述语的内容应与前转的 SETUP 消息中的一样，除了按 6.5.2.6 节中 2) 条规定编码的业务量管理任选。

若对方在 CONNECT 消息中未返回 ATM 业务量描述语信息单元且当网络节点处理相应的 SETUP 消息时未协商业务量参数，节点应在 CONNECT 消息中可选择地包括 ATM 业务量描述语信息单元。ATM 业务量描述语的内容应与前转的 SETUP 消息中的一样，除了按 6.5.2.6 节中 2) 的规定编码的业务量管理任选。

6.5.2.7 呼叫建立失败

除了 6.5.2.3 节中描述的过程，下列原因应使用。若后侧判断所请求的业务不可用或不能处理呼叫，则后侧应按 6.5.3.3 节和附录 B 启动折回。原因和折回原因码为下列之一。

- #38 网络无序；
- #41 暂时失败；
- #58 承载能力目前不可用；
- #63 业务或选择不可用，未规定；
- #65 承载能力未实现。

...

6.5.2.8 通用标识传送过程

见 UNI 4.0 信令规范 2.2.2 节。

6.5.3 呼叫/连接清除

对于折回，呼叫/连接清除过程的附加过程在附录 B 中描述。

6.5.3.1 术语

见 ITU-T 建议 Q.2931 的 5.4.1 节。

6.5.3.2 例外情况

在一般情况下，呼叫清除应在 UNI 启动。在 PNNI，呼叫清除可由 PNNI 的任一侧启动。它可作为

UNI 启动的呼叫清除的响应或由于故障、管理动作或其他例外条件启动呼叫清除。呼叫清除过程在 6.5.3.3 节中定义。上述原则的例外情况如下：

后侧可在收到一个 SETUP 消息后通过响应一个 RELEASE COMPLETE 消息拒绝一个呼叫/连接；释放呼叫参考并进入空状态。

6.5.3.3 清除

除了 6.5.3.2 节中说明的例外情况，清除过程是对称的且可由 PNNI 的前侧或后侧启动。下列过程仅描述前侧启动清除的情况。

前侧启动清除通过发送一个 RELEASE 消息，启动定时器 T308；释放虚通路并进入释放请求状态。

PNNI 接口的后侧在收到 RELEASE 消息后应进入释放指示状态。收到这个消息后，后侧应释放虚通路并启动由通知 PNNI 呼叫控制实体来清除网络连接的过程。一旦用于呼叫的虚通路已被释放，后侧应作如下操作：

向前侧发送一个 RELEASE COMPLETE 消息；释放呼叫参考和虚通路（即连接标识）；进入空状态。

注：RELEASE COMPLETE 消息仅为本地并且不隐含端到端清除的应答。后侧在收到 RELEASE COMPLETE 消息后应：停止定时器 T308；释放虚通路；释放呼叫参考并返回空状态。

若定时器 T308 第一次超时，前侧应：向后侧重发 RELEASE 消息并带有第一个 RELEASE 消息中包含的原因值；再启动定时器 T308 并保留在释放请求状态。此外，前侧应指示原因为#102“定时器超时恢复”的第二个原因信息单元。如果定时器 T308 第二次超时前未收到来自后侧的 RELEASE COMPLETE 消息，前侧应：释放呼叫参考并返回空状态。附加的恢复过程(如启动再启动)是取决于实施。

6.5.3.4 清除冲突

当两侧同时发送与同一呼叫参考值相关的 RELEASE 消息时，清除冲突发生。若前侧或后侧在释放请求状态时收到一个 RELEASE 消息，则接收实体应：停止定时器 T308；释放呼叫参考和虚通路；进入空状态（不发送或接收 RELEASE COMPLETE 消息）。

6.5.4 呼叫/连接冲突

当 PNNI 接口两侧同时发送指示一同超过接口剩余资源的业务量参数的 SETUP 消息时，呼叫/连接冲突发生。PNNI 接口两侧应按 6.5.3 节带折回信息单元的过程清除呼叫/连接。原因和折回原因应如下：

#47：资源不可用，未规定；

#49：服务质量不可用；

#51：用户信元速率不可用。

6.5.5 再启动过程

UNI 4.0 信令规范的 5.5 节应做如下修改。

第一段以下列文字替代：PNNI 接口两侧应实施这些过程。

6.5.5.1 发送 RESTART

UNI 4.0 信令规范的 5.5.1 节应做如下修改。

第一段的第一句以下列文字替代：为将虚通路返回至空闲条件，PNNI 任一侧发送 RESTART 消息。

6.5.5.2 接收 RESTART

见 UNI 4.0 信令规范的 5.5.2 节。

6.5.5.3 再启动冲突

当 PNNI 接口两侧的信令实体同时发送一个 RESTART 消息时，发生再启动冲突。全局呼叫参考的呼叫参考标志用于再启动过程。当接口两侧同时启动再启动请求时，它们应被各自独立地处理。当规定同样的虚通路时，只有在相关再启动过程完成时它们才可被释放。

6.5.6 差错条件的处理

Q.2931 建议的 5.6 节应做如下修改：

第一段中，“Q.2931 用户—网络控制消息”改为“PNNI 信令消息”。

第一段和第三段的参考“5.6.1”和“5.6.8”节分别改为“6.5.6.1”和“6.5.6.8”。

6.5.6.1 协议鉴别差错

当收到带有协议鉴别符编码不为“PNNI 信令消息”的消息时，该消息应被忽略。

6.5.6.2 消息太短

见 Q.2931 建议的 5.6.2 节。

6.5.6.3 呼叫参考差错

1) 无效呼叫参考格式。

见 Q.2931 建议的 5.6.3.1 节。

2) 呼叫参考过程差错。

Q.2931 建议的 5.6.3.2 节应做如下修改。

项目“f”和“g”参考“5.6.12”和“5.6.11”节分别改为参考“6.5.6.12”和“6.5.6.11”。

6.5.6.4 消息类型或消息顺序差错

Q.2931 建议的 5.6.4 节应做如下修改。

这节的差错过程仅用在当消息兼容性指令表示语的标志置为“消息指令字段无意义”时。若它置为“按照明确指令”，则优先采用 6.5.7 节中的过程。

每当在除空状态的任何状态下收到非期望的消息（包括 ITU-T 已标准化但在这个规范中未包括的消息），除了 RELEASE、RELEASE COMPLETE 或不认识的消息，应返回带下列原因之一的 STATUS 消息。

#97: “消息类型不存在或未实现”；

#101: “消息与呼叫状态不兼容”。

这个过程有两个例外。第一个例外是当 PNNI 的前侧或后侧收到一个响应 SETUP 消息的非期望的 RELEASE 消息时，不发送 STATUS 或 STATUS ENQUIRY 消息。每当前侧或后侧收到一个非期望的 RELEASE 消息时，接收实体应：释放虚通路并清除连接；向另一侧返回 RELEASE COMPLETE 消息；释放呼叫参考；停止所有定时器；进入空状态并通知 PNNI 呼叫控制实体。

第二个例外是当 PNNI 的前侧或后侧收到一个非期望的 RELEASE COMPLETE 消息时，接收实体应：释放虚通路并清除连接；停止所有定时器；进入空状态并通知 PNNI 呼叫控制实体。

6.5.6.5 消息长度差错

Q.2931 建议的 5.6.5 节应做如下修改。

参考“5.6.6”节改为“6.5.6.6”。

6.5.6.6 一般消息差错

1) 信息单元顺序。

Q.2931 建议的 5.6.6.1 节应做如下修改。

参考“4.5.1”节改为“6.4.5.1”。

2) 重复的信息单元。

见 Q.2931 建议的 5.6.6.2 节。

3) 编码标准差错。

Q.2931 建议的 5.6.6.3 节应做如下修改。

参考“5.6.7.2”节改为“6.5.6.7.2”。

参考“5.6.8.2”节改为“6.5.6.8.2”。

6.5.6.7 必备信息单元差错

1) 必备信息单元丢失。

Q.2931 建议的 5.6.7.1 节应做如下修改。

第三段中参考“5.4”节改为“6.5.3”。

2) 必备信息单元内容差错。

Q.2931 建议的 5.6.7.1 节应做如下修改。

第一段中参考“5.7”节改为“6.5.7”。

第四段中参考“5.4”节改为“6.5.3”。

这节结尾处的注不用。

6.5.6.8 非必备信息单元差错

这节的差错过程仅用在当指令字段的标志（比特 5）置为“IE 指令字段无意义”时。若它置为“按照明确指令”，则优先采用 6.5.7 节中的过程。

1) 不认识的信息单元。

Q.2931 建议的 5.6.8.1 节应做如下修改。

第三段的第一句改为：若一个清除消息包括一个或多个不认识的信息单元，则使用下列过程。

在节的末尾加下列内容：

当在一个消息中有一个宽带锁定转换信息单元时，宽带锁定转换信息单元和所有后续的信息单元可当做不认识的信息单元处理。当一个原因信息单元的产生是为了报告这些转换信息单元的差错时，原因信息单元（若有）的诊断字段应仅包括宽带锁定转换信息单元的信息单元标识以及任何差错的非转换信息单元。

当在一个消息中有一个宽带非锁定转换信息单元，宽带非锁定转换信息单元和后续信息单元可当做不认识的信息单元处理。当一个原因信息单元的产生是为了报告这些转换信息单元的差错时，原因信息单元（若有）的诊断字段应仅包括宽带非锁定转换信息单元的信息单元标识以及任何差错的非转换信息单元。

2) 非必备信息单元内容差错。

Q.2931 建议的 5.6.8.2 节应做如下修改：第二段中的参考节“3”改为“6.3”。

3) 非期望的识别的信息单元。

Q.2931 建议的 5.6.8.3 节应做如下修改：参考节“5.6.8.1”改为“6.5.6.8.1”。

6.5.6.9 信令 AAL 复位

Q.2931 建议的 5.6.9 节应改为：

每当通过 AALESTABLISHINDICATION 原语收到来自 SAAL 层的信令 AAL 复位指示时，用下列过程：

- 1) 在清除阶段（状态 NN11 和 NN12）的呼叫不采取任何动作。
- 2) 在建立阶段（状态 NN1、NN3、NN6 和 NN9）的呼叫应保持。作为任选，状态询问过程可调用。
- 3) 在激活状态的呼叫应按 6.5 节的其他部分中的过程保持。

6.5.6.10 信令 AAL 故障

Q.2931 建议的 5.6.10 节应做如下修改：

第二列表的第二项中的参考节“5.6.11”改为“6.5.6.11”。

第四段的正文改为：

若在信令 AAL 重新建立之前定时器 T309 超时，受影响的实体应：释放虚通路并清除连接；释放呼叫参考；进入空状态并通知 PNNI 呼叫控制实体信令 AAL 重新建立故障。

这节的最后一段不用。

6.5.6.11 状态询问过程

Q.2931 建议的 5.6.11 节应做如下修改。

第一段的改为：

为了检查 PNNI 的前侧或后侧的呼叫状态的正确性，可发送请求呼叫状态的 STATUS ENQUIRY 消息。这可用于 6.5.6.9 和 6.5.6.10 节中描述的过程差错条件。

这节的最后一段改为：

若定时器 T322 超时且未收到 STATUS 消息，STATUS ENQUIRY 消息可被重发一次或多次直到收到

响应。STATUS ENQUIRY 消息的重发次数取决于实施。若他的重发次数超过限制, 呼叫应被清除。当清除呼叫时应使用的原因#41(暂时故障)。PNNI 呼叫控制实体应被通知呼叫故障。

6.5.6.12 接收一个 STATUS 消息

Q.2931 建议的 5.6.12 节应做如下修改。

下句应加在第二列表项目 c) 的结尾:

PNNI 呼叫控制实体应被通知呼叫故障。

6.5.7 带明确动作表示语的差错过程

6.5.7.1 非期望的或认识的消息类型

Q.2931 建议的 5.7.1 节应做如下修改。

节的第二段中的参考节“5.4.3 或 5.4.4”改为“6.5.3”。

在节的开始加如下一段:

本节的差错过程仅当指令字段的“传递请求”(比特 4)置为“无传递请求”时用于不认识的消息。若它在不认识的消息中置为“传递请求”, 则消息应被无差错检查地传递, 提供的下一个接口是 PNNI。

6.5.7.2 信息单元差错

Q.2931 建议的 5.7.2 节应做如下修改。

在节的开始加如下一段:

本节的差错过程仅当指令字段的“传递请求”(比特 4)置为“无传递请求”时用于不认识的消息。若它在认识的信息单元中置为“传递请求”, 则消息中的信息单元应被无差错检查地传递, 提供的下一个接口是 PNNI。

6.5.8 带有不全信息消息的处理

Q.2931 建议的 5.8 节应做如下修改。

参考节“5.6.7.1”和“5.7.1”分别改为“6.5.6.7.1”和“6.5.7”。

6.5.9 网络节点呼叫控制要求

可能参加到另一路由的节点必须保留起始收到的 SETUP 消息内容的备份, 包括 QoS 和用于决定路由和呼叫允许控制以及它产生的 DTL 的其他参数。备份也应保留在折回过程中由节点收到的任何被阻塞的节点或链路子字段内容。起始 SETUP 消息内容、附加的 DTL 和可能被阻塞的转移备份应被保留直到从下一节点收到 CONNECT 或 ALERTING 消息, 或直到呼叫被清除。

6.5.10 通知过程

见 Q.2931 建议的 5.9 节。

6.5.11 互通通知

若呼叫控制消息中包括进展表示语信息单元, 则用 6.5 节的过程。若进展消息中包括进展表示语信息单元, 则不发生状态变化, 但若进展描述是 No.1、No.2 或任选的 No.4, 则应停止除 T301 和 T322 外的监视定时器。

6.5.12 定时器列表

基本呼叫的定时器描述将遵从 Q.2931 建议的表 7-1 定时器定义。这个表中的值应用 T310 值(应为 30~120s)的例外, 详见 6.5 节。SETUP 消息的重发是任选。

6.6 点一点呼叫/连接控制过程

本节描述点到多点呼叫过程并应用 ITU-T Q.2971 建议。它支持点一点呼叫当用户平面信息是从一个主叫用户到一组被叫用户时。网络节点将指示在 PNNI 接口一个增加一方用户请求的到达。它是通过穿过接口传送一个 SETUP(见 6.6.1 节)或 ADD PARTY(见 6.6.2 节)来完成。点一点过程是基于 ITU-T Q.2971 建议的第 10 节。

6.6.1 起始方用户的建立

ITU-T Q.2971 建议的 10.2.1 节的过程应做如下修改。

因为过程用在 PNNI 接口的前侧和后侧之间, 将全文中的下列项目修改如下:

“用户—网络接口”改为“PNNI 接口”。

“网络”改为“后侧网络节点”。

“用户”改为“前侧网络节点”。

第 2 段不用。段落表示如下：

若一个 CALL PROCEEDING、ALERTING 或 CONNECT 消息……9.2 子项的过程（代替 10.2.2 子项的过程）。

Q.2971 建议的 9.2 节的修改如下。

a) 第一段改为如下。

PNNI 接口的前侧和后侧应按有下列例外的 6.5 节的过程。

b) 第二段删除句子。

端点参考信息单元中的指令表示语应编码为“丢弃信息单元并进展”。

c) 第四段不用，段落表示如下。

若一个 CALL PROCEEDING、ALERTING 或 CONNECT 消息是第一个消息……应按 9.2.1 子项的过程。

d) 9.2 节的最后两段不用，段落表示如下。

在终结接口，ADD PARTY，……和 DROP PARTY 消息不用。

在终结接口，收到 ADD PARTY，……消息应当作不认识的或非期望的消息处理。

e) 9.2.1 节不用。

6.6.2 增加一方用户

ITU-T Q.2971 建议的 10.2.2 节的过程应做如下修改。

因为过程用在 PNNI 接口的前侧和后侧之间，将全文中的下列项目更换如下：

“用户—网络接口”改为“PNNI 接口”。

“网络”改为“前侧网络节点”。

“用户”改为“后侧网络节点”。

对于 10.2.2.1~10.2.2.5 节，当发送 ADD PARTY REJECT 消息时，它应包括折回原因置为规定原因值的折回信息单元。

在 10.2.2.5 节中，加下列段：

若收到的 ADD PARTY REJECT 消息包括一个折回信息单元，则用附录 B 中包括的折回处理过程。

注：增加一方用户请求在增加一方用户表中，它没有用户状态。

6.6.3 去掉一方用户

ITU-T Q.2971 建议的 10.3 节的过程应做如下修改。

10.3.2 节项 a) “应用 5.4.2 子项/Q.2931 的呼叫清除过程”改为“应用 6.5.3.3 节的呼叫清除过程”。

在 10.3.3、10.3.4、10.3.5 和 10.3.6 节中，“5.4.2 子项/Q.2931 的清除过程”改为“6.5.3 节的清除过程”。

在 10.3.3 节中，在以“当网络收到一个 RELEASE 消息”开始的段落中的第四段后，加另一段：

若收到的 RELEASE 消息包括折回信息单元，则用附录 B 中包括的折回处理过程。

6.6.4 再启动过程

应用 ITU-T Q.2971 建议的 10.4 节的过程。

6.6.5 差错条件的处理

ITU-T Q.2971 建议的 10.5 节的过程应做如下修改。

在 9.5.7.2 节中，“5.4 子项/Q.2931 的清除过程”改为“6.5.3 节的清除过程”。

6.6.6 点到多点的定时器列表

Q.2971 建议的 13.2 节中规定的定时器应用除了应在 34~124s 范围内的 T399 的缺省值的值。

附录 A

(标准的附录)

指定转接序列

该附录用来说明指定转接序列。多级层次结构中，指定转接序列（DTL）用来指示一个连接需要经过一个对等组时的转接情况（例如：逻辑节点或可能的物理链路）。网络拓扑表示的结构性质使得需要引入一个 DTL 堆栈（后入先出表）来表示完整的发端路由。DTL 堆栈作为指定转接序列的信息单元在 SETUP 或 ADD PARTY 消息中发送。

本节规定的伪程序代码包含在 A5 节中。如果叙述文本中存在不明确的地方，则应参照伪程序代码解决。如果文本与伪程序代码出现不一致时，则应以文本为准。

DTL 处理程序在呼叫经过的每个节点上进行，该程序分为两个部分，首先该节点要确定是否有 DTL 从 DTL 堆栈中删除，并按照余下的 DTL 堆栈中的内容进行处理，并准备传送到下一个节点。最典型的情况是一个节点或增加 DTL 到堆栈中或从堆栈中删除 DTL。如果一个节点同时要向 DTL 堆栈中增加并删除 DTL，则必须先要删除其增加的 DTL。

呼叫通路上的每个 PNNI 节点可以根据其对 DTL 处理时的角色进行分类，从生成 DTL 的角度出发，节点可以分为：

- 1) DTL 起源节点；
- 2) 入口边界节点（进入入口边界节点）；
- 3) 既不是 DTL 起源节点，也不是入口边界节点；

从删除 DTL 的角度出发，节点可以分为：

- 1) DTL 终止节点；
- 2) 出口边界节点（离开边界节点）；
- 3) 既不是 DTL 终止节点，也不是出口边界节点。

A1 术语

呼叫进入一个对等组：在多级结构中当一个呼叫通过与其他对等组相连的物理链路，已经进展到对等组的边界节点，则该呼叫知道将要进入一个对等组。

呼叫离开一个对等组：在多级结构中当一个呼叫通过与其他对等组相连的物理链路，将要进展到对等组的边界节点，则该呼叫知道将要离开一个对等组。

A2 初始的 DTL 处理

A2.1 DTL 起源节点的程序

DTL 起源点通过路由字段应当计算出一条至被叫的通路，并在用于建立呼叫/连接的 SETUP 或 ADD PARTY 消息中发送。如果存在转接网选择信息单元，则转接网或任何被叫号码将用来确定通路。选择通路的最后一个节点必须能连通至被叫用户号码或者转接网所通告的成员范围（参见 5.8.1.3 节），该范围至少应为最高通路范围（定义为通路所使用的 PNNI 结构的最高层）。此外如果 SETUP 消息中存在连接范围选择信息单元，则通路范围必须位于呼叫的连接范围内（例如：连接范围 \leq 通路中任何节点的等级指示语 \geq 所通告的可到达的地址范围或转接网）。

当被叫用户号码是一组地址时（例如：该呼叫使用 ATM 的组播能力任播），则通路选择的目标应当是由被叫用户号码标识的 ATM 组的一个成员。在这种情况下，如果 SETUP 消息没有连接范围选择信息单元，则“本地网”作为缺省的连接范围用于通路选择（它并未包含在 SETUP 消息中）。

通路是通过一个 DTL 堆栈来表示的，如果没有发现适当的通路，则连接将被清除，并携带适当的返回原因。如果一个转接网选择信息单元存在，则会使用返回原因 #2(没有路由到规定的转接网)。否

则使用返回原因#3(没有到目的地的路由)。

起源点应了解的通路信息所包含的通路上的逻辑节点(和相关的逻辑链路)信息如下:

- 1) 要经过的起源点所在的对等组内的完整的逻辑节点表(和相关的逻辑链路)。
- 2) 用来确定要到达的包含终止节点的更高级通路的一组逻辑节点表(和相关的逻辑链路)。

DTL 起源节点应当在 SETUP 或 ADD PARTY 消息中添加一个编码为“后入先出”的宽带重复表示语信息单元,该信息单元后再跟随一个或多个 DTL 信息单元。DTL 信息单元将被推入栈中,这样 DTL 就与所经过的节点的顺序相反。例如:第一个出现指定转接序列信息单元就是包含 DTL 终止节点的逻辑节点,最后一个出现指定转接序列信息单元就是包含 DTL 起源节点的逻辑节点。

为了表示将要通过的节点序列,每个指定转接序列信息单元都应包含一个层次结构中单层的转接序列。该序列应当以该连接通过的该级中的第一个逻辑节点开始,以该连接通过的该级中的最后一个逻辑节点或逻辑链路结尾。当通路从层次结构中的一级变化到另一级后,则指定转接序列信息单元将被推入栈中,当前的指针被设置为 0,用来指示指定转接序列中的第一个逻辑节点(或者是逻辑链路)。

为了表示将要通过的节点序列,栈顶的指定转接序列信息单元应当包含 DTL 起源节点所处的低一级的对等组在内的所有逻辑节点序列。该序列以 DTL 起源节点或所在的对等组的逻辑节点,接下来是与之相邻最近的逻辑节点,结尾可以是 DTL 终止节点(如果在 DTL 起源节点的对等组内),也可能是对等组的出口边界节点(DTL 起源节点与 DTL 终止节点不在同一个对等组中)。转接指针设置为 0,用来指示当前起源节点。

A2.2 入口边界节点的程序

当一个呼叫进入一个对等组,边界节点将检查 SETUP 或 ADD PARTY 消息中的第一个指定转接序列信息单元。如果在接收链路上的公共对等组的级别上,当前指针指示的逻辑节点标识符与边界节点的祖先节点 ID 不等,则清除该呼叫并返回原因#41(暂时故障),并折回呼叫时可以携带以下原因:

- 1) 后续端阻塞指示语及返回原因#128(下一节点不可达)。
- 2) 折回原因#160(DTL 转接的不是本节点 ID)且 DTL 转接序列是阻塞的节点。

当退出该对等组时,如果逻辑端口标识符并未设置为 0,将标识该逻辑端口将用于进展呼叫。

边界节点应按照如下方法计算通过对等组的路由。

1) 应检查堆栈顶部的指定转接序列,如果当前的转接指针指向转接序列的末尾,它应执行 A2.2 节 b)条的程序,否则应以当前指针所指示的转接后的内容作为通路计算的目标,执行 A2.2 节的 c)条的程序。

2) 如果到达转接序列的末尾,如果堆栈未空,则应继续检查 DTL 堆栈的下一级。如果当前的转接指针所指示的下一级指定转接序列的逻辑节点标识符不是该边界节点祖先的节点标识符,则应清除该呼叫并携带返回原因#41(暂时故障)如果使用折回功能,则折回原因为#160(DTL 转接的不是本节点 ID)。当退出该对等组时,如果逻辑端口标识符并未设置为 0,且尚未规定逻辑端口,则逻辑端口标识符指示的逻辑端口将用来进展呼叫。如果逻辑端口标识符并未设置为 0,且未包含在先前递归过程检测到的所有逻辑端口中(例如:DTL 中的规定处于层次结构中的更低一级),这样就存在着与用于进展呼叫的端口出现冲突,这时可能会产生呼叫清除并使用返回原因#41(暂时故障)。

——如果当前的转接指针指示的是指定转接序列(DTL)的末尾,并且 DTL 的堆栈中仍有 DTL 存在,则递归执行 A2.2 2)规定的过程。

——如果当前的转接指针指示并不是指定转接序列(DTL)的末尾,它将按照当前转接指针指示的转接后的内容作为路由计算的目的地,执行 A2.2 3)的过程。

——如果 DTL 堆栈中已经没有 DTL,在决定那个节点是 DTL 终止节点之前,它应当保存该 DTL 的等级(在 DTL 堆栈的底部)作为通路范围,在这种情况下,它应执行 A3 规定的程序。否则,则应使用如下的内容作为路由选择的目的地,执行 A2.2 节 3)条规定的程序:

如果存在转接网选择信息单元,则使用转接网和被叫用户号码。

如果没有转接网选择信息单元,则使用被叫用户号码。

在这种情况下,任何指示用于呼叫进展的逻辑端口的端口 ID 将被忽略。

3) 只有在最低级的入口边界节点, 才能显示出到目的地的通路并进行计算。在通路末尾的节点必须具备到使用通告成员范围的目的地连通性(参见 5.8.1.3 节), 该通告成员范围至少应等于通路范围。(例如: 通路范围 \geq 通告的可到达的地址或转接网的范围)。当目的地的被叫用户号码是一组地址(ATM 的任意播呼叫), 则该通路必须指向 ATM 组的一个成员, 该成员应带有足够高的成员范围。

使用 DTL 描述的选择好的通路应被推入栈中, 这些 DTL 按照它们需要经过反序推入栈中。每个 DTL 应当包含一个按照在单一层面上经过顺序的转接表, DTL 中的当前转接指针应当被设置为 0, 用来指示第一个转接。如果没有到达目的地的合适的通路选择, 则呼叫应被清除并折回, 并且按照 B2.1 描述的程序, 带有相应的返回原因。

注: 在 A2.2 的程序里, 当前指针不需增加, 且没有 DTL 需要从堆栈中弹出。

接下来执行 A3 的程序, 否则呼叫或连接将被清除。

A2.3 既不是 DTL 起源节点也不是入口边界节点的程序。

应当检查 SETUP 或 ADD PARTY 消息中最上的一个 DTL 信息单元, 如果当前转接指针所指示的节点标识符与与该节点的节点标识符不等, 则使用 #41 原因(暂时故障)清除呼叫, 并且可以选择呼叫折回, 后续端阻塞指示语和折回原因 #128(下一节点不可达); 或者折回原因 #160(DTL 转接不是本节点 ID), 且 DTL 转接序列是一个阻塞的节点。

接下来执行 A3 的程序, 否则呼叫或连接将被清除。

A3 DTL 处理的下一步

如果 DTL 堆栈顶部的第一个转接序列信息单元的当前指针并没有指向 DTL 信息单元的最后一个转接项, 则转接指针应增加以下内容。

1) 如果指示的转接项与当前节点在同一级上是相邻的, 则应使用规定的逻辑端口(如果存在)把连接进展到下一转接项上。但是, 如果规定的逻辑端口不能与到下一转接项的逻辑链路相对应, 则呼叫应折回, 并使用折回原因 #128(下一节点不可达)、#2 原因(无路由到规定的转接网)和 #3 原因“无路由到目的地”。

2) 如果指示的转接项与当前节点不相邻, 则连接应折回, 并使用折回原因 #128(下一节点不可达)、#2 原因(无路由到规定的转接网)和 #3 原因(无路由到目的地)。

如果呼叫继续进行, 则 SETUP 和 ADD PARTY 消息中应包含修改后的 DTL 堆栈。

如果逻辑端口未设置为 0, 则指示的逻辑端口将用于进展呼叫。

如果当前的转接指针指示的是 DTL 的末尾, 则该 DTL 应当被从栈中弹出, 如果该 DTL 是堆栈中的最后一个, 则应保存该 DTL 的级别作为路由范围。如果在 DTL 堆栈中还有其他的 DTL, 则该节点应当检查下一个 DTL。如果下一个 DTL 的当前转接指针指示的逻辑节点标识符不是边界节点祖先的节点标识符之一, 则清除呼叫, 并使用 #41 原因(暂时故障)。如果折回, 则折回原因为 #160(DTL 转接不是本节点 ID)。如果逻辑端口标识符未设置为 0, 且不包含在先前递归过程所检测到的逻辑端口(例如: 在更低一级的 DTL 中规定), 这样就存在着与用于进展呼叫的端口出现冲突, 这时可能会产生呼叫清除并使用返回原因 #41(暂时故障)。如果逻辑端口未设置为 0, 且无逻辑端口被规定, 则端口标识符所指示的逻辑端口将用于进展呼叫。这些过程发现的(例如: 在规定的最低一级中的一个)第一个规定的逻辑端口将用于进展呼叫。

然后重复递归 A.3 规定的程序。

如果 DTL 堆栈已经为空, 则该节点为 DTL 终止节点。除了由原来收到的 DTL 堆栈顶上的当前转接指针规定的端口 ID 外, 其他用于指示呼叫进展的端口 ID 将被忽略。如果转接网选择信息单元存在, 且指示的转接网正好又由该节点服务, 则该节点使用 UNI 的程序将该呼叫进展到转接网。如果转接网选择信息单元不存在, 但被叫用户由该节点服务, 则该节点使用 UNI 的程序将该呼叫进展到被叫用户。DTL 终止节点到转接网或被叫用户的连通性必须通告成员范围高于或等于通路范围(例如: 通告范围 \leq 通路范围)。如果用户号码是一组地址(例如该呼叫使用 ATM 任播能力), 则呼叫可以递交给 ATM 组的

一个成员，该成员应带有足够高的成员范围。

如果 DTL 堆栈变为空，但没有足够的成员范围与被叫用户或转接网连通，则呼叫必须被清除或按照 B2.1 的程序折回。

A4 对 DTL 中包含端口 ID 的建议

DTL 中使用端口 ID 是任选的。尤其是端口 ID 可以设置为 0，用以指示到 DTL 堆栈中的下一节点可以使用任何端口。

这里只是建议使用端口 ID，并不要求一定要使用。当 DTL 堆栈中的下一节点在拓扑数据库中使用简单节点表示或缺省节点表示时，端口 ID 的应用就不必进行规定。这允许一个逻辑组中的入口边界节点，使用通过逻辑节点表示的子对等组的拓扑信息去选择一个与外部逻辑组相连的出口边界节点。这样的选择要优于通过起源节点去选择出口边界节点，因为起源节点可能并不了解子对等组的内部拓扑。但是如果还有其他结果出现(例如：在到下一节点的链路上使用不同的策略)，则不必考虑本建议。

当 DTL 堆栈中的下一节点使用带有一个或多个例外的复杂节点表示，则建议规定端口 ID。在这种情况下，起源节点在计算通路时利用对复杂拓扑信息通过下一节点，起源节点可以选择适合经过复杂节点的端口内的链路。入口边界节点进行的路由计算与之相比，它通常不考虑路由的含义，它们的选择是在目的地节点的进入端口之上的。

A5 DTL 程序的伪代码

在处理接收到的 DTL 堆栈和产生新的 DTL 堆栈时所要求的操作如下

- 1) 收到一个 PNNI SETUP 或 ADD PARTY 消息
- 2) 从 SETUP 或 ADD PARTY 消息中提取 DTL 堆栈
- 3) 栈顶 DTL 的当前转接项是否与本节点所处的路由层次的级别相对应？
 - a 是：设置当前节点和当前端口（本地变量）为栈顶 DTL 的当前转接项。
 - b 否：中止，差错产生
- 4) 当前的节点（从步骤 3）跳转）与路由层次中最低级的本节点相对应？
 - a 是：执行步骤 6)
 - b 否：当前的转接指针是否指向 DTL 中的最后一个单元？
 - b.1 是：DTL 堆栈中是否还有其他的 DTL？
 - b.1.1 是：检查堆栈中的下一个 DTL（本阶段不从堆栈中弹出任何 DTL）
 - b.1.1.1.1 是：当前的端口是否设置为 0（未规定）？
 - b.1.1.1.1.1 是：重新置当前端口为 DTL 中的当前转接项并执行 b
 - b.1.1.1.1.2 否：当前的端口是否包含在该 DTL 的当前转接端口中？
 - b.1.1.1.1.2.1 是：执行 b
 - b.1.1.1.1.2.2 否：中止，差错产生
 - b.1.1.1.2.2 否：中止，差错产生。
 - b.1.1.2 否：已经到达与被叫用户号码或规定转接网有可达性的逻辑组节点。设置路由范围（本地变量）等于该 DTL 的等级（在 DTL 堆栈底），并且设置目的地（本地变量）为 SETUP 和 ADD PARTY 消息中的被叫用户号码和/或规定的转接网。
 - b.2 否：设置目的地（本地变量）为 DTL 中的下一入口本阶段中并不增加当前转接指针。

5) 确定一条路由通过当前的节点（从 3 跳转）并且使用当前的端口（从步骤 3 和 4 跳转）到目的地。如果目的地是被叫用户号码和/或是规定的转接网，则通路必须连至一个与范围高于或等于通路范围的目的地有连通性的节点。并把得到的通路转换为一组 DTL 信息单元，并把得到的 DTL 信息单元放入当前的 DTL 堆栈。

- 6) 设置输出端口（本地变量）为栈顶 DTL 的当前转接项。并对 DTL 栈顶的 DTL 开始步骤 7)。
- 7) 当前的转接指针是否指向 DTL 中的最后一个单元？
 - a 是：是否是 DTL 堆栈中的最后一个 DTL？
 - a.1 是：已经到达与被叫用户号码或规定的转接网有直接可达性的节点，设置通路范围（本地变量）为该 DTL 的等级，并把 DTL 弹出栈。到被叫用户号码或规定的转接网的连通性所具有的范围应高于或等于通路范围。直接向被叫用户或转接网发送一个 UNI 的 SETUP 消息。
 - a.2 否：把 DTL 弹出栈，DTL 堆栈的下一个 DTL 的当前的转接项是否与本节点所处的某一路由级相对应。
 - a.2.1 是：输出端口是否设置为 0？
 - a.2.1.1 是：重新设置输出端口为 DTL 中当前的转接项，执行步骤 7)。
 - a.2.1.2 否：输出端口是否包含在该 DTL 的当前转接端口中？
 - a.2.1.2.1 是：执行步骤 7)
 - a.2.1.2.2 否：中止，差错产生。
 - a.2.2 否：中止，差错产生。
 - b 否：继续
- 8) 增加栈顶 DTL 的当前转接指针。
- 9) 将得到的 DTL 堆栈重新放回至 SETUP 或 ADD PARTY 消息中。
- 10) 通过输出端口，直接向栈顶 DTL 的当前转接项发送 PNNI 的 SETUP 或 ADD PARTY 消息。

规定一条通路的方法和将得到的通路转换为 DTL 可以使用多种描述方式。这些可能的描述方式取决于如何划分路由算法和 DTL 规范。

总结上述算法，其主要步骤如下：

- 1) 确定当前节点（包括当前的层次等级），步骤 3)；
- 2) 确定通路选择的目的地，步骤 4)；
- 3) 确定路由并将通路放入 DTL 堆栈中，步骤 5)；
- 4) 确定输出端口，步骤 6)；
- 5) 从堆栈中弹出完整的 DTL，步骤 7)；
- 6) 增加当前转接指针步骤 8)。

A6 域外路由和 DTL

域外路由定义为 PNNI 路由字段以外的路由，在许多情况下，管理域外网络的组织不希望暴露他们的内部网络拓扑，所以尽管也使用了 DTL，但可能不需要提供详细的域外路由。没有必要对世界范围内的 PNNI 路由字段使用一致的等级定义作出保证。因此如果域外路由使用 DTL，不可能出现通路上所有的 DTL 使用一致的等级表示语（每个节点 ID 的第一个八位位组）。

因此当一个通路使用域外路由时，DTL 应终止于在 PNNI 路由字段中通告域外路由的节点（节点或特定的端口）。

附录 B

(标准的附录)

折回过程

本附录描述了在折回时附加的清除呼叫过程和 PNNI 节点的动作。折回是通过在第一个呼叫清除消息中包含折回信息单元来指示的 (RELEASE、RELEASE COMPLETE 或 ADD PARTY REJECT)。这些过程是 6.3.5 节描述的清除呼叫过程的补充过程。

B1 折回过程的范围

当一个呼叫在 PNNI 的字段中不能被进展下去的时候, 就可能出现折回。本标准用来说明在 PNNI 字段中要求折回的情况。无论何时, 规范指出的呼叫清除使用特定原因代码, 除非明确地指出, 否则折回过程不执行。

任何呼叫无论通过何种方法进展到被叫用户, 只要是由被叫用户拒绝则不进行折回。这些情况包括由于在被叫用户端的 UNI 出现故障而造成的由被叫用户的拒绝, 以及由于终端或用户系统内的原因而产生的拒绝。在这种情况下, 将返回一个带有原因信息单元的 RELEASE 或 RELEASE COMPLETE 消息, 这会使呼叫逐段清除到主叫用户。这时没有折回的情况发生, 也不会另外的路由上进行试呼。

由 DTL 终止节点确定的到被叫用户的 UNI 不能承载呼叫而拒绝的情况则可能产生折回, 同样的呼叫拒绝也可能发生在 PNNI 接口上, 但这种情况并不要求折回。

PNNI 中使用折回的过程可以分为以下 3 类:

可达性差错;

资源差错;

DTL 处理差错。

DTL 处理差错也可以认为是可达性差错的一个子集, 如同通常上的连通性变化的结果。

除了上面的 3 类, 当在某些节点进行通路选择时, 确定没有通路可以满足策略限制时, 也可能造成呼叫的折回。策略限制的规范在 PNNI 的下一版本中要求, 所以本标准中未包含违背策略的定义和与之相关折回原因编码。

B2 折回原因

对于呼叫清除的过程, 原因信息单元在第一个呼叫清除消息中是必备的。在折回过程中, 折回信息单元同原因信息单元一样都是必备的。折回信息单元必须包括一个折回原因编码, 它用来在 PNNI 字段中替代原因信息单元。

注: 当阻塞的节点或链路变化后, 折回信息单元中的原因编码和诊断可以被修改。特别是当折回通过一个生成包括阻塞节点或链路的 DTL 的入口边界节点时, 更需要如此。

B2.1 可达性差错

一般来讲, 可达性差错用来指示未找到去目的地的通路, 这种情况适用于在 SETUP 或 ADD PARTY 消息中的 DTL 内的中间的目的地点, 同时也适用于最终的目的地点 (转接网和被叫用户)。这种类型的故障说明在收到的 DTL 堆栈中的范围里, 没有通路到达目的地。这种情况不同于通路存在而要求的 QoS 不能满足的情况 (该类情况在 A2.2 资源差错中讨论)。

如果呼叫折回是由于可达性差错造成的, 则折回信息单元中规定的阻塞的转接项应当是一条阻塞的链路。如果下一节点、转接网络或被叫用户地址不能通过该节点的任何逻辑链路到达, 则阻塞链路的端口 ID 应当设置为 0。如果下一节点、转接网络或被叫用户地址不能通过 DTL 堆栈中规定的端口到达, 则阻塞链路的端口 ID 应当设置为该端口 ID。对于不能到达的转接网和被叫用户地址, 则阻塞链路后的节点 ID 应当设置为全 0。

B2.1.1 目的地不可达和转接网不可达

当一个节点收到的 SETUP 或 ADD PARTY 消息中包含的 DTL 指示的节点是通路上的最后一个节点，但从本节点没有高于或等于与被叫用户和转接网的通路范围的连通性，此时返回一个原因编码。在这种情况下，在节点的链路状态数据库中也没有对被叫用户或转接网的任何可达性信息，则使用原因编码 #2(没有路由到达指定的转接网)或原因 #3(没有到目的地的路由)。清除呼叫，同样的如果最大匹配地址前缀是由该节点或该节点的祖先通告的总结地址，则使用原因编码 #2(没有路由到达指定的转接网)或原因 #3(没有到目的地的路由)清除呼叫。

如果节点的链路状态数据库中包含了到被叫用户或转接网的可达性信息，但只是在一个或几个节点中（这些节点不是这个节点的祖先），则呼叫折回时使用 #2 折回原因(没有路由到指定的转接网)或 #3 原因(目的地不可达)。这种情况可能发生在当一个呼叫进入到分区的对等组中的一个特定的分区，这个折回原因编码说明了一个节点认为尽管目的地不在本分区内而产生的折回，但目的地可能仍旧可以通过其他分区到达。

在分区的对等组内，可能有多于一个逻辑节点要通告关于给定目的地的可达性，正在通告的内容可以由两个逻辑节点通告的一个概括地址。任何一个包含在概括地址中特定的目的地只可能通过其中的一个分区到达。否则就不可能产生分区的情况。如果规定了 #3 折回原因(目的地不可达)，则生成的 DTL 导致向错误的分区发出了 SETUP 或 ADD PARTY 消息的节点应当：1) 也可以判断出其它分区；2) 不可以判断出其它分区。对于情况 1)，则可以使用另外一条路由重新进行试呼；对于情况 2)，则向前方折回或清除呼叫。

一个节点只有当其收到的 DTL 在范围内才重新在另一路由上进行试呼。如果所有可能的通路和分区都被尝试过以后，则应通过反复的执行这些过程以确定是清除呼叫或向前方折回。

B2.1.2 下一节点不可达

如果 DTL 堆栈中的下一转接项从本节点不能直达的话，则返回该原因代码。无论这种情况何时发生，则呼叫都折回并使用 #128 折回原因(下一节点不可达)。如果转接网选择信息单元存在，则原因信息单元中使用的原因代码应设置为 #2(无路由到特定的转接网)，否则将设置为 #3 “无路由到达目的地”

B2.2 资源差错

B2.2.1 由于业务类别造成的资源差错

由于不能支持业务类别或承载类别时，可以产生折回，详细内容参见 6.5.2.3 节的 1)条和 6.5.2.7 节。

B2.2.2 由于业务量和 QoS 参数造成的资源差错

用来支持呼叫的资源是通过包含在 SETUP 或 ADD PARTY 消息中的业务量参数和/或 QoS 参数来计算的。资源差错用来指示未发现满足业务量和 QoS 参数要求的进行呼叫的通路。

不能满足业务量和 QoS 参数要求的情况可以通过通路选择时的 GCAC 算法(参见 5.13 节)或实际 CAC 算法(参见 6.5.2.3 节)来检测。由于在 PNNI 字段内没有足够的资源而被拒绝的呼叫通常要求折回。

如果 ATM 业务量描述语信息单元中要求的用户信元率不能被满足的话，则呼叫被折回并使用 #37 折回原因(用户信元率不可用)。同时也包括修改阻塞的节点/链路的相关基本 CAC 参数。

如果未发现满足要求的最大 CTD、峰值到峰值的 CDV、CLR（一个呼叫的一个或两个方向的通路），则呼叫被折回并使用 #49 折回原因(QoS 不可用)。指示造成呼叫拒绝的特定 QoS 参数可以通过设置适当的比特来进行判断，如：CTD 不可用、CDV 不可用或 CLR 不可用。

由于资源不足造成阻塞发生时，必须要确定阻塞是由于在前一链路的后端资源不足（相同资源要求的呼叫请求可能其他端口上被接受）；还是由于后续链路的前端资源不足（相同资源要求的呼叫请求可能其他端口上被接受）；或者是在节点内部的资源不足（通过该节点的所有资源要求的所有呼叫请求都可能被拒绝）。根据不同的响应，应执行 B3.1.3、B3.1.2 和 B3.1.1 的程序。

B2.2.3 由于 VPCI/VCI 分配造成的资源差错

当前方不能分配 VPCI（对于 SVP）或 VPCI/VCI 对（对于 SVC）时，应当执行 B3.1.2 规定的程序。如果没有其他路由可以试呼的话，或者是其他路由由故障，则将使用 #45 折回原因(无可用的 VPCI/VPI)。

按照 6.5.2.2 节的 1)条和 6.5.2.2 节的 2)条讨论的内容, 由于在后续方 VPCI/VCI 分配的资源差错导致的折回将使用 #35 折回原因(请求的 VPCI/VCI 不可用), 或者是 #45 折回原因(无可用的 VPCI/VCI)。当 VPCI/VCI 资源差错发生在 PNNI 接口的后续方, 则折回信息单元中的阻塞的转接类型必须设置为“呼叫或用户已经在该接口的后方被阻塞”。

B2.3 DTL 处理差错

当一个节点收到的 DTL 信息单元中, 由当前转接指针指示的逻辑节点标识符不能标识节点或祖先节点之一, 则产生折回。当连通性发生变化时, 可能出现暂时的这种 DTL 处理差错特别是当对等组头发生变化并且产生了一个新的父逻辑组节点。这种情况下产生的折回信息单元可以是:

- 1) 后续端阻塞指示语和折回原因 #128(下一节点不可达)。
 - 2) 折回原因 #160(DTL 转接项不是本节点 ID), 如附录 A 中讨论的内容一样。
- 列出的 DTL 转接项为阻塞的节点。

B3 折回等级和阻塞转接的过程

本节用来描述用以产生、解释和修改折回等级、阻塞的转接类型和阻塞的转接标识符的程序。这些过程提供了一种对从阻塞点、经过中间节点到允许另外选择路由进行呼叫的一个或多个节点的折回管理机制。

折回过程使用的折回等级和阻塞的转接类型用来确定哪个节点允许重新选择替换路由或替换链路。为了指示发生故障节点或链路, 折回信息单元中除了阻塞的转接类型外, 还必须包括一个阻塞的转接标识符子字段。应当避免重复试呼在替换路由中阻塞的节点或链路。

B3.3 节给出了程序使用的伪代码。本节规定的伪程序代码包含在 A5 节中。如果叙述文本中存在不明确的地方, 则应参照伪程序代码解决。如果文本与伪程序代码出现不一致时, 则应以文本为准。

B3.1 阻塞节点的程序

在阻塞的节点的折回根据不同的情况执行的程序也不相同。在输入端口、输出端口或节点自身, 这些程序将分别在 B3.1.3、B3.1.2 和 B3.1.1 节中讨论。

B3.1.1 阻塞发生在一个节点

当阻塞发生在一个节点, 则启动折回程序并发送一个适当的呼叫/连接清除消息 (RELEASE、RELEASE COMPLETE 或 ADD PARTY REJECT), 该消息中应包含一个折回信息单元。折回信息单元中必须包含一个折回等级子字段, 该子字段的值应设置为 DTL 栈顶的 DTL 中的第一个节点 ID。阻塞的转接类型必须设置为“阻塞的节点”, 阻塞的转接标识符必须是在收到的 SETUP 或 ADD PARTY 消息中栈顶 DTL 的当前转接指针所指示的在相应的层次等级中节点自身的节点 ID。折回信息单元中还应包含折回原因子字段。

B3.1.2 链路前端的阻塞

当链路前端的节点内的 CAC 过程确定没有足够的资源可利用时, 则链路阻塞被确定为是链路前端的阻塞。如果一个呼叫的链路前端的节点是一个入口边界节点, 则将执行 B3.2.2 节的程序。否则, 如果存在其他链路可以满足该节点收到的 SETUP 或 ADD PARTY 消息中 DTL 的要求, 则在另一替换路由上试呼。

如果没有替换路由可以进行试呼, 或者是替换路由故障的话, 则链路前端的该节点必须通过发送包含折回信息单元的呼叫清除消息 (RELEASE、RELEASE COMPLETE 或者 ADD PARTY REJECT 消息) 折回该呼叫或增加用户请求。折回信息单元中必须包含一个折回等级子字段, 该子字段的值应设置为栈顶 DTL 中的第一个节点 ID 的等级。阻塞的转接类型子字段必须设置为“阻塞的链路”, 阻塞的转接标识符必须指示出阻塞链路的标识符。阻塞链路前端的节点标识符和端口标识符应设置为接收到的 SETUP 或 ADD PARTY 消息中栈顶 DTL 的当前指针所指示的节点和端口 ID。如果当前转接项不是 DTL 中的最后一个节点, 则阻塞链路的后续节点应设置为栈顶 DTL 中的下一个节点 ID; 如果该节点是该呼叫的出口边界节点, 则被设置的下一节点的节点 ID 应根据收到的 DTL 堆栈来确定; 但如果该节点是 DTL

的终止节点，则阻塞链路的后续节点标识符应设置为全 0。

B3.1.3 链路后端的阻塞

当链路后端节点内的 CAC 过程认为没有足够的资源可用时，则链路阻塞被确定为是链路后端的阻塞。在这种情况下，节点必须通过发送包含折回信息单元的呼叫清除消息（RELEASE、RELEASE COMPLETE 或者 ADD PARTY REJECT 消息）折回该呼叫或增加用户请求。折回信息单元中必须包含一个折回等级子字段，该子字段的值应设置为栈顶的 DTL 中的第一个节点 ID。阻塞的转接类型必须设置为“呼叫或增加一方用户请求在该接口的后续端被阻塞”，折回信息单元中还应包含折回原因子字段。

B3.2 收到一个带有折回信息单元的呼叫清除消息

当一个节点收到一个包含有折回信息单元的呼叫清除消息（RELEASE、RELEASE COMPLETE 或者 ADD PARTY REJECT 消息）后，该节点应首先检查阻塞的转接类型的指示是否为“呼叫或增加一方用户请求在该接口的后续端被阻塞”。如果是的话，则将执行 B3.2.1 的程序；否则，如果该节点针对该呼叫产生的任何 DTL 等级高于或等于折回等级，则将执行 B3.2.2 的程序。否则该节点必须通过朝主叫方向发送适当的呼叫清除消息折回这个呼叫或增加一方用户请求（RELEASE 或者 ADD PARTY REJECT 消息），并包含关于前一接口未进行变化的折回信息单元。

B3.2.1 接收到一个指示该接口阻塞的呼叫清除消息

当收到的呼叫清除消息（RELEASE、RELEASE COMPLETE 或者 ADD PARTY REJECT 消息）的折回信息单元中的阻塞转接类型为“呼叫或增加一方用户请求在该接口的后续端被阻塞”，则将执行如下过程。如果链路前端的节点是呼叫的一个入口边界节点，则执行 B3.2.2 的程序。否则如果存在其他链路可以满足该节点收到的 SETUP 或 ADD PARTY 消息中 DTL 的要求，则在另一替换路由上试呼。此外，如果折回原因为 #35(请求的 VPCI/VCI 不可用)时，只有在这种情况下，SETUP 消息才会在该条阻塞的链路上使用不同的 VPCI（对于 SVP）或 VPCI/VCI（对于 SVC）重新发送。

如果没有替换路由可以进行试呼，或者是替换路由故障的话，则该节点应继续折回该呼叫或增加一方用户请求。阻塞的转接类型子字段必须改为指示“阻塞的链路”，且阻塞的转接标识符必须被插入到折回信息单元中。阻塞链路前端的节点标识符和端口标识符应设置为接收到的 SETUP 或 ADD PARTY 消息中栈顶 DTL 的当前指针所指示的节点和端口 ID。如果当前转接项不是 DTL 中的最后一个节点，则阻塞链路的后续节点应设置为栈顶 DTL 中的下一个节点 ID；如果该节点是该呼叫的出口边界节点，则被设置的下一节点的节点 ID 应根据收到的 DTL 堆栈来确定（参见 A3 节）。但如果该节点是 DTL 的终止节点，则阻塞链路的后续节点标识符应设置为全 0。折回信息单元中必须包含一个折回等级子字段，该子字段的值应设置为 DTL 栈顶的 DTL 中的第一个节点 ID 的等级。在完成对上述对折回信息单元的修改后，该节点应发送关于前一接口一个适当的呼叫清除消息（朝主叫方）。

B3.2.2 在折回级的入口边界节点收到一个呼叫清除消息

如果到达了一个在折回级上产生 DTL 的节点，该节点必须确定是在替换路由上重新呼叫，还是继续折回呼叫或增加一方用户请求。如果继续折回，则呼叫应按照 B3.2.2 节的 2) 条规定的内容继续进展；如果在替换路由上试呼，则呼叫应按照 B.3.2.2 节 1) 条规定的内容继续进展。

1) 替换路由。

如果在替换路由上试呼，则路由计算应与原始收到 SETUP 或 ADD PARTY 消息中的 DTL 一致，但新的路由中不能包含收到的呼叫清除消息（RELEASE 或 ADD PARTY REJECT 消息）折回信息单元中包含的阻塞的节点和链路。除了这些程序由入口边界节点执行的结果外，其他关于 SETUP 和 ADD PARTY 消息的进展应同原始收到的 SETUP 和 ADD PARTY 消息一样。

2) 折回到更高一级。

如果未在替换路由上试呼或在替换路由无法找到一条合适的通路通过对等组，那么折回应进入到更高一级对等组的入口边界节点。折回级应设置为收到的 SETUP 或 ADD PARTY 消息中栈顶 DTL 的第一个节点 ID 的等级。折回信息单元中的阻塞节点和链路标识符必须进行修改，把一个由于对等组了解的一个节点或链路映射为一个由父对等组了解的节点或链路。

如果 RELEASE 和 ADD PARTY REJECT 消息中的折回信息单元只有对等组内部节点和/或链路作为阻塞的节点和链路, 则逻辑节点对应的对等组将被列为阻塞。在这种情况下, 阻塞的节点 ID 应当同收到的 SETUP 或 ADD PARTY 消息中栈顶 DTL 的当前转接指针指示的节点 ID 相匹配。

如果至少有一条路由试呼在退出对等组的链路上阻塞, 则退出用逻辑组节点表示的对等组的逻辑链路应列为阻塞。应明确的是阻塞链路的前端节点 ID 和端口 ID 应当设置为由收到的 SETUP 或 ADD PARTY 消息中的栈顶 DTL 的当前转接指针指示的内容。链路阻塞的后续节点 ID 应当设置为通路计算的目的地节点 ID, 应按照 A2.2 节的程序根据收到的 DTL 确定。但如果通路计算的目的地不是一个节点, 则阻塞链路后续节点 ID 应设置为全 0。

B3.2.3 对于点—多点呼叫/连接的附加程序

在点—多点的呼叫/连接中除了收到的是 RELEASE 或 RELEASE COMPLETE 消息, 且 ADD PARTY 表中还有排队的 ADD PARTY 请求之外, 其他用于处理点—多点的呼叫/连接中折回呼叫的清除消息的程序同处理点—点的呼叫/连接的程序相同。

如果网络节点在替换路由上未对收到 RELEASE 或 RELEASE COMPLETE 消息的增加一方用户请求进行试呼, 且折回信息单元中阻塞的转接是该接口的后续端, 则网络节点应向前一网络节点对每个 ADD PARTY 请求发送一个 ADD PARTY REJECT 消息, 其中的折回指示应符合前一子节的规定。

如果网络节点在替换路由上未对收到 RELEASE 或 RELEASE COMPLETE 消息的增加一方用户请求进行试呼, 且折回信息单元中阻塞的转接不是该接口的后续端, 则网络节点应当对收到 RELEASE 或 RELEASE COMPLETE 的增加一方用户请求折回。并且可以:

1) 通过发送一个 SETUP 消息, 进展 ADD PARTY 表中的一个 ADD PARTY 请求, 其他 ADD PARTY 请求处于未完成。

2) 折回所有 ADD PARTY 表中其 DTL 中包含阻塞转接类型的 ADD PARTY 请求, 并且通过发送一个 SETUP 消息, 进展 ADD PARTY 表剩余的一个 ADD PARTY 请求(如果存在), 其他 ADD PARTY 请求处于未完成。

如果网络节点在替换路由上对收到 RELEASE 或 RELEASE COMPLETE 消息的增加一方用户请求进行试呼, 则节点应当确定 ADD PARTY 请求能否被满足以及如何被满足。如果增加该请求到一个处于激活状态的分支, 可以满足该请求的话, 则应向相关的后续节点发送一个 ADD PARTY 消息。如果请求了一个新的分支, 且该分支处于空(NULL)状态, 它也应向相关的后续节点发送一个 ADD PARTY 消息。如果请求了一个新的分支, 当该分支处于呼叫启动、输出呼叫进行或呼叫递交状态, 则应该使该增加一方用户请求进行排队。可能多于一个分支需要满足所有排队的 ADD PARTY 请求。如果 ADD PARTY 请求不能被满足, 则网络节点应向前一节点发送一个关于该 ADD PARTY 请求的 ADD PARTY REJECT 消息。其中的折回指示应符合前一子节的规定。节点取得一个 ADD PARTY 请求是否可以被满足, 以及如何进行处理的方法由实施决定。

B3.3 折回程序的伪代码

当在一个节点检测到阻塞或应执行如下操作。

1) 由于资源不足引起的阻塞是在:

a. 整个节点

返回的呼叫或增加一方用户清除消息中包含:

折回等级=栈顶 DTL 的节点 ID 的等级

阻塞转接类型=阻塞节点

阻塞的节点 ID=栈顶 DTL 的当前转接的节点 ID

b. 下列链路的前端

该节点是否增加了 DTL? (对该呼叫是否是入口边界节点?)

b.1 是: 执行 2.b.1.1 的程序

b.2 否: 是否有其它可用链路可以满足收到的 DTL?

b.2.1 是：在替换链路上试呼

b.2.2 否：返回的呼叫或增加一方用户清除消息中包含：

折回等级=栈顶 DTL 的节点 ID 的等级

阻塞转接类型=阻塞链路

阻塞链路的前端节点 ID=栈顶 DTL 的当前转接的节点 ID

阻塞链路的端口 ID=栈顶 DTL 的当前转接的端口 ID

阻塞的链路的后续节点 ID

(如果不是出口边界节点或 DTL 终止节点)

=栈顶 DTL 的下一节点 ID (当前的转接项后)

(如果是退出节点)

=根据收到的 DTL 栈确定的下一节点 (在 DTL 的伪代码的步骤 7 和 8)

(如果是 DTL 终止节点)

=全 0

c. 前段链路的后端

返回的呼叫或增加一方用户清除消息中包含：

折回等级=栈顶 DTL 的节点 ID 的等级

阻塞转接类型=呼叫或增加一方用户请求在该接口的后续端被阻塞

当收到一个带有折回信息单元的呼叫清除消息时应执行如下操作：

2) 阻塞的转接类型=呼叫或增加一方用户请求在该接口的后续端被阻塞？

a. 是：折回原因是“请求的 VPCI/VCI 不可用”？

a.1 是：重新在相同的链路上使用不同的 VPCI/VCI 试呼。

a.2 否：该节点是否增加了 DTL (对该呼叫是否是入口边界节点)？

a.2.1 是：执行 2)b.1 的程序

a.2.2 否：是否有其他可用链路可以满足收到的 DTL？

a.2.2.1 是：在替换链路上试呼

a.2.2.2 否：返回的呼叫或增加一方用户清除消息中包含：

折回等级=栈顶 DTL 的节点 ID 的等级

阻塞转接类型=阻塞链路

阻塞链路的前端节点 ID=栈顶 DTL 的当前转接的节点 ID

阻塞链路的端口 ID=栈顶 DTL 的当前转接的端口 ID

阻塞的链路的后续节点 ID

(如果不是出口边界节点或 DTL 终止节点)

=栈顶 DTL 的下一节点 ID (当前的转接项后)

(如果是退出节点)

=根据收到的 DTL 栈确定的下一节点 (在 DTL 的伪代码的步骤 7 和 8)

(如果是 DTL 终止节点)

=全 0

b. 否：该节点产生的 DTL 等级高于或等于折回等级？

b.1 是：该节点产生的 DTL 的在折回等级中已到达(该节点是一个入口边界节点或 DTL 终止节点)。在替换路由上试呼？

b.1.1 是：(SETUP 和 ADD PARTY 消息已经被保存)

是否存在一条替换路由未与收到的 DTL 相违背，并且未包含收到的 RELEASE 消息中指示的阻塞的转接项。

b.1.1.1 是：在新通路上发送 SETUP 或 ADD PARTY 消息
(新通路必然要导致在折回的 DTL 上要生成新的 DTL)

b.1.1.2 否：

折回到 DTL 堆栈中的更高一级 **

折回等级=收到的栈顶 DTL 的节点 ID 的等级

如果逻辑组节点的前端链路发生阻塞

阻塞转接类型=呼叫或增加一方用户请求在该接口的后续
端被闭塞

如果阻塞原因发生在逻辑组节点内

阻塞转接类型=阻塞的节点

阻塞的节点 ID=收到的栈顶 DTL 当前转接项的节点 ID

如果逻辑组节点的后续链路发生阻塞

阻塞转接类型=阻塞的链路

阻塞链路的前端节点 ID=收到的栈顶 DTL 当前转接项
的节点 ID

阻塞链路的端口 ID=收到的栈顶 DTL 当前转接项的
端口 ID

阻塞链路的后续节点 ID=根据收到的 DTL 栈确定目的
地(在 DTL 的伪代码的步骤 4.b)，如果目的不是一个
节点，则设置为全 0。

b.1.2 否：

折回到 DTL 堆栈中的更高一级 **

折回等级=收到的栈顶 DTL 的节点 ID 的等级

如果逻辑组节点的前端链路发生阻塞

阻塞转接类型=呼叫或增加一方用户请求在该接口的后续端被阻塞

如果阻塞原因发生在逻辑组节点内

阻塞转接类型=阻塞的节点

阻塞的节点 ID=收到的栈顶 DTL 当前转接项的节点 ID

如果逻辑组节点的后续链路发生阻塞

阻塞转接类型=阻塞的链路

阻塞链路的前端节点 ID=收到的栈顶 DTL 当前转接项的节点 ID

阻塞链路的端口 ID=收到的栈顶 DTL 当前转接项的端口 ID

阻塞链路的后续节点 ID=根据收到的 DTL 栈确定下一目的地(在
DTL 的伪代码的步骤 4.b)，如果目的不是一个节点，则设置为
全 0。

b.2 否：将折回进展到前一接口

** 表示至少一个路由的试呼在退出对等组的链路上阻塞(该节点是一个入口边界节点)？

b.1.2.1/b.1.1.2.1 是：退出对等组的链路列为阻塞

b.1.2.2/b.1.1.2.2 否：逻辑组节点表示的对等组应列为阻塞。

B4 折回信息单元中携带修改拓扑状态参数

在许多情况下，资源差错造成的阻塞之所以发生是因为使用了过期或不准确的路由参数值。为了把现有的参数值增加到发端或入口边界节点的数据库信息中，规定在 5.8.1.1 节 3)条中的 GCAC 路由参数可以作为折回原因 #37(用户信元率不可用)的诊断被包含，这样使得随后的呼叫建立路由变动更加有效。

一个节点启动呼叫折回可以包括作为诊断信息的修改后的 GCAC 的参数值。其中之允许包含一个方向的修改的参数值，这是因为对于给定的链路每个节点只负责出局方向的发端路由参数值。修改后的 GCAC 参数适用于相同业务类别的呼叫请求。如果包含了 AvCR、CRM 和 VF，则使用复杂的 GCAC（参见 5.13.4.2 节）；如果只包含了 AvCR，则使用简单的 GCAC（参见 5.13.4.3 节）。

对于阻塞的链路，包含的值应设置为当时立即通告的 PTSE（对于上行节点侧的上下链路，也可以存在于 Hello 的 ULIA 中）中的值。对于阻塞的节点，包含的值应设置为 PTSE 中作为在复杂节点说明中的缺省值。讨论的这些值认为是当前值。本地节点总是了解其拓扑数据库的参数。这些在给定节点的链路状态数据库中过时的值认为是上次通告的值。

当生成了一个带有资源差错的折回信息单元，则节点启动的呼叫折回应包含适当参数的相关当前值。最后一次通告的值可能已经与当前值不匹配。当前值可能是最新的或者是更为准确的值，该值可能在阻挡完成后，在 PTSE 中发送。当前值可能与最后一次发送的 PTSE 的值相匹配，该 PTSE 与呼叫进展可能是向该节点的“信中的交叉”。当前值也可能是只有在一些更有效的事件或定时器恢复后才需要通告的内容。

处理修改参数值外，还应规定方向和端口 ID。对于阻塞的节点，其方向字段应规定为“前向”，端口 ID 设置为 0，对于阻塞的链路，“前向”用来指示呼叫在链路的前端被阻塞，“后向”用来指示呼叫在链路的后端被阻塞。如果方向被设置为“前向”，则在链路的前端节点，对按照规定端口的所有出局方向修改拓扑状态参数。

如果方向被设置为“后向”，则端口 ID 由启动折回呼叫的节点设置为 0。修改的参数值应对于收到 SETUP 或 ADD PARTY 消息的物理链路或 VPC。当收到带有后续端阻塞指示语的折回信息单元的前端节点，它必须要插入一个端口 ID 用来标识是一个 PNNI 端口。对于上行链路，修改的参数应适用于阻塞链路的前端节点入局方向的规定端口。对于水平链路，修改的参数适用于在阻塞链路的后续节点出局方向的相关端口。

只要呼叫折回到一个新的层次等级，则在每个入口边界节点必须被修改带有修改拓扑状态参数值的端口 ID。但只有端口合并到其他逻辑端口（或者在一个入口边界节点，或者对于阻塞在上行链路的后端，可能是前端节点），则修改的参数值和端口 ID 必须使用合并链路的值替代，或者是舍弃。

附录 C

(标准的附录)

软 PVC 程序

本附录描述了用来建立软 PVC 的程序，SPVC 的建立是管理动作，PNNI 中支持的 PVCC 和 PVPC 与 PVCC 和 PVPC 端点相对应。

两个网络节点间的软 PVPC/PVCC 的建立和释放是 PVC 的管理动作。在专用 ATM 网络地址同网络接口对应时，软 PVPC/PVCC 的连接端点是根据分配的唯一包括 SEL 的 ATM 地址进行标识（UNI3.1 和 4.0 信令规范）。软 PVPC/PVCC 在两个网络接口间建立，因此网络接口就成为软 PVPC/PVCC 的端点。

软 PVPC/PVCC 端点拥有 PVPC/PVCC，并且负责建立和释放连接。这个网络接口是主叫端点。如果由于交换系统故障或链路故障造成软 PVPC/PVCC 的交换部分断开，则主叫端点应负责重新尝试建立连接。重新尝试建立连接的频次由实施决定。

在软 PVPC/PVCC 建立之前，必须应当有方法能够唯一地标识 PVPC/PVCC 的端点，主叫端点的标识在主叫方号码信息单元中，软 PVPC/PVCC 的被叫方用来标识目的地网络节点的网络接口。网管系统提供软 PVPC/PVCC 端点的 ATM 地址和两个端点间使用的 VPI/VCI 值。

本附录描述了用于建立点一点软 PVPC/PVCC 和点一多点的软 PVCC 的信令程序。除了下述内容有所区别外，其他程序同建立点一点和点一多点的 SVC 程序相同。

C1 消息

AAL 参数信息单元、宽带高层信息、宽带低层信息、被叫用户子地址和主叫用户子地址信息单元用来传送未包含在 SETUP、CONNECT、ADD PARTY 和 ADD PARTY ACK 消息的端到端信息。

C2 点到点软 PVPC/PVCC 程序

本附录规定的程序使用下述第 6.5 节规定的基本呼叫/连接控制程序的附加程序。

因为 PVPC/PVCC 是出于管理目的而建立的，因此其参数与建立软 PVPC/PVCC 程序是分开的，并且不能进行端到端的协商。这些不需要协商的参数是业务量参数和 QoS 参数。如果交换系统不能支持 SETUP 消息中规定的这些参数，则该呼叫将折回，折回原因为 #47(资源不可用，未规定)。

C2.1 启动 PVPC/PVCC 建立

作为 PVPC/PVCC “拥有者”的连接端点将启动 PVPC/PVCC 的建立。当 PVPC/PVCC 开始配置时；或者是作为 PVPC/PVCC “拥有者”的交换节点变为工作后（例如：加电）；或者是从故障中恢复后，将发送 SETUP 消息。

VP 的承载业务类或 X 包含在用于标识端点间建立的 VPC 和 VCC 连接的宽带承载能力中，为了在目的地网络节点标识 PVPC/PVCC 的 VPI/VCI，被叫方的软 PVPC/PVCC 信息单元应当包含在 SETUP 消息中。被叫方号码信息单元应包含配置的对端 PVPC/PVCC 连接端点标识符，主叫方号码信息单元应包含 PVPC/PVCC 连接端点自身的标识符。

C2.2 在主叫的网络接口收到 CONNECT 消息

当起源节点收到一个 CONNECT 消息，它将 PVPC/PVCC 置为工作状态，如果连接消息包含被叫方软 PVPC/PVCC 信息单元，则被叫连接点和用户的 PVPC/PVCC 段的 VPI 或 VPI/VCI 值应传递到管理实体。

C2.3 在被叫的网络接口收到 SETUP 消息

当在被叫连接端点收到 SETUP 消息，则使用 6.2.5.4 节和 6.2.5.6 节规定的程序。被叫方号码标识作为软 PVPC/PVCC 被叫连接端点的网络接口。

C2.3.1 最后的 PVPC 段的 VPI 值的分配/选择

主叫连接端点应为软 PVPC 的被叫连接端点指示如下内容之一：

- a) 任何 VPI；
- b) 要求的 VPI。

在使用 b)的情况下，如果指示的 VPI 可用，则被叫连接点对呼叫使用该 VPI；在使用 a)的情况下，被叫连接点选择任何可用的 VPI。

包含在 CONNECT 消息中的被叫方软 PVPC/PVCC 信息单元中指示了被选择的 VPI 值。选择类型字段编码为“分配的值”。

在使用 b)的情况下，如果 VPI 不可用，则发送原因值为 #34(请求的被叫方软 PVPC/PVCC 不可用)的 RELEASE COMPLETE 消息。

C2.3.2 最后的 PVCC 段的 VPI/VCI 值的分配/选择

主叫连接点应为软 PVCC 的被叫连接端点指示如下内容之一：

- a) 任何 VPI；任何 VCI。
- b) 要求的 VPI；要求的 VCI。

在使用 b)的情况下，如果指示的 VPI/VCI 可用，则被叫连接点对呼叫使用该 VPI/VCI；在使用 a)的情况下，被叫连接点选择任何可用的 VPI/VCI。

包含在 CONNECT 消息中的被叫方软 PVPC/PVCC 信息单元中指示了被选择的 VPI/VCI 值。选择类型字段编码为“分配的值”。

在使用 b)的情况下，如果 VPI/VCI 不可用，则发送原因值为 #34(请求的被叫方软 PVPC/PVCC 不可用)的 RELEASE COMPLETE 消息。

C3 点到多点的 PVCC 程序

本附录规定的程序使用 6.6 节规定的点到多点呼叫/连接控制程序。

C3.1 在起源接口上增加一方用户

点一点点的 PVCC 的连接建立必须要有根连接节点启动。当 PVCC 开始配置；或者是由网管增加的新用户；或者是根连接节点变为工作状态（例如：加电）；或者是从故障中恢复，则根连接节点向一个叶连接节点发送 SETUP/ADD PARTY 消息。

C3.2 第一方用户的建立

点一点点 PVCC 的第一方用户建立总是通过发送 SETUP 消息启动的，承载业务类或 X 包含在用于标识端点间建立的 VCC 连接的宽带承载能力中。为了在目的地网络节点标识 PVCC 的 VPI/VCI 的值，被叫方的软 PVPC/PVCC 信息单元包含在 SETUP 消息中。被叫用户号码信息单元应当包含在配置的对端 PVCC 连接点的标识符中；主叫方号码信息单元应包含 PVCC 连接端点自身的标识符。

C3.2.1 在主叫网络接口接收到 CONNECT 消息

采用 C2.2 规定的程序。

C3.2.2 在被叫网络接口接收到 SETUP 消息

采用 C2.3 规定的程序。

C3.3 增加一方用户

在到第一个叶用户的连接建立后，通过向每个叶用户发送 ADD PARTY 消息，连接可以建立到其他叶用户。为了让用户能够在目的地网络节点标识 PVCC 的 VPI/VCI，被叫方软 PVPC/PVCC 信息单元包含在 ADD PARTY 消息中。被叫用户号码信息单元应包含在在配置的叶 PVCC 连接点的标识符中；主叫方号码信息单元应包含 PVCC 连接端点自身的标识符。

C3.3.1 在主叫网络接口上收到一个 ADD PARTY ACK 消息

如果 ADD PARTY ACK 消息中包含了被叫方软 PVPC/PVCC 信息单元，则被叫连接端点和用户间 PVCC 段的 VPI/VCI 的值应当传递到管理实体。

C3.3.2 在被叫网络接口上收到一个 ADD PARTY 消息

收到 ADD PARTY 消息后, 采用 6.6.2 节规定的程序。被叫方号码用来标识作为 PVCC 的被叫连接端点的网络端口。

C3.3.3 最后的 PVCC 段的 VPI/VCI 和端点参考的分配/选择

主叫连接端点应当使用如下之一指示软 PVPC/PVCC 的被叫端点。

- a) 任何 VPI; 任何 VCI。
- b) 要求的 VPI; 要求的 VCI。

在使用 b)的情况下, 如果指示的 VPI/VCI 可用, 则被叫连接点对呼叫使用该 VPI/VCI; 在使用 a)的情况下, 被叫连接点选择任何可用的 VPI/VCI 和端点参考。

包含在 ADD PARTY ACK 消息中的被叫方软 PVPC/PVCC 信息单元中指示了被选择的 VPI/VCI 值。选择类型字段编码为“分配的值”。

在使用 b)的情况下, 如果 VPI/VCI 不可用, 则发送 PNNI 规定的原因值为 #34(请求的被叫方软 PVPC/PVCC 不可用)的 ADD PARTY REJECT 消息。

附录 D
(标准的附录)
结构化常量

PNNI 规范定义了如下结构化常量。

DdfaultAdminWeight=5040

该常量的取值有如下原因：

● 5040 具有能够被 2、3、4、5、6、7、8、9、10、12、14、15、16、18、20 整除的特性，而得到管理加权的整数又比缺省值小。

● 当一组管理加权用于整个网络时，该缺省值允许通过增加数倍后，使其与从起源到目的地的加权相同。

● 这样允许管理者可以配置比缺省值小的管理加权。

ExpiredAge=zero(0)

在 PTSE 头中，PTSE 存活时间用来指示该 PTSE 已经超时，并应从拓扑数据库中删除。

GoupLeaderIncrmet=50

为了保证灵活性和稳定性，当一个系统决定作为对等组领导者进行工作时，则由 **GoupLeaderIncrmet** 增加其通告的对等组领导者优先级。

InitiaSequenceNumber=one(1)

所有第一次产生的 PTSE 的头中都应包含该 **InitiaSequenceNumber** 的值。

MaxLeadership=255

PGL 选择协议的最高头优先级。

MaxSequenceNumber=2³²-1

在 PTSE 头中 PTSE 的最大顺序号。

OverLoadRetryTime=300s

当一个节点处于拓扑数据库过负荷的状态时，该时间间隔用来表示数据库重新同步尝试的周期。

RCCQoSClass=0(未规定 QoS)

用于 RCC 的 QoS 类。

附录 E
(标准的附录)
结构变量

以下是 PNNI 规范中使用的结构变量。

AvCR-PM: 缺省值为 50, 允许范围 1~99。

最后通告的可用信元速率范围 $AvCR \pm (AvCR \times AvCR-PM/100)$ 中的变化百分比无意义。

AvCR-mT: 缺省值为 3, 允许范围 1~99。

小于自最后通告值的 AvCR 中的数量变化的 MaxCR 的百分比无意义, 即使当 AvCR-PM 的规则会指示变化有意义。

CDV-PM: 缺省值为 25, 允许范围 1~99。

最后通告的可用信元时延变化 $AvCR \pm (AvCR \times AvCR-PM/100)$ 中的变化百分比无意义。

ConfiguredLinkAggregationToken: 缺省值为 0。

链路合并令牌将在 Hello 协议中被通告。

DSRxmInterval: 缺省值为 5s。

一个节点在它重发前一个数据库总结包之前的等待时间 (以 s 计)。

HelloInterval: 缺省值为 15s。

在没有事件触发的 Hello 的情况下, 节点在一个链路上发送的 Hello 之间的时间间隔 (以 s 计)。

HorizontalLinkInactivityTime: 缺省值为 120s。

一个节点在还未收到和处理 LGN Horizontal Link IG 时继续通告一个 Horizontal Link 的时间 (以 s 计)。

InactivityFactor: 缺省值为 5。

在未收到一个 Hello 且在 Hello FSM 通告一条链路坏了之前所允许通过的 HelloInterval 的数目。

InitialLGNSVCTimeout: 缺省值为 4s。

推迟建立一个 RCC SVC 以便减少对一个新 LGN 的多个 SETUP 同步的时间间隔。

maxCTD-PM: 缺省值为 50, 允许范围 1~99。

最后通告的最大信元传送时延 $maxCTD \pm (maxCTD \times maxCTD-PM/100)$ 中的变化百分比无意义。

MinHelloInterval: 缺省值为 1s; 最小值为 0.1s。

连续发送的 Hello 的最小间隔。

MinPTSEInterval: 缺省值为 1s; 最小值为 0.1s。

更改任何给定的 PTSE 的时间间隔。换言之, PTSE 的新情况的发送间隔不能比每一个 MinPTSEInterval 更短。

OverrideDelay: 缺省值为 30s。

在 PGL 选举中, 一个节点等待所有节点同意哪一个应是所选举的节点的时间。

PeerDelayedAck Interval: 缺省值为 1s。

延迟的 PTSE 应答包的最小传送时间间隔。

PGLInitTime: 缺省值为 15s。

在 PGL 选举中 PGLInitTimer 的初始值。节点延迟规定它的 PGL 选择直到定时器超时。

PTSELifetimeFactor: 缺省值为 200%。

这是用来计算自身起源的 PTSE 的初始生存时间。初始生存时间置为 PTSERefresh Interval 和 PTSELifetimeFactor 的结果。

PTSERefresh Interval: 缺省值为 1800s。

在无触发的修改时, 自身起源的 PTSE 的重新起源之间的时间。一个节点以此速率重新起源 PTSE

以防止这些 PTSE 被其他节点清掉。

PTSERetransmissionInterval: 缺省值为 5s。

重发未应答的 PTSE 的时间间隔。在每个 PTSERetransmissionInterval 时间, PTSE 将被重发, 除非通过收到是: 一个应答包规定了 PTSE 情况; 还是同样的情况或一个由 flooding 产生的更近的情况来明确应答。

RCCMaximumBurstSize: 缺省值为 171 个信元。

一个 RCC 的双向请求 CLP=0+1 的最大突发大小。

RCCPeakCellRate: 缺省值为每秒 906 个信元。

一个 RCC 的双向请求 CLP=0+1 的峰值信元率。

RCCSustainableCellRate: 缺省值为 453 个信元/s。

一个 RCC 的双向请求 CLP=0+1 的可持续信元率。

ReElectionInterval: 缺省值为 15s。

一个节点发现它无法连到当前的 PGL, 等待重新启动 PGL 选举处理的时间间隔。

RequestRxmInterval: 缺省值为 5s。

在一个节点发送一个新的 PTSE 请求包请求还未收到的上一个 PTSE 请求包的 PTSE 之前的时间。

RetryLGNSVCTimeout: 缺省值为 30s。

一个建立 RCC SVC 的失败的尝试和同样 RCC SVC 的一个新的 SETUP 之间的延迟时间间隔。

SVCCalledIntegrityTime: 缺省值为 50s。

这个缺省值是用来初始化在收到由相邻节点起源的 LGN-LGN SVC 的节点的 SVCIntegrity Timer。

SVCCallingIntegrityTime: 缺省值为 35s。

这个缺省值是用来初始化在启动一个 LGN-LGN SVC 的节点的 SVCIntegrityTimer。

附录 F
(标准的附录)
PNNI 层次配置

本附录说明了配置 PNNI 层次的一个优选方法。PNNI MIB 允许用这个方法配置层次。层次的其他方法未提及。

PNNI 层次中每一个最低层次节点必须知道它自己的地址、零个或多个可到达的地址总结、它的节点 ID 和对等组 ID。

节点的缺省等级为 96。PNNI 层次不必按同一“深度”到达任一特定的 PNNI 路由范围。每一个最低等级对等组可从等级 0~104 配置。

层次和对等组 ID 均可被配置。若层次已被配置，则对等组 ID 是尾部填 0 的节点地址的等级比特的前缀。

节点 ID 也可基于等级被缺省；具体做法举例见 5.3.3 节。

若对等组中的所有节点已赋以相同 12 个八位位组前缀且是那个对等组唯一的地址，最低层次的结构可以因此被简化。这样节点 ID 和对等组 ID 不必在每个节点中分别配置。可用另一对等组 ID 和节点 ID 配置最低等级节点。

每个节点必须通告端系统，经过它可到达那些端系统。在某种情况下这可组成每一个可到达主机的主机地址表。为了允许规划 PNNI 路由，建议用一小组地址前缀总结主机可达性表示至多个主机的可达性。若最低等级节点有一个不为 104 的等级，它有一个占节点地址 13 个八位位组前缀的缺省总结地址。在配置节点时必须能通告其他总结地址前缀是没有它还是代之以缺省总结地址。

每个节点也需要知道描述节点和相邻节点间每个链路的特性以及节点本身特性的拓扑状态参数。在最低等级允许基于链路和节点的物理特性的拓扑状态参数缺省。在较高等级，这些拓扑状态参数可基于合并的较低等级链路和节点来计算。

不能成为 PGL 的节点（即使在最低等级）不需附加的结构信息。这些节点需要的完全最小结构信息仅为地址。

每一个对等组的 PGL 必须知道其父对等组的对等组 ID。能成为 PGL 的每一节点必须知道它的头优先级以及它的父对等组 ID 的选择。较高等级父对等组 ID 的配置可由人工配置整个 ID 或一个等级。当只配置一个等级时，ID 是较低等级对等组 ID 的前缀。

在一个给定层次，PGL 在一个也作为在那个交换系统参加每一个子孙对等组的作为 PGL 的节点例示的交换系统中例示。因此，若这个节点故障，它会影响大量对等组。这可由有一个存在于较高等级且有高 PGL 优先级的最低等级节点改善。这样的节点可负责那个层次以上的 PGL。

每一个节点的更详细的配置在附录 H 中规定的 PNNI MIB 中提供。

附录 G
(标准的附录)
PNNI 最小子集

本附录定义了实施一个完整 PNNI 交换系统的不同子集的最小功能要求。4 个 PNNI 交换系统子集按如下考虑：

- 最小功能交换系统：能支持多层次 PNNI 中内部链路的交换系统。
- PGL/LGN 交换系统：能例示 LGN 的交换系统。
- 边界节点交换系统：能支持外部链路的交换系统。
- 带 LGN 对等支持的边界节点交换系统：支持 LGN 对等的最低等级的边界节点交换系统。

边界节点和 PGL/LGN 子集是最小功能交换系统子集的扩展。带 LGN 对等支持子集的边界节点是边界节点子集的扩展。这个关系在图 G1 中示出。

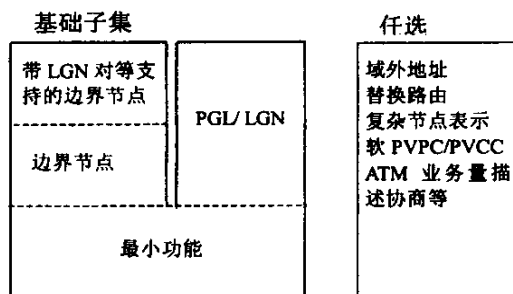


图 G.1 PNNI最小子集

所有的交换系统应支持最小功能交换系统的能力要求。其它能力任选。

表 G1 示出对应上述定义的每一个不同 PNNI 最小子集的必备和任选能力。

表 G1 必备和任选能力

No.	能力	交换系统
1	支持内部链路	M
2	版本协商	M
3	信息组标签	M
4	内部物理链路上的 Hello 协议	M
5	数据库同步	M
6	泛播	M
7	理解所有定义的 PTSE 类型	M
8	节点信息 PTSE 的起源	M
9	水平链路 PTSE 的起源	M
10	内部可到达地址 PTSE 的起源	M
11	PGL 选举投票	M
12	完成缺省内部地址归纳	M

续表 G1

No.	能力	交换系统
13	点到点呼叫	M
14	点到多点呼叫	M
15	单个 QoS 参数的信令	M
16	端到端连接完成通知	M
17	ATM 任播	M
18	产生多等级 DTL	M
19	遵循多等级 DTL	M
20	折回	M (注 1)
21	外部链路支持	B
22	外部链路上的 Hello 协议	B
23	产生上行链路的 PTSE	B, P
24	入和出边界节点 DTL 处理	B
25	入口边界节点折回处理	B
26	基于 SVC 的 RCC	N, P
27	基于 SVC 的 RCC Hello 协议	N, P
28	LGN 水平链路的 Hello 协议	N, P
29	PGL 能力	P
30	链路汇集	N, P
31	节点汇集	P
32	内部和域外地址归纳	P
33	域外可达地址	O
34	作为折回结果的替换路由	O
35	VPC 上的 Hello 协议	O
36	随路信令	O
37	ATM 话务描述符协商	O
38	SVP 业务	O
39	支持软 PVPC 和 PVCC	O
40	点到点呼叫的 ABR 信令	O
41	通用标识符传送	O
42	帧丢弃	O (注 2)
43	PNNI 链路上的 ILMI	O

M: 对于所有类别的交换系统 (即最小功能子集) 均为必备。

B: 边界节点交换系统要求; 否则任选。

N: 带 LGN 对等支持的边界节点交换系统要求; 否则任选。

P: PGL/LGN 交换系统要求; 否则任选。

O: 对于所有类别的交换系统均为任选。

注 1: 另一路由的支持是任选。

注 2: 帧丢弃指示的传送是必备的。

附录 H
(标准的附录)
PNNI 管理信息库

本附录定义了实施一个完整 PNNI 交换系统的管理信息数据库。

PNNI-MIB DEFINITIONS ::= BEGIN

IMPORTS

MODULE-IDENTITY, OBJECT-TYPE, OBJECT-IDENTITY,
Counter32, Gauge32, Integer32, enterprises
FROM SNMPv2-SMI
TEXTUAL-CONVENTION, RowStatus, DisplayString,
TimeStamp, TruthValue
FROM SNMPv2-TC
InterfaceIndex, ifIndex
FROM IF-MIB
AtmTrafficDescrParamIndex
FROM ATM-MIB
MODULE-COMPLIANCE, OBJECT-GROUP
FROM SNMPv2-CONF;

pnnimib MODULE-IDENTITY

LAST-UPDATED "9602270000Z"
ORGANIZATION "The ATM Forum"
CONTACT-INFO

"The ATM Forum
2570 West El Camino Real, Suite 304
Mountain View, CA 94040-1313 USA
Phone: +1 415-949-6700
Fax: +1 415-949-6705
info@atmforum.com"

DESCRIPTION

"The MIB module for managing ATM Forum PNNI routing."

REVISION "9602270000Z"

DESCRIPTION

"Initial version of the MIB for monitoring and controlling
PNNI routing."

::= { atmPnni 1 }

-- The object identifier subtree for ATM Forum PNNI MIBs

```
atmForum      OBJECT IDENTIFIER ::= { enterprises 353 }
atmForumNetworkManagement OBJECT IDENTIFIER ::= { atmForum 5 }
atmfPnni      OBJECT IDENTIFIER ::= { atmForumNetworkManagement 4 }
```

```
pnniMIBObjects OBJECT IDENTIFIER ::= { pnniMIB 1 }
```

```
Unsigned32 ::= TEXTUAL-CONVENTION
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"This definition, which is in compliance with RFC 1902, is a
temporary inclusion in the PNNI MIB until such time as MIB
compilers are upgraded and thereby can accept references to
the new definitions in RFC 1902."
```

```
REFERENCE
```

```
"RFC 1902"
```

```
SYNTAX      Gauge32
```

```
PnniAtmAddr ::= TEXTUAL-CONVENTION
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"The ATM address used by the network entity. The address
types are: no address (0 octets), and NSAP (20 octets)."
```

```
REFERENCE
```

```
"ATM Forum PNNI 1.0 Section 5.2"
```

```
SYNTAX      OCTET STRING (SIZE(0|20))
```

```
PnniNodeIndex ::= TEXTUAL-CONVENTION
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"An index that identifies a logical PNNI entity within the
managed system.
```

```
The distinguished value zero indicates the null instance or
no instance in the PnniNodeCfgParentNodeIndex. In all
other cases, the distinguished value zero indicates a
logical entity within the switching system that manages
routes only over non-PNNI interfaces.
```

```
By default, only the node identified by node index one is
```

created, and all PNNI interfaces are associated with that node."

SYNTAX Integer32 (0..65535)

PnniNodeId ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"A PNNI node ID - this is used to identify the logical PNNI node."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.3.3"

SYNTAX OCTET STRING (SIZE(22))

PnniPortId ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"A PNNI port ID - this is used to identify a point of attachment of a logical link to a given logical node.

The values 0 and 0xffffffff have special meanings in certain contexts and do not identify a specific port.

The distinguished value 0 indicates that no port is specified."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.3.4"

SYNTAX Unsigned32

PnniAggrToken ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"A PNNI aggregation token - this is used to determine which links to a given neighbor node are to be aggregated and advertised as a single logical link."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.3.5"

SYNTAX Unsigned32

PnniPeerGroupId ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"A PNNI peer group ID."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.3.2"

SYNTAX OCTET STRING (SIZE(14))

PnniLevel ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"A PNNI routing level indicator."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.3.1"

SYNTAX Integer32 (0..104)

PnniSvccRccIndex ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The value of this object identifies the SVCC-based RCC for which the entry contains management information."

SYNTAX Integer32

AtmAddrPrefix ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"A prefix of one or more ATM End System Addresses. The significant portion of a prefix is padded with zeros on the right to fill 19 octets."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.2"

SYNTAX OCTET STRING (SIZE(19))

PnniPrefixLength ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The number of bits that are significant in an ATM address prefix used by PNNI."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.2"

SYNTAX Integer32 (0..152)

PnniMetricsTag ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"An index into the pnniMetricsTable. The suffix tag is used to indicate that there may be many related entries in the table further discriminated by other index terms."

SYNTAX Integer32 (0..MAX)

ServiceCategory ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Indicates the service category."

REFERENCE

"ATM Forum Traffic Management 4.0 Section 2"

SYNTAX INTEGER { other(1),
cbr(2),
rtVbr(3),
nrtVbr(4),
abr(5),
ubr(6) }

ClpType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Indicates the CLP type of a traffic stream."

SYNTAX INTEGER { clpEqual0(1), clpEqual0Or1(2) }

TnsType ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"Indicates the type of network identification of a specified transit network."

REFERENCE

"ATM Forum UNI Signalling 4.0 Section 2 4.5.22/Q.2931"

SYNTAX INTEGER { nationalNetworkIdentification(2),
other(8) }

TnsPlan ::= TEXTUAL-CONVENTION

STATUS current

REFERENCE

"RFC 1902"

::= { 0 0 }

-- the base group

pnniBaseGroup OBJECT IDENTIFIER ::= { pnniMIBObjects 1 }

pnniHighestVersion OBJECT-TYPE

SYNTAX PnniVersion

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The highest version of the PNNI protocol that the software in this switching system is capable of executing."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.6.1"

::= { pnniBaseGroup 1 }

pnniLowestVersion OBJECT-TYPE

SYNTAX PnniVersion

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The lowest version of the PNNI Protocol that the software in this switching system is capable of executing."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.6.1"

::= { pnniBaseGroup 2 }

pnniDtlCountOriginator OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of DTL stacks that this switching system has originated as the DTLOriginator and placed into signalling messages. This includes the initial DTL stacks computed by this system as well as any alternate route (second, third choice etc.) DTL stacks computed by this switching system in response to crankbacks."

::= { pnniBaseGroup 3 }

pnniDtlCountBorder OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"The number of partial DTL stacks that this switching system has added into signalling messages as an entry border node. This includes the initial partial DTL stacks computed by this system as well as any alternate route (second, third choice etc.) partial DTL stacks computed by this switching system in response to crankbacks."

::= { pnniBaseGroup 4 }

pnniCrankbackCountOriginator OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"The count of the total number of connection setup messages including DTL stacks originated by this switching system that have cranked back to this switching system at all levels of the hierarchy."

::= { pnniBaseGroup 5 }

pnniCrankbackCountBorder OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"The count of the total number of connection setup messages including DTLs added by this switching system as an entry border node that have cranked back to this switching system at all levels of the hierarchy. This count does not include Crankbacks for which this switching system was not the crankback destination, only those crankbacks that were directed to this switching system are counted here."

::= { pnniBaseGroup 6 }

pnniAltRouteCountOriginator OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"The total number of alternate DTL stacks that this

switching system has computed and placed into signalling messages as the DTLOriginator."

::= { pnniBaseGroup 7 }

pnniAltRouteCountBorder OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of alternate partial DTL stacks that this switching system has computed and placed into signalling messages as an entry border node."

::= { pnniBaseGroup 8 }

pnniRouteFailCountOriginator OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of times where the switching system failed to compute a viable DTL stack as the DTLOriginator for some call. It indicates the number of times a call was cleared from this switching system due to originator routing failure."

::= { pnniBaseGroup 9 }

pnniRouteFailCountBorder OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of times where the switching system failed to compute a viable partial DTL stack as an entry border node for some call. It indicates the number of times a call was either cleared or cranked back from this switching system due to border routing failure."

::= { pnniBaseGroup 10 }

pnniRouteFailUnreachableOriginator OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of times where the switching system failed

to compute a viable DTL stack as the DTLOriginator because the destination was unreachable, i.e., those calls that are cleared with cause #2 `specified transit network unreachable' or cause #3 `destination unreachable' in the cause IE."

```
::= { pnniBaseGroup 11 }
```

pnniRouteFailUnreachableBorder OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of times where the switching system failed to compute a viable partial DTL stack as an entry border node because the target of the path calculation was unreachable, i.e., those calls that are cleared or cranked back with cause #2 `specified transit network unreachable' or cause #3 `destination unreachable' in the cause IE."

```
::= { pnniBaseGroup 12 }
```

-- node table

pnniNodeTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniNodeEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The pnniNodeTable collects attributes that affect the operation of a PNNI logical node.

There is a single row in this table for each PNNI peer group that the managed system is expected or eligible to become a member of."

REFERENCE

"ATM Forum PNNI 1.0 Annex F"

```
::= { pnniMIBObjects 2 }
```

pnniNodeEntry OBJECT-TYPE

SYNTAX PnniNodeEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table, containing information about a PNNI

logical node in this switching system."

REFERENCE

"ATM Forum PNNI 1.0 Annex F"

INDEX { pnniNodeIndex }

::= { pnniNodeTable 1 }

PnniNodeEntry ::=

SEQUENCE {

pnniNodeIndex	PnniNodeIndex,
pnniNodeLevel	PnniLevel,
pnniNodeId	PnniNodeId,
pnniNodeLowest	TruthValue,
pnniNodeAdminStatus	INTEGER,
pnniNodeOperStatus	INTEGER,
pnniNodeDomainName	DisplayString,
pnniNodeAtmAddress	PnniAtmAddr,
pnniNodePeerGroupId	PnniPeerGroupId,
pnniNodeRestrictedTransit	TruthValue,
pnniNodeComplexRep	TruthValue,
pnniNodeRestrictedBranching	TruthValue,
pnniNodeDatabaseOverload	TruthValue,
pnniNodePtses	Gauge32,
pnniNodeRowStatus	RowStatus

}

pnniNodeIndex OBJECT-TYPE

SYNTAX PnniNodeIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A value assigned to a node in this switching system that uniquely identifies it in the MIB."

::= { pnniNodeEntry 1 }

pnniNodeLevel OBJECT-TYPE

SYNTAX PnniLevel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The level of PNNI hierarchy at which this node exists. This attribute is used to determine the default node ID and the default peer group ID for this node. This object may only be written when pnniNodeAdminStatus has the value down."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.3.1, Annex F"

DEFVAL { 96 }

::= { pnniNodeEntry 2 }

pnniNodeId OBJECT-TYPE

SYNTAX PnniNodeId
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The value the switching system is using to represent itself as this node. This object may only be written when pnniNodeAdminStatus has the value down.

If pnniNodeLowest is true, then the default node ID takes the form defined in Section 5.3.3 for lowest level nodes, with the first octet equal to pnniNodeLevel, the second octet equal to 160, and the last 20 octets equal to pnniNodeAtmAddress.

If pnniNodeLowest is false, then the default node ID takes the form defined in Section 5.3.3 for logical group nodes, with the first octet equal to pnniNodeLevel, the next fourteen octets equal to the value of pnniNodePeerGroupId for the child node whose election as PGL causes this LGN to be instantiated, the next six octets equal to the ESI of pnniNodeAtmAddress, and the last octet equal to zero."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.3.3, Annex F"

::= { pnniNodeEntry 3 }

pnniNodeLowest OBJECT-TYPE

SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"Indicates whether this node acts as a lowest level node or whether this node is a logical group node that becomes active when one of the other nodes in this switching system becomes a peer group leader. The value 'false' must not be used with nodes that are not PGL/LGN capable.

This object may only be written when pnniNodeAdminStatus has the value down."

```
DEFVAL { true }
::= { pnniNodeEntry 4 }
```

pnniNodeAdminStatus OBJECT-TYPE

```
SYNTAX      INTEGER { up(1), down(2) }
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

DESCRIPTION

"Indicates whether the administrative status of the node is up (the node is allowed to become active) or down (the node is forced to be inactive).

When pnniNodeAdminStatus is down, then pnniNodeOperStatus must also be down."

```
DEFVAL { up }
```

```
::= { pnniNodeEntry 5 }
```

pnniNodeOperStatus OBJECT-TYPE

```
SYNTAX      INTEGER { up(1), down(2) }
```

```
MAX-ACCESS  read-only
```

```
STATUS      current
```

DESCRIPTION

"Indicates whether the node is active or whether the node has yet to become operational. When the value is down, all state has been cleared from the node and the node is not communicating with any of its neighbor nodes."

```
::= { pnniNodeEntry 6 }
```

pnniNodeDomainName OBJECT-TYPE

```
SYNTAX      DisplayString
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

DESCRIPTION

"The name of the PNNI routing domain in which this node participates. All lowest-level PNNI nodes with the same pnniNodeDomainName are presumed to be connected."

```
DEFVAL { "" }
```

```
::= { pnniNodeEntry 7 }
```

pnniNodeAtmAddress OBJECT-TYPE

```
SYNTAX      PnniAtmAddr
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

DESCRIPTION

"This node's ATM End System Address. Remote systems wishing to exchange PNNI protocol packets with this node should direct packets or calls to this address.

This attribute may only be written when pnniNodeAdminStatus has the value down."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.2.2"

::= { pnniNodeEntry 8 }

pnniNodePeerGroupId OBJECT-TYPE

SYNTAX PnniPeerGroupId

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The Peer Group Identifier of the peer group that the given node is to become a member of.

The default value of this attribute has the first octet equal to pnniNodeLevel, the next pnniNodeLevel bits equal to the pnniNodeLevel bits starting from the third octet of pnniNodeId, and the remainder padded with zeros.

This object may only be written when pnniNodeAdminStatus has the value down."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.3.2, Annex F"

::= { pnniNodeEntry 9 }

pnniNodeRestrictedTransit OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Specifies whether the node is restricted to not allowing support of SVCs transiting this node. This attribute determines the setting of the restricted transit bit in the nodal information group originated by this node."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.2.3"

DEFVAL { false }

::= { pnniNodeEntry 10 }

pnniNodeComplexRep OBJECT-TYPE

SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"Specifies whether this node uses the complex node representation. A value of 'true' indicates that the complex node representation is used, whereas a value of 'false' indicates that the simple node representation is used. This attribute determines the setting of the nodal representation bit in the nodal information group originated by this node."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.2.3"

::= { pnniNodeEntry 11 }

pnniNodeRestrictedBranching OBJECT-TYPE

SYNTAX TruthValue
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"Indicates whether the node is able to support additional point-to-multipoint branches. A value of 'false' indicates that additional branches can be supported, and a value of 'true' indicates that additional branches cannot be supported. This attribute reflects the setting of the restricted branching bit in the nodal information group originated by this node."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.2.3"

::= { pnniNodeEntry 12 }

pnniNodeDatabaseOverload OBJECT-TYPE

SYNTAX TruthValue
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"Specifies whether the node is currently operating in topology database overload state. This attribute has the same value as the Non-transit for PGL Election bit in the nodal information group originated by this node."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.2.3"

::= { pnniNodeEntry 13 }

pnniNodePtses OBJECT-TYPE

SYNTAX Gauge32
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"Gauges the total number of PTSEs currently in this node's topology database(s)."

::= { pnniNodeEntry 14 }

pnniNodeRowStatus OBJECT-TYPE

SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"To create, delete, activate and de-activate a Node."

::= { pnniNodeEntry 15 }

-- PGL election table

pnniNodePglTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniNodePglEntry
MAX-ACCESS not-accessible
STATUS current

DESCRIPTION

"Peer group leader election information for a PNNI node in this switching system."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.10.1"

::= { pnniMIBObjects 3 }

pnniNodePglEntry OBJECT-TYPE

SYNTAX PnniNodePglEntry
MAX-ACCESS not-accessible
STATUS current

DESCRIPTION

"An entry in the table, containing PGL election information of a PNNI logical node in this switching system."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.10.1"

AUGMENTS { pnniNodeEntry }

::= { pnniNodePglTable 1 }

```

PnniNodePglEntry ::=
    SEQUENCE {
        pnniNodePglLeadershipPriority    INTEGER,
        pnniNodeCfgParentNodeIndex      PnniNodeIndex;
        pnniNodePglInitTime              Integer32,
        pnniNodePglOverrideDelay         Integer32,
        pnniNodePglReelectTime           Integer32,
        pnniNodePglState                  INTEGER,
        pnniNodePreferredPgl             PnniNodeId,
        pnniNodePeerGroupLeader          PnniNodeId,
        pnniNodePglTimeStamp              Timestamp,
        pnniNodeActiveParentNodeId       PnniNodeId
    }

```

pnniNodePglLeadershipPriority OBJECT-TYPE

SYNTAX INTEGER (0..205)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The Leadership priority value this node should advertise in its nodal information group for the given peer group. Only the value zero can be used with nodes that are not PGL/LGN capable. If there is no configured parent node index or no corresponding entry in the pnniNodeTable, then the advertised leadership priority is zero regardless of this value."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.10.1.2"

DEFVAL { 0 }

::= { pnniNodePglEntry 1 }

pnniNodeCfgParentNodeIndex OBJECT-TYPE

SYNTAX PnniNodeIndex

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The local node index used to identify the node that will represent this peer group at the next higher level of hierarchy, if this node becomes peer group leader. The value 0 indicates that there is no parent node."

REFERENCE

"ATM Forum PNNI 1.0 Annex F"

DEFVAL { 0 }

::= { pnniNodePglEntry 2 }

pnniNodePglInitTime OBJECT-TYPE

SYNTAX Integer32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The amount of time in seconds this node will delay advertising its choice of preferred PGL after having initialized operation and reached the full state with at least one neighbor in the peer group."

REFERENCE

"ATM Forum PNNI 1.0 Annex G PGLInitTime"

DEFVAL { 15 }

::= { pnniNodePglEntry 3 }

pnniNodePglOverrideDelay OBJECT-TYPE

SYNTAX Integer32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The amount of time in seconds a node will wait for itself to be declared the preferred PGL by unanimous agreement among its peers. In the absence of unanimous agreement this will be the amount of time that will pass before this node considers a two thirds majority as sufficient agreement to declare itself peer group leader, abandoning the attempt to get unanimous agreement."

REFERENCE

"ATM Forum PNNI 1.0 Annex G OverrideDelay"

DEFVAL { 30 }

::= { pnniNodePglEntry 4 }

pnniNodePglReelectTime OBJECT-TYPE

SYNTAX Integer32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The amount of time in seconds after losing connectivity to the current peer group leader, that this node will wait before re-starting the process of electing a new peer group leader."

REFERENCE

"ATM Forum PNNI 1.0 Annex G ReElectionInterval"

DEFVAL { 15 }

::= { pnniNodePglEntry 5 }

pnniNodePglState OBJECT-TYPE

SYNTAX INTEGER {
 starting(1),
 awaiting(2),
 awaitingFull(3),
 initialDelay(4),
 calculating(5),
 awaitUnanimity(6),
 operPgl(7),
 operNotPgl(8),
 hungElection(9),
 awaitReElection(10)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the state that this node is in with respect to the Peer Group Leader election that takes place in the node's peer group. The values are enumerated in the Peer Group Leader State Machine."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.10.1.1.2"

::= { pnniNodePglEntry 6 }

pnniNodePreferredPgl OBJECT-TYPE

SYNTAX PnniNodeId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The Node ID of the node which the local node believes should be or become the peer group leader. This is also the value the local node is currently advertising in the 'Preferred Peer Group Leader Node ID' field of its nodal information group within the given peer group. If a Preferred PGL has not been chosen, this attribute's value is set to (all) zero(s)."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.10.1.1.6"

::= { pnniNodePglEntry 7 }

pnniNodePeerGroupLeader OBJECT-TYPE

SYNTAX PnniNodeId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The Node Identifier of the node which is currently operating as peer group leader of the peer group this node belongs to. If a PGL has not been elected, this attribute's value is set to (all) zero(s)."

::= { pnniNodePglEntry 8 }

pnniNodePglTimeStamp OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The time at which the current Peer Group Leader established itself."

::= { pnniNodePglEntry 9 }

pnniNodeActiveParentNodeId OBJECT-TYPE

SYNTAX PnniNodeId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The Node Identifier value being used by the Peer Group Leader to represent this peer group at the next higher level of the hierarchy. If this node is at the highest level of the hierarchy or if no PGL has yet been elected the PNNI Protocol Entity sets the value of this attribute to (all) zero(s)."

::= { pnniNodePglEntry 10 }

-- initial timer values table

pnniNodeTimerTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniNodeTimerEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table of initial PNNI timer values and significant change thresholds."

```
::= ( pnniMIBObjects 4 )
```

pnniNodeTimerEntry OBJECT-TYPE

SYNTAX PnniNodeTimerEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table, containing initial PNNI timer values and significant change thresholds of a PNNI logical node in this switching system."

AUGMENTS { pnniNodeEntry }

```
::= { pnniNodeTimerTable 1 }
```

PnniNodeTimerEntry ::=

SEQUENCE {

pnniNodePtseHolddown	Integer32,
pnniNodeHelloHolddown	Integer32,
pnniNodeHelloInterval	Integer32,
pnniNodeHelloInactivityFactor	Integer32,
pnniNodeHlinkInact	Integer32,
pnniNodePtseRefreshInterval	Integer32,
pnniNodePtseLifetimeFactor	INTEGER,
pnniNodeRxmtInterval	Integer32,
pnniNodePeerDelayedAckInterval	Integer32,
pnniNodeAvcrPm	INTEGER,
pnniNodeAvcrMt	INTEGER,
pnniNodeCdvPm	INTEGER,
pnniNodeCtdPm	INTEGER

}

pnniNodePtseHolddown OBJECT-TYPE

SYNTAX Integer32

UNITS "100 milliseconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The initial value for the PTSE hold down timer that will be used by the given node to limit the rate at which it can re-originate PTSEs. It must be a positive non-zero number."

REFERENCE

"ATM Forum PNNI 1.0 Annex G MinPTSEInterval"

DEFVAL { 10 }

```
::= { pnniNodeTimerEntry 1 }
```

pnniNodeHelloHolddown OBJECT-TYPE

SYNTAX Integer32
UNITS "100 milliseconds"
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The initial value for the Hello hold down timer that will be used by the given node to limit the rate at which it sends Hellos. It must be a positive non-zero number."

REFERENCE

"ATM Forum PNNI 1.0 Annex G MinHelloInterval"

DEFVAL { 10 }

::= { pnniNodeTimerEntry 2 }

pnniNodeHelloInterval OBJECT-TYPE

SYNTAX Integer32
UNITS "seconds"
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The initial value for the Hello Timer.

In the absence of triggered Hellos, this node will send one Hello packet on each of its ports on this interval."

REFERENCE

"ATM Forum PNNI 1.0 Annex G HelloInterval"

DEFVAL { 15 }

::= { pnniNodeTimerEntry 3 }

pnniNodeHelloInactivityFactor OBJECT-TYPE

SYNTAX Integer32
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The value for the Hello Inactivity factor that this node will use to determine when a neighbor has gone down."

REFERENCE

"ATM Forum PNNI 1.0 Annex G InactivityFactor"

DEFVAL { 5 }

::= { pnniNodeTimerEntry 4 }

pnniNodeHlinkInact OBJECT-TYPE

SYNTAX Integer32
UNITS "seconds"
MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The amount of time a node will continue to advertise a horizontal (logical) link for which it has not received and processed a LGN Horizontal Link information group."

REFERENCE

"ATM Forum PNNI 1.0 Annex G HorizontalLinkInactivityTime"

DEFVAL { 120 }

::= { pnniNodeTimerEntry 5 }

pnniNodePtseRefreshInterval OBJECT-TYPE

SYNTAX Integer32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The initial value for the Refresh timer that this node will use to drive (re-)origination of PTSEs in the absence of triggered updates."

REFERENCE

"ATM Forum PNNI 1.0 Annex G PTSERefreshInterval"

DEFVAL { 1800 }

::= { pnniNodeTimerEntry 6 }

pnniNodePtseLifetimeFactor OBJECT-TYPE

SYNTAX INTEGER (101..1000)

UNITS "percent"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The value for the lifetime multiplier, expressed as a percentage. The result of multiplying the pnniNodePtseRefreshInterval attribute value by this attribute value is used as the initial lifetime that this node places into self-originated PTSEs."

REFERENCE

"ATM Forum PNNI 1.0 Annex G PTSELifetimeFactor"

DEFVAL { 200 }

::= { pnniNodeTimerEntry 7 }

pnniNodeRxmtInterval OBJECT-TYPE

SYNTAX Integer32

UNITS "seconds"

MAX-ACCESS read-create
 STATUS current
 DESCRIPTION
 "The period between retransmissions of unacknowledged
 Database Summary packets, PTSE Request packets, and PTSPs."
 REFERENCE
 "ATM Forum PNNI 1.0 Annex G DSRxmtInterval,
 RequestRxmtInterval, PTSERetransmissionInterval"
 DEFVAL { 5 }
 ::= { pnniNodeTimerEntry 8 }

pnniNodePeerDelayedAckInterval OBJECT-TYPE

SYNTAX Integer32
 UNITS "100 milliseconds"
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION
 "The minimum amount of time between transmissions of
 delayed PTSE acknowledgement packets."
 REFERENCE
 "ATM Forum PNNI 1.0 Annex G PeerDelayedAckInterval,
 Appendix G"
 DEFVAL { 10 }
 ::= { pnniNodeTimerEntry 9 }

pnniNodeAvcrPm OBJECT-TYPE

SYNTAX INTEGER (1..99)
 UNITS "percent"
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION
 "The proportional multiplier used in the algorithms that
 determine significant change for AvCR parameters, expressed
 as a percentage."
 REFERENCE
 "ATM Forum PNNI 1.0 Section 5.8.5.2.5.4, Annex G AvCR_PM"
 DEFVAL { 50 }
 ::= { pnniNodeTimerEntry 10 }

pnniNodeAvcrMt OBJECT-TYPE

SYNTAX INTEGER (1..99)
 UNITS "percent"
 MAX-ACCESS read-create
 STATUS current

DESCRIPTION

"The minimum threshold used in the algorithms that determine significant change for AvCR parameters, expressed as a percentage."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.5.2.5.4, Annex G AvCR_mT"

DEFVAL { 3 }

::= { pnniNodeTimerEntry 11 }

pnniNodeCdvPm OBJECT-TYPE

SYNTAX INTEGER (1..99)

UNITS "percent"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The proportional multiplier used in the algorithms that determine significant change for CDV metrics, expressed as a percentage."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.5.2.5.6, Annex G CDV_PM"

DEFVAL { 25 }

::= { pnniNodeTimerEntry 12 }

pnniNodeCtdPm OBJECT-TYPE

SYNTAX INTEGER (1..99)

UNITS "percent"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The proportional multiplier used in the algorithms that determine significant change for CTD metrics, expressed as a percentage."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.5.2.5.5, Annex G maxCTD_PM"

DEFVAL { 50 }

::= { pnniNodeTimerEntry 13 }

-- nodal SVCC-based RCC variables table

pnniNodeSvccTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniNodeSvccEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table of variables related to SVCC-based routing control channels.."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.5"

::= { pnniMIBObjects 5 }

pnniNodeSvccEntry OBJECT-TYPE

SYNTAX PnniNodeSvccEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table, containing SVCC-based RCC variables of a PNNI logical node in this switching system."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.5"

AUGMENTS { pnniNodeEntry }

::= { pnniNodeSvccTable 1 }

PnniNodeSvccEntry ::=

SEQUENCE {

pnniNodeSvccInitTime Integer32,

pnniNodeSvccRetryTime Integer32,

pnniNodeSvccCallingIntegrityTime Integer32,

pnniNodeSvccCalledIntegrityTime Integer32,

pnniNodeSvccTrafficDescriptorIndex AtmTrafficDescrParamIndex

}

pnniNodeSvccInitTime OBJECT-TYPE

SYNTAX Integer32

UNITS "seconds"

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The amount of time this node will delay initiating establishment of an SVCC to a neighbor with a numerically lower ATM address, after determining that such an SVCC should be established."

REFERENCE

"ATM Forum PNNI 1.0 Annex G InitialLGNSVCTimeout"

DEFVAL { 4 }

::= { pnniNodeSvccEntry 1 }

pnniNodeSvccRetryTime OBJECT-TYPE

SYNTAX Integer32
 UNITS "seconds"
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION
 "The amount of time this node will delay after an apparently still necessary and viable SVCC-based RCC is unexpectedly torn down, before attempting to re-establish it."
 REFERENCE
 "ATM Forum PNNI 1.0 Annex G RetryLGNSVCTimeout"
 DEFVAL { 30 }
 ::= { pnniNodeSvccEntry 2 }

pnniNodeSvccCallingIntegrityTime OBJECT-TYPE

SYNTAX Integer32
 UNITS "seconds"
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION
 "The amount of time this node will wait for an SVCC, which it has initiated establishment of as the calling party, to become fully established before giving up and tearing it down."
 REFERENCE
 "ATM Forum PNNI 1.0 Annex G SVCCallingIntegrityTime"
 DEFVAL { 35 }
 ::= { pnniNodeSvccEntry 3 }

pnniNodeSvccCalledIntegrityTime OBJECT-TYPE

SYNTAX Integer32
 UNITS "seconds"
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION
 "The amount of time this node will wait for an SVCC, which it has decided to accept as the called party, to become fully established before giving up and tearing it down."
 REFERENCE
 "ATM Forum PNNI 1.0 Annex G SVCCalledIntegrityTime"
 DEFVAL { 50 }
 ::= { pnniNodeSvccEntry 4 }

pnniNodeSvccTrafficDescriptorIndex OBJECT-TYPE

SYNTAX AtmTrafficDescrParamIndex

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An index into the atmTrafficDescrParamTable defined in RFC 1695. This traffic descriptor is used when establishing switched virtual channels for use as SVCC-based RCCs to/from PNNI logical group nodes."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.5.2, Annex G
RCCMaximumBurstSize, RCCPeakCellRate,
RCCSustainableCellRate"

::= { pnniNodeSvccEntry 5 }

-- scope mapping table

pnniScopeMappingTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniScopeMappingEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The pnniScopeTable contains the mappings of membership and connection scope from organizational scope values (used at UNI interfaces) to PNNI scope (i.e. in terms of PNNI routing level indicators)."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.3.6"

::= { pnniMIBObjects 6 }

pnniScopeMappingEntry OBJECT-TYPE

SYNTAX PnniScopeMappingEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table, containing scope mapping information for a PNNI logical node in this switching system."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.3.6"

AUGMENTS { pnniNodeEntry }

::= { pnniScopeMappingTable 1 }

PnniScopeMappingEntry ::=

SEQUENCE {

pnniScopeLocalNetwork

PnniLevel,

pnniScopeLocalNetworkPlusOne	PnniLevel,
pnniScopeLocalNetworkPlusTwo	PnniLevel,
pnniScopeSiteMinusOne	PnniLevel,
pnniScopeIntraSite	PnniLevel,
pnniScopeSitePlusOne	PnniLevel,
pnniScopeOrganizationMinusOne	PnniLevel,
pnniScopeIntraOrganization	PnniLevel,
pnniScopeOrganizationPlusOne	PnniLevel,
pnniScopeCommunityMinusOne	PnniLevel,
pnniScopeIntraCommunity	PnniLevel,
pnniScopeCommunityPlusOne	PnniLevel,
pnniScopeRegional	PnniLevel,
pnniScopeInterRegional	PnniLevel,
pnniScopeGlobal	PnniLevel
}	

pnniScopeLocalNetwork OBJECT-TYPE

SYNTAX PnniLevel
 MAX-ACCESS read-create
 STATUS current

DESCRIPTION

"The highest level of PNNI hierarchy (i.e. smallest PNNI routing level) that lies within the organizational scope value localNetwork(1)."

DEFVAL { 96 }

::= { pnniScopeMappingEntry 1 }

pnniScopeLocalNetworkPlusOne OBJECT-TYPE

SYNTAX PnniLevel
 MAX-ACCESS read-create
 STATUS current

DESCRIPTION

"The highest level of PNNI hierarchy (i.e. smallest PNNI routing level) that lies within the organizational scope value localNetworkPlusOne(2)."

DEFVAL { 96 }

::= { pnniScopeMappingEntry 2 }

pnniScopeLocalNetworkPlusTwo OBJECT-TYPE

SYNTAX PnniLevel
 MAX-ACCESS read-create
 STATUS current

DESCRIPTION

"The highest level of PNNI hierarchy (i.e. smallest PNNI

routing level) that lies within the organizational scope value localNetworkPlusTwo(3)."

DEFVAL { 96 }

::= { pnniScopeMappingEntry 3 }

pnniScopeSiteMinusOne OBJECT-TYPE

SYNTAX PnniLevel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The highest level of PNNI hierarchy (i.e. smallest PNNI routing level) that lies within the organizational scope value siteMinusOne(4)."

DEFVAL { 80 }

::= { pnniScopeMappingEntry 4 }

pnniScopeIntraSite OBJECT-TYPE

SYNTAX PnniLevel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The highest level of PNNI hierarchy (i.e. smallest PNNI routing level) that lies within the organizational scope value intraSite(5)."

DEFVAL { 80 }

::= { pnniScopeMappingEntry 5 }

pnniScopeSitePlusOne OBJECT-TYPE

SYNTAX PnniLevel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The highest level of PNNI hierarchy (i.e. smallest PNNI routing level) that lies within the organizational scope value sitePlusOne(6)."

DEFVAL { 72 }

::= { pnniScopeMappingEntry 6 }

pnniScopeOrganizationMinusOne OBJECT-TYPE

SYNTAX PnniLevel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The highest level of PNNI hierarchy (i.e. smallest PNNI

routing level) that lies within the organizational scope value organizationMinusOne(7)."

DEFVAL { 72 }

::= { pnniScopeMappingEntry 7 }

pnniScopeIntraOrganization OBJECT-TYPE

SYNTAX PnniLevel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The highest level of PNNI hierarchy (i.e. smallest PNNI routing level) that lies within the organizational scope value intraOrganization(8)."

DEFVAL { 64 }

::= { pnniScopeMappingEntry 8 }

pnniScopeOrganizationPlusOne OBJECT-TYPE

SYNTAX PnniLevel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The highest level of PNNI hierarchy (i.e. smallest PNNI routing level) that lies within the organizational scope value organizationPlusOne(9)."

DEFVAL { 64 }

::= { pnniScopeMappingEntry 9 }

pnniScopeCommunityMinusOne OBJECT-TYPE

SYNTAX PnniLevel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The highest level of PNNI hierarchy (i.e. smallest PNNI routing level) that lies within the organizational scope value communityMinusOne(10)."

DEFVAL { 64 }

::= { pnniScopeMappingEntry 10 }

pnniScopeIntraCommunity OBJECT-TYPE

SYNTAX PnniLevel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The highest level of PNNI hierarchy (i.e. smallest PNNI

routing level) that lies within the organizational scope
value intraCommunity(11)."

DEFVAL { 48 }

::= { pnniScopeMappingEntry 11 }

pnniScopeCommunityPlusOne OBJECT-TYPE

SYNTAX PnniLevel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The highest level of PNNI hierarchy (i.e. smallest PNNI
routing level) that lies within the organizational scope
value communityPlusOne(12)."

DEFVAL { 48 }

::= { pnniScopeMappingEntry 12 }

pnniScopeRegional OBJECT-TYPE

SYNTAX PnniLevel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The highest level of PNNI hierarchy (i.e. smallest PNNI
routing level) that lies within the organizational scope
value regional(13)."

DEFVAL { 32 }

::= { pnniScopeMappingEntry 13 }

pnniScopeInterRegional OBJECT-TYPE

SYNTAX PnniLevel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The highest level of PNNI hierarchy (i.e. smallest PNNI
routing level) that lies within the organizational scope
value interRegional(14)."

DEFVAL { 32 }

::= { pnniScopeMappingEntry 14 }

pnniScopeGlobal OBJECT-TYPE

SYNTAX PnniLevel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The highest level of PNNI hierarchy (i.e. smallest PNNI

routing level) that lies within the organizational scope value global(15)."

```
DEFVAL { 0 }
 ::= ( pnniScopeMappingEntry 15 )
```

-- Summary advertising table

pnniSummaryTable OBJECT-TYPE

```
SYNTAX          SEQUENCE OF PnniSummaryEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
```

"A list of the summary address prefixes that may be advertised by the specified logical PNNI entity."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.9.2"

```
::= ( pnniMIBObjects 7 )
```

pnniSummaryEntry OBJECT-TYPE

```
SYNTAX          PnniSummaryEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
```

"An entry in the table, containing summary address prefix information in this switching system."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.9.2"

```
INDEX          { pnniNodeIndex,
                 pnniSummaryAddress,
                 pnniSummaryPrefixLength }
```

```
::= ( pnniSummaryTable 1 )
```

PnniSummaryEntry ::=

```
SEQUENCE {
    pnniSummaryAddress          AtmAddrPrefix,
    pnniSummaryPrefixLength    PnniPrefixLength,
    pnniSummaryType            INTEGER,
    pnniSummarySuppress        TruthValue,
    pnniSummaryState           INTEGER,
    pnniSummaryRowStatus       RowStatus
}
```

pnniSummaryAddress OBJECT-TYPE

```
SYNTAX          AtmAddrPrefix
```

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The ATM End System Address prefix for the summary."
::= { pnniSummaryEntry 1 }

pnniSummaryPrefixLength OBJECT-TYPE

SYNTAX PnniPrefixLength
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"The prefix length for the summary."
::= { pnniSummaryEntry 2 }

pnniSummaryType OBJECT-TYPE

SYNTAX INTEGER { internal(1), exterior(2) }
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"The type (e.g. internal or exterior) of summary being described."
DEFVAL { internal }
::= { pnniSummaryEntry 3 }

pnniSummarySuppress OBJECT-TYPE

SYNTAX TruthValue
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"Determines what is done with addresses that are being summarized by the instance. The default value (e.g. false) will indicate that the summary should propagate into the peer group. Network Management will be able to set the value of this attribute to 'suppress' (e.g. true), which suppresses the summary and any reachable addresses it summarizes from being advertised into the peer group."
DEFVAL { false }
::= { pnniSummaryEntry 4 }

pnniSummaryState OBJECT-TYPE

SYNTAX INTEGER {
advertising(1),
suppressing(2),
inactive(3)

```

    }
    MAX-ACCESS    read-only
    STATUS        current
    DESCRIPTION
        "Indicates whether the summary is currently being advertised
        by the node within the local switching system into its peer
        group."
    ::= { pnniSummaryEntry 5 }

```

pnniSummaryRowStatus OBJECT-TYPE

```

    SYNTAX        RowStatus
    MAX-ACCESS    read-create
    STATUS        current
    DESCRIPTION
        "To create, delete, activate and de-activate a summary."
    ::= { pnniSummaryEntry 6 }

```

-- Interface table

pnniIfTable OBJECT-TYPE

```

    SYNTAX        SEQUENCE OF PnniIfEntry
    MAX-ACCESS    not-accessible
    STATUS        current
    DESCRIPTION
        "The pnniIfTable contains the attributes necessary to
        configure a physical interface on a switching system which
        is capable of being used for PNNI routing. Interfaces may
        represent physical connection points (i.e. copper/fiber
        connection points) or VPCs which have been configured for
        PNNI's use. Each interface is attached to a specific
        lowest-level node within the switching system.

```

An ifIndex is used as the instance ID to uniquely identify the interface on the local switching system. This index has the same value as the ifIndex object defined in RFC 1573 for the same interface, since this table correlates with the ifTable in RFC 1573.

One row in this table is created by the managed system for each row in the ifTable that has an ifType of atm(37) or atmLogical(80)."

```

    ::= { pnniMIBObjects 8 }

```

pnniIfEntry OBJECT-TYPE

SYNTAX PnniIfEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION

"An entry in the table, containing PNNI specific interface information in this switching system."

INDEX { ifIndex }
 ::= { pnniIfTable 1 }

PnniIfEntry ::=

SEQUENCE {
 pnniIfNodeIndex PnniNodeIndex,
 pnniIfPortId PnniPortId,
 pnniIfAggrToken PnniAggrToken,
 pnniIfVPCapability TruthValue,
 pnniIfAdmWeightCbr Unsigned32,
 pnniIfAdmWeightRtVbr Unsigned32,
 pnniIfAdmWeightNrtVbr Unsigned32,
 pnniIfAdmWeightAbr Unsigned32,
 pnniIfAdmWeightUbr Unsigned32,
 pnniIfRccServiceCategory ServiceCategory,
 pnniIfRccTrafficDescrIndex AtmTrafficDescrParamIndex
 }

pnniIfNodeIndex OBJECT-TYPE

SYNTAX PnniNodeIndex
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION

"Identifies the node within the switching system that the interface is directly attached to."

DEFVAL { 1 }
 ::= { pnniIfEntry 1 }

pnniIfPortId OBJECT-TYPE

SYNTAX PnniPortId
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"The Port Identifier of the port as selected by the PNNI protocol entity for the given interface. This value has meaning only within the context of the node to which the port is attached."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.3.4"

::= { pnniIfEntry 2 }

pnniIfAggrToken OBJECT-TYPE

SYNTAX PnniAggrToken

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The configured aggregation token for this interface. The aggregation token controls what other links the link associated with this interface will be aggregated together with."

REFERENCE

"ATM Forum PNNI 1.0 Sections 5.3.5, 5.10.3.1"

DEFVAL { 0 }

::= { pnniIfEntry 3 }

pnniIfVPCapability OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Indicates whether the interface is capable of having VPCs established within it or not."

This object may only have the value 'true' for physical ATM interfaces, i.e. those with an ifType of atm(37)."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.14.9.1 Table 5-34"

::= { pnniIfEntry 4 }

pnniIfAdmWeightCbr OBJECT-TYPE

SYNTAX Unsigned32 (1..MAX)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The administrative weight of this interface for the constant bit rate service category."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.4"

DEFVAL { 5040 }

::= { pnniIfEntry 5 }

pnniIfAdmWeightRtVbr OBJECT-TYPE

SYNTAX Unsigned32 (1..MAX)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The administrative weight of this interface for the real-time variable bit rate service category."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.4"

DEFVAL { 5040 }

::= { pnniIfEntry 6 }

pnniIfAdmWeightNrtVbr OBJECT-TYPE

SYNTAX Unsigned32 (1..MAX)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The administrative weight of this interface for the non-real-time variable bit rate service category."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.4"

DEFVAL { 5040 }

::= { pnniIfEntry 7 }

pnniIfAdmWeightAbr OBJECT-TYPE

SYNTAX Unsigned32 (1..MAX)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The administrative weight of this interface for the available bit rate service category."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.4"

DEFVAL { 5040 }

::= { pnniIfEntry 8 }

pnniIfAdmWeightUbr OBJECT-TYPE

SYNTAX Unsigned32 (1..MAX)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The administrative weight of this interface for the unspecified bit rate service category."

REFERENCE


```

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.4"
DEFVAL { 5040 }
::= { pnniIfEntry 9 }

```

pnniIfRccServiceCategory OBJECT-TYPE

```

SYNTAX      ServiceCategory
MAX-ACCESS  read-write
STATUS      current

```

DESCRIPTION

"The service category used for the PNNI routing control channel (VCI=18) on this interface."

REFERENCE

"ATM Forum PNNI 1.0 Sections 5.5.2, 5.5.3"

```

::= { pnniIfEntry 10 }

```

pnniIfRccTrafficDescrIndex OBJECT-TYPE

```

SYNTAX      AtmTrafficDescrParamIndex
MAX-ACCESS  read-write
STATUS      current

```

DESCRIPTION

"The traffic descriptor index referring to the entry in the atmTrafficDescrParamTable defined in RFC 1695 that specifies the traffic allocation for the PNNI routing control channel (VCI=18) on this interface."

REFERENCE

"ATM Forum PNNI 1.0 Sections 5.5.2, 5.5.3, Annex G
RCCMaximumBurstSize, RCCPeakCellRate,
RCCSustainableCellRate"

```

::= { pnniIfEntry 11 }

```

-- link table

pnniLinkTable OBJECT-TYPE

```

SYNTAX      SEQUENCE OF PnniLinkEntry
MAX-ACCESS  not-accessible
STATUS      current

```

DESCRIPTION

"This table contains the attributes necessary to describe the operation of logical links attached to the local switching system and the relationship with the neighbor nodes on the other end of the links. Links are attached to a specific node within the switching system. A concatenation of the Node Index of the node within the

local switching system and the port ID are used as the instance ID to uniquely identify the link. Links may represent horizontal links between lowest level neighboring peers, outside links, uplinks, or horizontal links to/from LGNs.

The entire pnniLink object is read-only, reflecting the fact that this information is discovered dynamically by the PNNI protocol rather than configured."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.6"

::= { pnniMIBObjects 9 }

pnniLinkEntry OBJECT-TYPE

SYNTAX PnniLinkEntry
MAX-ACCESS not-accessible
STATUS current

DESCRIPTION

"An entry in the table, containing information about a link attached to a PNNI logical node in this switching system."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.6"

INDEX { pnniNodeIndex,
pnniLinkPortId }

::= { pnniLinkTable 1 }

PnniLinkEntry ::=

SEQUENCE {

pnniLinkPortId PnniPortId,
pnniLinkType INTEGER,
pnniLinkVersion PnniVersion,
pnniLinkHelloState PnniHelloState,
pnniLinkRemoteNodeId PnniNodeId,
pnniLinkRemotePortId PnniPortId,
pnniLinkDerivedAggrToken PnniAggrToken,
pnniLinkUpnodeId PnniNodeId,
pnniLinkUpnodeAtmAddress PnniAtmAddr,
pnniLinkCommonPeerGroupId PnniPeerGroupId,
pnniLinkIfIndex InterfaceIndex,
pnniLinkSvccRccIndex PnniSvccRccIndex,
pnniLinkRcvHellos Counter32,
pnniLinkXmtHellos Counter32
}

pnniLinkPortId OBJECT-TYPE

SYNTAX PnniPortId
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION

"The Port Identifier of the link as selected by the local node. This value has meaning only within the context of the node to which the port is attached."

::= { pnniLinkEntry 1 }

pnniLinkType OBJECT-TYPE

SYNTAX INTEGER {
 unknown(1),
 lowestLevelHorizontalLink(2),
 horizontalLinkToFromLgn(3),
 lowestLevelOutsideLink(4),
 uplink(5),
 outsideLinkAndUplink(6)
 }
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"Indicates the type of link being described."

::= { pnniLinkEntry 2 }

pnniLinkVersion OBJECT-TYPE

SYNTAX PnniVersion
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"For horizontal and outside links between lowest-level nodes and for links of unknown type, this attribute indicates the version of PNNI routing protocol used to exchange information over this link. If communication with the neighbor node has not yet been established, then the Version is set to 'unknown'. For uplinks (where the port ID is not also used for the underlying outside link) or links to/from LGNs, the Version is set to 'unknown'."

::= { pnniLinkEntry 3 }

pnniLinkHelloState OBJECT-TYPE

SYNTAX PnniHelloState
 MAX-ACCESS read-only
 STATUS current

DESCRIPTION

"For horizontal and outside links between lowest-level nodes and for links of unknown type, this attribute indicates the state of the Hello protocol exchange over this link. For links to/from LGNs, this attribute indicates the state of the corresponding LGN Horizontal Link Hello State Machine. For uplinks (where the port ID is not also used for the underlying outside link), this attribute is set to notApplicable."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.6.2.1.2"

::= { pnniLinkEntry 4 }

pnniLinkRemoteNodeId OBJECT-TYPE

SYNTAX PnniNodeId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the node identifier of the remote (neighboring) node on the other end of the link. If the pnniLinkType is 'outside link and uplink', this is the node identifier of the lowest-level neighbor node on the other end of the outside link. If the remote node ID is unknown or if the pnniLinkType is 'uplink', this attribute is set to all zeros."

::= { pnniLinkEntry 5 }

pnniLinkRemotePortId OBJECT-TYPE

SYNTAX PnniPortId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the port identifier of the port at the remote end of the link as assigned by the remote node. If the pnniLinkType is 'outside link and uplink', this is the port identifier assigned by the lowest-level neighbor node to identify the outside link. If the remote port ID is unknown or if the pnniLinkType is 'uplink', this attribute is set to zero."

::= { pnniLinkEntry 6 }

pnniLinkDerivedAggrToken OBJECT-TYPE

SYNTAX PnniAggrToken

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the derived aggregation token value used on this link. For horizontal links between lowest-level nodes and when the link type is not yet known, this attribute takes the value of zero."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.10.3.1"

::= { pnniLinkEntry 7 }

pnniLinkUpnodeId OBJECT-TYPE

SYNTAX PnniNodeId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"For outside links and uplinks, this attribute contains the Node Identifier of the upnode (the neighbor node's identity at the level of the common peer group). When the upnode has not yet been identified, this attribute is set to zero. For horizontal links or when the link type is not yet known, this attribute is set to zero."

::= { pnniLinkEntry 8 }

pnniLinkUpnodeAtmAddress OBJECT-TYPE

SYNTAX PnniAtmAddr

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"For outside links and uplinks, this attribute contains the ATM End System Address used to establish connections to the upnode. When the upnode has not yet been identified, this attribute is set to zero. For horizontal links or when the link type is not yet known, this attribute is set to zero."

::= { pnniLinkEntry 9 }

pnniLinkCommonPeerGroupId OBJECT-TYPE

SYNTAX PnniPeerGroupId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"For outside links and uplinks, this attribute contains the peer group identifier of the lowest level common Peer Group in the ancestry of the neighboring node and the node within the local switching system. The value of this attribute

takes on a value determined by the Hello exchange of hierarchical information that occurs between the two lowest-level border nodes. When the common peer group has not yet been identified, this attribute is set to zero. For horizontal links or when the link type is not yet known, this attribute is set to all zeros."

```
::= { pnniLinkEntry 10 }
```

pnniLinkIfIndex OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"For horizontal and outside links between lowest-level nodes and for links of unknown type, this attribute identifies the interface to which the logical link corresponds.

For all other cases, the value of this object is zero."

```
::= { pnniLinkEntry 11 }
```

pnniLinkSvccRccIndex OBJECT-TYPE

SYNTAX PnniSvccRccIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"For horizontal links to/from LGNs, this attribute identifies the SVCC-based RCC used to exchange information with the neighboring peer logical group node. If the pnniLinkType is not 'horizontal link to/from LGN', this attribute shall take the value of zero."

```
::= { pnniLinkEntry 12 }
```

pnniLinkRcvHellos OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"For horizontal and outside links between lowest-level nodes and for links of unknown type, this attribute contains a count of the number of Hello Packets received over this link. If the pnniLinkType is 'horizontal link to/from LGN' or 'uplink', this attribute is set to zero."

```
::= { pnniLinkEntry 13 }
```

pnniLinkXmtHellos OBJECT-TYPE

SYNTAX Counter32
 MAX-ACCESS read-only
 STATUS current

DESCRIPTION

"For horizontal and outside links between lowest-level nodes and for links of unknown type, this attribute contains a count of the number of Hello Packets transmitted over this link. If the pnniLinkType is 'horizontal link to/from LGN' or 'uplink', this attribute is set to zero."

::= { pnniLinkEntry 14 }

-- neighboring peer table

pnniNbrPeerTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniNbrPeerEntry
 MAX-ACCESS not-accessible
 STATUS current

DESCRIPTION

"The pnniNbrPeer Object contains all the attributes necessary to describe the relationship a node in this switching system has with a neighboring node within the same peer group. A concatenation of the Node Identifier of the node within the local switching system and the neighboring peer's Node Identifier is used to form the instance ID for this object.

The entire pnniNbrPeer object is read-only, reflecting the fact that neighboring peers are discovered dynamically by the PNNI protocol rather than configured."

REFERENCE

"ATM Forum PNNI 1.0 Sections 5.7, 5.8"

::= { pnniMIBObjects 10 }

pnniNbrPeerEntry OBJECT-TYPE

SYNTAX PnniNbrPeerEntry
 MAX-ACCESS not-accessible
 STATUS current

DESCRIPTION

"An entry in the table, containing information about this node's relationship with a neighboring peer node."

REFERENCE

"ATM Forum PNNI 1.0 Sections 5.7, 5.8"

```
INDEX      { pnniNodeIndex,
             pnniNbrPeerRemoteNodeId }
 ::= { pnniNbrPeerTable 1 }
```

PnniNbrPeerEntry ::=

```
SEQUENCE {
    pnniNbrPeerRemoteNodeId      PnniNodeId,
    pnniNbrPeerState             INTEGER,
    pnniNbrPeerSvccRccIndex      PnniSvccRccIndex,
    pnniNbrPeerPortCount         Gauge32,
    pnniNbrPeerRcvDbSums         Counter32,
    pnniNbrPeerXmtDbSums         Counter32,
    pnniNbrPeerRcvPtspS         Counter32,
    pnniNbrPeerXmtPtspS         Counter32,
    pnniNbrPeerRcvPtseReqs       Counter32,
    pnniNbrPeerXmtPtseReqs       Counter32,
    pnniNbrPeerRcvPtseAcks       Counter32,
    pnniNbrPeerXmtPtseAcks       Counter32
}
```

pnniNbrPeerRemoteNodeId OBJECT-TYPE

```
SYNTAX      PnniNodeId
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
```

"The Node Identifier of the neighboring peer node."

```
::= { pnniNbrPeerEntry 1 }
```

pnniNbrPeerState OBJECT-TYPE

```
SYNTAX      INTEGER {
                npdown(1),
                negotiating(2),
                exchanging(3),
                loading(4),
                full(5)
            }
```

```
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
```

"Indicates the state of this node's Neighboring Peer State Machine associated with pnniNbrPeerRemoteNodeId."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.7.2"

```
::= { pnniNbrPeerEntry 2 }
```


pnniNbrPeerSvccRccIndex OBJECT-TYPE

SYNTAX PnniSvccRccIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Identifies the SVCC-based RCC being used to communicate with the neighboring peer if one exists. If both the local node and the neighboring peer node are lowest-level nodes, this attribute is set to zero."

::= { pnniNbrPeerEntry 3 }

pnniNbrPeerPortCount OBJECT-TYPE

SYNTAX Gauge32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of the total number of ports that connect to the neighboring peer. If the neighboring peer only communicates via an SVCC-based RCC, the value of this attribute is set to zero. Otherwise it is set to the total number of ports to the neighboring peer in the Hello state 2-WayInside."

::= { pnniNbrPeerEntry 4 }

pnniNbrPeerRcvDbSums OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of the number of Database Summary Packets received from the neighboring peer."

::= { pnniNbrPeerEntry 5 }

pnniNbrPeerXmtDbSums OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of the number of Database Summary Packets transmitted to the neighboring peer."

::= { pnniNbrPeerEntry 6 }

pnniNbrPeerRcvPtsps OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "A count of the number of PTSPs received from the
    neighboring peer."
 ::= { pnniNbrPeerEntry 7 }
```

pnniNbrPeerXmtPtsps OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "A count of the number of PTSPs (re)transmitted to the
    neighboring peer."
 ::= { pnniNbrPeerEntry 8 }
```

pnniNbrPeerRcvPtseReqs OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "A count of the number of PTSE Request packets received from
    the neighboring peer."
 ::= { pnniNbrPeerEntry 9 }
```

pnniNbrPeerXmtPtseReqs OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "A count of the number of PTSE Request packets transmitted
    to the neighboring peer."
 ::= { pnniNbrPeerEntry 10 }
```

pnniNbrPeerRcvPtseAcks OBJECT-TYPE

```
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "A count of the number of PTSE Ack packets received from the
    neighboring peer."
 ::= { pnniNbrPeerEntry 11 }
```

pnniNbrPeerXmtPtseAcks OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A count of the number of PTSE Ack packets transmitted to the neighboring peer."

::= { pnniNbrPeerEntry 12 }

-- neighboring peer port table

pnniNbrPeerPortTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniNbrPeerPortEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table of all ports in Hello state 2-Way Inside to a given neighboring peer node. A concatenation of the Node Index of the node within the local switching system, the neighbor's Node Identifier and the Interface Index of the port being described forms the instance ID for this object. This object is only used for lowest-level nodes."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.7.1 Port ID List"

::= { pnniMIBObjects 11 }

pnniNbrPeerPortEntry OBJECT-TYPE

SYNTAX PnniNbrPeerPortEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table, containing information about a port in the Hello state 2-Way Inside from a PNNI logical node in this switching system to a neighboring peer node."

INDEX { pnniNodeIndex,
pnniNbrPeerRemoteNodeId,
pnniNbrPeerPortId
}

::= { pnniNbrPeerPortTable 1 }

PnniNbrPeerPortEntry ::=

SEQUENCE {

pnniNbrPeerPortId

PnniPortId,

```

        pnniNbrPeerPortFloodStatus    TruthValue
    }

```

pnniNbrPeerPortId OBJECT-TYPE

```

    SYNTAX      PnniPortId
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION

```

"The port ID of a port to the neighboring peer that is in the Hello state 2-Way Inside."

```
 ::= { pnniNbrPeerPortEntry 1 }
```

pnniNbrPeerPortFloodStatus OBJECT-TYPE

```

    SYNTAX      TruthValue
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION

```

"Indicates whether the port is being used for transmission of flooding and database synchronization information to the neighboring peer."

```
 ::= { pnniNbrPeerPortEntry 2 }
```

-- pnni SVCC-based routing control channel table

pnniSvccRccTable OBJECT-TYPE

```

    SYNTAX      SEQUENCE OF PnniSvccRccEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION

```

"A table containing the attributes necessary to analyze the operation of the PNNI protocol on SVCC-based Routing Control Channels. This entire object is read-only, reflecting the fact that SVCC-based RCCs are established dynamically during operation of the PNNI protocol rather than configured."

REFERENCE

"ATM Forum PNNI 1.0 Sections 5.5.6, 5.6.3.1"

```
 ::= { pnniMIBObjects 12 }
```

pnniSvccRccEntry OBJECT-TYPE

```

    SYNTAX      PnniSvccRccEntry
    MAX-ACCESS  not-accessible
    STATUS      current

```

DESCRIPTION

"An entry in the table, containing information about an SVCC-based RCC from a PNNI logical node in this switching system."

REFERENCE

"ATM Forum PNNI 1.0 Sections 5.5.6, 5.6.3.1"

INDEX { pnniNodeIndex,
pnniSvccRccIndex }

::= { pnniSvccRccTable 1 }

PnniSvccRccEntry ::=

```
SEQUENCE {
    pnniSvccRccIndex          PnniSvccRccIndex,
    pnniSvccRccVersion        PnniVersion,
    pnniSvccRccHelloState     PnniHelloState,
    pnniSvccRccRemoteNodeId   PnniNodeId,
    pnniSvccRccRemoteAtmAddress PnniAtmAddr,
    pnniSvccRccRcvHellos      Counter32,
    pnniSvccRccXmtHellos      Counter32,
    pnniSvccRccIfIndex        InterfaceIndex,
    pnniSvccRccVpi            INTEGER,
    pnniSvccRccVci            INTEGER
}
```

pnniSvccRccIndex OBJECT-TYPE

SYNTAX PnniSvccRccIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An index into the node's tables of SVCC-based RCCs."

::= { pnniSvccRccEntry 1 }

pnniSvccRccVersion OBJECT-TYPE

SYNTAX PnniVersion

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The version of the PNNI routing protocol used to exchange information with the neighbor node."

::= { pnniSvccRccEntry 2 }

pnniSvccRccHelloState OBJECT-TYPE

SYNTAX PnniHelloState

MAX-ACCESS read-only

STATUS current
DESCRIPTION
"The state of the Hello protocol exchange over the SVCC-based RCC.

Note: the Down state indicates that the SVCC establishment is in progress."

::= { pnniSvccRccEntry 3 }

pnniSvccRccRemoteNodeId OBJECT-TYPE

SYNTAX PnniNodeId
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"The remote node at which the SVCC-based RCC terminates."

::= { pnniSvccRccEntry 4 }

pnniSvccRccRemoteAtmAddress OBJECT-TYPE

SYNTAX PnniAtmAddr
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"The ATM End System Address to which SVCC establishment is attempted."

::= { pnniSvccRccEntry 5 }

pnniSvccRccRcvHellos OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"A count of the number of Hello Packets received over this SVCC-based RCC."

::= { pnniSvccRccEntry 6 }

pnniSvccRccXmtHellos OBJECT-TYPE

SYNTAX Counter32
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"A count of the number of Hello Packets transmitted over this SVCC-based RCC."

::= { pnniSvccRccEntry 7 }

pnniSvccRccIfIndex OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The interface from which the SVCC-based RCC leaves the switching system. If the SVCC-based RCC has not yet been established, then this attribute takes the value of zero."

::= { pnniSvccRccEntry 8 }

pnniSvccRccVpi OBJECT-TYPE

SYNTAX INTEGER (0..4095)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The VPI used at the interface from which the SVCC-based RCC leaves the switching system. If the SVCC-based RCC has not yet been established, then this attribute takes the value of zero "

::= { pnniSvccRccEntry 9 }

pnniSvccRccVci OBJECT-TYPE

SYNTAX INTEGER (0..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The VCI used at the interface from which the SVCC-based RCC leaves the switching system. If the SVCC-based RCC has not yet been established, then this attribute takes the value of zero "

::= { pnniSvccRccEntry 10 }

-- PTSE table

pnniPtseTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniPtseEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The pnniPtse object contains the attributes that describe the most recent instances of PTSEs in a node's topology database. A concatenation of the Node Identifier of the local node that received the PTSE, the originating Node's

Node Identifier and the PTSE Identifier are used to form the instance ID for an instance of this object."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.2"

::= { pnniMIBObjects 13 }

pnniPtseEntry OBJECT-TYPE

SYNTAX PnniPtseEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table, containing information about a PTSE in the topology database of a PNNI logical node in this switching system."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.2"

INDEX { pnniNodeIndex,
pnniPtseOriginatingNodeId,
pnniPtseId }

::= { pnniPtseTable 1 }

PnniPtseEntry ::=

SEQUENCE {

pnniPtseOriginatingNodeId	PnniNodeId,
pnniPtseId	Unsigned32,
pnniPtseType	INTEGER,
pnniPtseSequenceNum	Unsigned32,
pnniPtseChecksum	Unsigned32,
pnniPtseLifeTime	Unsigned32,
pnniPtseInfo	OCTET STRING
}	

pnniPtseOriginatingNodeId OBJECT-TYPE

SYNTAX PnniNodeId

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The Node Identifier of the node that originated the PTSE."

::= { pnniPtseEntry 1 }

pnniPtseId OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The value of the PTSE Identifier assigned to the PTSE by its originator."

::= { pnniPtseEntry 2 }

pnniPtseType OBJECT-TYPE

SYNTAX INTEGER {
 other(1),
 nodalStateParameters(96),
 nodalInformation(97),
 internalReachableAddresses(224),
 exteriorReachableAddresses(256),
 horizontalLinks(288),
 uplinks(289)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The type of information contained in the PTSE."

::= { pnniPtseEntry 3 }

pnniPtseSequenceNum OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The sequence number of the instance of the PTSE as it appears in the local topology database."

::= { pnniPtseEntry 4 }

pnniPtseChecksum OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of the PTSE checksum as it appears in the local topology database."

::= { pnniPtseEntry 5 }

pnniPtseLifeTime OBJECT-TYPE

SYNTAX Unsigned32 (0..65535)

UNITS "seconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of the remaining lifetime for the given PTSE as it appears in the local topology database."

::= { pnniPtseEntry 6 }

pnniPtseInfo OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(0..65535))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"An unformatted hexadecimal dump of the PTSE contents in full.

Note: If the size of the PTSE contents is larger than the maximum size of SNMP packets then this is truncated."

::= { pnniPtseEntry 7 }

-- pnni map table

pnniMapTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniMapEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table containing attributes necessary to find and analyze the operation of all links and nodes within the PNNI hierarchy, as seen from the perspective of a local node. An instance of a pnniMap Object describes a link in terms of a node at one end of the link. Normally there will be two instances of the pnniMap object in the MIB for each horizontal link. The two instances provide information for Network management to map port identifiers from the nodes at both ends to the link between them. A concatenation of the Local Node Index, Originating Node Identifier and Originating Port Identifier are used to form the instance ID for this object.

This entire object is read-only, reflecting the fact that the map is discovered dynamically during operation of the PNNI protocol rather than configured."

::= { pnniMIBObjects 14 }

pnniMapEntry OBJECT-TYPE

```

SYNTAX      PnniMapEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "An entry in the table, containing connectivity information
    about a node or link in the PNNI routing domain, as seen
    from the perspective of a PNNI logical node in this
    switching system."
INDEX       { pnniNodeIndex,
              pnniMapOriginatingNodeId,
              pnniMapOriginatingPortId,
              pnniMapIndex }
 ::= { pnniMapTable 1 }

```

```

PnniMapEntry ::=
SEQUENCE {
    pnniMapOriginatingNodeId      PnniNodeId,
    pnniMapOriginatingPortId      PnniPortId,
    pnniMapIndex                   INTEGER,
    pnniMapType                     INTEGER,
    pnniMapPeerGroupId              PnniPeerGroupId,
    pnniMapAggrToken                PnniAggrToken,
    pnniMapRemoteNodeId             PnniNodeId,
    pnniMapRemotePortId            PnniPortId,
    pnniMapVPCapability             TruthValue,
    pnniMapPtseId                   Unsigned32,
    pnniMapMetricsTag               PnniMetricsTag
}

```

pnniMapOriginatingNodeId OBJECT-TYPE

```

SYNTAX      PnniNodeId
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "The node identifier of the node whose connectivity within
    itself or to other nodes is being described."
 ::= { pnniMapEntry 1 }

```

pnniMapOriginatingPortId OBJECT-TYPE

```

SYNTAX      PnniPortId
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "The port identifier of the port as assigned by the

```

originating node, to which the port is attached."
 ::= { pnniMapEntry 2 }

pnniMapIndex OBJECT-TYPE

SYNTAX INTEGER (0..65535)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An index into the set of link and nodal connectivity associated with the originating node and port. This index is needed since there may be multiple entries for nodal connectivity from a specific node and port pair, in addition to any entry for a horizontal link or uplink."

::= { pnniMapEntry 3 }

pnniMapType OBJECT-TYPE

SYNTAX INTEGER {
 horizontalLink(1),
 uplink(2),
 node(3)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The type of PNNI entity being described by this entry in the table."

::= { pnniMapEntry 4 }

pnniMapPeerGroupId OBJECT-TYPE

SYNTAX PnniPeerGroupId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Identifies the peer group of the originating node."

::= { pnniMapEntry 5 }

pnniMapAggrToken OBJECT-TYPE

SYNTAX PnniAggrToken

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"For horizontal links to/from LGNs and for uplinks, this attribute contains the derived aggregation token value for this link. For nodes and for horizontal links between

lowest-level nodes, this attribute is set to zero."
 ::= { pnniMapEntry 6 }

pnniMapRemoteNodeId OBJECT-TYPE

SYNTAX PnniNodeId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"For horizontal links and uplinks, this attribute contains the node identifier of the node at the other end of the link from the originating node. If unknown, the PNNI protocol entity sets this attribute's value to (all) zero(s). For nodes, this attribute's value is set to (all) zero(s)."

::= { pnniMapEntry 7 }

pnniMapRemotePortId OBJECT-TYPE

SYNTAX PnniPortId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"For horizontal links and uplinks, this attribute contains the port identifier of the port at the remote end of the link as assigned by the remote node. If unknown, the PNNI protocol entity sets this attribute's value to zero.

For nodes, this attribute contains the port identifier of the port at the other end of the spoke or bypass from the originating port. When the originating port ID is zero, a value of zero indicates the default radius. When the originating port ID is non-zero, a value of zero indicates the nodal nucleus."

::= { pnniMapEntry 8 }

pnniMapVPCapability OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates whether VPCs can be established across the PNNI entity being described by this entry in the pnniMapTable."

::= { pnniMapEntry 9 }

pnniMapPtseId OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of the PTSE Identifier for the PTSE being originated by the originating node which contains the information group(s) describing the PNNI entity."

::= { pnniMapEntry 10 }

pnniMapMetricsTag OBJECT-TYPE

SYNTAX PnniMetricsTag (1..MAX)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"An arbitrary integer that is used to associate a set of traffic parameters that are always advertised together. Within this set, the parameters are distinguished by the service categories and direction to which a set of parameters apply."

::= { pnniMapEntry 11 }

-- nodal map table

pnniMapNodeTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniMapNodeEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A list of nodes as seen from the perspective of a local node. The pnniMapNodeTable contains all information learned by the local node from nodal information PTSEs. This entire object is read-only, reflecting the fact that the map is discovered dynamically during operation of the PNNI protocol rather than configured."

::= { pnniMIBObjects 15 }

pnniMapNodeEntry OBJECT-TYPE

SYNTAX PnniMapNodeEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table, containing information about a node

in the PNNI routing domain, as seen from the perspective of a logical node in this switching system."

```
INDEX      { pnniNodeIndex,
             pnniMapNodeId }
 ::= { pnniMapNodeTable 1 }
```

PnniMapNodeEntry ::=

```
SEQUENCE {
    pnniMapNodeId          PnniNodeId,
    pnniMapNodePeerGroupId PnniPeerGroupId,
    pnniMapNodeAtmAddress  PnniAtmAddr,
    pnniMapNodeRestrictedTransit TruthValue,
    pnniMapNodeComplexRep  TruthValue,
    pnniMapNodeRestrictedBranching TruthValue,
    pnniMapNodeDatabaseOverload TruthValue,
    pnniMapNodeIAMLeader   TruthValue,
    pnniMapNodeLeadershipPriority INTEGER,
    pnniMapNodePreferredPgl PnniNodeId,
    pnniMapNodeParentNodeId PnniNodeId,
    pnniMapNodeParentAtmAddress PnniAtmAddr,
    pnniMapNodeParentPeerGroupId PnniPeerGroupId,
    pnniMapNodeParentPglNodeId PnniNodeId
}
```

pnniMapNodeId OBJECT-TYPE

```
SYNTAX      PnniNodeId
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
```

"Identifies the node whose nodal information is being described."

```
::= { pnniMapNodeEntry 1 }
```

pnniMapNodePeerGroupId OBJECT-TYPE

```
SYNTAX      PnniPeerGroupId
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
```

"Identifies the peer group of the originating node."

```
::= { pnniMapNodeEntry 2 }
```

pnniMapNodeAtmAddress OBJECT-TYPE

```
SYNTAX      PnniAtmAddr
MAX-ACCESS  read-only
```

STATUS current

DESCRIPTION

"The ATM End System Address of the originating node."

::= { pnniMapNodeEntry 3 }

pnniMapNodeRestrictedTransit OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates whether the originating node is restricted to only allow support of SVCs originating or terminating at this node. A value of 'true' indicates that the transit capabilities are restricted, i.e., transit connections are not allowed, whereas a value of 'false' indicates that transit connections are allowed. This attribute reflects the setting of the restricted transit bit received in the nodal information PTSE of the originating node."

::= { pnniMapNodeEntry 4 }

pnniMapNodeComplexRep OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates whether the originating node uses the complex node representation. If the value is 'true', the spokes and bypasses that make up the complex node representation should be found in the pnniMapTable. This attribute reflects the setting of the nodal representation bit received in the nodal information PTSE of the originating node."

::= { pnniMapNodeEntry 5 }

pnniMapNodeRestrictedBranching OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates whether the originating node is able to support additional branches. If the value is 'false', then it can support additional branches. This attribute reflects the setting of the restricted branching bit received in the nodal information PTSE of the originating node."


```
::= { pnniMapNodeEntry 6 }
```

pnniMapNodeDatabaseOverload OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates whether the originating node is currently operating in topology database overload state. This attribute has the same value as the Non-transit for PGL Election bit in the nodal information group originated by this node."

```
::= { pnniMapNodeEntry 7 }
```

pnniMapNodeIamLeader OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates whether the originating node claims to be peer group leader of its peer group. This attribute reflects the setting of the 'I am Leader' bit received in the nodal information PTSE of the originating node."

```
::= { pnniMapNodeEntry 8 }
```

pnniMapNodeLeadershipPriority OBJECT-TYPE

SYNTAX INTEGER (0..255)

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The Leadership priority value advertised by the originating node."

```
::= { pnniMapNodeEntry 9 }
```

pnniMapNodePreferredPgl OBJECT-TYPE

SYNTAX PnniNodeId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Identifies the node which the originating node believes should be or is peer group leader of its peer group. If the originating node has not chosen a Preferred PGL, this attribute's value is set to (all) zero(s)."

```
::= { pnniMapNodeEntry 10 }
```

pnniMapNodeParentNodeId OBJECT-TYPE

SYNTAX PnniNodeId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"When the originating node is a peer group leader, indicates the node ID of the parent LGN. If the originating node is not peer group leader of its peer group, this attribute's value is set to (all) zero(s)."

::= { pnniMapNodeEntry 11 }

pnniMapNodeParentAtmAddress OBJECT-TYPE

SYNTAX PnniAtmAddr

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"When the originating node is a peer group leader, indicates the ATM address of the parent LGN. If the originating node is not peer group leader of its peer group, this attribute's value is set to (all) zero(s)."

::= { pnniMapNodeEntry 12 }

pnniMapNodeParentPeerGroupId OBJECT-TYPE

SYNTAX PnniPeerGroupId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"When the originating node is a peer group leader, indicates the node's parent peer group ID. If the originating node is not peer group leader of its peer group, this attribute's value is set to (all) zero(s)."

::= { pnniMapNodeEntry 13 }

pnniMapNodeParentPglNodeId OBJECT-TYPE

SYNTAX PnniNodeId

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"When the originating node is a peer group leader, identifies the node elected as peer group leader of the parent peer group. If the originating node is not peer group leader of its peer group, this attribute's value is set to (all) zero(s)."

```
 ::= { pnniMapNodeEntry 14 }
```

```
-- address map table
```

```
pnniMapAddrTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF PnniMapAddrEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

"The pnniMapAddr MIB Object contains a list of all reachable addresses from each node visible to the local node. The Local Node Index, Advertising Node ID, Advertised Port ID, Reachable Address, and Address prefix length are combined to form an instance ID for this object. The entire object is read-only, reflecting the fact that reachable addresses are discovered during dynamic operation of the PNNI protocol rather than configured."

```
 ::= { pnniMIBObjects 16 }
```

```
pnniMapAddrEntry OBJECT-TYPE
```

```
SYNTAX PnniMapAddrEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

"An entry in the table, containing information about an address prefix reachable from a node in the PNNI routing domain, as seen from the perspective of a PNNI logical node in this switching system."

```
INDEX { pnniNodeIndex,
        pnniMapAddrAdvertisingNodeId,
        pnniMapAddrAdvertisedPortId,
        pnniMapAddrIndex }
```

```
 ::= { pnniMapAddrTable 1 }
```

```
PnniMapAddrEntry ::=
```

```
SEQUENCE {
```

```
    pnniMapAddrAdvertisingNodeId PnniNodeId,
    pnniMapAddrAdvertisedPortId PnniPortId,
    pnniMapAddrIndex             INTEGER,
    pnniMapAddrAddress           AtmAddrPrefix,
    pnniMapAddrPrefixLength     PnniPrefixLength
}
```

pnniMapAddrAdvertisingNodeId OBJECT-TYPE

SYNTAX PnniNodeId
MAX-ACCESS not-accessible
STATUS current

DESCRIPTION

"The node ID of a node advertising reachability to the address prefix."

::= { pnniMapAddrEntry 1 }

pnniMapAddrAdvertisedPortId OBJECT-TYPE

SYNTAX PnniPortId
MAX-ACCESS not-accessible
STATUS current

DESCRIPTION

"The port identifier used from the advertising node to reach the given address prefix."

::= { pnniMapAddrEntry 2 }

pnniMapAddrIndex OBJECT-TYPE

SYNTAX INTEGER (1..MAX)
MAX-ACCESS not-accessible
STATUS current

DESCRIPTION

"An arbitrary index that is used to enumerate all of the addresses advertised by the specified node."

::= { pnniMapAddrEntry 3 }

pnniMapAddrAddress OBJECT-TYPE

SYNTAX AtmAddrPrefix
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"The value of the ATM End System Address prefix."

::= { pnniMapAddrEntry 4 }

pnniMapAddrPrefixLength OBJECT-TYPE

SYNTAX PnniPrefixLength
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"The Prefix length to be applied to the ATM End System Address prefix."

::= { pnniMapAddrEntry 5 }

-- TNS map table

pnniMapTnsTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniMapTnsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A list of all reachable transit networks from each node visible to the local node. The Local Node Index, Advertising Node ID, Advertised Port ID, Transit Network Type, Transit Network Plan, and Transit Network ID are combined to form an instance ID for this object. The entire object is read-only, reflecting the fact that reachable transit networks are discovered during dynamic operation of the PNNI protocol rather than configured.."

::= { pnniMIBObjects 17 }

pnniMapTnsEntry OBJECT-TYPE

SYNTAX PnniMapTnsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table, containing information about a transit network reachable from a node in the PNNI routing domain, as seen from the perspective of a PNNI logical node in this switching system."

INDEX { pnniNodeIndex,
pnniMapTnsAdvertisingNodeId,
pnniMapTnsAdvertisedPortId,
pnniMapTnsType,
pnniMapTnsPlan,
pnniMapTnsId }

::= { pnniMapTnsTable 1 }

PnniMapTnsEntry ::=

SEQUENCE {

pnniMapTnsAdvertisingNodeId PnniNodeId,

pnniMapTnsAdvertisedPortId PnniPortId,

pnniMapTnsType TnsType,

pnniMapTnsPlan TnsPlan,

pnniMapTnsId DisplayString

}

pnniMapTnsAdvertisingNodeId OBJECT-TYPE

SYNTAX PnniNodeId

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The node ID of a node advertising reachability to the transit network."

::= { pnniMapTnsEntry 1 }

pnniMapTnsAdvertisedPortId OBJECT-TYPE

SYNTAX PnniPortId

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The port identifier used from the advertising node to reach the given transit network."

::= { pnniMapTnsEntry 2 }

pnniMapTnsType OBJECT-TYPE

SYNTAX TnsType

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The type of network identification used for this transit network."

::= { pnniMapTnsEntry 3 }

pnniMapTnsPlan OBJECT-TYPE

SYNTAX TnsPlan

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The network identification plan according to which network identification has been assigned."

::= { pnniMapTnsEntry 4 }

pnniMapTnsId OBJECT-TYPE

SYNTAX DisplayString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of the transit network identifier."

::= { pnniMapTnsEntry 5 }

-- pnni metrics table

pnniMetricsTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniMetricsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This entity's table of PNNI parameters either associated with a PNNI entity or for the connectivity between a PNNI node and a reachable address or transit network."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3"

::= { pnniMIBObjects 18 }

pnniMetricsEntry OBJECT-TYPE

SYNTAX PnniMetricsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A set of parameters that applies to the connectivity from a certain node and port to another node or port or to one or more reachable address prefixes and/or transit networks, for one (or more) particular service category(s). Note that there can be multiple sets of parameters with the same tag, in which case all sets apply to the specified connectivity."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3"

INDEX { pnniNodeIndex,
pnniMetricsTag,
pnniMetricsDirection,
pnniMetricsIndex }

::= { pnniMetricsTable 1 }

PnniMetricsEntry ::=

SEQUENCE {

pnniMetricsTag PnniMetricsTag,

pnniMetricsDirection INTEGER,

pnniMetricsIndex Integer32,

pnniMetricsClasses INTEGER,

pnniMetricsGcacClp ClpType,

pnniMetricsAdminWeight Unsigned32,

```

pnniMetrics1      Unsigned32,
pnniMetrics2      Unsigned32,
pnniMetrics3      Unsigned32,
pnniMetrics4      Unsigned32,
pnniMetrics5      Unsigned32,
pnniMetrics6      Unsigned32,
pnniMetrics7      Unsigned32,
pnniMetrics8      Unsigned32,
pnniMetricsRowStatus RowStatus
}

```

pnniMetricsTag OBJECT-TYPE

```

SYNTAX      PnniMetricsTag (1..MAX)
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION

```

"An arbitrary integer that is used to associate a set of traffic parameters that are always advertised together. Within this set, the parameters are distinguished by the service categories and direction to which a set of parameters apply."

```
 ::= { pnniMetricsEntry 1 }
```

pnniMetricsDirection OBJECT-TYPE

```

SYNTAX      INTEGER { incoming(1), outgoing(2) }
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION

```

"The direction, with respect to the advertising node, in which the parameters in this entry apply."

```
 ::= { pnniMetricsEntry 2 }
```

pnniMetricsIndex OBJECT-TYPE

```

SYNTAX      Integer32 (1..MAX)
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION

```

"An index into the set of parameters associated with the given tag and direction."

```
 ::= { pnniMetricsEntry 3 }
```


pnniMetricsClasses OBJECT-TYPE

SYNTAX INTEGER(0..31)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The service categories to which this set of parameters applies. This is an integer used as a bit mask with each bit that is set representing a single service category for which the resources indicated are available. Bit 5 represents CBR, bit 4 represents real-time VBR, bit 3 represents non-real-time VBR, bit 2 represents ABR, and bit 1 (LSB) represents UBR."

REFERENCE

"ATM Forum Traffic Management 4.0 Section 2,
ATM Forum PNNI 1.0 Section 5.8.1.1.3.1"

```
::= { pnniMetricsEntry 4 }
```

pnniMetricsGcacClp OBJECT-TYPE

SYNTAX ClpType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Indicates whether the advertised GCAC parameters apply for CLP=0 traffic or for CLP=0+1 traffic."

REFERENCE

"ATM Forum PNNI 1.0 Sections 5.8.1.1.3.1, 5.13.4.1"

```
::= { pnniMetricsEntry 5 }
```

pnniMetricsAdminWeight OBJECT-TYPE

SYNTAX Unsigned32 (1..MAX)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The administrative weight from the advertising node to the remote end of the PNNI entity or to the reachable address or transit network, for the specified service categories.

If this metric is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.4"

```
DEFVAL { 5040 }
```

```
::= { pnniMetricsEntry 6 }
```

pnniMetrics1 OBJECT-TYPE

```
SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"An alternate routing parameter from the advertising node to the remote end of the PNNI entity or to the reachable address or transit network, for the specified service categories.

For information learned from PNNI nodes, this is the maximum cell rate in cells per second for the specified service categories.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.7"

```
DEFVAL { 'FFFFFFFF'h }
```

```
::= { pnniMetricsEntry 7 }
```

pnniMetrics2 OBJECT-TYPE

```
SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"An alternate routing parameter from the advertising node to the remote end of the PNNI entity or to the reachable address or transit network, for the specified service categories.

For information learned from PNNI nodes, this is the available cell rate in cells per second for the specified service categories.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.8"

```
DEFVAL { 'FFFFFFFF'h }
```

```
::= { pnniMetricsEntry 8 }
```

pnniMetrics3 OBJECT-TYPE

```
SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"An alternate routing parameter from the advertising node to the remote end of the PNNI entity or to the reachable address or transit network, for the specified service categories.

For information learned from PNNI nodes, this is the maximum cell transfer delay in microseconds for the specified service categories.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.3"

```
DEFVAL { 'FFFFFFFF'h }
```

```
::= { pnniMetricsEntry 9 }
```

pnniMetrics4 OBJECT-TYPE

```
SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"An alternate routing parameter from the advertising node to the remote end of the PNNI entity or to the reachable address or transit network, for the specified service categories.

For information learned from PNNI nodes, this is the cell delay variation in microseconds for the specified service categories.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.2"

```
DEFVAL { 'FFFFFFFF'h }
```

```
::= { pnniMetricsEntry 10 }
```

pnniMetrics5 OBJECT-TYPE

```
SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"An alternate routing parameter from the advertising node to the remote end of the PNNI entity or to the reachable address or transit network, for the specified service categories.

For PNNI, this is the cell loss ratio for CLP=0 traffic for the specified service categories. The cell loss ratio value is computed as $10^{*(-n)}$ where 'n' is the value returned in this variable.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.5"

```
DEFVAL { 'FFFFFFFF'h }
```

```
::= { pnniMetricsEntry 11 }
```

pnniMetrics6 OBJECT-TYPE

```
SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"An alternate routing parameter from the advertising node to the remote end of the PNNI entity or to the reachable address or transit network, for the specified service categories.

For PNNI, this is the cell loss ratio for CLP=0+1 traffic for the specified service categories. The cell loss ratio value is computed as $10^{*(-n)}$ where 'n' is the value returned in this variable.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.6"

```
DEFVAL { 'FFFFFFFF'h }
::= { pnniMetricsEntry 12 }
```

pnniMetrics7 OBJECT-TYPE

```
SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"An alternate routing parameter from the advertising node to the remote end of the PNNI entity or to the reachable address or transit network, for the specified service categories.

For information learned from PNNI nodes, this is the cell rate margin in cells per second for the specified service categories.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.9"

```
DEFVAL { 'FFFFFFFF'h }
::= { pnniMetricsEntry 13 }
```

pnniMetrics8 OBJECT-TYPE

```
SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"An alternate routing parameter from the advertising node to the remote end of the PNNI entity or to the reachable address or transit network, for the specified service categories.

For information learned from PNNI nodes, this is the variance factor in units of 2⁽⁻⁸⁾ for the specified service categories.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.10"

```
DEFVAL { 'FFFFFFFF'h }
::= { pnniMetricsEntry 14 }
```

```
pnniMetricsRowStatus OBJECT-TYPE
```

```
SYNTAX      RowStatus
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
```

```
    "To create, delete, activate and de-activate a set of
    metrics."
```

```
::= { pnniMetricsEntry 15 }
```

```
--
```

```
-- PNNI Routing Tables
```

```
--
```

```
pnniRoutingGroup OBJECT IDENTIFIER ::= { pnniMIBObjects 19 }
```

```
pnniRouteBaseGroup OBJECT IDENTIFIER ::= { pnniRoutingGroup 1 }
```

```
pnniRouteNodeNumber OBJECT-TYPE
```

```
SYNTAX      Gauge32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
```

```
    "The number of current precalculated PNNI routes to PNNI
    nodes that are not invalid."
```

```
::= { pnniRouteBaseGroup 1 }
```

```
pnniRouteAddrNumber OBJECT-TYPE
```

```
SYNTAX      Gauge32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
```

```
    "The number of current PNNI routes from nodes in the PNNI
    routing domain to addresses and transit networks that are
    not invalid."
```

```
::= { pnniRouteBaseGroup 2 }
```

```
-- Table of routes to other nodes
```

pnniRouteNodeTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniRouteNodeEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This entity's PNNI Routing table (of routes to other nodes)."

```
::= { pnniRoutingGroup 2 }
```

pnniRouteNodeEntry OBJECT-TYPE

SYNTAX PnniRouteNodeEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A particular route to a particular destination node, under a particular policy."

```
INDEX { pnniNodeIndex,
        pnniRouteNodeClass,
        pnniRouteNodeDestNodeId,
        pnniRouteNodeDTL }
```

```
::= { pnniRouteNodeTable 1 }
```

PnniRouteNodeEntry ::=

SEQUENCE {

pnniRouteNodeClass	ServiceCategory,
pnniRouteNodeDestNodeId	PnniNodeId,
pnniRouteNodeDTL	Integer32,
pnniRouteNodeDestPortId	PnniPortId,
pnniRouteNodeProto	INTEGER,
pnniRouteNodeTimeStamp	TimeStamp,
pnniRouteNodeInfo	OBJECT IDENTIFIER,
pnniRouteNodeGcacClp	ClpType,
pnniRouteNodeFwdMetricAW	Unsigned32,
pnniRouteNodeFwdMetric1	Unsigned32,
pnniRouteNodeFwdMetric2	Unsigned32,
pnniRouteNodeFwdMetric3	Unsigned32,
pnniRouteNodeFwdMetric4	Unsigned32,
pnniRouteNodeFwdMetric5	Unsigned32,
pnniRouteNodeFwdMetric6	Unsigned32,
pnniRouteNodeFwdMetric7	Unsigned32,
pnniRouteNodeFwdMetric8	Unsigned32,
pnniRouteNodeBwdMetricAW	Unsigned32,
pnniRouteNodeBwdMetric1	Unsigned32,
pnniRouteNodeBwdMetric2	Unsigned32,

```

pnniRouteNodeBwdMetric3    Unsigned32,
pnniRouteNodeBwdMetric4    Unsigned32,
pnniRouteNodeBwdMetric5    Unsigned32,
pnniRouteNodeBwdMetric6    Unsigned32,
pnniRouteNodeBwdMetric7    Unsigned32,
pnniRouteNodeBwdMetric8    Unsigned32,
pnniRouteNodeVPCapability  TruthValue,
pnniRouteNodeStatus        RowStatus
}

```

pnniRouteNodeClass OBJECT-TYPE

SYNTAX ServiceCategory

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Indicates the service category with which this forwarding table entry is associated."

::= { pnniRouteNodeEntry 1 }

pnniRouteNodeDestNodeId OBJECT-TYPE

SYNTAX PnniNodeId

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The node ID of the destination node to which this route proceeds, and at which the DTL stack for this route terminates."

::= { pnniRouteNodeEntry 2 }

pnniRouteNodeDTL OBJECT-TYPE

SYNTAX Integer32 (1..MAX)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The index into the owning PNNI node's DTL table of the DTL stack that goes with this route."

::= { pnniRouteNodeEntry 3 }

pnniRouteNodeDestPortId OBJECT-TYPE

SYNTAX PnniPortId

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The port ID of the destination node at which the route

terminates. A port ID of zero indicates the node nucleus. When the destination node is represented by the simple node representation, this value should be set to zero."

```
DEFVAL { 0 }
 ::= { pnniRouteNodeEntry 4 }
```

pnniRouteNodeProto OBJECT-TYPE

```
SYNTAX      INTEGER {
                other(1), -- not specified
                local(2), -- e.g. ilmi
                mgmt(3), -- configured by management,
                        -- for example by SNMP or console
                        -- the following are all dynamic
                        -- routing protocols
                pnni(4) -- ATM Forum PNNI
            }
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The routing mechanism via which this route was learned."

```
 ::= { pnniRouteNodeEntry 5 }
```

pnniRouteNodeTimeStamp OBJECT-TYPE

```
SYNTAX      TimeStamp
```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The time at which this route was last updated or otherwise determined to be correct. Note that no semantics of 'too old' can be implied except through knowledge of the routing protocol by which the route was learned."

```
 ::= { pnniRouteNodeEntry 6 }
```

pnniRouteNodeInfo OBJECT-TYPE

```
SYNTAX      OBJECT IDENTIFIER
```

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"A reference to MIB definitions specific to the particular routing protocol which is responsible for this route, as determined by the value specified in the route's pnniRouteNodeProto value. If this information is not

present, its value should be set to the OBJECT IDENTIFIER zeroDotZero."

```
DEFVAL { zeroDotZero }
 ::= { pnniRouteNodeEntry 7 }
```

pnniRouteNodeGcacClp OBJECT-TYPE

```
SYNTAX      ClpType
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
```

"For PNNI, indicates whether any advertised GCAC parameters apply for CLP=0 traffic or for CLP=0+1 traffic."

```
 ::= { pnniRouteNodeEntry 8 }
```

pnniRouteNodeFwdMetricAW OBJECT-TYPE

```
SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
```

"The cumulative administrative weight calculated for the forward direction of this route. If this metric is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.4"

```
DEFVAL { 'FFFFFFFF'h }
 ::= { pnniRouteNodeEntry 9 }
```

pnniRouteNodeFwdMetric1 OBJECT-TYPE

```
SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
```

"An alternate routing parameter for the forward direction of this route.

For information learned from PNNI nodes, this is the maximum possible cell rate (in cells per second) for the forward direction of the route.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.7"

```
DEFVAL { 'FFFFFFFF'h }
```

```
::= { pnniRouteNodeEntry 10 }
```

pnniRouteNodeFwdMetric2 OBJECT-TYPE

```
SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"An alternate routing parameter for the forward direction of this route.

For information learned from PNNI nodes, this is the Available cell rate (in cells per second) for the forward direction of the route. Further information on available bandwidth may be obtainable by reference to the nodal advertisements of the nodes in the path.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.8"

```
DEFVAL { 'FFFFFFFF'h }
```

```
::= { pnniRouteNodeEntry 11 }
```

pnniRouteNodeFwdMetric3 OBJECT-TYPE

```
SYNTAX      Unsigned32
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"An alternate routing parameter for the forward direction of this route.

For information learned from PNNI nodes, this is the cumulative Maximum Cell Transfer Delay (in microseconds) for the forward direction of the route.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.3"

```
DEFVAL { 'FFFFFFFF'h }
```

```
::= { pnniRouteNodeEntry 12 }
```

pnniRouteNodeFwdMetric4 OBJECT-TYPE

```
SYNTAX      Unsigned32
```

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An alternate routing parameter for the forward direction of this route.

For information learned from PNNI nodes, this is the cumulative Cell Delay Variation (in microseconds) for the forward direction of the route.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.2"

DEFVAL { 'FFFFFFFF'h }

::= { pnniRouteNodeEntry 13 }

pnniRouteNodeFwdMetric5 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An alternate routing parameter for the forward direction of this route.

For information learned from PNNI nodes, this is the cumulative Cell Loss Ratio for CLP=0 traffic for the forward direction of the route. The cell loss ratio value is computed as $10^{*(-n)}$ where 'n' is the value returned in this variable.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.5"

DEFVAL { 'FFFFFFFF'h }

::= { pnniRouteNodeEntry 14 }

pnniRouteNodeFwdMetric6 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An alternate routing parameter for the forward direction of

this route.

For information learned from PNNI nodes, this is the cumulative Cell Loss Ratio for CLP=0+1 traffic for the forward direction of the route. The cell loss ratio value is computed as $10^{*(-n)}$ where 'n' is the value returned in this variable.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.6"

DEFVAL { 'FFFFFFFF'h }

::= { pnniRouteNodeEntry 15 }

pnniRouteNodeFwdMetric7 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An alternate routing parameter for the forward direction of this route.

For information learned from PNNI nodes, this is the Cell Rate Margin (in cells per second) for the forward direction of the route.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.9"

DEFVAL { 'FFFFFFFF'h }

::= { pnniRouteNodeEntry 16 }

pnniRouteNodeFwdMetric8 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An alternate routing parameter for the forward direction of this route.

For information learned from PNNI nodes, this is the

Variance Factor (in units of $2^{*(-8)}$) for the forward direction of the route.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.10"

DEFVAL { 'FFFFFFFF'h }

::= { pnniRouteNodeEntry 17 }

pnniRouteNodeBwdMetricAW OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The administrative weight calculated for the backward direction of this route. If this metric is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.4"

DEFVAL { 'FFFFFFFF'h }

::= { pnniRouteNodeEntry 18 }

pnniRouteNodeBwdMetric1 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An alternate routing parameter for the backward direction of this route.

For information learned from PNNI nodes, this is the maximum possible cell rate (in cells per second) for the backward direction of the route.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.7"

DEFVAL { 'FFFFFFFF'h }

::= { pnniRouteNodeEntry 19 }

pnniRouteNodeBwdMetric2 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An alternate routing parameter for the backward direction of this route.

For information learned from PNNI nodes, this is the Available cell rate (in cells per second) for the backward direction of the route. Further information on available bandwidth may be obtainable by reference to the nodal advertisements of the nodes in the path.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.8"

DEFVAL { 'FFFFFFFF'h }

::= { pnniRouteNodeEntry 20 }

pnniRouteNodeBwdMetric3 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An alternate routing parameter for the backward direction of this route.

For information learned from PNNI nodes, this is the cumulative Maximum Cell Transfer Delay (in microseconds) for the backward direction of the route.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.3"

DEFVAL { 'FFFFFFFF'h }

::= { pnniRouteNodeEntry 21 }

pnniRouteNodeBwdMetric4 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An alternate routing parameter for the backward direction

of this route.

For information learned from PNNI nodes, this is the cumulative Cell Delay Variation (in microseconds) for the backward direction of the route.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.2"

DEFVAL { 'FFFFFFFF'h }

::= { pnniRouteNodeEntry 22 }

pnniRouteNodeBwdMetric5 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An alternate routing parameter for the backward direction of this route.

For information learned from PNNI nodes, this is the cumulative Cell Loss Ratio for CLP=0 traffic for the backward direction of the route. The cell loss ratio value is computed as $10^{*(-n)}$ where 'n' is the value returned in this variable.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.5"

DEFVAL { 'FFFFFFFF'h }

::= { pnniRouteNodeEntry 23 }

pnniRouteNodeBwdMetric6 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An alternate routing parameter for the backward direction of this route.

For information learned from PNNI nodes, this is the cumulative Cell Loss Ratio for CLP=0+1 traffic for the

backward direction of the route. The cell loss ratio value is computed as $10^{**(-n)}$ where 'n' is the value returned in this variable.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.6"

DEFVAL { 'FFFFFFFF'h }

::= { pnniRouteNodeEntry 24 }

pnniRouteNodeBwdMetric7 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An alternate routing parameter for the backward direction of this route.

For information learned from PNNI nodes, this is the Cell Rate Margin (in cells per second) for the backward direction of the route.

If this parameter is not used, its value should be set to 0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.9"

DEFVAL { 'FFFFFFFF'h }

::= { pnniRouteNodeEntry 25 }

pnniRouteNodeBwdMetric8 OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"An alternate routing parameter for the backward direction of this route.

For information learned from PNNI nodes, this is the Variance Factor (in units of $2^{*(-8)}$) for the backward direction of the route.

If this parameter is not used, its value should be set to

0xFFFFFFFF."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.1.3.10"

DEFVAL { 'FFFFFFFF'h }

::= { pnniRouteNodeEntry 26 }

pnniRouteNodeVPCapability OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This attribute indicates whether a VPC setup on this route is possible."

::= { pnniRouteNodeEntry 27 }

pnniRouteNodeStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The row status variable, used according to row installation and removal conventions."

::= { pnniRouteNodeEntry 28 }

-- Table of DTL stacks for routes to other nodes

pnniDTLTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniDTLEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The set of all DTL stacks used for the pre-computed routes maintained by this managed entity."

::= { pnniRoutingGroup 3 }

pnniDTLEntry OBJECT-TYPE

SYNTAX PnniDTLEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A segment of a DTL stack. The complete DTL stack is formed by traversing the rows of the table for which the pnniDTLIndex is the same. Level transitions are indicated

using the pnniDTLinkType column."

```
INDEX {
    pnniNodeIndex,
    pnniDTLIndex,
    pnniDTLEntryIndex
}
::= { pnniDTLTable 1 }
```

```
PnniDTLEntry ::=
SEQUENCE {
    pnniDTLIndex      Integer32,
    pnniDTLEntryIndex Integer32,
    pnniDTLNodeId     PnniNodeId,
    pnniDTLPortId     PnniPortId,
    pnniDTLLinkType   INTEGER,
    pnniDTLStatus     RowStatus
}
```

```
pnniDTLIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..65535)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The index in the node's DTL table of this DTL stack."
    ::= { pnniDTLEntry 1 }
```

```
pnniDTLEntryIndex OBJECT-TYPE
    SYNTAX      Integer32 (1..200)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The index in the current DTL stack of this entry."
    ::= { pnniDTLEntry 2 }
```

```
pnniDTLNodeId OBJECT-TYPE
    SYNTAX      PnniNodeId
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The node which is this hop in the DTL stack."
    ::= { pnniDTLEntry 3 }
```

```
pnniDTLPortId OBJECT-TYPE
    SYNTAX      PnniPortId
```

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The port from the pnniDTLNodeId to use as the exit. If the DTL stack does not care, this is coded as zero."

::= { pnniDTLEntry 4 }

pnniDTLLinkType OBJECT-TYPE

SYNTAX INTEGER {

invalid (1), -- An invalid link

horizontal (2), -- A normal link within
-- the containing peer group

uplink (3), -- A link going up a
-- level

last (4) -- The last entry in the
-- DTL stack

}

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The type of link out from this node (pnniDTLNodeId). This is well defined even if the specific port is not specified."

::= { pnniDTLEntry 5 }

pnniDTLStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The row status variable, used according to row installation and removal conventions."

::= { pnniDTLEntry 6 }

-- Table of routes from nodes to reachable addresses

pnniRouteAddrTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniRouteAddrEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table containing all the attributes necessary to determine what the PNNI entity believes is reachable in

terms of ATM End System Addresses and to determine which nodes are advertising this reachability. This table is also used to configure static routes to reachable address prefixes. The local node index that received the reachability information, reachable address, address prefix length, and an index that distinguishes between multiple listings of connectivity to a given address prefix from a given local node are combined to form an instance ID for this object.."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.3"

::= { pnniRoutingGroup 4 }

pnniRouteAddrEntry OBJECT-TYPE

SYNTAX PnniRouteAddrEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table, containing information about a reachable address prefix."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.3"

INDEX { pnniNodeIndex,
pnniRouteAddrAddress,
pnniRouteAddrPrefixLength,
pnniRouteAddrIndex }

::= { pnniRouteAddrTable 1 }

PnniRouteAddrEntry ::=

SEQUENCE {

pnniRouteAddrAddress	AtmAddrPrefix,
pnniRouteAddrPrefixLength	PnniPrefixLength,
pnniRouteAddrIndex	Integer32,
pnniRouteAddrIfIndex	InterfaceIndex,
pnniRouteAddrAdvertisingNodeId	PnniNodeId,
pnniRouteAddrAdvertisedPortId	PnniPortId,
pnniRouteAddrType	INTEGER,
pnniRouteAddrProto	INTEGER,
pnniRouteAddrPnniScope	PnniLevel,
pnniRouteAddrVPCapability	TruthValue,
pnniRouteAddrMetricsTag	PnniMetricsTag,
pnniRouteAddrPtseId	Unsigned32,
pnniRouteAddrOriginateAdvertisement	TruthValue,
pnniRouteAddrInfo	OBJECT IDENTIFIER,
pnniRouteAddrOperStatus	INTEGER,

```

        pnniRouteAddrTimeStamp      TimeStamp,
        pnniRouteAddrRowStatus      RowStatus
    }

```

pnniRouteAddrAddress OBJECT-TYPE

```

SYNTAX      AtmAddrPrefix
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "The value of the ATM End System Address prefix."
 ::= { pnniRouteAddrEntry 1 }

```

pnniRouteAddrPrefixLength OBJECT-TYPE

```

SYNTAX      PnniPrefixLength
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "The prefix length to be applied to the ATM End System
    Address prefix."
 ::= { pnniRouteAddrEntry 2 }

```

pnniRouteAddrIndex OBJECT-TYPE

```

SYNTAX      Integer32 (1..65535)
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "An index into the set of listings of connectivity to a
    given address prefix from a given local node."
 ::= { pnniRouteAddrEntry 3 }

```

pnniRouteAddrIfIndex OBJECT-TYPE

```

SYNTAX      InterfaceIndex
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
    "The local interface over which the reachable address can be
    reached. The value zero indicates an unknown interface or
    reachability through a remote node."

```

This object may only have a non-zero value if the value of the corresponding instance of pnniRouteAddrProto is other than 'pnni', pnniRouteAddrType is other than 'reject', and the node identified by pnniRouteAddrAdvertisingNodeId is instantiated within this switching system."

```

 ::= { pnniRouteAddrEntry 4 }

```

pnniRouteAddrAdvertisingNodeId OBJECT-TYPE

SYNTAX PnniNodeId
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION

"The node ID of a node advertising reachability to the address prefix. If the local node index is zero, then the advertising node ID must be set to all zeros."

::= { pnniRouteAddrEntry 5 }

pnniRouteAddrAdvertisedPortId OBJECT-TYPE

SYNTAX PnniPortId
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION

"The port identifier used from the advertising node to reach the given address prefix."

DEFVAL { 0 }

::= { pnniRouteAddrEntry 6 }

pnniRouteAddrType OBJECT-TYPE

SYNTAX INTEGER {
 other(1), -- not specified by this MIB
 reject(2), -- route which discards
 -- traffic
 internal(3),
 exterior(4)
 }

MAX-ACCESS read-create
 STATUS current
 DESCRIPTION

"The type (e.g. internal or exterior) of reachability from the advertising node to the address prefix.

Reject(2) refers to an address prefix which, if matched, indicates that the message should be discarded as unreachable. This is used in some protocols as a means of correctly aggregating routes."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.3"

DEFVAL { exterior }

::= { pnniRouteAddrEntry 7 }

pnniRouteAddrProto OBJECT-TYPE

```

SYNTAX      INTEGER {
                other(1), -- not specified
                local(2), -- e.g. ilmi
                mgmt(3), -- configured by management,
                        -- for example by SNMP or console
                        -- the following are all dynamic
                        -- routing protocols
                pnni(4) -- ATM Forum PNNI
            }

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The routing mechanism via which the connectivity from the advertising node to the reachable address prefix was learned."

```
 ::= { pnniRouteAddrEntry 8 }
```

pnniRouteAddrPnniScope OBJECT-TYPE

SYNTAX PnniLevel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The PNNI scope of advertisement (i.e. level of PNNI hierarchy) of the reachability from the advertising node to the address prefix."

REFERENCE

"ATM Forum PNNI 1.0 Sections 5.3.6, 5.9.1"

```
 ::= { pnniRouteAddrEntry 9 }
```

pnniRouteAddrVPCapability OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Indicates whether VPCs can be established from the advertising node to the reachable address prefix."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.14.9.1 Table 5-34"

```
 ::= { pnniRouteAddrEntry 10 }
```

pnniRouteAddrMetricsTag OBJECT-TYPE

SYNTAX PnniMetricsTag

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The index into the pnniMetricsTable for the traffic parameter values that apply for the connectivity from the advertising node to the reachable address prefix. There will be one or more entries in the pnniMetricsTable whose first instance identifier matches the value of this variable.

If there are no parameters associated with this reachable address prefix then the distinguished value zero is used."

DEFVAL { 0 }

::= { pnniRouteAddrEntry 11 }

pnniRouteAddrPtseId OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"For reachable addresses learned via PNNI, this attribute contains the value of the PTSE Identifier for the PTSE being originated by the originating node which contains the information group(s) describing the reachable address. For reachable addresses learned by means other than PNNI, this attribute is set to zero."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.2"

::= { pnniRouteAddrEntry 12 }

pnniRouteAddrOriginateAdvertisement OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Whether or not the reachable address specified by this entry is to be advertised by the local node into its PNNI routing domain.

This object may only take on the value 'true' when the value of the corresponding instance of pnniRouteAddrProto is other than 'pnni'."

DEFVAL { true }

::= { pnniRouteAddrEntry 13 }

pnniRouteAddrInfo OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"A reference to MIB definitions specific to the particular routing protocol which is responsible for this reachable address prefix, as determined by the value specified in the route's pnniRouteAddrProto value. If this information is not present, its value should be set to the OBJECT IDENTIFIER zeroDotZero."

DEFVAL { zeroDotZero }

::= { pnniRouteAddrEntry 14 }

pnniRouteAddrOperStatus OBJECT-TYPE

SYNTAX INTEGER {

inactive(1),

active(2), -- i.e. reachability to this

-- prefix exists and is not

-- being advertised in PNNI

advertised(3)

}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates whether the reachable address prefix is operationally valid and whether it is being advertised by this node."

::= { pnniRouteAddrEntry 15 }

pnniRouteAddrTimeStamp OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates when the connectivity from the advertising node to the reachable address prefix became known to the local node."

::= { pnniRouteAddrEntry 16 }

pnniRouteAddrRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"To create, delete, activate and de-activate a reachable address prefix."

```
::= { pnniRouteAddrEntry 17 }
```

-- Table of routes from nodes to reachable transit networks

pnniRouteTnsTable OBJECT-TYPE

SYNTAX SEQUENCE OF PnniRouteTnsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table containing all the attributes necessary to determine what transit networks the PNNI entity believes are reachable and to determine which nodes are advertising this reachability. This table is also used to add static routes to reachable transit networks. The local node index which received the reachability information, type of network identification, network identification plan, transit network identifier, and an index that distinguishes between multiple listings of connectivity to a given transit network from a given local node are combined to form an instance ID for this object."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.3.2"

```
::= { pnniRoutingGroup 5 }
```

pnniRouteTnsEntry OBJECT-TYPE

SYNTAX PnniRouteTnsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the table, containing information about a reachable transit network."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.3.2"

```
INDEX { pnniNodeIndex,
        pnniRouteTnsType,
        pnniRouteTnsPlan,
        pnniRouteTnsId,
        pnniRouteTnsIndex }
```

```
::= { pnniRouteTnsTable 1 }
```

PnniRouteTnsEntry ::=

```
SEQUENCE {
    pnniRouteTnsType          TnsType,
    pnniRouteTnsPlan          TnsPlan,
    pnniRouteTnsId            DisplayString,
    pnniRouteTnsIndex         Integer32,
    pnniRouteTnsIfIndex       InterfaceIndex,
    pnniRouteTnsAdvertisingNodeId PnniNodeId,
    pnniRouteTnsAdvertisedPortId PnniPortId,
    pnniRouteTnsRouteType     INTEGER,
    pnniRouteTnsProto         INTEGER,
    pnniRouteTnsPnniScope     PnniLevel,
    pnniRouteTnsVPCapability   TruthValue,
    pnniRouteTnsMetricsTag     PnniMetricsTag,
    pnniRouteTnsPtseId        Unsigned32,
    pnniRouteTnsOriginateAdvertisement TruthValue,
    pnniRouteTnsInfo          OBJECT IDENTIFIER,
    pnniRouteTnsOperStatus     INTEGER,
    pnniRouteTnsTimeStamp     TimeStamp,
    pnniRouteTnsRowStatus     RowStatus
}
```

pnniRouteTnsType OBJECT-TYPE

```
SYNTAX      TnsType
MAX-ACCESS  not-accessible
STATUS      current
```

DESCRIPTION

"The type of network identification used for this transit network."

```
::= { pnniRouteTnsEntry 1 }
```

pnniRouteTnsPlan OBJECT-TYPE

```
SYNTAX      TnsPlan
MAX-ACCESS  not-accessible
STATUS      current
```

DESCRIPTION

"The network identification plan according to which network identification has been assigned."

```
::= { pnniRouteTnsEntry 2 }
```

pnniRouteTnsId OBJECT-TYPE

```
SYNTAX      DisplayString
MAX-ACCESS  not-accessible
STATUS      current
```

DESCRIPTION

"The value of the transit network identifier."

::= { pnniRouteTnsEntry 3 }

pnniRouteTnsIndex OBJECT-TYPE

SYNTAX Integer32 (1..65535)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An index into the set of listings of connectivity to a given transit network from a given local node."

::= { pnniRouteTnsEntry 4 }

pnniRouteTnsIfIndex OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The local interface over which the transit network can be reached. The value zero indicates an unknown interface or reachability through a remote node."

This object may only have a non-zero value if the value of the corresponding instance of pnniRouteTnsProto is other than 'pnni' and the node identified by pnniRouteTnsAdvertisingNodeId is instantiated within this switching system."

::= { pnniRouteTnsEntry 5 }

pnniRouteTnsAdvertisingNodeId OBJECT-TYPE

SYNTAX PnniNodeId

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The node ID of a node advertising reachability to the transit network. If the local node index is zero, then the advertising node ID must also be set to zero."

::= { pnniRouteTnsEntry 6 }

pnniRouteTnsAdvertisedPortId OBJECT-TYPE

SYNTAX PnniPortId

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The port identifier used from the advertising node to

reach the given transit network."

DEFVAL { 0 }

::= { pnniRouteTnsEntry 7 }

pnniRouteTnsRouteType OBJECT-TYPE

SYNTAX INTEGER {

other(1), -- not specified by this MIB
exterior(4)
}

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The type (e.g. exterior or other) of reachability from the advertising node to the transit network."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.1.3"

DEFVAL { exterior }

::= { pnniRouteTnsEntry 8 }

pnniRouteTnsProto OBJECT-TYPE

SYNTAX INTEGER {

other(1), -- not specified
local(2), -- e.g. ilmi
mgmt(3), -- configured by management,
-- for example by SNMP or console
-- the following are all dynamic
-- routing protocols
pnni(4) -- ATM Forum PNNI
}

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The routing mechanism via which the connectivity from the advertising node to the transit network was learned."

::= { pnniRouteTnsEntry 9 }

pnniRouteTnsPnniScope OBJECT-TYPE

SYNTAX PnniLevel

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The PNNI scope of advertisement (i.e. level of PNNI hierarchy) of the reachability from the advertising node to

the transit network."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.3.6"

::= { pnniRouteTnsEntry 10 }

pnniRouteTnsVPCapability OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Indicates whether VPCs can be established from the advertising node to the reachable transit network."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.14.9.1 Table 5-34"

::= { pnniRouteTnsEntry 11 }

pnniRouteTnsMetricsTag OBJECT-TYPE

SYNTAX PnniMetricsTag

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The index into the pnniMetricsTable for the traffic parameter values that apply for the connectivity from the advertising node to the transit network. There will be one or more entries in the pnniMetricsTable whose first instance identifier matches the value of this variable.

If there are no parameters associated with this transit network then the distinguished value zero is used."

DEFVAL { 0 }

::= { pnniRouteTnsEntry 12 }

pnniRouteTnsPtseId OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"For reachable transit networks learned via PNNI, this attribute contains the value of the PTSE Identifier for the PTSE being originated by the originating node which contains the information group(s) describing the transit network. For reachable transit networks learned by means other than PNNI, this attribute is set to zero."

REFERENCE

"ATM Forum PNNI 1.0 Section 5.8.2"

::= { pnniRouteTnsEntry 13 }

pnniRouteTnsOriginateAdvertisement OBJECT-TYPE

SYNTAX TruthValue
 MAX-ACCESS read-create
 STATUS current

DESCRIPTION

"Whether or not the transit network specified by this entry is to be advertised by the local node into its PNNI routing domain.

This object may only take on the value 'true' when the value of the corresponding instance of pnniRouteNodeProto is other than 'pnni'."

DEFVAL { true }

::= { pnniRouteTnsEntry 14 }

pnniRouteTnsInfo OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER
 MAX-ACCESS read-create
 STATUS current

DESCRIPTION

"A reference to MIB definitions specific to the particular routing protocol which is responsible for this transit network, as determined by the value specified in the route's pnniRouteTnsProto value. If this information is not present, its value should be set to the OBJECT IDENTIFIER zeroDotZero."

DEFVAL { zeroDotZero }

::= { pnniRouteTnsEntry 15 }

pnniRouteTnsOperStatus OBJECT-TYPE

SYNTAX INTEGER {
 inactive(1),
 active(2), -- i.e. reachability to this
 -- transit network exists and is
 -- not being advertised in PNNI
 advertised(3)
 }

MAX-ACCESS read-only
 STATUS current

DESCRIPTION

"Indicates whether the reachable transit network is

operationally valid and whether it is being advertised by this node."

::= { pnniRouteTnsEntry 16 }

pnniRouteTnsTimeStamp OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates how long the connectivity from the advertising node to the reachable transit network has been known to the local node."

::= { pnniRouteTnsEntry 17 }

pnniRouteTnsRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"To create, delete, activate and de-activate a reachable transit network."

::= { pnniRouteTnsEntry 18 }

-- conformance information

pnniMIBConformance

OBJECT IDENTIFIER ::= { pnniMIB 2 }

pnniMIBCompliances

OBJECT IDENTIFIER ::= { pnniMIBConformance 1 }

pnniMIBGroups

OBJECT IDENTIFIER ::= { pnniMIBConformance 2 }

-- compliance statements

pnniMIBCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"The compliance statement for entities which implement the PNNI MIB.

Groups of PNNI objects required for management of a minimum function node are identified by the suffix MinGroup.

Groups of PNNI objects required for management of a border node are identified by the suffix BorderGroup.

Groups of PNNI objects required for management of a PGL/LGN capable node are identified by the suffix LgnGroup.

Groups of optional PNNI objects are identified by the suffix OptionalGroup."

MODULE -- this module

```
MANDATORY-GROUPS { pnniGeneralMinGroup,
                    pnniNodeMinGroup,
                    pnniNodePglMinGroup,
                    pnniNodeTimerMinGroup,
                    pnniScopeMinGroup,
                    pnniIfMinGroup,
                    pnniLinkMinGroup,
                    pnniNbrPeerMinGroup,
                    pnniNbrPeerPortMinGroup }
```

OBJECT pnniNodeId

MIN-ACCESS read-only

DESCRIPTION

"Support for manual configuration of node IDs is optional."

OBJECT pnniNodeLowest

MIN-ACCESS read-only

DESCRIPTION

"Only switching systems that are PGL/LGN capable are allowed to provide write/create access to the pnniNodeLowest object."

OBJECT pnniNodeRestrictedTransit

MIN-ACCESS read-only

DESCRIPTION

"Support for the restricted transit capability is optional."

OBJECT pnniNodeComplexRep

MIN-ACCESS read-only

DESCRIPTION

"The ability to generate the complex node representation is only required for PGL/LGN capable switching systems, and is otherwise optional."

OBJECT pnniNodeRowStatus

SYNTAX INTEGER { active(1) }

MIN-ACCESS read-only

DESCRIPTION

"The ability to create more than one node in a switching system is optional."

OBJECT pnniNodePglLeadershipPriority

MIN-ACCESS read-only

DESCRIPTION

"Only switching systems that are PGL/LGN capable are allowed to provide write/create access to the pnniNodePglLeadershipPriority object."

OBJECT pnniIfNodeIndex

MIN-ACCESS read-only

DESCRIPTION

"Write access to the pnniIfNodeIndex object is optional. It only applies when there can be multiple lowest-level nodes in the switching system."

OBJECT pnniIfVPCapability

MIN-ACCESS read-only

DESCRIPTION

"The ability to support switched virtual paths is optional."

::= { pnniMIBCompliances 1 }

-- units of conformance

pnniGeneralMinGroup OBJECT-GROUP

OBJECTS

pnniHighestVersion,
pnniLowestVersion,
pnniDtlCountOriginator,
pnniCrankbackCountOriginator,
pnniAltRouteCountOriginator,
pnniRouteFailCountOriginator,
pnniRouteFailUnreachableOriginator
}

STATUS current

DESCRIPTION

"A collection of general PNNI objects required for management of a minimum function switching system."

::= { pnniMIBGroups 1 }

pnniGeneralBorderGroup OBJECT-GROUP

OBJECTS

```
pnniDt1CountBorder,  
pnniCrankbackCountBorder,  
pnniAltRouteCountBorder,  
pnniRouteFailCountBorder,  
pnniRouteFailUnreachableBorder  
}
```

STATUS current

DESCRIPTION

"A collection of general PNNI objects required for management of a border node."

::= { pnniMIBGroups 2 }

pnniNodeMinGroup OBJECT-GROUP

OBJECTS {

```
pnniNodeLevel,  
pnniNodeId,  
pnniNodeLowest,  
pnniNodeAdminStatus,  
pnniNodeOperStatus,  
pnniNodeDomainName,  
pnniNodeAtmAddress,  
pnniNodePeerGroupId,  
pnniNodeRestrictedTransit,  
pnniNodeComplexRep,  
pnniNodeRestrictedBranching,  
pnniNodeDatabaseOverload,  
pnniNodePtses,  
pnniNodeRowStatus  
}
```

STATUS current

DESCRIPTION

"A collection of per node PNNI objects required for management of a minimum function switching system."

::= { pnniMIBGroups 3 }

pnniNodePglMinGroup OBJECT-GROUP

OBJECTS {

```
pnniNodePglLeadershipPriority,  
pnniNodePglInitTime,  
pnniNodePglReelectTime ,  
pnniNodePglState,
```

```
pnniNodePreferredPgl,  
pnniNodePeerGroupLeader,  
pnniNodePglTimeStamp,  
pnniNodeActiveParentNodeId  
}
```

STATUS current

DESCRIPTION

"A collection of per node PGL election related PNNI objects required for management of a minimum function switching system."

::= { pnniMIBGroups 4 }

pnniNodePglLgnGroup OBJECT-GROUP

```
OBJECTS {  
    pnniNodeCfgParentNodeIndex,  
    pnniNodePglOverrideDelay  
}
```

STATUS current

DESCRIPTION

"A collection of per node PGL election related PNNI objects required for management of a PGL/LGN capable switching system."

::= { pnniMIBGroups 5 }

pnniNodeTimerMinGroup OBJECT-GROUP

```
OBJECTS {  
    pnniNodePtseHolddown,  
    pnniNodeHelloHolddown,  
    pnniNodeHelloInterval,  
    pnniNodeHelloInactivityFactor,  
    pnniNodePtseRefreshInterval,  
    pnniNodePtseLifetimeFactor,  
    pnniNodeRxmtInterval,  
    pnniNodePeerDelayedAckInterval,  
    pnniNodeAvcrPm,  
    pnniNodeAvcrMt,  
    pnniNodeCdvPm,  
    pnniNodeCtdPm  
}
```

STATUS current

DESCRIPTION

"A collection of per node PNNI objects required for management of timers and significant change thresholds in a minimum function switching system."

```
::= { pnniMIBGroups 6 }
```

```
pnniNodeTimerLgnGroup OBJECT-GROUP
```

```
OBJECTS {  
    pnniNodeHlinkInact  
}
```

```
STATUS current
```

```
DESCRIPTION
```

```
"A collection of per node PNNI objects required for  
management of timers in a PGL/LGN capable switching  
system."
```

```
::= { pnniMIBGroups 7 }
```

```
pnniNodeSvccLgnGroup OBJECT-GROUP
```

```
OBJECTS {  
    pnniNodeSvccInitTime,  
    pnniNodeSvccRetryTime,  
    pnniNodeSvccCallingIntegrityTime,  
    pnniNodeSvccCalledIntegrityTime,  
    pnniNodeSvccTrafficDescriptorIndex  
}
```

```
STATUS current
```

```
DESCRIPTION
```

```
"A collection of per node SVCC-based RCC related PNNI  
objects required for management of a PGL/LGN capable  
switching system."
```

```
::= { pnniMIBGroups 8 }
```

```
pnniScopeMinGroup OBJECT-GROUP
```

```
OBJECTS {  
    pnniScopeLocalNetwork,  
    pnniScopeLocalNetworkPlusOne,  
    pnniScopeLocalNetworkPlusTwo,  
    pnniScopeSiteMinusOne,  
    pnniScopeIntraSite,  
    pnniScopeSitePlusOne,  
    pnniScopeOrganizationMinusOne,  
    pnniScopeIntraOrganization,  
    pnniScopeOrganizationPlusOne,  
    pnniScopeCommunityMinusOne,  
    pnniScopeIntraCommunity,  
    pnniScopeCommunityPlusOne,  
    pnniScopeRegional,  
}
```

```
        pnniScopeInterRegional,  
        pnniScopeGlobal  
    )
```

STATUS current

DESCRIPTION

"A collection of per node scope mapping related PNNI objects required for management of a minimum function switching system."

```
::= { pnniMIBGroups 9 }
```

pnniSummaryLgnGroup OBJECT-GROUP

OBJECTS {

```
    pnniSummaryType,  
    pnniSummarySuppress,  
    pnniSummaryState,  
    pnniSummaryRowStatus  
}
```

STATUS current

DESCRIPTION

"A collection of PNNI objects required for controlling address summarization."

```
::= { pnniMIBGroups 10 }
```

pnniIfMinGroup OBJECT-GROUP

OBJECTS {

```
    pnniIfNodeIndex,  
    pnniIfPortId,  
    pnniIfVPCapability,  
    pnniIfAdmWeightCbr,  
    pnniIfAdmWeightRtVbr,  
    pnniIfAdmWeightNrtVbr,  
    pnniIfAdmWeightAbr,  
    pnniIfAdmWeightUbr,  
    pnniIfRccServiceCategory,  
    pnniIfRccTrafficDescrIndex  
}
```

STATUS current

DESCRIPTION

"A collection of per interface PNNI objects required for management of a minimum function switching system."

```
::= { pnniMIBGroups 11 }
```

pnniIfBorderGroup OBJECT-GROUP

OBJECTS {

```
    pnniIfAggrToken
  }
```

STATUS current

DESCRIPTION

"A collection of per interface PNNI objects required for management of a border node."

```
::= { pnniMIBGroups 12 }
```

pnniLinkMinGroup OBJECT-GROUP

OBJECTS {

```
    pnniLinkType,
    pnniLinkVersion,
    pnniLinkHelloState,
    pnniLinkRemoteNodeId,
    pnniLinkRemotePortId,
    pnniLinkIfIndex,
    pnniLinkRcvHellos,
    pnniLinkXmtHellos
  }
```

STATUS current

DESCRIPTION

"A collection of per link PNNI objects required for management of a minimum function switching system."

```
::= { pnniMIBGroups 13 }
```

pnniLinkBorderOrLgnGroup OBJECT-GROUP

OBJECTS {

```
    pnniLinkDerivedAggrToken,
    pnniLinkUpnodeId,
    pnniLinkUpnodeAtmAddress,
    pnniLinkCommonPeerGroupId
  }
```

STATUS current

DESCRIPTION

"A collection of per link PNNI objects required for management of a border node or a PGL/LGN capable switching system."

```
::= { pnniMIBGroups 14 }
```

pnniLinkLgnGroup OBJECT-GROUP

OBJECTS {

```
    pnniLinkSvccRccIndex
  }
```

STATUS current

DESCRIPTION

"A collection of per link PNNI objects required for management of a PGL/LGN capable switching system."

::= { pnniMIBGroups 15 }

pnniNbrPeerMinGroup OBJECT-GROUP

OBJECTS {

pnniNbrPeerState,
pnniNbrPeerPortCount,
pnniNbrPeerRcvDbSums,
pnniNbrPeerXmtDbSums,
pnniNbrPeerRcvPtsps,
pnniNbrPeerXmtPtsps,
pnniNbrPeerRcvPtseReqs,
pnniNbrPeerXmtPtseReqs,
pnniNbrPeerRcvPtseAcks,
pnniNbrPeerXmtPtseAcks
}

STATUS current

DESCRIPTION

"A collection of per neighboring peer PNNI objects required for management of a minimum function switching system."

::= { pnniMIBGroups 16 }

pnniNbrPeerLgnGroup OBJECT-GROUP

OBJECTS {

pnniNbrPeerSvccRccIndex
}

STATUS current

DESCRIPTION

"A collection of per neighboring peer PNNI objects required for management of a PGL/LGN capable switching system."

::= { pnniMIBGroups 17 }

pnniNbrPeerPortMinGroup OBJECT-GROUP

OBJECTS {

pnniNbrPeerPortFloodStatus
}

STATUS current

DESCRIPTION

"A collection of per port to neighboring peer PNNI objects required for management of a minimum function switching system."

::= { pnniMIBGroups 18 }

pnniSvccRccLgnGroup OBJECT-GROUP

OBJECTS {

pnniSvccRccVersion,
pnniSvccRccHelloState,
pnniSvccRccRemoteNodeId ,
pnniSvccRccRemoteAtmAddress,
pnniSvccRccRcvHellos,
pnniSvccRccXmtHellos,
pnniSvccRccIfIndex,
pnniSvccRccVpi,
pnniSvccRccVci
}

STATUS current

DESCRIPTION

"A collection of per SVCC-based RCC PNNI objects required
for management of a PGL/LGN capable switching system."

::= { pnniMIBGroups 19 }

pnniPtseOptionalGroup OBJECT-GROUP

OBJECTS {

pnniPtseType,
pnniPtseSequenceNum,
pnniPtseChecksum,
pnniPtseLifeTime,
pnniPtseInfo
}

STATUS current

DESCRIPTION

"A collection of optional per PTSE PNNI objects."

::= { pnniMIBGroups 20 }

pnniMapOptionalGroup OBJECT-GROUP

OBJECTS {

pnniMapType,
pnniMapPeerGroupId,
pnniMapAggrToken,
pnniMapRemoteNodeId,
pnniMapRemotePortId,
pnniMapVPCapability,
pnniMapPtseId,
pnniMapMetricsTag
}

STATUS current

DESCRIPTION

"A collection of optional PNNI objects used to create a map of nodes and links in the PNNI routing domain."

::= { pnniMIBGroups 21 }

pnniMapNodeOptionalGroup OBJECT-GROUP

OBJECTS {

pnniMapNodePeerGroupId,
pnniMapNodeAtmAddress,
pnniMapNodeRestrictedTransit,
pnniMapNodeComplexRep,
pnniMapNodeRestrictedBranching,
pnniMapNodeDatabaseOverload,
pnniMapNodeIAmLeader,
pnniMapNodeLeadershipPriority,
pnniMapNodePreferredPgl,
pnniMapNodeParentNodeId,
pnniMapNodeParentAtmAddress,
pnniMapNodeParentPeerGroupId,
pnniMapNodeParentPglNodeId
}

STATUS current

DESCRIPTION

"A collection of optional PNNI objects used to create a map of nodes in the PNNI routing domain."

::= { pnniMIBGroups 22 }

pnniMapAddrOptionalGroup OBJECT-GROUP

OBJECTS {

pnniMapAddrAddress,
pnniMapAddrPrefixLength
}

STATUS current

DESCRIPTION

"A collection of optional PNNI objects used to create a map of reachable addresses in the PNNI routing domain."

::= { pnniMIBGroups 23 }

pnniMapTnsOptionalGroup OBJECT-GROUP

OBJECTS {

pnniMapTnsId
}

STATUS current

DESCRIPTION

"A collection of optional PNNI objects used to create a map of reachable transit networks in the PNNI routing domain."
 ::= { pnniMIBGroups 24 }

pnniMetricsOptionalGroup OBJECT-GROUP

```
OBJECTS {
    pnniMetricsClasses,
    pnniMetricsGcacClp,
    pnniMetricsAdminWeight,
    pnniMetrics1,
    pnniMetrics2,
    pnniMetrics3,
    pnniMetrics4,
    pnniMetrics5,
    pnniMetrics6,
    pnniMetrics7,
    pnniMetrics8,
    pnniMetricsRowStatus
}
```

STATUS current

DESCRIPTION

"A collection of optional PNNI objects used to manage metrics and attributes associated with PNNI entities."
 ::= { pnniMIBGroups 25 }

pnniRouteGeneralOptionalGroup OBJECT-GROUP

```
OBJECTS {
    pnniRouteNodeNumber,
    pnniRouteAddrNumber
}
```

STATUS current

DESCRIPTION

"A collection of optional PNNI objects."
 ::= { pnniMIBGroups 26 }

pnniRouteNodeOptionalGroup OBJECT-GROUP

```
OBJECTS {
    pnniRouteNodeDestPortId,
    pnniRouteNodeProto,
    pnniRouteNodeTimeStamp,
    pnniRouteNodeInfo,
    pnniRouteNodeGcacClp,
    pnniRouteNodeFwdMetricAW,
```

```

pnniRouteNodeFwdMetric1,
pnniRouteNodeFwdMetric2,
pnniRouteNodeFwdMetric3,
pnniRouteNodeFwdMetric4,
pnniRouteNodeFwdMetric5,
pnniRouteNodeFwdMetric6,
pnniRouteNodeFwdMetric7,
pnniRouteNodeFwdMetric8,
pnniRouteNodeBwdMetricAW,
pnniRouteNodeBwdMetric1,
pnniRouteNodeBwdMetric2,
pnniRouteNodeBwdMetric3,
pnniRouteNodeBwdMetric4,
pnniRouteNodeBwdMetric5,
pnniRouteNodeBwdMetric6,
pnniRouteNodeBwdMetric7,
pnniRouteNodeBwdMetric8,
pnniRouteNodeVPCapability,
pnniRouteNodeStatus
}

```

STATUS current

DESCRIPTION

"A collection of optional PNNI objects used to manage precalculated routes to nodes in the PNNI routing domain."

::= { pnniMIBGroups 27 }

pnniDTLOptionalGroup OBJECT-GROUP

```

OBJECTS {
    pnniDTLNodeId,
    pnniDTLPortId,
    pnniDTLLinkType,
    pnniDTLStatus
}

```

STATUS current

DESCRIPTION

"A collection of optional PNNI objects used to manage precalculated routes to nodes in the PNNI routing domain."

::= { pnniMIBGroups 28 }

pnniRouteAddrOptionalGroup OBJECT-GROUP

```

OBJECTS {
    pnniRouteAddrIfIndex,
    pnniRouteAddrAdvertisingNodeId,
    pnniRouteAddrAdvertisedPortId,
}

```

```
pnniRouteAddrType,  
pnniRouteAddrProto,  
pnniRouteAddrPnniScope,  
pnniRouteAddrVPCapability,  
pnniRouteAddrMetricsTag,  
pnniRouteAddrPtseId,  
pnniRouteAddrOriginateAdvertisement,  
pnniRouteAddrInfo,  
pnniRouteAddrOperStatus,  
pnniRouteAddrTimeStamp,  
pnniRouteAddrRowStatus  
}
```

STATUS current

DESCRIPTION

"A collection of optional PNNI objects used to manage routes to reachable addresses in the PNNI routing domain."

::= { pnniMIBGroups 29 }

pnniRouteTnsOptionalGroup OBJECT-GROUP

OBJECTS {

```
pnniRouteTnsIfIndex,  
pnniRouteTnsAdvertisingNodeId,  
pnniRouteTnsAdvertisedPortId,  
pnniRouteTnsRouteType,  
pnniRouteTnsProto,  
pnniRouteTnsPnniScope,  
pnniRouteTnsVPCapability,  
pnniRouteTnsMetricsTag,  
pnniRouteTnsPtseId,  
pnniRouteTnsOriginateAdvertisement,  
pnniRouteTnsInfo,  
pnniRouteTnsOperStatus,  
pnniRouteTnsTimeStamp,  
pnniRouteTnsRowStatus  
}
```

STATUS current

DESCRIPTION

"A collection of optional PNNI objects used to manage routes to reachable transit networks in the PNNI routing domain."

::= { pnniMIBGroups 30 }

END