

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

H04L 12/46 (2006.01)

H04L 12/56 (2006.01)



[12] 发明专利说明书

专利号 ZL 200610061053.X

[45] 授权公告日 2009年3月4日

[11] 授权公告号 CN 100466588C

[22] 申请日 2006.6.5

[21] 申请号 200610061053.X

[73] 专利权人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

[72] 发明人 汪大海

[56] 参考文献

US2003/0007495A1 2003.1.9

US2006/0045102A1 2006.3.2

CN1553668A 2004.12.8

WO2006/048769A1 2006.5.11

CN1777328A 2006.5.24

审查员 杨颖

[74] 专利代理机构 深圳中一专利商标事务所

代理人 张全文

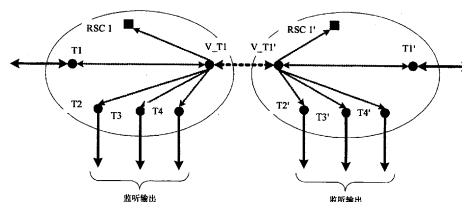
权利要求书 3 页 说明书 16 页 附图 4 页

[54] 发明名称

基于虚拟中继实现承载互通的方法及系统

[57] 摘要

本发明提供一种基于虚拟中继实现承载互通的方法，包括以下步骤：a. 逻辑功能元件和逻辑功能控制器配置虚拟中继数据；b. 所述逻辑功能元件进行拓扑分析；以及 c. 所述逻辑功能元件将拓扑分析得出的拓扑连接变化信息所涉及的虚拟中继端点 V_T1 与另一个虚拟中继端点 V_T2 或真实端点 T1 进行组合。本发明还提供一种基于虚拟中继实现承载互通的系统。本发明既能解决网关内互通的问题，又不降低网关的物理端口密度，在纯 TDM 组网应用下尤其有优势。另外，其实现灵活简单，无对具体电信业务的依赖性。



- 1、一种基于虚拟中继实现承载互通的方法，包括以下步骤：
 - a. 逻辑功能元件和逻辑功能控制器配置虚拟中继数据；
 - b. 所述逻辑功能元件进行拓扑分析；以及
 - c. 所述逻辑功能元件将拓扑分析得出的拓扑连接变化信息所涉及的虚拟中继端点V_T1与另一个虚拟中继端点V_T2或真实端点T1进行组合。
- 2、根据权利要求1所述的方法，其特征在于：所述步骤a中逻辑功能元件配置其两端的端点ID及所属虚拟逻辑功能元件ID；所述逻辑功能控制器为虚拟中继数据配置相应的呼叫路由、中继群及中继电路ID。
- 3、根据权利要求1所述的方法，其特征在于：所述步骤c中如果将所述虚拟中继端点V_T1与真实端点组合，则进一步包括以下步骤：
 - c1. 查找到所述虚拟中继端点V_T1和其另一侧虚拟中继端点V_T1'存储各自入末点和出末点信息的控制块；以及
 - c2. 根据所述拓扑连接变化信息对所述控制块进行处理。
- 4、根据权利要求3所述的方法，其特征在于：如果所述拓扑连接变化是形成所述虚拟中继端点V_T1与真实端点的隔离，则所述步骤c之后还包括以下步骤：
 - d1. 检查真实端点是否在所述虚拟中继端点V_T1的入末点和出末点信息列表中，如果是则进入步骤d2，如果不是则无须进一步处理；
 - d2. 检查真实端点是否在所述虚拟中继端点V_T1的入末点信息列表中，如果是则将真实端到另一侧虚拟中继端点V_T1'的所有出末点方向的连接断开，并将其从虚拟中继端点V_T1的入末点信息列表中清除；以及
 - d3. 检查真实端点是否在所述虚拟中继端点V_T1的出末点信息列表中，如果是则将另一侧虚拟中继端点V_T1'入末点到真实端点方向的连接断开，并将其从虚拟中继端点V_T1的出末点信息列表中清除。
- 5、根据权利要求3所述的方法，其特征在于：如果所述拓扑连接变化是形

成所述虚拟中继端点V_T1到真实端点的单向拓扑,则所述步骤c之后还包括以下步骤:

e1. 检查真实端点是否在所述虚拟中继端点V_T1的入末点信息列表中,如果是则将真实端到另一侧虚拟中继端点V_T1'的所有出末点方向的连接断开,并将其从虚拟中继端点V_T1的入末点信息列表中清除; 以及

e2. 检查真实端点是否在所述虚拟中继端点V_T1的出末点信息列表中,如果是则无须继续操作。

6、根据权利要求3所述的方法,其特征在于: 如果所述拓扑连接变化是形成所述虚拟中继端点V_T1到真实端点的双向拓扑,则所述步骤c之后还包括以下步骤:

f1. 检查真实端点是否在所述虚拟中继端点V_T1的入末点信息列表中,如果是则无须进一步处理,如果不是则将真实端点和另一侧虚拟中继端点V_T1'的所有出末点添加一个单向的连接; 以及

f2. 检查真实端点是否在所述虚拟中继端点V_T1的出末点信息列表中,如果是则无须进一步处理,如果不是则将另一侧虚拟中继端点V_T1'的入末点和真实端点添加一个单向的连接,并将真实端点添加到所述虚拟中继端点V_T1的出末点信息列表中。

7、根据权利要求1所述的方法,其特征在于: 所述步骤c中如果将所述虚拟中继端点V_T1与另一个虚拟中继端点V_T2组合,则进一步包括以下步骤:

c'1. 检查所述另一个虚拟中继端点V_T2是否在所述虚拟中继端点V_T1的入末点和出末点信息列表中,如果是则进入步骤c'2,如果不是则将其添加到所述虚拟中继端点V_T1的入末点和出末点信息列表中; 以及

c'2. 将所述虚拟中继端点V_T1对应的另一侧虚拟中继端点V_T1'的入末点和所述另一个虚拟中继端点V_T2对应的另一侧虚拟中继端点V_T2'的出末点添加一个单向的连接。

8、根据权利要求7所述的方法，其特征在于：所述虚拟中继端点V_T1'的入末点和虚拟中继端点V_T2'的出采点均为真实端点。

9、一种基于虚拟中继实现承载互通的系统，其特征在于：包括逻辑功能控制器和受所述逻辑功能控制器控制的逻辑功能元件，所述逻辑功能元件和逻辑功能控制器配置有虚拟中继数据，所述逻辑功能元件进行拓扑分析，并将拓扑分析得出的拓扑连接变化信息所涉及的虚拟中继端点V_T1与另一个虚拟中继端点V_T2或真实端点进行组合。

10、根据权利要求9所述的系统，其特征在于：所述逻辑功能控制器是媒体网关控制器，所述逻辑功能元件是媒体网关。

11、根据权利要求10所述的系统，其特征在于：当所述虚拟中继连接两个虚拟媒体网关时，所述虚拟媒体网关均位于同一物理媒体网关上。

基于虚拟中继实现承载互通的方法及系统

技术领域

本发明涉及通信领域，尤其是涉及一种基于虚拟中继实现承载互通的方法及系统。

背景技术

在分离架构下，呼叫控制面和承载控制面的主体分别是MGC（媒体网关控制器）和MGW（媒体网关）。MGW接受MGC的控制，接口协议称作媒体网关控制协议，在Itu-t定义为H.248系列规范。

MGW为呼叫/会话提供具体的承载资源。在H.248 协议模型中，Context（上下文）和Termination（端点）是最重要的两个概念。其中，Termination是对MGW上承载资源的抽象，由网关自行分配Id来标识，MGC通过这个Id来引用MGW上的承载资源并对其进行操作。一个Termination可以用来标识一个TDM时隙，可以用来标识AAL2的连接，也可以用来表示一个RTP流，如此等等。而Context则用来将多个Termination关联起来，以体现其对呼叫/会话的参与关系，通常对于普通呼叫/会话，一个Context中只包含两个Termination，而对于特殊业务，如多媒体呼叫、多方通话/会议和监听等，则可以允许有2个以上的Termination。具体请参见H.248协议。

MGW和MGW之间通过承载联通起来。一个物理的MGW可以虚拟成多个逻辑的MGW，这些逻辑的MGW被称为VMGW（Virtual MGW，虚拟媒体网关），每个VMGW对其MGC来说都作为一个单独的物理MGW来看待。MGC负责呼叫控制，可以管理多个MGW（对于每个这样的MGW，既可能是物理的，也可能是虚拟的），需要为呼叫在所管理的MGW之间路由并建立承载路径。一个呼叫/会话可能只在一个MGC控制的区域内就可以完成，也可能需要跨越多个MGC，此时MGC需要处理相互之间的路由和承载路径的建立问题。示意如图1。

在图1中，USER 1到USER 2之间的呼叫经过了两个MGC的控制域，分别为MGC 1和MGC 2，MGC为其建立的承载路径经过了MGW 1、MGW 2和MGW 3三个MGW，其中MGW 1本身就是一个物理MGW，而MGW 2和MGW 3则是同一个物理MGW上的两个VMGW。很明显，从物理MGW角度看，上述MGW 2和MGW 3涉及网关内承载互通的问题。

通常情况下，一个承载路径在一个MGW（从MGC角度看到的MGW，即要么是一个未划分VMGW的物理MGW，要么就是一个VMGW）内只涉及一个Context，比如对于普通呼叫这种做法是适用的。对应上图的承载路径，示意如图2。

通常采用一个Context并不表明就不可以采用多个Context。实际上会发现，将原来一个MGW对应单个Context的方式改变成一个MGW对应2个甚至多个Context是有好处的——特别是从MGC的角度。比如可能获得以下的好处：

1>MGC的处理模型可以统一到跨MGW呼叫模型中，即只涉及单个MGW的呼叫也可以当作跨两个MGW的呼叫来处理——本来MGC就要管理跨MGW的呼叫模型，而单Context方式下MGC还要多管理一种MGW内呼叫的模型；

2>MGC呼叫信令功能的Incoming-Half和Outgoing-Half可以获得一个更松耦合的设计——单Context形式下，这个Context是Incoming-Half和Outgoing-Half要共同维护的东西，这就是一种耦合性，而一旦Incoming-Half和Outgoing-Half可以各自拥有一个Context，则这种耦合性就消失了（注：对于 Incoming-Half和Outgoing-Half的概念，参考BICC协议可以获得良好的概念）；

3>对于某些特殊业务模型，比如MPTY、监听、移动放音时切换，多Context设计可以带来灵活性和方便性——以MPTY为例，如果多方通话采用单一Context设计，则会议桥上下文的放音和监听需要进行复杂的拓扑管理，实现代价非常大，而采用多Context设计之后，会议桥可以放在一个单独的上下文内，放音和监听在每个Call Leg上下文中进行，则这种复杂的拓扑管理就避免了（请参考3GPP TS23.205规范，对MPTY的单Context设计和多Context设计有良好清晰

的阐述)。很显然，一旦这么有好处的东西被付诸实施，则带来了单个MGW内（这个MGW可以是一个未划分VMGW的物理MGW，也可以是一个VMGW）承载互通的问题。

总结上面的描述，获得问题如下：

(1) 一个PMGW划分成多个VMGW，一个VMGW上Context存在与另一个VMGW上的Context进行承载互通的可能性。

(2) 一个未划分VMGW的PMGW内，其Context之间存在承载互通的可能性。

(3) 一个VMGW内的Context之间存在承载互通的可能性。

(4) 利用前面三种可能性，还可以组合出更复杂的情况，例如以三方通话为例，会议桥单独占用一个上下文，即拥有三个Termination的那个Context。

无论MGW怎样实现这些Context之间的承载互通，都要以一种对MGC透明的方式来处理，因为MGC仍然认为这是实实在在的承载，比如：两个VMGW虽然都在同一个PMGW上，但是对于MGC来说，它们是两个独立的MGW，MGC仍然会认为这两个MGW与MGW之间是通过真实的承载联通起来的。当然，以非透明的方式实现也是可能的，但肯定是一个私有接口，严重丧失了开放性，导致一个厂商的MGW不能和其它厂商的MGC互通。

目前主流的承载有三种类型，即TDM、IP和ATM，通过真实的承载实现上述互通也是有可能的。比如，3GPP对WCDMA应用就正在探讨一种通过真实IP承载互通（使用标准的Nb接口和IPBCP协议，Nb接口定义请参见3GPP TS29.415协议，IPBCP协议请参见ITU-T Q.1970）的可能性。另外，考虑通过真实TDM中继在MGW外部环回的方式也可以实现上述互通，此时就是以真实的TDM承载的方式。采用真实承载进行互通的最大缺点是占用实际的物理承载资源，导致MGW设备的端口密度大大降低，极端的情况下可能降低一半左右——比如MGC以每个普通呼叫在一个MGW上占用两个Context的方式实现时。这对设备的规格损失是不可接受的。另一方面，组网复杂，比如外部环回TDM中继方式仍然要

配置相应的中继电缆，导致太多的多余走线。

发明内容

本发明要解决的技术问题是克服上述现有技术存在的不足，提供一种针对同一媒体网关内的Context间实现承载互通的基于虚拟中继实现承载互通的方法及系统。

为解决上述技术问题，本发明提供一种基于虚拟中继实现承载互通的方法，包括以下步骤：

- a. 逻辑功能元件和逻辑功能控制器配置虚拟中继数据；
- b. 所述逻辑功能元件进行拓扑分析；以及
- c. 所述逻辑功能元件将拓扑分析得出的拓扑连接变化信息所涉及的虚拟中继端点V_T1与另一个虚拟中继端点V_T2或真实端点T1进行组合。

进一步地，所述步骤a中逻辑功能元件配置其两端的端点ID及所属虚拟逻辑功能元件ID；所述逻辑功能控制器为虚拟中继数据配置相应的呼叫路由、中继群及中继电路ID。

进一步地，所述步骤c中如果将所述虚拟中继端点V_T1与真实端点组合，则进一步包括以下步骤：

- c1. 查找到所述虚拟中继端点V_T1和其另一侧虚拟中继端点V_T1'存储各自入末点和出末点信息的控制块；以及
- c2. 根据所述拓扑连接变化信息对所述控制块进行处理。

进一步地，如果所述拓扑连接变化是形成所述虚拟中继端点V_T1与真实端点的隔离，则所述步骤c之后还包括以下步骤：

d1. 检查真实端点是否在所述虚拟中继端点V_T1的入末点和出末点信息列表中，如果是则进入步骤d2，如果不是则无须进一步处理；

d2. 检查真实端点是否在所述虚拟中继端点V_T1的入末点信息列表中，如果是则将真实端点到另一侧虚拟中继端点V_T1'的所有出末点方向的连接断开，并将其从虚拟中继端点V_T1的入末点信息列表中清除；以及

d3. 检查真实端点是否在所述虚拟中继端点V_T1的出末点信息列表中, 如果是则将另一侧虚拟中继端点V_T1'入末点到真实端点方向的连接断开, 并将其从虚拟中继端点V_T1的出末点信息列表中清除。

进一步地, 如果所述拓扑连接变化是形成所述虚拟中继端点V_T1到真实端点的单向拓扑, 则所述步骤c之后还包括以下步骤:

e1. 检查真实端点是否在所述虚拟中继端点V_T1的入末点信息列表中, 如果是则将真实端点到另一侧虚拟中继端点V_T1'的所有出末点方向的连接断开, 并将其从虚拟中继端点V_T1的入末点信息列表中清除; 以及

e2. 检查真实端点是否在所述虚拟中继端点V_T1的出末点信息列表中, 如果是则无须继续操作。

进一步地, 如果所述拓扑连接变化是形成所述虚拟中继端点V_T1到真实端点的双向拓扑, 则所述步骤c之后还包括以下步骤:

f1. 检查真实端点是否在所述虚拟中继端点V_T1的入末点信息列表中, 如果是则无须进一步处理, 如果不是则将真实端点和另一侧虚拟中继端点V_T1'的所有出末点添加一个单向的连接; 以及

f2. 检查真实端点是否在所述虚拟中继端点V_T1的出末点信息列表中, 如果是则无须进一步处理, 如果不是则将另一侧虚拟中继端点V_T1'的入末点和真实端点添加一个单向的连接, 并将真实端点添加到所述虚拟中继端点V_T1的出末点信息列表中。

进一步地, 所述步骤c中如果将所述虚拟中继端点V_T1与另一个虚拟中继端点V_T2组合, 则进一步包括以下步骤:

c'1. 检查所述另一个虚拟中继端点V_T2是否在所述虚拟中继端点V_T1的入末点和出末点信息列表中, 如果是则进入步骤c'2, 如果不是则将其添加到所述虚拟中继端点V_T1的入末点和出末点信息列表中; 以及

c'2. 将所述虚拟中继端点V_T1对应的另一侧虚拟中继端点V_T1'的入末点

和所述另一个虚拟中继端点V_T2对应的另一侧虚拟中继端点V_T2'的出末点添加一个单向的连接。

进一步地,所述虚拟中继端点V_T1'的入末点和虚拟中继端点V_T2'的出末点均为真实端点。

本发明还提供一种基于虚拟中继实现承载互通的系统,包括逻辑功能控制器和受所述逻辑功能控制器控制的逻辑功能元件,所述逻辑功能元件和逻辑功能控制器配置有虚拟中继数据,所述逻辑功能元件进行拓扑分析,并将拓扑分析得出的拓扑连接变化信息所涉及的虚拟中继端点V_T1与另一个虚拟中继端点V_T2或真实端点T1进行组合。

进一步地,所述逻辑功能控制器是媒体网关控制器,所述逻辑功能元件是媒体网关。

进一步地,当所述虚拟中继连接两个虚拟媒体网关时,所述虚拟媒体网关均位于同一物理媒体网关上。

本发明具有以下有益的效果:本发明为背景中总结的问题提供了一种清晰的解决思路和可实际实施的完整方法。该方法最大优点在于既能解决网关内互通的问题,又不降低网关的物理端口密度,在纯TDM组网应用下尤其有优势。另外,其实现灵活简单,无对具体电信业务的依赖性。

附图说明

图1是MGW和MGW之间通过承载联通的示意图。

图2是对应图1的承载路径示意图。

图3是虚拟中继示意图。

图4是本发明第一实施例的示意图。

图5是基于图4的物理拓扑连接示意图。

图6是本发明第二实施例的示意图。

图7是基于图6的物理拓扑连接示意图。

具体实施方式

以下结合附图对本发明进行详细描述。

在分离架构下，一个灵活的可支持综合业务并具有长期演进能力的MGW可以是下面的逻辑结构：

OAM Units: 提供整个系统的操作维护功能，如安全管理、配置管理、日志管理、告警管理、性能管理、跟踪管理等等，同时提供对外部运营支撑系统的接口。

Narrow Band Units: 提供设备对窄带侧的接口和协议处理功能，基于TDM技术，提供对TDM承载的低层次抽象。通常，TDM交换所需要的物理资源（如交换网）在此处提供。

Broad Band Units: 提供到宽带侧的接口和协议处理功能，可以基于ATM技术，也可以基于IP技术，也可以同时基于ATM和IP技术，提供对IP/ATM等承载的低层次抽象。

Codec Pools: 提供媒体处理资源，如码转换TC、声学回声抑制EC、电学回声抑制EC、语音质量增强VQE、电路交换数据互通 IWF、多媒体互通MUX、会议通道，等等。

Control Units: 提供到MGC的垂直接口，处理并根据来自MGC的控制指令，驱动内部NarrowBand Units、Broad Band Units、Codec Pool等组件工作。通常在此处实现对Context、Termination、拓扑等高层次资源的抽象和管理。

上述结构是个逻辑上的表示，具体到物理实体（如单板、硬件、软件）的划分和对应可以根据自己的需要选择合适的方案实现，不影响本发明的技术实施。

本发明基于一种虚拟TDM中继的思路，即TDM承载是虚拟的，非真实存在的资源。

从MGC角度来看，这种虚拟中继和真实的TDM中继完全一样，管理和操作方式完全一样。在呼叫控制层面，MGC可以将这种中继用于各种共路信令或者

承载无关的信令中，如SS7、Q.931，等等。在配置管理层面，其在路由、中继群、中继电路等数据的管理和组织上也是和真实的TDM中继一样的。在MGW控制层面，MGC可以使用真实TDM承载的操作方式对其进行操作。总之，这种虚拟中继对MGC完全透明，所有与真实TDM承载的差异都在MGW内部进行屏蔽。

一个真实的中继两端各连接着一个设备，在本文就是MGW，这两个MGW可以是同一个VMGW或PMGW，也可以是不同的VMGW或PMGW。这个中继各占用两个Termination Id，分别体现其与两端设备的连接关系。虚拟中继也类似，只不过当其连接两个不同的VMGW时，这两个VMGW必须在同一个PMGW上（这是很显然的，因为两个物理的设备不可能通过完全虚拟的非真实存在的东西连接起来），也正是如此，从一个MGW连接到另一个MGW，当这两个MGW处于不同的PMGW上时，其间必然是真实承载。此处要强调一下，每个虚拟中继也有两个Termination Id，设分别为T1和T2，则如果一个承载是由虚拟中继提供的，则承载一端肯定对应着Termination T1，另一端对应着Termination T2。示意如图3。

从MGW角度看，虚拟承载相对于真实承载来说，其区别主要是在拓扑管理上。在MGW内，一个真实的承载资源可以很好的和另外的真实资源（可以是其它的承载资源，也可以是一些承载/媒体互通资源，如TC等）实现交换连接，而虚拟承载则缺乏这样的拓扑连接点。虚拟承载必须通过某种途径将其与MGW内的某个拓扑连接点挂接起来，才能实现与其它承载/资源的拓扑操作——这个“其它承载”既可以是真实的承载，也可以是另外的虚拟承载，当是另外的虚拟承载时，这个“另外的虚拟承载”也必须经过上述的挂接处理，最终体现为真实的承载/资源挂接。

一个承载实际上体现为Context对外的一个连接，或者说媒体流。在H.248框架下，一个媒体流是与Termination关联起来的，通常一个Termination只有一个媒体流，比如对于TDM承载来说，因为其物理特性的限制就注定只有一个媒体流

(注：实际上，一个TDM承载上也可以承载多个流，不过这多个流必然是经过复用等处理的，本文为了描述方便，将这样经过处理的流仍然看作单个流看待，不影响本发明的技术实施)。而对于其它承载类型，如IP/ATM，一个Termination则可以很方便地提供多个逻辑上独立的流，不过目前直接以这种多流方式实现的应用并不多(即使是VIG这样的多媒体应用，其也是通过多个Termination实现多个流，而不是单个Termination承载多个流)，而且流只不过是Termination的一个更细粒度的逻辑划分而已，如果将Termination的定义下移一级，则Termination就和流完全等同了。所以，再一次，本发明为了描述方便，直接将Termination与单个承载流等同看待，不影响本发明的技术实施，而且，将这种等同看待的实现扩展到多流Termination的情形，在技术上是显而易见的——只需要Termination与流相关的内容抽取出来，并且下移一层就可以构造出流所需的信息，然后将上一级的Termination与下一级的多个流予以关联即可。

从拓扑的角度，一个Termination，对于其上对外的承载流来说，只可能源自Context内的某单个点，即在Context内只可以有一个物理的入向流——如果有多个流流入，则这多个流必然要经过混音、MUX等资源的汇聚处理，而汇聚处理之后的输出再作为Termination的输入，此时从Termination角度看仍然是单一的入向流。这个入向流可以直接来自同一Context内部的其它Termination，也可以来自Context内的其它资源，如编解码器、放音资源、混音资源、MUX资源，等等。这是入向流的情况，对于出向流则可以存在多个，即Termination收自外部(相对于Context边界)的承载流可以送往Context内的多个宿点，比如，一个复制到Context内的DTMF检测器，一个复制到Context内的对端Termination，其它N个流以复制的方式输送到多个监听用的Termination。

根据上面的分析，可以看出一个Termination可以很容易抽象出以下特征：至多一个入向连接，至多N(这个N可根据各个系统的需求自行确定，不影响本发明的技术实施)个出向连接。对应每个连接是一个EndPoint(末点)，这些EndPoint可以是其它Termination，也可以是Context中的资源与此Termination连接

的点。这些EndPoint具体的物理实现可以是多种多样的，如：在一个基于GE通信的组件/平面中它可以体现为MAC地址加上一定的Session标识（用来区分到/源自同一个MAC的多个流），而在一个基于TDM通信的组件/平面中它可以体现为TDM交换网边缘的时隙号，等等，具体是怎样的实现可依赖于系统本身的需要进行选择，不影响本发明的技术实施。这些连接在Context的LifeCycle中是动态变化的，对应的EndPoint与此Termination的连接绑定关系也是动态变化的，很明显需要作为Termination的关联动态数据进行管理。

同样，对于虚拟中继也具有这种特征，只不过本发明的虚拟中继是虚拟的TDM中继，所以要求这些连接的EndPoint对其表现出TDM特性，即是TDM时隙，这并不难做到，并且和真实TDM承载的要求和做法是一致的：在实际的系统中，如果这个EndPoint本身就是TDM承载或者资源，则自然满足了要求，否则插入相应的资源进行转换即可，比如对于IP/ATM承载，在其和虚拟中继之间插入TC即可实现TDM特性的体现——此时虚拟中继看到的EndPoint是TC的TDM时隙。而且，前面说过，虚拟承载本身不具有挂接点，则上述特征中多个连接事实上也构成了虚拟承载与其它真实承载/资源挂接的唯一机会——对于这样的连接，只要其EndPoint不是另外的虚拟中继端点，则这个EndPoint就构成了这个虚拟中继端点的挂接点之一。

上述抽象实现了虚拟中继与真实承载/资源的挂接，但这并不能使虚拟中继构成真正的物理上的拓扑连接。如果要构成物理上的拓扑连接，则需要将一个虚拟中继两个Termination所在的Context关联起来考虑，示意如图4。期望形成的物理拓扑连接将如图5所示。

拓扑连接与虚拟中继端点的连接端点的关系示意如下表：

V_T1 :

In_EndPoint	Out_EndPoints
T1	T1
	T2
	T3
	T4
	RSC 1

V_T1' :

In_EndPoint	Out_EndPoints
T1'	T1'
	T2'
	T3'
	T4'
	RSC 1'

拓扑连接表:

Source	Destination
T1	T1'
	T2'
	T3'
	T4'
	RSC 1'

Source	Destination
T1'	T1
	T2
	T3
	T4
	RSC 1

从上图可以看出，最终的物理拓扑的形式实际上就是把V_T1的In_EndPoint（入末点）与V_T1'的Out_EndPoints（出末点）单向连接起来，再把V_T1'的In_EndPoint与V_T1的Out_EndPoints单向连接起来。到了这儿，实现思路已经很明确了，现在主要的问题有两个。

一个问题是如何得出每个虚拟中继端点的In_EndPoint和Out_EndPoints信息？实际上，这是H.248协议的基本功能，即通过拓扑描述，指明Context中的其它承载端点与虚拟中继端点之间是怎样的拓扑结构，是隔离、单向还是双向（缺省是双向）：当一个承载端点与虚拟中继端点是隔离的时候，它对虚拟中继端点的In_EndPoint和Out_EndPoints都没有贡献；当一个承载端点存在到虚拟中继端点的单向拓扑时，该承载端点就形成了虚拟中继端点的In_EndPoint；当一个承载端点存在源自虚拟中继端点的单向拓扑时，该承载端点就形成了虚拟中继端点的Out_EndPoints之一；当一个承载端点存在与虚拟中继端点的双向拓扑时，该承载端点既是虚拟中继端点的In_EndPoint，又是其Out_EndPoints之一。另外，还有放音、收号、MUX、混音等资源也可以形成虚拟中继端点的In_EndPoint和Out_EndPoints，其判定准则和承载端点类似的，只不过不是明确的拓扑描述的

方式，比如，H.248可以指示对虚拟中继放音，则MGW分配的放音资源就会形成到虚拟中继端点的单向拓扑，继而成为虚拟中继端点的In_EndPoint。要注意的一种情形是，多个资源/承载点到当前考察端点的拓扑可能存在冲突（冲突主要存在于到被考察端点的入向拓扑上，因为这个方向上对入向流的数目明确限制为1），比如一个放音资源对V_T1放音，同时又指示另一个承载端点存在到V_T1的单向或者双向拓扑结构，则就导致V_T1上有两个入向流（一个来自放音资源，另一个来自承载端点），这种冲突的通常解决机制是取就近优先（关联关系密切、操作时间靠后等）原则，在上例中，放音资源形成的入向连接被认为是优先的，所以承载端点形成的入向连接被暂时忽略，直到放音结束之后才恢复有效。

第二个问题是如何将两个端点（如前述的V_T1和V_T1'）的In_EndPoint/Out_EndPoints信息关联起来处理？实际上，因为这两个端点都是针对同一个虚拟中继的，虚拟中继在MGW配置了，这两个端点的关联关系就产生了。甚至，最简单的实现，是将这两个端点的这类EndPoint信息共享控制块，而其它的信息单独放在端点本身的控制块，端点控制块有到EndPoint控制块的索引。示意如下：

下面再总结一下处理流程，如下（沿用前面的MGW逻辑结构中的模块划分）：

首先：MGW上必须配置虚拟中继数据，如两端的Termination Id，所属VMGW的Id，等等；MGC上为虚拟中继数据配置相应的呼叫路由、中继群、中继电路Id等，用于VMGW内或者统一PMGW内VMGW间的互通。当有呼叫涉及这种互通时，MGC能够通过这些配置数据控制MGW在相应的Context中添加相应的端点。

对于每个Context，MGW会接受来自MGC的操作请求（如添加/删除端点、改变拓扑连接关系、指示在端点放音/停止放音、指示在端点上进行/停止DTMF检测，等等），同时MGW内部也可能产生一些激励事件（如放音结束，等等），

对于这些内容，Control Units会进行相应的分析处理，其中很重要的一项（也是本发明最相关的一项）就是拓扑分析和处理。在拓扑分析中，会得出以下变化信息（注意只关注变化信息即可）：哪些拓扑连接是需要删除的，哪些拓扑连接是需要新增的。

无论是需要删除的还是需要新增的，如果这个拓扑连接关连的端点不涉及虚拟中继，则其处理和不实现本发明的MGW实现一致，MGW可以选择自己所方便的方式进行实现，而不影响本发明的技术实施。如果不是这样，则其中一侧必然是虚拟中继端点，假设为V_T1（设该中继的另一侧Termination Id为V_T1'），另一侧可能是另一个虚拟中继端点V_T2，也可能是一个真实的承载或者资源EndPoint，设为T1。先看后一种情况，即V_T1和T1的组合，下面看如何处理：

根据前面的描述，可以很容易找到V_T1和V_T1'存储In_EndPoint/Out_EndPoints信息的控制块。

如果这个拓扑变化的目标是形成V_T1和T1隔离，则检查T1是否在In_EndPoint/Out_EndPoints列表中出现（实际上，只需要很小的编程技巧就可以实现直接的判断而不需遍历查找，不影响本发明的技术实施，此处不赘述），如果不出现，则说明原来就是隔离状态，无须进一步处理。接下来检查T1是否出现在V_T1的In_EndPoint中，如果是，则将T1到V_T1'的所有Out_EndPoints方向的连接断开，并且将其从V_T1的In_EndPoint中清除。再检查（此处的“再”和上面是顺序执行的关系，因为T1可能同时出现在V_T1的In_EndPoint和Out_EndPoints中）T1是否出现在V_T1的Out_EndPoints当中，如果是，则将V_T1'的In_EndPoint到T1方向的连接断开——如果先前存在这样的连接的话，接着将其从V_T1的Out_EndPoints中清除。

如果这个拓扑变化的目标是形成V_T1到T1的单向拓扑，则检查T1是否在V_T1的In_EndPoint中出现，如果是，则将T1到V_T1'所有Out_EndPoints方向的连接断开，并将T1从V_T1的In_EndPoint中清除。接下来，检查T1是否已经在

V_T1的Out_EndPoints当中，如果已在，则无须继续操作，否则，将V_T1'的In_EndPoint和T1做一个单向的连接处理，并将T1添加到V_T1的Out_EndPoints当中。

如果这个拓扑变化的目标是形成V_T1到T1的双向拓扑，则先检查T1是否在V_T1的In_EndPoint中出现，如果已经出现，则无须附加处理，否则将T1和V_T1'所有Out_EndPoints添加一个单向的连接处理。再检查T1是否在V_T1的Out_EndPoints中出现，如果已经出现，则无须附加处理，否则将V_T1'的In_EndPoint和T1添加一个单向的连接，并将T1添加到V_T1的Out_EndPoints当中。

注：上面对In_EndPoint和Out_EndPoint的检查顺序并不是强制性要求（但可能是一个比较简洁的实现方式），可以选择自己方便的方式加以实现，不影响本发明的技术实施。

实际上到这儿，背景技术中介绍的四个问题问题的前三个就已经解决了，而最后一个问题是前三个问题的组合情况，导致的结果就是出现本处将要介绍的V_T1和V_T2组合的情况（实际上，这最后一个问题通过管理的手段是可以限制使其不出现的，如：如果对一个普通呼叫MGC要求在一个VMGW上占用两个Context，则禁止PMGW到VMGW的逻辑划分就可以做到这一点），即本发明的第二实施例。

请参照图6所示，这种组合情况实际上是一个虚拟中继串接的过程，这个串接过程相当于将一个虚拟中继在原来的基础上叠加了一个延伸处理，延伸之后的两个虚拟中继在效果上等同于一个虚拟中继。拓扑连接与虚拟中继端点的连接端点的关系示意如下表：

In_EndPoint	Out_EndPoints
V_T2	RSC 2
	V_T2

In_EndPoint	Out_EndPoints
T7	T7
	T6

In_EndPoint	Out_EndPoints
V_T1	V_T1

In_EndPoint	Out_EndPoints
RSC 1	T1

据此，上图所应形成的物理拓扑应该如图7所示。拓扑连接表如下：

拓扑连接表：

Source	Destination
RSC 1	T6
	T7

Source	Destination
T7	RSC 2
	T1

在图6中实际已经给出了实现的提示，即一个虚拟中继端点的In_EndPoint/Out_EndPoints域可以设置成另一个虚拟中继端点，而实际物理拓扑操作时，则是一个将其递归向下展开的过程。举例说明如下：

4.1>参照图6-7，取一小部分来举例：在Context C3中T7有到V_T1'方向的拓扑，而V_T1'位于虚拟中继另一端的对应端点为V_T1，V_T1的Out_EndPoints域包括RSC 2和V_T2。

4.2>根据3>以前的实现，将T7连向RSC 2的处理是很显然的，现在就是轮到T7连到V_T2，而V_T2又是一个虚拟中继端点，其不可能直接联网，则将其递归展开：V_T2位于虚拟中继另一端的对应端点为V_T2'，V_T2'的Out_EndPoints域为T1，将T7连向T1即可，整个过程结束。

很显然，配置数据是在MGW的OMA Units完成的，而上述的整个分析和处理是在Control Units完成的，而具体的物理拓扑的形成则涉及网关内的所有模块，对于虚拟中继来说，TDM总线起着在这些模块间衔接的作用。这是和未采用本发明方案的MGW实现是完全相同的，厂商可根据自己的需要自行选择合适的方式加以实现，不影响本发明的技术实施。

另外，很显然，这个发明的应用不限于网关类设备，只要是采用TDM电路

提供承载连接并且涉及内部互通的设备都可以使用。比如，一个传统的交换机，可以看作是两个逻辑功能组件：MGC和MGW，通过内部接口绑定在一起，此时这种内部互通的方法很明显也是适用的。

本发明既能解决网关内互通的问题，又不降低网关的物理端口密度，在纯TDM组网应用下尤其有优势。另外，其实现灵活简单，无对具体电信业务的依赖性。

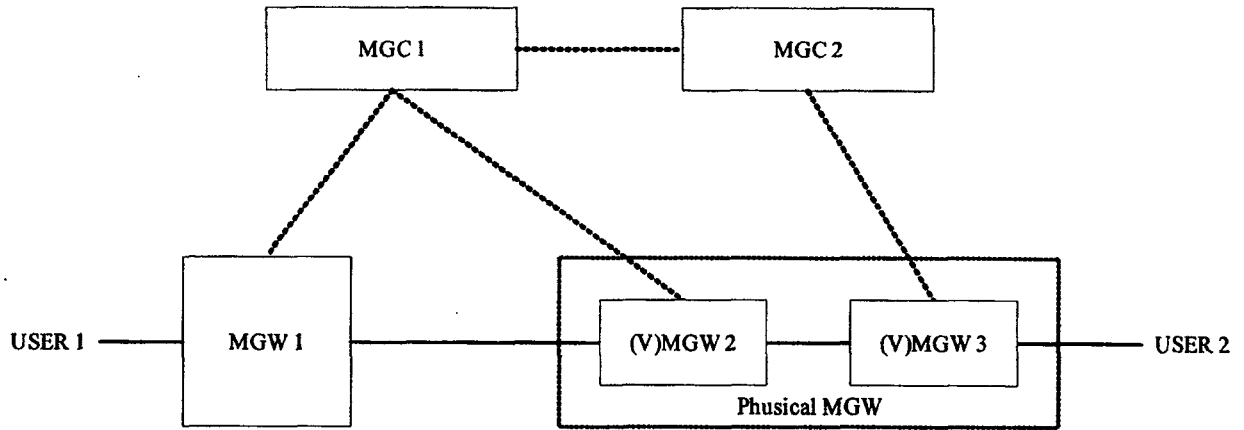


图 1

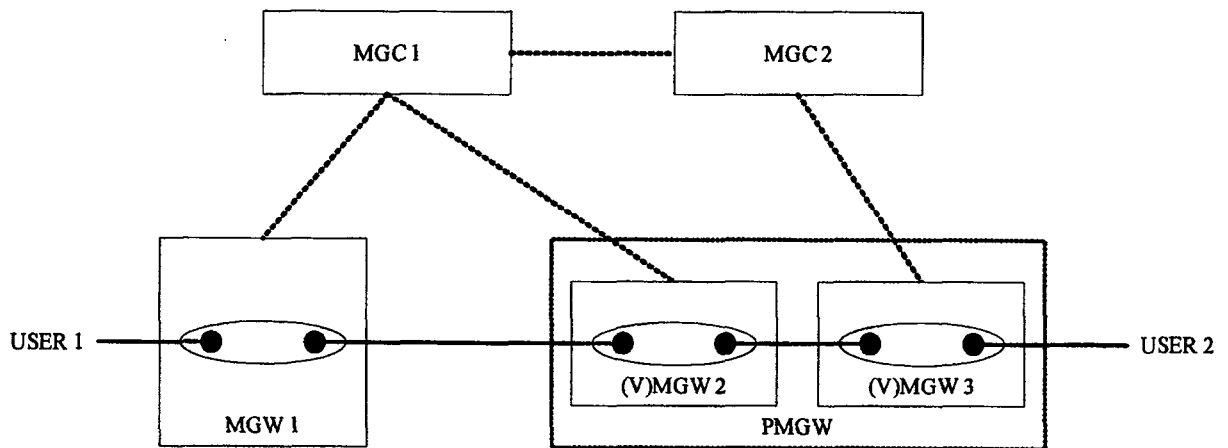


图 2

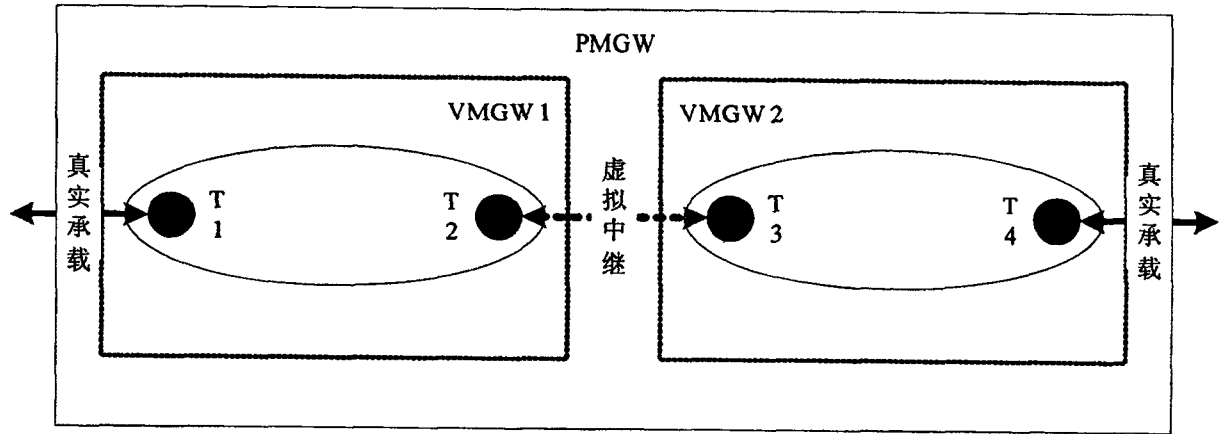


图 3

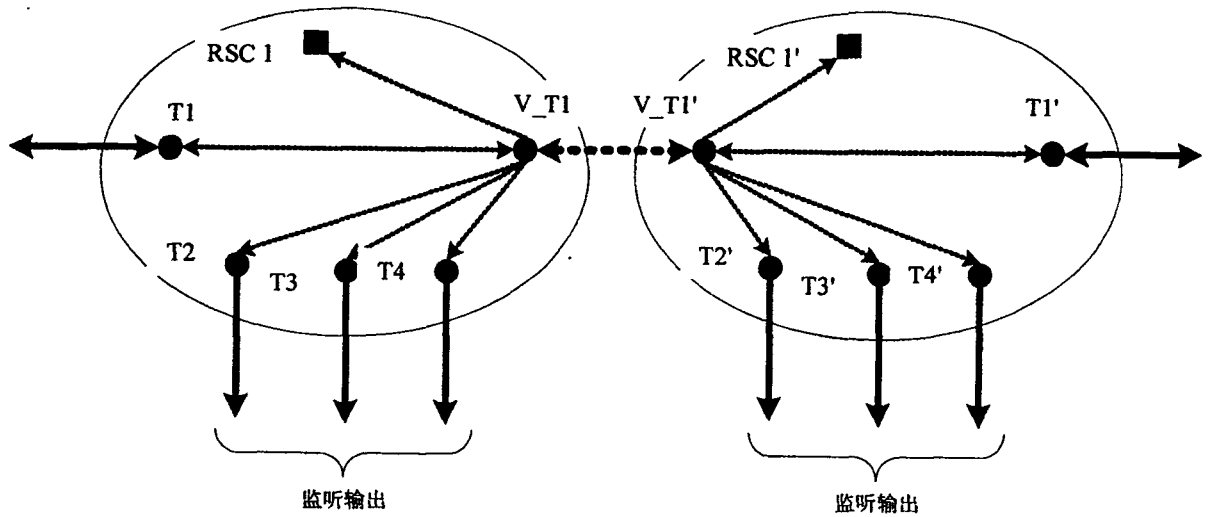


图 4

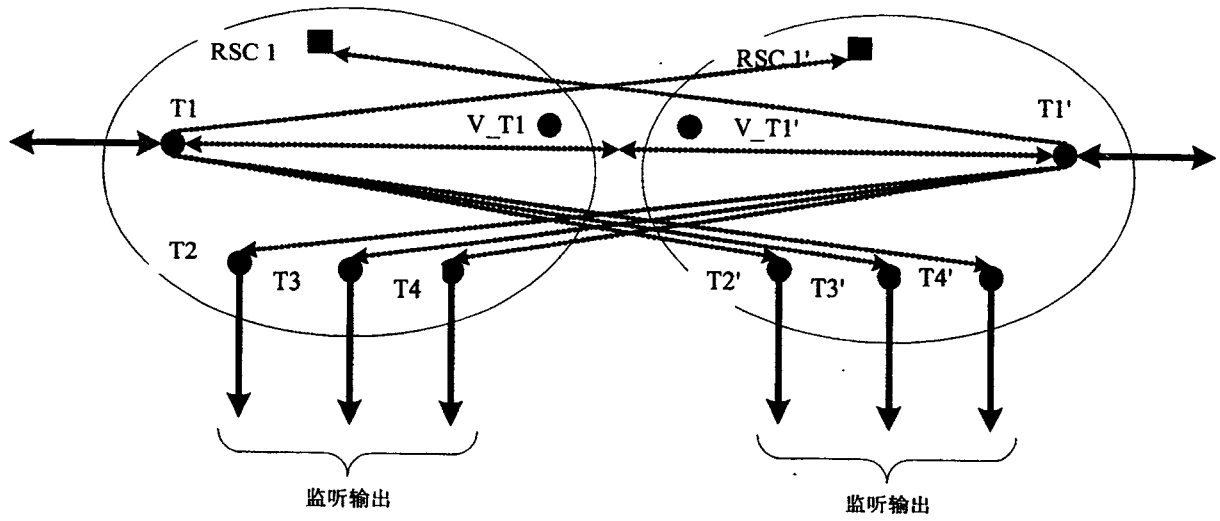


图 5

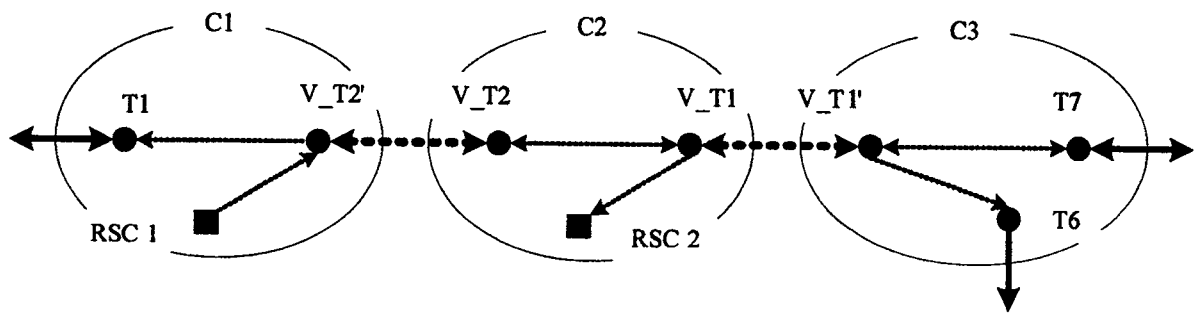


图 6

