



(12) 发明专利

(10) 授权公告号 CN 1663193 B

(45) 授权公告日 2010.05.12

(21) 申请号 02810757.8

H04L 12/28(2006.01)

(22) 申请日 2002.05.24

H04L 12/64(2006.01)

(30) 优先权数据

01113121.6 2001.05.29 EP

01114652.9 2001.06.19 EP

(85) PCT申请进入国家阶段日

2003.11.27

(86) PCT申请的申请数据

PCT/EP2002/005715 2002.05.24

(87) PCT申请的公布数据

W002/098070 EN 2002.12.05

(73) 专利权人 汤姆森许可贸易公司

地址 法国布洛里

(72) 发明人 让-巴普蒂斯特·亨利

塞巴斯蒂安·佩洛特

(74) 专利代理机构 中科专利商标代理有限责任

公司 11021

代理人 戎志敏

(51) Int. Cl.

H04L 12/56(2006.01)

H04L 12/46(2006.01)

H04L 12/403(2006.01)

(56) 对比文件

EP 1113625 A, 2001.07.04, 全文.

US 6389496 B1, 2002.05.14, 第8栏第55行—第10栏第50行, 第20栏第25行—第20栏第57行, 图7, 图30.

EP 0933900 A, 1999.08.04, 全文.

WO 0138996, 2001.05.31, 全文.

审查员 刘斌

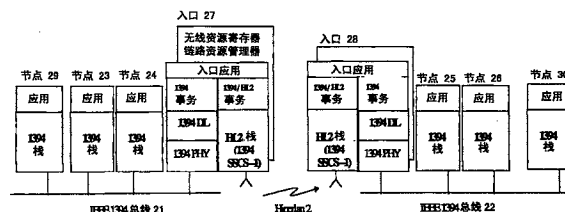
权利要求书 1 页 说明书 15 页 附图 5 页

(54) 发明名称

用于管理通信网络中的链路资源的方法

(57) 摘要

本发明涉及一种用于包括总线的管理通信网络中的链路资源的方法, 该总线由至少一个链路互联以形成一个其中至少一个链路对于某些与总线相连节点的协议层来说是透明的网络。本发明方法的特征在于包括步骤: 使至少一个链路上的可用资源信息对于认识链路的应用是可用的。



1. 一种用于管理包括至少两条总线的通信网络中的链路资源的方法,所述总线由该链路连接而成一个网络,其中该网络关于由总线使用的协议构成了单一虚拟总线,所述方法包括以下步骤:

在网络的节点中提供一个认识链路的应用,以获得链路资源预约信息;

访问所述的信息,以在该链路上的设备之间建立连接;以及

作为该信息的功能,确定该连接是否可以被建立。

2. 根据权利要求1所述的方法,其特征在于所述认识链路的应用从把总线连接到链路的入口之一获得所述信息。

3. 根据权利要求2所述的方法,其特征在于所述信息包括链路上的可用带宽。

4. 根据权利要求2或3所述的方法,其特征在于所述信息包括链路上的可用信道。

5. 根据权利要求4所述的方法,其特征在于一个应用通过以下步骤在网络的两个节点之间建立连接:

a 在总线上预约用于连接的信道和带宽;

b 识别要被连接的节点之间的通路上的链路;

c 如果步骤a成功,为所述通路上的每一个链路预约资源。

6. 根据权利要求5所述的方法,其特征在于为每一个链路预约资源的步骤包括步骤;

- 使应用请求预约与在步骤a从各个链路预约到的带宽和信道相同的带宽和信道。

7. 根据权利要求6所述的方法,其特征在于为每一个链路预约资源的步骤进一步包括步骤:

- 由应用将要与入口连接的节点的标识符发送到资源要被预约链路的入口;

- 作为被连接节点的功能,使链路判断请求的带宽是否可用;

- 作为此判断的功能,接受或拒绝带宽预约请求。

8. 根据权利要求7所述的方法,其特征在于为每一个链路预约资源的步骤进一步包括步骤:

- 使链路检查被请求的信道的可用性,如果该链路上的信道是可用的,接受信道预约;

- 检查是否达到了链路上可预约信道的最大值,并且在肯定的情况下拒绝其他的预约。

9. 根据权利要求7的方法,该方法包括了提供一个表示网络的信道可用性的寄存器和一个表示各个链路的信道可用性的寄存器的步骤,其特征在于两个寄存器都包括相同的信道数和各个链路可以并行预约的信道数的最大值,所述最大值小于或等于寄存器中的信道数,在接受了信道预约之后,为链路提供资源信息的入口检查已经被并行预约的信道是否达到了最大值,并在肯定的情况下修改链路的寄存器以表示对于该链路而言没有可用的信道。

用于管理通信网络中的链路资源的方法

技术领域

[0001] 本发明涉及一种用于管理通信网络中的链路资源的方法,该链路可以是无线链路,特别地该网络包括了在无线链路帮助下互连的有线通信总线。

背景技术

[0002] 在包括多个互连的传输媒介的网络中,各个传输媒介的可用带宽没有必要相同。

[0003] 例如,如果考虑到由 5GHz ETSI BRAN HiperLAN 2 无线链路连接在一起的两个 IEEE1394 有线串行总线,每个有线总线的可用带宽可以是 100Mb/s 或更大,而无线链路上的可用带宽可能被限制在 30Mb/s 之内。因此,对于使用这种链路的连接而言,无线链路造成了瓶颈。

[0004] 图 1 为这种网络的一种示例图。该网络包括两条总线 11 和 12,分别将源设备 13 与宿设备 14 以及宿设备 15 与源设备 16 连接起来。例如源设备 13 可以是数字 VCR,宿设备 14 和 15 为显示器而源设备 16 为调谐器。总线 11 还与入口 17 相连接,而总线 12 与入口 18 相连,这两个入口形成了总线 11 和 12 之间的无线链路。假设在 IEEE 1394 层,该无线链路对于设备 13 和 16 是透明的,即就是说这些设备认为它们处于同一物理总线上。

[0005] 假设无线链路上可用带宽为 30Mb/s,在 VCR13 与显示器 15 之间建立 20Mb/s 的第一连接。由于链路对设备 16 和 14 来说是透明的,可以在调谐器 16 与显示器 14 之间建立第二连接,但如果第二连接所需要的带宽大于剩余的无线带宽,不管大多少,显示器 14 也不能显示正确的图像。至于涉及到的创建第二连接的应用,在 IEEE1394 层仍然可以毫无意外地建立连接,这是因为该层中链路的透明性。

[0006] 参考文件 EP-A-0933900 公开了一种通过 1394 桥来互联 1394 总线的方法。在该文献中,每条总线和桥都由独立的总线构成,并且该方法允许来自不同总线的两个节点进行带宽和信道的预约,从而允许这两个节点之间的同步通信。该方法能在每条独立的总线上进行三个相应的带宽预约,一个用于发送节点,一个用于桥,一个用于接收节点。

[0007] 与上述的文献相反,在本发明中,连接总线的链路是透明的。这就是说,一条总线上的节点能够看见互联的总线上的所有节点,如同属于同一个网络一样。

发明内容

[0008] 本发明的目的是一种用于管理包括至少两条总线的通信网络中的链路资源的方法,所述总线由该链路连接而成一个网络,其中该网络关于由总线使用的协议构成了单一虚拟总线,所述方法包括以下步骤:

[0009] 在网络的节点中提供一个认识链路(link-aware)的应用,以获得链路资源预约信息;

[0010] 访问所述的信息,以在该链路上的设备之间建立连接;以及

[0011] 作为该信息的函数,确定该连接是否可以被建立。

[0012] 根据实施例,每个链路包括了多个入口,链路的至少一个入口包括了有关其链路

的所述资源信息。

[0013] 根据实施例,信息包括链路上的可用带宽。该信息还包括链路上的可用信道。

[0014] 根据实施例,一个应用(application)通过包括以下步骤的方法在网络的两个节点之间建立连接:

[0015] a 在总线上预约(reserving)用于连接的信道和带宽;

[0016] b 识别要被连接的节点之间通路上的链路;

[0017] c 如果步骤 a 成功,为所述通路上的每一个链路预约资源。

[0018] 根据实施例,为每一个链路预约资源的步骤包括步骤:

[0019] - 使该应用请求预约与在步骤 a 从各个链路预约的带宽和信道相同的带宽和信道。

[0020] 根据实施例,为每一个链路预约资源的步骤进一步包括步骤:

[0021] - 该应用将要被连接节点的标识符发送到资源要被预约链路的入口;

[0022] - 作为被连接节点的功能,使链路判断请求的带宽是否可用;

[0023] - 作为此判断的功能,接受或拒绝带宽预约请求。

[0024] 根据实施例,为每一个链路预约资源的步骤进一步包括步骤:

[0025] - 使链路检查被请求的信道的可用性,如果该链路上的信道是可用的,接受信道预约;

[0026] - 检查是否已经达到了链路上被预约信道最大值并且在肯定情况下,拒绝其他的预约。

[0027] 根据实施例,对于提供一个表示网络的信道可用性的寄存器和一个表示各个链路的信道可用性的寄存器的步骤,其中上述两个寄存器都包括相同的信道数和各个链路可以并行预约的信道数的最大值,所述最大值小于或等于寄存器中的信道数,在接受了信道预约之后,为链路提供资源信息的入口检查已经被并行预约的信道是否达到了最大值,并在肯定的情况下修改链路的寄存器以表示对于该链路而言没有可用的信道。

附图说明

[0028] 在附图的帮助下,在对无限制实施例的说明中会公开本发明其他的特性及优势,在附图中:

[0029] - 已说明过的图 1 所示为包括了无线链路的网络框图,并显示了由该链路构成的带宽瓶颈;

[0030] - 图 2 为根据本发明实施例的网络框图,该网络包括无线链路并显示了在构成无线链路的入口设备中的软件栈和实现本发明用到的寄存器;

[0031] - 图 3 所示为预约信道方法的流程图,该方法利用了指出了无线链路上可用信道的寄存器;

[0032] - 图 4 所示为释放被预约的信道方法的流程图;

[0033] - 图 5 所示为根据本发明不同实施例的链路上带宽的使用和预约的框图;

[0034] - 图 6 所示为释放图 5 所示带宽的框图;

[0035] - 图 7 所示为根据本发明不同实施例,在不同类型网络之间共享的多入口链路的框图;

[0036] - 图 8 所示为图 1 中网络的框图,这里为了减少无线链路的使用,调整了节点与总线连接。

具体实施方式

[0037] 尽管本实施例是基于使用 ETSI BRAN HiperLAN 2 无线链路,也可以使用其他技术,例如执行 Internet 协议 (IP) 的链路。此外,尽管在下面说明的实施例中,有线总线之间的链路是无线的,但没有必要必须如此。

[0038] 有关 IEEE1394 的信息可参看由 IEEE 公布的 IEEE 文件 1394-1995 及 IEEE 1394a-2000。

[0039] 1、基本信息

[0040] 为了简化说明,将在附图的帮助下介绍本实施例,这些附图给出了通常各包括两个入口的链路。不过,本发明也可以应用于包括两个以上入口的链路。

[0041] 图 2 所示的网络由两个 IEEE1394 总线 (也称为设备群集) 21 和 22 形成,每个总线各包括有线 IEEE1394 总线、多个设备 (分别为节点 23、24、29 以及 25、26、30) 以及通往无线 Hiperlan 2 链路的入口。所有节点均为 IEEE1394 设备。应用可以认识到无线链路的存在和位置。

[0042] 根据本实施例,入口 27 和 28 对其他设备来说不是透明的,也可以认为这两个入口是其各自总线上的节点,即,在总线复位之后其性质为物理标识符。然而在 IEEE1394 层的层面上,链路是透明的:认为所有的节点均位于单一总线上并因此在复位之后的性质为物理标识符。当执行总线复位时,入口 27 和 28 发出自身的识别分组,以表示其各自群集上的节点。

[0043] 各个节点都包括 IEEE1394 软件栈,即就是物理层、链路层、事务层以及应用层。每一个入口在其有线总线接口中也包括这些层。最后,这些入口利用 Hiperlan 2 协议栈进行通信。

[0044] 2、链路资源管理器与相关联的寄存器

[0045] 根据本发明,入口使与无线链路相关的资源预约信息对于不同群集中节点的应用是可用的。根据本实施例,该信息包括链路上的可用带宽和信道。该信息保留在无线链路的至少一个入口的“无线资源寄存器”中,并可以由应用或不同总线上节点的中间件 (middleware) 访问到。

[0046] 众所周知如 IEEE1394 1995 标准定义的 IEEE1394 总线包括了所谓的同步资源管理器 (或‘IRM’),该管理器保留了两个寄存器,‘BANDWIDTH_AVAILABLE’和 ‘CHANNELS_AVAILABLE’,分别指出了总线上的可用带宽和在最多 64 个同步信道中的可用信道。

[0047] 由于根据本实施例涉及 IEEE1394 层的无线链路是透明的,所以网络中存在单一的 IRM。

[0048] 与上述标准所定义的相同,表 1 给出 IRM BANDWIDTH_AVAILABLE 寄存器的格式:

[0049]	0 19 比特	bw_remaining 13 比特
--------	------------	-----------------------

[0050] 表 1

[0051] bw_remaining 的初始值等于 0X1333 个 (在 10 进制系统中为 4915) 带宽单元。当

每次节点的应用期望预约有线总线上的带宽时,应用会检查此值并且如果有足够可用带宽时,减去所需的带宽。否则不能建立连接。

[0052] 与上述标准所定义的相同,表 2 给出了 IRM CHANNELS_AVAILABLE 寄存器的格式:

[0053]	channels_available_hi
	channels_available_lo

32 比特

[0054] 表 2

[0055] 该寄存器的初始值为 0xFFFFFFFF。每一位表示了特定信道的可用性。' channels_available_hi' 的最低可用位表示信道 0 的可用性,而 ' channels_available_lo' 的最高可用位表示信道 63 的可用性。当每次应用期望预约信道时,应用会检查该寄存器,选择可用信道并将寄存器的对应位置为 ' 0' 。当没有足够多的可用信道时,不能建立连接。

[0056] 为了预约带宽或信道,首先由应用通过读取请求消息读取寄存器。应用接收包含了该寄存器内容的读取响应消息。在检查完被请求资源的可用性之后,应用锁定对应的寄存器的访问权并写入适当的新值。锁定和写入操作由应用发送锁定请求以及 IRM 发送锁定响应来完成。根据 IEEE1394 1995,分开进行对各个寄存器的访问。根据 IEEE1394 2000,万一有锁定请求不利冲突,由于锁定响应发回了实际值,则没有必要进行读取。

[0057] 根据本实施例,资源管理器在链路层实现。以下论述中这种资源管理器被称为“链路资源管理器”或 LRM,以区别于 IEEE1394 1995 定义的 IRM。

[0058] 根据本实施例,LRM 在无线链路的一个入口执行。它管理的两个寄存器, LINK_BANDWIDTH_AVAILABLE 和 LINK_CHANNEL_AVAILABLE 分别表示了无线链路上的可用带宽和可用信道(除了其他信息之外)。

[0059] 表 3 给出了根据本实施例的 LRM LINK_BANDWIDTH_AVAILABLE 寄存器的格式:

[0060]	Source_PhyID	Sink_PhyID	f	信道	LRM_bw_remaining
	6 比特	6 比特	1	6 比特	13 比特

[0061] 表 3

[0062] LRM_bw_remaining 字段和 IRM BANDWIDTH_AVAILABLE 寄存器中 bw_remaining 字段的作用相同。假设链路上的可用带宽与有线总线上的可用带宽不同,则该参数的初始值也不同。如果链路上的可用带宽较低,这个字段的初始值也就较低,而如果链路上的可用带宽较高,由于限制因子是与无线链路相连接的总线上的带宽,所以当进行预约时,该参数永远也不会到 0。

[0063] Source_PhyID 字段包含了源设备的物理标识符。Sink_PhyID 字段包含了宿设备的物理标识符。这些字段的内容由进行预约的应用来定义(利用锁定请求消息),目的是形成讲话者节点(源)与收听者节点(宿)的身份 LRM。通过这些标识符,LRM 可以判断与 LRM_BANDWIDTH_AVAILABLE 寄存器的 ' link_bw_remaining' 字段所指出的带宽不相同的实际可用带宽,这将在下文中进行解释。

[0064] 根据不同的实施例,请求的 Source_PhyID 和 Sink_PhyID 字段包含了要被预约的连接之间入口的物理标识符。

[0065] 在入口多于两个的无线链路中,任意两个入口之间的可用带宽可能是不同的,例如是入口间传送质量的函数。为了指出这些入口对之间的实际带宽,LRM 必须知道哪

两个入口是连接的一部分。LRM 可以从讲话者和收听者的身份推断出入口。在特定情况下, 'link_bw_remaining' 字段指出的带宽就不能反映任意两个入口间的实际可用带宽。

[0066] 根据本实施例, 'link_bw_remaining' 字段指出了无线链路的任意两对入口之间的最大可用带宽。如果其请求的数量大于 'link_bw_remaining' 字段指出的数量, 则应用可以立即判断出不可能预约带宽。在其他所有情况下, LRM 首先计算涉及到预约的两个端口间的可用带宽并最后在锁定响应中接受或拒绝预约, 并在响应的 'link_bw_remaining' 字段给出源和宿之间的真实可用带宽。

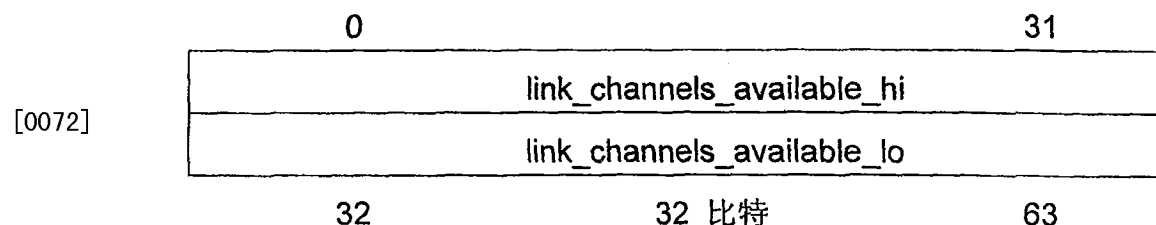
[0067] 根据一个变化实施例, 'link_bw_remaining' 字段包含假值 (dummyvalue), 此值大于任意应用可能请求的带宽。在这种情况下, 由于应用不能从其读取请求中推断出任何的无效带宽信息, 所以 LRM 始终会拒绝带宽预约。

[0068] 在将无线链路严格限制为两个入口的情况下, 链路上的可用带宽不是源和宿设备的函数。根据不同的实施例, 在包括了这种链路的网络中, 不需要 LRM BANDWIDTH_AVAILABLE 寄存器的前两个字段。不过, 还有可能在无线链路上的两个入口使用这些字段。

[0069] 'Channel' 字段包括由应用进行预约所请求的信道数目。此数目与分配给 IRM 的 CHANNEL_AVAILABLE 寄存器的数目相同, 就像后面会看到的, 这是因为应用在向无线链路的 LRM 进行预约之前, 首先要向 IEEE1394 网络的 IRM 进行预约。为了访问 LRMLINK_BANDWIDTH_AVAILABLE, 通过锁定请求将该字段的内容提供给 LRM。

[0070] 'f' 字段包括用于指示链路上被请求信道的可用性的标记。该字段用于锁定响应, 以指示被请求的信道是可用 ('0') 还是无效 ('1')。这由 LRM 设置。

[0071] 表 4 定义了 LRM LINK_CHANNELS_AVAILABLE 寄存器的格式。此格式与 IRM CHANNELS_AVAILABLE 寄存器的格式相同, 尽管二者功能并不同。



[0073] 表 4

[0074] 该寄存器的内容与 IRM 的 CHANNELS_AVAILABLE 寄存器的内容有着相同的意义。

[0075] 无线链路允许的信道最大值可能会小于或大于 IEEE1394 总线 64 个信道的界限。尽管链路容许比 IEEE1394 信道数更多的信道数目, 但只能使用 64, 这是由于 IEEE1394 网络的 IRM 仅限于这个数字。

[0076] 如果无线链路允许更少的信道数目, 则使用下列算法。图 3 给出的算法用于预约机制而图 4 的用于释放机制。

[0077] 将 link_channels_available(link_channels_available_hi 和 link_channels_available_lo) 位设为 1 作为初始值。

[0078] 当应用请求被链路的 LRM 分配为空闲且不是最后一个可能的信道时, 接受预约该信道并将该信道的标记设为 '0'。

[0079] 当应用请求信道且该信道为链路 (尽管在 IRM 的 CHANNELS_AVAILABLE 寄存器中可能还有可用信道) 上最后一个可用信道时, 接受预约该信道并将 LINK_CHANNELS_

AVAILABLE 寄存器的所有标记设为 ‘0’，这是为了表示出通过这个链路已经没有可用信道了。

[0080] LRM 将真正被应用预约的寄存器的信道列表保存在存储器中以与那些因为无线链路强加的限制而被简单地设为无效的信道相比较。

[0081] 当应用释放信道时，将指定信道的标记设为 ‘1’。如果该链路已被完全预约，这次信道释放还会触发将所有未预约信道的标记设为 ‘1’ 的操作。

[0082] 管理 LINK_CHANNELS_AVAILABLE 寄存器和管理 IRM 的 CHANNELS_AVAILABLE 寄存器的方式是不同的。前者会接受读取请求，但不会接受锁定请求。这样，应用会检查无线链路上的信道是否保持可用。LRM 会遵循 BANDWIDTH_AVAILABLE 寄存器上的应用锁定请求进行实际的信道预约。

[0083] 3、寄存器的使用

[0084] LINK_BANDWIDTH_AVAILABLE 寄存器除了只指示可用带宽之外，还有更多的功能。当利用读取请求进行读取时，返回链路上的可用带宽，但锁定请求不仅要触发带宽的预约，还会触发信道的预约。于是只需要在读取模式下访问 LINK_CHANNELS_AVAILABLE 寄存器，在锁定模式下则不需要。

[0085] 希望进行预约的应用首先执行 IRM 预约，接着利用获得的信道数目和预约的带宽进行源节点和宿节点之间无线链路上的预约。

[0086] 当在 LINK_BANDWIDTH_AVAILABLE 寄存器上执行锁定请求时，调用者必须提供 IEEE1394 网络上源设备的 PhyID 号、IEEE1394 网络上宿设备的 PhyID 号、被请求信道（IRM 已对该信道进行了分配）以及新的带宽值，即读取请求过程中读出的值减去被预约的值。

[0087] 用一条消息（即锁定请求）来执行所有的分配（带宽和信道）。如果链路 LRM 有能力处理其入口设备之间不同的带宽，则 LRM 使用源和宿 PhyID。

[0088] 通过表 5 至表 7 可以说明 LINK_BANDWIDTH_AVAILABLE 寄存器的行为，表 5 至表 7 所示分别为初始值、在读取请求之后获得的值以及在锁定请求之后获得的值。

[0089]

初始值				
0	0	0	0	4915 或更小
6 比特	6 比特	1	6 比特	13 比特

[0090] 表 5

[0091]

读出的值				
zeros	zeros	0	zeros	Link_bw_remaining
6 比特	6 比特	1	6 比特	13 比特

[0092] 表 6

[0093]

锁定结果				
忽略	忽略	i	忽略	条件写
6 比特	6 比特	1	6 比特	13 比特

[0094] 表 7

[0095] 其中，（ ‘i’ =忽略 ）

[0096] 当执行读取请求时,link_bw_remaining 值返回读取响应。

[0097] 锁定请求包括预先读取的带宽值 (根据 IEEE1394 1995 词汇表为 arg_value 字段) 和新的带宽值 (' data_value')。LRM 将 link_bw_remaining 字段的当前值与锁定请求接收到的值进行比较。如果这些值相等,由于在当前读取和锁定请求之间没有同意其他请求,则进行预约。写入到 link_bw_remaining 字段的新带宽值不必是在锁定请求和锁定响应中发送的值,这是由于前面讨论的用于多入口链路的计算规则。换句话说,下一次的读取请求没有必要导致保留上一次锁定请求发送的带宽值。

[0098] LINK_BANDWIDTH_AVAILABLE 寄存器锁定请求失败的可能原因有 :

[0099] 1 在当前读取和锁定请求的时间间隔内更改了链路上的可用带宽,所以锁定请求的 ' arg_value' 无效。LRM 管理这种情况的方式与 IRM 相同。

[0100] 2 链路上源和宿之间的可用带宽不够。当读取寄存器时,因为返回 link_bw_remaining 字段的值可能是该链路上用于最优传输的可用带宽,所以可能检测不到这一点。由于 LRM 知道来自锁定请求消息的 PhyID 并能够计算出用于该特定连接的真实可用带宽,于是容量由相关入口确定的带宽不同的链路可能会拒绝恢复源和宿 PhyID 之间通路上真实带宽的锁定。

[0101] 3 需要的信道不再可用。这只有在其他应用分配了其他信道 (因为被请求的信道仍然被分配在 IRM) 且这个其他信道是链路上最后一个可用信道才有可能发生,尽管在 IRM 的 CHANNELS_AVAILABLE 寄存器中某些信道仍保持空闲。在这种情况下,通过链路的 LRM 不能建立连接。为了反映这种情况,LRM 在锁定响应中将 ' f' 位置为 1。

[0102] 通过表 8 和表 9 可以说明 LINK_CHANNELS_AVAILABLE 寄存器的行为,表 8 和表 9 所示分别为初始值、在读取请求之后获得的值以及在锁定请求之后获得的值。

初始值	
[0103]	0
	0
32 比特	

[0104] 表 8

读取的值	
[0105]	在 LINK_BANDWIDTH_AVAILABLE 寄存器上的最后一次成功锁定
	在 LINK_BANDWIDTH_AVAILABLE 寄存器上的最后一次成功锁定
32 比特	

[0106] 表 9

[0107] 在应用从任意源节点和宿节点之间链路的 LRM 获得需要的预约失败的情况下,必须请求已预约资源的存储单元分配。

[0108] 4、利用 LINK_BANDWIDTH_AVAILABLE 寄存器的例子

[0109] 参照表 10 至表 16,在下文中给出了一个利用 LRM 的 LINK_BANDWIDTH_AVAILABLE

寄存器的例子。假设在本例的结构中链路最多支持两个信道。

[0110] 以总可用带宽为 4915 单元为例，表 10 给出了寄存器的初始值。

[0111]	0	0	0	0	4915
	6	6	1	6	13

[0112] 表 10

[0113] 图 2 中节点 23 的应用希望分配节点 23(源)和节点 25(宿)之间的链路上的信道和带宽。首先,遵循 IEEE1394 1995 规则,在 IRM 请求信道和带宽的分配。例如已分配的信道为信道 7 且带宽为 500 个单元。于是节点 23 的应用执行 LRM 的 LINK_BANDWIDTH_AVAILABLE 寄存器上的锁定请求,参数如表 11 所示。

[0114]	Arg_value Data_value	源	宿		信道	带宽
		0	0	0	0	4915
		23	25	0	7	4415
		6	6	1	6	13

[0115] 表 11

[0116] 如果接受了锁定(并且更新了 LINK_CHANNELS_AVAILABLE 寄存器),按照表 12 的参数给出响应。

[0117]

New_value	23	25	0	7	4415
	6	6	1	6	13

[0118] 表 12

[0119] 接着,节点 26 决定分配节点 26(源)和节点 24(宿)之间的链路上的信道和带宽。假设 IRM 已经进行了预约。作为例子,该信道为信道号 33 且需要的带宽为 1000 个单元。同时,如表 13 所示,节点 25 的应用执行 LRM 上的锁定请求且请求信道 12 以及源节点 29 和宿节点 30 之间的 2000 个带宽单元。

[0120]

Arg_value Data_value	0	0	0	0	4415
	29	30	0	12	2415
	6	6	1	6	13

[0121] 表 13

[0122] 如果接受了锁定(并且更新了 LINK_CHANNELS_AVAILABLE 寄存器),则按照表 14 给出响应。

[0123]	New_value	29	30	0	12	2415
		6	6	1	6	13

[0124] 表 14

[0125] 由于该链路只能提供两个信道,拒绝节点 26 应用的锁定请求。

[0126]	Arg_value	0	0	0	0	4415
	Data_value	26	24	0	33	3415
		6	6	1	6	13

[0127] 表 15

[0128] 拒绝了锁定响应且寄存器的字段成为：

[0129]	New_value	26	24	1	33	2415
		6	6	1	6	13

[0130] 表 16

[0131] 将‘f’标记设为 1,表示在该链路上再也没有信道了。link_bw_remaining 字段显示出最终的更新值,即就是同意节点 25 的应用的请求之后。

[0132] 锁定响应包括了 Source_PhyID、Sink_PhyID 和 Channel 字段的值,以帮助呼叫方恢复其请求的内容。

[0133] 当预约或释放资源时,可以采用以下操作：

[0134] “分配新值”：

[0135] 该操作是分配新信道和带宽的数量。信道为可用、空闲的,所需的带宽是可用的。该操作被称为分配是因为在请求(‘arg_value’)下发送的预先读取的带宽值要大于写入到 link_bw_remaining 字段(‘data_value’)的新值。将 channel_available 标记置为 0。

[0136] LRM 记住了信道和已分配带宽之间的关系。

[0137] 根据不同的实施例,可以预约信道而不预约带宽。这种可能性可以由入口方决定。

[0138] “分配更多”：

[0139] 如果有足够多的可用带宽,该操作是向预先被分配的信道分配附加的带宽。LRM 根据锁定请求的信道号对应着已被分配的信道这一事实来识别这种操作。将请求的带宽之差加到已被该信道预约的带宽中。channel_available 标记保持为 0。入口不能阻止在已分配信道预约带宽的应用发送另一个会在总线上引起混乱的数据流(同 1394-1995 一样)。

[0140] “完全释放”：

[0141] 该操作是释放信道以及分配给该信道带宽。LRM 将该操作识别为释放是因为锁定请求(包含了读取请求的结果)的(‘arg_value’)字段小于其‘data_value’(包含了要写入到 link_bw_remaining 字段的新值)。前两个字段之差等于(或大于)分配给该信道的带宽这一事实显示出资源的释放为完全释放的情况。将 LINK_CHANNELS_AVAILABLE 寄存器的信道标记置为空闲(1)。在锁定请求指出不止要释放已分配带宽情况下,只释放已分配的带宽。

[0142] “部分释放”：

[0143] 该操作是只释放给定信道的部分带宽。LRM 根据锁定请求 bw_remaining 字段的‘arg_value’小于其‘data_value’这一事实将请求的操作识别为释放,而因为差值

小于分配给该信道的带宽,则就是部分的。信道在 LINK_CHANNELS_AVAILABLE 寄存器中保持预约 (对应的标记保持为值 0)。减去信道被释放的带宽并且使 LRM 与其保持联系。

[0144] 在预约了链路上所有可用信道的情况下,将 LINK_CHANNELS_AVAILABLE 寄存器中所有的标记置为 0。LRM 仍然保持那些真正被应用预约的信道列表,以和那些只是在寄存器中被标记的信道相比较。当 LRM 接收到用于“分配更多”操作的锁定请求时,只有当该请求对应着真正预约的信道时,才会给该请求分配附加的带宽。

[0145] 5、利用 LINK_CHANNELS_AVAILABLE 寄存器的例子

[0146] 分别参照给出不同时刻寄存器内容的表 17 至表 21, 下文中给出了一个利用 LINK_CHANNELS_AVAILABLE 寄存器的例子。根据本例, 无线链路的容量为三个信道。

[0147] 最初,没有使用无线链路上的信道。LINK_CHANNELS_AVAILABLE 寄存器的状态由表 17 给出:

[0148]

[illegible]

[0149] 表 17

[0150] 例如第一个应用预约信道 7。分配之后寄存器的状态由表 18 给出。

[0151]

[illegible]

[0152] 表 18

[0153] 接着第二个应用预约信道 33, 寄存器的状态变为:

[0154]

[illegible]

[0155] 表 19

[0156] 在该链路上只剩下一个可用信道。当第三个应用分配信道 2 时,寄存器的状态变为:

[0157]

[illegible]

[0158] 表 20

[0159] 由于该链路不能提供更多信道,将所有的信道标记置为0。如果第四个应用读取了该链路的寄存器,它就会推断出没有信道可以预约了。

[0160] 现在,如果第二个应用释放了信道 33,寄存器的状态变为:

[0161]

[illegible]

[0162] 表 21

[0163] 现在在无线链路上有一个信道是可用的:所有未使用信道的标记置为 1。

[0164] 有利地,首先需要由全局 IRM 来预约信道这一事实保证了不会出现与在多个无线链路上预约信道识别相关的冲突。如果首先不由 IRM 进行分配,则 LRM 不可能预约信道。

[0165] 网络的应用可能认识或不认识到无线链路。在认识链路的应用创建了一个或多个无线链路的情况下,这些链路的 LRM 可以通过其自身预约请求的资源,或根据其他实施例,给由认识链路的应用做出的连接以优先权。

[0166] 6、连接建立的方法

[0167] 这里简要给出几种根据本实施例的认识链路使用的连接方法。这些方法涉及连接的打开、连接的破坏、总线复位之后连接的建立以及对已有连接的连接覆盖。

[0168] 在下面几个方法的例子中,为了判断在讲话者节点与听话者节点之间的通路是否有一个或几个资源被预约的链路,应用需要获得网络的拓扑。

[0169] 应用可以从在总线复位之后建立一拓扑图的 IEEE1394 物理层得到该拓扑图。该拓扑图表示了不同的节点是如何连接的。

[0170] 根据本实施例,每一个入口在其存储器中包含了用于指示它是入口的标识符。为了判断这些节点是否是入口,一旦应用得到拓扑图——这只在复位之后需要——就会访问网络上所有节点的存储器。当在拓扑图中连续检测到两个入口时,会识别链路。

[0171] 6.1 打开连接

[0172] 为了打开连接,应用必须:

[0173] 1 判断网络的拓扑图并检查在源节点与宿节点之间的通路上是否有一个或几个链路。

[0174] 2 如果呼叫方客户（即应用）没有给出带宽，在源节点的输出插头控制寄存器（‘oPCR’，与 IEC61883 的定义相同，说明了 IEEE1394 总线上同步数据流的传送）上执行读取请求以获取所需的带宽。

[0175] 3 在下面基础上执行读取请求：

[0176] IRM BANDWIDTH_AVAILABLE 寄存器（为了获得 IEEE1394 网络上可用带宽的数量）；

[0177] IRM CHANNELS AVAILABLE 寄存器（为了获得 IEEE1394 网络上可用信道的列表）；

[0178] LRM LINK_BANDWIDTH_AVAILABLE 寄存器（为了获得各个链路上可用带宽的数量），用于由连接交叉形成的各个链路；

[0179] LRM LINK_CHANNELS_AVAILABLE(为了检查信道的可用性),用于由连接交叉的各个链路;

[0180] 宿节点的输入插头控制寄存器 (IEC61883 定义的 'iPCR'), 用于核实该节点为空闲且可以接收数据流。

[0181] 4 在下面的基础上执行锁定请求:

[0182] IRM BANDWIDTH_AVAILABLE 寄存器 (为了在 IEEE1394 网络上预约带宽);

[0183] IRM CHANNELS_AVAILABLE 寄存器 (为了在 IEEE1394 网络上预约信道);

[0184] 5 如果请求成功了, 在下面的基础上执行锁定请求:

[0185] LRM LINK_BANDWIDTH_AVAILABLE 寄存器, 用于由连接交叉形成的各个链路 (为了预约各个链路上的带宽), 该寄存器包括源 PhyID、宿 PhyID 以及在 IRM 已经被分配的信道号;

[0186] 源的 oPCR (为了调整该寄存器以显示出连接的存在)

[0187] 宿的 iPCR (为了调整该寄存器以显示出连接的存在)

[0188] 6.2 关闭连接

[0189] 为了关闭连接, 应用必须:

[0190] 1 在下面的基础上执行读取请求:

[0191] 源的 oPCR 核实该源没有被覆盖 (即就是只连接了一个收听者节点部分)

[0192] IRM BANDWIDTH_AVAILABLE 寄存器;

[0193] LRM LINK_BANDWIDTH_AVAILABLE 寄存器, 用于由连接交叉形成的各个链路。

[0194] 2 在下面的基础上执行锁定请求:

[0195] IRM BANDWIDTH_AVAILABLE 寄存器

[0196] IRM CHANNELS_AVAILABLE 寄存器

[0197] LRM LINK_BANDWIDTH_AVAILABLE 寄存器, 用于由连接交叉形成的各个链路

[0198] 源节点的 oPCR

[0199] 宿节点的 iPCR

[0200] 6.2 网络复位之后重建连接

[0201] 总线复位由链路传播。无论复位发生在何处, 创建连接的应用都能意识到总线复位的发生。基本上, 重建所遵循的规则与创建连接的规则相同。LRM 寄存器与 IRM 寄存器复位的方式相同。

[0202] 6.3 覆盖连接

[0203] 为了覆盖连接, 应用必须:

[0204] 1 得到拓扑图, 并检查在源节点与宿节点之间的通路上是否有一个或几个链路。推断出哪一个节点已经被连接预约。

[0205] 2 在源的 oPCR 上执行读取请求, 以了解所需的带宽 (如果呼叫方客户没有给出带宽)。

[0206] 3 在下面的基础上执行读取请求:

[0207] LRM LINK_BANDWIDTH_AVAILABLE 寄存器 (为了获得各个链路上可用带宽的数量), 用于由连接交叉形成且没有被预约的各个链路。

[0208] LRM LINK_CHANNELS_AVAILABLE 寄存器 (为了检查信道的可用性), 用于由连接交叉形成且没有被预约的各个链路。

[0209] 宿节点的 iPCR, 用于核实该节点为空闲。

[0210] 4 在下面的基础上执行锁定请求：

[0211] LRM LINK_BANDWIDTH_AVAILABLE 寄存器（为了获得各个链路上可用带宽的数量），用于由连接交叉形成且没有被预约的各个链路；

[0212] LRM LINK_CHANNELS_AVAILABLE 寄存器，用于由连接交叉形成的各个链路，该寄存器包括源 PhyID、宿 PhyID 以及在 LRM 已经分配的信道号。

[0213] 源节点的 oPCR

[0214] 宿节点的 iPCR

[0215] 利用上述，应用可以检测到是否建立了连接以及哪个链路未能提供充分的资源。根据不是这里所述目的的方法，为了避免失败的连接，可以建议应用更改网络的拓扑图。

[0216] 图 8 表示了图 1 的网络，其中源 13 和 16 分别与总线 12 和 11 相连以响应应用对用户的指示。

[0217] 7、带宽使用的改善

[0218] 如果第一个应用在可提供 30Mb/s 的 HiperLAN2 链路上预约了 20Mb/s 带宽，但只用了 10Mb/s，浪费了 10Mb/s 的带宽。需要 15Mb/s 的第二个应用就不能实现预约。

[0219] 根据变化实施例，链路的 LRM 核实与预约带宽相比应用的实际使用带宽并得到可用差额以用于其他应用的分配。

[0220] 图 5 所示的竖条表示了链路上的可用带宽。出于本例的目标，总可用带宽为 10 个单元。对于每一个应用预约的信道来说，LRM 保留了两个参数。第一个参数（下文中标为 'R'）表示被预约带宽。第二个参数（标为 'O'）表示被预约带宽与信道上应用实际使用带宽之差。O 表示用于该信道的最优化带宽。

[0221] 根据图 5 中例子，三个应用 (A1, A2, A3) 依次预约带宽。字母 R 和 O 旁边的数值分别表示了用于各个应用的带宽值。

[0222] 第一个应用 A1 在信道 7 上预约了 3 个单元（步骤 1），但 LRM 推断出 A1 只使用了一个。将用于 A1 的参数 O 置为值 2（步骤 2）。接着应用 A2 在信道 33 上预约了 4 个单元（步骤 3），但只使用了两个：用于该应用的参数 O 也为值 2（步骤 4）。最后，应用 A3 在信道 2 上预约了 4 个单元并且全部都使用了（步骤 5）。这样可以将 3 个单元分配给其他的应用并且 LRM 适当地设置新的 link_bw_available 值。

[0223] 图 6 表示了当应用开始释放带宽时图 5 中的带宽。由于应用没有意识到自己并没有把预约的带宽全部留给自己使用，它们会利用适当的锁定请求释放预约的数量。

[0224] 假设应用 A1 希望释放其信道和带宽（步骤 6）。它会发送试图将 link_bw_available 值增大到 3 的锁定请求，即三个单元。LRM 会利用包括 link_bw_available 值增加三个单元的正锁定请求来假装释放应用 A1 的三个单元，但实际上只会用实际使用的单元数量，即 1，来有效地增加该值。应用于 A3 和 A2 的资源释放概念是相同的。

[0225] 根据这个过程，应用可能必须使在其锁定请求中使用大于在 LINK_BANDWIDTH_AVAILABLE 寄存器中给出的最大带宽的带宽值。例如在步骤 8，为了释放带宽的 4 个单元，由于寄存器的当前值是 8，应用 A2 会在其锁定请求中使用期待值（'data_value'）12。

[0226] 不过，应用的带宽释放肯定不会导致使用的带宽值超过 0X1FFF，这是因为该值表示的是为该目标预约的 13 位编码的最大值。因此，LRM 提供的一个信道上带宽的改善不会大于 0X1FFF 减去链路上可用带宽的最大值。

[0227] 例如,链路带宽为 0X1333 时,用于任意一个信道改善的最大值不会超过 0XCCC。

[0228] 8、利用单一链路的多个网络互连

[0229] 根据变化实施例,几个类型的网络可以同时使用网络。图 8 所示为包含了四个入口 74 至 77 的链路框图。入口 74 和 75 分别与两个 IEEE1394 总线 70 和 71 相连接。入口 76 和 77 分别与两个 IP 网络 72 和 73 相连接。链路以透明方式连接两个总线 70 和 71,也就是说,在较低层认为所有的总线设备在单一总线上。对于 IP 网络来说也是如此:认为所有的 IP 设备在单一 IP 网络上。总线 70/71 组成的网络以及 IP 网 72/73 的网络均忽略了其他网络的存在。它们仍然可以共享使用由入口 74 到 77 形成的链路资源。入口可以包括在单一设备中或由分离设备形成(例如在无线链路的情况下)。

[0230] 入口必须将网络资源分配给所有网络类型。根据本实施例,会给各种类型的网络提供单独的 LRM。在本例的情况下,给 IEEE1394 总线的网络提供了一个 LRM,并给 IP 网络提供了一个 LRM。在下文中将这两个 LRM 称为 LRM-1394 和 LRM-IP。LRM-IP 向 IP 设备提供的接口不需要与 LRM1394 向 LRM1394 节点提供的相同。

[0231] 由于这两个 LRM 被分配了相同的资源,它们根据链路实际的操作交换有关资源分配的信息。

[0232] 前面已经讨论过,即使链路的容量更大,例如在两个总线之间链路上使用的带宽通常也不会超过总线上的可用带宽。在现在这种情况下,几个网络竞争相同资源这种情况对于仅在确定环境中的给定网络会有限制效果。

[0233] 考虑下面这种情况:

[0234] 链路带宽大于 IEEE1394 带宽。

[0235] 在这种情况下,除非被 IP 网预约的带宽使总的链路带宽减少到小于 0X1333 带宽单元(IEEE1394 单元),否则在 IP 网上预约的带宽不会减少 LRM-1394 的 LINK_BANDWIDTH_AVAILABLE 寄存器中 bw_remaining 的值。

[0236] 将同样的行为应用于信道。

[0237] 预约是真实的(即 IEEE1394 网络的带宽和信道预约)

[0238] 9、LRM 选择

[0239] 本发明作用于点对点链路(每个网络只包括一对入口)和多点网络(每个网络包括多于两个入口)。在这两种情况下,在给定网络中 LRM 设备的选择存在两种可能性:一个入口(对每个网络而言)承担 LRM 的任务且所有应用向该设备发送消息;或是各个应用向与其连接的入口发送消息且链路内部的专用协议使入口同步。

[0240] 换句话说,在第一种情况下,由从所有候选入口中选出的单一入口来管理 LRM,各个应用都了解当选的入口,而在第二种情况下,没有进行选择且各个应用都不知道 LRM 的位置,只是简单地将请求发送到距离它们最近的入口。

[0241] 在包括了链路的网络为 IEEE1394 总线网络情况下,现在对这两种可能性进行说明。

[0242] (a) 单一 LRM 入口选择

[0243] 一个入口承担 LRM 的任务。在网络或总线复位期间执行选择。这里将介绍两种选择 LRM 入口的选择方法。第一种方法基于对网络拓扑图的分析。第二种方法基于入口的选择,随后是向应用宣布或公开当选 LRM 的物理识别符。

[0244] 拓扑算法

[0245] 根据本方法, 候选入口中 PhyID 值最大的入口当选为 LRM。这种算法的优点在于非常简单, 这是因为很容易利用到来自 IEEE1394 层的拓扑。在每次总线复位之后, 应用必须检查拓扑以确定各个链路的 LRM。

[0246] 专用链路选择

[0247] 该方法执行了链路 (根据链路的特殊协议, 即入口之间) “内部” 的动态协议, 这意味着在总线复位之后可以更换 LRM (例如如果删除了是 LRM 的入口)。

[0248] 而且, 为了使 IEEE1394 应用了解哪个节点表示 LRM, 在入口的配置存储器 (' config rom') 中定义特定的条目, 这个新条目存在于所有入口设备的配置存储器中。

[0249] 表 22 给出了根据本例的存储器条目的格式。可能是其他格式。

[0250]

8 比特	1 比特	6 比特
Key_ID = 30 ₁₆	s 预约	Phy ID

[0251] 表 22

[0252] ‘s’ 为同步位。当 ‘s’ 为 ‘1’ 时, 认为 Phy ID 值不是最新值且现在正在改变。这可能发生在例如总线复位之后。当 ‘s’ 等于 ‘0’ 时, Phy ID 值表示用于此链路的 LRM 的 Phy ID。更新 Phy ID 值的方法是链路专用的。

[0253] (b) 无 LRM 入口选择

[0254] 根据第二种方法, 各个应用向与其连接的入口发送消息 (即读取请求和锁定请求)。每一个入口执行 LRM 寄存器。链路内部的专用协议实现了入口之间的同步以避免同时发生的多锁定请求引起的冲突。

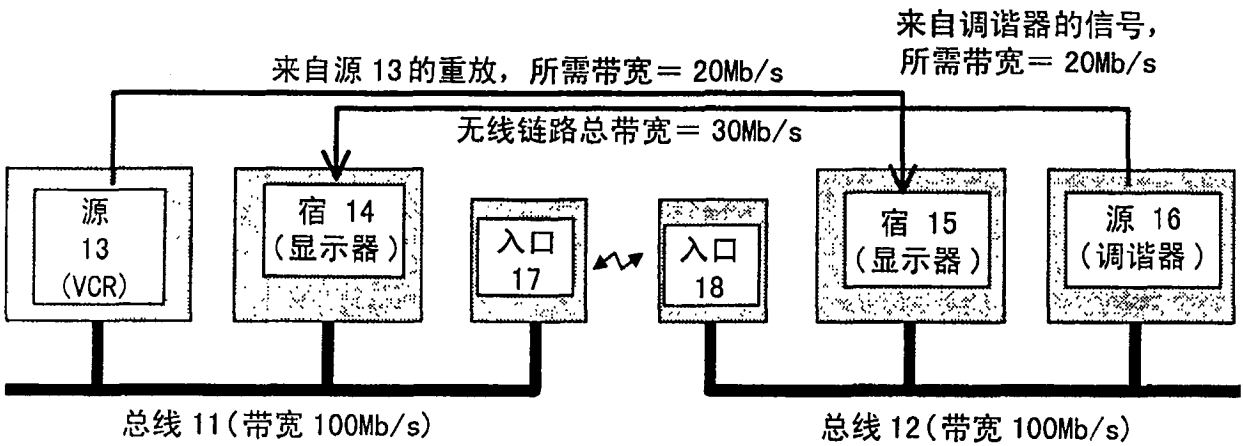


图 1

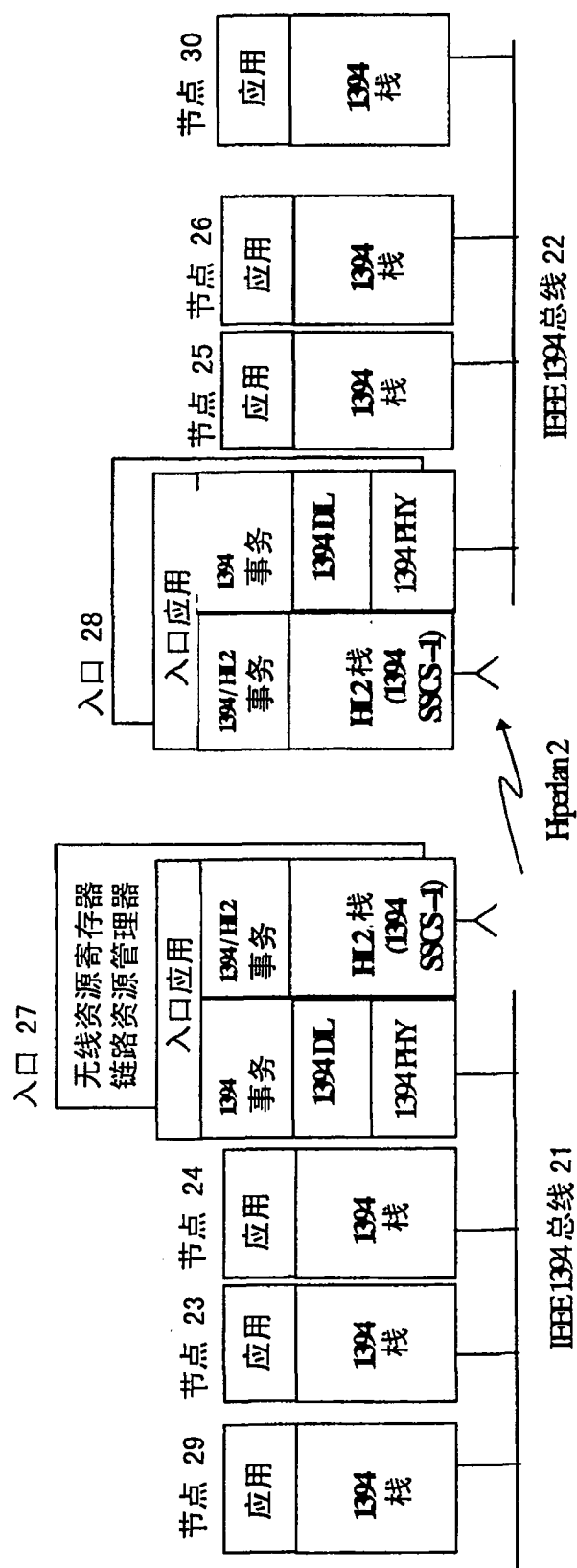


图 2

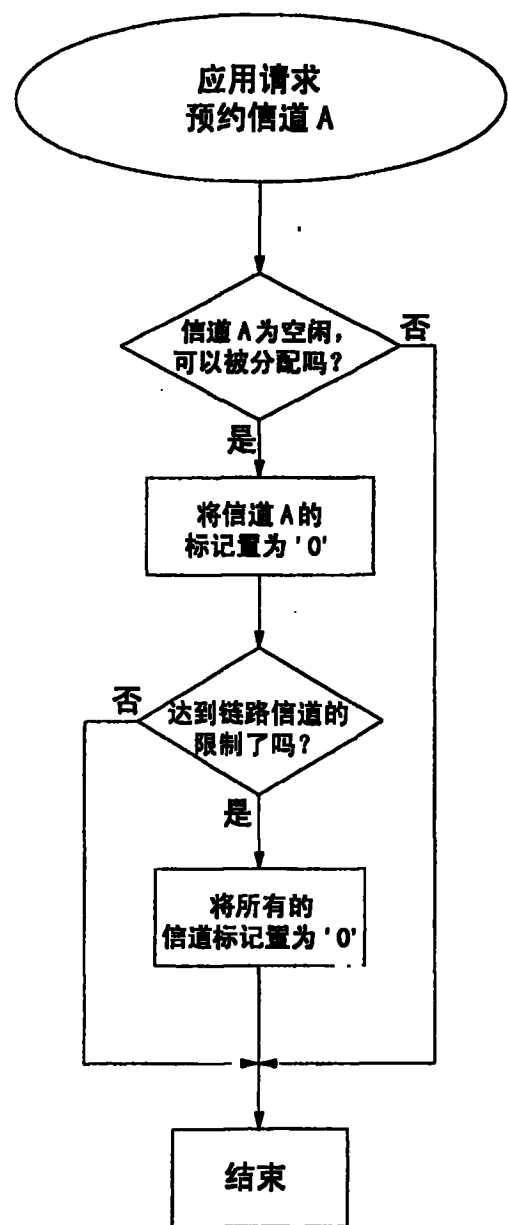


图 3

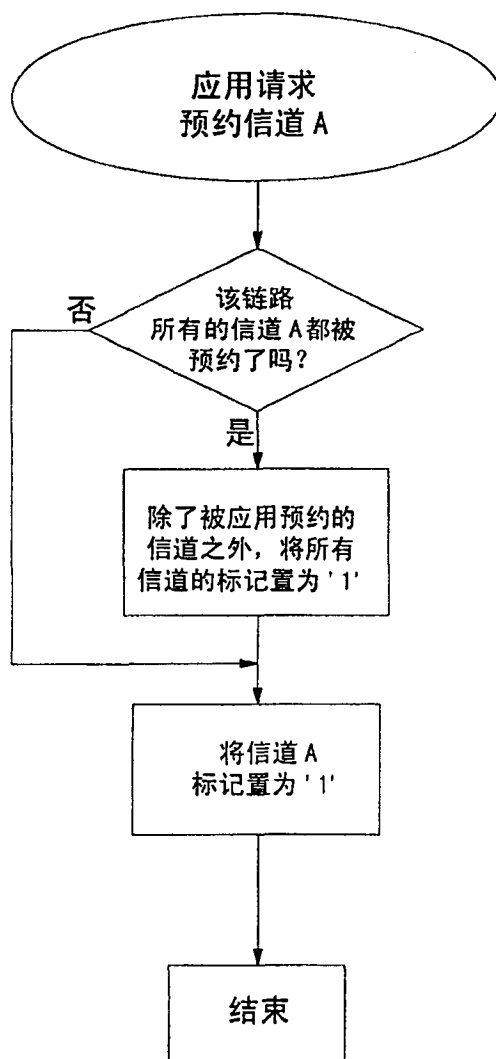


图 4

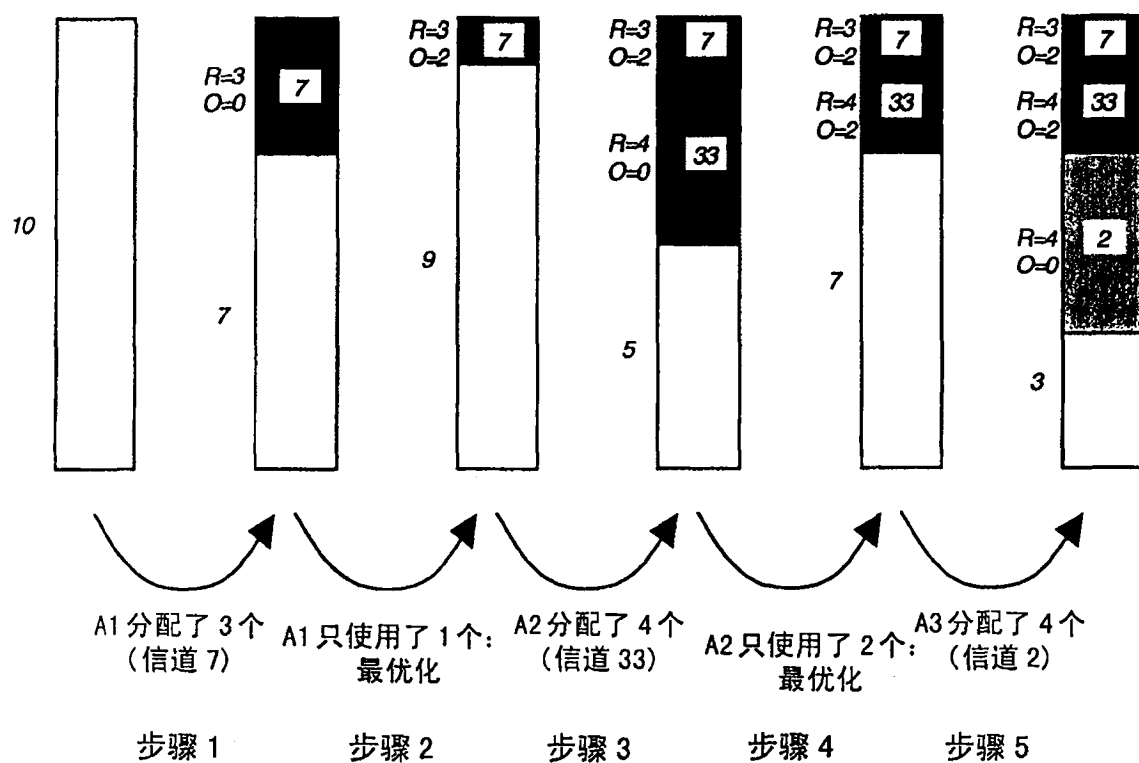


图 5

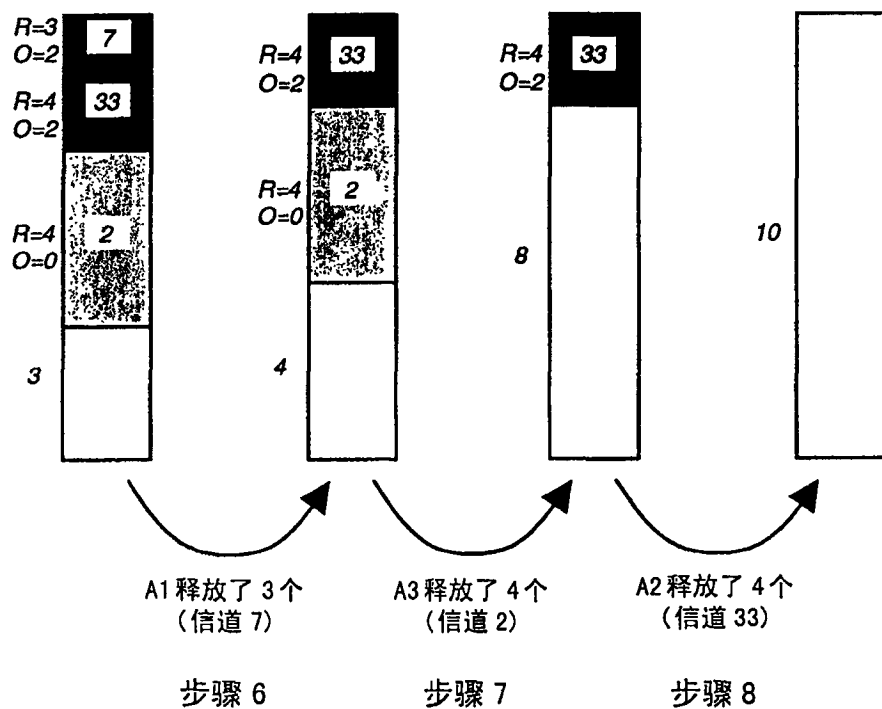


图 6

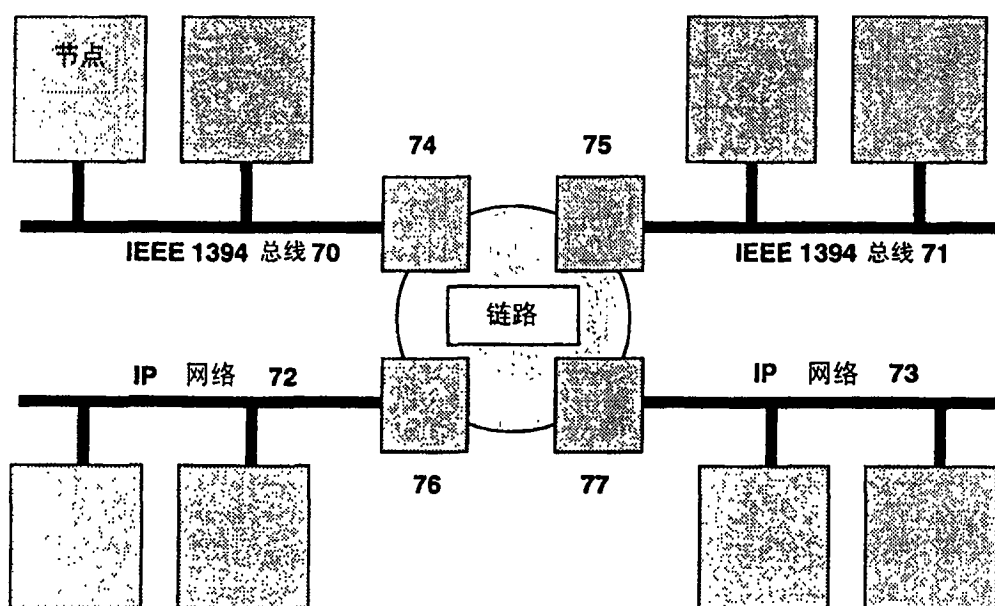


图 7

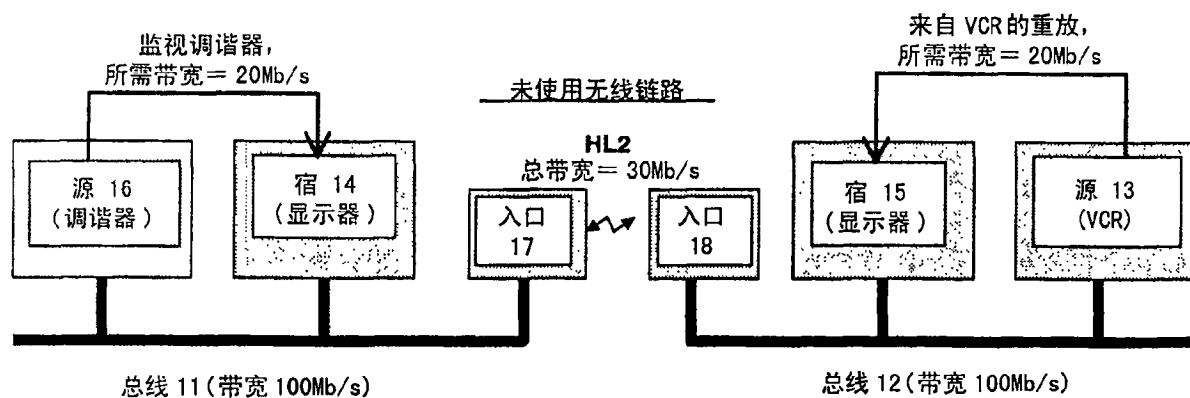


图 8