



中华人民共和国通信行业标准

YD/T 936—1997
eqv ITU-T T.125:1994

音像和视听会议业务的 多点通信服务协议

**Multipoint communication service protocol specification
for audiographics and audiovisual conferencing sevice**

1997-11-28 发布

1998-04-01 实施

中华人民共和国邮电部 发布

目 次

前言	III
1 范围	1
2 引用标准	1
3 定义	1
4 缩略语	2
5 MCS 协议综述	3
6 运输(层)服务的使用	8
7 MCSPDU 的结构	9
8 MCSPDU 的编码	22
9 MCSPDU 的路由	22
10 MCSPDU 的含义	25
11 MCS 提供者信息库	44
12 过程单元	47

前　　言

本标准规定在多点通信域分级结构上运行的协议。它规定在一系列传输连接上控制交换协议信息和过程的格式。该协议的目的是实现 ITU-T 建议 T.122 定义的多点通信服务。

本标准根据国内通信行业急需而制定。

本标准等效采用 ITU-T 建议 T.125(1994 年版)。本标准中删除了上述建议中的提示的附录(附录 A、B、C、D、E、F 和 G)，同时删除了第 13 章。

本标准由邮电部电信科学研究院提出并归口。

本标准起草单位：邮电部电信传输研究所。

本标准主要起草人：聂秀英 李守静

中华人民共和国通信行业标准

音像和视听会议业务的多点通信服务协议

Multipoint communication service protocol specification for audiographics and audiovisual conferencing service

YD/T 936—1997
eqv ITU-T T. 125;1994

1 范围

本标准规定由 MCS 提供者到对等 MCS 提供者传送数据和控制信息的单一协议的规程,以及用于传送数据和控制信息的 MCS 协议数据单元的结构和编码。本标准从以下几方面进行规定:在对等 MCS 提供者之间通过交换 MCS 协议数据单元的交互作用;在 MCS 提供者和 MCS 用户之间通过交换 MCS 原语的交互作用和在 MCS 提供者和运输服务提供者之间通过交换运输层服务原语的交互作用。

本标准适用于支持 MCS 和希望在开放系统环境下互连的系统间的多对等通信场合。

2 引用标准

下列建议和其他参考文献通过在本标准中引用构成本标准的条文。在出版时,所示版本均为有效。所有建议和其他参考文献都会被修订,鼓励本标准的所有使用者探讨应用下列建议和其他参考文献最新版本的可能性。

ITU-T 建议 T. 122(1993)	用于声像和视听会议业务的多点通信服务
ITU-T 建议 T. 123(1993)	用于声像和视听会议应用的协议栈
CCITT 建议 X. 200(1988)	CCITT 应用的开放系统互连参考模型
CCITT 建议 X. 214(1988)	CCITT 应用的开放系统互连运输服务定义
CCITT 建议 X. 208(1988)	抽象句法表记法 1(ASN. 1)的规范
CCITT 建议 X. 209(1988)	抽象句法表记 1(ASN. 1)的基本编码规则规范
ITU-T 建议草案 X. 691 ISO/IEC DIS 8825-2	信息技术 — 开放系统互连 — ASN. 1 编码规则规范 — 分组编码规则(PER)规范

3 定义

本标准基于 CCITT 建议 X. 200 中的概念,并利用其中定义的下列术语:

- a) 流量控制;
- b) 重装;
- c) 再组合;
- d) 分段;
- e) 排序;
- f) 分割;
- g) 传输句法;
- h) 传输连接;
- i) 传输连接端点标识符;
- j) 传输服务;
- k) 传输服务接入点;
- l) 传输服务接入点地址;

m) 运输服务数据单元。

本标准也基于 ITU-T 建议 T.122 中开发的概念，并利用其中定义的下列术语：

- a) 控制型 MCSAP；
- b) MCS 连入；
- c) MCS 信道；
- d) MCS 连接；
- e) MCS 域；
- f) MCS 域选择符；
- g) MCS 专用信道；
- h) MCS 专用信道管理者；
- i) MCS 提供者；
- j) MCS 服务接入点；
- k) MCS 用户；
- l) MCS 用户标识符；
- m) 顶级 MCS 提供者。

本标准采用下列定义。

- a) MCS 服务数据单元：在从发送者到各接收者的传送期间，其标识符保持不变的一定数量的 MCS 用户数据。特别是 MCS-SEND-DATA 请求或 MCS-UNIFORM-SEND-DATA 请求的内容。
- b) MCS 接口数据单元：在单一的交互作用中，穿过 MCSAP 在 MCS 用户和 MCS 提供者之间运输的信息单元。每一 MCS 接口数据单元包含接口控制信息，也可能包含全部或部分 MCS 服务数据单元。
- c) MCS 协议数据单元：在 MCS 协议中交换的信息单元，包括协调其联合操作的 MCS 提供者之间传输的控制信息和可能代表为其提供服务的 MCS 用户而传送的数据。
- d) MCS 数据传送优先级：从发送者到各接收者的传输期间该值保持不变。根据所支持的不同数据传送优先级级数的 MCS 域参数，两个或多个最低优先级可能收到相同的服务质量。共分 4 级：最高级、高级、中级和低级。
- e) 有效的 MCSPDU：其结构和编码符合本标准的 MCSPDU。
- f) 无效的 MCSPDU：非有效的 MCSPDU。
- g) 协议差错：以与本标准的规程不一致的方式使用 MCSPDU。
- h) 连接 MCSPDU：Connect-Initial、Connect-Response、Connect-Additional、Connect-Result 中的任何一个。
- i) 域 MCSPDU：非连接 MCSPDU 的任何 MCSPDU。
- j) 数据 MCSPDU、SDrq、SDin、USrq、USin 中的任何一个。
- k) 控制 MCSPDU：非数据 MCSPDU 的任何域 MCSPDU。
- l) 初始 TC：MCS 连接中第一个传输连接，用于交换最高优先级的控制 MCSPDU 和数据 MCSPDU。
- m) 附加 TC：属于 MCS 连接中后续的传输连接，用于交换较低优先级的数据 MCSPDU。
- n) MCS 提供者子树：在 MCS 域的范围内，它由 MCS 提供者本身和它的 MCS 连入加上分级结构上下属于它的所有 MCS 提供者及它们的 MCS 连入组成。
- o) MCS 提供者的高度：在 MCS 域的范围内，它是比所有分级结构上下属的 MCS 提供者的最大高度高 1 的那一个高度。没有下属的 MCS 提供者的高度为 1。

4 缩略语

在本标准中使用下列缩略语。

MCS 多点通信服务

MCSAP	MCS 服务接入点
MCSPDU	MCS 协议数据单元
TC	传输连接
TS	传输服务
TSAP	传输服务接入点
TSDU	运输服务数据单元

5 MCS 协议综述

5.1 MCS 分级模型

MCS 提供者利用 ITU-T 建议 T.122 中定义的 MCS 原语通过 MCSAP 与 MCS 用户通信。这些原语可以是使用 MCS 连接在对等 MCS 提供者之间交换 MCSPDU 的起因或结果,也可以是单一 MCS 提供者中所发生动作的起因或结果。MCSPDU 交换发生在管理同一 MCS 域的 MCS 提供者之间。

一个 MCS 提供者可以有多个对等 MCS 提供者,每一个对等 MCS 提供者直接地通过不同 MCS 连接或间接地通过一个对等 MCS 提供者到达该 MCS 提供者。根据 MCS 域中执行的数据传输优先级的等级数,MCS 连接由一个或多个传输连接组成。通过一对 TSAP 使用的运输层服务来影响协议交换。

图 1 示出了该 MCS 分级结构的模型。

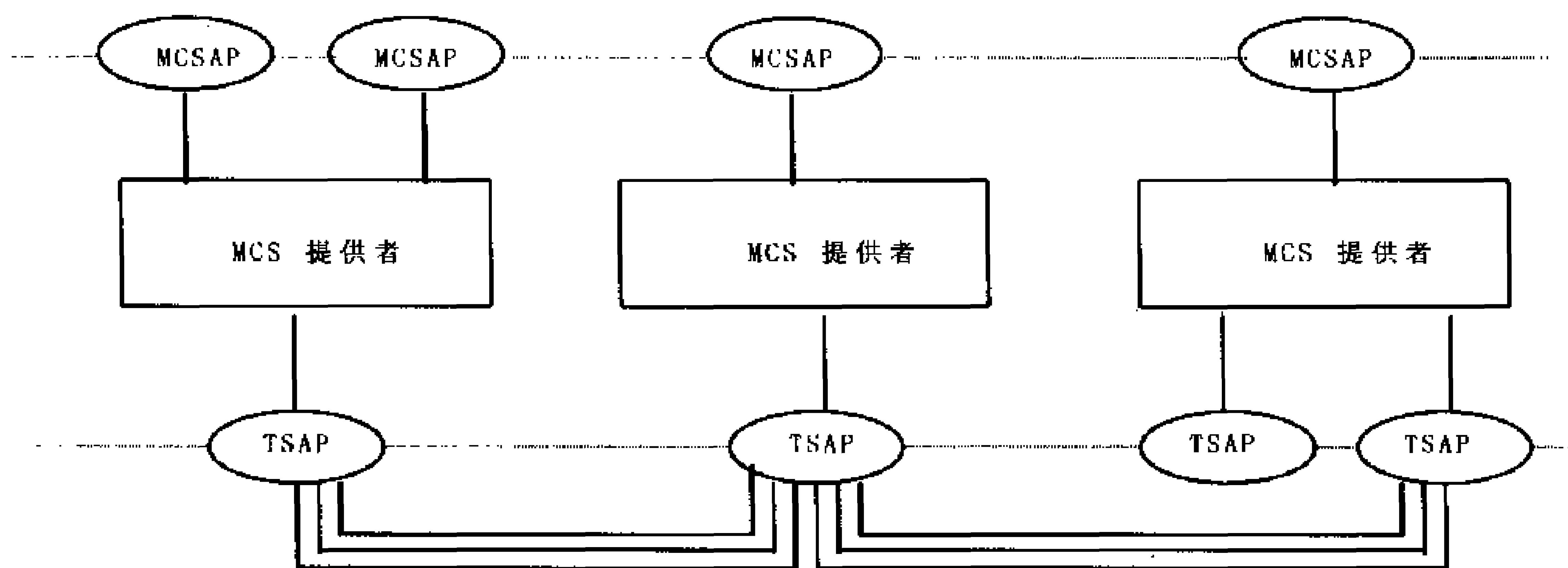


图 1 MCS 分级结构模型

5.2 MCS 分级结构提供的服务

MCS 协议支持 ITU-T 建议 T.122 定义的服务。使用表 1 列出的各原语将信息传送到 MCS 或从 MCS 传送出信息。

5.3 运输层执行的服务

MCS 协议采用 CCITT 建议 X.214 中定义的面向连接的运输层服务子集。利用表 2 中列出的各原语从 TS 提供者传送出信息或将信息传输到 TS 提供者。

5.4 MCS 分级结构的功能

表 1 表明与每一 MCS 原语相关联的 MCS 功能单元和 MCSPDU。在第 7 章中规定了 MCSPDU 原语和 MCSPDU 之间的关系可以简单到任一方向上的起因和结果。例如, MCS-ATTACH-USER 请求生成 AUrq, AUcf 生成 MCS-ATTACH-USER 证实。其他情况可能较复杂。例如, MCS-CONNECT-PROVIDER 的完成要求交换作为 4 个阶段原语的附加效应的附加 MCSPDU。同样 5 个 MCSPDU 中的任何一个都可能引起 MCS-DETACH-USER 指示,4 个 MCSPDU 中的任何一个可能引起 MCS-CHANNEL-EXPTEL 指示。

5.4.1 域管理

MCS 分级结构保持组成 MCS 域的 MCS 连接的整体性。MCS 连接是有方向性的,分级结构中的一端高于另一端。在每一个域的顶端有唯一一个 MCS 提供者。

建立一个 MCS 连接将两个域合并成一个域。MCS 分级结构确保保留一个顶级提供者。它解决唯一一致性或可能出现的外部拥有关系的冲突。

拆除 MCS 连接将一个域分为两部分,包含有顶级提供者的部分仍存在,另一部分自行消除。

MCS 分级结构唯一地标识连入域的用户。用户可以通过 MCS 原语之间的相互作用意识到彼此的存在。当域中的某个用户退出时,MCS 分级结构通知域中所有用户。MCS 分级结构复原退出用户的任何资源。

表 1 MCS 原语

功能单元	原语	相关 MCSPDU
域管理	MCS-CONNECT-PROVIDER 请求 MCS-CONNECT-PROVIDER 指示 MCS-CONNECT-PROVIDER 响应 MCS-CONNECT-PROVIDER 证实 (附加效应)	Connect-Initial Connect-Initial Connect-Response Connect-Response Connect-Additional Connect-Result PDir EDrq MCrq MCCf PCin MTrq MTcf PTin
	MCS-DISCONNECT-PROVIDER 请求 MCS-DISCONNECT-PROVIDER 指示	DPum DPum RJum
	MCS-ATTACH-USER 请求 MCS-ATTACH-USER 证实	AUrq AUcf
	MCS-DETACH-USER 请求 MCS-DETACH-USER 指示	DURq DUin MCcf PCin MTcf PTin
信道管理	MCS-CHANNEL-JOIN 请求 MCS-CHANNEL-JOIN 证实	CJrq CJcf
	MCS-CHANNEL-LEAVE 请求 MCS-CHANNEL-LEAVE 指示	CLrq MCcf PCin
	MCS-CHANNEL-CONVENE 请求 MCS-CHANNEL-CONVENE 证实	CCrq CCcf
	MCS-CHANNEL-DISBAND 请求 MCS-CHANNEL-DISBAND 指示	CDrq MCcf PCin
	MCS-CHANNEL-ADMIT 请求 MCS-CHANNEL-ADMIT 指示	CArq CAin
	MCS-CHANNEL-EXPTEL 请求 MCS-CHANNEL-EXPTEL 指示	CErq CEin CDin MCcf PCin

续表 1

功能单元	原语	相关 MCSPDU
数据传输	MCS-SEND-DATA 请求 MCS-SEND-DATA 指示	SDrq SDin
	MCS-UNIFORM-SEND-DATA 请求 MCS-UNIFORM-SEND-DATA 指示	USrq USin
令牌管理	MCS-TOKEN-GRAB 请求 MCS-TOKEN-GRAB 证实	TGrq TGcf
	MCS-TOKEN-INHIBIT 请求 MCS-TOKEN-INHIBIT 证实	TIrq TICf
	MCS-TOKEN-GIVE 请求 MCS-TOKEN-GIVE 指示 MCS-TOKEN-GIVE 响应 MCS-TOKEN-GIVE 证实	TVrq TVin TVrs TVcf
	MCS-TOKEN-PLEASE 请求 MCS-TOKEN-PLEASE 指示	TPrq TPin
	MCS-TOKEN-RELEASE 请求 MCS-TOKEN-RELEASE 证实	TRrq TRcf
	MCS-TOKEN-TEST 请求 MCS-TOKEN-TEST 指示	TTrq TTcf

5.4.2 信道管理

MCS 分级结构记录 MCS 域中哪些部分包含加入给定信道的一个或多个用户,这样就可以优化到拟接收数据的目的地的数据传送。

MCS 分级结构将用户标识符视作单一成员信道,仅允许所指明的用户加入该信道。在请求时,它可以生成专用信道,仅允许被允许的用户加入专用信道或将当前没有其他用户加入的信道作为公共信道进行分配。

5.4.3 数据传送

对于加入信道的用户 MCS 分级结构保持已排序的数据流,事实上,信道就变成了具有零个目的地到全广播的一系列地址的多分发表。

作为缺省情况,MCS 分级结构通过 MCS 连接最短路径将数据传送到每一接收者。作为选用,它通过顶级 MCS 提供者为指定的 MCS 服务数据单元选择路由,从而确保在所有接收者(包括发送者)处按一致排序接收数据。

MCS 分级结构识别数据传输的一个或多个优先级并给予优先处理。通过分段,它允许 MCS 具有无限长的 MCS 服务数据。

MCS 分级结构控制域中所有数据流。若一个接收者无能力以提供的速率接收数据最后将产生引起发送者阻塞的“反压”。如果用户不能保持最低的接收速度,用户可能被无条件地拆连。

只要源和目的地用户保持连入且目的地用户保持加入到信道,MCS 分级结构确保无差错地接收传送来的数据。然而,高优先级数据具有发送数据的优先权,同时过多的数据可能无限地延迟优先级较低的数据的传送。

表 2 传输服务原语

原语	使用	参数	使用
T-CONNECT 请求 T-CONNECT 指示	×	被叫地址 主叫地址 加速数据选项 服务质量 TS 用户数据	× × — × —
T-CONNECT 响应 T-CONNECT 证实	×	响应地址 加速数据选项 服务质量 TS 用户数据	— — × —
T-DATA 请求 T-DATA 指示	×	TS 用户数据	×
T-EXPEDITED-DATA 请求 T-EXPEDITED-DATA 指示	—	TS 用户数据	—
T-DISCONNECT 请求 T-DISCONNECT 指示	×	TS 用户数据	—
	×	原因 TS 用户数据	—

注：
 × MCS 协议假定该特征永远有效；
 — MCS 协议不使用该特征。

5.4.4 令牌管理

MCS 分级结构在顶级 MCS 提供者处执行令牌操作，因而确保一致性和排它性。

5.5 分级处理

MCS 域中的分层处理示于图 2。

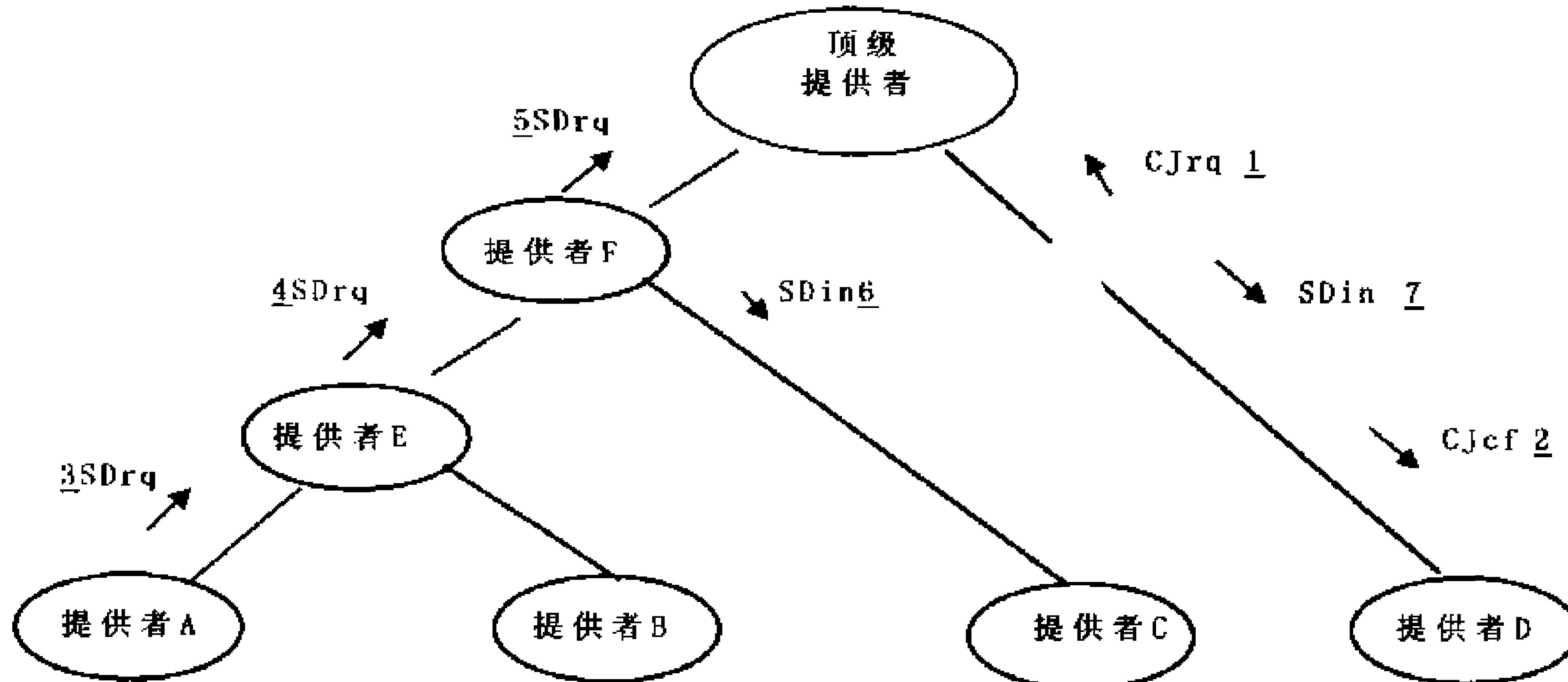


图 2 MCS 域中分层处理

图中的节点表示 MCS 提供者，标有标签的箭头表示 MCS PDU。此例集中于通过 MCS 提供者之间的

连接建立了域之后和正在开始扩展数据传送使用的一段时间。在第一步提供者 D 以用户名义请求加入到数据将要在其上传送的信道;第二步,该请求被成功地证实;第三步,连入到提供者 A 的用户发送数据,开始向上传送相应的 SDrq。假使仅有连入到提供者 A、C 和 D 的用户已加入到数据将在其上发送的信道,在第六步和第七步向下反射请求 MCSPDU SDin;第四步,提供者 E 得知无其他下属需要接收数据便简单地上传 SDrq;得知提供者 C 已经表示出对该信道的兴趣,提供者 F 在第五步向上前转 SDrq,同时,在第六步向下反射。

MCS 提供者并不特别地注意它们在分级结构中的高度,除非它担任保持域高度的整体限制的角色,并在广义上讲它们是顶级提供者或不是顶级提供者。顶级 MCS 提供者没有向上连接。所有其他的提供者仅有一个向上连接。

MCS 提供者记录有关 MCS 域内它的子树中使用的信道和令牌的信息。

MCS 提供者记录子树中用户加入的信道,对于每一个这样的信道加入的起点,也就是说从哪一个连入或从哪一个下属 MCS 连接加入。它记录子树中分配的用户标识符和它们的起始地点。它还记录专用信道,此专用信道或者有一个管理者或者有一个孩子树中承认的用户,并且还记录相关的用户标识符。

MCS 提供者记录子树中用户抓获或禁止的令牌,并记录相关的用户标识符。

MCS 提供者监视它的子树发起的请求,以证实初始用户标识符被合法地分配给起始连入或向下的 MCS 连接。这样便在顶级 MCS 提供者周围生成保护环,并限制了可能在别的相关域中恶意的参与者的破坏程度。

在一般的情况下(有一些例外),MCS 分级结构的操作可描述如下:

- a) 在 MCS 连入处调用的 MCS 原语请求在相应 MCS 提供者处生成 MCSPDU,并将其向上发送接近到顶级 MCS 提供者。在那里,MCSPDU 在保持有关 MCS 域的所有信息处起作用。
- b) 为将结果返回到请求连入,在顶级 MCS 提供者处可生成证实 MCSPDU。将其传送出去的 MCS 提供者按照对它们子树操作的影响更新它们的记录。通过在每一后续的向下跳变处考虑本地记录将证实选路到初始用户标识符。
- c) 可能生成指示 MCSPDU 来代替将采取的有关行动,以通知其他接入者。指示可以被复制并在连到受影响的用户的几个连接上向下发送。MCS 提供者也可以用操作的影响作为处理指示部分来更新它们的记录。

上面的描述是想要创建概念性的框架综述。有关在 MCS 提供者中所记录的信息和如何处理指定的 MCSPDU 的全部细节在后面的各章中规定。

其中的例外如下:一些请求,特别是 CJrq 和 CLrq,可能在离顶级 MCS 提供者不远的级上停止上升,一些指示,特别是 SDin,可能在低于该顶级 MCS 提供者级上生成。某个 MCSPDU(TVrs)属于响应范畴。而一些 MCSPDU 可以作为不同于 MCSPDU 的处理的继续,由 MCS 提供者生成,如紧随 DUin 之后的 CLrq。

5.6 域参数

管理单一 MCS 域的 MCS 提供者按照下列参数分配资源并执行程序。在整个域内这些参数值是完全相同的。

- a) 可以同时使用的 MCS 的最大信道数。它包括任一用户加入的信道、已经分配的用户标识符和已经创建的专用信道。
- b) 可同时被分配的最大的用户标识符数。它是在前一个参数限制内的子限制。
- c) 可同时被抓获或禁止的最大的令牌标识符数。
- d) 执行的数据传输优先级数。它等于一个 MCS 连接中 TC 的数。MCS 用户仍然可以用限制之外的优先级发送和接收数据。然而,这样的优先级可能被作为执行的最低优先级一样对待。
- e) 强制吞吐量。尽管整个流量控制将域中的数据传输限制到速度最低的接收者的速度,但绝对不能允许接收者以任意慢的速度运行。否则,会议中的一方可能会阻碍所有其他方。该参数命令 MCS 提供者

强制每一 MCS 连入和在每一个向下进行 MCS 连接处强制一个最低接收速率。违反者将有分别被无条件地踢除或拆断的危险。

- f) 最大高度。它限制所有 MCS 提供者,特别是顶级提供者的高度。
- g) 域 MCSPDU 的最大尺寸。整个流量控制基于在 MCS 提供者(非连接 MCSPDU)中的缓存域 MCSPDU。为简单起见,采用固定大小的缓存存储器。MCS 提供者不应生成较大的 MCSPDU。该限制可以被分组成单一控制 MCSPDU 的信息量,并建议不受限的用户数据应在数据 MCSPDU 中分段的地点。

- h) 协议版本。它采用定义不同域 MCSPDU 编码的两个值之一。

注:MCS 提供者在一给定场合可以用已参数化的本地资源限制进行操作。这些限制可能包括用于缓存等待传输的 MCSPDU 的存贮量、MCS 连入者的最大数和连接到其他提供者的 MCS 的最大数。这些参数是本地事宜而且不在整个 MCS 域内交换。

6 运输(层)服务的使用

6.1 运输(层)服务的模型

本段描述取自 CCITT 建议 X.214 的相应部分,假定不使用加速发送的数据组成的。

运输(层)服务向 TS 用户提供这些特征:

- a) 为交换 TSDU 而与另一个 TS 用户建立一条 TC 的方法。在同一对 TS 用户之间可能存在多于一条 TC。
- b) 在其建立时与每一 TC 相关联,通过该 TS 提供者请求、协商、认可,且由表示象吞吐量、传输时延、残余差错率和优先级这些特征参数所规定的某些服务质量的机会。
- c) 在 TC 上传送 TSDU 的方法。TSDU 的传输是透明的,在传送过程中 TS 提供者保持 TSDU 的边界、TSDU 的内容的一致性,TSDU 由整数个八位组构成。
- d) 接收 TS 用户可以用其控制发送 TS 用户可能用以发送数据的速率的方法。
- e) TC 的无条件因而可能为毁灭性的释放。

通过链接两个 TSAP 的一对队列,以抽象的方式模拟 TC 的操作。在信息流的每个方向上有一个队列。分别用一对队列模拟每一 TC。

该队列模型用来表示流量控制特征。队列具有有限的能力,但是,此能力既不必是固定的也不必是可确定的。作为在两个 TSAP 的交互作用的结果,将 TSDU 连接或拆断客体送入或移出队列。TS 用户将客体加入队列的能力是通过 TS 用户将客体从那个队列中移出的行为和该队列的状态确定。TS 提供者可以将其放到队列中的客体只能是拆断客体。TS 提供者将客体加到要控制的队列客体。客体通常由 TS 接收用户控制移出队列。客体一般按它们加入的顺序移出队列。一般移出的唯一例外是当且仅当后随者是拆断客体时,TS 提供者可以删除一个客体。

如果 TS 用户和 TS 提供者需要在 TSAP 处区分几个 TC,必须本地提供 TC 端点识别机制。然后,所有原语必须利用这一识别机制去识别它们所使用的 TC。该隐式识别不作为 TS 原语的参数示出,也不会与 T-CONNECT 的地址参数相混淆。

6.2 多连接的使用

MCS 连接由同一对 MCS 提供者之间的一个或多个 TC 组成。称所建立的第一条 TC 为初始 TC;称后续建立的那些 TC 为附加 TC。属于一 MCS 连接的所有 TC 由同一个 MCS 提供者建立,以响应 MCS-CONNECT-PROVIDER 请求。该请求包含主叫和被叫 TSAP 地址的地址参数。在生成的 T-CONNECT 请求中不修改地使用这些参数。

在整个 MCS 域内每一条 MCS 连接的 TC 数是不变的。该域参数等于执行的数据传输优先级的级数。因为每一条都是流量控制驱动器,因而需要不同的 TC。较低优先级数据块对较高优先级数据不应产生反压。为完成执行,必须在不同的 TC 上加载较低和较高优先级数据。

对于一个 TC 所请求的服务质量可以根据它们所建立的数据优先级进行改变。在整个 MCS 域上这些

服务质量目标不必一成不变。

服务质量中最大或平均吞吐量和传输时延较令人感兴趣。对实时响应,高优先级数据可获得低传输时延,但是可能不需要高吞吐量。另一方面,低优先级数据可获得高吞吐量但并不需要低传输时延。

尽管它不与 MCS 数据传输优先级的概念精确地对应,TC 优先级是服务质量的另一个方面。如果需要,它规定使它们的服务质量降低的 TC 的相对次序。为确保 MCS 优先级数据受到相应的优先对待,可能会与其他特征如低传输时延一起请求高 TC 优先级。

连接 MCSPDU 仅作为在 TC 的任一方向上运载的第一个 TSDU 出现。Connect-Initial 和 Connect-Response 穿过 MCS 连接的初始 TC。如果有附加的 TC,Connect-Additional 和 Connect-Result 穿过附加的 TC。

发出 T-CONNECT 请求的主叫 MCS 提供者通过它自身的活动来控制 TC 成为同一 MCS 连接的部分以及它们所代表的数据传输优先级。

接收 T-CONNECT 指示的被叫 MCS 提供者通常必须接受 TC 并在了解它的特征之前读出它的第一个 TSDU。Connect-Initial 将入 TC 识别为一个新 MCS 连接的开始。Connect-Additional 将入 TC 识别为进行中的 MCS 连接的部分。

Connect-Additional 包含由被叫 MCS 提供者分配的、且在初始 TC 上的 Connect-Response 中传送到主叫 MCS 提供者的值,该值将附加 TC 指定为属于同一 MCS 连接。该初始 TC 将附加的 TC 指定为属于相同的 MCS 连接。Connect-Additional 也明确地表明 TC 所代表的数据优先级。

在 TC 建立之后立即交换连接 MCSPDU。一旦 MCS 连接被证实,它变为 MCS 域分级结构中的一部分。然后,MCS 连接传输域 MCSPDU。

除了数据 MCSPDU 之外,域 MCSPDU 穿过 MCS 连接的初始 TC。数据 MCSPDU 穿过与它们的数据优先级相对应的 TC。如果规定的优先级超出了 MCS 域执行的数值,则数据 MCSPDU 穿过所执行的最低优先级的 TC。

如果 MCS 域仅执行一个优先级,它的每一个 MCS 连接由单一的 TC 组成。不使用由 Connect-Additional 或 Connect-Result 组成的情况,并且在提供者之间按顺序传送所有 MCSPDU。

6.3 传输连接释放

TS 提供者通过执行足够多的协议来克服下级网络服务中任何缺欠,以增加端到端连接的可靠性。MCS 提供者不复制上述功能。在传输失败的情况下,MCS 提供者不试图自动恢复。

通过 T-DISCONNECT 指示宣布不可恢复的差错。如果属于 MCS 连接的任何 TC 被拆断,其他的也立即被拆断。除非用户请求,才生成 MCS-DISCONNECT-PROVIDER 指示并给出原因“提供者初始”。

另一方面,MCS-DISCONNECT-PROVIDER 请求应当在另一端作为具有原因“用户请求”的指示出现。尽管 X.214 中保证最简单级的传输协议不允许在 T-DISCONNECT 上传输用户数据。因此,在 MCSPDU 中传送拆断原因码。在接收端,该 MCSPDU 强迫 MCS 提供者拆断传输它的 MCS 连接。

7 MCSPDU 的结构

下面是用 CCITT 建议 X.208 的 ASN.1 表记法来规定 MCSPDU 的结构。这些 MCSPDU 的使用和特征在第 9 章和 10 章中进一步描述。

MCS-PROTOCOL DEFINITIONS ::=

BEGIN

-- 部分 1: 基本 MCS 类型

ChannelId ::= INTEGER (0..65535)

-- 范围是 16 比特

StaticChannelId ::= *ChannelId* (1..1000)

-- 永久已知的那些

```

DynamicChannelId ::= ChannelId (1001..65535)      -- 创建和删除的那些

UserId ::= DynamicChannelId                         -- 由 Attach-User 创建的
                                                    -- 由 Detach-User 删除的

PrivateChannelId ::= DynamicChannelId              -- 由 Channel-Convene 创建的
                                                    -- 由 Channel-Disband 删除的

AssignedChannelId ::= DynamicChannelId            -- 由 Channel-Join 0 创建的
                                                    -- 由 Detach-User 删除的
                                                    -- 全部为永久已知的

TokenId ::= INTEGER (1..65535)

TokenStatus ::= ENUMERATED
{
    notInUse          (0),
    selfGrabbed       (1),
    otherGrabbed      (2),
    selfInhibited     (3),
    otherInhibited    (4),
    selfRecipient     (5),
    selfGiving         (6),
    otherGiving        (7)
}

DataPriority ::= ENUMERATED
{
    top               (0),
    high              (1),
    medium             (2),
    low                (3)
}

Segmentation ::= BIT STRING
{
    begin             (0),
    end               (1)
} (SIZE (2))

DomainParameters ::= SEQUENCE
{
    maxChannelIds INTEGER(0..MAX),                  -- 对使用的信道标识符的限制
                                                    -- 静态的+用户标识符+专用的+分配的

    maxUserIds INTEGER (0..MAX),                     -- 单独对用户标识符的子限制

    maxTokenIds INTEGER (0..MAX),                   -- 对使用的令牌标识符的限制
                                                    -- 抓获的+被禁止的+正在给出的+不可给出的
}

```

+ 已给出的

```

numPriorities INTEGER (0..MAX),
    -- MCS 连接中 TC 数
minThroughput INTEGER (0..MAX),
    -- 每秒钟强迫的八位组数
maxHeight INTEGER (0..MAX),
    -- 对提供者高度的限制
maxMCSPDUsizE INTEGER (0..MAX),
    -- 对域 MCSPDU 的八位组限制
protocolVersion INTEGER (0..MAX)
}

-- 部分 2: 连接提供者

Connect-Initial ::= [APPLICATION 101] IMPLICIT SEQUENCE
{
    callingDomainSelector OCTET STRING,
    calledDomainSelector OCTET STRING,
    upwardFlag BOOLEAN,
        -- 如果被叫提供者较高, 则为 TRUE
    targetParameters DomainParameters,
    minimumParameters DomainParameters,
    maximumParameters DomainParameters,
    userData OCTET STRING
}

Connect-Response ::= [APPLICATION 102] IMPLICIT SEQUENCE
{
    result Result,
    calledConnectId INTEGER (0..MAX),
        -- 为识别同一 MCS 连接
        -- 的附加 TC 由被叫
        -- 提供者分配的
    domainParameters DomainParameters,
    userData OCTET STRING
}

Connect-Additional ::= [APPLICATION 103] IMPLICIT SEQUENCE
{
    calledConnectId INTEGER (0..MAX),
    dataPriority DataPriority
}

Connect-Result ::= [APPLICATION 104] IMPLICIT SEQUENCE
{
    result Result
}

```

— 部分 3: 域合并

```

PDis ::= [APPLICATION 0] IMPLICIT SEQUENCE -- 最终域指示
{
    heightLimit INTEGER (0..MAX)
                                -- 对 MCSPDU 接收者的限制
}
EDrq ::= [APPLICATION 1] IMPLICIT SEQUENCE -- 建立域请求
{
    subHeight INTEGER (0..MAX),
                                -- MCSPDU 发送者在域中的高度
    subInterval INTEGER (0..MAX)
                                -- 用毫秒表示的它吞吐量强迫间隔
}
ChannelAttributes ::= CHOICE
{
    static [0] IMPLICIT SEQUENCE
    {
        channelId      StaticChannelId
                        -- 加入的为隐含的 TRUE
    },
    userId [1] IMPLICIT SEQUENCE
    {
        joined         BOOLEAN,
        userId         UserId
                        -- 如果用户加入到它的用户标识符, 则为 TRUE
    },
    private [2] IMPLICIT SEQUENCE
    {
        joined         BOOLEAN,
                        -- 如果下面加入了信道标识符, 则为 TRUE
        channelId      PrivateChannelId,
        manager        UserId,
        admitted      SET OF UserId
                        -- 跨越多个 MCrq
    },
    assigned [3] IMPLICIT SEQUENCE
    {
        channelId      AssignedChannelId
                        -- 加入的为隐含的 TRUE
    }
}
MCrq ::= [APPLICATION 2] IMPLICIT SEQUENCE -- 合并信道请求

```

```

{
    mergeChannels          SET OF ChannelAttributes,
    purgeChannelIds       SET OF ChannelId
}

MCcf ::= [APPLICATION 3] IMPLICIT SEQUENCE -- 合并信道证实
{
    mergeChannels          SET OF ChannelAttributes,
    purgeChannelIds       SET OF ChannelId
}

PCin ::= [APPLICATION 4] IMPLICIT SEQUENCE -- 驱除信道指示
{
    detachUserIds          SET OF UserId,
    -- 驱除用户标识符信道
    purgeChannelIds        SET OF ChannelId
    -- 驱除其他信道
}

TokenAttributes ::= CHOICE
{
    grabbed                [0] IMPLICIT SEQUENCE
    {
        tokenId             TokenId,
        grabber              UserId
    },
    inhibited              [1] IMPLICIT SEQUENCE
    {
        tokenId             TokenId,
        inhibitors           SET OF UserId          -- 可能跨越多个 MCreq
    },
    giving                 [2] IMPLICIT SEQUENCE
    {
        tokenId             TokenId,
        grabber              UserId,
        recipient             UserId
    },
    ungivable              [3] IMPLICIT SEQUENCE
    {
        tokenId             TokenId,
        grabber              UserId
        -- 接受者已经拆连
    },
    given                  [4] IMPLICIT SEQUENCE
    {
        tokenId             TokenId,
    }
}

```

```

recipient          UserId
                  -- 抓获者释放或被拆断
}
}

MTrq ::= [APPLICATION 5] IMPLICIT SEQUENCE -- 合并令牌请求
{
  mergeTokens      SET OF TokenAttributes,
  purgeTokenIds   SET OF TokenId
}

MTcf ::= [APPLICATION 6] IMPLICIT SEQUENCE -- 合并令牌指示
{
  mergeTokens      SET OF TokenAttributes,
  purgeTokenIds   SET OF TokenId
}

PTin ::= [APPLICATION 7] IMPLICIT SEQUENCE -- 驱除令牌指示
{
  purgeTokenIds   SET OF TokenId
}

-- 部分 4:拆连提供者

DPum ::= [APPLICATION 8] IMPLICIT SEQUENCE -- 拆连提供者最后的结论
{
  reason           Reason
}

RJum ::= [APPLICATION 9] IMPLICIT SEQUENCE -- 拒绝 MCS-PDU 最后的结论
{
  diagnostic       Diagnostic,
  initialOctets   OCTET STRING
}

-- 部分 5:连入/拆连用户

AUrq ::= [APPLICATION 10] IMPLICIT SEQUENCE -- 连入用户请求
{
}

AUcf ::= [APPLICATION 11] IMPLICIT SEQUENCE -- 连入用户证实
{
  result           Result,
  initiator        UserId OPTIONAL
}

DUrq ::= [APPLICATION 12] IMPLICIT SEQUENCE -- 拆连用户请求
{
  reason           Reason,
  userIds          SET OF UserId
}

```

```

}

DUin ::= [APPLICATION 13] IMPLICIT SEQUENCE -- 拆连用户指示
{
    reason           Reason,
    userIds         SET OF UserId
}

-- 部分 6:信道管理

CJrq ::= [APPLICATION 14] IMPLICIT SEQUENCE -- 信道加入请求
{
    initiator       UserId,
    channelId        ChannelId
    -- 可能为零
}

CJcf ::= [APPLICATION 15] IMPLICIT SEQUENCE -- 信道加入证实
{
    result          Result,
    initiator       UserId,
    requested       ChannelId,
    -- 可能为零
    channelId        ChannelId OPTIONAL
}

CLrq ::= [APPLICATION 16] IMPLICIT SEQUENCE -- 信道离开请求
{
    channelIdss      SET OF ChannelId
}

CCrq ::= [APPLICATION 17] IMPLICIT SEQUENCE -- 信道召集请求
{
    initiator       UserId
}

CCcf ::= [APPLICATION 18] IMPLICIT SEQUENCE -- 信道离开证实
{
    result          Result,
    initiator       UserId,
    channelId        PrivateChannelId OPTIONAL
}

CDrq ::= [APPLICATION 19] IMPLICIT SEQUENCE -- 信道拆散请求
{
    initiator       UserId,
    channelId        PrivateChannelId
}

CDin ::= [APPLICATION 20] IMPLICIT SEQUENCE -- 信道拆散指示
{
    channelId        PrivateChannelId
}

```

}

CArq ::= [APPLICATION 21] IMPLICIT SEQUENCE -- 信道允许请求

{

initiator	UserId,
channelId	PrivateChannelId,
userIds	SET OF UserId

}

CAin ::= [APPLICATION 22] IMPLICIT SEQUENCE -- 信道允许指示

{

initiator	UserId,
channelId	PrivateChannelId,
userIds	SET OF UserId

}

CErq ::= [APPLICATION 23] IMPLICIT SEQUENCE -- 信道驱除请求

{

initiator	UserId,
channelId	PrivateChannelId,
userIds	SET OF UserId

}

CEin ::= [APPLICATION 24] IMPLICIT SEQUENCE -- 信道驱除指示

{

channelId	PrivateChannelId,
userIds	SET OF UserId

}

-- 部分 7: 数据传送

SDrq ::= [APPLICATION 25] IMPLICIT SEQUENCE -- 发送数据请求

{

initiator	UserId,
channelId	ChannelId,
dataPriority	DataPriority,
segmentation	Segmentation,
userData	OCTET STRING

}

SDin ::= [APPLICATION 26] IMPLICIT SEQUENCE -- 发送数据指示

{

initiator	UserId,
channelId	ChannelId,

```

dataPriority          DataPriority,
segmentation         Segmentation,
userData             OCTET STRING
}

USrq ::= [APPLICATION 27] IMPLICIT SEQUENCE -- 统一发送数据请求
{
  initiator           UserId,
  channelId           ChannelId,
  dataPriority        DataPriority,
  segmentation        Segmentation,
  userData            OCTET STRING
}

USin ::= [APPLICATION 28] IMPLICIT SEQUENCE -- 统一发送数据指示
{
  initiator           UserId,
  channelId           ChannelId,
  dataPriority        DataPriority,
  segmentation        Segmentation,
  userData            OCTET STRING
}

-- 部分 8:令牌管理

TGrq ::= [APPLICATION 29] IMPLICIT SEQUENCE -- 令牌抓获请求
{
  initiator           UserId,
  tokenId              TokenId
}

TGcf ::= [APPLICATION 30] IMPLICIT SEQUENCE -- 令牌抓获证实
{
  result               Result,
  initiator           UserId,
  tokenId              TokenId,
  tokenStatus          TokenStatus
}

Tlreq ::= [APPLICATION 31] IMPLICIT SEQUENCE -- 令牌禁止请求
{
  initiator           UserId,
  tokenId              TokenId
}

Tlcf ::= [APPLICATION 32] IMPLICIT SEQUENCE -- 令牌禁止证实
{

```

```

result          Result,
initiator      UserId,
tokenId         TokenId,
tokenStatus    TokenStatus
}

TVrq ::= [APPLICATION 33] IMPLICIT SEQUENCE -- 令牌给出请求
{
  initiator      UserId,
  tokenId         TokenId,
  recipient      UserId
}

TVin ::= [APPLICATION 34] IMPLICIT SEQUENCE -- 令牌给出指示
{
  initiator      UserId,
  tokenId         TokenId,
  recipient      UserId
}

TVrs ::= [APPLICATION 35] IMPLICIT SEQUENCE -- 令牌给出响应
{
  result          Result,
  recipient      UserId,
  tokenId         TokenId
}

TVcf ::= [APPLICATION 36] IMPLICIT SEQUENCE -- 令牌给出证实
{
  result          Result,
  initiator      UserId,
  tokenId         TokenId,
  tokenStatus    TokenStatus
}

TPrq ::= [APPLICATION 37] IMPLICIT SEQUENCE -- 令牌申请请求
{
  initiator      UserId,
  tokenId         TokenId
}

TPin ::= [APPLICATION 38] IMPLICIT SEQUENCE -- 令牌申请指示
{
  initiator      UserId,
  tokenId         TokenId
}

```

TRrq ::= [APPLICATION 39] IMPLICIT SEQUENCE -- 令牌释放请求

```
{
    initiator          UserId,
    tokenId           TokenId
}
```

TRcf ::= [APPLICATION 40] IMPLICIT SEQUENCE -- 令牌释放证实

```
{
    result            Result,
    initiator        UserId,
    tokenId          TokenId,
    tokenStatus      TokenStatus
}
```

TTrq ::= [APPLICATION 41] IMPLICIT SEQUENCE -- 令牌检测请求

```
{
    initiator        UserId,
    tokenId          TokenId
}
```

TTcf ::= [APPLICATION 42] IMPLICIT SEQUENCE -- 令牌检测证实

```
{
    initiator        UserId,
    tokenId          TokenId,
    tokenStatus      TokenStatus
}
```

-- 部分 9: 状态码

Reason ::= ENUMERATED -- 在 DPum、DUrq、DUin 中

```
{
    rn-domain-disconnected      (0),
    rn-provider-initiated     (1),
    rn-token-purged           (2),
    rn-user-requested         (3),
    rn-channel-purged         (4)
}
```

Result ::= ENUMERATED -- 在连接、响应、证实中

```
{
    rt-successful              (0),
    rt-domain-merging          (1),
    rt-domain-not-hierarchical (2),
    rt-no-such-channel        (3),
    rt-no-such-domain         (4),
    rt-no-such-user            (5),
}
```

```

    rt-not-admitted          (6),
    rt-other-user-id         (7),
    rt-parameters-unacceptable (8),
    rt-token-not-available   (9),
    rt-token-not-possessed    (10),
    rt-too-many-channels     (11),
    rt-too-many-tokens       (12),
    rt-too-many-users        (13),
    rt-unspecified-failure   (14),
    rt-user-rejected         (15)
}

```

Diagnostic ::= ENUMERATED -- 在 RJum 中

```

{
    dc-inconsistent-merge      (0),
    dc-forbidden-PDU-downward  (1),
    dc-forbidden-PDU-upward    (2),
    dc-invalid-BER-encoding    (3),
    dc-invalid-PER-encoding    (4),
    dc-misrouted-user          (5),
    dc-unrequested-confirm     (6),
    dc-wrong-transport-priority (7),
    dc-channel-id-conflict     (8),
    dc-token-id-conflict       (9),
    dc-not-user-id-channel     (10),
    dc-too-many-channels       (11),
    dc-too-many-tokens         (12),
    dc-too-many-users          (13)
}

```

-- 部分 10, MCSPDU 清单

```

ConnectMCSPDU ::= CHOICE
{
    connect-initial           Connect-Initial,
    connect-response          Connect-Response,
    connect-additional         Connect-Additional,
    connect-result            Connect-Result
}

```

DomainMCSPDU ::= CHOICE

```

{
    pdin          PDIn,
    edrq          EDrq,
    mcrq          MCrq,
}

```

mcdf	MCdf,
pcin	PCin,
mtrq	MTrq,
mtcf	MTcf,
ptin	PTin,
dpum	DPum,
rjum	RJum,
aurq	AUrq,
aucf	AUcf,
durq	DURq,
duin	DUin,
cjrq	CJrq,
cjcf	CJcf,
clrq	CLrq,
ccrq	CCrq,
ccdf	CCdf,
cdrq	CDrq,
cdin	CDin,
carq	CArq,
cain	CAin,
cerq	CErq,
cein	CEin,
sdrq	SDrq,
sdin	SDin,
usrq	USRq,
usin	USin,
tgrq	TGrq,
tgcf	TGcf,
tirq	TIRQ,
ticf	TICf,
tvrq	TVrq,
tvin	TVin,
tvrs	TVrs,
tvcf	TVcf,
tprq	TPrq,
tpin	TPin,
trrq	TRrq,
trcf	TRcf,
ttrq	TTrq,
ttcf	TTcf

}

END

8 MCSPDU 的编码

穿过属于 MCS 连接的 TC, 作为 TSDU 传输每一个 MCSPDU。对连接 MCSPDU 在大小上没有限制。用 MCS 域参数限制域 MCSPDU 的大小。

使用编码标准 ASN.1 数据值在对等 MCS 提供者之间传送 MCSPDU。本协议有两个版本, 这两个版本仅在编码规则规范上不同:

——版本 1; 对所有的 MCSPDU 使用 CCITT 建议 X.209 的基本编码规则。

——版本 2; 对连接 MCSPDU 使用基本的编码规则, 同时对所有后续域 MCSPDU 使用 ISO/IEC 8825 的分组编码规则。特别地, BASIC-PER 的 ALIGNED 变量应用于 ASN.1 类型 DomainMCSPDU。应将生成的比特串转换为整数个八位组。该串的开始位应与第一个八位组的最有效位相对应。

协商协议版本包括在初始 TC 上 Connect-Initial 和 Connect-Response MCSPDU 的交换, 这两个 MCSPDU 总是用基本编码规则进行编码。在域 MCSPDU 开始之前, 后随的任何 Connect-Additional 和 Connect-Result MCSPDU 也一样。域 MCSPDU 从 TC 上传输的第二个 TSDU 开始。

直到分组编码规则已经被采纳为 ITU-T 建议或 ISO/IEC 国际标准的一部分时才使用本协议的版本 2。

注:

1 分组编码规则生成更紧凑的 MCSPDU 头。

2 在 BER(基本编码规则)和 PER(分组编码规则)包含有分配每一编码的 MCSPDU 尾部的足够信息的情况下, BER 和 PER 均为自定界的。可能会引起有关不必要利用 TSDU 以及本协议能在非标准运输(层)服务上执行的争论, 该服务在不保留 TSDU 边缘的情况下运送八位组信息流。然而, 这种方法更易受执行差错的影响。如果丢失了 MCSPDU 之间的边缘, 恢复起来将很困难。

9 MCSPDU 的路由

9.1 连接 MCSPDU

图 3 规定了连接 MCSPDU 的交换。

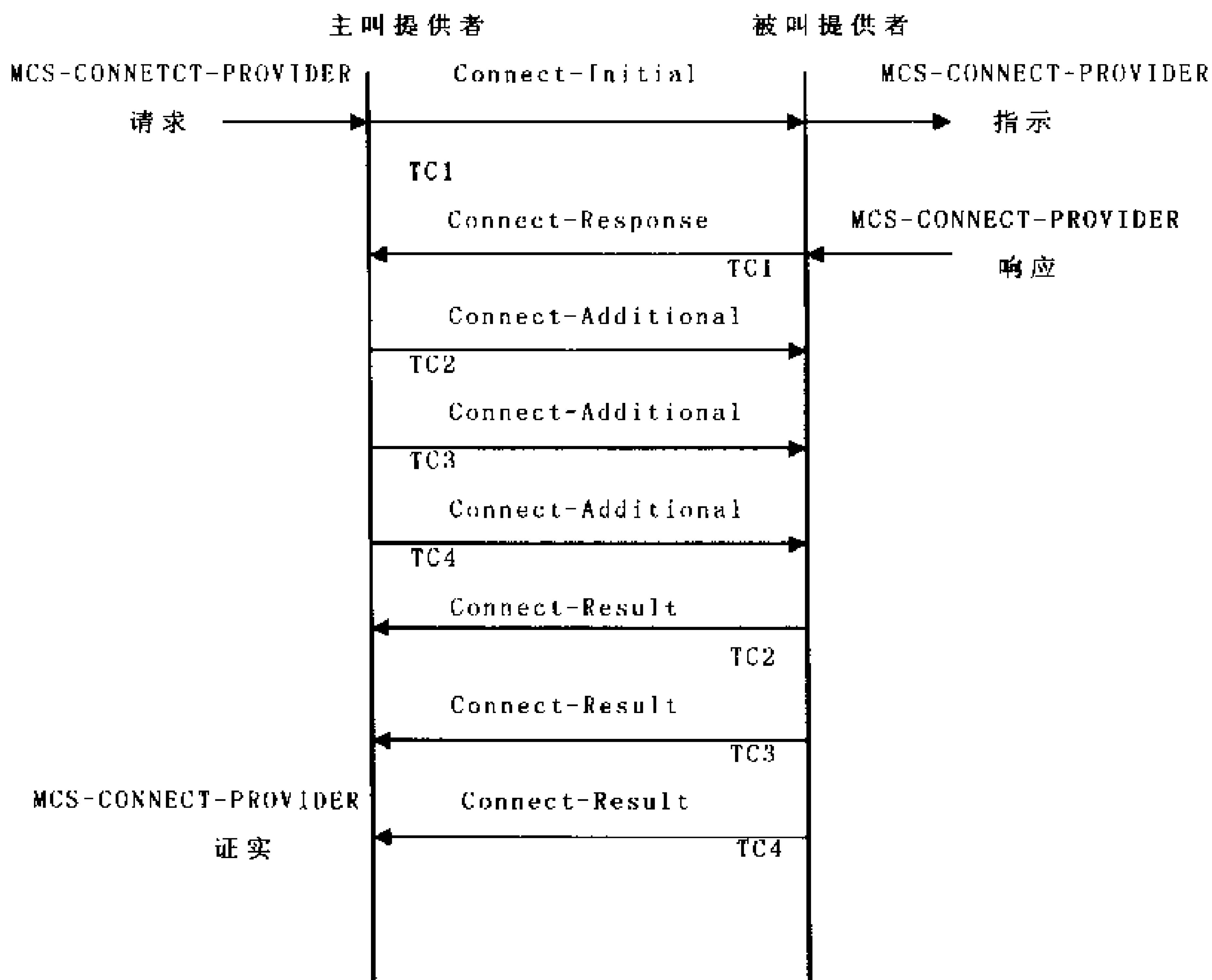
接收到 Connect-Response 之后, 主叫 MCS 提供者查看域中执行的数据传送优先级数的协商值。为图示的方便, 在被叫 MCS 提供者处, 附加 TC 上的连接 MCSPDU 严格按顺序 2、3、4 表示出来。实际上, 可以不按请求它们的顺序建立传输连接。确实, Connect-Additional 可以按不同于发送它的次序到达, 使得 Connect-Result 也不按该顺序返回。或对于后者那种情况, 即使按顺序发送, 也可以在传输过程中重新排序。主叫 MCS 提供者在发送第一个 Connect-Additional 之前, 不必等所有的附加 TC 均被建立。被叫 MCS 提供者在返回第一个 Connect-Result 之前不必等所有的 Connect-Additional MCSPDU 全部到达。无论按何种顺序收到所有的成功结果后, 主叫 MCS 提供者将产生成功的 MCS-CONNECT-PROVIDER 证实。

在一些中间点处的不成功的 Connect-Response 或 Connect-Result 或 T-DISCONNECT 指示将引起目前属于该 MCS 连接的所有 TC 拆断, 并生成不成功的 MCS-CONNECT-PROVIDER 证实。

MCS-CONNECT-PROVIDER 请求规定两个 MCS 提供者中的哪一个高。该分层关系确定了到域 MCSPDU 的后续路由, 并且与主叫和被叫 MCS 提供者之间的距离是不相关的。例如, 对该 MCS 分级结构, 下一步将合并两个先前独立的域资源。较低的 MCS 提供者生成 MCrq 和 MTrq, 并将 MCrq 和 MTrq 穿过新 MCS 连接传输到较高的 MCS 提供者。根据设置向上标志的不同方式, 这些 MCSPDU 发送的方向可能是主叫到被叫或被叫到主叫。

9.2 域 MCSPDU

表 3 规定了域 MCSPDU 的路由。



注：附加 TC 的数量和次序可以改变。

图 3 连接 MCS-PDU 的消息流

表 3 域 MCS-PDU 的路由

类别	MCS-PDU	TC	方向
请求	EDrq MCrq MTrq AUrq DUrq CJrq CLrq CCrq CDrq CArq CErq TGrq TIrq TVrq TPrq TRrq TTrq	I	向上
	SDrq USrq	A	
指示	PDin PCin PTin DUin CDin CAin CEin TVin TPin	I	向下
	SDin USin	A	
响应	TVrs	I	向上

续表 3

类别	MCSPDU	TC	方向
证实	MCef MTcf AUcf CJcf CCef TGcf TLcf TVcf TRef TTcf	I	向下
最终结果	DPum RJum	I	向上或向下

注：

I: 该 MCSPDU 通过初始 TC。

A: 根据它的数据传送优先级, MCSPDU 可能通过附加 TC。

向上: 向顶 MCS 提供者传送 MCSPDU。

向下: 远离顶 MCS 提供者传送 MCSPDU。

如果 MCS 提供者生成或前转请求类 MCSPDU, 它在唯一的向上 MCS 连接上传送。在中间的某些 MCS 提供者处可能消耗掉 EDrq、CJrq 和 CLrq。其他请求无需证实就上升到顶级 MCS 提供者, 并由顶级 MCS 提供者使其起作用, 除非确定了请求的内容无效, 在此情况下, 可以忽略它的 MCSPDU。

当 MCS 提供者生成或前转指示类 MCSPDU(即在内容上可能有所变化的拷贝), 就按下列规则在零个或多个 MCS 连接上向下传送:

a) 在所有的 MCS 连接上向下发送 PDin。将它包含的高度限值减 1。接收到具有高度限值为零的该 MCSPDU 的 MCS 连接时将拆断。

b) 在所有的 MCS 连接上向下传输 PCin。不改变前转的用户标识符集, 以便所有的被拆连的用户均被通知到保留下来的那些用户。前转的其他信道标识符集可能被限制在子树中的那些用户之中。在正处在合并到较高域的前顶级 MCS 提供者处, 用户标识符和其他信道标识符均被限制为已经被证实了的接受较高域的那些标识符。

c) 在所有的 MCS 连接上向下发送 PTin。前转的令牌标识符集可能被限制为子树中使用的那些标识符。在目前正处在合并到较高域的前顶级 MCS 提供者处, 令牌标识符均被限制为已经被证实了的接受到较高域的那些标识符。

d) 在所有的 MCS 连接上向下发送 DUin。不改变前转的用户标识符集, 以便所有被拆连的用户被通知到保留下来的那些用户。在正处在合并到较高域的前顶级 MCS 提供者处, 用户标识符均被限制到已经被证实了由较高域接受的那些用户标识符。

e) 在它们的子树中包含专用信道管理者或任何被允许的用户的所有的 MCS 连接上向下发送 CDin。

f) 在它们的子树中包含有一个或多个受影响的用户的所有 MCS 连接上向下发送 CAin 和 CEin。前转的用户标识符集可能被限制到保留在子树中的那些标识符。

g) 在它们的子树中包含有指明的接收者的单一 MCS 连接上向下发送 TVin。

h) 在它们的子树中包含有已抓获、禁止或正在给出令牌的用户的所有的 MCS 连接上向下发送 TPin。

i) 除了生成 SDin 时, 在 SDrq 到达的连接上不回送 SDin 的情况下, 在通过它们加入特定信道的所有 MCS 连接上向下传送 SDin 和 USin。

如果它们包含的标识符集为空, 那么不必前转指示 PCin、PTin、DUin、CAin 和 CEin。如果 MCS 提供者生成或前转响应类 MCSPDU, 它在唯一的 MCS 连接上向上传送。除非它的内容被确认为无效的, 它上升到顶级 MCS 提供者, 且顶级 MCS 提供者使其起作用。

如果 MCS 提供者生成或前转证实类 MCSPDU, 按下列规则在单一的 MCS 连接上向下传递该 MCSPDU。

a) MCcf 以相反的方向折回到还没有被证实应答的最早的 MCrq 的路径上。这需要每一 MCS 提供者保持悬挂请求的先入先出队列。

b) MTcf 按相反的方向折回到还没有被证实应答的最早的 MTrq 的路径上。这就要求每一 MCS 提供者保持悬挂请求的先入先出队列。

c) AUcf 以相反的方向折回到还没有被证实应答的较早的 AUrq 的路径。如果不止悬挂一个请求, 是哪一个 AUrq 并不是确定关键的, 但为公平起见, 每一 MCS 提供者应保持有一先入先出队列。对于发送 AUrq, MCS 提供者应记录它所包含的用户标识符正在分配给哪一个子树。

d) 其他证实类 MCSPDU 包含有一先前通过 AUcf 的动作, 按说明分配初始者用户标识符。在朝向分配用户标识符的子树的 MCS 连接上向下传递这些 MCSPDU。以这种方式继续下去, 它们甚至会返回到管理该请求 MCS 连入的提供者。

在像请求一样的处理过程中生成证实。由顶级 MCS 提供者生成除 CJcf 外的所有证实。

如果 MCS 提供者生成最终结果类 MCSPDU, 它在单一的连接上向上或向下传送。DPum 命令接收 MCS 提供者拆断传输它的 MCS 连接。RJum 拒绝具有协商码的错误 MCSPDU 并访问将它传递到拆断的 MCS 提供者。最终结果不被前转。

10 MCSPDU 的含义

表 4 到表 50 重述第 8 章中定义的每一 MCSPDU 的内容。

表 4 Connect-Initial MCSPDU

内 容	源	宿
主叫域选择符	请求	指示
被叫域选择符	请求	指示
向上标志	请求	指示
目标域参数	请求	指示
最小域参数	请求	指示
最大域参数	请求	指示
用户数据	请求	指示

10.1 Connect-Initial

Connect-Initial 由 MCS-CONNECT-PROVIDER 请求生成。它作为新 MCS 连接的初始 TC 上的第一个 TSDU 予以发送。在接收端, 它生成 MCS-CONNECT-PROVIDER 指示。

主叫传输地址和被叫传输地址为 MCS-CONNECT-PROVIDER 请求和指示的附加参数。它们变为 T-CONNECT 的参数并非显式地以任何 MCSPDU 进行传送。同一对传输地址将用来请求属于同一 MCS 连接的所有 TC。

传输的服务质量是 MCS-CONNECT-PROVIDER 请求的附加参数, 但不是指示的参数。服务质量可以从一个 TC 变化到另一个 TC, 且有效的质量仅在建立给定 TC 的过程中揭示。由于直到已经协商了所实现的数据优先级数的域参数后才知道所需的附加 TC 数, 该原语不能同时完全协商它们的传输服务质量。只要它满足任何主叫 MCS 提供者选用项规定的并由每个单一的 T-CONNECT 指示的任何最小项, 然后被叫 MCS 提供者应自动地以提供的服务质量接入 TC。

对每一提供者, 域选择符值的解释是本地的事宜。这些域选择符值是具有地址特征的八位组串。通过

设置 MCS 提供者的过程可以确定可接受的值。不止一个值可以选择同一域。非特定域选择符为一长度为零的八位组串。通过本地约定可以将其解释为某一明确的值。

向上标志规定新 MCS 连接的方向;如果认为被叫提供者比主叫提供者高,则其值为真,否则为假。MCS 提供者在它参与的 MCS 连接方向上的域分级结构上起一定的作用。提供者将不允许有两个到较高提供者的连接。没有连接到较高提供者的提供者应作为顶级提供者起作用。

Connect-Initial 的目标域参数应分别处于所规定的最小和最大值之间。MCS 提供者应修正所请求的域参数以便反应它执行的限制或强制域现有成员之间已经同意的值。这样可能增加最小值并减小最大值。仅为了将它们保持在区间之内而改变目标,MCS 提供者应准备兑付落在它所提供的值范围之间的任何响应。

用户数据是任意的八位组串。它的长度可以为零。

MCS 提供者在其能力范围内自动地接收每一个 TC,不使用 T-CONNECT 中的用户数据。数据传输中接收到的第一个 TSDU(或者是 Connect-Initial 或者是 Connect-Additional MCSPDU)确定了 TC 的性质。如果内容为不可接受的,被叫 MCS 提供者可能立即拆断该 TC。希望的反应是返回 Connect-Response 或 Connect-Result,当可能出现这种情况时,解释 MCS 连接失败的原因,然后拆断主叫 MCS 提供者。

10.2 Connect-Response

Connect-Response 由 MCS-CONNECT-PROVIDER 响应生成。它是在新 MCS 连接的初始 TC 上反方向发送的第一个 TSDU。它将 MCS 连接的接受传送到主叫 MCS 提供者,该主叫 MCS 提供者然后开始建立所需要的任何附加的 TC。

表 5 Connect-Response MCSPDU

内 容	源	宿
结果	响应	证实
域参数	响应	证实
被叫连接标识符	被叫提供者	主叫提供者
用户数据	响应	证实

如果结果是成功的,该 MCSPDU 固定运行中的域参数。在这些中,执行的 MCS 数据传送优先级数等于 MCS 连接的 TC 数。如果它超过 1,应创建附加的 TC 并通过交换 Connect-Additional 和 Connect-Result MCSPDU 将其限制到该 MCS 连接上。

被叫连接标识符作为具有该初始 TC 的被叫 MCS 提供者处的附加 TC 与该初始 TC 相关联的方法。为此目的单独选择它的值。在被叫提供者的进展中它将唯一地识别一条 MCS 连接。在 MCS-CONNECT-PROVIDER 完成之后,该标识符没有持续特征。

在 Connect-Response 中传送 MCS-CONNECT-PROVIDER 证实的大多数参数。如果结果是不成功或不需要附加的 TC,将立即生成证实。否则,它将延迟到知道将附加的 TC 组合到该 MCS 连接的结果为止。

10.3 Connect-Additional

Connect-Additional 在接收到 Connect-Response 之后生成。它作为在一条新 MCS 连接的附加 TC 上的第一个 TSDU 发送。数据优先级按顺序取值高、中和低,直到所需要的附加 TC 数。

表 6 Connect-Additional MCSPDU

内 容	源	宿
被叫连接标识符	主叫提供者	被叫提供者
数据优先级	主叫提供者	被叫提供者

10.4 Connect-Result

在接收到 Connect-Additional 之后生成 Connect-Result。它是新 MCS 连接的附加 TC 上反方向传送的第一个 TSDU。

表 7 Connect-Result MCSPDU

内 容	源	宿
结果	被叫提供者	证实

如果任何结果均不成功,则立即生成 MCS-CONNECT-PROVIDER 证实。与 MCS 连接相关联的所有 TC 将被拆断,同时将忽略它们传送的任何 MCSPDU。

否则,成功的结果应等待每一个附加的 TC。这些可能不按顺序返回。当收集到了所有的附加 TC 后,将生成成功的 MCS-CONNECT-PROVIDER 证实。

在 MCS-CONNECT-PROVIDER 证实之后,域 MCSPDU 可能流过 MCS 连接,每一 MCS 连接属于一个单一域。在 MCS 提供者管理多于一个域的配置时,携带 MCSPDU 的 MCS 连接确定它们应用到哪一个域。本章余下部分的域 MCSPDU 的描述建立在单个域的范围上。

10.5 PDin

在成功地完成 MCS-CONNECT-PROVIDER 证实之后生成 PDin。为确保没有建立环,它查明新 MCS 连接上 MCS 提供者的分级结构。为强制域的最大高度,顶级 MCS 提供者也生成 PDin。

表 8 PDin MCSPDU

内 容	源	宿
高度限制	前顶或顶	子序

在新 MCS 连接的较低端,由已不再是域顶的提供者生成 PDin。它的内容初始化到域的最大高度的域参数。在所有 MCS 连接上向下传送 PDin。

如果需要,在顶级 MCS 提供者处以同样的方式生成 PDin。

无论何处接收到 PDin,查看它包含的高度限值。如果大于 0,将限值减 1,同时在所有的 MCS 连接上向下前转 PDin。另一方面,值 0 意味着接收者距顶级 MCS 提供者太远了。它将作出向上拆断 MCS 连接的反应。它删去整个子树并帮助修改域的高度。

当出现环时,PDin 的高度限值必须减小到零。检查该环的那个提供者将打破该环,其方式是从域中删去环中的所有提供者和它们的下属。

注:具有纯本地 MCS 连接情况的 MCS 提供者不能阻止生成环。它能确保在所有次连接中最多有一个向上连接,但它不能确保向上连接不环回到某下级提供者。当生成环时,直接原因是顶级 MCS 提供者错误地向上连接。规定创建 MCS 连接的控制器的应用必须争取避免这样的错误。

10.6 EDrq

在成功地完成 MCS-CONNECT-PROVIDER 证实之后生成 EDrq。它向上传送在提供者高度上的变化和它们的强制吞吐量间隔。无论何时它的高度或间隔发生变化,MCS 提供者均生成 EDrq。

当增加或减少 MCS 连接时或当下级提供者通过 EDrq 报告有变化时,可能改变 MCS 提供者的高度。强制作为域参数规定的最小吞吐量的监测间隔可能根据调整到下级报告的间隔或由于其他原因而改变。如果任一值变化,MCS 提供者将 EDrq 传送到它的直接上级。

表 9 EDrq MCSPDU

内 容	源	宿
域中的高度	下级	较高提供者
强制吞吐量间隔	下级	较高提供者

10.7 MCrq

在成功地完成 MCS-CONNECT-PROVIDER 证实之后生成 MCrq。为将它们混合到合并域中,它向上传送由前顶级 MCS 提供者保持的信道属性。

表 10 MCrq MCSPDU

内 容	源	宿
合并的信道	前顶	顶级提供者
消除的信道标识符	中间提供者	顶级提供者

可以用多到 MCSPDU 大小的域限值的多信道属性填充 MCrq。按照第 7 章中 ASN.1 定义的详细论述,使用的 4 种信道的每一种(静态、用户标识符、专用的、分配的)具有它的相关属性集。这些保留在顶级 MCS 提供者的信息库中并部分地复制到使用信道的子树中。当合并两个域时,通过 MCS-CONNECT-PROVIDER 的作用,在较低域中使用的那些信道必须合并到较高域的信息库中或从较低域中清除出去。该决定由合并域的顶级 MCS 提供者负责。

应单独考虑每一信道。如果域对使用的信道限制允许该信道标识符,且该信道标识符没有发生信道标识符的冲突,较高域将包括它。固定信道标识符的使用绝不会出现冲突。如果较低域中的专用信道在较高域中也用作专用信道标识符,且管理者有相同的用户标识符,则它的使用是不会发生冲突的。不允许所有其他相同的使用组合,此时需要将信道标识符从较低域中消除掉。

如果专用信道具有一个较大的被允许的用户集,它的属性可能不适合单一的 MCrq,并应在多 MCSPDU 中向上发送。然而第二个或后续的合并相同专用信道的请求应该延时到已经收到响应第一个请求的 MCcf 之后。只有那时才会知道域限值是否已经允许信道在较高域中使用。如果第一个请求失败,它不应该对用户的余下子集重复上面的过程。

每一 MCrq 按同样的顺序从顶级 MCS 提供者引出一个 MCcf 答复。MCcf 不包含明显地标识前 MCrq 的任何内容。应仅按接收这些 MCSPDU 的次序送回答复。在前顶之上的 MCS 提供者应记录每一未答复的 MCrq 及它的到达之处,这样就可能通过同一条 MCS 连接返回相应的 MCcf。

为确保将它们合理地分配到 MCrq 起始的子树,中间 MCS 提供者应使专用信道属性中所表明的用户标识符有效。应删去无效的用户标识符。如果删去了专用信道管理者,应从合并请求中删去所有的信道属性,并且在要消除的集中应该仅包含信道标识符。除了确定该用户标识符的有效性外,中间 MCS 提供者不应改变 MCrq 的内容。

在前顶级提供者开始为合并提交静态的、分配的或专用的信道属性之前,前顶级提供者应等待所有用户标识符各自的证实和所有的令牌标识符已经合并到合并域中或被消除掉的证实。

10.8 MCcf

MCcf 答复前面的 MCrq。它返回相同的信道标识符集和属性子集。未加入到合并域中的信道属性作为被消除的信道标识符予以记录。

返回被接受的信道标识符,该信道标识符具有已经输入到顶级 MCS 提供者信息库中的属性。为保持一致,中间提供者应更新它们的信息库。

表 11 MCcf MCS-PDU

内 容	源	宿
合并信道	顶级提供者	中间提供者
消除的信道标识符	顶级提供者	前顶级提供者

仅通过标识符列出从较低域中消除掉的信道。如果在较高域中使用了相同的信道标识符，则将它们保持原样。中间的提供者将原样前转消除掉的信道标识符。

MCS 提供者利用一对一答复的知识将 MCcf 送回初始 MCrq 源中。MCcf 返回到生成 MCrq 的前顶级提供者。由于合并的信道已经保留在悬挂答复的信息库中，在那里可以将它们忽略。如对 PCin 一样，应删去消除的信道标识符。

中间 MCS 提供者应证实在专用信道属性中表明的用户标识符被分配到了 MCcf 选路的那个子树。如果初始 MCrq 为有效时，专用信道管理者已经被断开并同时被重新分配到其他地方，中间提供者应生成 CDrq，该 CDrq 使专用信道变为域合并者的牺牲者，并将信道标识符移入清除的集中。如果所允许的用户已经被重新分配到其他地方，在该信道中应不包括它们。

10.9 PCin

接收到 MCcf 后在前顶级提供者处生成 PCin。它向下广播并从下级提供者中消除特定信道标识符的使用。

表 12 PCin MCS-PDU

内 容	源	宿
拆断用户标识符	前顶	下级
消除信道标识符	前顶	下级

根据当前使用的信道标识符，清除的影响是：如果为一用户标识符信道，向所有的用户发送 MCS-DETACH-USER 指示；如果是一静态或分配的信道标识符，向加入的用户发送 MCS-CHANNEL-LEAVE 指示；如果是一专用信道标识符，向管理者发送 MCS-CHANNEL-DISBAND 指示，并向允许的用户发送 MCS-CHANNEL-EXPTEL 指示。

由于已经知道在它的较低域中使用的所有信道，前顶级提供者能仅利用信道标识符来生成适当的指示。然而它的下级可能仅有部分的知识。必须告知它们哪些信道标识符代表拆连的用户，为此种情况总要生成一指示和哪些代表其他类型的信道，此种情况仅在下级提供者使用该信道的情况下为它们生成一指示。这样，PCin 将要清除的信道标识符分为这样的两类。

除了由于 PCin 的接收者不再是顶级提供者，它不需要生成 CDin 或 TVcf 作为附加效应，通过 PCin 清除用户标识符应与通过 DUin 删掉用户标识符具有同样的结果。

注：在 PCin 中提供者可能收到没有加入的静态和分配的信道标识符或它的连入既不是管理者也不是允许用户的专用信道标识符。信息库中保留的使用记录允许提供者禁止对这样的信道标识符的初始指示。

10.10 MTrq

成功地接收到 MCS-CONNECT-PROVIDER 证实之后生成 MTrq。它向上传送前顶级提供者持有的令牌属性，这样它们就可以并入到合并域中。

表 13 MTrq MCSPDU

内 容	源	宿
合并令牌	前顶	顶级提供者
消除的令牌标识符	中间提供者	顶级提供者

可以用多达 MCSPDU 大小的域限制的多令牌属性填充 MTrq。如第 7 章中用 ASN.1 定义的详细论述,使用的令牌的每一状态(抓获的、禁止的、正在给出的、不可给出的、给出的)有它的相关属性集。在顶级 MCS 提供者的信息库中持有这些属性集,并部分地复制到使用令牌的子树中。当两个域合并时,通过 MCS-CONNECT-PROVIDER 的作用在较低域中使用的令牌必须合并到较高域的信息库中或从较低域中清除出去。该决定由合并域的顶级提供者负责。

应单独考虑每一令牌。如果使用的令牌的域限制允许且令牌标识符在使用上没有冲突,较高域应被扩展来包括它。如果较低域中禁止的令牌标识符在较高域中也被禁止,此种使用并不发生冲突。不允许所有其他的各种相同使用的组合,需要从较低域中将其清除。

如果一令牌具有较大的禁止用户集,它的属性不适合于单一的 MTrq 并应在多个 MCSPDU 中向下发送。然而,对同一被禁止的令牌的第二个或后续的请求将被延迟到接收到对第一个请求的答复 MTcf 之后。直到那时才知道域限制是否已允许在较高域中使用令牌。如果第一次请求失败了,对其余的禁止者子集不重复上面的过程。

每一 MTrq 按相同顺序从顶级 MCS 提供者处得到一 MTcf。MTcf 不包含任何明显地标识前 MTrq 的内容。仅按接收到这些 MCSPDU 的次序返回答复。在前顶级提供者之上的提供者应记录每一未回答的 MTrq 和它到达的地点,以便于可以通过同一 MCS 连接返回相应的 MTcf。

中间 MCS 提供者应使令牌属性中表明的用户标识符有效,以确保合理地将它们分配到起始 MTrq 的子树。应删除无效的用户标识符。如果抓获者或接收者中的一方(但并非两者)被删除,正处于给出状态的令牌应保持被抓获状态。如果通过该用户标识符的删除,令牌变为被释放,它的所有属性应从合并请求中删去,且清除集中仅应包含该令牌标识符。即使所有的禁止者均被删除,禁止的令牌在 MTrq 中应保持被禁止状态,而因为通过其他的 MCSPDU 禁止者可能还存在,则属性集变为空集。除了使该用户标识符变为有效外,中间 MCS 提供者不应改变 MTrq 的内容。

在开始向合并者提交令牌属性之前,前顶级提供者应等待对每一用户标识符已经被合并或清除掉的单独的证实。

10.11 MTcf

MTcf 答复 MTrq。它返回相同的令牌标识符集及其属性子集。未合并到合并域中的令牌属性记作被清除的令牌标识符。

表 14 MTcf MCSPDU

内 容	源	宿
合并令牌	顶级提供者	中间提供者
消除令牌标识符	顶级提供者	前顶

返回接受的令牌标识符连同已经输入到顶级 MCS 提供者信息库中的属性。为保持一致,中间的提供者将更新它们的信息库。

仅通过标识符列出从较低域中清除掉的令牌。如果在较高的域中使用了相同的令牌,它们将保持不变。中间提供者在不作任何改变的情况下前转令牌标识符。

利用一对一对答的知识,MCS 提供者将 MTcf 返回到起始 MTrq 的源处。MTcf 返回到生成 MTrq 的前顶级提供者。由于它们已经保留在悬挂一对答的信息库中,在那里可以忽略合并的令牌。如 PTin 一样,清除的令牌标识符应被删除。

中间 MCS 提供者应证实在令牌属性中表明的用户标识符被分配到 MTcf 迁回到的子树。如果起始 MTrq 为有效时,已经拆断了任何用户标识符并将其重新分配到其他地方,中间提供者应为它们生成具有原因码“信道清除”的 DURq,使它们成为域合并者的牺牲者。如果通过删除该用户标识符没有被禁止的令牌标识符变为释放的,则它将被移入清除集。

10.12 PTin

在接收到 MTcf 之后,前顶级提供者生成 PTin。将其向下广播并从下级提供者中清除特定的令牌标识符的使用。

表 15 PTin MCSPDU

内 容	源	宿
消除令牌标识符	前顶	下级

清除令牌的影响是严重的:向任何已经抓获、禁止或正在被给某一令牌标识符的任何用户发送 MCS-DETACH-USER 指示。提供者应通过为受影响的用户生成带有原因“令牌清除”的 DURq 来执行该影响。

注:可以预料将来可能修订 T.122,在这种情况下允许 MCS-TOKEN-RELEASE 指示。这样,即使已经剥夺了它使用令牌的权力仍允许受影响的用户保持它的连入。

10.13 DPum

由 MCS-DISCONNECT-PROVIDER 请求生成 DPum。反过来,在 MCS 连接的另一端生成 MCS-DISCONNECT-PROVIDER 指示。DPum 强制接收者拆断传送它的 MCS 连接。

当检测到在域分级结构中出现环的差错条件时,MCS 提供者也可生成 DPum。在这种情况下,原因为“非用户请求”。

表 16 DPum MCSPDU

内 容	源	宿
原因	请求的提供者	指示

10.14 RJum

当接收到一个无效的 MCSPDU 或检测到一个 MCS 协议差错时,MCS 提供者生成 RJum。由于不可能从不应出现的情况下恢复,它要求在 MCS 连接的另一端的对等提供者拆断。

RJum 诊断差错并返回出错 TSDU 的初始部分,典型的应有与所能适合的最大尺寸 MCSPDU 一样多的八位组。接收提供者可以选择拆断或保留。

表 17 RJum MCSPDU

内 容	源	宿
诊断	拒绝提供者	被拒绝的提供者
初始八位组	拒绝提供者	被拒绝的提供者

10.15 AUrq

由 MCS-ATTACH-USER 请求生成 AUrq。它上升到顶级 MCS 提供者,该顶级提供者返回一 AUcf 答复。如果域对用户标识符的限制允许,则生成一个新用户标识符。

表 18 AURq MCSPDU

内 容	源	宿
(无)	(无)	(无)

AURq 除了包含它的 MCSPDU 类型外不包含任何信息。由承载该 MCSPDU 的 MCS 连接确定用户要连入的域。生成的用户标识符仅有的初始特征是它的唯一确定性。

MCS 提供者将记录每一个接收到的未回答的 AURq, 及通过其到达的 MCS 连接, 以便能够将回复 AUcf 回送到同一个源。为公平地分布回复, 每一提供者应为该目的保留一先入先出队列。

10.16 AUcf

接收到 AURq 后, 顶级 MCS 提供者生成 AUcf。返回到请求提供者, 它生成 MCS-ATTACH-USER 证实。

表 19 AUcf MCSPDU

内 容	源	宿
结果	顶级提供者	证实
初始者(任选项)	顶级提供者	证实

当且仅当结果为成功时, AUcf 包含一用户标识符。接收到成功 AUcf 的提供者将用户标识符输入到它们的信息库中。

利用一对一回复的知识, MCS 提供者将 AUcf 返回到初始 AURq 的源。传送 AUcf 的提供者将通过 MCS 连接向下通知已分配了这个新的用户标识符, 以便以后在其他请求中出现时, 可以使该用户标识符有效。

10.17 DURq

由 MCS-DETACH-USER 请求生成 DURq。如果有效, 它上升到顶级 MCS 提供者, 该顶级 MCS 提供者从它的信息库中删去该用户, 并将该变化通过广播 DUin 以通知其他提供者。

表 20 DURq MCSPDU

内 容	源	宿
原因	请求提供者	顶级提供者
用户标识符	请求提供者	顶级提供者

MCS-DETACH-USER 请求生成包含有原因“用户请求”和单一用户标识符的 DURq。

当拆断向下 MCS 连接时, MCS 提供者也应生成 DURq。在那一点上, 在该受影响的子树中所有用户均被失去并应记作具有原因“域拆断”的拆连。如果用户标识符的分配正在悬挂, 下一个答复是 MCcf 或者是 AUcf, 将被停止并不返回到拆断子树中它的源处。它的提供者也生成 DURq, 以删去不可分配的用户标识符。

接收 DURq 的提供者应使得它包含的用户标识符有效, 以确保将它们被合理地分配到起始的子树。应删除无效的用户标识符。如果没有用户标识符存在, 应忽略 DURq。

在 DURq 中包含的用户标识符直到提供者接收到 DUin 时才从信息库中删去。这样就保持了与顶级 MCS 提供者的一致性。

注: 如果在 MCS 域中执行不止一个数据优先级, 在同一个用户较早以低优先级发送数据之前 DUin 可以到达一个给定

提供者。即使通过 DUin 报告发送者已经拆连之后,该协议不阻止向该连入投递数据。

10.18 DUin

顶级 MCS 提供者在接收到 DURq 后生成 DUin。向下将 DUin 广播到所有其他提供者,并在所有连入者处生成 MCS-DETACH-USER 指示。

在存在的连入处,DUin 为它所包含的每一用户标识符生成一 MCS-DETACH-USER 指示。被通知的用户以前是否知道被拆连用户的存在是无关紧要的。

接收到包含它本身用户标识符的 DUin 后,MCS 连入者不再存在。作为用户拆连的结果除用户标识符信道本身外,任何变为未加入的信道应通过 CLRq 释放。

表 21 DUin MCSPDU

内 容	源	宿
原因	顶级提供者	指示
用户标识符	顶级提供者	指示

接收到 DUin 的提供者应从它们的信息库中删去指定的用户标识符。如果没有其他被允许的用户,应删除由拆断用户管理的专用信道。如果其他用户存在,管理者的删除将使该顶级提供者向它们多点广播一 CDin。任何抓获的、正在给出的或由拆连用户禁止的令牌应相应地调整它们的状态。所希望的令牌接收者的删除将使得该顶级提供者向令牌给出者生成一不成功的 TVcf,除非它已经释放了该令牌或它本身也已拆断。

10.19 CJrq

由 MCS-CHANNEL-JOIN 请求生成 CJrq。如果有效,它上升到生成 CJcf 回复的、具有足够信息的 MCS 提供者。它可能是顶级 MCS 提供者。

表 22 CJrq MCSPDU

内 容	源	宿
初始者	请求提供者	较高提供者
信道标识符	请求	较高提供者

初始 MCS 连入的用户标识符由接收到初始请求的 MCS 提供者提供。随后接收到 CJrq 的提供者应使用用户标识符有效以确保合理地将其分配到初始子树。如果用户标识符无效,应忽略该 MCSPDU。

注:这尽可能地允许 CJrq 快速地跟踪向下流的清除初始用户标识符。首先接收到 PCin 的提供者此后不久可能接收到含有无效用户标识符的 CJrq。这是一般情况,并不引起拒绝 MCSPDU。

CJrq 可能上升到在它的信息库中已经有请求的信道标识符的 MCS 提供者。为与顶级 MCS 提供者一致,任何这样的提供者将同意是否请求应随后出现。如果请求失败,该提供者应生成不成功的 CJcf。如果它成功且该提供者已经被加入到同一信道,该提供者应生成一成功的 CJcf。在这两种情况下,MCS-CHANNEL-JOIN 在不必访问顶级 MCS 提供者的情况下完成。否则,如果请求成功但信道未被加入,提供者应向上前转 CJrq。

如果 CJrq 上升到顶级 MCS 提供者,请求的信道标识符可能是零,因为它是无效的用户标识符,它不在信息库中。如果域对使用的信道数的限制允许,应生成新分配的信道标识符,并将其在成功的 CJcf 中返回。如果请求的信道标识符处在静态范围内且域对使用信道数的域限制允许,信道标识符应输入到信息库中并同样在成功的 CJcf 中返回。

否则,仅当信道标识符已经在顶级 MCS 提供者的信息库中时,请求将成功。仅有该用户自身才能加入到用户标识符信道。专用信道标识符仅可以由它的管理者先前允许的用户加入。任何用户均可以加入

分配的信道标识符。

10.20 CJcf

较高 MCS 提供者在收到 CJrq 之后生成 CJcf。返回到请求提供者后, 它生成 MCS-CHANNEL-JOIN 证实。

表 23 CJcf MCSPDU

内 容	源	宿
结果	较高提供者	证实
起始者	较高提供者	MCS-PDU 选路
被请求的	较高提供者	证实
信道标识符(任选的)	较高提供者	证实

当且仅当结果为成功时,CJcf 包含所加入的信道标识符。

请求的信道标识符与 CJrq 中的相同。这就帮助了初始连入者将 MCS-CHANNEL-JOIN 证实与先前的请求相联系。由于 CJrq 不需要上升到顶级提供者, 证实可不按顺序出现。

如果结果是成功的,CJrq 将接收 MCS 提供者加入到特定的信道。然后较高提供者将用户发送到该信道的任何数据传送给它。只要它的任何连入者或下级提供者这样做了, 提供者就应保持加入到信道。要离开信道, 提供者将生成 CLrq。

接收到成功 CJcf 的提供者应将该信道标识符输入到它们的信息库中。如果已经不在那里, 根据它的范围, 信道标识符类型应为静态的或分配的。

应沿着初始用户标识符方向前转 CJcf。如果因为 MCS 连接不再存在, 用户标识符则为不可到达的, 提供者应确定是否它有理由保持加入到该信道。如果无理由, 它应生成 CLrq。

10.21 CLrq

为将其本身从信道集中移出,MCS 提供者生成 CLrq。该移动可以是来自加入到信道的最后一个连入者的 MCS-CHANNEL-LEAVE 请求。作为后果, 如果较高提供者也失去加入的原因, CLrq 继续上升。

接收到 CLrq 的提供者应停止将用户在该特定信道上发送的任何数据选路到承载它的 MCS 连接。当最后连入者或下级提供者离开信道时,MCS 提供者应生成相应的 CLrq。

表 24 CLrq MCSPDU

内 容	源	宿
信道标识符	请求提供者	较高提供者

10.22 CCrq

MCS-CHANNEL-CONVENE 请求生成 CCrq。如果有效, 它上升到顶级 MCS 提供者, 顶级 MCS 提供者返回 CCcf 作为回复。如果域对信道标识符的限制允许, 则生成一新专用信道。

CCrq 包含有初始用户标识符, 如对 CJrq 的解释一样, 该初始用户标识符应有效。

请求者成为专用信道的管理者。初始时信道未加入且它的管理者是仅有的被允许的用户。

表 25 CCrq MCSPDU

内 容	源	宿
起始者	请求提供者	顶级提供者

10.23 CCcf

在接收到 CCRq 后,顶级提供者生成 CCcf。返回到请求提供者后,它生成 MCS-CHANNEL-CON-
VENE 证实。

表 26 CCcf MCSPDU

内 容	源	宿
结果	顶级提供者	证实
起始者	顶级提供者	MCS-PDU 选路
信道标识符(任选的)	顶级提供者	证实

当且仅当结果为成功的时,CCcf 包含有专用信道标识符。

接收到成功的 CCcf 的提供者应将信道标识符作为具有作为它的管理者的初始用户标识符的专用信道输入到它们的信息库中。

沿初始用户标识符的方向前转 CCcf。如果因为 MCS 连接不再存在,用户标识符是不可到达的,不需要采取任何特别的动作,DUin 必须过后到达以通知初始者已经拆连了。由于初始者为它的管理者,这将从信息库中删除该信道标识符。

10.24 CDrq

由 MCS-CHANNEL-DISBAND 请求生成 CDrq。如果有效,它上升到顶级 MCS 提供者,该顶级 MCS 提供者删去专用信道标识符并生成 CDin。

表 27 CDrq MCSPDU

内 容	源	宿
起始者	请求的提供者	顶级提供者
信道标识符	请求	顶级提供者

为解散一信道,MCS 提供者在其本身初始化状态下也可生成 CDrq。

CDrq 包含初始用户标识符,初始用户标识符应为有效的,以确保将其合法地分配到起始子树。如果初始者不为专用信道的管理者,如同在信息库中所记录的一样,应忽略该 MCSPDU。

10.25 CDin

在接收到 CDrq 后,在顶级 MCS 提供者处生成 CDin。向下将其多点广播到它们的子树中包含有管理者或允许的用户的提供者。如果提供者发起,它对被允许的用户生成具有原因为“信道解散”的 MCS-CHANNEL-EXPTEL 指示,对管理者生成 MCS-CHANNEL-DISBAND 指示。

表 28 CDin MCSPDU

内 容	源	宿
信道标识符	顶级提供者	指示

当专用信道的管理者被拆连时,顶级 MCS 提供者也可以生成 CDin。

接收到 CDin 的提供者应从它们的信息库中删去该信道。

10.26 CArq

MCS-CHANNEL-ADMIT 请求生成 CArq。如果有效,它上升到顶级 MCS 提供者,该顶级 MCS 提供者允许特定的用户加入专用信道并多点广播 CAin,以通知它们所在的子树中的提供者。

CArq 包含初始用户标识符,如对 CDrq 的解释一样,该初始用户标识符应为有效。

由于只有顶级 MCS 提供者知道整个用户群,CArq 的其他用户标识符(表示被允许的那些用户)在顶级 MCS 提供者处应为有效。无效的那些应从结果 CAin 中删去。

表 29 CArq MCSPDU

内 容	源	宿
起始者	请求提供者	顶级提供者
信道标识符	请求	顶级提供者
用户标识符	请求	顶级提供者

直到提供者接收到 CAin 后才允许 CArq 中包含的用户标识符加入专用信道。这与顶级 MCS 提供者保持一致。

10.27 CAin

在接收到 CArq 后,顶级 MCS 提供者生成 CAin。然后向下多点广播到它们的子树中包含有新近被允许的用户的那些提供者。在受影响的连入处生成 MCS-CHANNEL-ADMIT 指示。

接收到 CAin 的提供者应正常地更新它们的信息库中的该信道,允许在它们的子树中的指定用户。然而,如果提供者是作为 MCS-CONNECT-PROVIDER 的结果正处于合并状态的较低域的前顶,它可以通过带有原因“信道清除”对受影响的用户标识符生成 DUrq 来拒绝允许。

表 30 CAin MCSPDU

内 容	源	宿
起始者	顶级提供者	指示
信道标识符	顶级提供者	指示
用户标识符	顶级提供者	MCSPDU 选路

10.28 CErq

由 MCS-CHANNEL-EXPEL 请求生成 CErq。如果有效,它上升到顶级 MCS 提供者,顶级 MCS 提供者从该专用信道中驱除特定的用户,并将 CEin 多点广播出去以通知它们所在的子树中的提供者。

CErq 包含有初始用户标识符,如对 CDrq 的解释一样,该用户标识符应为有效。

表示将被驱除的用户的 CErq 的其他用户标识符应在顶级 MCS 提供者处有效,该顶级 MCS 提供者独自知道整个允许的用户集。不被允许的那些应从结果 CEin 中删除。

CErq 中包含的用户标识符直到提供者接收到 CEin 后才被驱除。这样就与顶级 MCS 提供者保持了一致。

表 31 CErq MCSPDU

内 容	源	宿
起始者	请求提供者	顶级提供者
信道标识符	请求	顶级提供者
用户标识符	请求	顶级提供者

10.29 CEin

接收到 CErq 后,在顶级 MCS 提供者处生成 CEin。然后将其向下多点广播到它们子树中,它包含有

被驱除用户的提供者。它在受影响的连入者处生成带有原因“用户请求”的 MCS-CHANNEL-EXPEL 指示。

表 32 CEin MCSPDU

内 容	源	宿
信道标识符	顶级提供者	指示
用户标识符	顶级提供者	MCS-PDU 选路

接收到 CEin 的提供者将在它们的信息库中更新信道,从被允许加入信道的用户集中删去特定用户。如果允许加入专用信道的用户集为空集且管理者不在该子树中,应从信息库中删除该信道标识符。否则,作为驱除的结果,如果信道变为未加入的,提供者应生成相应的 CLrq。

对每一目标子树,前转 CEin 的提供者应计算是否后来包含允许加入专用信道的任何连入者。如果没有,提供者应推论出相应的下级提供者不再加入到专用信道中,而且,不必等待 CLrq,立即用将该影响更新其信息库。

10.30 SDrq

由 MCS-SEND-DATA 请求生成 SDrq。如果有效,它上升到顶级 MCS 提供者。沿着该路由提供者可由它生成具有相同内容的 SDin 并将其向下多点广播。

SDrq 包含有初始用户标识符,如对 CJrq 的解释一样,该初始用户标识符应为有效。

如果信道标识符在接收 MCS 提供者的信息库中列为专用信道并且 SDrq 的初始者不是被允许的用户,应忽略该 MCSPDU。

考虑到域中执行的优先级数,传送 SDrq 的初始或附加的 TC 应与它的数据优先级相匹配。应拒绝由于错误的 TC 而在 MCS 连接上到达的 MCSPDU。

表 33 SDrq MCSPDU

内 容	源	宿
起始者	请求的提供者	较早提供者
信道标识符	请求	较高提供者
数据优先级	请求	较高提供者
分段	请求的提供者	较高提供者
用户数据	请求	较高提供者

提供者应设置分段标志“开始”和“结束”,以表明 SDrq 中的数据与 MCS 服务数据单元的关系。只要这不破坏用户数据的整体性,提供者具有分段和再重组 MCSPDU 的自由,该 MCSPDU 是同一个 MCS 服务数据单元的部分。然而,由于在整个域中 MCSPDU 的最大大小为一常量,在这样的变换中没有什么好处。

提供者应由 SDrq 生成具有相同内容的 SDin,并将其传送到除了向上传送 SDrq 的下级提供者之外的所有加入到该特定信道中的提供者。除非信道在提供者的信息库中列为其子树中的用户标识符,它将向上前转 SDrq。

10.31 SDin

接收到 SDrq 后,在较高的 MCS 提供者处生成 SDin,将其向下多点广播,并在加入到该信道的所有连入者处生成 MCS-SEND-DATA 指示。

表 34 SDin MCSPDU

内 容	源	宿
起始者	较高提供者	指示
信道标识符	较高提供者	指示
数据优先级	较高提供者	指示
分段	较高提供者	指示提供者
用户数据	较高提供者	指示

考虑到域中执行的优先级数,传递 SDin 的初始或附加的 TC 将与它的数据优先级相匹配。应删除由于错误的 TC 在 MCS 连接上到达的 MCSPDU。

分段标志开始和结束允许将用户数据重组为完整的 MCS 服务数据单元。应在来自同一信道的相同用户的 SDin MCSPDU 的范围内以相同的优先级解释这些标志。要重组的分段流可能会与其他 MCSPDU 以及采用其他优先级来自其他信道的其他用户的数据相同。

如何将服务数据单元指示给连入的 MCS 用户是本地执行事宜。一种可能是将每一个 MCSPDU 连同包含的分段标志段作为一个分离的接口数据单元投递。另一种可能的实现是在接收提供者中重组收到的 MCSPDU,在这种情况下,应对大的服务数据单元作某些规定并应将服务数据单元按开始到达的相对次序反射给该用户。

接收到 SDin 的提供者应将其前转到加入该信道的所有下级。

10.32 USrq

由 MCS-UNIFORM-SEND-DATA 请求生成 USrq。如果有效,它上升到顶级 MCS 提供者。该顶级 MCS 提供者由它生成具有相同内容的 USin 并将其向下多点广播。

表 35 USrq MCSPDU

内 容	源	宿
起始者	请求提供者	顶级提供者
信道标识符	请求	顶级提供者
数据优先级	请求	顶级提供者
分段	请求的提供者	顶级提供者
用户数据	请求	顶级提供者

USrq 包含有初始用户标识符,如对 CJrq 的解释一样,该初始用户标识符应为有效的。

如果该信道标识符在接收 MCS 提供者的信息库中被列为专用信道并且 USrq 的初始者不是被允许的用户,应该忽略该 MCSPDU。

考虑到域中执行的优先级数,传送 USrq 的初始或附加的 TC 应与它的数据优先级相匹配。应拒绝由错误的 TC 在 MCS 连接上到达的 MCSPDU。

提供者应设置分段标志:开始标志和结束标志,以表明 USrq 中的用户数据与 MCS 服务数据单元边缘的关系。只要这不打乱用户数据的完整性,提供者具有分段和重组 MCSPDU 的自由,该 MCSPDU 为同一 MCS 服务数据单元的一部分。然而,由于在整个域中 MCSPDU 的最大尺寸为一常量,以这样的方式管理操作没有什么好处。

顶级 MCS 提供者应由 USrq 生成具有相同内容的 USin。

10.33 USin

接收到 USrq 后,顶级 MCS 提供者生成 USin。将其向下多点广播并在加入到该信道的所有连入处生成 MCS-UNIFORM-SEND-DATA 指示。

表 36 USin MCSPDU

内 容	源	宿
起始者	顶级提供者	指示
信道标识符	顶级提供者	指示
数据优先级	顶级提供者	指示
分段	顶级提供者	指示提供者
用户数据	顶级提供者	指示

考虑到域中执行的优先级的数量,传递 UDin 的初始或附加的 TC 应与它的数据优先级相匹配。应拒绝由错误的 TC 在 MCS 连接上到达的 MCSPDU。

分段标志开始和结束允许将用户数据重组为完整的 MCS 服务数据单元。应在以相同的优先级来自同一信道的相同用户的 SDin MCSPDU 的范围内来解释这些标志。要重组的分段流可能会与其他 MCSPDU 和以其他优先级来自其他信道的其他用户的数据相间。

如何将服务数据单元指示给连入的 MCS 用户是本地执行事宜。

接收到 USin 的提供者应将其前转到加入该信道的所有下级。

10.34 TGrq

由 MCS-TOKEN-GRAB 请求生成 TGrq。如果有效,它上升到顶级 MCS 提供者,顶级 MCS 提供者返回 TGcf 回复。

表 37 TGrq MCSPDU

内 容	源	宿
起始者	请求提供者	顶级提供者
令牌标识符	请求	顶级提供者

TGrq 包含有初始用户标识符,如对 CJrq 的解释一样,该初始用户标识符应为有效的。

如果令牌为自由的且域对令牌的限制允许,它应变为被抓获的。如果该令牌仅被请求用户禁止,应变为被抓获的。否则,令牌的状态将不改变。

10.35 TGcf

接收到 TGrq 后,顶级 MCS 提供者生成 TGcf。返回到请求提供者后,它生成 MCS-TOKEN-GRAB 证实。

表 38 TGcf MCSPDU

内 容	源	宿
结果	顶级提供者	证实
起始者	顶级提供者	MCS-PDU 选路
令牌标识符	顶级提供者	证实
令牌状态	顶级提供者	证实

如果令牌先前是自由的或由同一用户将令牌从禁止的转为抓获的,结果应为成功的。其他的结果是令牌太多或令牌不可得到。后者应用于已经由请求者抓获的令牌;检查令牌状态便可断定令牌状态。

接收到 TGcf 的提供者为保持与返回的状态相一致应更新它们信息库中的令牌状态。

应将 TGcf 沿初始用户标识符的方向前转。如果由于 MCS 连接不再存在,用户标识符不可到达,不需要采取特别的操作,由于过后一定会到达一个 DUin 以通知初始者已经拆连了。这将释放信息库中对令牌标识符的保持。

10.36 TIrq

由 MCS-TOKEN-INHIBIT 请求生成 TIrq。如果有效,它上升到顶级 MCS 提供者。该顶级 MCS 提供者返回 TIcf 回复。

表 39 TIrq MCSPDU

内 容	源	宿
起始者	请求提供者	顶级提供者
令牌标识符	请求	顶级提供者

TIrq 包含有初始用户标识符,如对 CJrq 的解释一样,该初始用户标识符应为有效的。

如果该令牌为自由的且使用令牌数的域限制允许,它应变为被禁止的。如果令牌由请求用户抓获,它将变为被禁止的。如是令牌已经为被禁止的,则将请求者加入到禁止者集中。否则,令牌的状态将不改变。

10.37 TIcf

接收到 TIrq 后,在顶级 MCS 提供者处生成 TIcf。返回到请求提供者后,它生成 MCS-TOKEN-INHIBIT 证实。

如果令牌先前是自由的或被禁止的或由同一用户将令牌从抓获的转为被禁止的,结果为成功的。其他的结果是令牌太多或令牌不可得到。

表 40 TIcf MCSPDU

内 容	源	宿
结果	顶级提供者	证实
起始者	顶级提供者	MCSPDU 选路
令牌标识符	顶级提供者	证实
令牌状态	顶级提供者	证实

接收到 TIcf 的提供者应在它们的信息库中更新令牌状态,以确保与返回的状态一致。

该 MCSPDU 以与 TGcf 相同的方式选择路由。

10.38 TVrq

由 MCS-TOKEN-GIVE 请求生成 TVrq。如果有效,它上升到顶级 MCS 提供者,顶级 MCS 提供者生成 TVin 或不成功的 TVcf。

表 41 TVrq MCSPDU

内 容	源	宿
起始者	请求提供者	顶级提供者
令牌标识符	请求	顶级提供者
接收者	请求	顶级提供者

TVrq 包含有初始用户标识符,如对 CJrq 的解释一样,该初始用户标识符应为有效的。

如果请求者抓获了该令牌,且拟接收者存在,应将 TVin 传送到接收者。否则,请求将失败,令牌的状态将不改变,应向请求者传送具有结果令牌不可拥有或无这样的用户的 TVcf。

10.39 TVin

接收到 TVrq 后,在顶级 MCS 提供者处生成 TVin。返回到拟接收者后,它生成 MCS-TOKEN-GIVE 指示。

接收到 TVin 的提供者通常将它们的信息库中令牌标识符更新到由初始者给接收者的状态。然而,作为 MCS-CONNECT-PROVIDER 的结果如果提供者是正在合并的较低域的前顶,它可以通过生成含有原因为“域合并”的 TVrs 来拒绝所提供的令牌。

表 42 TVin MCSPDU

内 容	源	宿
起始者	顶级提供者	指示
令牌标识符	顶级提供者	指示
接收者	顶级提供者	MCSPDU 选路

应沿接收者用户标识符的方向前转 TVin。如果因为 MCS 连接不再存在,用户标识符为不可到达的,由于随后必定到达一 DUin 来通知该接收者已经拆连。则不需要进行任何特殊的操作。这将释放信息库中对该令牌标识符的保持。

10.40 TVrs

由 MCS-TOKEN-GIVE 响应生成 TVrs。如果有效,它上升到顶级 MCS 提供者,顶级提供者生成 TVcf 以通知输出令牌的捐献者。

表 43 TVrs MCSPDU

内 容	源	宿
结果	响应	顶级提供者
接收者	响应提供者	顶级提供者
令牌标识符	响应提供者	顶级提供者

成功的结果应表明接收者接受了所提供的令牌。

接收到该原语响应的 MCS 提供者提供响应 MCS 连入者用户标识符。随后接收到 TVrs 的提供者应使用用户标识符有效,以确保将其合理地分配到初学者树。如果用户标识符无效,应忽略该 MCSPDU。

如果在提供者的信息库中未将该令牌标识符列为正在交送接收者状态,应忽略 MCSPDU。若结果为成功的,因而捐献者仍抓获该令牌,令牌的状态应被更新为由接收者抓获;否则,根据捐献者是否位于提供者的子树中,它将返回到由捐献者抓获或从信息库中将其删除。如果该令牌标识符已经被捐献者释放且结果为不成功的,则应从提供者的信息库中将该令牌删去。

如果 MCSPDU 无效且被忽略,它将向上前转。顶级 MCS 提供者应按上面的规定处理 TVrs。另外,如果捐献者没有释放该令牌,顶级提供者应生成包含有与 TVrs 具有相同结果的 TVcf。

10.41 TVcf

接收到 TVrs 后,在顶级 MCS 提供者处生成 TVcf。返回到请求提供者时,它生成 MCS-TOKEN-GIVE 证实。

表 44 TVcf MCSPDU

内 容	源	宿
结果	顶级提供者	证实
起始者	顶级提供者	MCSPDU 选路
令牌标识符	顶级提供者	证实
令牌状态	顶级提供者	证实

如果不能将令牌提供给拟接收者,顶级提供者接收到 TVrs 后也应生成 TVcf。它替代生成 TVin。如果在接收到 TVrs 之前接收者已经拆连,也生成带有结果为“没有这个用户”的 TVcf。

接收到 TVcf 的提供者为保持与返回状态的一致性应更新它们信息库中的令牌状态。

该 MCSPDU 按与 TGcf 一样的方式选择路由。

10.42 TPrq

由 MCS-TOKEN-PLEASE 请求生成 TPrq。如果有效,它上升到顶级 MCS 提供者,该顶级 MCS 提供者将 TPin 多点广播,以告知该令牌的当前用户。

TPrq 包含有初始用户标识符,如对 CJrq 的解释一样,该初始用户标识符应为有效的。

表 45 TPrq MCSPDU

内 容	源	宿
起始者	请求提供者	顶级提供者
令牌标识符	请求提供者	顶级提供者

10.43 TPin

接收到 TPrq 后,在顶级 MCS 提供者处生成 TPin。将其向下多点广播并生成 MCS-TOKEN-PLEASE 指示。

接收到 TPin 的提供者将其前转到其子树中包含有已经抓获、禁止或正在给出该指定令牌的用户的所有下级。

表 46 TPin MCSPDU

内 容	源	宿
起始者	顶级提供者	指示
令牌标识符	顶级提供者	指示

10.44 TRrq

MCS-TOKEN-RELEASE 请求生成 TRrq。如果有效,它上升到顶级 MCS 提供者,该顶级 MCS 提供者返回 TRcf 作为答复。

表 47 TRrq MCSPDU

内 容	源	宿
起始者	请求提供者	顶级提供者
令牌标识符	请求提供者	顶级提供者

TRrq 包含初始用户标识符,如对 CJrq 的解释一样,该初始用户标识符应为有效的。

如果请求者抓获该令牌,它将变为自由的。如果该令牌为被禁止的,应从禁止者集中移出该请求者;如果该集变空,令牌将变为自由的。如果令牌正处于从请求者中给出的过程中,它应进入某一独特的被给予拟接收者的中间状态,悬挂 TVrs 的接收。否则,令牌的状态将不改变。

10.45 TRcf

接收到 TRrq 后,在顶级 MCS 提供者处生成 TRcf。返回到请求提供者时,它生成 MCS-TOKEN-RELEASE 证实。

表 48 TRcf MCSPDU

内 容	源	宿
结果	顶级提供者	证实
起始者	顶级提供者	MCSPDU 选路
令牌标识符	顶级提供者	证实
令牌状态	顶级提供者	证实

如果请求者抓获或禁止该令牌或请求者正处于将令牌给出的过程中,结果应为成功的。其他可能的结果是令牌不可获得。

接收到 TRcf 的提供者将在它们的信息库中更新令牌状态,以保持与返回状态的一致性。

按与 TGcf 一样的方式为该 MCSPDU 选择路由。

10.46 TTrq

MCS-TOKEN-TEST 请求生成 TTrq。如果有效,它上升到顶级 MCS 提供者,该顶级 MCS 提供者返回 TTcf 作为回复。

TTrq 包含初始用户标识符,如对 CJrq 的解释一样,该初始用户标识符应为有效的。

表 49 TTrq MCSPDU

内 容	源	宿
起始者	请求提供者	顶级提供者
令牌标识符	请求	顶级提供者

10.47 TTcf

接收到 TTrq 后,在顶级 MCS 提供者处生成 TTcf。返回到请求提供者时,它生成 MCS-TOKEN-TEST 证实。

接收 TTcf 的提供者应在它们的信息库中找到与返回状态相一致的令牌。

按与 TGcf 一样的方式为该 MCSPDU 选择路由。

表 50 TTcf MCSPDU

内 容	源	宿
起始者	顶级提供者	MCSPDU 选路
令牌标识符	顶级提供者	证实
令牌状态	顶级提供者	证实

11 MCS 提供者信息库

11.1 分级复制

尽管一个 MCS 提供者可以管理多个域,但它分别管理每一个域。从逻辑上讲它对每一个域保留不同的信息库,以记录使用的信道和令牌资源的状态。后面的描述建立在单个域的基础上。

一个域需要管理的 MCS 资源为信道标识符和令牌标识符,用户标识符是信道标识符的子集。域参数限制了可以同时使用的每类标识符数。这允许提供者计算在充分利用域的最坏情况下信息库所需的存储空间的大小。

在一个域的分级结构中,在任何给定的 MCS 提供者处使用的标识符稳定在它的直接上级使用的那些标识符的子集中。有关标识符的信息记录在可用于支持包含那个标识符的 MCS 服务处。为及时更新数据信息,更广泛的记录信息使得 MCSPDU 业务量中增加了附加的花销。由于在 MCSPDU 传输延时的限制内,提供者中记录的信息与在较高提供者中记录的那些信息一致,因而可以说该提供者信息库是通过域分级结构部分地复制的。

随着域的第一条 MCS 连接的建立,域参数变为固定不可变的。某些提供者在缺少在每一类中规定的标识符最大数目的容量可能以虚假的借口协商加入一个域。它可能推测在分级结构中处在它所在的那样低位置将不会被要求保留有多于整个信息库中的一小部分内容。这样的一个提供者不可能支持具有他们希望的全部 MCS 服务的连入和下级。然而,在实际上它们能力已经超出了之后,这样的一个提供者才能够作为域的平等成员出现。该策略可能有助于具有有限能力的终端节点的参与。

在顶级提供者处首先使用标识符。通过有选择的 MCSPDU 的下行流,下级提供者使用标识符。以同样的自顶向下的方式删除大多数标识符的使用。因为在传输中保留有删除的标识符的 MCSPDU,因而在下级提供者将标识符记为在使用之中,而它的上级提供者又未记录的那段时间内有必要有间隔。然而,不允许这种情况。按它们传送的次序接收和处理控制 MCSPDU。MCSPDU 的处理结果(包含有信道和令牌标识符的生成和删除)在注意力移到下一个输入事件之前才起作用。

前段的一个例外是删除静态的和分配的信道标识符。尽管由 CJcf 的下行使得这些信道标识符处于使用状态,但却以相反的次序删除它们,即从下向上删除。特别地,当来自连入的 MCS-CHANNEL-LEAVE 请求和来自下级提供者的 CLrq MCSPDU 一同组合以释放未加入的信道时,将其删除。这样的转变促使 CLrq 进一步向上传送。因此,在这两种情况下,提供者记录的处于使用状态的信道标识符一定为上级提供者所记录的处于使用状态的信道标识符的绝对子集。这是数据传送之前加速信道管理的最优化设计的偶然结果。

MCcf、AUcf、CJcf、CCcf 和 CAin 使得信道标识符处于使用状态;由 MCcf、PCin、DUin、CLrq、CDin 和 CEin 删除信道标识符。MTcf、TGcf、TICf 和 TVin 使令牌标识符处于使用状态。由 MTcf、PTin、TRcf、TVrs 和 TVcf 删除它们。在给定提供者处使用一标识符时,作为起因的 MCSPDU 可能前转到零个、一个、几个或所有的下级提供者。例如,随着单独用户被允许加入某一专用信道或从某一专用信道驱除标识符的使用可能逐渐地增加或减少。当从给定的提供者处删除一标识符时,影响它的 MCSPDU 被前转到可能仍将该标识符记录为在使用状态的所有下级。

连入到该域的用户最终将标识符的使用与该用户对信道或令牌的动作联系起来(尽管如解释的一样,通过 MCSPDU 的传输在通信变化中,可能有些延时。)。给定的 MCS 提供者处稳定记录处于使用状态的标识符是那些在该提供者的子树上由某用户实际使用的标识符。因而,它们是任何较高提供者稳定记录的那些标识符的子集。

删除用户标识符的必然结果是删除子树中属于唯一用户的信道标识符和令牌标识符。

下列各节中规定了考虑信道标识符和令牌标识符处于使用状态的准则。

11.2 信道信息

4 种信道具有其各自的准则,以确定一给定连入是否被认为是正在使用的信道标识符,以及它是否应

在提供者的信息库中出现：

- 静态信道标识符(范围 1~1000)处于使用中,如果用户通过一成功的 MCS-CHANNEL-JOIN 证实加入该信道且未通过 MCS-CHANNEL-LEAVE 请求或指示离开该信道。
- 用户标识符信道处于使用状态,如果由成功的 MCS-ATTACH-USER 证实将其分配给一用户并且用户未通过 MCS-DETACH-USER 请求或指示拆连。
- 专用信道标识符处于使用状态,如果用户通过成功的 MCS-CHANNEL-CONVENE 证实创建该信道或用户已经通过 MCS-CHANNEL-ADMIT 指示被允许加入它并且未通过 MCS-CHANNEL-EXPTEL 指示驱出,同时信道也未通过 MCS-CHANNEL-DISBAND 请求或指示解散。
- 分配的信道标识符处于使用状态,如果用户通过成功的 MCS-CHANNEL-JOIN 证实已加入到该信道,且未通过 MCS-CHANNEL-LEAVE 请求或指示离开该信道。

对处于使用状态的信道标识符应记录下列信息：

- 它代表的信道类型(静态的、用户标识符、专用的或分配的)。
- 该信道由哪个 MCS 连入,以及是通过哪些 MCS 到下级提供者的连接加入的。
- 如果对一用户标识符信道,朝向它的方向,也就是分配到用户标识符的那个本地 MCS 连入或者是在用户所在的子树中连到下级提供者的那个向下 MCS 连接。
- 如果是专用信道,召集该专用信道的管理者的用户标识符(无论管理者本身是否在该提供者的该子树中)以及已经允许加入到该信道的提供者子树中的所有用户标识符。

按第 10 章的解释,使用对信道标识符所记录的那些信息,以使请求 MCSPDU 有效并为指示和证实 MCSPDU 选择路由。

11.3 令牌信息

图 4 示出了令牌标识符的状态转变。

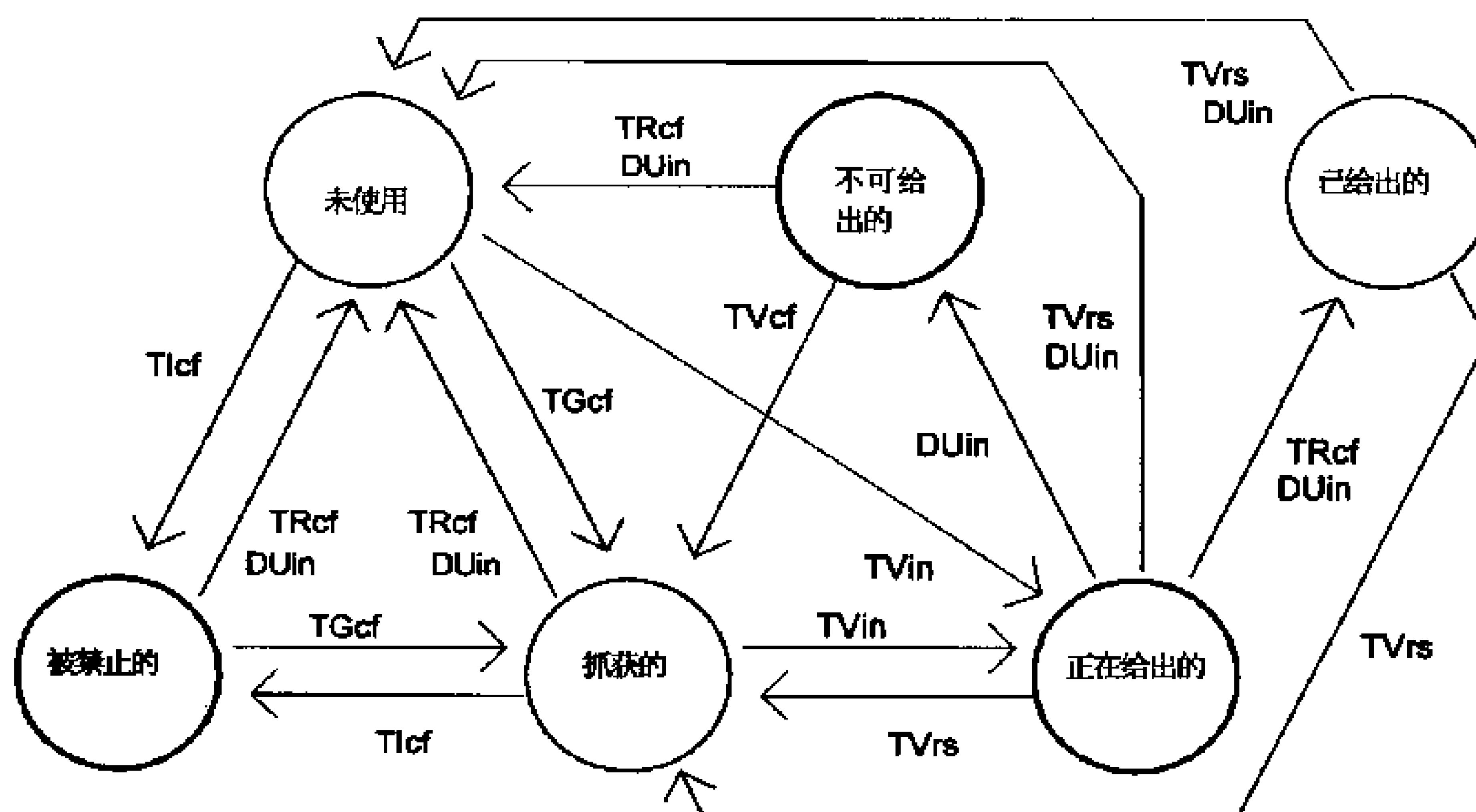


图 4 令牌标识符的状态转变

单独的令牌标识符可以由单个的用户抓获或由一个或多个用户禁止。TVin 的动作沿着从顶级 MCS 提供者朝向拟接收者的域分级结构分支将状态转变到“正在给出”。如果接收者的提供者用 TVrs 响应 TVin 之前该接收者已经拆连,该状态将变为不可给出的。另一种情况,如果捐献者明显地释放了令牌或拆断了,就断定其为已给出的状态。在交出令牌的过程中,朝向捐献者的域分级结构的分支和朝向接收者的分支至少在顶级 MCS 提供者处相交。令牌的状态仅沿着该相交处从被抓获的状态变为正在给出状态,也可能变为不可给出或已给出的状态。

令牌的用户作为抓获者、禁止者、接收者或者同时作为抓获者和接收者(当将令牌给它本身时)与它相联系:

a) 用户为抓获者,如果用户已通过成功的 MCS-TOKEN-GRAB 证实占有令牌且未通过 MCS-TOKEN-RELEASE 请求或成功的 MCS-TOKEN-GIVE 证实释放它也未用成功的 MCS-TOKEN-INHIBIT 证实转变它或已经通过成功的 MCS-TOKEN-GIVE 响应接受所提供的令牌。

b) 用户为禁止者,如果用户通过成功的 MCS-TOKEN-INHIBIT 证实占有令牌且未通过成功的 MCS-TOKEN-RELEASE 请求释放它,也未通过成功的 MCS-TOKEN-GRAB 证实转变它。

c) 用户为接收者,如果用户已经通过 MCS-TOKEN-GIVE 指示提供给它一个令牌且未通过不成功的 MCS-TOKEN-GIVE 响应释放该令牌。

对正在使用的令牌标识符应记录下列信息:

- a) 在该 MCS 提供者处的令牌标识符的状态(不必与顶级提供者处的一致);
- b) 如果处于被抓获或不可给出状态,该提供者子树中的抓获者的用户标识符;
- c) 如果处于正在给出状态,抓获者的用户标识符(无论抓获者是否在该提供者的子树中);
- d) 如果处于给出或已给出状态,该提供者子树中提供者的用户标识符;
- e) 如果处于禁止的状态,已经禁止该令牌的提供者子树中的所有用户标识符集。

对使用的令牌标识符所记录的信息,按第 10 章中的解释使用,以使响应 MCSPDU 有效并为指示 MCSPDU 选择路由。

下级提供者处的令牌标识符的状态不必与顶级 MCS 提供者的内容一致。这是因为令牌捐献者一般不处理 TVin 或 TVrs 且接收者一般不处理捐献者的 TRcf。图 5 示出了在复杂的令牌交互作用中可能出现的状态。

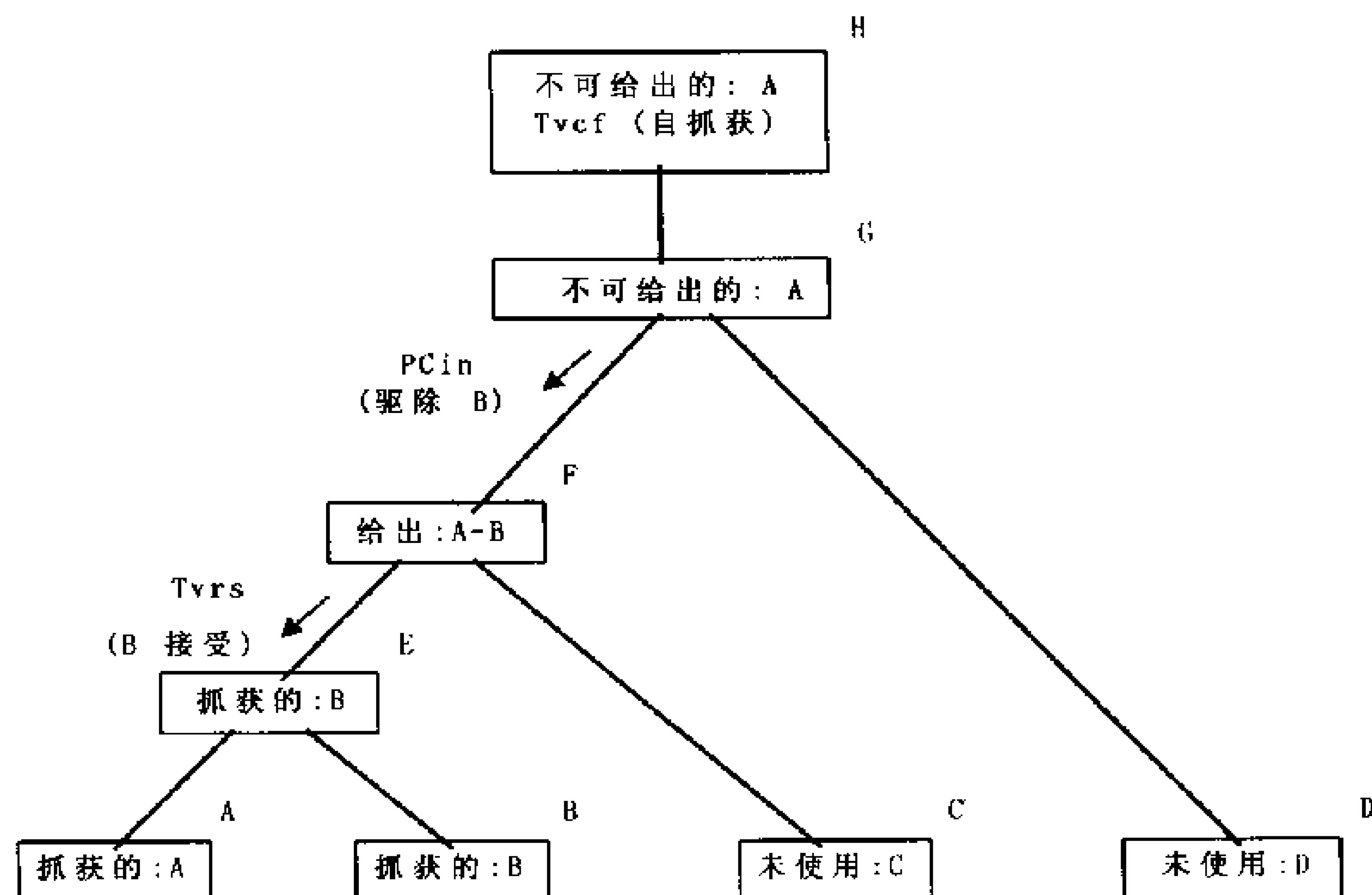


图 5 在复杂的交互作用中可能出现的令牌状态

该图集中说明在提供者 A 到 H 的信息库中的一个令牌标识符。假定先前情况是连入到提供者 A 的用户 A 将该令牌给予连入到提供者 B 的用户 B。然而,在用户 B 响应之前,提供者 G 将该域向上连接到一新顶级提供者 H,并开始用户的域合并。由于在信道标识符上的冲突,新域不能接受 B,该新域通过 PCin 启动它的拆连。如图所示,该 MCSPDU 已经通过提供者 G 和 D 部分完成它的传送过程,结果是,提供者 G 已经将它的令牌状态从正在给出状态调整到不可给出状态并以该状态将其合并到新域。具有这样一种状态的令牌出现的该新顶级提供者 H 排队传送不成功的 TVcl,以将该令牌返回到用户 A。在所示时刻,该

令牌最终由用户 B 接受,同时 TVrs 沿着域分层结构上接收者的分支向上传送,将每一提供者处该令牌的状态从正在给出变换到由 B 抓获。两个 MCSPDU 目前会聚到提供者 F,且先到的那一个确定它后序的状态;或者返回到由 A 抓获或者暂时地由 B 抓获,然后在用户 B 被拆连后再变为未使用状态。无论哪种情况,当新顶级提供者传送它所悬挂的 TVcf 时,令牌将固定在由 A 抓获状态。

12 过程单元

12.1 MCS 顺序

由于 MCS 提供者在单一的 TC(MCS 连接的初始 TC)上传送控制 MCSPDU,在任何一对 MCS 提供者之间按顺序保持控制 MCSPDU。MCS 提供者应处理接收到的 MCSPDU 并应按相同次序传送任何生成的输出 MCSPDU。这应用于被简单地前转或下转到域分层结构以及应用于传送来的 MCSPDU,例如将请求或响应转换成指示或证实。如果由于沿 TC 的流量控制的后压有必要为后面的传送排队输出,则在 MCS 提供者中应保持 MCSPDU 的排序。

不必按顺序保留不同优先级的数据 MCSPDU。相反,相关优先级的优点仅仅是当高优先级数据在低优先级数据之前到达时实现。这意味着它应在单独的 TC 上传送,并且分别在每一 MCS 提供者中排队。即使域中执行的数据优先级数少于最大数,较少的 TC 有效,那么选择这样做的提供者内部可能仍保留单独的队列。这实现了相关优先级的部分而并非全部的优点。

MCS 提供者应保持在给定优先级上传送的数据 MCSPDU 的次序。这比由 ITU-T 建议 T. 122 所规定的限制要严,建议 T. 122 仅保证在同一目的地的信道上按给定优先级传送的服务数据单元的排序。

最高优先级的控制 MCSPDU 和数据 MCSPDU 在同一初始 TC 上传送,且应引起 MCS 提供者相同的注意。较低优先级的数据可能落在后面。超前的控制指示可能实际上在先传送的较低优先级数据之前到达。

12.2 输入流量控制

MCS 提供者具有有时相互冲突的目标:快速地将数据通过域,而不管临时阻塞某些接收者;给予传送者平等的接入可用带宽的机会,并防止一些参加者落后于正在接收同一多点广播数据的对等体之后太远。

MCS 是保证用户数据完整性的可靠服务。由于提供者在不能立即传送数据时,已经限制了存储 MCSPDU 的能力,它必须具有预防有时拒绝将来输入的能力。由于与传送服务的接口的细节为本地的事宜,抽象的影响必定是在将来某时在撤除流量控制时,完整地按顺序接收入 TSDU,在 TC 管道中保持入 TS-DU。当 TC 管道填满时,远端 MCS 提供者可能发现它们会受传送将来 MCSPDU 的反压的阻止,并且可能需要调用类似的防御措施。

在 MCS 协议中没有明显地标明流量控制。这是较低层的功能,重复将是一种浪费。因此,通过前 TC 的媒体很难断定远端提供者是否阻止将来的输入。为尽可能地满足冲突的目标,建议下面描述的原则。

MCS 提供者可以给予每一入 TC 一个固定量的缓冲区,这些缓冲区可以在使用反压之前用 MCSPDU 来填充。按本协议的规定处理每个缓冲区,然后给输出分配零个或多个出 TC。可能立即出现输出或可能由于传送管道满而延时。直到将它输出到最后的 TC 上,才告知缓冲区通过 TC 到达的该 TC 的输入量。在该输入量输出到所有需要的 TC 后,它将作为对那个输入量的增量而循环。当用完了缓冲区时,就停止在相应的 TC 上进一步的输入。在考虑了 TC 是否为向上或向下的 MCS 连接的一部分和它所代表的数据优先级后才可以设置输入量。

缓存可以减小传送者和接收者之间在速率上的差异。输入量可以阻止任一 TC 独自控制资源。它也能限定同一多点广播数据的两个接收者收到的数据失步的可能差异。然而,所建议的方案不足以明显地预测使用的所有样式,而且有时在具有可接受的替代方式时可能减慢通过域的数据传送速率。较好的流量控制方案的发明(本地执行以及不需要附加的 MCSPDU 通信)可能是决定产品的因素。

12.3 强制吞吐量

与输入流量控制相反,MCS 协议中对强制吞吐量的支持是明显的。首先,强制的速率为通过 MCS-

CONNECT-PROVIDER 协商的域参数。其次,在采取相反动作之前,每一 MCS 提供者通过 EDrq 将监视吞吐量的时间间隔传送给它的上级。为描述强制吞吐量间隔需要提供者共享一些共同的行为准则。然而,强制吞吐量保留各自技术发明的余地。

在每一接收者处强制最小输入速率是对控制器应用可用的选择,该控制器应用建立一个域的 MCS 连接。通过域参数强制吞吐量参数选项,该选项以每秒八位组数的形式表示。当很直观地看出不应允许一方以任意慢的速度运行,而在其他方之间防碍数据传送时,在太严格地寻求强制吞吐量方面是有危险的。

多传送者的复杂样式是一个难题。强制吞吐量策略作用于其上的反压种类可能不简单的是不正常的慢接收者的结果。首先,向下的 MCSPDU 必须与从下面来但被反射回去的 MCSPDU(特别是 SDin)竞争以取得注意。然后通过一对等提供者的经验性的向下流,可能仅获得标称带宽的一小部分并可能随着加入到对等的其他连接和连入者数动态地变化。其次,只能希望最高优先级 MCSPDU 在 MCS 连接上连续地传送。由于从其他源以较高优先级接收到的业务量密度较低的对等提供者可能将较低优先级适当地阻塞较长时间。最后,如果按在 MCS 连接上接受到业务量之前必须等待的受阻 MCSPDU 的长短来测量,在瞬间吞吐量方面可能有很大的不同。在其他方面,这可能取决于其他源传送了多少 MCSPDU 并以何种次序在该对等提供者中排队。

无论如何,在已知数据传递方较一致的一些实际情况下,强制吞吐量是一个有价值的选择。这可解释为在强制 MCS 提供者所规定的时间间隔上,对每一直接的 MCS 连入和每一 MCS 向下连接所要求的 MCSPDU 的最小输出量。每一 MCSPDU 输出,包括控制和数据,应计算吞吐量,就如同它是域参数允许的最大量。到 MCS 连入的 MCSPDU 输出应意味着相关联的原语指示和证实的传递。到向下 MCS 连接的输出将意味着在传送服务接口处无反压并接受到相应的 TC 管道。

不论为何种数据优先级,只要到给定连入或向下连接上有一个或多个 MCSPDU 在排队,就应监视输出。只要队列为空,就应停止监视,表明无强制作用。同时,对为补偿进一步的减速而必须记录好的行为不予保证。一旦反压阻止输出任何 MCSPDU,且需要将它排入队列,应重新开始监测。当一个或多个 MCSPDU 排在队列中时,应在设定的时间间隔内记录实际的输出数。

每个 MCS 提供者应选择强制吞吐量间隔。该间隔应长到在最小吞吐量情况下允许至少一个具有最大长度的 MCSPDU 输出。MCS 提供者应通过向上传送 EDrq 通知它的上级有关所选择的间隔和任何后续的变化。提供者应在被监视的吞吐量不满足要求期间,在任何区间结束时阻止 MCS 连入或向下 MCS 连接。它应拆连该用户或拆断该连接。

上级提供者可以将它们的强制吞吐量间隔设置得比任何下属的强制吞吐量间隔加上一些响应时间还要长。其目的是鼓励最低提供者首先检测问题所在并进行强制作用。当在某个较短的间隔的结束后移出一违反者时,上级提供者应有足够的时间来检测吞吐量并将其恢复到适当水平。如果它们动作太快,它们可能处罚比应当处罚的子树更大的子树并打破了域中的良性循环。

预见在紧张的应用情况下,吞吐量可能偶然下降,设置域参数的控制器应用在它们的最低需求上应有所保留。如果它们的焦点是简单地防止接收者完全停止接收任何数据,它们可能将最小吞吐设置得非常低。

知道了最大 MCSPDU 长度和强制的吞吐量速率,控制器可以计算在检测到阻塞并克服它之前必须经过的最小间隔。

12.4 域配置

ITU-T 建议 T.122 不提供 MCS 提供者所支持的、用于配置域集的机制。一定要认为这是本地的事宜,其标准可能是将来研究的课题。该协议假定 MCS 提供者将认为一些域选择者有效而另一些无效。它规定了要交换的域选择符作为建立 MCS 连接通信的一部分。

MCS 提供者隐式地参与域参数的协商。无论是主叫方还是被叫方,按照为其设置的限制来限制它所允许的参数值范围。一旦任何一个用户已经连入到域或一旦已经建立了第一条 MCS 连接,MCS 提供者应冻结域参数的协商。

12.5 域合并

作为 MCS-CONNECT-PROVIDER 的结果,域被合并。如果此时将一个域或另一个域安排为空,那么合并时没有困难。然而,在大多数情况下,必须采取措施来更新保留的顶级提供者的信息库,以包含前顶级提供者的信息库并解决可能变为明显的任何冲突。在第 10 章中详细地论述了这些措施。

为帮助理解,按图 6 到图 9 的顺序示出了一个域合并的例子。此处,提供者 E 表示前顶,该前顶通过加粗线的 MCS 连接到一中间提供者 F 而加入一新域。是提供者 E 还是提供者 F 初始该 MCS 连接并不重要。

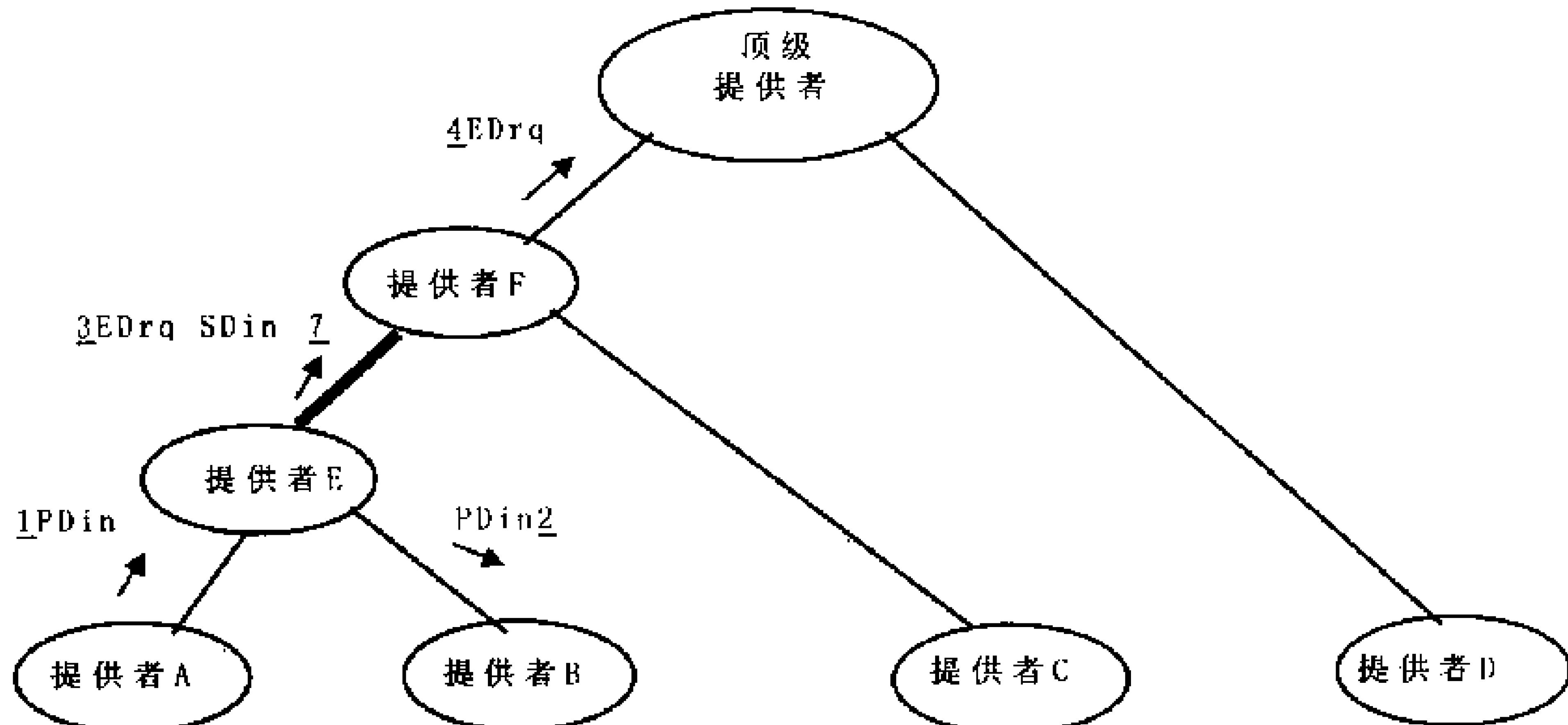


图 6 域合并第一步:建立分级结构

因发现自己处在向上连接的较低端,提供者 E 认为有责任执行该合并。由于在信息库处于变化之时进行其他活动是不安全的,所以提供者 E 停止接受来自其子树的输入。对于已经从提供者 A 和 B 生成的或在完成合并之前生成的新提供者所传出的任何 MCSPDU, 将予以保护, 并留作以后处理。然而, 不阻止 MCSPDU 的下行流,无论该流是由提供者 E 生成的还是由上面前转的。

由于不允许向上的用户请求,直到完成合并,提供者 E 将接收的证实 MCSPDU 只有 MCcf 和 MTcf, 这些将证实或去除已经使用的信道和令牌标识符。较高域的指示 MCSPDU PCin、PTin、DUin 和 CDin 也可能到达,从较低域中删除信道和令牌标识符,该较低域的合并已经被单独证实。因为较低域的未证实的标识符指一些不同于较高域的相同的标识符,因而,保护它们不被删除。必须遵守 PDin 以强迫域高度限制。正在进行合并时,提供者 E 不必应用保留指示。特别地,直到完成了合并,数据在先前单独的域之间的数据传送才需要指示;直到至少已经挑出运载它们的信道时才能传送它们。可以将新的标识符投入使用的两个指示使用起来不方便。为保持信息库的一致性,如果提供者 E 拒绝 CAin 或 TVin, 它将按第 10 章中的规定动作。

提供者 E 的第一个动作是向下发送 PDin 以确保最后的 MCS 连接没有创建这样的环,该环将使每一个域分级结构仅有一个顶级提供者的原则无效,并向上传送 EDrq,以报告它所在的高度和强制吞吐量间隔。已经从高度 2 上升到高度 3 的提供者 F 向顶级提供者传送 EDrq, 顶级提供者然后到达高度 4。

在域合并的第二阶段,提供者 E 向上发送 MCreq, 其数目与包含在它的信息库中的用户标识符数一样多。证实与上级域不发生冲突的用户标识符,同时以相同类的 MCcf 去除剩余的用户标识符。提供者 E 由 MCcf 生成 PCin, 以报告在它的子树上所进行的所有去除。该阶段在所有的用户标识符或者被明显地证实或者被明显地去除时结束。

阶段 3 与阶段 2 类似, 使用一平行的 MCSPDU 集, 但它关心的是令牌标识符而不是用户标识符信道。如果开始没有合并用户标识符,后面的 MTrq 部分可能作为无效而被拒绝, 同时没有必要去除受影响的令牌标识符。在一个被禁止的令牌情况下,禁止用户的整个集可能不适合单一的 MCSPDU。提供者 E

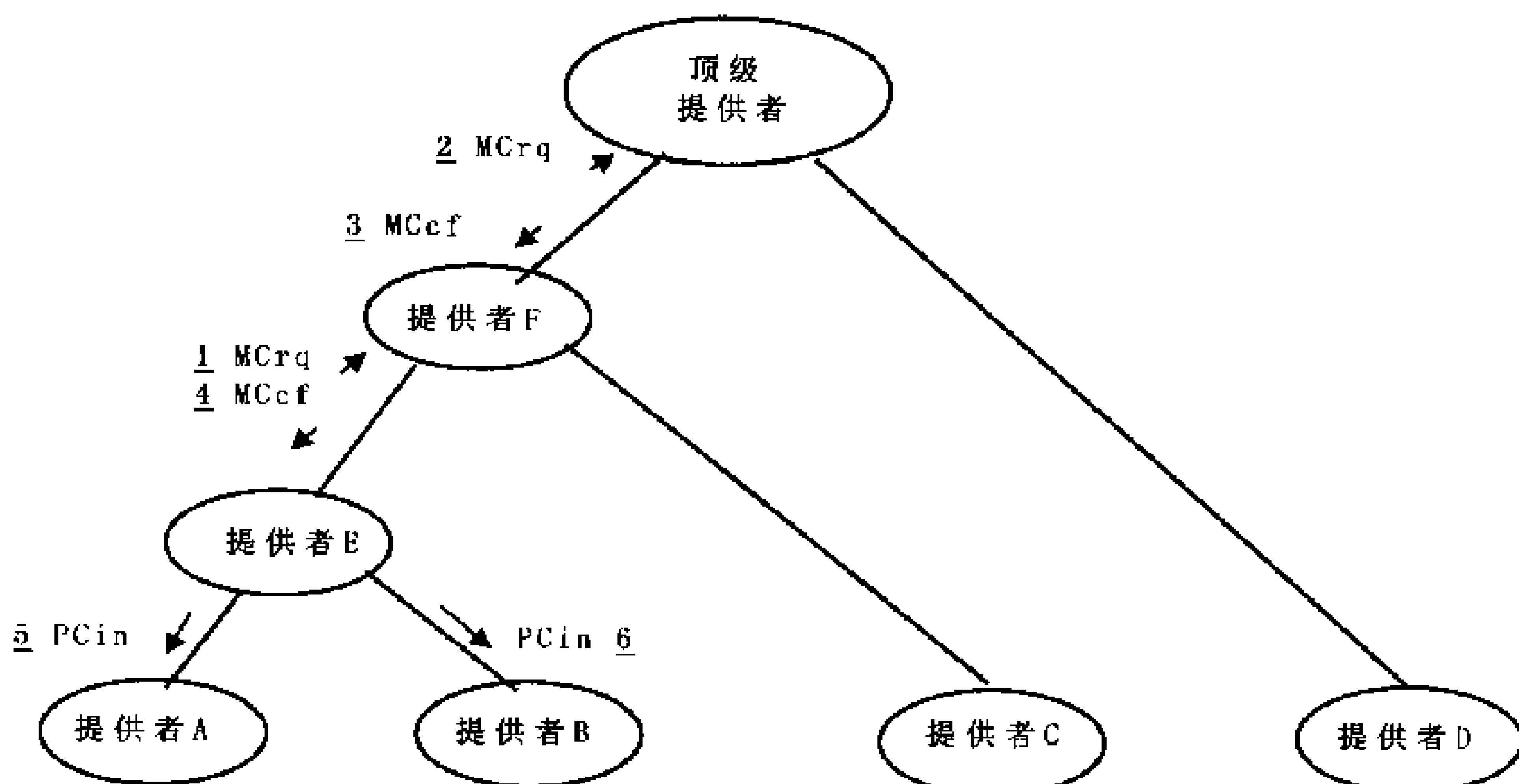


图 7 域合并第二步：合并用户标识符信道

在再发送带有其余用户标识符的被禁止的令牌之前，等待对它向上发送的第一个子集的证实。这样就防止了由于有太多的令牌标识符处于使用状态而拒绝第一集，而接受后面的余下部分，致使信息库产生混乱。该阶段在所有的令牌标识符或者已经被明显地证实或被明显地去除时结束。

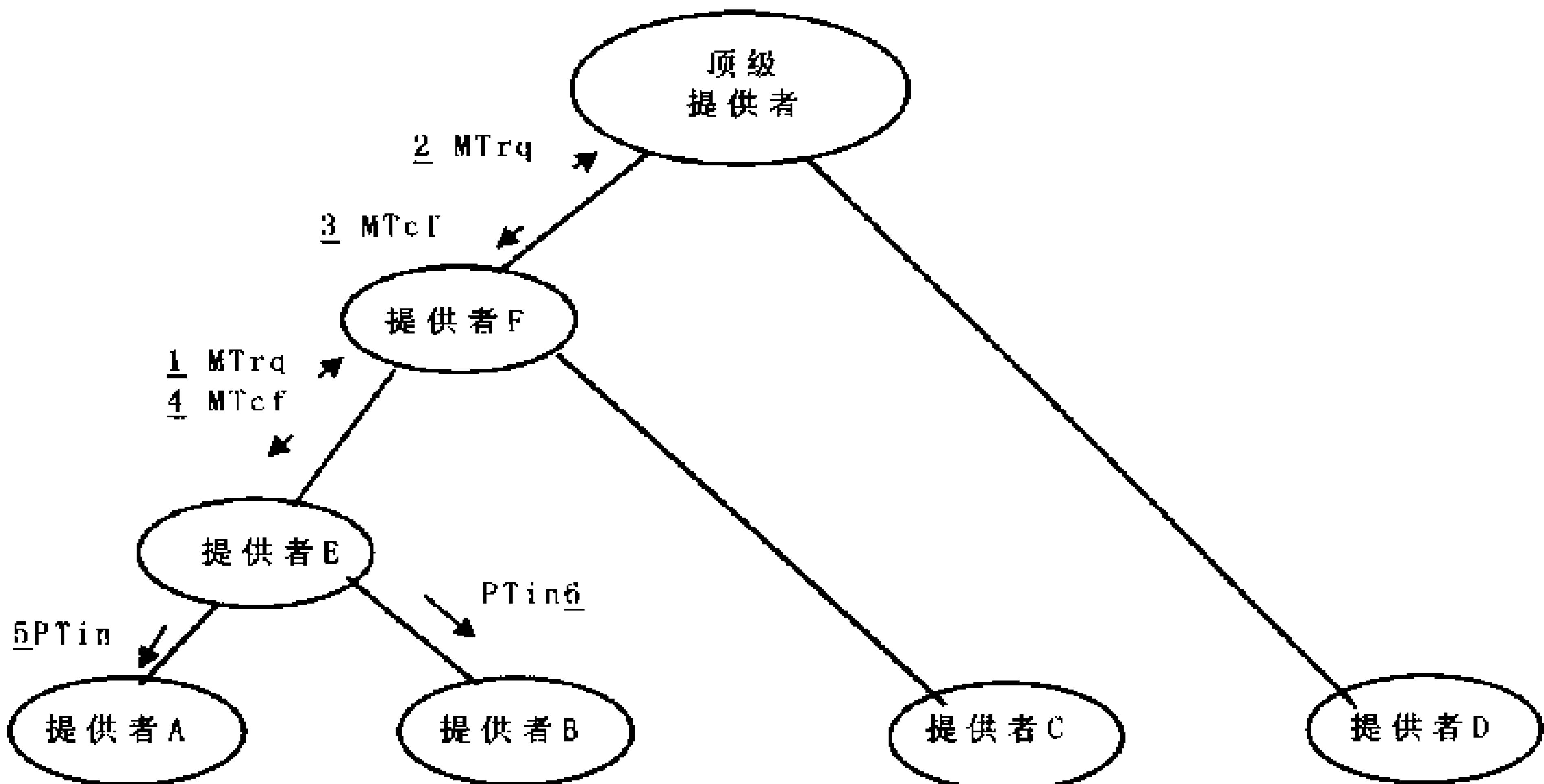


图 8 域合并第三步：合并令牌标识符

阶段 4 包含与阶段 2 相同的 MCS-PDU，但是它们包含不同的信道标识符。已经考虑了用户标识符信道，剩下的信道是静态的、专用的和分配的信道标识符。如果开始没有合并用户标识符，后面 MCrq 的部分可能作为无效而被拒绝，同时将不必去除受影响的信道标识符。在专用信道的情况下，信道管理者允许的整个用户集可能不适合单一的 MCS-PDU。提供者 E 在再发送带有其余用户标识符的专用信道之前等待对它向上发送的初始子集的证实。这样就防止了由于有太多的信道标识符处于使用状态而拒绝初始集，而接受后面余下部分，致使信息库的混乱。该阶段在所有的其余信道标识符或者已经被明显地证实或被明显地去除时结束同时完成域合并。

注：首先合并令牌标识符使它们专有的所有权在抑制数据流冲突上更有效。否则，由于证实了信道标识符，在通过令牌去除表现出该冲突之前数据可能会在域之间泄漏。

12.6 域拆断

当拆断向上的 MCS 连接时, MCS 提供者应通过拆连它的所有直接的 MCS 连入和拆断所有它的其他 MCS 连接而去除该域的子树。一般情况下,因为受影响的提供者没有为其向上发送的请求 MCSPDU 的记录,不可能接收到与请求 MCSPDU 相对应的证实 MCSPDU,所以它不能在其本身的子树中建立一残余的域。

当拆断了向下 MCS 连接时,MCS 提供者将为它的子树中余下的所有用户生成 MCSPDU DU_{rq},给出原因“域拆断”。

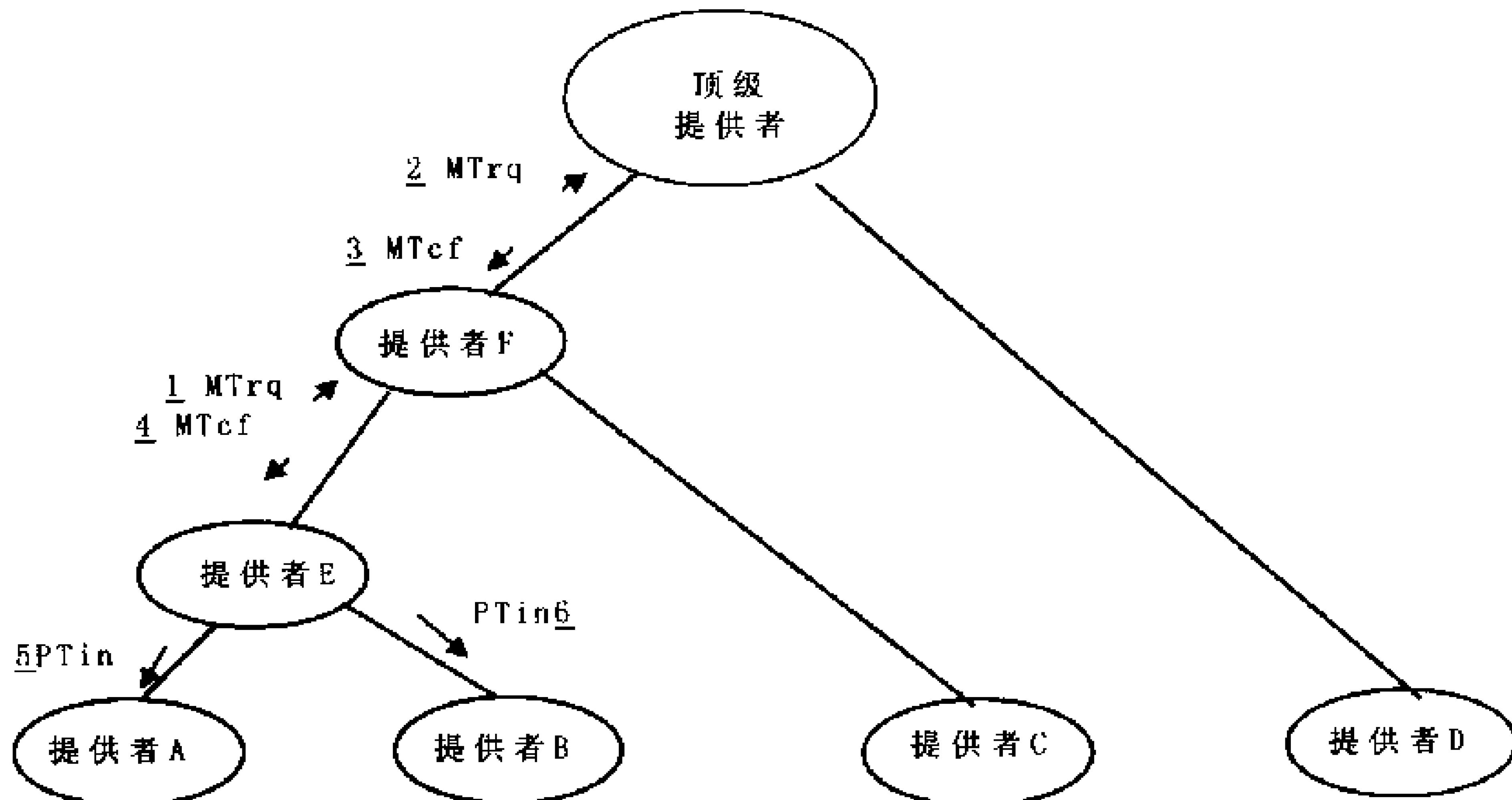


图 9 域合并第四步：合并余下的信道标识符

12.7 信道标识符分配

在处理原语请求 MCS-ATTACH-USER、MCS-CHANNEL-JOIN 零和 MCS-CHANNEL-CONVENE 期间,在顶级 MCS 提供者处动态地分配范围在 1001 以上的信道标识符。分配的值不需要适合任何特定的模式。再者希望值在允许的范围上随机地分配。这使得下面的情况更加可能出现,即对于独立运行了一段实际时间的两个域,过后可以被合并,而在它们各自的信道标识符分配上没有冲突,这也防止了当一个应用释放后,再分配给不同的应用时,在单一域中标识符循环得太快。使用 MCS 的应用在新的具体化返回之前应将时间调整到消失信道或用户标识符。

在需要活动域无缝隙合并的情况下,可以通过从子范围中选择对每一提供者唯一的信道标识符来避免冲突。通过动态地将信道标识符 1001...65535 分配为几个带的方式来创建子范围。再者,在子范围内,可以在一个方向上,从上或从下顺序地分配标识符。使用子范围分配的提供者仍应遵守该协议的所有方面,包括域合并的过程。然后它们应调节可能缺乏唯一子范围的对等提供者。

注:提供者如何从中选择分配的子范围是本地执行的事宜。网络管理系统可以划分预安排的会议。

不同于信道标识符,不分配令牌标识符,对它们值 1001 的划分线没有意义。具有给定值的自由令牌标识符在任何时刻都可被抓获或被禁止,且仅受一次使用的域总数的限制。

12.8 令牌状态

在第 7 章中正式地规定了令牌状态。它作为令牌证实 MCSPDU 的构成体返回并用于更新下级提供者的信息库中对令牌标识符的记录。不必直接地通过证实原语向初始用户报告令牌状态,但可以间接地通过结果值报告。

当不止一个令牌状态值描述给定用户与指定令牌标识符的关系时,优先级的次序应如下。应首先报告有自接收,如果它适合,它提醒用户它们必须响应 MCS-TOKEN-GIVE 指示。下一个优先者是自给出,以提醒用户它们被加入到未完成的操作,然后是自抓获或自禁止。在次序的最后是保留状态值,表示令牌的

当前状态是其他方独自使用的结果。

为在它们的信息库中正确地更新令牌状态,MCS 提供者取决于对令牌状态值的指定优先级。
