



# [12] 发明专利申请公开说明书

[21] 申请号 200510088804.2

[43] 公开日 2006年2月1日

[11] 公开号 CN 1728688A

[22] 申请日 2005.7.29

[21] 申请号 200510088804.2

[30] 优先权

[32] 2004.7.29 [33] US [31] 10/901,873

[71] 申请人 卓联半导体股份有限公司

地址 加拿大安大略省

[72] 发明人 张荣峰 涂圣晖 C·巴拉克

余旭亮

[74] 专利代理机构 北京纪凯知识产权代理有限公司  
代理人 沙捷

权利要求书6页 说明书19页 附图9页

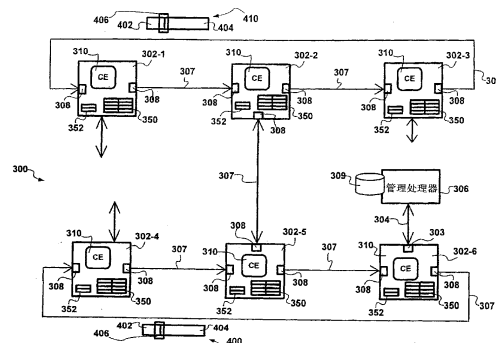
## [54] 发明名称

交换节点堆叠中的交换节点的远程控制

## [57] 摘要

本发明提供了一种在堆叠中的交换网络节点通过带内消息进行远程管理的方法和装置。在启动、重新启动、复位时，堆叠中的交换节点缺省为保留的交换节点标识符，堆叠端口缺省设置为阻塞状态。每个通过阻塞状态接收到的命令帧都被转发到每个交换节点的控制引擎，并且用当前交换节点标识符确认。每个载有保留网络节点标识符的确认帧都会触发确认交换节点的配置。交换节点和管理处理器跟踪关于事件的中断状态向量。中断确认过程用来跟踪生成的中断。交换节点的配置通过由管理处理器发送并发往交换节点相关的控制引擎的命令帧来执行。通过将控制帧发往管理处理器所连接的交换节点和其管理端口来请求管理处理器提供服务。本发明的优点得自交换节点的配置，其中使用小于堆叠中交换节点数目的适当数目

的管理处理器来根据处理、控制、配置带宽要求向堆叠中相应的交换节点提供服务。堆叠中交换节点的带内配置和控制减少了配置、构造、管理以及维护的成本。



- 1、在交换网络节点堆叠中参与的多个交换网络节点中的一个交换网络节点包括：
- 5 a. 至少一个堆叠端口，用来与所述堆叠中至少一个其它交换节点交换帧；
- b. 交换节点标识寄存器，用来指定所述交换节点的交换节点标识；
- c. 管理节点标识寄存器，用来指定所述堆叠中所述多个交换节点中联接管理处理器的交换节点的交换节点标识符；以及
- 10 d. 转发列表，为每个所述堆叠中的交换节点指定向所述交换节点转发帧所经过的相应堆叠端口。
- 2、根据权利要求 1 所述的交换节点，其中，所述堆叠具有单环、多环、星形拓扑结构的一种。
- 15 3、根据权利要求 1 所述的交换节点，其中每一个堆叠端口都具有一个状态指定寄存器，用来将所述堆叠端口设置成为转发状态和阻塞状态之一。
- 20 4、根据权利要求 1 所述的交换节点，进一步包括分类引擎，其选择性地区分控制帧和数据帧。
- 5、根据权利要求 1 所述的交换节点，进一步具有与管理处理器连接的管理端口。
- 25 6、根据权利要求 1 所述的交换节点，进一步包括：
- a. 多个本地外部端口，用于接收要被所述堆叠中的所述多个交换节点处理的数据帧；
- b. 交换数据库，用于存储关于经由哪个外部端口能够到达哪个网络地址的交换信息，该网络端口是所述堆叠中多个交换节点的多个外部端
- 30 口中的一个；

- c. 搜索引擎, 用于选择性地确定经由所述多个本地外部端口中的哪一个转发数据帧; 以及
- d. 命令引擎, 如果数据帧是从针对没有通过所述搜索引擎确定的外部端口的帧以及从针对与所述堆叠中与另一个交换节点相关的确定的外部端口的帧中选择的帧中选择的帧, 则其用于在所述数据帧中选择性地嵌入一个帧堆叠标签 (FTAG), 以用于在堆叠中转发所述数据帧。
- 5
- 7、根据权利要求 1 所述的交换节点, 进一步包括:
- a. 至少一个模块, 其用来执行与在所述交换节点处处理帧相关的动作, 该动作包括发起一个中断和生成一个响应之一; 以及
- 10 b. 命令引擎, 用来生成一个要发送到管理处理器来响应中断的出现的控制帧, 用于执行封装在接收到的控制帧中的命令, 并且用于确认一个接收到的控制帧。
- 15 8、根据权利要求 7 所述的交换节点, 其中, 所述模块进一步包括本地中断向量, 来跟踪与所述模块相关的多个中断事件。
- 9、根据权利要求 7 所述的交换节点, 进一步包括一个命令中断状态向量, 用于跟踪由所述交换节点生成的多个中断事件。
- 20 10、一种用于远程管理堆叠中多个交换网络节点的管理处理器, 包括:
- a. 识别接收到的控制帧的装置;
- b. 确认所述接收到的控制帧的装置;
- c. 跟踪交换节点状态的储存库;
- 25 d. 封装生成控制帧的命令的装置; 以及
- e. 用来转发生成的控制帧的传输装置。
- 11、根据权利要求 10 所述的管理处理器, 进一步包括:
- a. 接收封装在数据帧中的地址解析请求的装置;
- 30 b. 网络节点地址解析装置; 以及 c. 发出封装在数据帧中的地址解析响应的装置。

- 12、一种在交换网络节点的堆叠中发现多个交换网络节点的方法，所述堆叠中每个交换网络节点具有指定了一个保留标识值的交换节点标识符，和至少一个堆叠端口，其被配置以在阻塞状态中转发所有命令帧到命令引擎以及根据在转发状态中的转发列表所指定的交换信息转发命令帧，所述方法包括交换节点的步骤有：
- a. 经由配置在所述阻塞状态中的堆叠端口，接收一个发往具有所述保留识别值的交换节点的命令帧；
  - b. 转发所述命令帧到所述命令引擎；
  - c. 确认具有所述交换节点标识符的命令帧；
  - 10 d. 将所述交换节点标识符设置成在封装命令的接收的命令帧中所指定的新值，该命令指定将交换节点标识符改变为不同于所述保留值的值；以及
  - e. 根据转发列表中的交换信息将后续的命令帧转发到具有保留标识值的交换节点，该帧经由转发状态中配置的堆叠端口接收。
- 15
- 13、一种在交换网络节点的堆叠中发现多个交换网络节点的方法，所述堆叠中的每个交换网络节点具有指定了保留标识值的交换节点标识符，以及至少一个堆叠端口，其被配置以在阻塞状态中转发所有命令帧到命令引擎以及根据在转发状态中的转发列表所指定的交换信息转发命令帧，该方法包括管理处理器的循环步骤有：
- a. 发送指定具有保留标识值的目的地交换节点的命令帧；
  - b. 接收来自所述交换节点的确认，其指定来自新发现的交换节点的保留标识值；
  - c. 将所述新发现的交换节点的交换节点标识符配置为与保留标识值不同的唯一标识值；
  - 20 d. 从所述新发现并配置的交换节点取得堆叠端口标识符；
  - e. 选择一个堆叠端口；以及
  - f. 设置所选择的堆叠端口为转发状态。
- 25
- 30 14、根据权利要求 13 所述的发现方法，进一步包括配置所述堆叠中新发现的交换节点的转发列表。

15、根据权利要求 13 所述的发现方法，进一步包括配置所述堆叠中新发现的交换节点的交换数据库。

16、一种向交换网络节点堆叠中的交换网络节点提供管理处理器服务的方法，该方法包括：

- a. 将关于所提供的服务的数据封装到一个帧中；
- b. 将帧堆叠标签（FTAG）与该帧相关联；
- c. 将所述交换节点的交换节点标识符写入 FTAG 的目的地交换节点字段；以及
- 10 d. 将载有 FTAG 的帧向所述交换节点发送。

17、根据权利要求 16 所述的方法，其中所述帧是一个控制帧，该方法进一步包括：

- a. 将为控制帧保留的分类动作值写到 FTAG 的分类动作字段；
- 15 b. 将为所述交换节点处的控制引擎保留的目的地端口值写到 FTAG 的目的地端口字段。

18、根据权利要求 16 所述的方法，其中，所述帧为路由数据帧，该方法进一步包括：

- 20 a. 解析所述数据帧的目的地地址到所述堆叠的一个外部发出目的地端口；
- b. 将为控制帧保留的分类动作值写到 FTAG 的分类动作字段；以及
- c. 将解析的目的地端口的目的地端口标识符写到 FTAG 的目的地端口字段。

25

19、一种在交换网络节点堆叠的交换网络节点处理帧的方法，包括：

- a. 如果接收到的帧载有与为控制帧保留的分类动作值不同的分类动作值，则基于与交换节点相关的交换数据库所存储的交换信息选择性地转发接收到的帧；
- 30 b. 如果接收到的帧载有为控制帧保留的分类动作值，则基于与交换节点相关存储的转发列表所存储的交换信息选择性地转发接收到的

帧；

c. 对每个请求管理处理器服务的帧，选择性地将帧的分类动作值改变成为控制帧保留的分类动作值。

5 20、根据权利要求 19 所述的方法，其中将帧基于保持在交换数据库中的交换信息进行转发，该方法进一步包括：

a. 修改所述帧相关的目的地交换节点字段，来指定所述堆叠中的一个解析的目的地交换节点；以及

10 b. 修改所述帧相关的目的地端口字段，来指定解析的目的地交换节点的解析的外部发出端口。

21、根据权利要求 19 所述的方法，其中将接收的帧基于转发列表中存储的交换信息进行转发，该方法进一步包括：

15 a. 修改所述帧相关的目的地交换节点字段，来指定所述堆叠中连接管理处理器的交换节点；以及

b. 修改所述帧相关的目的地端口字段，来指定在目的地交换节点处，管理处理器所连接的管理端口。

22、根据权利要求 19 所述的方法，进一步包括：

20 a. 接收一个帧，其载有为控制帧保留的分类动作值、并且在与所述帧相关的目的地端口字段载有外部发出端口的标识符；以及

b. 利用在帧头中指定的目的地网络地址和指定的外部发出端口标识符之间的联系更新所述交换数据库。

25 23、根据权利要求 19 所述的方法，进一步包括：

a. 接收载有为控制帧保留的分类动作值、本地交换节点的标识符，以及为命令引擎保留的目的地端口标识符的帧；

b. 执行所述帧中指定的命令；以及

c. 确认该帧的接收。

30

24、根据权利要求 23 所述的方法，进一步包括将执行命令所得到的应

答封装到确认帧中。

- 25、根据权利要求 19 所述的方法，进一步包括如果在帧中所指定的目的地地址不能根据交换数据库中指定的交换信息解析，则确定该帧请求管理处理器服务。
- 5

## 交换节点堆叠中的交换节点的远程控制

### 5 技术领域

本发明涉及在通信网络中的交换网络节点的控制，尤其涉及控制堆叠中多个交换网络节点的方法，其中堆叠中仅有少数几个交换网络节点中具有附加其上的管理处理器。

### 10 背景技术

在分组交换通信领域中，交换网络节点是用来指导分组业务到达适合的网络目的地的。交换网络节点可能在受管理的模式下运行，在此种模式中，该交换节点应用一管理处理器提供的服务，或者可能在自由模式下运行，在此种模式中该交换节点独立运行。两种运行模式都要求如管理处理器所提供的管理模式，例如对交换节点的学习功能，不过在自由模式下不承受相关的管理开销。在正常的运行中，传输到和来自管理处理器的主要资源包括：不能通过硬件分类分析的数据分组或者需要特定软件支持的分组；包含统计计数器翻转的硬件触发中断；媒体接入控制（MAC）地址查询表的插入或删除；或者关于资源可用性的通知。成本涉及管理处理器本身的开发、实现、配置和管理。

在其他文章中描述的各种相关的解决方法针对于与管理处理器的优化和管理开销的减少有关的问题。

交换网络节点的一种配置的情况包括将一般协同定位的多个交换网络节点堆叠在一个设备托架（equipment shelf）上。

交换机堆叠是全体代表一个单独的逻辑交换网络节点的来提供更高的总吞吐量的一组交换网络节点的结构。例如，假设单一的网络交换机 102 包括 24 个快速以太网端口和 4 吉比特以太网端口。虽然网络交换机 102 可以支持 6.4 Gbps，如图 1 所示，将多个这样的网络交换机 102 级联可以增加总的系统吞吐量。如图 1a) 所示的堆叠结构 100 通过将 3 个交换节点 102 配置成环形结构 104 可以传递 13.2Gbps 的总吞吐量。如图 1b) 所示的堆叠结构 110 通过将 6 个交换节点 102 配置成



双环结构（104）可以传递 22.4Gbps 的总吞吐量。如图 1c）所示的堆叠结构 120 将 3 个交换节点 102 配置成星形结构可以传递 17.6Gbps 的总吞吐量。

5 虽然总吞吐量的增加使堆叠配置很理想，但是其面临着在这样的堆叠中如何配置和控制交换节点 102 的难题。

图 2 所示是现有技术管理下的交换节点配置。如图 2a）所示的配置 200，每个交换节点 102 有一个单独的管理处理器 204 来对其进行控制。虽然这是一个简单的方法但同时也是昂贵的。如图 2b）所示的配置 210 中，整个交换节点堆叠通过一个管理处理器 206 控制。依照这种方  
10 种方法，上述的管理处理器 206 能够控制和配置由堆叠中的交换节点 102 所定义的单域。管理处理器 206 通过单独的控制平面（control plane）208 发送信号或消息给堆叠中所有的交换节点 102，通常像共享介质那样实现。如图 2a）和图 2b）所显示的，每一个交换节点 102 保留了一个专用端口来保留管理处理器 204/206 的服务，并且共享的管理处理器  
15 配置 210 将承担配置、构造、管理和维护共享介质 208 所需的成本。

因此，在交换节点堆叠中提供交换节点的控制和结构时需要解决上述问题。

## 发明内容

20 根据本发明的一个方面，提供了交换网络节点堆叠中参与的多个交换网络节点中的一个交换网络节点。该交换节点包括：至少一个用于与堆叠中至少另一个交换节点交换帧的堆叠端口，用来指定交换节点的交换节点标识的交换节点标识寄存器，用来指定堆叠中的多个节点中连接有管理处理器的交换节点的交换节点标识符的管理节点标识  
25 寄存器，以及用来为堆叠中每个节点指定转发帧到上述交换节点所要通过的相应的堆叠端口的转发列表。

根据本发明的另一方面，提供了一种用于远程管理堆叠中的多个交换网络节点的管理处理器。该管理处理器包括：识别接收到的控制帧的装置；确认收到的控制帧的装置；用来跟踪交换节点状态的储存库；封装生成控制帧的命令的装置；以及将生成的控制帧转发的传输  
30 装置。

根据本发明的另一个方面，提供了一种在交换网络节点堆叠中发现多个交换网络节点的方法，其中，堆叠中的每个交换网络节点都有指定了保留的标识值的交换节点标识符，并且至少配置一个堆叠端口在阻塞状态时转发所有的命令帧给命令引擎，在转发状态中根据转发列表中指定的交换信息转发命令帧。由交换节点执行的该方法的步骤包括：通过一个在阻塞状态中设置的堆叠端口接收去往具有该保留的标识值的交换节点命令帧；转发该命令帧到命令引擎；确认带有交换节点标识符的命令帧；将交换节点标识符设置为在接收到的命令帧中指定的新的值，该命令帧封装了指定改变交换节点标识符为不同于保留值的值的命令；依据转发列表中指定的交换信息，将后续的命令帧转发到具有经由堆叠端口接收的保留的标识值的交换节点，该端口被配置成转发状态。

根据本发明的另一个方面，提供了一种在交换网络节点堆叠中发现多个交换网络节点的方法，其中，堆叠中的每个交换网络节点都具有指定了一个保留的标识值的交换节点标识符，并且至少配置一个堆叠端口以在阻塞状态时转发所有的命令帧给命令引擎，并根据在转发状态中转发列表中指定的交换信息转发命令帧。由管理处理器执行的该方法的循环步骤包括：传送指定了具有保留标识值的目的地交换节点命令帧；接收指定了来自新发现的交换节点的保留标识值的交换节点的确认；配置新发现的交换节点的交换节点标识符为一个与保留的标识值不同的唯一标识值；从新发现的配置的交换节点查询堆叠端口标识符；选择一个堆叠端口；并将被选择的堆叠端口设置为转发状态。

根据本发明的另一个方面，提供了一种方法来向交换网络节点堆叠中的一个交换网络节点提供管理处理器服务。该方法包括：封装关于所提供的服务的数据到一个帧中；将帧堆叠标签（FTAG）与该帧相关联；将交换节点的交换节点标识符写入 FTAG 的目的地交换节点字段；将带有 FTAG 的帧向该交换节点传输。

根据本发明的另一个方面，提供了一种在交换网络节点堆叠中处理交换网络节点处的帧的方法。该方法包括：如果接收的帧具有的分类动作值不同于为控制帧保留的分类动作值，那么就根据储存在与该

交换节点相关的交换数据库中的交换信息有选择地转发接收到的帧；  
如果接收的帧具有的分类动作值是为控制帧保留的，那么就根据储存在  
与该交换节点相关的转发列表的交换信息有选择地转发接收到的  
帧；以及有选择地将帧的分类动作值改变成为请求管理处理器服务的  
5 每个帧所用的控制帧保留的分类动作值。

本发明的优点来源于网络节点的工程配置，其中根据处理、控制、  
配置带宽的要求，应用了少于堆叠中交换节点数量的适当数量的管理  
处理器来向堆叠中的相应交换节点提供服务。堆叠中交换节点的带内  
配置和控制减少了配置、构造、管理和维护的开销。

10

### 附图说明

本发明的特点和优越性将在下文参照附图对示例性实施方式的详  
细描述中变得更加明显。

图 1 的 a、b 和 c 是示出提供了增加的总吞吐量的示例性交换节点  
15 堆叠配置的示意图；

图 2 的 a 和 b 是说明现有技术的管理交换节点堆叠配置的示意图；

图 3 是示出依照本发明的示例性实施方式在交换节点堆叠中实现  
远程交换节点控制的元件的示意图；

图 4 是流程图，示出了依照本发明的示例性实施方式的管理处理  
20 器在交换节点堆叠中实现远程控制和配置所执行的处理步骤；

图 5 是流程图，示出了依照本发明的示例性实施方式的在交换节  
点堆叠中由管理处理器远程控制和配置的交换节点所执行的处理步  
骤；

图 6 是依照本发明的示例性实施方式示出的中断确认过程的步骤  
25 的处理图；

图 7 是依照本发明的示例性实施方式示出的堆叠初始化过程的处  
理图；

图 8 的 a 和 b 是示出具有环状结构和发现的控制帧环的堆叠中交  
换网络节点初始化的示意图；

图 9 是依照本发明的示例性实施方式示出控制帧环检测的示意图；  
30 以及

图 10 是依照本发明的示例性实施方式示出的一个堆叠中多个控制和配置域的示意图，其中每个域都有相关的管理处理器。

应该注意的是在附图中类似的特征具有类似的标记。

## 5 具体实施方式

参照图 3，依照本发明的示例性实施方式，堆叠 300 中的网络交换机 302 由一台连接 304 到其中一个交换网络节点 302 的管理处理器 306 配置和控制，而不为堆叠 300 中的每个交换网络节点 302 配置单独的管理处理器。本文提供的方法是涉及如堆叠 300 中的交换节点 302 的远程配置和控制的方法。

堆叠结构中远程结构及控制配置的示例性元件包括相互连接交换节点 302 间的链路，参照连接堆叠端口 308 的堆叠链路 307。

依照本发明的示例性实施方式，通过堆叠端口 308 发送的每一帧 400 都包括具有如下示例性格式的帧堆叠标签 (FTAG) 406：

15

字段	字段说明
分类器动作	0- 普通 L2/L3 搜索。 1- 基于流的列表查询。使用流字段。 2- 发往管理处理器的分组，或者来自管理处理器有单一目的地的分组。应用目的地交换节点字段确定目的地交换节点是否是远程的。如果是远程的，就应用目的地交换节点/堆叠端口列表来确定一个发出端口。否则，根据一组 ID 字段转发到本地端口或本地命令引擎。 3- 来自管理处理器的有多个目的地的分组。应用一组 ID 字段确定一个或多个发出端口。
目的地 或 流 ID 或 组 ID	分组所要到达的交换节点 ID 和端口 ID 或 流 ID 或 多播组 ID。

字段	字段说明
原始来源	分组所来自的堆叠中的交换节点 ID 和端口 ID。
流控制允许	表示分组的原始来源端口是允许流控制的。
传送优先级	发送分组的优先级，用于排队和调度。
丢弃优先级	丢弃分组的优先级，用于排队之前的 WRED。
使用优先级比特	表示 FTAG 中储存的优先级比特取代其他任何方法确定分组的优先级
VLAN	两字节的标签，包括使用优先级比特、CFI 以及 VLAN ID
VLAN 标签	表示分组包括一个 VLAN 标签头
取代 VLAN 动作	表示在本分组中没有执行 VLAN 动作（插入、剥离（stripping）、置换）
多点传播	表示该分组是否是多播分组
FTAG 签名	这些比特被典型地设置为“FF”

FTAG 406 的大小与该实现的开销相互关联，所应用的 FTAG406 的大小留在设计时选择，其对于每一种配置都是在开销和所支持的功能之间的一种平衡。

5 每一个 FTAG406 都包含关于经处理的帧的信息，例如，包括该帧是单播还是多播，发送优先级，VLAN 联系（VLAN association），始发“来源”——帧 400 第一次被接收到所通过的交换节点 302 以及外部端口，等等。如下面将要更具体地描述的，分类器动作字段确定每一个帧 400 应该被堆叠 300 中的每个交换节点 302 转发到哪里。

10 依照本发明的示例性实施方式，虽然大多数通过堆叠端口 308 传输的帧都确实包含 FTAG 406，但是 MAC 流控制帧是一个例外。因此，为了区别包含 FTAG 406 的帧和少数不包含的，FTAG 签名是必需的。这样的 FTAG 签名示范性地识别跟在前同步码（preamble）后面的首几个比特/字节作为 FTAG，而不是实际的帧数据 404。FTAG 406 和 FTAG  
15 签名可以与帧头 402 相联系，但这并不作为对本发明的限制。

依照本发明的示例性实施方式，堆叠 300 中的每一个交换节点 302 都包含一个命令引擎 310。命令引擎 310 的示例性功能包括：解码从管

- 理处理器 306 接收到的命令帧 410，执行其中包含的命令，和生成一个确认和/或响应。如果管理处理器 306 是“远程的，”也就是说，不是直接附加在从属交换节点 302 上的，那么，命令引擎 310 用作管理处理器 306 的本地代理，配置交换网络节点 302 的逻辑并且响应已经实现的远程控制。另外，当需要服务和/或支持的时候，为了中断远程的管理处理器 306，控制引擎 310 能够初始化对外的命令帧 410。下表显示了示例性命令帧 410 的示例性格式：

字段	说明
目的地 MAC 地址	命令帧的目的地 MAC 地址
源 MAC 地址	命令帧的来源 MAC 地址
以太型	总是设置成 A0A0 来表示是一个命令帧
序号	通过监视丢失的命令帧生成一条用于远程控制的可靠信道
控制帧信息	如果命令帧是一个 CPU 的中断，就显示所要中断的线程的 ID。 管理处理器或命令引擎。如果是 0，就代表管理处理器发布的此帧，如果是 1，就代表命令引擎发布的此帧。 设备 ID：对于由管理处理器发出的帧，显示帧的目的地的设备（交换网络节点/端口的 ID）。 对于要传向管理处理器的帧，显示帧的来源的设备（交换节点/端口）。
命令操作码	如果该帧是一个响应，设置该比特为 1，表示原命令是无效的或未执行的。 否则，表示命令/响应的类型
数据	命令/响应的内容

- 命令帧 410 的操作码用来辨识被编码在帧数据 404 中的命令或响应的类型。下面列出命令操作编码的样例，尽管在实际应用中用到的远程命令/请求的种类事实上是不限制的。

操作码	定义
000001	来自管理处理器的读存储器请求，或响应。
000010	来自管理处理器的写存储器请求，或确认。
000011	来自管理处理器的写寄存器请求，或确认。
000100	来自管理处理器的读寄存器请求，或响应。
000101	来自管理处理器的将 MAC 地址加入列表的请求，或确认
000110	来自管理处理器的查询列表中的 MAC 地址的请求，或响应
000111	来自管理处理器的将 MAC 地址从列表删除的请求，或确认。
001000	命令引擎通知管理处理器发生统计计数器翻转
001001	命令引擎通知管理处理器队列占用
001010	命令引擎通知管理处理器连接失败/失败转移(fail-over)?
001011	命令引擎通知管理处理器 MAC 地址已经加入到列表中
001101	命令引擎通知管理处理器 MAC 地址已经从列表中删除

命令帧 410 的实际的帧数据 404 内容根据命令帧 410 的操作码改变。例如，操作码是“000011”（寄存器写请求）的命令帧 410 的内容包括将要写入的寄存器的地址，以及相关的值。相反地，操作码是“001000”（统计计数器翻转）的命令帧的内容包括生成该命令帧 410 的交换节点 302 中所有的硬件统计计数器的位图，只有在那些计数器发生翻转的比特位置为逻辑高电平“1”。

依照本发明的示范性实施方式，堆叠 300 中的每个交换节点 302 包括一个转发列表 350，下面描述的例子针对堆叠 300 中的交换节点 302-2:

目的地交换机 302	堆叠端口 308
1	B
2	X (自己)
3	B
4	D
5	D
6	D

在堆叠 300 中的每一个交换节点 302, 本地转发列表 350 存储了堆叠端口的 ID, 帧 400/410 必须通过该端口转发来到达堆叠 300 中的另一个交换节点 302。

5 依照本发明的示范性实施方式, 除了转发列表 350 外, 堆叠 300 中的每一个交换节点 302 还存储它自己的唯一 ID, 以及它的控制管理处理器 306 直接连接的交换节点 302 的 ID。此信息通常被存储于本地寄存器 352。

依照本发明的示范性实施方式, 上述的因素协同工作使带内控制帧能够在受远程控制的交换节点堆叠 300 中传送。

10 根据第一示例性情况, 管理处理器 306 生成一个针对远程交换节点 302 的命令。

参照图 4, 管理处理器 306 封装 422 该命令到一个命令帧 410, 包括 FTAG 406。设置 424 该 FTAG 406 的“分类器动作”字段, 例如设为“2”, 同时将“目的地交换节点”字段设置 426 成命令要发送到的交换节点 302 的交换节点 ID。“目的地端口”字段被设置 428 成被保留来表示要发送到的目的地交换节点 302 的命令引擎 310 的值。根据本发明的示范性实施方式, 如果堆叠 300 中没有包含多于 29 个物理端口的交换节点 302, 那么特定值“31”可以用在“目的地端口”字段来表示控制帧 410 要到达的命令引擎 310。并且, 命令帧 410 被发送 430 到要  
20 目的交换节点 302。

参照图 5, 当命令帧 410 穿过一个堆叠 300 时, 在沿途的每一个交换节点 302, 堆叠端口 308 的接口上或其附近的交换节点硬件, 媒体接入控制 (MAC) 模块, 通过其 FTAG 签名识别 432 命令帧 410。MAC 模块通过一个交换节点 302 的分类器抽取 434 FTAG 406 以便使用。

25 搜索及分类引擎观测 436 在 FTAG 406 中指定的“分类器动作”字段被设置为“2”, 这表示 FTAG 406 指定的控制帧的目的地将取代数据库搜索或分类的结果。

如果交换节点 302 的搜索及分类引擎检测 438 到 FTAG 306 中“目的地交换节点”等于作决定的交换节点 302 的交换节点 ID, 那么命令帧 410 必须传送至一个本地端口或者本地命令引擎 310。如果“目的地端口”字段被设置 440 成命令引擎 310 的保留值, 那么搜索及分类引  
30



引擎就会转发 442 命令帧 410 给本地命令引擎 310。

一旦目的地命令引擎 310 接收到了命令帧 410，则该命令引擎 310 就会通过读取和解释操作码来解码该命令并且执行 444 该命令。执行 444 一个命令一般涉及与交换节点 302 的其他硬件模块的相互作用，要  
5 被读取或者更新的相关寄存器或记录可以位于此节点。

如果搜索及分类引擎检测到 438 FTAG 406 中的“目的地交换节点”不等于从属交换节点 ID，那么该命令帧 410 一定是发送到堆叠 300 中的另一个交换节点 302 的。在这种情况下，搜索及分类引擎使用指定的目的地交换节点 ID 作为索引参考 446 转发列表 350 来确定对控制帧  
10 410 正确的发出堆叠端口，并且通过确定的堆叠端口 308 转发 448 该控制帧 410。

根据第二种情况，命令引擎响应来自远程管理处理器 306 的命令。

当命令引擎 310 完成了一条命令的执行后，该命令引擎 310 可以确认完成，并且在一般情况下，但不是必需地，以相关信息进行响应。  
15 在确认/响应中，命令引擎 310 封装 450 一个响应/确认到命令帧 410 中，其中包括 FTAG 406。该 FTAG 406 的“分类器动作”字段被设置 452 为“2”，并且“目的地交换节点”字段被设置 454 成控制管理处理器 306 直接附加的交换节点 302。“目的地端口”字段被设置 456 成为管理处理器端口的 ID。命令引擎 310 写入 (450) 响应帧 410 的内容，例  
20 如从一个或多个寄存器中读出的数据。

根据本发明的示例性实施方式，在交换节点 302 处理控制帧 410 的相关开销通过以下方式响应/确认由管理处理器 306 发送的控制帧 410 而减少：通过在响应控制帧 410 已经生成 450 时，用相应的响应控制帧 410 覆盖接收到的已经存储到交换节点 302 中的请求控制帧 410，  
25 同时假设每一个响应控制帧 410 最大不超过接收到的请求控制帧 410。

然后，确认/响应控制帧 410 经由在转发列表 350 中指定的堆叠端口 308 发送 458。如果该确认/响应控制帧 410 的目的地是本地交换节点 302，那么可知该确认/响应控制帧 410 将通过回环堆叠端口发送 458，然后过程从步骤 432 继续。

30 如果响应控制帧的“目的地交换节点”说明指定了 438 本地交换节点 ID，其中，本地交换节点 302 生成了确认/响应控制帧 410，并且

“目的地端口”是 460 管理处理器 306 所连接的端口 303，那么命令引擎 310 就会发送 462 该控制帧 410 到该本地管理处理器端口 303。

如果响应帧的“目的地交换节点”说明不是指定 438 本地交换节点 ID，那么控制管理处理器 306 就连接到堆叠 300 中的另一交换节点 5 302。在这种情况下，命令引擎 310 参考 446 转发列表 350 来确定正确的发出堆叠端口 308，来转发 448 响应控制帧 410 到控制管理处理器 306 直接连接的交换网络节点 302。

如上面关于第一种情况的描述，该响应控制帧 410 继续通过堆叠 300，一个交换节点 302 接着一个交换节点 302 地被转发 448，直到找到 10 正确的目的地交换节点 302。一旦到了正确的 438 目的地交换节点 302，本地控制引擎 310 就会检测到 460 FTAG 406 中指定的“目的地端口”就是该管理处理器端口 303，并且确认/响应控制帧 410 将被转发 462 到管理处理器 306。

根据第三种情况，命令引擎 310 发送一个中断到远程管理处理器 15 306。

当交换节点 302 的任何硬件模块请求一个管理处理器中断时，本地命令引擎 310 将中断信息编码 466 到一个具有 FTAG 406 的命令帧 410。FTAG 406 的“分类器动作”字段设置 452 为“2”，“目的地交换节点”字段设置 454 为控制管理处理器 306 直接连接的交换节点 302 20 的交换节点 ID。“目的地端口”字段也被设置 456 成管理处理器端口 303。

命令帧 410 的转发如上所述关于第二种情况进行。

根据第四种情况，搜索/分类引擎重新引导一个帧到管理处理器 306。

在某些情况下，搜索/分类引擎可以通过将 FTAG 406 包括到 472 从属帧 (subject frame) 400 中来重新引导 470 一个帧 400 到管理处理器 306。例如，如果第 3 层帧 400 的目的地 IP 地址在本地交换节点 302 的数据库里没有找到，那么交换节点 302 可以转发 458 该帧 400 到提供路由功能的管理处理器 306。另外，其它一些特定类型的帧 400，例如 30 如网桥协议数据单元 (BPDU) 或者生长树协议 (STP) 帧 400，可以被本地交换节点 302 的搜索/分类引擎捕获并且以特定处理转发到管理

处理器 306。一旦 FTAG 406 “附加” 472 到这样的帧 400 上，FTAG 406 的“分类器动作”字段就被设置 452 成“2”，“目的地交换节点”字段被设置 454 成控制管理处理器 306 直接连接的交换节点 302。“目的地端口”字段也被设置 456 成管理处理器端口 303。修改后的帧 400 的转发 458 如上面关于第二种情况的描述进行。

根据第五种情况，管理处理器 306 发送帧 400 到远程交换节点 302 以通过端口转发。

参照图 4，在大多数情况下，非命令帧 400 从管理处理器 306 发送是在管理处理器 306 解析了 478 一个帧的 IP 地址之后，也就是，如上面关于第四种情况的描述，响应于转发 458 到管理处理器 306 的帧 400。在这种情况下，管理处理器 306 在发送该帧 400 到远程交换节点 302 前修改 480 接收到 476 的帧 400 的 FTAG 406 以及有效载荷。FTAG 406 的“分类器动作”字段设置 482 为“2”，并且“目的地交换节点”以及“目的地端口”字段根据解析的 IP 地址被设置 484/486 成源交换节点的 ID 以及发出端口 ID。

帧 400 的转发如参考第一种情况的以上描述来进行。

当帧最终到达目的地交换节点 302 时，本地的搜索/分类引擎识别 432 并重新引导 490 帧 400 发往指定发出端口。在此过程中，搜索/分类引擎学习 492 新的目的地 IP 地址/发出端口的关联。

因此上面描述的五种情况说明了带内配置及控制帧的转发。

帧通过外部端口在交换节点 302 被接收，并且被提供 501 FTAG。

根据第六种情况，帧 400 通过执行在每一跳的第二层或第三层数据库搜索，在堆叠 300 的堆叠端口 308 之间进行转发 458。这种帧 400 的 FTAG 406 中的“分类器动作”字段被设置 500 成“0”。作为选择，第一跳的交换节点 302 在分类并存储流 ID 于 FTAG 406 中后，可以分配 504 给该帧 400 一个流 ID。通过设置 506 “分类器动作”为“1”，搜索/分类引擎可以迫使 508 后续的跳跃利用前面分配的流 ID 来取代本地数据库搜索的结果。

因此上面六种示例性情况详细说明了帧在堆叠 300 中的转发。

根据本发明的示范性实施方式，提供了中断确认协议。

当单独的管理处理器 306 通过带内消息控制堆叠 300 中的多个交

换网络节点 302 时，带内通信信道的可靠性成为了一个关键性问题。

参照对第 1、2 种情况的描述，命令引擎 310 确认接收到的每一个命令帧 410，这样管理处理器 306 可以监测发出的命令是否实际到达了它们的目的地，并且是否被执行。如果管理处理器 306 没有接收到确认，那么必然会认为命令帧 410 丢失，因此会重传命令帧 410。

有可能的是命令确实被接收并执行，但是确认本身丢失或者延时。这种情况使得管理处理器 306 发送多次相同的命令，从而使得远程交换节点 302 意外地将这个没有确认的命令执行多次。

基于本文描述的示例性实施方式的实现应该尽可能确保管理处理器 306 发送的命令如果被执行了两次是无害的。

有利的是，大多数用于控制和配置堆叠 300 中的交换网络节点 302 所需的命令是无害的，这样的例子包括读/写寄存器，或者交换数据库更新。

然而，回想上文的描述，一旦请求了管理处理器 306 中断，命令引擎 310 就会生成一个命令帧 410。为了确保可靠的带内通道，命令引擎 310 通过监测确认（410）来确定是否每一个中断消息都实际地到达了管理处理器 306。如上所述，命令引擎 310 可能因为来自管理处理器 306 的确认的丢失或者延时错误地发送同一个中断两次，从而引起管理处理器 306 误解命令引擎的意图。设想管理处理器 306 从交换节点 302-3 接收到了两个时间间隔很短的中断消息，每一个中断指示“端口 7 接收到的总字节”的统计计数器翻转。管理处理器 306 可能错误地认为该计数器实际上翻转了两次，而事实上可能只是因为同一个中断消息发送了两次。

根据本发明的示范性实施方式，通过使用图 6 所示的中断确认协议 600 解决该不明确性。中断确认协议包括三个参与实体：相关的交换节点 302 的中断用户模块 602、本地交换节点 302 上的相关命令引擎 310，以及管理处理器 306。下面的术语将在文中被用来描述一个示例性的中断确认协议的实现：

——中断用户模块 602 是与中断事件初始化有关的从属交换节点的硬件模块。从上面的例子可知，当在 MAC 模块中的统计计数器翻转时，管理处理器 306 将被通知。这个 MAC 模块代表中断用户

模块 602。

——本地中断向量 604：中断用户模块 602 可以经历几种触发管理  
处理器 306 中断的事件，该中断用户模块 602 必须保存指明哪个或  
5 哪些事件被用来触发管理处理器 306 的位图。这个被称作本地中断  
向量 604 的位图被存储在本地从属交换节点 302 内。

——命令中断状态向量 606：命令引擎 310 为受控交换节点 302 的  
每个中断用户模块 602 维护一个中断状态向量 606。客户的中断请  
求是未决的，一直到管理控制器 306 已经确认它被收到。这个未决  
10 状态在 CMD 中断状态向量 606 中反映为逻辑高电平“1”。

——线程：当中断用户模块 602 触发了一个或者多个中断事件时，  
命令引擎 310 就会表述并且发送包含该中断用户模块 620 的中断状  
态的命令帧 410，然后等待来自管理处理器 306 的确认。命令引擎  
310 可以同时服务于多个用户 602；也就是说，在任意给定的时间  
可以有多个这样的命令帧 410，或者它们相应的确认进行传送。这  
15 种多个并行的操作被称为线程。对于每个线程，需要命令引擎 310  
和管理处理器 306 维护小量的状态信息。

——序号：命令帧 410 以及确认（410）通过线程号以及对每个发  
送/确认的命令帧 410 加一的序号进行辨识。序号包含在如上所述  
20 的命令帧 410 头中。要为每个线程维护一个计数器。

相应地，管理处理器 306 通过与 CMD 中断状态向量信息相关联的  
储存库 309 跟踪该信息。

当中断事件 650 继而发生时，中断用户模块 602 设置 652 相应的  
本地中断向量 604 的相应特位。中断触发开始于中断用户模块 602 请  
求访问 654 命令引擎 310。当访问命令引擎被准许 656 后，命令引擎  
25 310 更新 658 与该中断用户模块 602 相关的 CMD 中断状态向量 606。  
依照本发明的示范性实施方式的实现，新的 CMD 中断状态向量 606  
是旧的向量 606 和客户的本地中断向量 604 的按位逻辑或。并且，中  
断用户模块 602 清除 660 其本地中断向量 604。

在任意比特位置包含逻辑高电平“1”的任何 CMD 中断状态向量  
30 606 都请求发送命令帧 410 到管理处理器 306 来代表该中断用户模块  
602。依照本发明的一个示范性实施方式，活动线程的数目代表了在交

换网络节点 302 的受控资源，因此当多个线程中的一个被释放的时候，命令引擎 310 选择一个符合条件的中断用户模块 602 来使用它。例如，为服务器选择中断用户模块 602 的仲裁算法可以遵守循环规则。当选择的线程被占用时，该线程的序号就会增加。

5       命令引擎 310 生成 662 命令帧 410，并且代表中断用户模块 602 发送 664 该命令帧 410。该命令帧 410 包含该中断用户模块 602 所用的 CMD 中断状态向量 606（位图）的当前值。通过线程号和线程的当前序号识别该命令帧 410。在命令引擎 310 等待确认（410）时，该线程仍保持被占用状态。

10       如果接收 670 到一个针对特定线程的确认，但是确认的序号与线程的序列计数器的值不匹配 672，或者该线程已经被释放 674，那么命令引擎 310 就会忽略该确认，从而减小多个确认的有害效果。

      如果接收到针对某个被占用线程的确认（410），并且确认的序号与线程的序列计数器 676 的值匹配，那么该确认是有效的。命令引擎  
15 310 更新 678 与中断用户模块 602 相关联的 CMD 中断状态向量 606。依照本发明的实例性实施方式的实现，新的 CMD 中断状态向量 606 是旧向量和确认（410）中包含的向量的反向（inverse）进行位逻辑与的结果。然后，线程被释放。

      如果在（可编程的）超时阶段 680 后仍没有接收到有效的确认 410，  
20 那么命令引擎 310 重新开始执行如上所述的表示 662 以及发布 664 另一个命令帧 410 的步骤。

      对于每一个线程，管理处理器 306 一直跟踪控制帧 410 中接收 664 到的最后一个序号，以及与该序号相对应的第一个接收到的中断状态向量 606。

25       如果接收 664 到一个针对特定线程的命令帧 410，并且帧的序号与最后接收到的序号不匹配 680，那么该控制帧 410 被视为有效的。管理处理器 306 根据控制帧 410 中指定的中断状态向量 606 的值，或者更明确地说根据其中指定的逻辑高电平比特执行动作 680。

      管理处理器 306 表达 684 并且发送 670 一个确认帧 410。该确认帧  
30 410 包含与相应的命令帧 410 相同的序号，并且有相同的中断状态向量 606 内容。管理处理器 306 记录接收到的序号以及命令帧的内容（有效

载荷在这种情况下是中断状态向量 606)。

如果接收到一个针对某个线程的命令帧 410，并且该帧的序号与最后接收到的序号匹配 679，那么这个控制帧 410 是无效的，它是一个由于第一次触发中断的确认 670 没有到达从属交换节点 302 而重复发送的中断。管理处理器 306 不会关于该命令帧 410 中的向量内容执行任何动作。管理处理器 306 表达 684 并且发送 670 (另) 一个确认帧 410。该确认包含与命令帧相同的序号以及先前存储的内容而不是当前无效命令帧 410 的内容。管理处理器 306 不记录接收到的序号和内容。

在上面的描述中，管理处理器 306 和堆叠 300 中的远程控制交换节点 302 之间的相互作用假设了堆叠 300 中的每一个 302 都知道自己的 ID、管理处理器 306 连接的交换节点 302 的 ID，以及到达堆叠 300 中的每一个交换节点 302 的正确的堆叠端口 308——也就是说，堆叠 300 中的每一个交换节点 302 都被存储在转发列表 350 和寄存器 352 中的信息进行了预配置。

这三类信息在初始化时提供给堆叠 300 中的每一个交换节点 302。看起来的恶性循环是明显的：堆叠 300 必须被初始化来配置该堆叠 300。因为由管理处理器 306 实现的远程控制不能在缺少上述三类信息的情况下执行——也就是说，除非交换节点 302 被初始化——可以预期初始化本身不能通过远程控制执行。

根据本发明的示例性实施方式，提供了对远程交换网络节点初始化的支持。

依照本发明的示例性实施方式的一个实现，使用两个附加的基本元素：

——交换节点缺省 ID——在初始化前，堆叠 300 中的每一个交换节点 302 都有等于保留 ID “0” 的交换节点 ID。由于 “0” 是一个保留 ID，所以在初始化后的正常工作阶段，堆叠 300 中没有交换节点 302 可能具有等于 0 的其交换节点 ID。

——端口阻塞——每个堆叠端口 308 可以被设置成“阻塞状态”或者“转发状态”任何之一。当堆叠端口 308 处于阻塞状态时，引入的命令帧 410 总是被直接发送给本地命令引擎 310。当堆叠端口 308 处于转发状态时，引入的命令帧 410 根据 FTAG 406 以及转发列表

350 中存储的信息被转发。在缺省状态，每一个堆叠端口 308 都处于阻塞状态。

根据本发明的示例性实施方式的一个实现，堆叠探索及初始化过程 700 使用如图 7 所示的深度优先搜索（depth-first search）。下面是简单的说明：

管理处理器 306 初始化过程 700 开始于对与其最接近的交换节点 302 的初始化，初始化是通过生成 702 一个具有 FTAG 406 的命令帧 410，并且“目的地交换节点”等于“0”，“目的地端口”等于命令引擎的保留端口值来进行的。命令帧 410 通过堆叠端口 308 发送 704。最近 10 的交换节点 302 的搜索引擎接收 704 命令帧 410 并且将其转发到命令引擎 310。

命令引擎 310 按照特定的操作码执行 706 命令帧 410 中的命令。命令引擎 310 也确认 708 命令 410，通过原命令帧 410 到达（704）的路径发送 708 该确认 410 到本地堆叠端口 308。

15 管理处理器 306 接收 710 确认，然后具有通过发送一系列命令帧 410 配置 710 最接近的交换节点 302 所必需的信息。最接近的交换节点 302 的交换节点 ID 被重新分配 712 为“1”。

在进一步处理前，管理处理器 306 选择 714 一个传出堆叠端口 308 来搜索下一个。此处应用深度优先搜索算法。管理处理器 306 配置 20 716/718 交换节点 ID 1 的转发列表 350 来引导去往交换节点 ID 0 的帧到达被选择的发出端口 308。

管理处理器 306 再一次生成 720 并且发送 722 一个“目的地交换节点”等于“0”、“目的地端口”等于命令引擎值的保留值的命令帧 410。已经初始化的交换节点 ID 1 检测到 724 发往交换节点 302-0 的控制帧 25 410，并且发送 726 该命令帧 410 到预先配置的被选择的堆叠端口 308。堆叠 300 中的下一个通过堆叠端口 308 可到达的交换节点 302 是未初始化的，并且因此根据缺省指定自己为交换节点 ID 0。当接收到命令帧 410 时，转发该帧到本地命令引擎 310。堆叠 300 中的第二个交换节点 302 的配置过程与上文描述的最接近管理处理器 306 的交换节点 302 30 的配置相似，最终将交换节点 ID 设置为“2”。

堆叠 300 中的所有交换节点 302 的配置都按这种方式继续。管理



处理器 306 发送一个命令帧 410 到“交换节点 ID 0”，该帧只能由已经初始化的交换节点 302 转发，直到遇到第一个未初始化的 302。然后，管理处理器 306 分配一个新的 ID 给新遇到的交换节点 302 并且按需要配置该节点。在这个过程中，管理处理器 306 配置在已经配置过的交换节点 302 中的转发列表，这样对下一个交换节点 ID 0 的搜索通过一个之前未搜索的堆叠端口 308 进行。

上文描述的方法基本上解决了堆叠初始化的问题。图 8 显示了一个遗留的问题。

在图 8a) 中，管理处理器 306 已经初始化了交换机 302-1、302-2、302-3。当管理处理器 306 试图发现第四个交换节点 302 时，它生成了一个带有 FTAG 的命令帧 410，如所述的，其中“目的地交换节点”等于“0”。在发送这个命令帧 410 之前，管理处理器 306 配置 302-1、302-2、302-3 中的转发列表来沿着粗线所示的路径转发去往交换节点 ID 0 的命令帧 410。

图 8b) 描述了如果交换节点 302-3 实际上被连接回到了交换节点 302-1，形成上文所述的环形 104 交换节点堆叠 100 中的回路的情况下，将发生什么问题。在这种情况下，发往交换节点 ID 0 的命令帧 410 将在堆叠 300 中一圈接一圈地中继来搜索并不存在的未配置的交换节点 302。相似地，没有一个交换节点 302 可以打破这个循环，因为环形 104 中的每一个交换节点 302 都已经通过管理处理器 302 配置来转发这样的命令帧 410 并且从而如配置过的节点那样运行。

环路检测是拓扑探测中一个很关键的方面。依照实例性的实施方式，一个简单的监测和解决环路的方法在图 9 中示出。在通过一个未搜索的堆叠端口 308 转发命令帧 410 之前，管理处理器 306 配置 902 所有的还没有作为阻塞端口成为探索路径的一部分的堆叠端口 308，并且配置 904 探索路径中的堆叠端口 308 作为转发端口。

因此，当命令帧 410 又循环回到交换节点 302-1 时，它就到达了一个阻塞端口 (308)，并且如上文所述，被立刻转发到本地命令引擎 310。本地命令引擎 310 执行该命令，并且发回一个确认到管理处理器 306。管理处理器 306 接收该确认，并且从 FTAG 406 中的“源交换节点”字段观察到该确认 (410) 来自交换节点 ID 1，而不是交换节点 ID 0，这

标志着在拓扑结构中发现了一个环。

在上文中,已经关于由单一的管理处理器 306 管理的单一堆叠 300 的交换节点 302 对本发明进行了描述。为了实现特定的应用和/或例如提供载荷共享,一个堆叠的交换节点 302 可以被两个或多个管理处理器 5 器 306 控制,如图 10 所示。只要每个交换节点 302 得知自己的控制管理处理器 306 的位置,那么上文描述的方法和算法就可以不经修改地应用。每一个管理处理器 306 以及相应的受控交换节点 302 组成一个管理域。堆叠的探索和初始化大致上仍然相同,与其区别可以参照图 10 很好地进行解释。

10 在图 10 中,管理处理器 306-A 以及管理处理器 306-B 都如上文所述初始化它们自己的域。当两个管理处理器都试图初始化同一个交换节点 302-S 时,发生了冲突。为了解决这个冲突,假设管理处理器 306-A 先初始化交换节点 302-S。当管理处理器 306-B 试图通过发送命令帧 410 来重初始化网络节点 302-S 时,交换节点 302-S 的命令引擎 310 将  
15 发回相应的确认到管理处理器 302-A,因为管理处理器 306-A 是根据当前配置的交换节点 302-S 的控制管理处理器 306。当管理处理器 302-A 接收到一个针对从来没有发送过的命令帧 410 的确认时,管理处理器 302-A 就会识别这个冲突。在接收到这样的确认后,管理处理器 306-A 以及 306-B 就会直接通信,并且协商哪个交换节点将被各自控制。

20 本文所示的实施方式仅仅是示范性的,本领域的技术人员将会理解上述实施方式的变形可以在不偏离本发明的精神的情况下产生。本发明的范围仅由附加的权利要求限定。

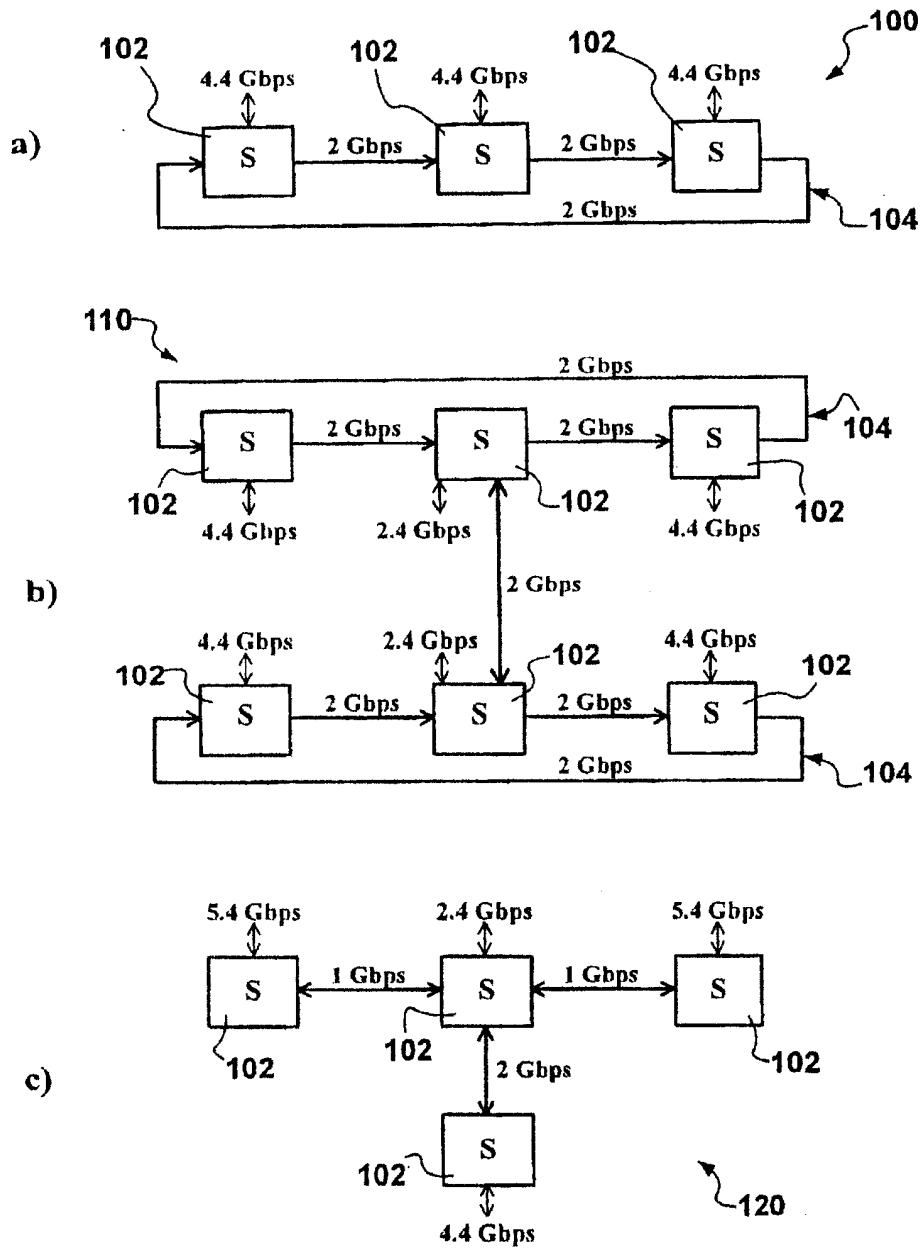


图1

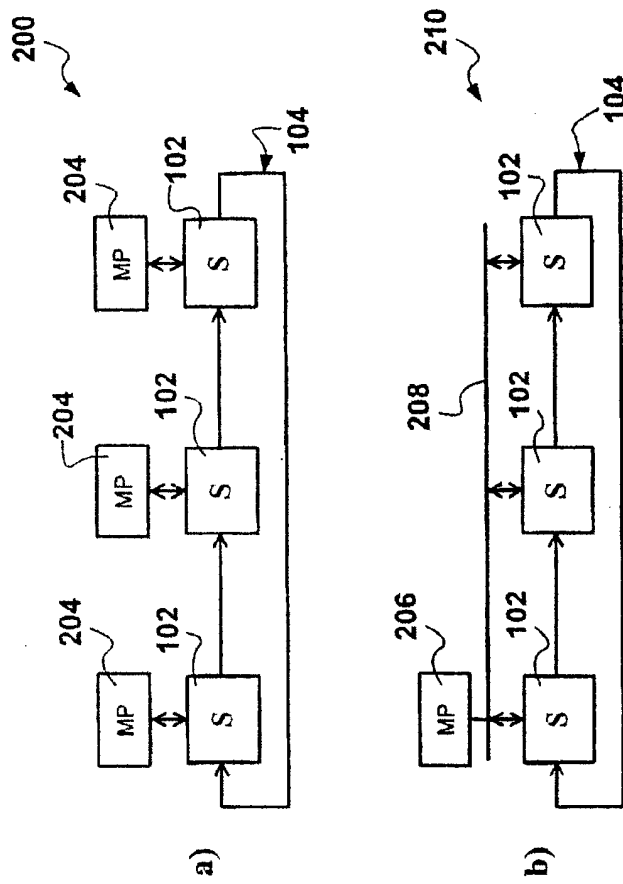


图2

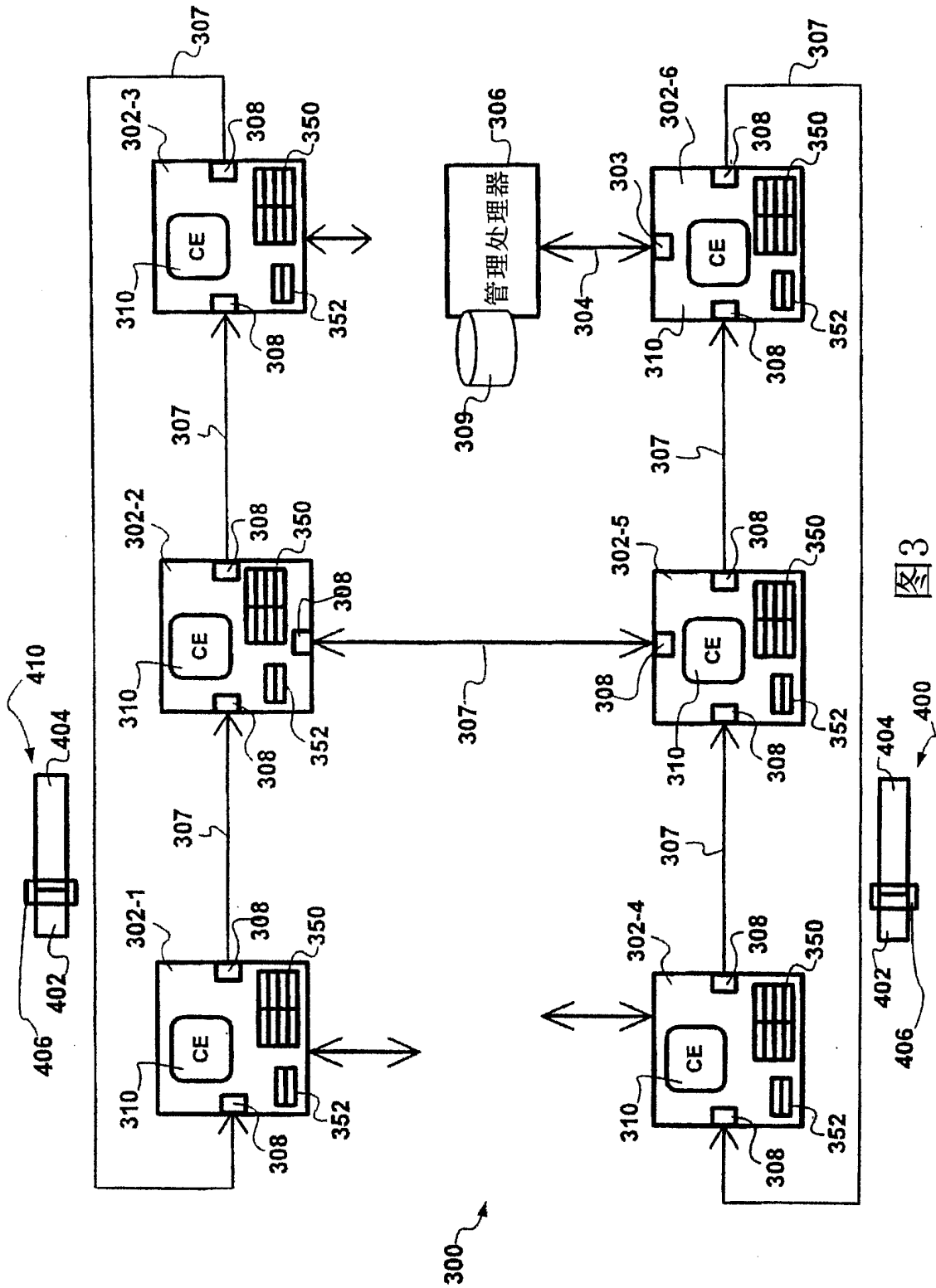


图3

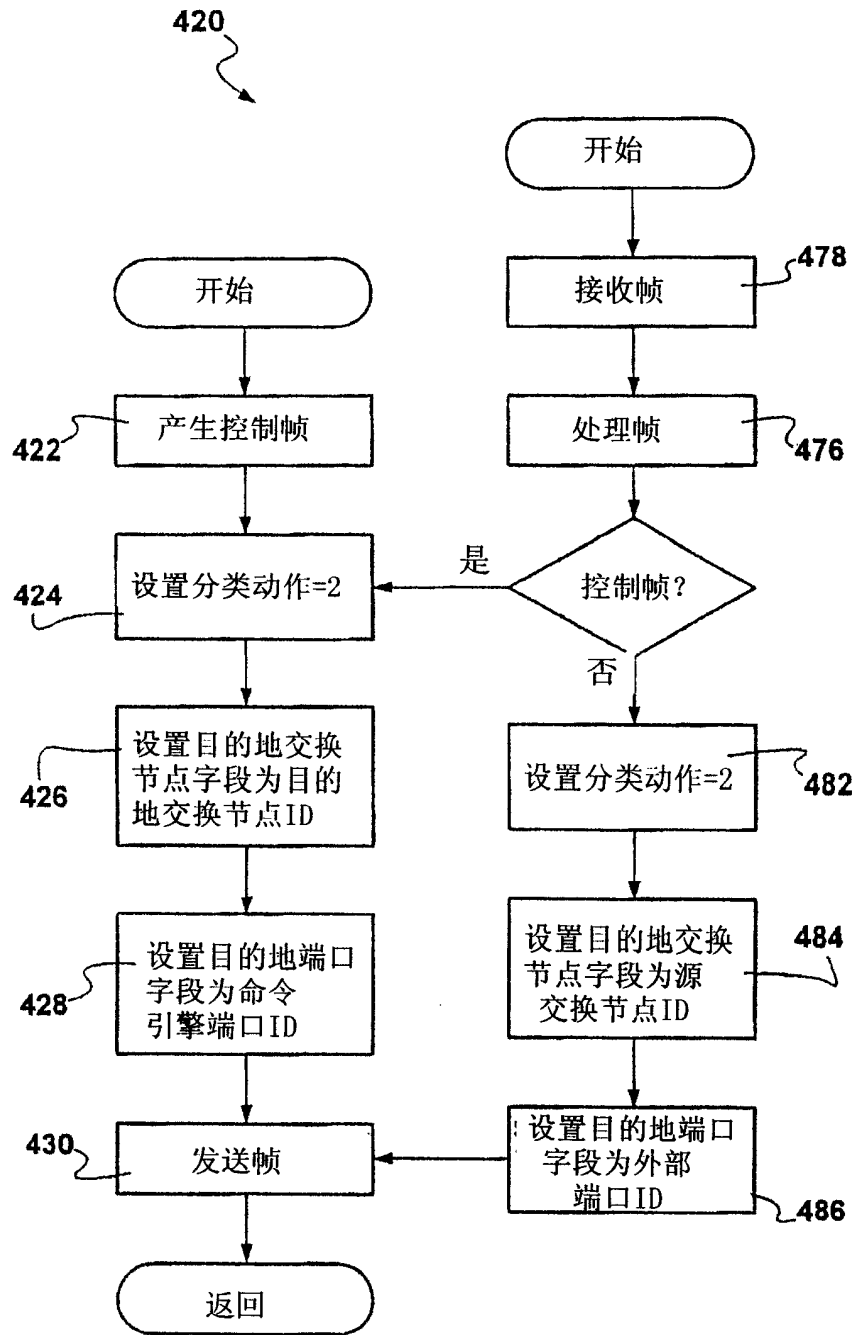


图4



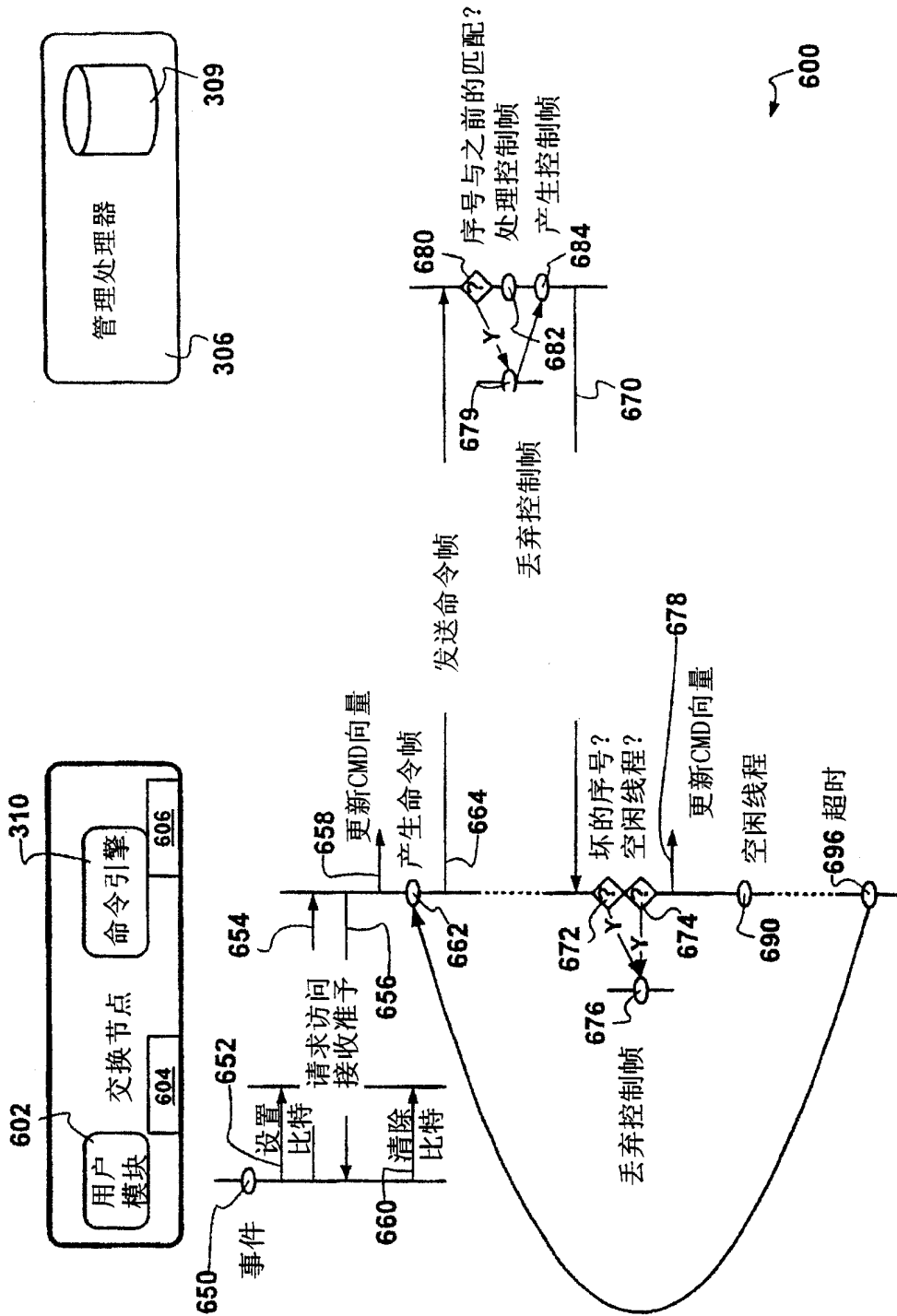


图6



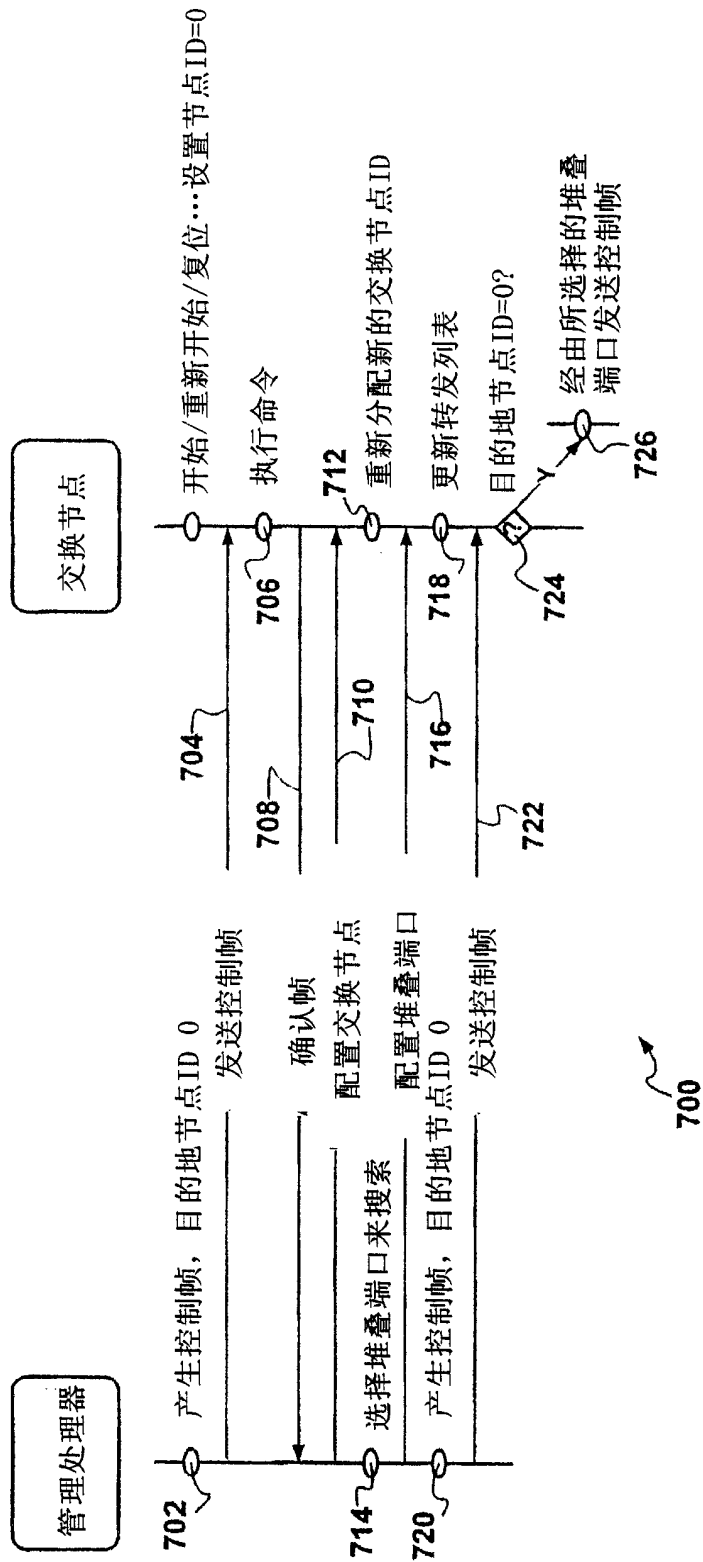


图7

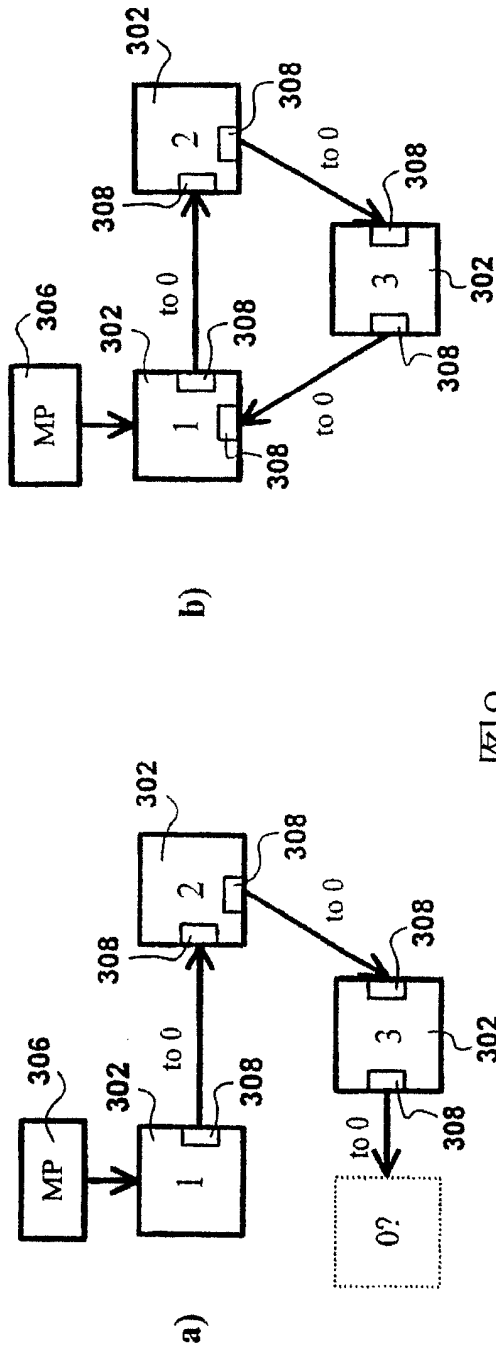


图8

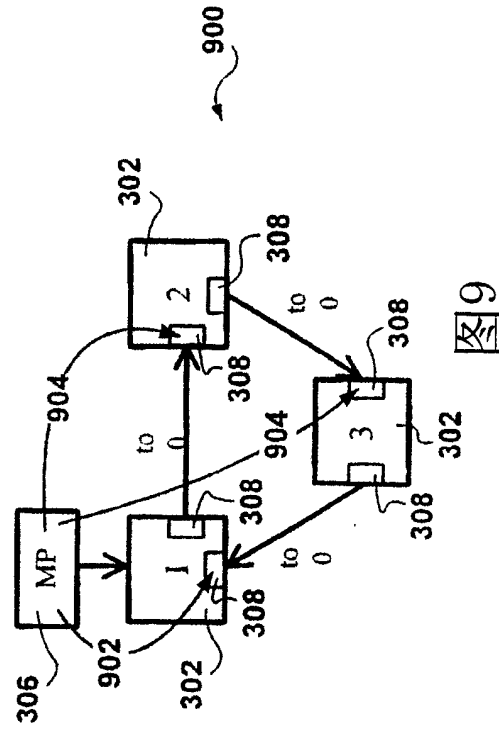


图9

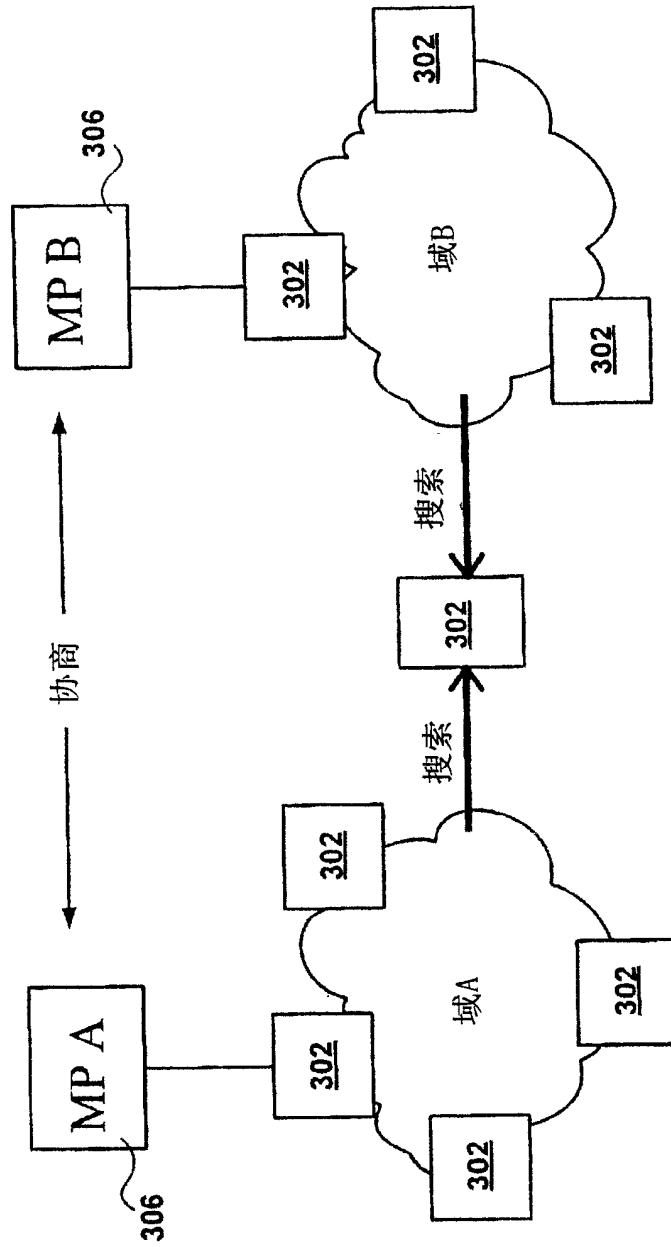


图10