

**YD**

# 中华人民共和国通信行业标准

YD/T 1631-2007

---

## 同步数字体系（SDH）虚级联及链路容量 调整方案技术要求

Technical Requirements for Synchronous Digital Hierarchy(SDH)  
virtual concatenation and link capacity adjustment scheme

2007-04-16 发布

2007-10-01 实施

---

中华人民共和国信息产业部 发布

## 目 次

前 言	II
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 缩略语	2
5 VC 级联	3
5.1 级联定义	3
5.2 虚级联	4
6 基于 VC 虚级联的 LCAS	9
6.1 基本方法	9
6.2 控制包	9
6.3 LCAS 操作过程	15
7 LCAS 和非 LCAS 之间互通	17
7.1 LCAS 发送机和非 LCAS 接收机之间互通	17
7.2 非 LCAS 发送机和 LCAS 接收机之间互通	17
8 性能要求	17
8.1 最大可补偿差分时延	17
8.2 RS_Ack 超时时间 (可选)	17
8.3 等待恢复 (WTR) 时间 (可选)	17
8.4 拖延 (HO) 时间 (可选)	17
8.5 业务受损时间 (可选)	17
8.6 宿端删除 (RMV) 等待时间 (可选)	17
9 管理要求	18
9.1 性能统计	18
9.2 告警	18
9.3 配置	18
附录 A (规范性附录) 相邻级联	19
附录 B (规范性附录) 相邻级联和虚级联互通	20
附录 C (规范性附录) LCAS 协议	24
附录 D (资料性附录) LCAS 时序图	32
附录 E (资料性附录) 拖延期间无损更改带宽	38
附录 F (资料性附录) VCG 连续数据映射	39
附录 G (资料性附录) 和映射结构有关的 POH 字节	40

## 前 言

本标准对应于国际电信联盟 – 电信标准部门 (ITU-T) G.7042/Y.1305: 2006《虚级联信号的链路容量调整方案》(英文版), 与 ITU-T G.7042/Y.1305 的一致性程度为非等效。

本标准与 ITU-T G.7042/Y.1305 相比主要变化如下:

- 增加了“VC 级联”(见第 5 章);
- 增加了“性能要求”(见第 8 章);
- 增加了“管理要求”(见第 9 章);
- 增加了规范性附录“相邻级联”(见附录 A);
- 增加了规范性附录“相邻级联和虚级联互通(可选)”(见附录 B);
- 增加了资料性附录“VCG 连续数据映射”(见附录 F);
- 增加了资料性附录“和映射结构相关的 POH 字节”(见附录 G)。

本标准的附录 A、附录 B 和附录 C 为规范性附录, 附录 D、附录 E、附录 F 和附录 G 为资料性附录。

本标准由中国通信标准化协会提出并归口。

本标准起草单位: 信息产业部电信研究院

上海贝尔阿尔卡特股份有限公司

中兴通讯股份有限公司

华为技术有限公司

本标准主要起草人: 赵文玉 徐一军 彭松涛 沈永红 郁志勇 程永刚 杨 洋

# 同步数字体系（SDH）虚级联及链路容量调整方案技术要求

## 1 范围

本标准规定了 SDH 虚级联（VCAT）及链路容量调整方案（LCAS）的技术要求，主要包括 SDH 系统中 VCAT 的定义、VCAT 信号中 LCAS 的实现方法、LCAS 的实现过程、LCAS 和非 LCAS 的互通、VCAT 及 LCAS 的性能要求和管理要求等。本标准适用于 SDH 网络中多种 VC 颗粒（VC-12/VC-3/VC-4）的 VCAT 及 LCAS。

## 2 规范性引用文件

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

ITU-T G.7042/Y.1305 (2006)	虚级联信号的链路容量调整方案
ITU-T G.707/Y.1322 (2003)	数字同步序列（SDH）网络节点接口
ITU-T G.783 (2004)	数字同步序列（SDH）设备功能块特性
ITU-T G.808.1	通用保护倒换—线性路径和子网保护
ITU-T G.806	传送设备特性—描述方法和通用功能
ITU-T Z.100	规范描述语言（SDL）

## 3 术语和定义

下列术语和定义适用于本标准。

### 3.1

#### 链路 link

网络中从终结功能到终结功能之间的连接，可与虚级联组中的成员以及虚级联组自身相关联。

### 3.2

#### 成员 member

隶属于虚级联组的单个服务层容器。

### 3.3

#### 虚容器级联 virtual container concatenation

较小容量容器复用为单个较大容量容器的过程，有相邻级联和虚级联两种方式。

### 3.4

#### 相邻级联 contiguous concatenation

多个相邻的较小虚容器复用为单个较大虚容器，并且在贯穿全部的传送中保持相邻的级联带宽。这就要求级联带宽经历的每一个网元都必须支持相邻级联功能。

## 3.5

**虚级联 virtual concatenation**

多个相邻或非相邻的较小虚容器复用为单个较大虚容器，并且这些较小的虚容器在传送过程中相互独立，而在传输的终结点处将这些较小虚容器重新组合成一个相邻带宽。这就要求仅在通道开始和终结的网元处支持虚级联功能，而对于中间历经的网元不作要求。

## 3.6

**虚级联组 virtual concatenation group**

连接到相同虚级联链路并且实现共址成员路径终结功能的组。

## 4 缩略语

下列缩略语适用于本标准。

AIS	Alarm Indication Signal	告警指示信号
ATM	Asynchronous Transfer Mode	异步传送模式
CRC	Cyclic Redundancy Check	循环冗余校验
CTRL	Control field sent from source to sink	源到宿发送的控制字段
DNU	Do Not Use	不使用
DQDB	Distributed Queue Dual Bus	分布队列双总线
EMS	Element Management System	网元管理系统
EOS	End Of Sequence	序列尾
FC	Fibre Channel	光纤通道
FDDI	Fibre Distributed Data Interface	光纤分布式数据接口
GFP	Generic Framing Procedure	通用成帧规程
GID	Group Identification	组识别
HDLC	High-level Data Link Control	高层数据链路控制
HO	Hold Off	拖延
IEC	Incoming Error Count	输入错误统计
LAPS	Link Access Procedure – SDH	链路接入规程 – SDH
LCAS	Link Capacity Adjustment Scheme	链路容量调整方案
LCASC	Link Capacity Adjustment Scheme Controller	链路容量调整方案控制器
LOA	Loss Of Alignment	对齐丢失
LOM	Loss Of Multiframe	复帧丢失
LSB	Least Significant Bit	最低有效比特位
MAN	Metropolitan Area Network	城域网
MFI	MultiFrame Indicator	复帧指示
MI	Management Information	管理信息
MND	Member Not Deskewable	成员不可用
MSB	Most Significant Bit	最高有效比特位
MST	Member Status	成员状态

MSU_L	Member Status Unavailable, LCAS enabled criteria	成员状态不可用, LCAS 使能准则
NMS	Network Management System	网络管理系统
NORM	Normal Operating Mode	正常操作模式
ODUk	Optical channel Data Unit-k	光通路数据单元-k
POH	Path OverHead	通道开销
PPP	Point-to-Point Protocol	点到点协议
RMV	Remove	删除
RTTI	Received Trail Trace Identifier	接收路径踪迹标识
RS-Ack	Re-Sequence Acknowledge	重排序确认
SDH	Synchronous Digital Hierarchy	同步数字体系
Sk	Sink	宿端
So	Source	源端
SQ	Sequence Indicator	序列指示
SQM	Sequence Indicator Mismatch	序列指示失配
TCM	Tandem Connection Monitoring	串联连接监视
TLCR	Total Loss of Capacity Receive	宿端丢失全部容量
TLCT	Total Loss of Capacity Transmit	源端丢失全部容量
TSD	Trail Signal Degrade	路径信号劣化
TU-n	Tributary Unit-n	支路单元
TUG	Tributary Unit Group	支路单元组
VC	Virtual Container	虚容器
VCAT	Virtual Concatenation	虚级联
VCG	Virtual Concatenation Group	虚级联组
WTR	Wait To Restore	等待恢复
X <sub>A</sub>	Actual number of members of a virtual concatenated group	虚级联组实际成员个数
X <sub>M</sub>	Maximum size of a virtual concatenated group	虚级联组最大成员个数
X <sub>P</sub>	Number of provisioned members in a virtual concatenated group	虚级联组已指派成员个数

## 5 VC 级联

### 5.1 级联定义

当标准的虚容器 (VC-12、VC-3、VC-4) 不能有效地传送净荷时, 可采用 VC 级联的方式来传送净荷。VC 级联适用于以下几种情况:

- 当净荷要求采用 VC-12 颗粒传送但其容量大于 1 个 VC-12 时, 可采用 VC-12 的级联来传送;
- 当净荷要求采用 VC-3 颗粒传送但其容量大于 1 个 VC-3 时, 可采用 VC-3 的级联来传送;
- 当净荷要求采用 VC-4 颗粒传送但其容量大于 1 个 VC-4 时, 可采用 VC-4 的级联来传送。

目前定义了两种 VC 级联方式: 相邻级联和虚级联。这两种级联方式的共同特点是, 在通道终结处都可提供标准容器 C-n 的 X 倍级联带宽, 而两者的差异主要体现在通道终结点之间的传送过程中。对于

相邻级联而言,在整个传送过程中,级联带宽保持不变,而虚级联则把级联带宽分割为单个的 VC-n 进行传送,在传输的终点再重新组合这些单个的 VC-n 为完整的级联带宽。另外,相邻级联要求业务通道通过的每个网络节点都必须支持相邻级联功能,而虚级联则仅要求业务通道终结点设备支持虚级联功能。

相邻级联和虚级联并不是完全独立的,两者可互通,具体要求参见本标准附录 B。相邻级联和虚级联采用以下表示方法:

a) 相邻级联表示为: VC-n-Xc

其中:

VC-n——虚容器的大小等级,取值分别为 VC-12、VC-3、VC-4 等;

X——级联的虚容器的个数,取值与虚容器等级有关,如对于 VC-4,取值分别为 4/16/64/256 等;

c——级联类型为相邻级联。

b) 虚级联表示为: VC-n-Xv

其中:

VC-n——虚容器的大小等级,取值分别为 VC-12、VC-3、VC-4 等。

X——级联的虚容器的个数,取值与虚容器等级有关。对于 VC-3 和 VC-4,取值范围为 1~256;对于 VC-12,取值范围为 1~64。

V——级联类型为虚级联。

注:对于 VC-3 和 VC-4 的相邻级联定义,见附录 A;对于 VC-11、VC-2 等颗粒的相邻级联和虚级联定义,见 ITU-T G707/Y.1322 第 11 章。

## 5.2 虚级联

### 5.2.1 VC-3/VC-4虚级联 (VC-3-Xv/VC-4-Xv, X=1...256)

VC-3-Xv/VC-4-Xv 提供了等同于 X 个 C-3/C-4 容器 (VC-4/3-Xc) 的传送能力,可由 C-3-X/C-4-X 类结构来表示,其净荷的容量为  $X \times 48384/149760 \text{ kbit/s}$ ,如图 1 和图 2 中所示。净荷被映射进 X 个独立的 VC-3/VC-4 中以组成 VC-3-Xv/VC-4-Xv (即 VCG),具体映射过程参见附录 F。每个 VC-3/VC-4 有自己的 POH,其中 POH 字节中的 H4 作为下文中定义的虚级联特定的序号和复帧指示,其它与映射结构相关的 POH 字节参见附录 G。

VC-3-Xv/VC-4-Xv 中的每个 VC-3/VC-4 在网络中是独立传输的。由于每个 VC-3/VC-4 传输延时的不同,不同的 VC-3/VC-4 之间在宿端将产生差分时延。为了访问相邻的净荷区,这些时延差必须补偿并且单个的 VC-3/VC-4 需要重新排序。重新排序的处理必须能容忍至少  $125\mu\text{s}$  的差分时延。

在 VC-3/VC-4 虚级联中,采用时长为 512ms 的两级复帧可容忍  $125\mu\text{s}$  及其以上 (可到 256ms) 的差分时延,如表 1 所示。第一级复帧采用 H4 字节的比特 5~8 作为 4 比特的复帧指示器 (MFI1)。每过一个基帧 MFI1 增加 1, MFI1 数值在 0~15 之间循环。第二级的复帧指示器 (MFI2) 采用 8 个比特来标识,这 8 个比特由第一级复帧中的第一个复帧的基帧 0 和 1 的 H4 字节的部分比特所组成,即基帧 0 的 H4 的比特 1~4 为 MFI2 的比特 1~4,基帧 1 的 H4 的比特 1~4 为 MFI2 的比特 5~8。MFI2 在每过一个第一级复帧时增加 1,数值在计数 0~255 之间循环。这样,两级复帧一共是 4096 帧 ( $256 \times 16$ ),占用时长为 512ms ( $4096 \times 125\mu\text{s}$ )。

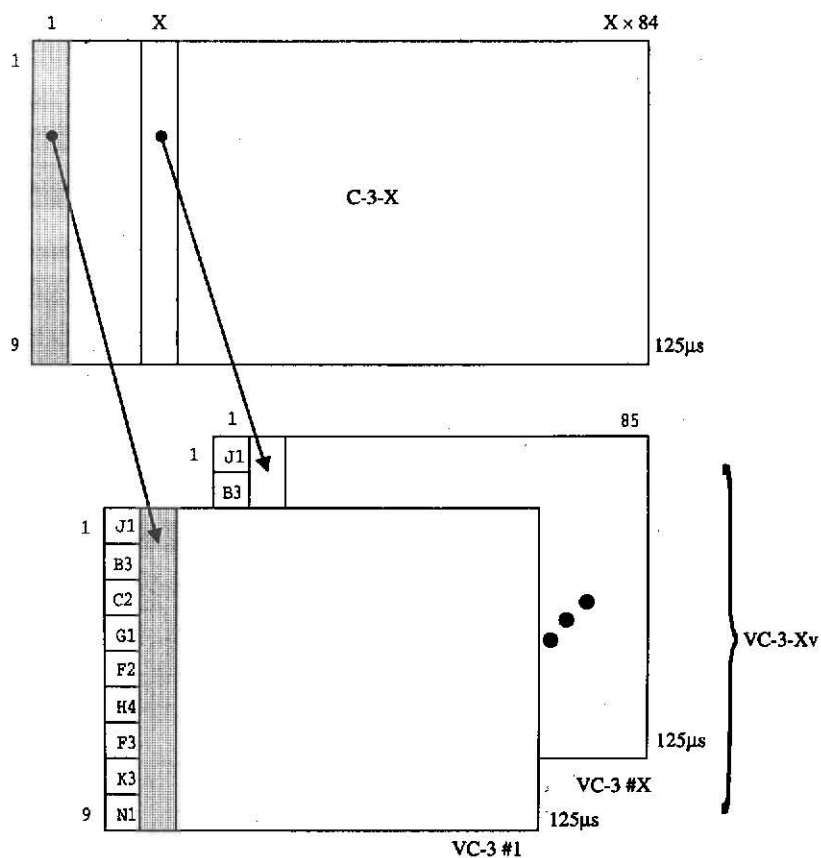


图 1 VC-3-Xv 结构

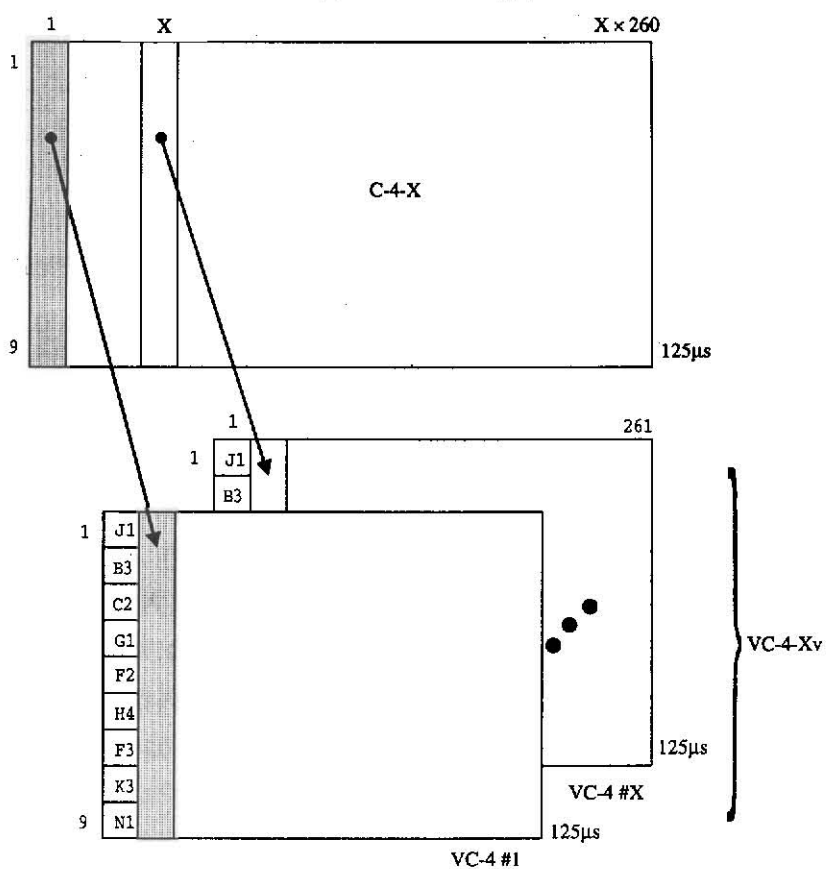


图 2 VC-4-Xv 结构



表 1 包含 VC-3-Xv/VC-4-Xv 序列和复帧指示的 H4 字节编码

H4 字节								第一级复帧编号	第二级复帧编号
比特 1	比特 2	比特 3	比特 4	比特 5	比特 6	比特 7	比特 8		
				第一级复帧指示 MFI1 (比特 1~4)					
序列指示 MSB (比特 1~4)				1	1	1	0	14	n-1
序列指示 LSB (比特 5~8)				1	1	1	1	15	
第二级复帧指示 MFI2 MSB (比特 1~4)				0	0	0	0	0	n
第二级复帧指示 MFI2 LSB (比特 5~8)				0	0	0	1	1	
保留 ("0000")				0	0	1	0	2	
保留 ("0000")				0	0	1	1	3	
保留 ("0000")				0	1	0	0	4	
保留 ("0000")				0	1	0	1	5	
保留 ("0000")				0	1	1	0	6	
保留 ("0000")				0	1	1	1	7	
保留 ("0000")				1	0	0	0	8	
保留 ("0000")				1	0	0	1	9	
保留 ("0000")				1	0	1	0	10	
保留 ("0000")				1	0	1	1	11	
保留 ("0000")				1	1	0	0	12	
保留 ("0000")				1	1	0	1	13	
序列指示 SQ MSB (比特 1~4)				1	1	1	0	14	
序列指示 SQ LSB (比特 5~8)				1	1	1	1	15	
第二级复帧指示 MFI2 MSB (比特 1~4)				0	0	0	0	0	n+1
第二级复帧指示 MFI2 LSB (比特 5~8)				0	0	0	1	1	
保留 ("0000")				0	0	1	0	2	

序列指示 (SQ) 用来区分 VC-3-Xv/VC-4-Xv 中分离的 VC-3/VC-4 的序列或者次序, 并把他们合并以组成 C-3-X/C-4-X 类结构的相邻净荷, 如图 3 中所示。VC-3-Xv/VC-4-Xv 中每一个 VC-3/VC-4 都有一个固定且唯一的序号, 范围在 0~(X-1) 之间。传送 C-4-X 类结构的第 1, X+1, 2X+1, .....259X+1 列数据的 VC-4 的序列指示为 0; 传送 C-4-X 类结构的第 2, X+2, 2X+2, .....259X+2 列数据的 VC-4 的序列指示为 1; 依此类推, 传送 C-4-X 类结构的第 X, X+X, 2X+X, .....259X+X 列数据的 VC-4 的序列指示为 X-1。类似地, 传送 C-3-X 类结构的第 1, X+1, 2X+1, .....83X+1 列数据的 VC-3 的序列指示为 0; 传送 C-3-X 类结构的第 2, X+2, 2X+2, .....83X+2 列数据的 VC-3 的序列指示为 1; 依此类推, 传送 C-3-X 类结构的第 X, X+X, 2X+X, .....83X+X 列数据的 VC-3 的序列指示为 X-1。对于固定带宽的应用, 序列号是固定分配且不可配置的。这使得不采用回溯就能检查 VC-3-Xv/VC-4-Xv 的组成。8 个比特的序列号 (支持 X 值的范围最大为 256) 用 H4 的比特 1~4 来传输, 第一级复帧第十五基帧 H4 的比特 1~4 组成 SQ 的 1~4 比特, 第十六帧 H4 的比特 1~4 组成 SQ 的 5~8 比特, 如表 1 中所示。

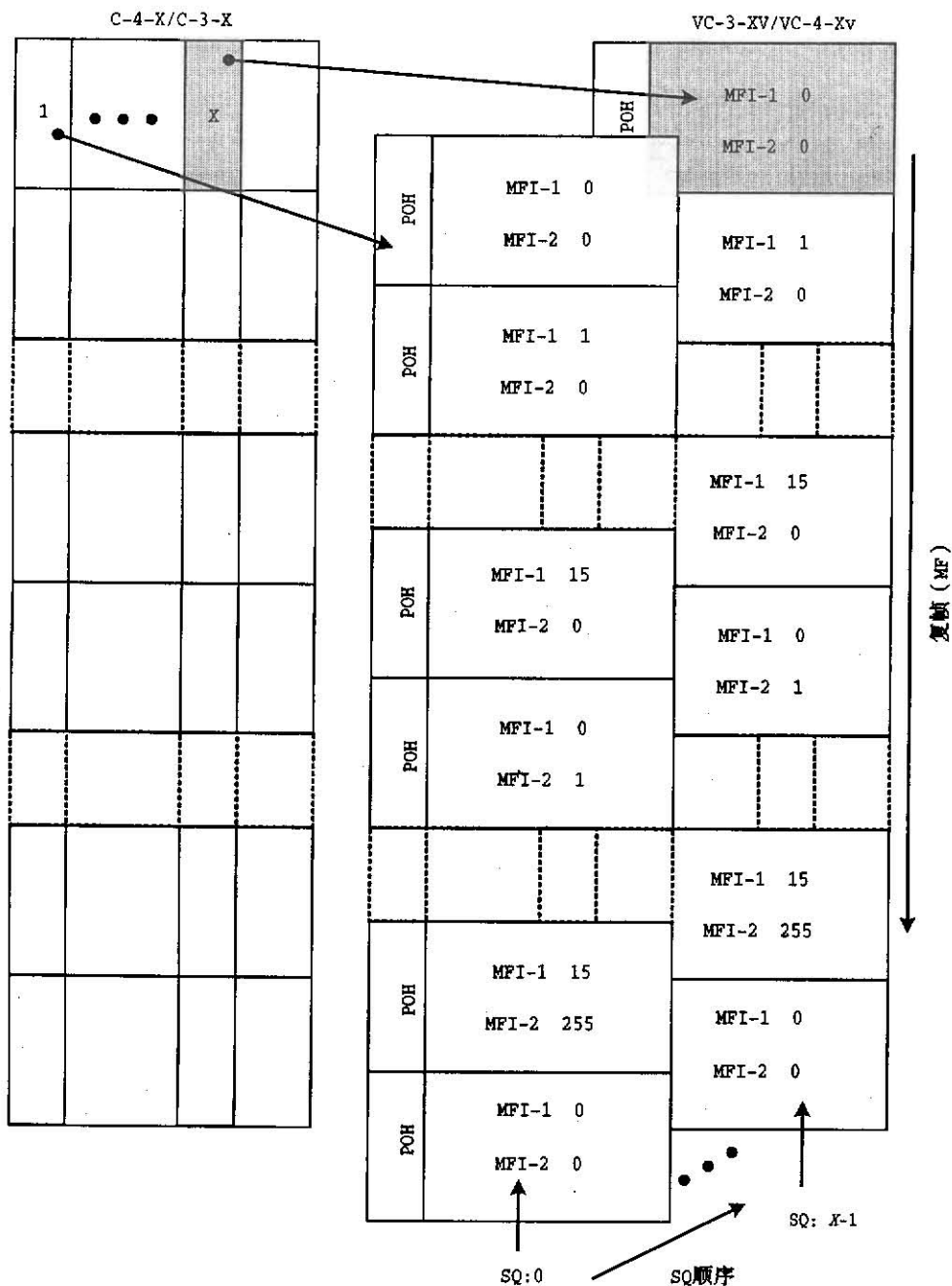


图3 VC-3-Xv/VC-4-Xv 复帧和序列指示

### 5.2.2 VC-12虚级联 (VC-12-Xv, X=1...64)

VC-12-Xv 提供了等同于 X 个容器 C-12 的传送能力, 其由 C-12-X 类结构表示, 可传送的净荷容量为  $X \times 2176\text{ kbit/s}$ , 如图 4 中所示。净荷映射进 X 个独立的 VC-12 中以形成 VC-12-Xv (即 VCG), 具体映射过程参见附录 D。每个 VC-12 有自己的 POH。POH 字节中的 V5 (b5~b7) 和 K4 (b1) 作为信号标记和扩展信号标记, 指示了 VC-12-Xv 的组成状态。其它与映射结构相关的 POH 字节参见附录 G。

VC-12-Xv 中的每一个 VC-12 都是独立地在网络上传输, 因此, 不同的 VC-12 之间存在差分时延。这样, VC-12 的顺序和排列将会发生改变。在终结点, 独立的 VC-12 必须重新排列和组合以便重新建立相邻级联的容器。重新排列的处理必须至少能容忍  $125\mu\text{s}$  的差分时延。

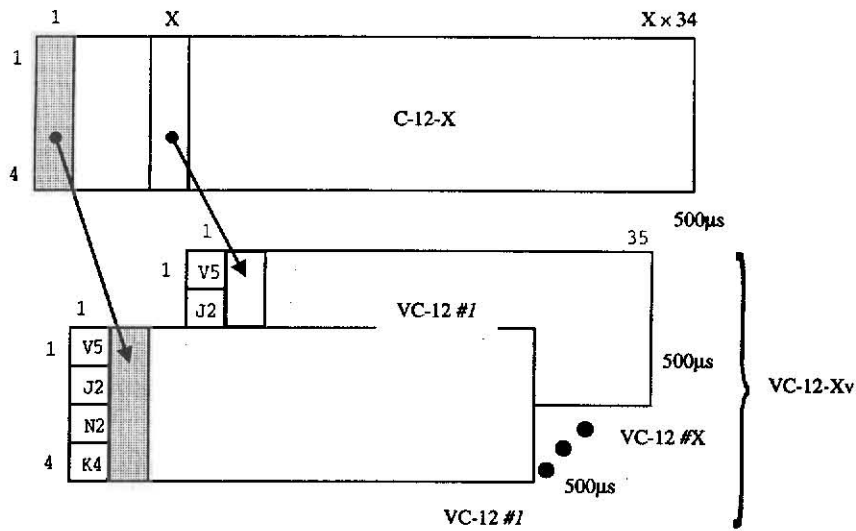


图4 VC-12-Xv 结构

在表 2 中说明了 VC-12-Xv 所能携带的净荷容量。

表 2 VC-12-Xv 虚级联容量

VC-12-Xv	X	容量	步长
	1~ 64 (注)	2176~139264kbit/s	2176 kbit/s

注: 受限 于 64 是因为在 K4 字节比特 2 的复帧中, 6 个比特用于序列指示

在重组隶属于同一虚级联组中分离的 VC-12 之前, 需要解决的问题是:

- 补偿每个 VC-12 之间所经历的差分时延;
- 获取每个 VC-12 的序列指示。

低阶通道 VC-12 POH 字节 K4 的比特 2 用于在发送端和执行 VC-12 虚级联信号重组的接收端之间传送所需要的信息, 即定义了 K4 比特 2 的 32 比特复帧来传递低阶通道虚级联信息, 如表 3 中所示。由于 K4 的比特 2 由 4 帧的映射复帧 (第一级复帧) 来提供 ( $4 \times 125\mu s = 500\mu s$ ), 因此该 32 比特复帧 (第二级复帧) 每 16ms ( $32 \times 500\mu s$ ) 或每 128 帧 ( $32 \times 4$ ) 重复一次。

表 3 K4 比特 2 复帧结构 (第二级复帧)

比特编号																															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
帧计数					序列指示							R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
R 保留比特																															

在表 3 的复帧中, 比特 1~5 用于低阶虚级联的帧计数, 比特 6~11 用于低阶虚级联的序列指示, 剩余的 21 个比特保留为将来使用, 都被置为全 “0” 且被接收机忽略。低阶虚级联的帧计数字段提供了度量差分时延的尺度, 两级复帧的时长为 512ms (映射复帧结构占用时间 16ms, 而第二级复帧帧计数字段为 5 个比特, 因此, 总计持续时间为  $32 \times 16 = 512ms$ ), 可容忍 125 μs 及其以上 (可到 256ms) 的差分时延。

低阶虚级联序列指示用于确认 VC-12-Xv 中独立的 VC-12 的序列或者顺序, 以便组成相邻的 C-12-X 类净荷, 如图 4 中所示。VC-12-Xv 中的每一个 VC-12 都有一个固定且惟一的序列号, 其取值范围为 0~(X-1) 之间。传送 C-12-X 类结构净荷的第 1, X+1, 2X+1, ..., 34X+1 列的 VC-12 的序列指示为 0; 传送 C-12-X 类结构净荷的第 2, X+2, 2X+2, ..., 34X+2 列的 VC-12 的序列指示为 1; 依此类推, 传送 C-12-X

类结构净荷的第  $X$ ,  $X+X$ ,  $2X+X$ , ...,  $34X+X$  列的 VC-12 的序列指示为  $X-1$ 。在要求固定带宽需求的应用中, 序列号固定分配且不可配置, 这样, 在不使用回溯的情况下就可查看 VC-12- $X_v$  组成。

需要注意的是, 在 K4 比特 2 中的低阶虚级联信息的相位必须和 K4 比特 1 中的扩展信号标签相一致, 即虚级联的 VC-12 必须使用扩展信号标签 (即 K4 的比特 1), 否则 K4 比特 2 的复帧状态不能建立。

## 6 基于 VC 虚级联的 LCAS

### 6.1 基本方法

LCAS 在虚级联源和宿的适配功能中提供了控制机制, 能够无损地增加或减少 VCG 链路容量, 以满足带宽需求方面的应用, 同时也提供了临时删除失效链路成员的能力。当隶属于 VCG 的所有成员的传输都是无误码时, 无损带宽改变是可以保证的。LCAS 机制假设在链路容量调整 (如初始化、增加、减少或删除) 时, 网络和网元管理系统 (NMS/EMS) 负责单个链路成员端到端通道的建立或删除。在源端或宿端都可以发起 VCG 链路容量的增加或者删除。

LCAS 通常假设 VCG 的成员与其传送方向之间彼此无关, 这就意味着 LCAS 为非对称连接, 即前向传送的带宽独立于反向传送的带宽。基于此考虑, 附录 C 中的状态框图和附录 D 中的时序图中仅考虑非对称连接。

LCAS 的对称性连接待研究。

### 6.2 控制包

#### 6.2.1 控制包字段组成

发送机 (源端) 和接收机 (宿端) 之间链路容量变化的同步需要通过控制包来实现。每个控制包描述的是下一控制包发送期间的链路状态, 提前发送变化状态以便接收机一旦收到变化信息时便立即切换到新的配置。

控制包由不同特定功能的字段组成, 控制包包括从  $S_o$  到  $S_k$  以及从  $S_k$  到  $S_o$  发送的信息, 如图 5 中所示。

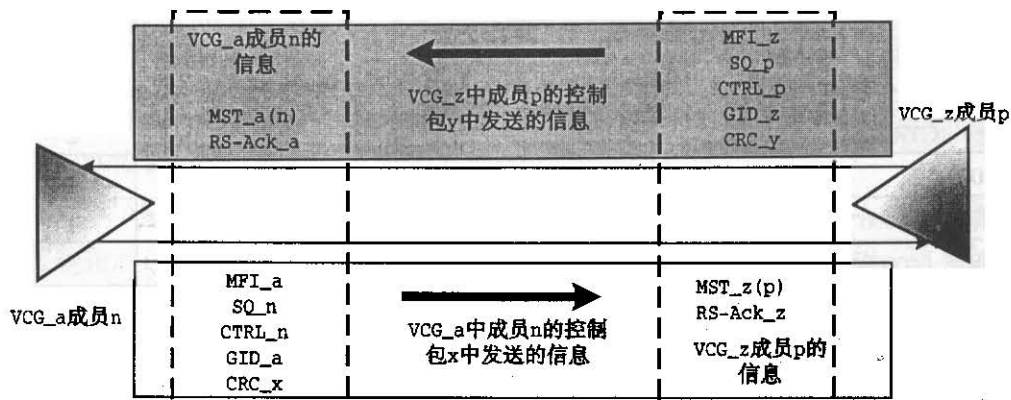


图 5 控制包信息分配

从  $S_o$  到  $S_k$  方向包含的控制包字段为:

- 复帧指示 (MFI) 字段;
- 序列指示 (SQ) 字段;
- 控制 (CTRL) 字段;
- 组识别 (GID) 比特。

从  $S_k$  到  $S_o$  包含的控制包字段为:

类结构净荷的第  $X, X+X, 2X+X, \dots, 34X+X$  列的 VC-12 的序列指示为  $X-1$ 。在要求固定带宽需求的应用中，序列号固定分配且不可配置，这样，在不使用回溯的情况下就可查看 VC-12- $Xv$  组成。

需要注意的是，在 K4 比特 2 中的低阶虚级联信息的相位必须和 K4 比特 1 中的扩展信号标签相一致，即虚级联的 VC-12 必须使用扩展信号标签（即 K4 的比特 1），否则 K4 比特 2 的复帧状态不能建立。

## 6 基于 VC 虚级联的 LCAS

### 6.1 基本方法

LCAS 在虚级联源和宿的适配功能中提供了控制机制，能够无损地增加或减少 VCG 链路容量，以满足带宽需求方面的应用，同时也提供了临时删除失效链路成员的能力。当隶属于 VCG 的所有成员的传输都是无误码时，无损带宽改变是可以保证的。LCAS 机制假设在链路容量调整（如初始化、增加、减少或删除）时，网络和网元管理系统（NMS/EMS）负责单个链路成员端到端通道的建立或删除。在源端或宿端都可以发起 VCG 链路容量的增加或者删除。

LCAS 通常假设 VCG 的成员与其传送方向之间彼此无关，这就意味着 LCAS 为非对称连接，即前向传送的带宽独立于反向传送的带宽。基于此考虑，附录 C 中的状态框图和附录 D 中的时序图中仅考虑非对称连接。

LCAS 的对称性连接待研究。

### 6.2 控制包

#### 6.2.1 控制包字段组成

发送机（源端）和接收机（宿端）之间链路容量变化的同步需要通过控制包来实现。每个控制包描述的是下一控制包发送期间的链路状态，提前发送变化状态以便接收机一旦收到变化信息时便立即切换到新的配置。

控制包由不同特定功能的字段组成，控制包包括从  $S_o$  到  $S_k$  以及从  $S_k$  到  $S_o$  发送的信息，如图 5 中所示。

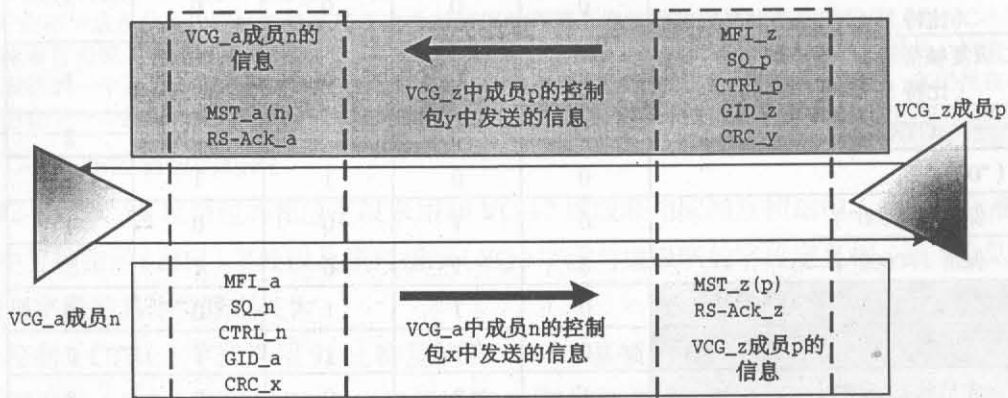


图 5 控制包信息分配

从  $S_o$  到  $S_k$  方向包含的控制包字段为：

- 复帧指示（MFI）字段；
- 序列指示（SQ）字段；
- 控制（CTRL）字段；
- 组识别（GID）比特。

从  $S_k$  到  $S_o$  包含的控制包字段为：

- 成员状态 (MST) 字段;
- 重排序确认 (RS-Ack) 比特。

注: 在 VCG 的所有成员中, MST 和 RS-ACK 是相一致的。

双向都包含的控制包字段为:

- 循环冗余校验 (CRC) 字段;
- 未使用比特保留并设置为“0”。

注: 为了保持一致的时间关系, 假设 LCAS 控制包在  $S_k$  处差分时分延补偿以后才进行相关处理。

### 6.2.2 VC-3-Xv/VC-4-Xv的控制包结构

VC-3-Xv/VC-4-Xv LCAS 控制包的相应字段采用和 VC-3/VC-4 虚级联相同的复帧结构 (参见本标准 5.2.1 节) 来承载, 如表 4 所示。其中复帧指示 (MFI) 字段以及序列指示 (SQ) 字段与虚级联中相关字段的定义相一致 (参见本标准表 1), 而其它控制包字段的具体位置定义为:

- 控制 (CTRL) 字段采用第一级复帧第 3 基帧的 H4 字节的 1~4 比特来标识;
- 组织别 (GID) 比特采用第一级复帧第 4 基帧的 H4 字节的第 4 比特来标识;
- 成员状态 (MST) 字段采用第一级复帧第 9 和第 10 基帧的 H4 字节的 1~4 比特来标识;
- 重排序确认 (RS-Ack) 比特采用第一级复帧第 11 基帧的 H4 字节的第 4 比特来标识;
- CRC 字段采用第一级复帧第 7 和第 8 基帧的 H4 字节的 1~4 比特来标识。

表 4 包含 VC-3-Xv/VC-4-Xv LCAS 控制字段的 H4 字节编码

H4 字节								第一级复帧编号	第二级复帧编号
比特 1	比特 2	比特 3	比特 4	比特 5	比特 6	比特 7	比特 8		
				第一级复帧指示 MFI1 (比特 1~4)					
序列指示 MSB (比特 1~4)				1	1	1	0	14	n-1
序列指示 LSB (比特 5~8)				1	1	1	1	15	
第二级复帧指示 MFI2 MSB (比特 1~4)				0	0	0	0	0	n
第二级复帧指示 MFI2 LSB (比特 5~8)				0	0	0	1	1	
CTRL				0	0	1	0	2	
GID ("000x")				0	0	1	1	3	
保留 ("0000")				0	1	0	0	4	
保留 ("0000")				0	1	0	1	5	
CRC-8				0	1	1	0	6	
CRC-8				0	1	1	1	7	
成员状态 MST				1	0	0	0	8	
成员状态 MST				1	0	0	1	9	
0	0	0	RS_Ack	1	0	1	0	10	
保留 ("0000")				1	0	1	1	11	
保留 ("0000")				1	1	0	0	12	
保留 ("0000")				1	1	0	1	13	
序列指示 SQ MSB (比特 1~4)				1	1	1	0	14	
序列指示 SQ LSB (比特 5~8)				1	1	1	1	15	



表 4 (续)

H4 字节								第一级 复帧 编号	第二级 复帧 编号
比特 1	比特 2	比特 3	比特 4	比特 5	比特 6	比特 7	比特 8		
				第一级复帧指示 MF11 (比特 1~4)					
第二级复帧指示 MF12 MSB (比特 1~4)				0	0	0	0	0	n+1
第二级复帧指示 MF12 LSB (比特 5~8)				0	0	0	1	1	
CTRL				0	0	1	0	2	
0	0	0	GID	0	0	1	1	3	
保留 ("0000")				0	1	0	0	4	
保留 ("0000")				0	1	0	1	5	
C1	C2	C3	C4	0	1	1	0	6	
C5	C6	C7	C8	0	1	1	1	7	
成员状态 MST				1	0	0	0	8	

另外, 在高阶虚级联 VC-3-Xv/VC-4-Xv 的 LCAS 中, 复帧指示与序列指示的关系如表 5 中所示。

表 5 VC-3-Xv/VC-4-Xv LCAS 复帧与成员编号关系

第 2 级复帧编号	成员编号				MST-复帧
0, 32, 64, 96, 128, 160, 192, 224	0	1	2	3	
	4	5	6	7	
1, 33, 65, 97, 129, 161, 193, 225	8	9	10	11	
	12	13	14	15	
30, 62, 94, 126, 158, 190, 222, 254	240	241	242	243	
	244	245	246	247	
31, 63, 95, 127, 159, 191, 223, 255	248	249	250	251	
	252	253	254	255	

注 1: 每个 VC-n-Xv 帧报告 8 个成员状态, 在第 2 级复帧 2ms 帧速的情况下, 256 个成员需要 32 帧, 即若仅有一条返回通路时, 成员状态每隔 64ms 才能刷新一次。  
注 2: 对于表格中成员状态比特的解释是基于这个时刻接收到第 2 级复帧的成员状态字的数值。在 VC-3 /VC-4 的情况下, 这意味着首先第 2 级复帧是从 H4[1-4][0]和 H4[1-4][1]中读取的, 是 0~255 间的一个值, 因此, 该值 (模为 32) 被用作是本表格的一个索引, 以便识别其后立刻从 H4[1-4][8]和 H4[1-4][9]中接收到的成员状态。这仍是在相同的第 1 级复帧中, 但在下一个控制包中

### 6.2.3 VC-12-Xv控制包的结构

VC-12-Xv LCAS 控制包的相应字段采用和 VC-12 虚级联相同的复帧结构 (参见本标准 5.2.2 节) 来承载。其中复帧指示 (MFI) 字段以及序列指示 (SQ) 字段与虚级联的字段定义相一致 (参见本标准表 3), 而其它控制字段的具体位置定义为:

- 控制 (CTRL) 字段采用 32 比特复帧第 12~15 基帧的 K4 字节的第 2 比特来标识;
- 组识别 (GID) 比特采用 32 比特复帧第 16 基帧的 K4 字节的第 2 比特来标识;
- 成员状态 (MST) 字段采用 32 比特复帧第 22~29 基帧的 K4 字节的第 2 比特来标识;
- 重排序确认 (RS-Ack) 比特采用 32 比特复帧第 21 基帧的 K4 字节的第 2 比特来标识;
- CRC 字段采用 32 比特复帧第 30~32 基帧的 K4 字节的第 2 比特来标识。

包含 VC-12-Xv LCAS 控制字段的 K4 (b2) 复帧编码见表 6。

表 6 包含 VC-12-Xv LCAS 控制字段的 K4 (b2) 复帧编码

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
帧指示				序列指示				CTRL				GID	保留 "0000"				RS-Ack	成员状态				C1	C2	C3							
CRC-3																															

另外，在低阶虚级联 VC-12-Xv 的 LCAS 中，复帧指示与序列指示的关系如表 7 所示。

表 7 VC-12-Xv LCAS 复帧与成员编号关系

帧编号	成员编号								MST-复帧
0, 8, 16, 24	0	1	2	3	4	5	6	7	
1, 9, 17, 25	8	9	10	11	12	13	14	15	
2, 10, 18, 26	16	17	18	19	20	21	22	23	
3, 11, 19, 27	24	25	26	27	28	29	30	31	
4, 12, 20, 28	32	33	34	35	36	37	38	39	
5, 13, 21, 29	40	41	42	43	44	45	46	47	
6, 14, 22, 30	48	49	50	51	52	53	54	55	
7, 15, 23, 31	56	57	58	59	60	61	62	63	

注：每个 VC-12-Xv 帧中报告 8 个成员的状态，这样 64 个成员状态的报告则需要帧速为 16ms/帧的 8 帧复帧。因此，若仅有一条返回通路时，成员状态每隔 128ms (8×16ms) 才能刷新一次

6.2.4 复帧指示 (MFI) 字段

在源端，VCG 的所有成员的 MFI 都相同。在宿端，MFI 应该用于为组中所有成员重新对齐净荷。MFI 用于确定相同 VCG 中不同成员之间的差分时延，更多细节参见 ITU-T G.806。

6.2.5 序列指示 (SQ) 字段

该字段包含分配给指定成员的序列编号。相同 VCG 的每个成员分配了一个唯一的序列指示，从 0 开始，和虚级联中定义的序列指示一致。

对于在控制字段中发送 IDLE 状态的成员而言，SQ 在宿端被忽略。VCG 中控制字段发送 IDLE 的成员 SQ 应设置为最高可能的值。

6.2.6 控制 (CTRL) 字段

控制字段用于从源到宿传递信息，其通过源来同步宿并提供组中每个成员的状态，具体的字段定义如表 8 中所示。

表 8 LCAS CTRL 字段定义

值 MSB...LSB	命令	含 义
0000	FIXED	这说明该端使用固定带宽 (非 LCAS 模式)
0001	ADD	该成员将被添加到组
0010	NORM	正常传输
0011	EOS	顺序指示并正常传输的末尾
0101	IDLE	该成员不属于组或准备删除
1111	DNU	不可用 (净荷)，宿端报告了 FAIL 状态

在源端 VCG 所有成员添加到 VCG (发送 CTRL=ADD) 之前，他们初始化时都发送 CTRL=IDLE 信



息。

### 6.2.7 组识别 (GID) 比特

该比特用于识别 VCG。相同 VCG 所有成员的 GID 比特在拥有相同 MFI 值的控制包中其值是相同的。

GID 为接收机提供了验证来源于一个发送机的所有到达成员的方法，其对于控制字段发送 IDLE 的成员无效。GID 内容是伪随机的，但并不要求接收机同步于接收的信息流。使用的伪随机模式是  $2^{15}-1$ 。

### 6.2.8 CRC 字段

为了简化虚级联开销变化的验证过程，采用 CRC 来保护控制包。在每个控制包收到以后进行 CRC 校验，如果校验失败，那么拒绝这些内容。若控制包通过 CRC 校验，则其携带的内容立即被采用。为了简化 MFI 复帧成帧，对于 MFI 单元而言，允许忽略控制包 CRC 校验的结果，这样 LCAS 复帧成帧过程可和非 LCAS 虚级联复帧成帧过程等同地使用 MFI 单元。

#### 6.2.8.1 CRC 乘/除处理

控制包中的比特可看作是二进制多项式系数，控制包传输的第一比特为最高位。特定的 CRC- $n$  [ $n$  为 8 (对应 VC-3-Xv/VC-4-Xv) 或 3 (对应 VC-12-Xv)] 块为控制包中的所有比特乘以  $x^n$  并除以 (模 2) 应用指定的生成多项式后的余数。该余数是最高维数为  $(n-1)$  的多项式。

当要标识整个块的内容为多项式时，块中的第一个比特，即比特 1 应看作为最高比特位。相应地， $C_1$  定义为该余数的最高比特位，而  $C_n$  为余数的最低比特位。

需要注意的是，在进行 CRC 计算时，VC-3-Xv/VC-4-Xv 和 VC-12-Xv 对于控制包标识块内容的处理有所不同。

对于 VC-3-Xv/VC-4-Xv 而言，CRC 计算内容由两个相邻第二级复帧相关内容所组成，如表 4 中所示，第  $n+1$  帧第二级复帧的 CRC 的计算内容包括第  $n$  帧第二级复帧的第 8~15 帧第一级复帧的 H4 字节的比特 1 到比特 4 和第  $n+1$  帧第二级复帧的第 0~5 帧第一级复帧的 H4 字节的比特 1 到比特 4 (第  $n$  帧第二级复帧的第 8 帧的 H4 字节的第一比特为最高位，比特位置依次递减)。若采用多项式  $M(x)$  来表示该数据块，则有

$$M(x) = a_{55}x^{55} + a_{54}x^{54} + \dots + a_1x + a_0 \quad (1)$$

在此指定生成多项式  $G(x)$  为

$$G(x) = x^8 + x^2 + x + 1 \quad (2)$$

那么  $M(x)$  乘以  $x^8$  后其多项式除以 (模 2)  $G(x)$  的多项式的余数结果可表示为 7 次 (或更低次) 多项式  $R(x)$ ，即

$$R(x) = C_1x^7 + C_2x^6 + C_3x^5 + C_4x^4 + C_5x^3 + C_6x^2 + C_7x + C_8 \quad (3)$$

其中系数  $C_1$ ~ $C_8$  分别为 CRC 字段中  $C_1$ ~ $C_8$  的值。

但对于 VC-12-Xv 来说，CRC 计算内容由当前复帧的相关内容所组成，如表 6 中所示，第  $n$  帧 CRC 的计算内容包括第  $n$  帧复帧的第 1~29 子复帧 (第 1 子复帧的 K4 字节的比特 2 为最高位，比特位置依次递减)。若采用多项式  $M(x)$  来表示该数据块，则有

$$M(x) = a_{28}x^{28} + a_{27}x^{27} + \dots + a_1x + a_0 \quad (4)$$

在此指定生成多项式  $G(x)$  为

$$G(x) = x^3 + x + 1 \quad (5)$$

那么  $M(x)$  乘以  $x^3$  后其多项式除以 (模 2)  $G(x)$  的多项式的余数结果可表示为 2 次 (或更低次) 多项式  $R(x)$ , 即

$$R(x) = C_1x^2 + C_2x + C_3 \quad (6)$$

其中系数  $C_1$ - $C_3$  分别为 CRC 字段中  $C_1$ - $C_3$  的值。

### 6.2.8.2 CRC 编码过程

CRC 编码过程如下:

- a) 控制包中的 CRC- $n$  比特替换为二进制 0;
- b) 控制包执行本标准 6.2.8.1 中所阐述的乘/除过程;
- c) 从乘/除过程得到的余数插入到控制包中的 CRC- $n$  位置。

如上面 a) 中所述, 生成的 CRC- $n$  比特并不影响乘/除过程的结果, CRC- $n$  比特位在乘/除运算过程中初始化为 0。

### 6.2.8.3 CRC 解码过程

解码过程如下:

- a) 接收到的控制包执行本标准 6.2.7.1 节中的除法过程[接收到的控制包内容在除法计算中包括 CRC 字段, 即此时执行除法的分子多项式为 63 次 (对于 VC-3-Xv/VC-4-Xv) 或 31 次 (对于 VC-12-Xv)];
- b) 若在解码器中计算的余数为 0, 则认为校验的控制包是无误码的。

### 6.2.9 成员状态 (MST) 字段

该字段包含相同虚级联组中 CTRL 为 ADD/NORM/EOS/DNU 的成员从宿到源传递的状态信息。从宿到源有两种成员状态: OK 或 FAIL (每个成员 1 个状态比特), OK=0, FAIL=1。由于每个控制包仅包含有限的比特位用于表示 MST 字段, 因此该字段扩展到多个控制包中携带, 即多个复帧中携带 MST。

VCG 中成员的数量可以为可分配范围 (例如, 对于 SDH 的高阶虚级联为 0~255, 低阶虚级联为 0~63) 内的任意值 (在非 LCAS 模式中, 接收机功能配置为期望固定数目的成员), 而且可以更改。对于每个成员而言, 宿端使用来自源端的 SQ 编号作为响应源端 MST 的编号。在这种模式下, 源端接收到的 MST 值将总是直接对应用其分配的 SQ 值。

为了便于接收机确定 VCG 中成员的个数, 需要注意的是: 在控制字段中, 最高非失效的有效成员将用于指示序列结束 (EOS)。VCG 可以拥有其它更高 SQ 值但控制字是不可用 (DNU) 状态的成员。

在 VCG 宿端, 所有成员初始化时均报告 MST=FAIL。当某个成员带有控制字段为 ADD (或在其添加后发送 EOS 或 NORM, 或网络故障恢复以后发送 DNU) 的控制包收到时, 将产生 MST=OK 的转变。所有未使用成员的 MST 和拥有控制字段为 IDLE 成员的 MST 将被设置为 FAIL。

### 6.2.10 重排序确认 (RS-Ack) 比特

当重新编排在 CTRL 字段状态为 NORM、DNU 或者 EOS 的发送成员的序列指示时, 或者当在宿端检测到这些成员的编号改变时, 每个 VCG 通过反转 (即从 “0” 到 “1” 或者从 “1” 到 “0”) RS-Ack 比特来发送通知到源端。具体而言, 触发 RS\_Ack 比特翻转的原因如下所列 (参见本标准附录 D):

——对于 VCG 中任何 SQ 值的改变 (在宿端, 对于控制字段为 DNU/NORM/EOS 的成员可以检测到 SQ 改变);

——CTRL=“ADD”→CTRL=“EOS”和/或 CTRL=“ADD”→CTRL=“NORM”（添加一个或更多的成员）；

——CTRL=“NORM”（或者“EOS”）→CTRL=“IDLE”（减少带宽或删除成员）；

——CTRL=“DNU”→CTRL=“IDLE”。

需要注意的是，在管理界面上执行 ADD 命令（即当发生 CTRL = IDLE → CTRL = ADD 转换）时，并没有 RS\_Ack 传输。事实上，仅当隶属于 VCG 的成员序列的改变在宿端能检测到时，RS\_Ack 才被触发。在添加新成员的第一阶段（IDLE 到 ADD 状态的转换），即使 SQ 分配可以发生，但并不影响 VCG，因此并不需要 RS\_Ack。

RS-Ack 比特仅在 VCG 所有成员的状态已经确认并且序列改变已经发生后才能被翻转。在 RS-Ack 没有发送到源端的情形下，宿端和源端之间的同步通过激活（在请求重新排序过程当中，或者改变组中成员编号期间）RS-Ack 超时来实现。超时的到期相当于在源端检测到 RS-Ack 比特的反转（参见附录 D）。RS-Ack 比特的反转或者 RS-Ack 超时的到期说明需考虑新的 MST 值。这意味着在包含 RS-Ack 的控制包中收到的 MST 值以及在随后的控制包中收到的 MST 值对应于新的序列指示。源端可使用该反转作为指示源端发起的改变已经接受和完成，并将开始接收新的 MST 状态信息。

只有当收到 RS-Ack 或者对于当前活动的更改请求的 RS-Ack 超时已经到期时，在 VCG 中才有新的变化，如在 VCG 中增删成员。

### 6.3 LCAS操作过程

#### 6.3.1 VCG容量增加

##### 6.3.1.1 添加成员

当添加一个成员时，该成员总是被分配一个比当前在 CTRL 代码中含有 EOS 或 DNU 状态的最高序列指示大一的序列指示。当添加多个成员时，这些成员每个都使用惟一的序列指示，这样一来对于每个添加成员将有惟一的 MST 响应。

在 ADD 命令之后，响应 MST = OK 的第一个成员将被分配为下一个最高的序列指示并更改其 CTRL 代码为 EOS，与此同时，当前占用最高序列指示成员更改其 CTRL 代码为 NORM（或保持为 DNU）。

当发送 CTRL=ADD 用于发起新成员加入时，将连续发送 CTRL=ADD，直到收到 MST=OK 为止。

在多于一个（例如， $x$  个）成员加入的情形下，此时同时接收到多于 1 个成员的 MST=OK，而且序列指示将是动态提供的，它们是当前最高序列指示（CTRL 代码为 EOS 或 DNU）之后  $x$  个连续的顺序标号。新添加的成员的 CTRL 代码为 EOS 或 DNU。对于当前序列指示最高成员而言，CTRL 代码将从 EOS 转变为 NORM（或保持为 DNU），与此同时最高序列指示的新成员的 CTRL 代码转变为 EOS。而其它新成员的 CTRL 代码将被设置为 NORM。

##### 6.3.1.2 添加成员净荷

添加成员的最后步骤就是对于该成员在虚级联开销控制包中的控制字段发送 NORM 或 EOS。对于新成员而言，包含净荷的第一容器帧将是紧随包含该成员 NORM/EOS 控制字段的控制包的最后比特（即 CRC）容器帧的容器帧。

#### 6.3.2 VCG容量减少

##### 6.3.2.1 LCAS 过程因网络故障临时删除成员

###### 6.3.2.1.1 临时删除成员

当 CTRL 字段为 NORM 或 EOS 的成员在网络中发生失效并在宿端（如 MSU\_L, TSD 等）检测到时，

宿端将在发送的 MST 中设置该成员状态为 FAIL。为了限制嵌套保护机制中保护切换动作的次数，宿端将通过拖延 (HO) 定时器将 MST 为 FAIL 的报告延迟一段时间。源端一旦检测到成员的 MST 为 FAIL，则将该成员的 CTRL 字段从 NORM 替换为 DNU (如果该成员失效之前的 CTRL 字段为 NORM)，或者从 EOS 替换为 DNU (如果该成员失效之前的 CTRL 字段为 EOS)。此时拥有最高序列指示的有效成员将在 CTRL 字段中发送 EOS。

#### 6.3.2.1.2 临时删除成员净荷

临时删除成员净荷时主要考虑以下两种情形：

——对于接收到 MSU\_L 情形而言，临时删除成员的最后步骤就是从 VCG 中删除该成员。在宿端，检测到 MSU\_L 缺陷以后立即启动删除操作。在源端，包含删除成员净荷的最后容器帧将是包含第一个 DNU 控制字段的控制包最后比特的容器帧。随后的容器帧将在净荷区置为全零。一旦在宿端接收到 DNU 控制字段，该相应成员的净荷将不被用于重建原始的 VCG 净荷。

——对于接收到 TSD 情形而言，临时删除成员的最后步骤就是从 VCG 中删除该成员。在宿端，该成员的净荷区将继续用于重建原始的 VCG 净荷。该成员净荷区的比特误码必须由 VCG 宿端从服务层到客户层的适配功能来处理。在源端，包含删除成员净荷的最后容器帧将是包含第一个 DNU 控制字段的控制包最后比特的容器帧。随后的容器帧将在净荷区置为全零。一旦在宿端接收到控制字段为 DNU，该成员的净荷区将从 VCG 中删除。

#### 6.3.2.1.3 临时删除成员复原

在宿端检测到导致临时删除成员的缺陷已清除时，宿端将发送的 MST 字段中设置该成员状态为 OK。为避免间歇波动缺陷导致的不稳定的影响，宿端将通过等待恢复 (WTR) 定时器把该成员 MST 为 OK 的报告拖延一段时间。源端一旦检测到该成员的 MST 为 OK，则或者将该成员控制字段从 DNU 替换为 NORM (如果该成员失效之前的 CTRL 字段为 NORM)，或者将该成员控制字段从 DNU 替换为 EOS，同时把前一成员的 CTRL 字段从 EOS 替换为 NORM (如果该成员失效之前的 CTRL 字段为 EOS)。

#### 6.3.2.1.4 临时删除成员净荷复原

临时删除恢复的最后步骤就是重新使用该成员的净荷区。该成员包含净荷数据的第一容器帧将是紧随该成员 CTRL 代码中第一个包含 NORM 或者 EOS 的控制包最后比特的容器帧的容器帧。

### 6.3.2.2 永久删除成员

#### 6.3.2.2.1 删除成员

当有成员被永久删除时，需要重新编号和更新。如果被永久删除的成员占用 VCG 中最高序列指示，那么占用次高序列指示的有效成员将改变其控制包中的控制字段为 EOS，同时被永久删除成员的控制包设置控制字段为 IDLE。倘若被永久删除成员占用组中最高序列指示并同时控制字段 CTRL 中发送的是 DNU，那么组中其它成员的序列指示和控制字段将不发生变化。如果被永久删除成员的序列指示不是组中的最高值，那么在被永久删除成员和最高序列指示成员之间的其它成员 (包括最高序列指示成员) 将更新控制包中的序列指示，同时更改被永久删除成员的控制字段状态，即从 NORM 更改为 IDLE。

需要注意的是，如果永久删除成员是由宿端首先发起并且删除的成员并不拥有最高成员编号，那么宿端仍然保持为最高序列指示成员将收到的 SQ 编号高于宿端新指派的大小 (这种情况持续到源端也删除这些成员为止)，这并不是故障情形。另外，如果有效成员的永久删除是由宿端发起，并且宿端的 REMOVE 定时器设置为“0”，这对于宿端数据重组将带来一些冲击，具体细节见附录 C.4.1。冲击持续的时间为从宿端删除成员开始 (开始发送 MST=FAIL)，到宿端接收到来自源端的 DNU 为止。

### 6.3.2.2.2 删除成员净荷

当通过在虚级联开销的控制包中发送 IDLE 控制字段来删除成员时,删除成员包含净荷数据的最后一容器帧将是包含带有 IDLE 控制字段的控制包的最后比特的容器帧。

## 7 LCAS 和非 LCAS 之间互通

### 7.1 LCAS发送机和非LCAS接收机之间互通

LCAS 发送机和非 LCAS 模式的非 LCAS 接收机可协同工作而不需做任何特殊考虑。LCAS 发送机将按照本标准 5.2 中所指定的方法来设置 MFI 和 SQ。接收机将忽略除 MFI 和 SQ 之外的所有比特,并根据接收到的 MFI 和 SQ 值来从虚容器中提取净荷。

这样,从宿端返回到源端的成员状态将一直是 MST=OK (即返回开销中的 MST 相关比特都为“0”)。

### 7.2 非LCAS发送机和LCAS接收机之间互通

LCAS接收机期望CTRL字段不是“0000”以及期望正确的CRC,但非LCAS发送机将在LCAS CTRL 字段和CRC字段发送“0000”。因此当LCAS接收机和非LCAS发送机互通时,当收到CTRL和CRC均为“0000”时,它将:

- 除了MFI和SQ之外,忽略其它所有信息;
- 使用针对虚级联定义的 MFI 和 SQ 缺陷检测。

## 8 性能要求

### 8.1 最大可补偿差分时延

最大可补偿差分时延是指相同 VCG 中不同成员进行异径传输后重新组合成 VCG 时所能容忍的相对传输时延的最大值。目前暂定义该值大于或等于 125 $\mu$ s。

### 8.2 RS\_Ack超时时间(可选)

RS\_Ack 超时时间是指当 VCG 源端在重新排序成员状态为 NORM、DNU 或者 EOS 的成员时,等待 VCG 宿端响应 RS\_Ack 比特反转的最长时间。其值大小待研究。

### 8.3 等待恢复(WTR)时间(可选)

等待恢复(WTR)时间是指 LCAS 状态机在服务层的故障(如 SDH 层的告警、误码等)消失后,为防止瞬态效应影响而采取的定时器延时控制。其值大小待研究。

### 8.4 拖延(HO)时间(可选)

拖延(HO)时间是指 LCAS 状态机在服务层出现故障(如 SDH 层的告警、误码等)时,为防止瞬态效应影响和协调嵌套的多层保护倒换机制而采取的定时器延时控制。其值大小待研究。

### 8.5 业务受损时间(可选)

基于 LCAS 协议的业务受损时间指的是,在使能 LCAS 协议的前提下,出现故障时采用 LCAS 引入的业务受损时间等于 LCAS 故障检测时间、LCAS 拖延时间、LCAS 成员增删完成时间等之和,其值大小待研究。

### 8.6 宿端删除(RMV)等待时间(可选)

宿端删除等待时间指的是,在宿端启动 REMOVE 命令以后,等待源端响应的的时间。其值大小待研究。

## 9 管理要求

### 9.1 性能统计

性能统计要求支持以下项目：

- a) VCG 正常成员个数 (可选)；
- b) VCG 故障成员个数 (可选)；
- c) VCG 成员总个数 (可选)。

### 9.2 告警

告警要求支持以下项目：

- a) 复帧丢失 (LOM)；
- b) 序列指示失配 (SQM) (可选)；
- c) VCG 失效 (可选)；
- d) 对齐丢失 (LOA) (可选)；
- e) 成员不可用 (MND) (可选)；
- f) SQ 编号矛盾 (可选)；
- g) 源端丢失部分容量 (PLCT) (可选)；
- h) 源端丢失全部容量 (TLCT) (可选)；
- i) 宿端丢失部分容量 (PLCR) (可选)；
- j) 宿端丢失全部容量 (TLCR) (可选)；
- k) 组识别 (GID) 比特错误 (可选)；
- l) EOS 丢失 (可选)；
- m) RS-ACK 超时 (可选)；
- n) RMV 超时 (可选)；
- o) CTRL 字未定义 (可选)。

### 9.3 配置

配置要求支持以下项目：

- a) VCAT 建立/删除配置；
- b) 源端 LCAS 使能/去使能配置 (可选)；
- c) 宿端 LCAS 使能/去使能配置 (可选)；
- d) VCG 中成员个数的配置；
- e) 虚容器 (VC) 大小的配置 (VC-12 和 VC-3/VC-4 等)；
- f) VCG 成员的增加或删除的配置；
- g) HO 超时时间的配置 (可选)；
- h) WTR 超时时间的配置 (可选)；
- i) RMV 超时时间的配置 (可选)。

附录 A  
(规范性附录)  
相邻级联

A.1 VC-4相邻级联 (VC-4-Xc, X=4, 16, 64, 256)

如图 A.1 所示, 一个 VC-4-Xc 提供了一个 X 个 C-4 容器的净荷区并由 VC-4-X 结构来表示。第一列是整个 VC-4-Xc 的 POH (例如 BIP-8 校验覆盖了 VC-4-Xc 所有的  $261 \times X$  列), 第二列到第 X 列是固定填充字节。

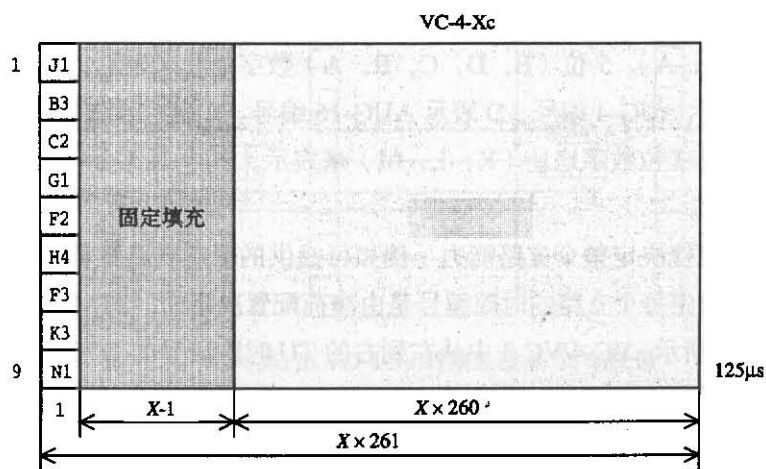


图 A.1 VC-4-Xc 结构

VC-4-Xc 在 STM-N 信号中相邻的 X 个 AU-4 中传输, VC-4-Xc 的第一列总是位于第一个 AU-4。第一个 AU-4 的指针指示了 VC-4-Xc 的 J1 字节的位置, 第二到 X 的指针被设置成级联指示, 表明连续的级联净荷。指针调整的实现是由 X 个级联的 AU-4 共同完成, 同时使用了  $X \times 3$  个填充字节。

当 X=4 时, VC-4-Xc 的净荷容量是 599 040 kbit/s; 当 X=16 时, VC-4-Xc 的净荷容量是 2 396 160 kbit/s; X=64 时, VC-4-Xc 的净荷容量是 9 584 640 kbit/s; X=256 时, VC-4-Xc 的净荷容量是 38 338 560 kbit/s。

附录 B  
(规范性附录)

相邻级联和虚级联互通

B.1 AU/TU编号要求

VC 互通的基础是采用相同的映射方式, 并且在 STM-N 帧中使用相同编号的时隙。

一个 STM-N 帧由  $N \times 270$  列(编号从 1 到  $N \times 270$ )组成, 开始的  $N \times 9$  列包含 SOH 和 AU-4/AU-4-Xc 指针, 剩余的  $N \times 261$  列包含高阶数据净荷(高阶支路)。高阶净荷列可以通过 2 位(B, A)、3 位(C, B, A)、4 位(D, C, B, A)、5 位(E, D, C, B, A) 数字地址来表示, 其中 A 表示 AU-3 编号、B 表示 AUG-1 编号、C 表示 AUG-4 编号、D 表示 AUG-16 编号、E 表示 AUG-64 编号。在 AU-4 结构帧的情况下, 净荷列可以通过 3 位数字地址(K, L, M)来表示, 其中 K 表示 TUG-3 编号、L 表示 TUG-2 编号、M 表示 TU-1 编号。

为提供一种简单、便捷决定整个支路能力(例如可提供的低阶支路数量)的方法, 净荷列被分配了一个时隙编号, 每一个帧中每个支路的时隙编号是由净荷配置决定的。STM-N 中从左到右的 AU 时隙编号如 G707 图 7-12~7-26 所示, VC-4/VC-3 中从左到右的 TU 时隙编号如 G707 图 7-27~7-29 所示。AU/TU 可以采用这些图中所示的连续序列的时隙编码, 或采用复用序列的编号。例如: VC-4 中 TU-12 时隙编号 1、2、3、4 对应复用序列指示为 (1, 1, 1)、(2, 1, 1)、(3, 1, 1)、(1, 2, 1), 如图 B.1 所示。

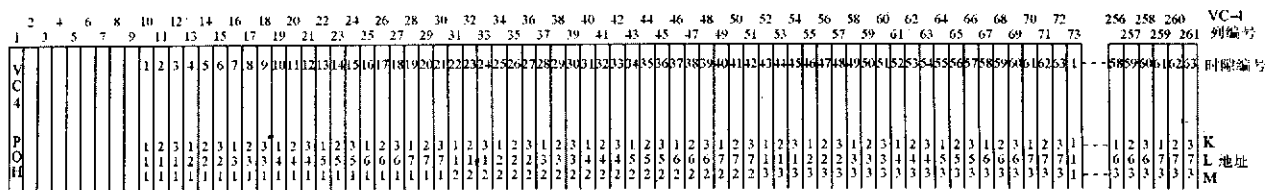


图 B.1 VC-4 中 TU-12 时隙编号示意

AU/TU 的编号要求详见 G707 第 7.3 节。

B.2 相邻级联和虚级联的互通(可选)

B.2.1 VC-4-Xc到VC-4-Xv的互通

当要求从 VC-4-Xc 到 VC-4-Xv 互通时(允许的 X 值为 4、16 和 64, 对于更高的 X 值有待研究), VC-4-Xc 净荷区(C-4-Xc) 将被插入到 VC-4-Xv 净荷中, 映射关系如表 B.1 所示。

表 B.1 VC-4-Xc 到 VC-4-Xv 的净荷映射

VC-4-Xc 列	VC-4-Xv	
	VC-4 编号	VC-4 列
X+1	1	2
...	...	...
2×X	X	2
2×X+1	1	3
...	...	...
261×X	X	261

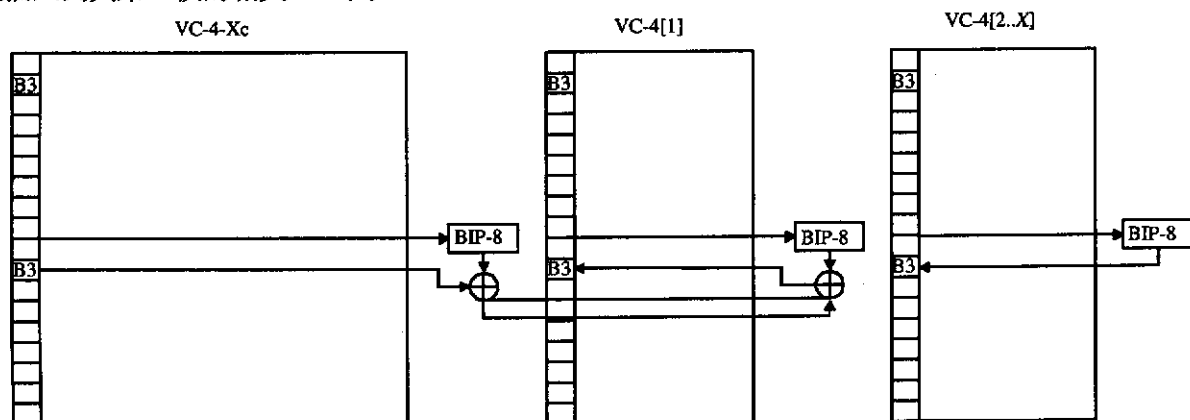
此时涉及以下 POH 的处理:

a) J1: VC-4-Xc 的该字节将被插入到 VC-4-Xv 的第 1 个 VC-4, 对于 VC-4-Xv 所有其它的 VC-4, 如果踪迹插入是使能的, 则将插入一个单独的 J1 踪迹字节; 如果踪迹插入不是使能的, 则将插入 VC-4-Xc



的 J1 字节。

b) B3: BIP-8 将计算 VC-4-Xc 的第  $n-1$  帧, 并和第  $n$  帧相关的 B3 相比较, 以判定比特误码的个数。对于 VC-4-Xv 而言, 第  $n-1$  帧每一个独立的 VC-4 都将计算 BIP-8。对于 VC-4-Xv 的第 1 个 VC-4, 由于 VC-4-Xc 中检测到比特误码的原因, 在第  $n$  帧中插入相关的 B3 前, 由于 VC-4-Xc 被检测到错误比特时 BIP-8 相应比特将反转, 这可以通过异或处理来完成, 如图 B.2 中所示。其它 VC-4 的 BIP-8 将不做任何修改插入到其第  $n$  帧的相关 B3 中。



⊕ 异或

图 B.2 VC-4-Xc 到 VC-4-Xv 转换过程中 B3 的处理

c) C2: VC-4-Xc 的该字节将被插入到 VC-4-Xv 信号的所有独立的 VC-4 中。

d) G1[1-4]: VC-4-Xc 该字节的比特 1 到比特 4 (REI) 将被插入到 VC-4-Xv 的第 1 个 VC-4 该字节的比特 1 到比特 4, VC-4-Xv 其它 VC-4 该字节的比特 1 到比特 4 将被置 0。

e) G1[5]: VC-4-Xc 该字节的比特 5 (RDI) 将被插入到 VC-4-Xv 的所有 VC-4 该字节的比特 5 中。

f) G1[6-7]: 作为加强的 RDI 来使用, 可选。

g) G1[8]: VC-4-Xc 该字节的比特 8 将被插入到 VC-4-Xv 的所有 VC-4 该字节的比特 8 中。

h) F2: VC-4-Xc 的 F2 字节将被插入到 VC-4-Xv 信号的第 1 个 VC-4 中, VC-4-Xv 其它 VC-4 的 F2 字节将被置为 00h。

i) F3: VC-4-Xc 的 F3 字节将被插入到 VC-4-Xv 信号的第 1 个 VC-4 中, VC-4-Xv 其它 VC-4 的 F3 字节将被置为 00h。

j) K3: VC-4-Xc 的 K3 字节将被插入到 VC-4-Xv 信号的第 1 个 VC-4 中, VC-4-Xv 其它 VC-4 的 K3 字节将被置为 00h。

k) N1[1-4]: 如果 VC-4-Xc 该字节的比特 1 到比特 4[输入错误统计 (IEC)] 包含编码 “1110” (输入 AIS), VC-4-Xv 所有 VC-4 该字节的比特 1 到比特 4 将被置为 “1110”; 如果 VC-4-Xc 该字节的比特 1 到比特 4 (IEC) 包含编码 “0000” (部分 TC 未装载), VC-4-Xv 所有 VC-4 该字节的比特 1 到比特 4 将被置为 “0000”。否则, VC-4-Xc 该字节的比特 1 到比特 4 将被插入到 VC-4-Xv 第 1 个 VC-4 该字节的比特 1 到比特 4 中, 并且 VC-4-Xv 其它 VC-4 该字节的比特 1 到比特 4 将被置为一个为 0 的 IEC (“1001”)。

l) N1[5-8]: VC-4-Xc 该字节的比特 5 到比特 8 将被拷贝至 VC-4-Xv 所有 VC-4 该字节的比特 5 到比特 8 中。

m) H4: 复帧处理 (H4[5-8], H4[1-4][0-1]), 同 5.2.1。

n) H4: 序列号处理 (H4[1-4][14-15]), 如 ITU-T G.707 中的规定, 每一个 VC-4 中将插入一个独立的序列号 SQ, 对于 VC-4[y] 的序列号是  $y-1$ 。

o) H4: H4[1-4][2-13]这些比特保留作为将来使用, 将置为“0000”。

**B.2.2 VC-4-Xv到VC-4-Xc的互通**

当要求从 VC-4-Xv 到 VC-4-Xc 互通时 (允许的 X 值为 4、16 和 64, 对于更高的 X 值有待研究), 需要进行错误监视处理、复帧处理、序列号处理、对齐处理和净荷处理等几个方面的处理。现分别描述如下:

错误监视处理由每个独立 VC-4 来完成, 包括:

——J1: 将从 J1 字节中恢复接收路径踪迹标识 (RTTI), 并使能接收路径踪迹标识以便网管使用。需要注意的是, 如果在 VC-4-Xv 到 VC-4-Xc 的互通过程中, VC-4[2...X]没有配置独立的踪迹字节, VC-4[2..X]期望的踪迹字节将设置为与第 1 个 VC-4 期望的踪迹字节相同, 或者将关闭这些 VC-4 踪迹管理。

——C2: 将恢复该信号标记比特, 并通过监视 C2 是否为“1111111”编码来检测 VC-AIS 条件。

复帧处理包括 H4[5~8]、H4[1~4][0~1]等开销字节的处理, 具体处理过程同 5.2.1 节。

序列号处理包括 H4[1~4]、[14~15]等开销字节的处理, 将从第 14、15 复帧 H4 字节的比特 1 到 4 中恢复接收到的序列指示 (SQ), 并且使能 SQ 以便网管使用。如果在连续  $m$  个 (3~10) 第 1 级复帧中接收到相同数值的序列指示, 新的序列指示则是可接受的。

在对齐处理过程中, 如果任何独立 VC-4 的缺陷功能均未激活, 则将独立的 VC-4 对齐为两级复帧的开始。对齐处理将包含至少 125 $\mu$ s 的差分延迟。如果对齐处理成功, 则将执行以下净荷和开销的相关处理。

在净荷处理过程中, VC-4-Xv 净荷区 (C-4-Xc) 将被插入到 VC-4-Xc 净荷中, 如表 B.2 中所示。

表 B.2 VC-4-Xv 到 VC-4-Xc 的净荷映射

VC-4-Xv		VC-4-Xc 列
VC-4 列	VC-4 编号	
2	1	$X + 1$
...	...	...
2	$X$	$2 \times X$
3	1	$2 \times X + 1$
...	...	...
261	$X$	$261 \times X$

此时涉及对以下 POH 的处理:

a) J1: VC-4-Xv 第 1 个 VC-4 的 J1 字节将被插入到 VC-4-Xc 中。

b) B3: 此时应计算 VC-4-Xv 中每个 VC-4 第  $n-1$  帧的 BIP-8 值, 并和每个 VC-4 的第  $n$  帧相关的 B3 相比较, 以判定比特误码的个数。VC-4-Xc 所有 VC-4 帧的比特误码都将被一起相加, 结果的限值为 8。在第  $n$  帧中插入 B3 之前, 首先计算 VC-4-Xc 第  $n-1$  帧的 BIP-8 值, 然后根据前面计算的比特误码个数反转该 BIP-8 相关比特, 最后将该值插入到第  $n$  帧的 B3 中, 如图 B.3 所示。

c) C2: VC-4-Xv 第 1 个 VC-4 的该字节插入到 VC-4-Xc 中。

d) G1[1~4]: VC-4-Xv 中所有 VC-4 的该字节的比特 1 到比特 4 (REI) 将被一起相加, 结果的限值为 8, 并且将插入到 VC-4-Xc 该字节的比特 1 到比特 4 中。

e) G1[5]: 如果 VC-4-Xv 中任何一个 VC-4 该字节的比特 5 (RDI) 包含编码“1”, 则 VC-4-Xc 该字节的比特 5 将置为“1”。

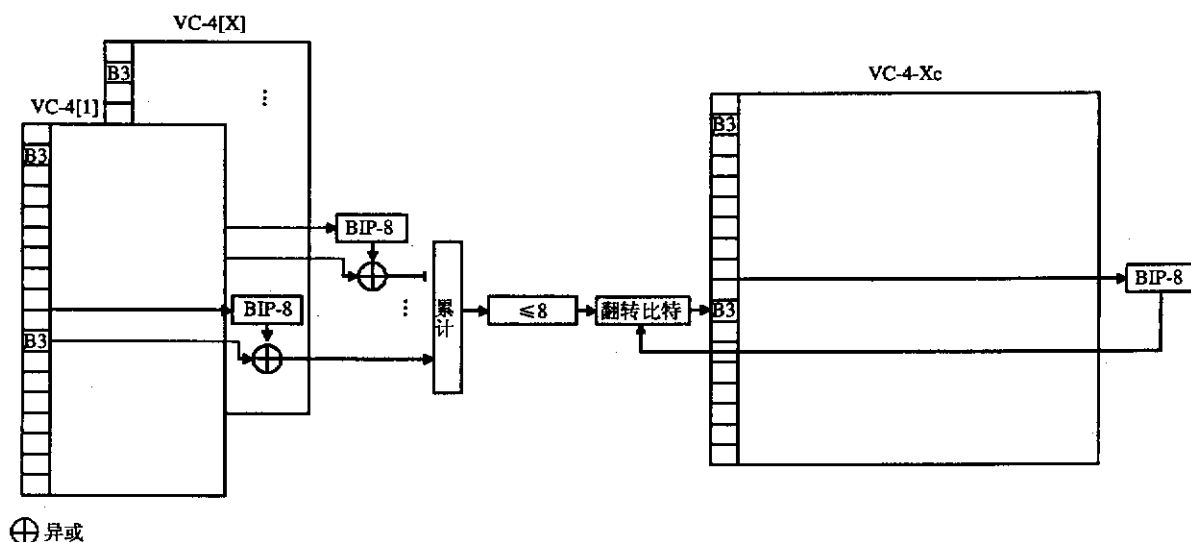


图 B.3 VC-4-Xv 到 VC-4-Xc 转换过程中 B3 的处理

- f) G1[6~7]: 作为增强的 RDI 来使用, 可选。
- g) G1[8]: VC-4-Xv 第 1 个 VC-4 该字节的比特 8 将被插入到 VC-4-Xc 该字节的比特 8 中。
- h) F2: VC-4-Xv 第 1 个 VC-4 的 F2 字节将被插入到 VC-4-Xc 中。
- i) F3: VC-4-Xv 第 1 个 VC-4 的 F3 字节将被插入到 VC-4-Xc 中。
- j) K3: VC-4-Xv 第 1 个 VC-4 的 K3 字节将被插入到 VC-4-Xc 中。
- k) N1[1~4]: 如果 VC-4-Xv 中任何 VC-4 该字节的比特 1 到比特 4 (IEC) 包含编码“1110” (输入 AIS), VC-4-Xc 该字节的比特 1 到比特 4 将被置为“1110”; 如果 VC-4-Xv 第 1 个 VC-4 该字节的比特 1 到比特 4 (IEC) 包含编码“0000” (部分 TC 未装载), VC-4-Xc 该字节的比特 1 到比特 4 将被置为“0000”。否则, VC-4-Xv 所有 VC-4 的 IEC 数值 (比特 1 到比特 4) 将被一起相加, 结果限值为 8, 并且将被作为 IEC 插入到 VC-4-Xc 该字节的比特 1 到比特 4 中。
- l) N1[5~8]: VC-4-Xv 第 1 个 VC-4 该字节的比特 5 到比特 8 将被插入到 VC-4-Xc 该字节的比特 5 到比特 8 中。
- m) H4: VC-4-Xc 的该字节将置为 0。
- 关于 VC-4-Xv 和 VC-4-Xc 之间互通处理有关的其它更多细节, 需满足 ITU-T G.783 中有关规定。

附 录 C  
(规范性附录)  
LCAS 协议

### C.1 LCAS协议

LCAS 是单向操作机制。如果要求双向增加或删除成员,则需要在相反方向重复一次操作,但这些双向操作是相互独立的,因此并不要求相互同步。当 VCG 中的成员传输无比特误码时,该机制允许在管理系统的控制下业务无损地增加或减小带宽。另外,LCAS 将自动地从 VCG 中临时删除失效成员。当失效条件消失后,LCAS 将自动添加该成员到 VCG 中。一般而言,通道层失效导致的成员删除对于虚级联组承载的业务并不是无损的,而故障修复后的自动添加对于承载业务而言则是无损的。

在该模型中,采用三个参数来描述虚级联组的大小:

——参数  $X_M$ , 用来说明虚级联组的最大成员个数。在不同传送网络技术的特定定义中限制该参数的大小(如 VC-4-Xv/VC-3-Xv 中为 256, 而 VC-12-Xv 中为 64), 而且在具体的实现中可能限制为更小的值。

——参数  $X_P$ , 用来说明虚级联组中已指派的成员个数。每个 ADD[i] ( $i$  为成员编号) 命令的完成将使得  $X_P$  加 1, 而每个 REMOVE[i] ( $i$  为成员编号) 命令的完成将使得  $X_P$  减 1。另外, 存在如下关系:  
 $0 \leq X_P \leq X_M$ 。

——参数  $X_A$ , 用来说明在单个成员失效的情况下, 虚级联组中受 LCAS 协议自动添加或删除成员而受影响的实际成员个数。存在如下关系:  $0 \leq X_A \leq X_P \leq X_M$ 。

每个参数可进一步分开描述: 当源(发送)端或者宿(接收)端处理需要指定时, 那么“T”或“R”则分别添加到术语中。例如,  $X_{PT}$  是在源(发送)端方向指派的成员个数, 而  $X_{AR}$  则是宿(接收)端方向的成员实际个数。

对于每个成员 ( $X_{MT}$  个), 在源端都存在状态机并且是以下 5 个状态之一:

- a) IDLE: 该成员没有被指派到级联组。
- b) NORM: 该成员被指派到 VCG 中, 而且从源端到宿端的通路正常。
- c) DNU: 该成员被指派到 VCG 中, 但是从源端到宿端的通路失效。
- d) ADD: 该成员正在加入到 VCG 之中。
- e) REMOVE: 该成员正在被从 VCG 中删除之中。

对于每个成员 ( $X_{MR}$  个), 在宿端存在状态机并且是以下 3 个状态之一:

- a) IDLE: 该成员在 VCG 中没有指派。
- b) OK: 该成员携带的信号没有出现失效情况(如 MSU\_L), 或者是收到添加该成员的请求并已获得确认; 当该成员信号出现劣化时(如 TSD), 成员保持在 OK 状态。
- c) FAIL: 该成员携带的信号经历了一些失效条件, 或者收到删除该成员的请求并已确认。

对于所有  $X_{MT}$  个源端和  $X_{MR}$  个宿端成员, 这些状态机是并发运行的。

为了说明在状态框图描述中可能出现的情况, 本节中采用了以下一些符号约定:

——从源端往宿端转发以下 5 类控制消息。成员将总是转发这些消息中的一种(因此总是有  $X_{MT}$  个消息被发送)。这些消息隶属于发送消息的成员:

- a)  $F_{IDLE}$  说明该容器当前不是 VCG 中的成员, 而且也没有未处理完的 ADD 请求;
- b)  $F_{ADD}$  请求添加该成员到 VCG 中;

- c)  $F_{\text{DNU}}$  说明该成员的净荷在 VCG 中没有使用;
- d)  $F_{\text{EOS}}$  说明在 VCG 中激活的成员是拥有最高序列指示的成员;
- e)  $F_{\text{NORM}}$  说明该成员是组中的正常成员, 而且也不拥有最高序列指示。

—— $C_{\text{EOS}}$  和  $C_{\text{NORM}}$  表示从成员 ( $i$ ) 到其前一序列成员 ( $i-1$ ) 的消息 (仅针对源端), 以说明一旦出现请求, 成员 ( $i-1$ ) 发送的控制字段应该改变。

—— $R_{\text{FAIL}}$  和  $R_{\text{OK}}$  表示从宿端到源端关于在宿端的所有成员状态的消息。所有宿端成员的状态通过每个成员的控制包返回到源端。源端可以从不同的成员中读取状态信息, 例如, 从成员 No.1 中读取信息, 若其不可用, 则从成员 No.2 中读取相同信息。一旦没有可用的返回带宽, 源端将使用最后收到的有效状态。

—— $M_{\text{ADD}}$  和  $M_{\text{REMOVE}}$  表示来自管理系统的消息, 用于添加或删除成员。删除操作将影响特定的成员。添加成员将总是发生在 VCG 的末端并被赋予新的最高序列指示。

—— $R_{\text{RS\_ACK}}$  表示一个比特, 用于确认在宿端检测到 VCG 中成员顺序的重新编排或者改变。该确认用于同步源端和宿端, 以便于消除网络时延的影响。由于在接受增加成员或删除成员请求时需要重编序列指示, 接收到的成员状态在一段时间不能使用, 而这段时间的长短主要取决于发送延时和成帧延时。

—— $C_{\text{RSQ}}$  表示从成员 ( $i$ ) 到其前一序列成员 ( $i+1$ ) 的消息 (仅针对源端), 以说明成员 ( $i+1$ ) 的序列指示将减小 1。

—— $S_{\text{RSQ}}$  表示从源端和宿端的成员状态机到 VCG 状态机的消息, 以说明序列指示的变化已经发生。

LCAS 协议在以下的状态框图中描述, 其中具有详细的状态转换图。

为了避免源端和宿端之间序列指示和相应接收的远端状态之间可能出现的不对齐现象, 在 VCG 中指派的成员个数  $X_p$  仅可在管理命令下改变。

在  $MSU\_L$  信号之前收到的序列指示将用于报告成员状态, 但其净荷并不用于原始信号的重建。如果失效成员被管理者永久删除, 那么其余成员将重新编号。对于网络中不可修复的失效因素导致的失效成员的替换必须通过 REMOVE 和 ADD 过程来进行。

本节描述的 LCAS 协议状态框图图例如图 C.1 中所示。

## C.2 LCAS协议分割

LCAS 协议的实现可以分割为:

- 部分协议在 VCG 源端实现;
- 部分协议在 VCG 宿端实现。

需要注意的是, 流程图中从源端到宿端的信息是基于 VCG 中每个成员的信息, 也即 SQ、CTRL、CRC 和 MFI, 而流程图中从宿端到源端的信息是对于 VCG 所有成员的公有定义。

利用流程图可以对 LCAS 协议的实现进一步分割为 4 部分:

a) 部分 1: 该部分实现源端发送 VCG 中每个成员的信息 (也即 SQ、CTRL、CRC 和 MFI) 到宿端的功能。SQ 信息在 VCG 成员之间交换, 同时 SQ 信息也发送到接收部分以控制成员 MST 的分配。

b) 部分 2: 该部分实现宿端接收来自源端的每个成员的信息并发送 SQ 和成员状态到下一协议分割部分 (部分 3) 的功能。

c) 部分 3: 该部分实现宿端发送关于 VCG 中所有成员信息的功能, 也即把 VCG 成员状态作为整体发送, 同时确认在 VCG 序列编号中检测到了变化。

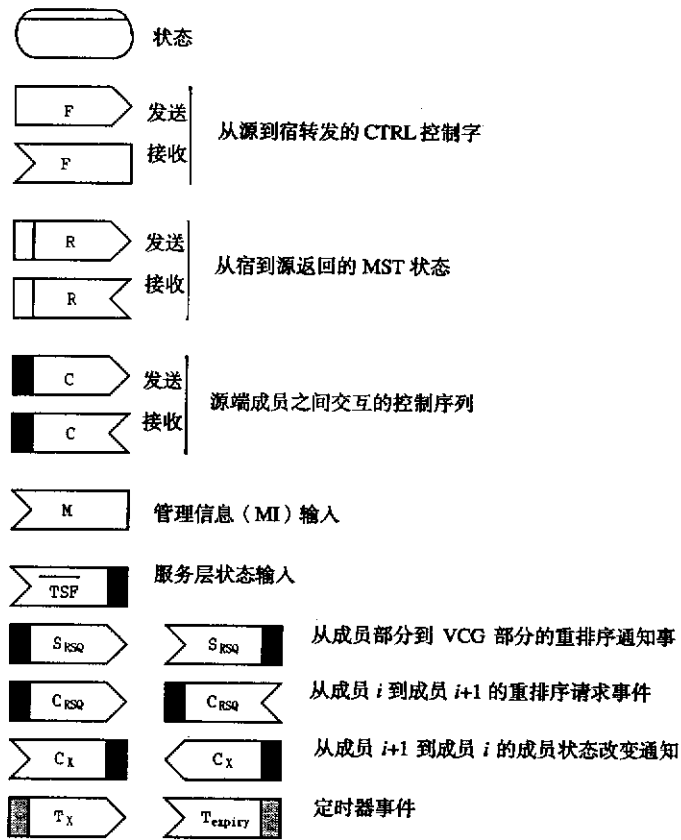


图 C.1 状态图图例

d) 部分 4: 该部分实现源端接收关于 VCG 所有成员信息的功能, 也即接收 VCG MST 和 RS-Ack, 并分配 MST 到每个 VCG 成员功能处理部分 (也即部分 1)。

图 C.2 中描述了不同协议分割部分以及这些部分之间的流程图信息, 图 C.3 中说明了 LCAS 协议不同分割部分之间事件的交互过程。

注: 图 C.2 与 ITU G.7042 的图 A.2 表征的内容相同, 但图 C.2 增加了对于协议分割部分的具体标注。

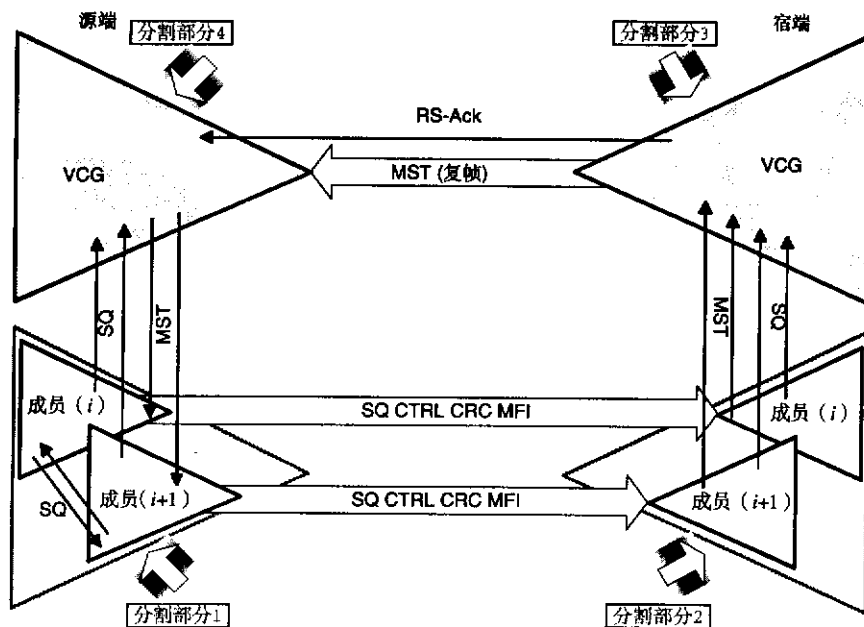


图 C.2 LCAS 协议分割

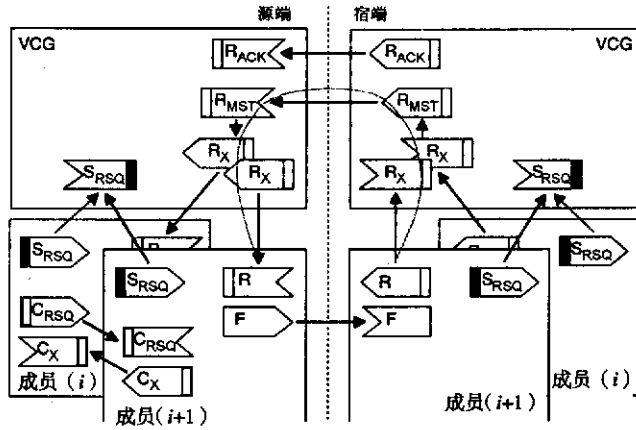
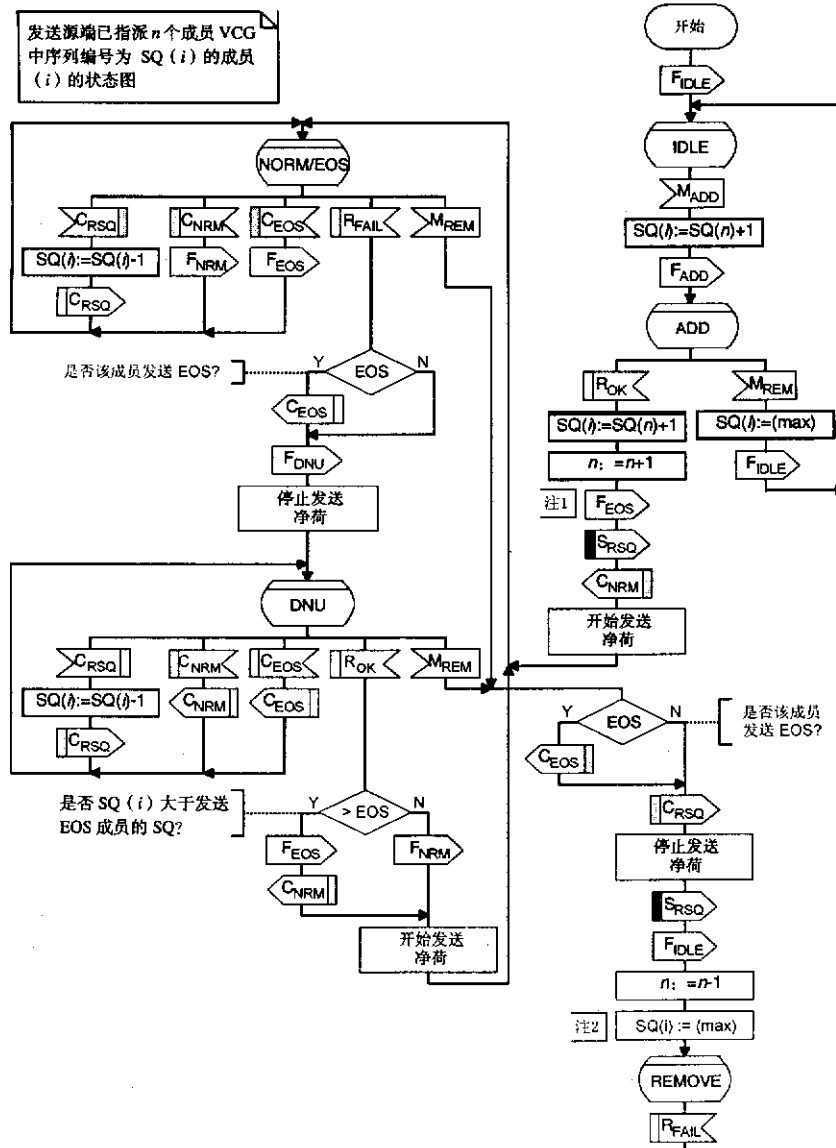


图 C.3 LCAS 协议事件流程图

### C.3 VCG中成员状态图

VCG 成员的源端状态图如图 C.4 中所示，宿端状态图如图 C.5 中所示，其中状态图的描述按照 ITU-T Z.100 规定进行。



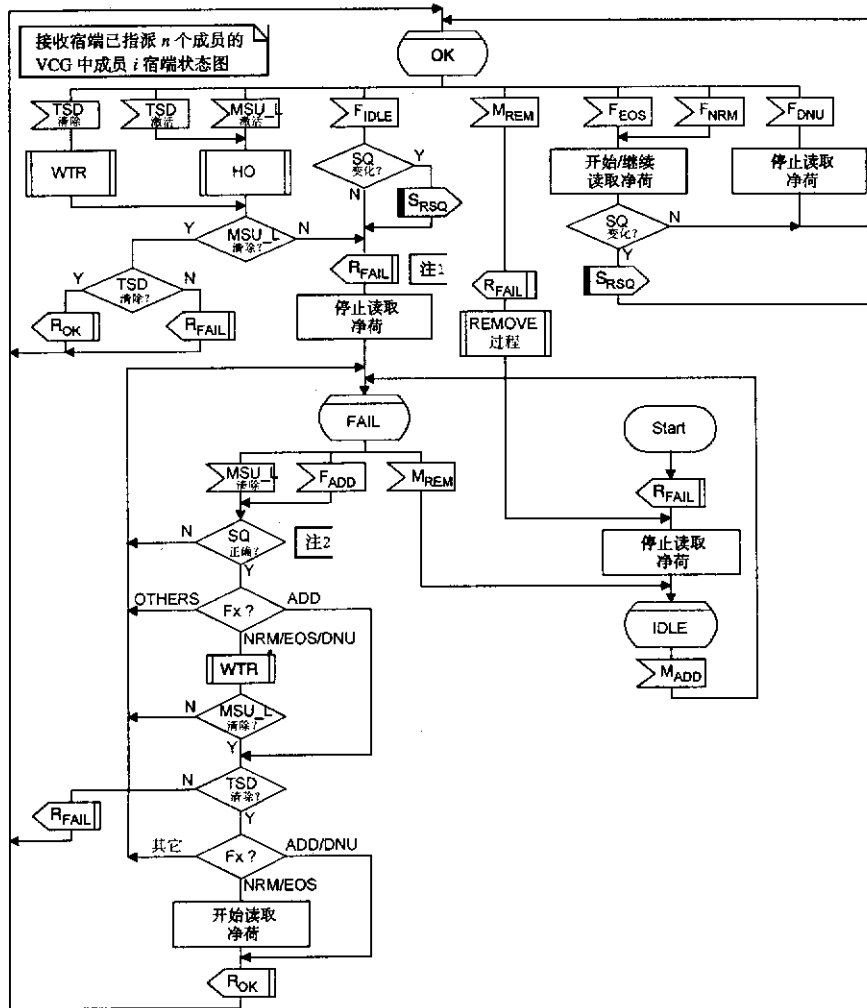
注 1: 在单个成员添加的情形时, 将发送 F<sub>EOS</sub>。否则, 在多个成员同时添加时, 激活的最高序号的成员将发送 F<sub>EOS</sub>, 其它新添加的成员发送 F<sub>NORM</sub>。

注 2: 删除成员  $i$  ( $0 \leq i < X_p$ ) 的 SQ 将被设置为最高的可能值, 成员  $i+1, \dots, X_p-1$  等的 SQ 被重新排列为  $i, \dots (X_p-2)$ 。

注 3: 在整个 VCG 中, RS\_Ack 为共有处理过程。

注 4:  $n \geq 0$  (本附录中  $n$  取值范围一致, 下文不再提及),  $SQ(0) = -1$ 。

图 C.4 源端状态图



注 1: 对于接收到 CTRL 为 IDLE 的成员, 宿端没有特定的 SQ 值。宿端根据 G806 附录 B 中对于无效 SQ 的生成规则, 在此生成 MST 为 FAIL。

注 2: 该判断用于核实当前成员的 SQ 编号与拥有 OK 状态的其它成员相比是否惟一。如果收到的 SQ 是惟一的 (即没有被 OK 状态的任何其它成员使用), 则遵循 Y 流程, 否则遵循 N 流程。

注 3: 对于特定成员 ( $i$ ), “拖延”和“等待恢复”过程是不可能同时有效的。

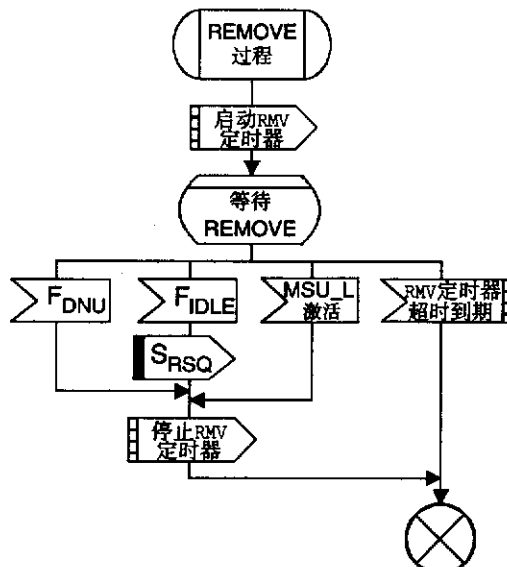
图 C.5 宿端状态图



C.4 过程状态图

C.4.1 REMOVE过程

该过程描述 REMOVE 定时器的激活和去激活过程，以便允许从宿端发起无损删除 VCG 成员。详细过程如图 C.6 所示。



注 1：“等待 REMOVE” 状态仅为暂时状态，用于获取源端已认可宿端删除成员的确切信息。只有该确认消息收到以后，成员净荷才被丢弃。这样，在 VCG 中源端发起的其它可能的更改将被排除在外。

注 2:在 RMV 定时器设置为 0 时,REMOVE 过程停止,在图 C.4 的宿端状态图中 REMOVE 过程忽略(即跳过 REMOVE 过程直接进入下一状态操作),此时在宿端发起的成员删除将导致 VCG 承载的数据临时性丢失。

图 C.6 REMOVE 过程

C.4.2 RS\_Ack过程

该过程描述 RS\_Ack 检测过程，用于验证接收到的 MST。单个成员激活的 RS\_Ack 处理过程是整个 VCG 的共有处理。详细过程如图 C.7 所示。

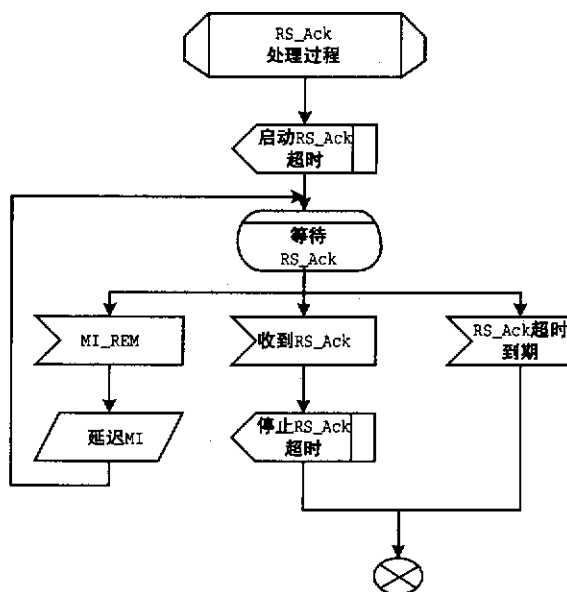


图 C.7 RS\_Ack 过程

注意在状态框图中：“等待 RS\_Ack” 状态仅是源端在接受新的 MST 值分配之前需要确认 MST 的暂时状态。这样，在 VCG 中源端发起的其它可能的更改将被排除在外。

C.4.3 WTR过程

该过程用于描述等待恢复定时器的激活和去激活过程，以便避免如 ITU-T G.808.1 中所描述的瞬态告警的影响。详细过程如图 C.8 所示。

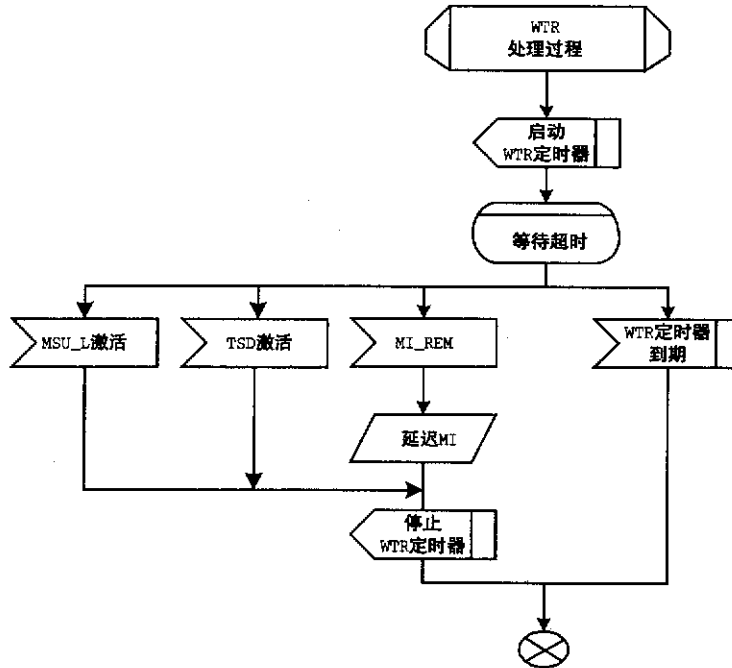


图 C.8 WTR 过程

C.4.4 HO过程

该过程描述了拖延定时器（HO）的激活和去激活处理过程，以便于限制如 G.808.1 中所描述的嵌套保护中倒换动作的次数。图 C.9 提供了该过程详细的 SDL 框图。

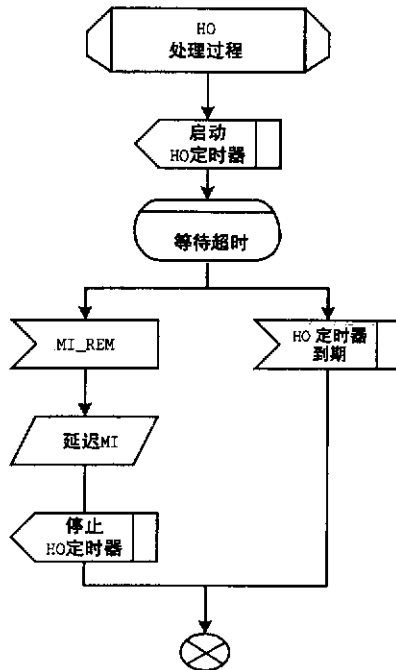


图 C.9 HO 过程

在拖延期间，有一些特殊情况将导致带宽更改时有损承载的业务，对于这些情况的详细分析参见附录 E。

### C.5 VCG状态图

#### C.5.1 源端VCG接收状态图

对于整个 VCG 而言，在源端接收侧状态机的状态将是以下两种状态之一：

a) 处理 MST。从某个成员中接收到的 MST 信息将被处理并分配到独立的成员中去；

b) 等待 RS-ACK。一个或多个源端成员状态机已指示序列编号的改变已经发生。只有当宿端通过发送 RS-ACK 来确认收到 CTRL 变化或者 RS-ACK 定时器超时到期时，MST 处理才能开始。详细的源端 VCG 接收状态图如图 C.10 所示。

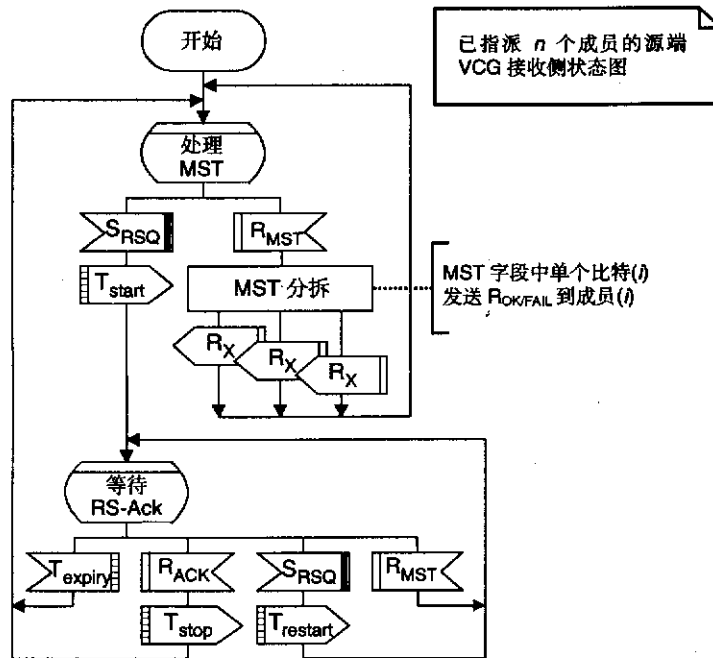


图 C.10 源端 VCG 接收状态图

#### C.5.2 宿端VCG发送状态图

对于整个 VCG 而言，在宿端发送侧状态机只有一种状态——组装 MST，即从每个成员中接收到的 MST 信息被收集到 MST 复帧并在所有成员上发送。一个或多个成员检测到的序列编号的改变将被发送到源端。详细的宿端 VCG 发送状态图如 C.11 所示。

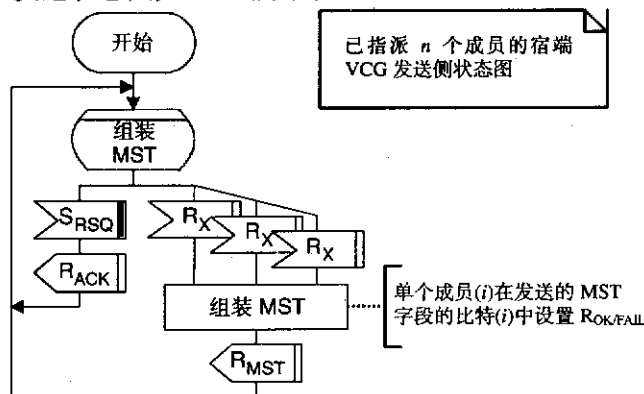


图 C.11 宿端 VCG 发送状态图

附录 D  
(资料性附录)  
LCAS 时序图

D.1 编号方法

虚级联组中的成员排序为  $0 \sim (n-1)$ ,  $n$  为组中所有成员的个数。

D.2 配置

当新容器指派为组中成员时, 必须为其分配如下内容:

- a) CTRL 设置为 IDLE (该代码说明新容器还没有使用)。
- b) SQ 设置为大于在 CTRL 代码中拥有 EOS 的当前最高序号成员的值。当 CTRL=IDLE 时 (没有使用) 该 SQ 值不被解释。建议 SQ 设置为最大可能的值, 但由于该最大值与实现技术相关, 因此在此不可能指出具体精确的值。在以下示例中, 值 (max) 用于指示最大值。
- c) GID 设置为当前虚级联组的组 ID。
- d) MST 设置为 1 (FAIL = 1, OK = 0)。

D.3 命令

D.3.1 增加VCG带宽 (ADD命令)

D.3.1.1 添加: (ADD) 在最后成员之后添加多个成员

本附录以成员个数为  $n$  的 VCG 组最后一个成员之后添加两个新成员为例来说明该添加过程, 其响应时序过程如图 D.1 和表 D.1 所示。

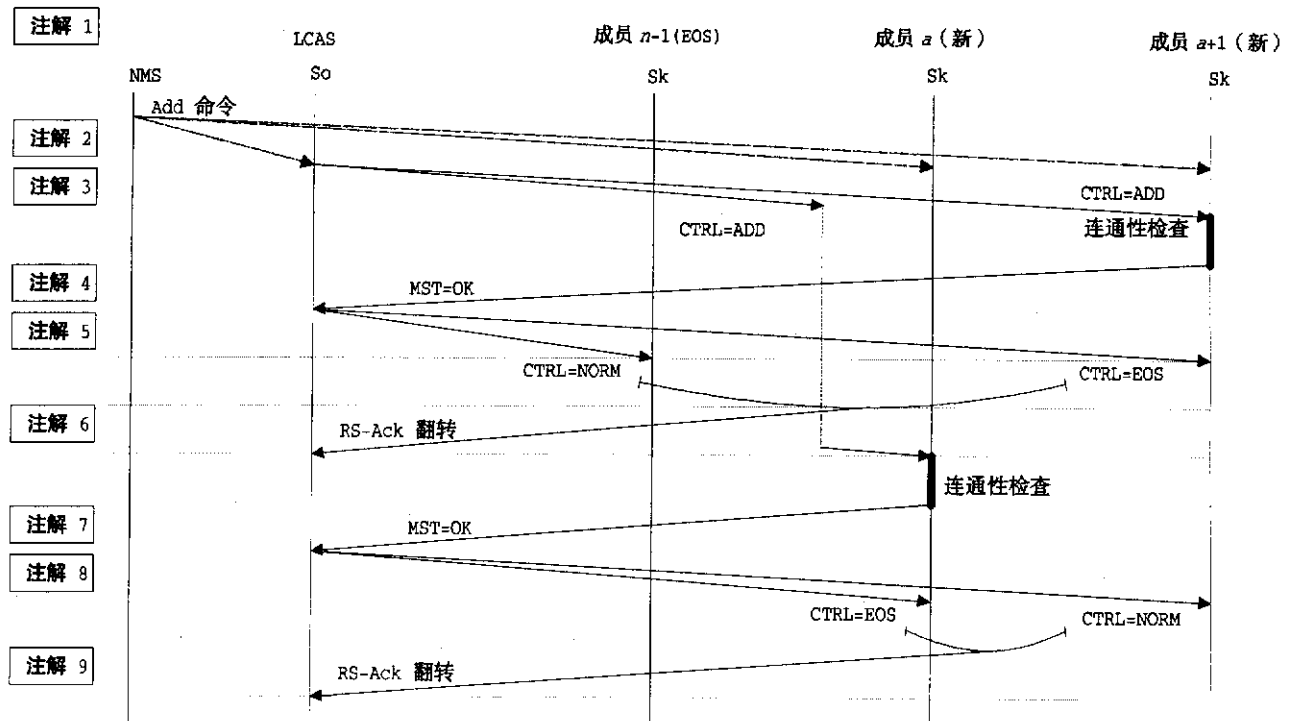


图 D.1 添加多个成员过程

表 D.1 添加多个成员时成员状态变化

注解	命令交互过程说明	成员 $n$			成员 $a$ (新)			成员 $a+1$ (新)			RS_Ack
		CTRL	SQ	MST	CTRL	SQ	MST	CTRL	SQ	MST	
1	初始情况	EOS	$n-1$	OK	IDLE	(max)	FAIL	IDLE	(max)	FAIL	0
2	NMS 发送 Add 命令到 So 和 Sk 的 LCASC	EOS	$n-1$	OK	IDLE	(max)	FAIL	IDLE	(max)	FAIL	0
3	So ( $a$ ) 发送 CTRL=ADD 和 SQ= $n$ ; So ( $a+1$ ) 发送 CTRL=ADD 和 SQ= $n+1$	EOS	$n-1$	OK	ADD	$n$	FAIL	ADD	$n+1$	FAIL	0
4	Sk ( $a+1$ ) 发送 MS=OK 到 So	EOS	$n-1$	OK	ADD	$n$	FAIL	ADD	$n+1$	OK	0
5	So ( $n-1$ ) 发送 CTRL = NORM; So ( $a+1$ ) 发送 CTRL=EOS 和 SQ= $n$	NORM	$n-1$	OK	ADD	$n+1$	FAIL	EOS	$n$	OK	0
6	由于序列发生变化, 故 RS-Ack 比特翻转	NORM	$n-1$	OK	ADD	$n+1$	FAIL	EOS	$n$	OK	1
7	Sk ( $a$ ) 发送 MST=OK 到 So	NORM	$n-1$	OK	ADD	$n+1$	OK	EOS	$n$	OK	1
8	So ( $a$ ) 发送 CTRL=EOS; So ( $a+1$ ) 发送 CTRL=NORM	NORM	$n-1$	OK	EOS	$n+1$	OK	NORM	$n$	OK	1
9	由于序列发生变化, 故 RS-Ack 比特翻转	NORM	$n-1$	OK	EOS	$n+1$	OK	NORM	$n$	OK	0

注 1: 在该示例的时序图中, 新成员 ( $a+1$ ) 在新成员 ( $a$ ) 之前以 MST=OK 响应, 其实这种响应次序是任意的, 并且第一个响应 MST=OK 的成员将被分配 SQ= $n$ , 然后下一个响应 MST=OK 的成员分配的 SQ= $n+1$ 。如果由于某些原因成员被添加以后在超时期间内没有响应 MST=OK 的话, 那么源端的 LCASC 报告该成员失效。

注 2: RS\_Ack 的初始值 “0” 可随意设置, 在示例中与 LCAS 机制相关的仅仅是 RS\_Ack 比特的反转。

### D.3.2 减小 VCG 带宽 (REMOVE 命令)

#### D.3.2.1 减小: (REMOVE) 计划删除多个成员但不包括最后成员

本附录以成员个数为 6 的 VCG 组中删除成员 4 和 5 为例来说明该删除过程, 其响应时序过程如图 D.2 和表 D.2 所示。需要说明的是, 本节示例的删除过程是从源端开始的, 对于宿端开始的删除过程, 与 D.3.3 节中网络故障导致的减小带宽过程类似。

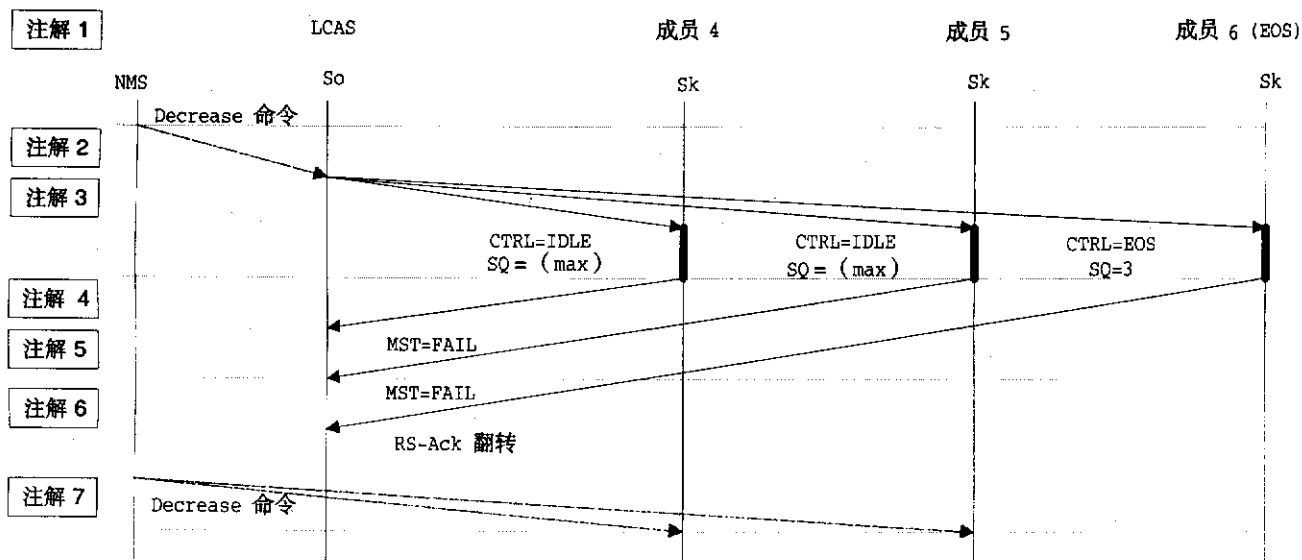


图 D.2 6 个成员中计划删除成员 4 和 5 过程

表 D.2 6 个成员中计划删除成员 4 和 5 时成员状态变化

注解	命令交互过程说明	成员 4			成员 5			成员 6			RS_Ack
		CTRL	SQ	MST	CTRL	SQ	MST	CTRL	SQ	MST	
1	初始情况	NORM	3	OK	NORM	4	OK	EOS	5	OK	0
2	NMS 发送 Decrease 命令到 So 的 LCASC	NORM	3	OK	NORM	4	OK	EOS	5	OK	0
3	So (3) 发送 CTRL=IDLE, SQ=(max) So (4) 发送 CTRL=IDLE, SQ=(max) So (5) 发送 SQ=3	IDLE	=(max)	OK	IDLE	=(max)	OK	EOS	3	OK	0
4	Sk (被删除) 发送 MST=FAIL 到 So	IDLE	=(max)	FAIL	IDLE	=(max)	OK	EOS	3	OK	1
5	Sk (被删除) 发送 MST=FAIL 到 So	IDLE	=(max)	FAIL	IDLE	=(max)	FAIL	EOS	3	OK	1
6	由于序列发生变化, 故 RS-Ack 比特翻转	IDLE	=(max)	FAIL	IDLE	=(max)	FAIL	EOS	3	OK	1
7	NMS 发送 Decrease 命令到 Sk LCASC	IDLE	=(max)	FAIL	IDLE	=(max)	FAIL	EOS	3	OK	1

在所有要删除的成员中, 源端 LCASC 设置为 CTRL=IDLE。

注 1: 组中其它成员的 CTRL 并不改变。

示例说明的是来自源端 LCASC 同时发生的 IDLE 命令删除了两个成员。在宿端一旦接收到 IDLE 命令, 则立即停止使用“已删除的”成员来重组数据。

然而来自宿端的响应可能不是同时发生的, 但由于 IDLE 命令拥有相同的 MFI 值, 所以这并不影响宿端的操作。从宿端到源端的响应当然是简单地确认该成员在宿端不再使用, 另外如果有需要, NMS 则解除对于该成员的配置。

注 2: 删除的成员如表 D.2 的注解 7 中所示解除指派。

在 REMOVE 功能中 SQ 调节的一般规则如下:

- a) 所有删除的成员重新分配大于发送 EOS 控制字段的成员 SQ 的 SQ 值, 即最高可能的值 (max);
- b) 所有保留的必需的成员重新分配连续的 SQ (从 SQ=0 开始)。

表 D.3 是对此过程最好的描述。

表 D.3 成员删除前后 SQ 变化情况

SQ 值变化	VC 编号						
	A	B	C	D	E	F	G
准备删除的成员			√	√			√
成员删除之前的 SQ 值	0	1	2	3	4	5	6
成员删除之后的 SQ 值	0	1	=(max)	=(max)	2	3	=(max)

注 3: RS\_Ack 的初始值“0”可随意选择, 在示例中与 LCAS 机制相关的仅仅是 RS\_Ack 比特的反转。

### D.3.2.2 减少: (REMOVE) 计划删除单个最后成员

计划删除单个最后成员的响应时序过程如图 D.3 和表 D.4 中所示。

表 D.4 计划删除单个 (最后) 成员时成员状态变化

注解	命令交互过程说明	成员 n-1			成员 n			RS_Ack
		CTRL	SQ	MST	CTRL	SQ	MST	
1	初始情况	NORM	n-2	OK	EOS	n-1	OK	0
2	NMS 发送 Decrease 命令到 So LCASC	NORM	n-2	OK	EOS	n-1	OK	0
3	So (删除) sends CTRL=IDLE, SQ=(max), So (n-2) 发送 CTRL=EOS	EOS	n-2	OK	IDLE	(max)	OK	0
4	由于序列发生变化, 故 RS-Ack 比特翻转	EOS	n-2	OK	IDLE	(max)	FAIL	1
5	同时 Sk (删除) 发送 MST=FAIL	EOS	n-2	OK	IDLE	(max))	FAIL	1
6	NMS 发送 Decrease 命令到 Sk LCASC	EOS	n-2	OK	IDLE	(max)	FAIL	1

注 1: 删除的成员如表 D.4 中的注解 6 中所示解除指派。

注 2: 在相同控制包中发送 RS\_Ack 反转, 以更新最近的 MST 值。

注 3: RS\_Ack 的初始值“0”可随意设置, 在示例中与 LCAS 机制相关的仅仅是 RS\_Ack 比特的反转。

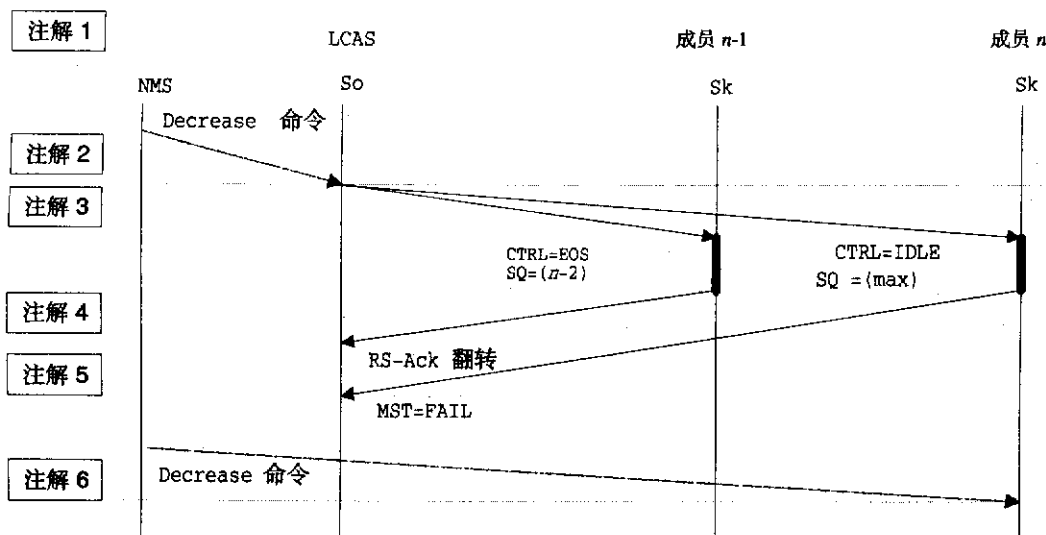


图 D.3 计划删除单个（最后）成员过程

D.3.3 由于故障（DNU命令）减小VCG带宽

D.3.3.1 单个最后成员故障导致减小带宽（DNU）

单个最后成员故障导致减小带宽的响应时序过程如图 D.4 和表 D.5 所示。

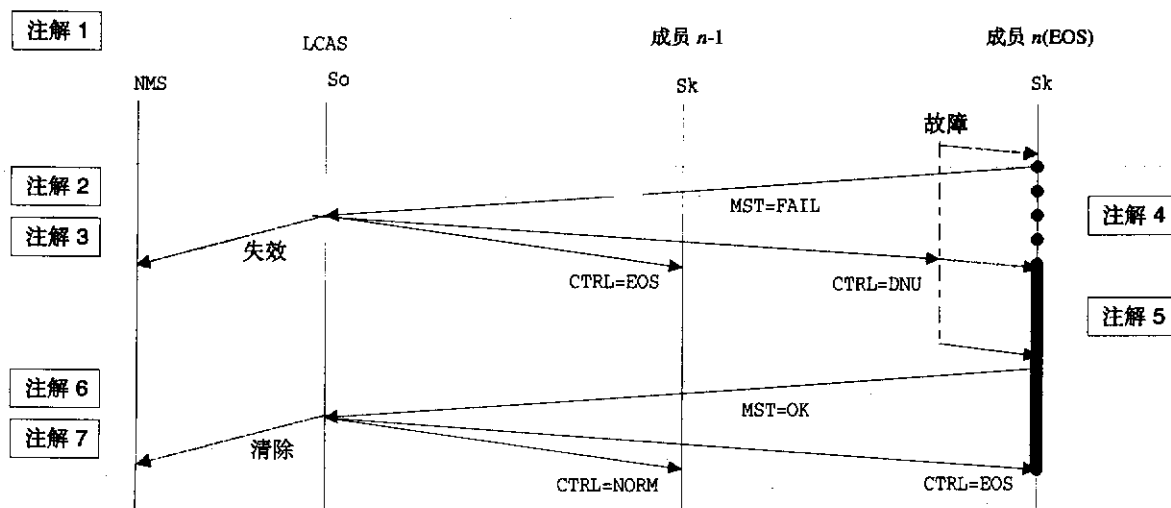


图 D.4 单个（最后）成员网络故障导致带宽减小过程

表 D.5 单个（最后）成员网络故障导致带宽减小时成员状态变化

注解	命令交互过程说明	成员 n-1			成员 n (EOS)			RS_Ack
		CTRL	SQ	MST	CTRL	SQ	MST	
1	初始情况	NORM	n-2	OK	EOS	n-1	OK	0
2	Sk (故障成员) 发送 MST=FAIL 到 So	NORM	n-2	OK	EOS	n-1	FAIL	0
3	So (故障成员) 发送 DNU; So (故障成员前一成员) 发送 EOS	EOS	n-2	OK	DNU	n-1	FAIL	0
4	参见本表之后文字说明	EOS	n-2	OK	DNU	n-1	FAIL	0
5	参见本表之后文字说明	EOS	n-2	OK	DNU	n-1	FAIL	0
6	网络故障清除, MST=OK 发送到 So	EOS	n-2	OK	DNU	n-1	OK	0
7	So (故障成员) CTRL 发送 EOS; So (故障成员前一成员) CTRL 发送 NORM	NORM	n-2	OK	EOS	n-1	OK	0

源端的 LCASC 在有故障的成员上设置 CTRL=DNU，同时在故障成员前的前一个成员上设置 CTRL=EOS。为了更清楚地解释表 D.5 中各个成员的状态变化过程，下面对于表中的一些注解进行进一步的解释。

——注解 3

虽然带宽和包含 EOS 的成员已发生变化，但由于该变化是临时的，因此并不触发 RS-Ack。

——注解 4

如果宿端检测到 MSU\_L，宿端立即开始级联组的重组，此时重组的对象为 CTRL 是 NORM 和 EOS 成员。在一段时间内（从宿端到源端的传输时间 + 源端的处理时间 + 源端到宿端的传输时间），由于所有成员在故障之前都发送，因此此时重组的数据是错误的。

如果宿端检测到 TSD，宿端将继续使用该成员的净荷。该成员净荷区的比特误码必须由 VCG 宿端的服务层到客户层的适配功能来处理。在一段时间内（从宿端到源端的传输时间 + 源端的处理时间 + 源端到宿端的传输时间），由于所有成员在故障之前都发送，因此此时重组的数据是错误的。

——注解 5

源端将停止在错误成员（由于它们回告为 MST=FAIL 并随后设置失效成员为 DNU）上发送数据，并仅在其它 CTRL 为 NORM 和 EOS 的成员上发送数据。

对于故障是 MSU\_L 的情形，从 CTRL=DNU 到达宿端直到宿端再收到 CTRL=EOS 这段时间内，VCG 带宽是减少的。对于故障是 TSD 的情形，从 CTRL=DNU 到达宿端以后，VCG 带宽就减少了。宿端 LCASC 并不知道什么时候数据完整性重新建立，这些操作在数据层处理。

——注解 7

当失效成员故障修复后，CTRL 从 DNU 变为 EOS 时，宿端将重新使用该成员净荷来重组数据。

注 1：如果失效的通路在故障清除之前进行了计划减小带宽而删除，那么宿端将无法在失效成员的控制包中看到该变化。这样一来，RS-Ack 在该计划减小带宽中并不翻转，同时源端将依赖于 RS-Ack 超时来继续处理 MST。VCG 的带宽将不受影响。

注 2：RS\_Ack 的初始值“0”可随意设置，在示例中与 LCAS 机制相关的仅仅是 RS\_Ack 比特的反转。

D.3.3.2 单个非最后成员故障导致减小带宽 (DNU)

单个非最后成员故障导致减小带宽的响应时序过程如图 D.5 和表 D.6 所示（VCG 总共 5 个成员，成员 4 出现故障）。

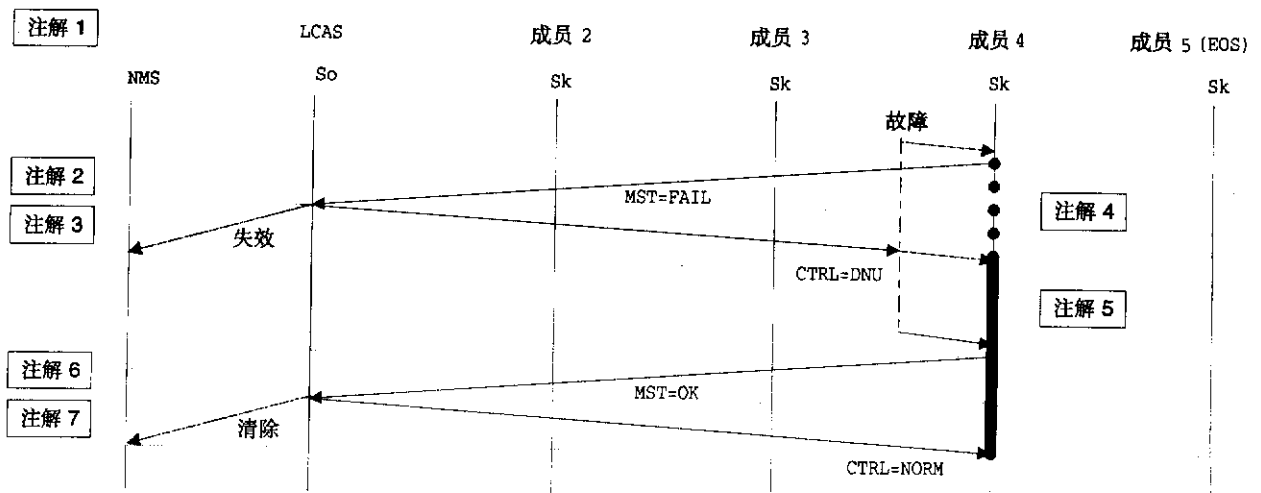


图 D.5 单个非最后成员网络故障导致减小带宽过程



表 D.6 单个非最后成员网络故障导致减少带宽时成员状态变化

注解	命令交互过程说明	成员 2			成员 3			成员 4			成员 5 (EOS)			RS_Ack
		CTRL	SQ	MST	CTRL	SQ	MST	CTRL	SQ	MST	CTRL	SQ	MST	
1	初始情况	NORM	1	OK	NORM	2	OK	NORM	3	OK	EOS	4	OK	0
2	Sk (故障成员) 发送 MST=FAIL 到 So	NORM	1	OK	NORM	2	OK	NORM	3	FAIL	EOS	4	OK	0
3	So (故障成员) 发送 CTRL=DNU	NORM	1	OK	NORM	2	OK	DNU	3	FAIL	EOS	4	OK	0
4	参见本表之后文本	NORM	1	OK	NORM	2	OK	DNU	3	FAIL	EOS	4	OK	0
5	参见本表之后文本	NORM	1	OK	NORM	2	OK	DNU	3	FAIL	EOS	4	OK	0
6	网络故障清除, 故障成员发送 MST=OK 到 So	NORM	1	OK	NORM	2	OK	DNU	3	OK	EOS	4	OK	0
7	故障成员的 CTRL 从 DNU 改变为 NORM	NORM	1	OK	NORM	2	OK	NORM	3	OK	EOS	4	OK	0

为了更清楚地解释表 D.6 中各个成员的状态变化过程, 下面对于表中的一些注解进行进一步的解释。

——注解 4

如果宿端检测到 MSU\_L, 宿端立即开始级联组的重组, 此时重组的对象为 CTRL 是 NORM 和 EOS 成员。在一段时间内 (从宿端到源端的传输时间 + 源端的处理时间 + 源端到宿端的传输时间), 由于所有成员在故障之前都发送数据, 因此此时重组的数据是错误的。

如果宿端检测到 TSD, 宿端将继续使用该成员的净荷。该成员净荷区的比特误码必须由 VCG 宿端的服务层到客户层的适配功能来处理。在一段时间内 (从宿端到源端的传输时间 + 源端的处理时间 + 源端到宿端的传输时间), 由于所有成员在故障之前都发送, 因此此时重组的数据是错误的。

——注解 5

源端将停止在错误成员 (由于它们回告为 MST=FAIL 并随后设置失效成员为 DNU) 上发送数据, 并仅在其它 CTRL 为 NORM 和 EOS 的成员上发送数据。

对于故障为 MSU\_L 的情形, 从 CTRL=DNU 到达宿端直到宿端再收到 CTRL=NORM 这段时间内, VCG 的带宽是减少的; 而对于故障为 TSD 的情形, 从 CTRL=DNU 到达宿端以后, VCG 带宽就减少了。宿端 LCASC 并不知道什么时候数据完整性重新建立, 这些处理在数据层完成。

——注解 7

当失效成员故障修复后, CTRL 从 DNU 改变为 NORM 时, 宿端将重新使用该成员净荷来重组数据。

注: RS\_Ack 的初始值“0”可随意设置, 在示例中与 LCAS 机制相关的仅仅是 RS\_Ack 比特的反转。

附录 E  
(资料性附录)  
拖延期间无损更改带宽

E.1 介绍

在附录 C.4.4 中所描述的拖延状态图仅指出了输入信号为 MI-REMOVE 命令，对于其它输入信号并没有明确指定，这将导致这些输入信号丢弃。LCAS 状态机因此将在拖延期间并不处理 CTRL 控制字，这将影响所承载的业务流量。

E.2 源端删除组成员

在源端执行的删除操作并不要求源宿状态机之间的同步，因此，在发送更改的信号到宿端之前，源端的删除操作已经执行了。既然源端已经把该成员排除在承载净荷的容器之外，那么宿端应该能够对于这一更改立即响应，即使其正在拖延状态。如果无法做到这一点，那么将丢失客户净荷，并持续到拖延定时器过期为止。

E.3 执行ADD命令时TSD激活

当执行成员添加时，宿端收到 CTRL=ADD 以后，宿端状态将更改为 OK 状态同时发送 MST=OK 到源端。源端更改该成员的 CTRL 状态为 NORM，发送 CTRL=NORM/EOS 并开始在该成员容器中发送净荷。如果同时在宿端激活了 TSD 缺陷，宿端将启动拖延定时器并不能对于 CTRL=NORM/EOS 的控制字段做出响应。客户净荷将被丢失，并持续到拖延定时器过期为止。

E.4 增强的HO过程

虽然上述事件在特定时段内出现，而且出现的概率非常低，但为了允许在传输误码导致的特殊情形中也可以实现无损的容量调整，在此增强了附录 C.4.4 中阐述的 HO 过程的处理功能。具体增强的 HO 过程的状态图如 E.1 所示。

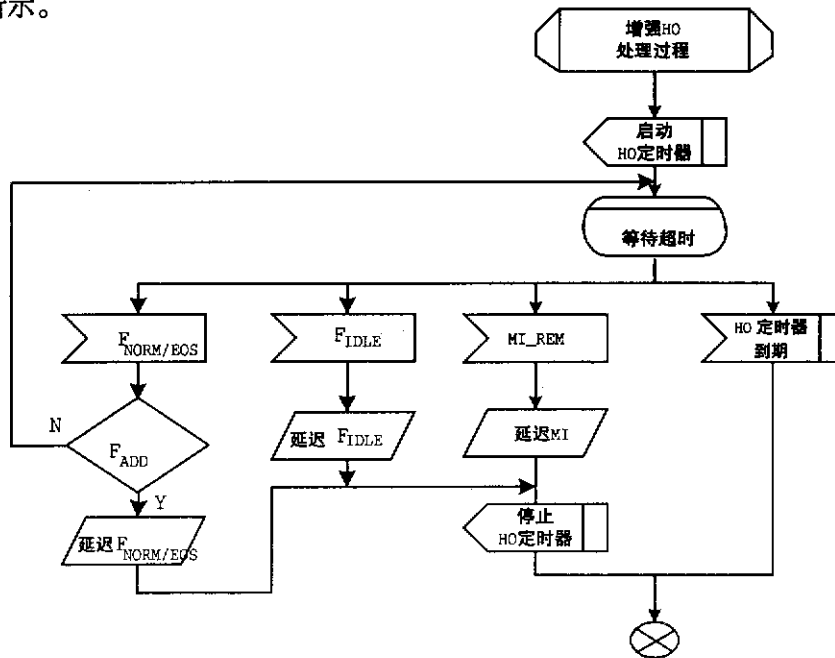


图 E.1 增强 HO 过程

附录 F  
(资料性附录)  
VCG 连续数据映射

如图 F.1 中所示, VCG 中连续数据的映射是基于字节并按照如下顺序来进行映射的:

- a) 第 X 个 VC;
- b) 第 C 列;
- c) 第 R 行。

也就是说, 第 1 个字节映射进第 1 个 VC-n #1 的第 1 列、第 1 行; 第 2 个字节映射进第 2 个 VC-n #2 的第 1 列、第 1 行; 第 3 个字节映射进第 3 个 VC-n #3 的第 1 列、第 1 行.....第 X 个字节映射进第 X 个 VC-n #X 的第 1 列、第 1 行。

第 X+1 个字节映射进第 1 个 VC-n #1 的第 2 列、第 1 行; 第 X+2 个字节映射进第 2 个 VC-n #2 的第 2 列、第 1 行; 第 X+3 个字节映射进第 3 个 VC-n #3 的第 2 列、第 1 行.....第 X+X 个字节映射进第 X 个 VC-n #X 的第 2 列、第 1 行。

依此类推, 直到  $X \times C \times R$  个连续字节映射完毕, 然后按照上述顺序重复进行映射。

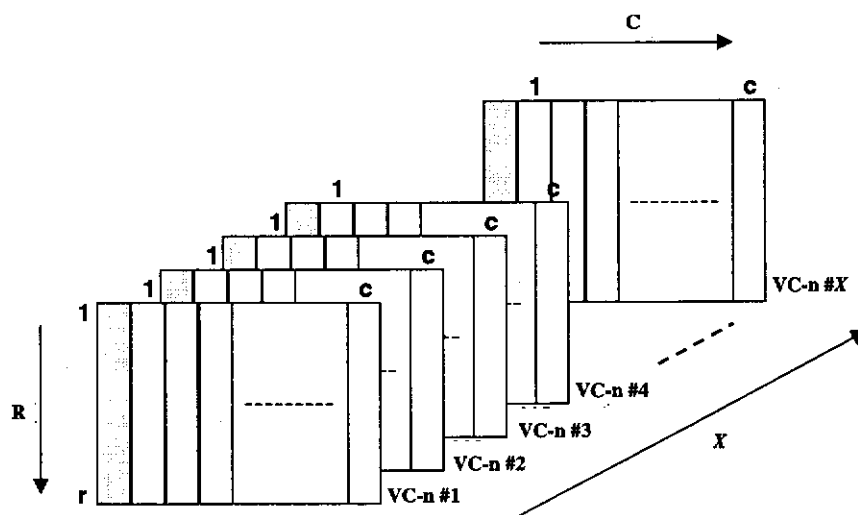


图 F.1 VCG 中连续数据映射顺序

附录 G  
(资料性附录)  
和映射结构有关的 POH 字节

## G.1 VC-3/VC-4的信号标记: C2字节

在 SDH 帧结构中分配了一个字节用来指示 VC-4-Xc/VC-4/VC-3 的组成或维持的状态。表 G1 列出了该字节对应的十六进制码及其主要含义。

表 G.1 C2 字节编码

高位 1 2 3 4 5 6 7 8	低位	十六进制 (注 1)	含 义
0 0 0 0 0 0 0 0		00	未装载或监控未装载信号(注 2)
0 0 0 0 0 0 0 1		01	预留(注 3)
0 0 0 0 0 0 1 0		02	TUG 结构
0 0 0 0 0 0 1 1		03	锁定支路单元方式 TU-n(注 4)
0 0 0 0 0 1 0 0		04	异步映射 34Mbit/s 或 45Mbit/s 到 C-3
0 0 0 0 0 1 0 1		05	实验性质的映射(注 9)
0 0 0 1 0 0 1 0		12	异步映射 140Mbit/s 到 C-4
0 0 0 1 0 0 1 1		13	ATM 映射
0 0 0 1 0 1 0 0		14	MAN DQDB [1]映射
0 0 0 1 0 1 0 1		15	FDDI[3]~[11]映射
0 0 0 1 0 1 1 0		16	HDLC/PPP 映射
0 0 0 1 0 1 1 1		17	为私有应用预留(注 10)
0 0 0 1 1 0 0 0		18	HDLC/LAPS 信号映射
0 0 0 1 1 0 0 1		19	为私有应用预留(注 10)
0 0 1 0 0 0 0 0		20	异步映射 ODUk (k=1, 2) 到 VC-4-Xv (X=17, 68)
0 0 0 1 1 0 1 0		1A	10 Gbit/s 以太网帧映射
0 0 0 1 1 0 1 1		1B	GFP 映射
0 0 0 1 1 1 0 0		1C	10 Gbit/s 光纤通道 (FC) 帧映射(注 8)
1 1 0 0 1 1 1 1		CF	预留(注 7)
1 1 0 1 0 0 0 0		D0	为私有应用预留(注 10)
... ..		...	
1 1 0 1 1 1 1 1		DF	
1 1 1 0 0 0 0 1		E1	为国内使用预留
... ..		...	
1 1 1 1 1 1 0 0		FC	
1 1 1 1 1 1 1 0		FE	测试信号, O.181 建议规范的映射(注 5)
1 1 1 1 1 1 1 1		FF	VC-AIS 缺陷指示(注 6)

注 1: 剩余 191 个备用码, 留作将来使用。

注 2: 值“0”表示“VC-3/VC-4/VC-4-XC”通道未装载或监控未装载。该值在一个开放连接的情况下以及一个监控的未装载信号没有包含任何负荷的情况下为“0”。

注 3: 值“1”仅适用表没有规定的映射码的情况, 但是 2000 年 10 月之后设计的设备在这种情况下应该使用值“5”。对于与老设备互通(即设计成仅传送“0”和“1”值)可采用下述条件:

——为了实现后向兼容, 老设备应该把除“0”值之外的任何值解释为通道已装载。

——为了实现前向兼容, 当收到来自老设备的“1”值, 新设备不应该产生一个负荷失配告警。

注 4: 为了实现后向兼容, 即使没有另外规定锁定模式字节同步映射, 也应仍按前述规定解释代码“03”。

注 5: 当 ITU-T O.181 建议规定的映射和 ITU-T G.707 (12/2003) 建议中规定的映射不相符时应归入此类。

注 6: 值“FF”表示 VC-AIS。如果没有有效的输入信号可用并且生成一个替代信号, 则由 TCM 源生成。

注 7: 为以前过时的 HDLC/PPP 帧信号映射分配的值。

注 8: 这些映射待研究, 是临时分配的信号标签。

注 9: 在本表没有定义的一种映射代码的情况下, 值“05”仅用于实验性质的活动。

注 10: 该代码值不符合将来的标准

## G.2 VC-12的信号标记: V5 (b5~b7)

V5 字节为 VC-12 通道提供误块检测、信号标记和通道状态功能。其中, V5 (b5~b7) 比特提供 VC-12 的信号标记, 如表 G.2 所示。

表 G.2 VC-12 V5 信号标记编码

b5	b6	b7	含义
0	0	0	未装载或监控的未装载
0	0	1	预留 (注 1)
0	1	0	异步映射
0	1	1	比特同步映射 (注 2)
1	0	0	字节同步映射
1	0	1	扩展信号标记, 和 K4 (b1) 配合使用 (注 1)
1	1	0	按 O.181 建议规定映射的测试信号 (注 3)
1	1	1	VC-AIS (注 4)

注 1: 在 2000 年 10 月之后设计的新设备不应该使用“001”。原来的老设备中“001”用于表示 VC-12 通道装载非特定净荷, 用于映射编码不在本表中的情况, 新设备使用为新设计定义的“101”和扩展信号标记 K4 (b1)。为了与只传送“000”和“001”的老设备互连, 需满足下面的规定:  
 ——为了实现后向兼容, 老设备应把接收到的任何非“000”的值解释为已装载。  
 ——为了实现前向兼容, 当从老设备收到“001”的值时, 新设备不应产生信号标记失配告警。

注 2: 在 VC-12 映射中, 为了后向兼容, 即使不再规定 2048 kbit/s 信号的比特同步映射, 也应按以前的规定解释为“011”。

注 3: 任何在 ITU-T 建议 O.181 中定义的任何非虚级联映射, 且无法对应到 ITU-T 建议 G.707 (12/2003) 中定义的映射, 可归入到此类。

注 4: “111”表示 VC-AIS。如果没有有效的输入信号可用并且生成一个替代信号, 则由 TCM 源生成

## G.3 VC-12的扩展信号标记: K4 (b1)

由于 V5 字节的 b5~b7 表示的信号标记状态不够用, 故将代码“101”定义为扩展的信号标记。它和 K4 的 b1 配合使用能表示更多的含义。如果 V5 (b5~b7) 比特的取值不为“101”, 扩展信号标记将是未定义的, 应被接收器忽略。K4 (b1) 比特只有一位, 也难以表示众多的客户信号类型。因此采用图 G.1 表示的 32 个 K4 (b1) 组成的复帧信号。

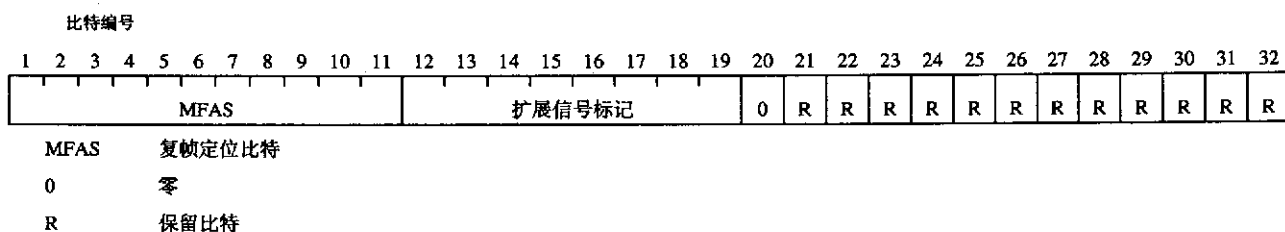


图 G.1 K4 (b1) 复帧

复帧的定位信号 (MFAS) 由“0111 1111 110”的图案组成, 位于复帧的第 1 到第 11 位。扩展信号标记用复帧的第 12 到第 19 位表示。复帧的第 20 位必须置为“0”, 剩余 12 位也应该设为“0”, 留待将来标准化, 接收器应忽略这些比特。

需要注意的是, 基于 K4 (b2) 的低阶通道虚级联复帧使用了 K4 (b1) 的复帧定位信号, 这意味着低阶通道虚级连功能需要考虑该比特位, 而且也不用确认 V5 (b5~b7) 是否为“101”。这样要求并不矛盾, 因为低阶虚级联净荷必须要求具有扩展信号标签。另外, 如果 K4 (b1) 复帧中留作将来保留的位使用的话, 应该避免出现连续 9 个 1 的情况 (会与 MFAS 冲突)。

VC-12 扩展标记编码见表 G.3。

表G.3 VC-12扩展标记编码

高位									低位		十六进制 (注1)	含 义
b12	b13	b14	b15	b16	b17	b18	b19					
0	0	0	0	0	0	0	0	0	0	00	保留(注2)	
0	0	...	0	0	0	1	...	1	1	07		
0	0	0	0	0	1	0	0	0	0	08	开发中的映射(注3)	
0	0	0	0	0	1	0	0	0	1	09	ATM映射	
0	0	0	0	0	1	0	1	0	0	0A	HDLC/PPP 帧信号映射	
0	0	0	0	0	1	0	1	1	1	0B	HDLC/LAPS 帧信号映射	
0	0	0	0	0	1	1	0	0	0	0C	按 O.181 建议规定映射的虚级连测试信号(注4)	
0	0	0	0	0	1	1	0	1	1	0D	GFP映射	
1	1	0	1	0	0	0	0	0	0	D0	为私有应用预留(注5)	
1	1	...	0	1	1	1	...	1	1	DF		
1	1	1	1	1	1	1	1	1	1	FF	保留	

注1: 有 225 个备用码留待将来使用。  
 注2: 值“00”到“07”保留以使表 8 中扩展和没有扩展的信号标记拥有一个惟一名称。  
 注3: 在本表没有定义的映射代码的情况下, 值“08”仅用于实验性质的活动。  
 注4: 任何在 ITU-T 建议 O.181 中定义的任何虚级连映射, 且无法对应到 ITU-T 建议 G.707 (12/2003) 中定义的映射, 归入到此类。  
 注5: 这些代码值不符合将来的标准