



[12] 发明专利申请公开说明书

[21] 申请号 03818330.7

[43] 公开日 2005 年 9 月 21 日

[11] 公开号 CN 1672387A

[22] 申请日 2003.7.31 [21] 申请号 03818330.7

[30] 优先权

[32] 2002.7.31 [33] US [31] 10/210,190

[86] 国际申请 PCT/US2003/023946 2003.7.31

[87] 国际公布 WO2004/012418 英 2004.2.5

[85] 进入国家阶段日期 2005.1.31

[71] 申请人 思科技术公司

地址 美国加利福尼亚州

[72] 发明人 马尔科·E·福斯基亚诺

贾斯廷·Q·陈

安巴里什·C·伦盖尔

[74] 专利代理机构 北京东方亿思知识产权代理有限公司

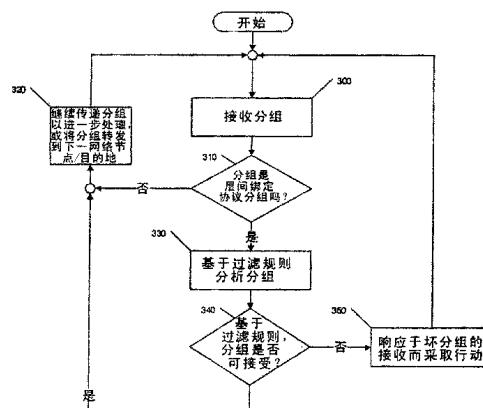
代理人 王 怡

权利要求书 6 页 说明书 14 页 附图 10 页

[54] 发明名称 用于层间绑定检验的方法和装置

[57] 摘要

本发明公开了一种用于检验分组的方法。该方法包括通过确定分组是不是层间绑定协议分组(310)，并且如果所述分组是层间绑定协议分组(310)，则检验所述分组(330)，从而处理所述分组。层间绑定协议分组指示第一网络层地址和第二网络层地址之间的绑定。



1. 一种网络设备，包括：
转发引擎；和
5 检验引擎，所述检验引擎耦合到所述转发引擎，其中
所述转发引擎被配置为如果分组是层间绑定协议分组，则将所述
分组转发到所述检验引擎，并且
所述检验引擎被配置为检验所述层间绑定协议分组。
2. 如权利要求 1 所述的网络设备，其中，所述检验引擎被配置为借助
10 被配置来检验所述层间绑定协议分组中所代表的层间绑定，来检验所述层
间绑定协议分组。
3. 如权利要求 2 所述的网络设备，其中所述层间绑定协议分组是地址
解析协议分组，并且所述层间绑定是互联网协议地址和媒体访问控制地址
之间的绑定。
- 15 4. 如权利要求 1 所述的网络设备，其中所述转发引擎被配置为确定所
述分组是不是所述层间绑定协议分组。
5. 如权利要求 4 所述的网络设备，其中所述转发引擎被配置为借助被
配置来分析所述分组的头部来做出所述确定。
6. 如权利要求 1 所述的网络设备，所述转发引擎还包括：
20 查找表；和
查找表控制器，所述查找表控制器耦合到所述查找表，其中
所述查找表控制器被配置为如果所述分组是所述层间绑定协议分
组，则使所述转发引擎将所述分组转发到所述检验引擎。
7. 如权利要求 1 所述的网络设备，其中所述转发引擎被配置为确定所
25 述分组是不是所述层间绑定协议分组。
8. 如权利要求 1 所述的网络设备，还包括：
第一端口设备，可通信地耦合到所述转发引擎；
第二端口设备，可通信地耦合到所述转发引擎和所述第一端口设备；
以及

端口设备控制器，被耦合以控制所述第二端口设备，并且被耦合以由所述转发引擎来控制。

9. 如权利要求 8 所述的网络设备，其中

- 5 所述转发引擎被配置为从所述第一端口设备接收所述分组，
 所述第二端口设备被配置为从所述第一端口设备接收所述分组，
 所述转发引擎还被配置为如果所述分组不是所述层间绑定协议分组，
 则使得所述端口设备控制器让所述第二端口设备来处理所述分组。

10. 如权利要求 1 所述的网络设备，还包括：

- 第一端口设备，可通信地耦合到所述转发引擎，其中
10 所述转发引擎被配置为从所述第一端口设备接收所述分组，
 所述检验引擎被配置为如果所述分组是所述层间绑定协议分组，
 则从所述转发引擎接收所述分组，并向所述转发引擎指示出所述层间
 绑定协议分组是可接受的。

11. 如权利要求 10 所述的网络设备，还包括：

- 15 第二端口设备，可通信地耦合到所述转发引擎和所述第一端口设备；
 和

 端口设备控制器，被耦合以控制所述第二端口设备，并且被耦合以由
 所述转发引擎来控制，其中

- 所述转发引擎还被配置为如果所述分组是所述层间绑定协议分组
20 并且如果所述层间绑定协议分组是可接受的，则使得所述端口设备控
 制器让所述第二端口设备来处理所述分组。

12. 一种用于检验分组的方法，包括：

- 通过确定分组是不是层间绑定协议分组来处理所述分组，所述层间绑
 定协议分组指示出第一网络层地址和第二网络层地址之间的绑定；以及
25 如果所述分组是层间绑定协议分组，则检验所述分组。

13. 如权利要求 12 所述的方法，其中所述检验操作确定所述绑定是不
 是可接受的绑定。

14. 如权利要求 12 所述的方法，其中所述分组是多个分组之一，并且
 所述方法还包括：

对所述分组执行所述处理；以及
对处理所述分组的速率进行限制。

15. 如权利要求 14 所述的方法，其中所述限制包括：
将所述分组的总数与一个限度进行比较；以及
5 如果所述分组的所述总数大于所述限度，则丢弃所述分组。

16. 如权利要求 14 所述的方法，其中所述分组是在端口上接收到的，
并且所述限制包括：

- 将所述分组的总数与端口关闭阈值进行比较；以及
如果所述分组的所述总数大于所述端口关闭阈值，则关闭所述端口。

- 10 17. 如权利要求 14 所述的方法，其中所述限制包括：
将所述分组的总数与端口丢弃阈值进行比较；以及
如果所述分组的所述总数大于所述端口丢弃阈值，则丢弃所述分组。

18. 一种网络元件，包括：

处理器；

- 15 耦合到所述处理器的计算机可读介质；以及
用于检验分组的计算机代码，所述计算机代码被编码在所述计算机可
读介质中，并被配置为使得所述处理器执行以下操作：

- 通过确定分组是不是层间绑定协议分组来处理所述分组，所述层
间绑定协议分组指示出第一网络层地址和第二网络层地址之间的绑
20 定；以及

如果所述分组是层间绑定协议分组，则检验所述分组。

19. 如权利要求 18 所述的计算机系统，其中所述检验操作确定所述绑
定是不是可接受的绑定。

20. 如权利要求 18 所述的计算机系统，其中所述分组是多个分组之
25 一，并且所述计算机代码还被配置为使得所述处理器执行以下操作：

对所述分组执行所述处理；以及

对处理所述分组的速率进行限制。

21. 如权利要求 20 所述的计算机系统，其中被配置为使得所述处理器
限制所述速率的计算机代码还被配置为使得所述处理器执行以下操作：

将所述分组的总数与一个限度进行比较；以及

如果所述分组的所述总数大于所述限度，则丢弃所述分组。

22. 如权利要求 20 所述的计算机系统，其中所述分组是在端口上接收到的，并且被配置为使得所述处理器限制所述速率的计算机代码还被配置为使得所述处理器执行以下操作：

将所述分组的总数与端口关闭阈值进行比较；以及

如果所述分组的所述总数大于所述端口关闭阈值，则关闭所述端口。

23. 如权利要求 20 所述的计算机系统，其中被配置为使得所述处理器限制所述速率的计算机代码还被配置为使得所述处理器执行以下操作：

- 10 将所述分组的总数与端口丢弃阈值进行比较；以及

如果所述分组的所述总数大于所述端口丢弃阈值，则丢弃所述分组。

24. 一种用于检验分组的计算机程序产品，包括：

- 第一组指令，所述第一组指令可在计算机系统上执行，并被配置为通过确定分组是不是层间绑定协议分组来处理所述分组，所述层间绑定协议
15 分组指示出第一网络层地址和第二网络层地址之间的绑定；

第二组指令，所述第二组指令可在所述计算机系统上执行，并被配置为如果所述分组是层间绑定协议分组，则检验所述分组；以及

计算机可读介质，其中所述计算机程序产品被编码在所述计算机可读介质中。

- 20 25. 如权利要求 24 所述的计算机程序产品，其中所述检验操作确定所述绑定是不是可接受的绑定。

26. 如权利要求 24 所述的计算机程序产品，其中所述分组是多个分组之一，并且所述计算机程序产品还包括：

- 第三组指令，所述第三组指令可在所述计算机系统上执行，并被配置
25 为对所述分组执行所述处理；以及

第四组指令，所述第四组指令可在所述计算机系统上执行，并被配置为对处理所述分组的速率进行限制。

27. 如权利要求 26 所述的计算机程序产品，其中所述第四组指令包括：

第一小组指令，所述第一小组指令可在所述计算机系统上执行，并被配置为将所述分组的总数与一个限度进行比较；以及

第二小组指令，所述第二小组指令可在所述计算机系统上执行，并被配置为如果所述分组的所述总数大于所述限度，则丢弃所述分组。

- 5 28. 如权利要求 26 所述的计算机程序产品，其中所述分组是在端口上接收到的，并且所述第四组指令包括：

第一小组指令，所述第一小组指令可在所述计算机系统上执行，并被配置为将所述分组的总数与端口关闭阈值进行比较；以及

- 10 第二小组指令，所述第二小组指令可在所述计算机系统上执行，并被配置为如果所述分组的所述总数大于所述端口关闭阈值，则关闭所述端口。

29. 如权利要求 26 所述的计算机程序产品，其中所述第四组指令包括：

- 15 第一小组指令，所述第一小组指令可在所述计算机系统上执行，并被配置为将所述分组的总数与端口丢弃阈值进行比较；以及

第二小组指令，所述第二小组指令可在所述计算机系统上执行，并被配置为如果所述分组的所述总数大于所述端口丢弃阈值，则丢弃所述分组。

- 20 30. 一种用于检验分组的装置，包括：

用于通过确定分组是不是层间绑定协议分组来处理所述分组的装置，所述层间绑定协议分组指示出第一网络层地址和第二网络层地址之间的绑定；以及

用于如果所述分组是层间绑定协议分组，则检验所述分组的装置。

- 25 31. 如权利要求 30 所述的装置，其中所述用于检验的装置确定所述绑定是不是可接受的绑定。

32. 如权利要求 30 所述的装置，其中所述分组是多个分组之一，并且所述装置还包括：

用于对所述分组执行所述处理的装置；以及

用于对处理所述分组的速率进行限制的装置。

33. 如权利要求 32 所述的装置，其中所述用于限制的装置包括：
用于将所述分组的总数与一个限度进行比较的装置；以及
用于如果所述分组的所述总数大于所述限度，则丢弃所述分组的装置。

- 5 34. 如权利要求 32 所述的装置，其中所述分组是在端口上接收到的，并且所述用于限制的装置包括：
用于将所述分组的总数与端口关闭阈值进行比较的装置；以及
用于如果所述分组的所述总数大于所述端口关闭阈值，则关闭所述端口的装置。

- 10 35. 如权利要求 32 所述的方法，其中所述用于限制的装置包括：
用于将所述分组的总数与端口丢弃阈值进行比较的装置；以及
用于如果所述分组的所述总数大于所述端口丢弃阈值，则丢弃所述分组的装置。

用于层间绑定检验的方法和装置

5 技术领域

本发明涉及信息网络领域，更具体地涉及用于对由层间绑定协议（inter-layer binding protocol）的分组所创建的绑定进行检验的方法和装置。

10 背景技术

可用于保护和穿透联网环境的工具库在数量和能力两方面都非常出色。这些工具中的一些是非常专门化的，而另一些是多目的的，并且作为用于更大的工具包的构建模块。这种工具中的一种是网络“嗅探器”（sniffer）。最一般形式的网络“嗅探”包括从网络拦截通信（例如帧或分组）和查看其内容。此能力的普及已经有一段时间，并且已被网络管理员（例如故障检修问题）、所谓的“破解者”（cracker，拦截密码和文件的人）等所采用。应当注意到，相对而言，网络嗅探近来才在交换网络上成为可能。正如人们可能希望的那样，如今出现了允许在交换网络上进行网络嗅探的工具。以下简要说明非交换网络的工作方式和如何对这种网络进行嗅探，以及交换网络的工作方式和如何对这些网络进行嗅探。

图 1A 是一般地示出非交换网络环境的体系结构的框图，所述体系结构在图 1A 中示出为网络 100。网络 100 包括多个节点（节点 102(1)-(N)），这些节点由集线器 106 耦合到路由器 104。节点 102(1)-(N)中的每个节点都耦合到集线器 106 的一个单独的端口（未示出）。在图 1A 所示的非交换网络环境中，存在网段的概念。段（segment）是一种存在于路由器、桥接器、集线器或交换机背后的网络体系结构，其中每个节点是从每个其他节点直接可寻址的。在某些网络中这也称为子网。于是，在图 1A 中将节点 102(1)-(N)示出为在一个段中（在图 1A 中示出为子网 108）。

在非交换环境中，以广播的方式来处理帧。即，当一个节点发送帧

时，该帧就被段上的每个节点“看见”。而每个节点又大致地检查该帧，以了解该帧是否被定址到给定节点。如果不是，则给定节点丢弃所述帧。然而，如果所述节点是预期的接收者，则所述节点接受所述帧以进行处理。为了这里的讨论，将节点 102(2)指定为采用嗅探主体（sniffing agent）的主机。节点 102(1)和 102(3)代表仅仅试图相互通信的“无辜者”。

图 1B 的流程图示出了当非交换网络中的一个和多个数据流被嗅探到时，该网络中流量的流动。为了让节点用作嗅探主体，该节点的网络接口被设置为“混杂”（promiscuous）模式。设置此模式通常需要在该节点处的根访问或管理员访问。在设置此模式之后，该节点的网络接口就会将这些帧传递到更高的网络层，并希望更高层处的某个软件会处理这些帧。

图 1B 所示的过程开始于节点 102(1)发送帧，并在该帧中指示出该帧将被发送到节点 102(3)（步骤 150）。集线器 106 随后将所述帧广播到它的每个活动端口（步骤 155）。节点 102(2)接收所述帧，并检查该帧中的目的地址（步骤 160 和 165）。因为节点 102(2)被设置成“混杂”模式（步骤 170），所以节点 102(2)接受所述帧（虽然该帧并未定址到节点 102(2)）（步骤 175）。如上所述，将节点的网络接口设置成“混杂”模式使该网络接口可以接受任意帧，而不管该帧中的地址（例如 MAC（媒体访问控制）地址）如何。然而，即使所述接口将会保存所述帧，通常也需要某种更高级的软件来处理数据。接下来，节点 102(1)-(N)中的其他节点（例如耦合到集线器 106 上的活动端口的其他节点）接收所述帧，并确定它们不是预期的主机（步骤 160、165 和 170），因此丢弃所述帧（步骤 180）。（当然，如果节点 102(2)的网络接口未被设置成“混杂”模式，节点 102(2)就也会忽略所述分组。）如果是预期目的地（节点 102(3)），则节点 102(3)也接收所述帧，并检查该帧的目的地址（步骤 160 和 165）。在节点 102(3)确定它是预期的主机之后（步骤 165），节点 102(3)进一步处理所述帧（步骤 185）。

为全面起见，应当注意，步骤 160/165/170/180，160/165/170/175 以及 160/165/185 可以进行种种调换或者同时发生，正如为了这里的讨论，对哪

个节点将会首先接收所述帧进行的预测并不重要一样。为了实用，可以假定这些操作同时发生，而不失一般性。

从图 1A 和 1B 中可以看出，非交换环境易受到嗅探。这种环境几乎无需嗅探主体一方做出额外的努力，因为集线器将帧广播到所有活动端口。

- 5 前面已经暗示了，几种这样的嗅探实用程序（utility）是存在并且公开可以获得的。这种可能性使得不道德者在被称为“中间人”（man-in-the-middle）攻击的设置中，当信息在“无辜”节点传递的时候查看该信息，而不被这些节点或其用户注意到。而且，一旦对数据流进行嗅探的黑客（hacker）将其自身插入到无辜方之间，该黑客就可以以完全透明的方式，自由地对任意一方生成各种回复。
- 10

图 2A 是一般地示出交换网络环境的体系结构的框图，所述体系结构在图 2A 中示出为网络 200。在交换网络环境中，网段的概念仍然存在，但这种网段仅包括交换机和所关心的节点，并且以直接的方式来处理帧。即，从第一节点到第二节点的帧仅通过在所述第一和第二节点之间完成连接所必需的交换机中的电路而发送。

15

网络 200 包括多个节点（节点 202(1)-(N)），这些节点由交换机 206 耦合到路由器 204。节点 202(1)-(N)中的每个节点都耦合到交换机 206 的一个单独的端口（未示出）。在图 2A 所示的交换网络环境中，存在网段的概念。段是一种存在于路由器、桥接器或交换机背后的网络体系结构，其中每个节点是从每个其他节点直接可寻址的。在某些网络中这也称为子网。于是，在图 1A 中将节点 202(1)和 202(3)示出为在一个段中（在图 2A 中示出为子网 208）。

20

图 2B 的流程图示出了交换网络中流量的正常流动。首先，节点 202(1)发送帧，并在该帧中指示出该帧将被发送到节点 202(3)（步骤 210）。交换机 206 随后检查所述帧，并确定应对哪个节点（端口）进行连接（步骤 220）。一旦做出了这一确定，交换机 206 就在节点 202(1)和节点 202(3)所分别耦合的端口之间配置连接（步骤 230）。交换机 206 随后将所述帧转发到其预期节点，即节点 202(3)（步骤 240）。一旦节点 202(3)接收到所述帧，节点 202(3)就检查所述帧的目的地址，以确定节点 202(3)是不是该

25

帧的预期目的地（步骤 250）。如果此地址确定指示出节点 202(3)不是正确的目的地，则不处理所述帧。否则，节点 202(3)就对所述帧执行理所当然的处理（步骤 260）。

这种操作模式具有某些固有的优点：

- 5 1) 较低的网络流量，因为不将帧广播到每个节点，换句话说是通过减少冲突域而带来较高的带宽。

2) 较低的节点处理开销，这是由于每个节点仅需处理要去往该节点的帧而导致的。

- 10 然而，也存在一些折衷。例如，给交换机带来了更高的开销处理要求，因为交换机必须忙于创建机器之间的虚拟连接。

可见，因为交换网络不广播大多数帧，所以交换网络不像非交换网络那样易遭受嗅探。然而，也有若干方法可用来对交换网络进行嗅探。这种方法的一个例子是地址解析协议（ARP）哄骗（spoofing），下面将对其进行简要论述。

- 15 互联网协议（IP）的基本操作之一是围绕 ARP（地址解析协议）请求和回复来进行的。一般而言，当网络上的第一节点想要与第二节点进行通信时，第一节点发送 ARP 请求。第二节点将会发送包括其 MAC 地址在内的 ARP 回复。即使在交换网络中，这个初始 ARP 请求也是以广播的方式发送的。第三节点可以制作并发送未经请求的假的 ARP 回复到第一节点。
- 20 此假 ARP 回复将会指明，第三节点具有第二节点的 IP 地址。由于第三节点将其自身表示为具有预期的 IP 地址，因此第一节点随后无意地将流量发送到第三节点。某些可用的工具专门用于向某些类别的机器（例如 NFS 服务器、HTTP 服务器等）发送假的 ARP 回复。一种这样的工具是“dsniff”，它适于对特定类型的流量进行嗅探。其他工具监听一般的
- 25 ARP 请求，并在该时刻发送假的 ARP 回复，并且非常适于对整个网络进行嗅探。要使这类攻击起作用，需要能够将接收到的帧继续转发到其预期目的地。这通常是通过某种内核级或应用级的 IP 转发来实现的。

尽管存在若干防御这种攻击的方法，但其中每种方法都有其自身的缺点。（应当注意，这些方法中的一些同时适用于非交换网络环境和交换网

络环境。) 这些方案包括 IP 过滤、端口安全性和路由安全性。

通过在交换机上实现 IP 过滤, 用户直接指定允许哪些流量从每个端口流入和流出。尽管可能有效, 但这种方法可能需要极大的努力来设置和管理, 尤其是在动态环境的情况下。

5 可替换地, 如果集线器或者交换机有能力实现端口安全性, 这种措施可以帮助网络的节点防御 MAC 泛滥(flooding) 和 MAC 哄骗攻击。此特征有效地防止集线器或交换机承认一个物理端口上的多于一个 MAC 地址。然而, 像许多安全性程序一样, 这对环境造成了约束, 并且增加了对管理过程和审核过程的要求。

10 而且, 不希望将安全性推进到网络节点级, 这有以下几个原因。首先, 其使得安全性的源对于能够访问这种节点的任何人可用。而且, 其还大大增加了对这种安全性措施进行管理的任务, 因为每个节点必须被单独配置为支持这种安全性措施。在节点的连接性动态变化的网络环境中(例如膝上型计算机), 这被证明是尤其复杂的。

15

发明内容

在一种实施方式中, 公开了一种网络设备。该网络设备包括转发引擎和检验引擎, 检验引擎耦合到转发引擎。转发引擎被配置为如果分组是层间绑定协议分组, 则将所述分组转发到检验引擎。检验引擎被配置为检验

20 层间绑定协议分组。

在一种实施方式中, 公开了一种用于检验分组的方法。该方法包括通过确定分组是不是层间绑定协议分组来处理所述分组, 并且如果所述分组是层间绑定协议分组, 则检验所述分组。层间绑定协议分组指示第一网络层地址和第二网络层地址之间的绑定。

25 以上是对发明内容的概述, 因此包含了对细节的必要简化、概括和省略; 因此, 本领域技术人员应该理解到, 所述发明内容仅仅是说明性的, 而绝非想要进行限制。在下文所阐述的非限制性的具体实施方式中, 本发明的其他方面、创造性特征和优点将会变得清楚, 这些方面、创造性特征和优点仅由权利要求书来定义。

附图说明

通过参考附图，本领域技术人员将会更好地理解本发明，并且清楚许多目的、特征和优点。

5 图 1A 是示出现有技术的非交换网络的框图。

图 1B 是示出现有技术的非交换网络的操作的框图。

图 2A 是示出现有技术的交换网络的框图。

图 2B 是示出现有技术的交换网络的操作的框图。

10 图 3 的流程图一般地示出了根据本发明实施方式的层间绑定检验的处理。

图 4A 是示出根据本发明实施方式的网络的框图。

图 4B 是更详细地示出图 4A 的网络的部分的框图。

图 5 是示出根据本发明实施方式的交换机的框图。

图 6 的框图更详细地示出了根据本发明实施方式的转发引擎的特征。

15 图 7 的流程图示出了在图 5 的网络元件上实现的例如图 3 所示的处理。

图 8 是根据本发明的对层间绑定协议分组进行速率限制处理的流程图。

不同附图中相同的标号用来指示相似或相同的项目。

20

具体实施方式

以下希望提供对本发明实施例的详细描述，而不用被理解为限制发明本身。确切地说，多种变化可以落入所附权利要求书所定义的本发明的范围之内。

25 引言

根据本发明实施方式的系统和方法提供了对层间绑定协议（ILBP）分组的检验，以防止出现以下情况，即错误的和/或未经授权的绑定生效，因而导致受到影响的网络中信息（例如分组）的错误路由。ILBP 的一个例子是地址解析协议（ARP），其将互联网协议（IP）地址（第 3 层）绑定到

媒体访问控制（MAC）地址（第 2 层）。ILBP 的另一个例子是 IP 版本 6 的网络发现协议（NDP）。这种系统和方法通过识别 ILBP 分组并仅分析这些分组来提供所述的保护，从而使该方法对吞吐率的影响最小。如果由于恶意或错误动作而导致分组不合法，则丢弃不合法的分组。而且，通过

5 在分析之前丢弃（多个）ILBP 分组，通过关闭发送大量这种分组的端口，或者其他适当的响应，可以实现速率限制。这防止了分组风暴（storm）和对这种分组进行分析的检验引擎的过载。

根据本发明的层间绑定检验的实施例

图 3 的流程图一般地示出了根据本发明实施方式的层间绑定检验的处理。该处理开始于对协议分组（或者帧或其他通信单元）的接收（步骤 300）。首先确定接收到的分组是绑定协议分组并且因而需要分析（步骤 310）。如果接收到的分组不是绑定协议分组，则传递该分组以用于进一步处理（或者将其转发到下一网络节点或目的节点）（步骤 320）。反之，如果接收到的分组是绑定协议分组，则基于过滤规则来分析该分组

15 （步骤 330）。接下来，基于所采用的过滤规则，确定是否可接受该分组（步骤 340）。如果接收到的分组是可接受的分组，则再次传递该分组以用于进一步处理（或者将其转发到下一网络节点或目的节点）（步骤 320）。然而，如果基于所采用的过滤规则，接收到的分组是不可接受的（步骤 340），则响应于所述坏分组的接收而采取适当的动作（步骤

20 350）。这些动作包括仅仅丢弃坏分组，将坏分组的接收记入日志，或者中断来自给定源节点的分组流，或者上述动作的组合，以及其他这种可能的动作。

如上所述，图 3 的流程图示出了根据本发明实施方式的处理。应该意识到，这里所讨论的操作可能包括由计算机系统的用户直接输入的命令或由专用的硬件模块执行的步骤，但优选实施方式包括由软件模块执行的步骤。这里所提及的步骤的功能可以对应于模块或模块的部分的功能。

25

这里所提及的操作可以是模块或模块的若干部分（例如软件、固件或硬件模块）。例如，虽然所描述的实施方式包括软件模块和/或包括手动输入的用户命令，但多个示例模块可以是专用的硬件模块。这里所讨论的软

件模块可以包括脚本、批处理或其它可执行文件，或者这种文件的组合和/或部分。软件模块可以包括编码于计算机可读介质上的计算机程序或其例程。

- 另外，本领域技术人员将会认识到，模块之间的边界仅仅是说明性的，替换性的实施例可以合并模块或者实施替换性的对模块功能的分解。例如，这里所讨论的模块可被分解成子模块以作为多个计算机进程来执行，并且还可以选择在多个计算机上执行。而且，替换性的实施方式可以将特定模块或子模块的示例组合起来。此外，本领域技术人员将会认识到，在示例性实施方式中所描述的操作仅用于说明。根据本发明，可以组合多个操作，或者可以将操作的功能分布到更多的操作中。

可替换地，这种操作可以包含在实现了这种功能的电路结构中，例如复杂指令集计算机（CISC）的微代码、被编程到可编程或可擦除/可编程器件中的固件、现场可编程门阵列（FPGA）的配置、门阵列或完全定制专用集成电路（ASIC）的设计，等等。

- 流程图中的每个框可以由模块（例如软件模块）或软件模块的一部分或者计算机系统用户使用诸如随后描述的计算机系统 800 之类的计算机系统来执行。因此，上述方法及其操作以及用于所述方法的模块可以在计算机系统上执行和/或从计算机可读介质执行，所述计算机系统被配置为执行所述方法的操作。所述方法可以包含在用于将计算机系统配置为执行所述方法的机器可读介质和/或计算机可读介质中。因此，软件模块可以被存储在和/或发送到计算机系统存储器，以配置计算机系统来执行该模块的功能。

- 这种计算机系统通常根据程序（在内部存储的指令列表，例如特定的应用程序和/或操作系统）来处理信息，并产生作为结果的经由 I/O 设备输出的信息。计算机进程通常包括执行（运行）程序或程序的一部分、当前程序值和状态信息，以及由操作系统用来管理进程执行的资源。父进程可以派生其他子进程，以帮助执行父进程的整体功能。因为父进程具体地派生子进程以执行父进程的整体功能中的一部分，所以子进程（以及孙子进程等等）所执行的功能可能有时被描述为是由父进程执行的。

这种计算机系统通常包括多个“并发”执行的多个计算机进程。计算机系统常常包括能够交替支持许多活动进程的单个处理单元。虽然多个进程可能看起来是被并发执行的，但是在任意给定时刻，单个处理单元实际上只执行一个进程。通过快速改变进程的执行，计算机系统看起来好像在并发执行进程。计算机系统在执行的不同阶段中，在多个进程之间复用计算机系统资源的能力称为多任务操作（multitasking）。具有多个处理单元的系统确切地说可以支持真正的并发处理，这种系统被称为多处理系统。当活动进程在多任务操作和/或多处理环境中执行时，这些进程常常被称为并发执行。

10 这里所描述的软件模块可以由这种计算机系统例如从计算机可读介质接收。所述计算机可读介质可以永久性地、可移动地或远程地耦合到所述计算机系统。计算机可读介质例如可以包括以下多种介质，但不局限于此：包括磁盘和磁带存储介质在内的磁存储介质。诸如光盘介质（例如 CD-ROM, CD-R 等）和数字视频盘存储介质之类的光存储介质。包括基
15 于半导体的存储器单元在内的非易失性存储器存储介质，例如 FLASH 存储器、EEPROM, EPROM, ROM 或专用集成电路。易失性存储介质，包括寄存器、缓冲器或缓存、主存储器、RAM 等。还有包括计算机网络、点到点电信和载波传输介质在内的数据传输介质。在基于 UNIX 的实施方式中，软件模块可以包含在文件中，所述文件可以是设备、终端、本地或
20 远程文件、套接口、网络连接、信号或其他通信或状态改变的手段。其他新的和各种类型的计算机可读介质可用来存储和/或发送这里所描述的软件模块。

图 4A 示出了根据本发明实施方式的网络 400。网络 400 包括多个用户节点（在图 4A 中示出为节点 410(1)-(N)），ISP 420 和互联网 430。图 4A
25 所示的 ISP 420 使用路由器 440 和交换机 450，支持节点 410(1)-(N)和互联网 430 之间的连接性。交换机 450 支持根据本发明实施方式的处理，保护点 410(1)-(N)以防御“中间人”攻击、嗅探和其他这种攻击，以及错误的层间绑定。应当注意的是，更一般而言，本发明的结构和处理可以包括在其他网络元件（例如路由器 440）中，而不失其一般性或适用性，因为这

种网络元件的功能可以组合。还应当注意，尽管这里的讨论利用了交换机（交换机 450），但这仅仅是说明性的实施例——本发明可以实现在集线器或其他环境中。

图 4B 更详细地示出了网络 400 的部分。现在将交换机 450 示出为包括转发引擎 460 和检验引擎 470。最简单地，转发引擎 460 确定将分组转发（即发送）到哪个端口（因而发送到哪个节点）。检验引擎 470 所执行的功能是要检验这些由转发引擎 460 转发以进行检验的分组。图 4B 还示出了分组通过交换机 450 可能经过的两条路径：正常路径 480 和检验路径 490。显然，在通常情况下分组遵循正常路径 480，例如从节点 410(1) 通过转发引擎 460 到预期目的地节点 410(2)。这些分组是交换机 450 所处理的分组中的绝大多数，因此由交换机 450 所执行的如图 3 所示的处理对这种流量的流动几乎没有影响。然而，对于所关心的层间绑定协议分组，遵循的是另一条路径，即检验路径 490。在以下的检验路径 490 中，例如将这种分组从给定节点（例如节点 410(1)）向另一个这种节点（例如节点 410(2)）发送，但由于它们是层间绑定协议分组，因此被转发引擎 460 转发到检验引擎 470 以进行检验。如果检验引擎 470 发现所述分组是可接受的，则检验引擎 470 随后将当前被检验的分组返回到转发引擎 460，以用于转发到其目的地（例如节点 410(2)）。

图 5 的流程图更详细地示出了交换机 450 和与本发明实施方式的描述有关的细节。如图 5 所示，交换机 450 包括多个线路卡（线路卡 500(1)-(N)），所述线路卡经由数据总线 530 和结果总线 540（和图 4B 一样）耦合到转发引擎（FE）510（和图 4B 中的转发引擎 460 一样）和处理器 520（和图 4B 中的检验引擎 470 一样）。线路卡 500(1)-(N) 中的每个线路卡都包括多个端口设备（在图 5 中示出为端口 ASIC 550(1,1)-(N,N)），所述端口设备由端口控制器来控制（在图 5 中示出为由端口 ASIC 控制器 560(1)-(N) 中的各个端口 ASIC 控制器来控制），所述线路卡还可包括其他结构。

如结合图 4B 所述，在根据本发明的系统中，尽管非 ILBP 分组仅仅被转发到其预期端口，但 ILBP 分组被识别并分析。根据图 5，在接收时，将（任意类型的）分组从端口 ASIC 550(1,1)-(N,N) 当中接收到该分组的那一

个端口 ASIC 发送到耦合于数据总线 530 的设备（例如端口 ASIC 550(1,1)-(N,N)当中的其他端口 ASIC 和 FE 510）。FE 510 随后确定该分组是不是 ILBP 分组。如果不是，则 FE 510 将该（非 ILBP）分组转发到端口 ASIC 550(1,1)-(N,N)当中预期的那一个端口 ASIC，这是通过向端口 ASIC 控制器 560(1)-(N)中相应的一个端口 ASIC 控制器指示出“保存在端口 ASIC 550(1,1)-(N,N)中给定的一个端口 ASIC 当中的分组的拷贝应当在相应的端口上发出”来实现的。

然而，如果所述分组是 ILBP 分组，则将该（ILBP）分组转发到处理器 520 以进行检验。处理器 520 检验所述分组（在软件控制下（这取决于专用处理器 520 具有怎样的体系结构，硬件和软件实现之间的设计决定等等）检验到一定程度）。如果所述分组包含不合法的绑定（例如具有 IP 和 MAC 地址间不合法绑定的坏 ARP 分组），则处理器 520 向 FE 510 和端口 ASIC 控制器 560(1)-(N)指示出“所考虑的分组应被丢弃”。因此，所述分组不会从交换机 450 发出。或者，如果所述分组包含合法绑定，则处理器 520 向 FE 510 和端口 ASIC 控制器 560(1)-(N)指示出“所考虑的分组是可接受的”，并且 FE 510 向端口 ASIC 控制器 560(1)-(N)中相应的一个端口 ASIC 控制器指示出“保存在端口 ASIC 550(1,1)-(N,N)中给定的一个端口 ASIC 当中的分组的拷贝应当在相应的端口上发出”。

图 6 的框图更详细地示出了转发引擎（FE）510 的特征。与图 5 相同，FE 510 在图 6 中被示出为耦合到数据总线 530 和结果总线 540。FE 510 经由数据总线接口 600 与数据总线 530 接口。以类似方式，FE 510 经由结果总线接口 610 与结果总线 540 接口。FE 510 中包含这样的结构，即查找表 620 在输入端处耦合到数据总线接口 600，并在输出端处耦合到结果总线接口 610，FE 510 还可包含其他结构。查找表 620 经由数据总线接口 600 接受来自数据总线 530 的分组，并在对给定分组执行分析之后，经由结果总线接口 610，在结果总线 540 上对交换机 450 的多种结构提供控制。查找表 620 由查找控制器 630 来控制，而查找控制器 630 又与处理器接口 640 相接口。

图 7 的流程图示出了在支持图 5 所示结构的网络元件上实现的例如图

3 所示的处理。因此，将根据图 5 和图 6 中所出现的结构来讨论图 7 所示的动作。图 7 所示的处理始于端口 ASIC (1,1)-(N,N)中的一个端口 ASIC 接收到分组（步骤 700）。该分组将会预期去往附接到端口 ASIC (1,1)-(N,N)中另一端口 ASIC 上的另一网络节点，所述另一端口 ASIC 通常是另一线路卡(1)-(N)上的端口 ASIC (1,1)-(N,N)之一。

在本发明和交换机 450 的某些实施方式中，所考虑的分组被传递到交换机 450 内所有可能的目的地（例如包括处理器 520、端口 ASIC 550(1,1)-(N,N)当中的其他端口 ASIC 和 FE 510）（步骤 720）。在这种情形下，尽管所有（或者一部分）可能的位置接收到了给定分组，妨碍了对实际处理该给定分组的指令（或控制信号）的接收，但这些目的地将会仅仅丢弃或覆写所述分组，忽略其接收。接下来，接收到给定分组之后，FE 510 接着对该分组进行预检验分析（步骤 730）。取决于 FE 510 是如何编程的，以及要检验的层间绑定协议分组，FE 510 此时可以采取多种动作。

尽管希望实现合理的大量这种处理，但对本领域技术人员来说很清楚的是，还希望能够以一种根据本发明实施方式的方式来加载在分析分组时所使用的规则，这是因为使用基于软件的规则可以容易地进行更新、动态分析、由于使用更简单的硬件而降低成本，还有其他这方面的优点。然后，确定接收到的分组是不是层间绑定协议分组（例如在图 7 的情况下是 ARP/RARP 分组）（步骤 740）。通常，在多数情况下，接收到的分组不会是层间绑定协议分组，因而在转发引擎生成结果之后，就将所述分组发送到端口 ASIC 550(1,1)-(N,N)中预期的一个端口 ASIC，以用于传输到预期节点（步骤 750）。这样，对分组进行的用于确定层间绑定是否可允许的分析不是典型分组的关键路径，因而不会对通过交换机 450 的分组流动产生大的影响。然而，如果所述分组是层间绑定协议分组（仍例如是 ARP/RARP 分组），则发送到处理器 520 的分组的拷贝由处理器 520 进行分析，以确保其中所断言（assert）的层间绑定是合法的（步骤 760）。如果所述分组中所断言的层间绑定是可接受的（步骤 770），则与从前一样，由预期的端口 ASIC 来发送所述分组（步骤 750）。如果确定所述分组是不可接受的（步骤 770），则对所述的坏分组进行处理（步骤

780)。

这种处理可以采取多种形式中的一种，所述形式包括在分析之后仅仅
丢弃分组。遗憾的是，一旦不友好者（例如黑客）掌握了这种区分伪造的
层间绑定的能力，则该不友好者随后就可以通过发送过于大量的这种分组
5 从而试图使对这种分组的分析泛滥，以试图压倒处理器 520 所执行的处
理。在这种情况下，可以设计对坏分组的处理以应付这种情况，如参照图
8 而更详细地描述的那样。

图 8 是用于对层间绑定协议 (ILBP) 分组进行速率限制，以避免被这
种分组的泛滥压倒的处理的流程图。图 8 的处理开始于确定这种分组的总
10 数是否已超过一个限度，所述限度可以是预设的或动态的（步骤 800）。
前面提到过，这种基于软件的分析的优点在于能够动态地对不同的分组流
量条件进行响应，因而在执行这种分析时考虑到网络状况。如果分组总数
超过了这个限度（步骤 800），则可以仅仅丢弃所述（多个）分组（步骤
810）。

应当注意到，FE 510 可以基于规则来执行多种动作，所述规则例如指
示出 FE 510 是否应丢弃分组（在其由处理器 520 分析之前），或者是使接
收到坏分组的流入端口关闭。在图 8 所示的例子中，如果接收到的分组的
总数（例如在给定单位时间内）超过了当前设置的限度（步骤 800），则
FE 在将分组发送到处理器 520 以用于分析之前仅仅丢弃分组，并向端口
20 ASIC 控制器 560(1)-(N)通知，端口 ASIC 550(1,1)-(N,N)中的一个或多个端
口 ASIC 所可能保存的坏分组的拷贝也应被丢弃（步骤 810）。

如果接收到的分组的总数（例如在给定单位时间内）未超过当前设置
的限度（步骤 800），则确定是否超过了第二限度，所述第二限度在这里
称为端口关闭限度（步骤 820）。这种条件可以指示出多种问题中的任意
25 一种，包括恶意或错误地引起的分组风暴。如果超过了端口关闭限度，则
关闭引起麻烦的（流入）端口（步骤 830）。这有效地将生成大量 ILBP 分
组的节点与网络的其余部分以及交换机 450 的内部结构（具体而言，FE
510 和处理器 520）断开。这例如防止了已经检测到发送以实现嗅探的分
组已被丢弃的第三方造成过载情况，以便迫使交换机接受坏的 ILBP 分组

等等。而且，如果接收到的分组的总数（例如在给定单位时间内）未超过当前设置的限度（步骤 800），则还确定是否超过了第三限度，所述第三限度在这里称为端口丢弃阈值（步骤 840）。这里，不合法的 ILBP 分组的数量超过了某个限度，因而应当丢弃引起麻烦的（多个）分组（步骤 5 810）。如果并未超过端口丢弃阈值，并且未关闭端口，则将所述（多个）分组传递到检验引擎（例如处理器 520）以进行检验（步骤 850）。

尽管已经示出和描述了本发明的特定实施方式，但对本领域技术人员来说显而易见的是，基于这里的教导，可以进行变化和修改，而不会脱离本发明及其更广的方面，因此所附权利要求的范围应覆盖所有本发明真正精神和范围之内所有这种变化和修改。而且，尽管参照这些具体实施方式而具体示出和描述了本发明，但本领域技术人员将会理解到，可以对其中的形式和细节进行前述及其他变化，而不会脱离本发明的精神和范围。10

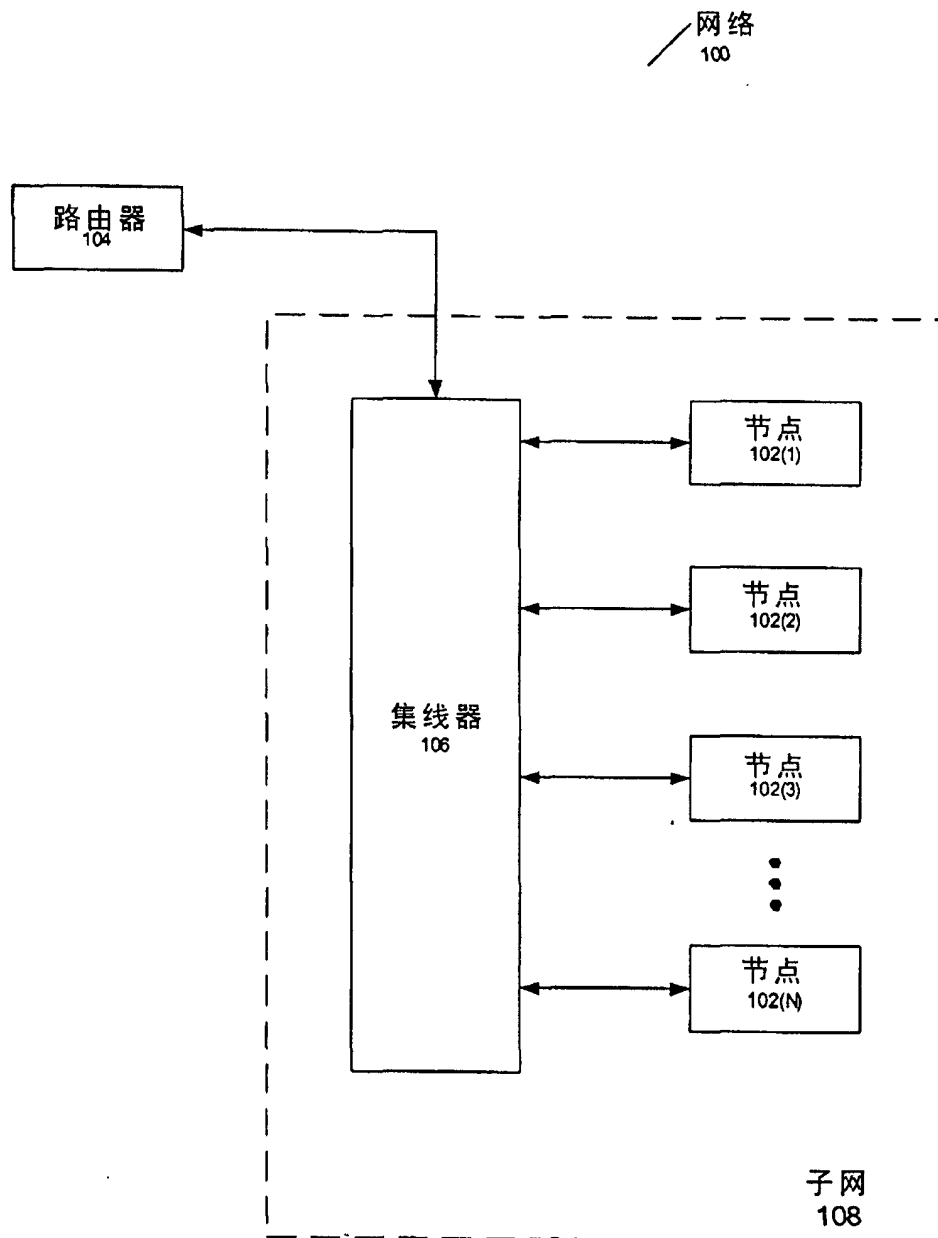


图1A
(现有技术)

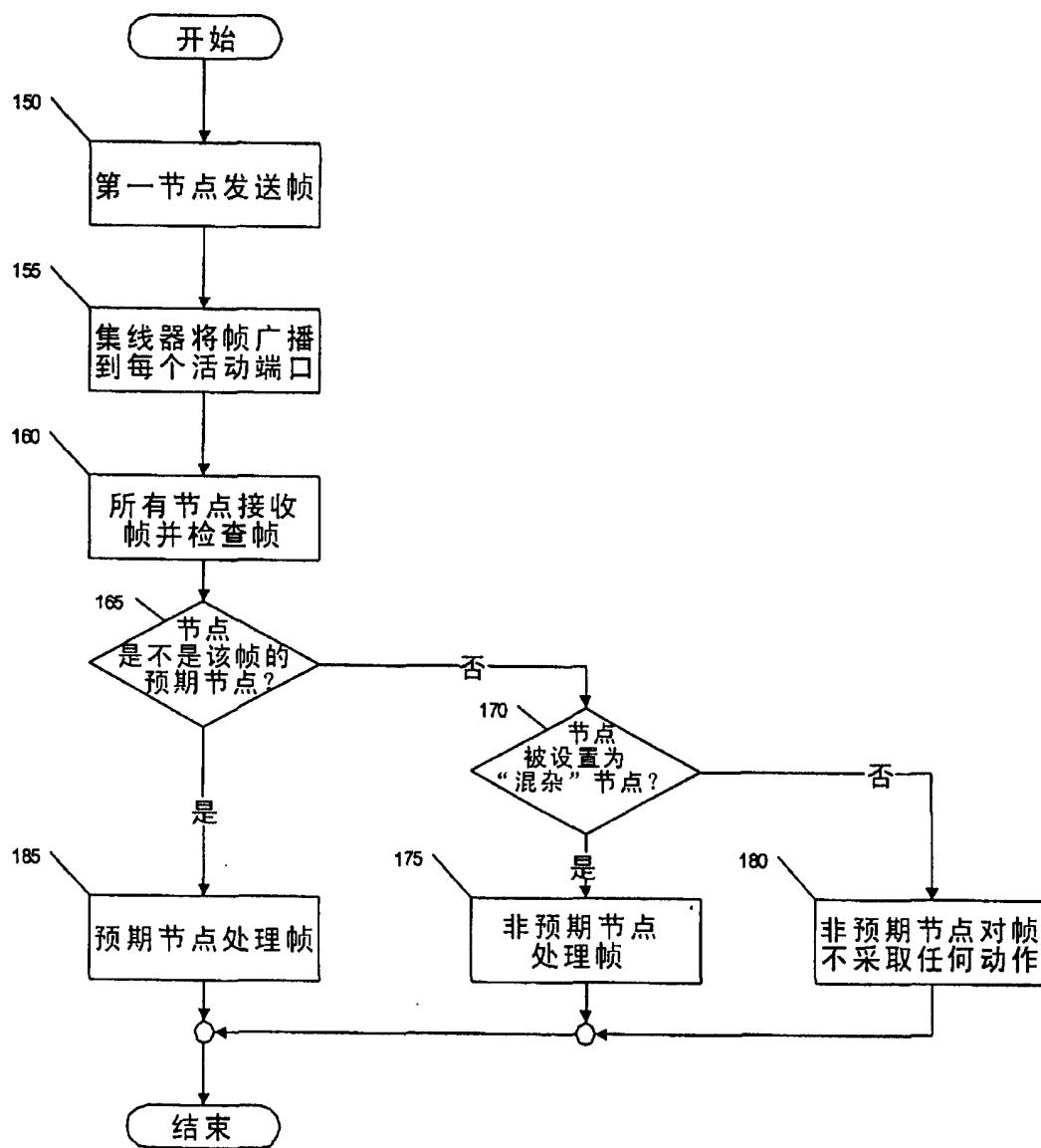


图1B
(现有技术)

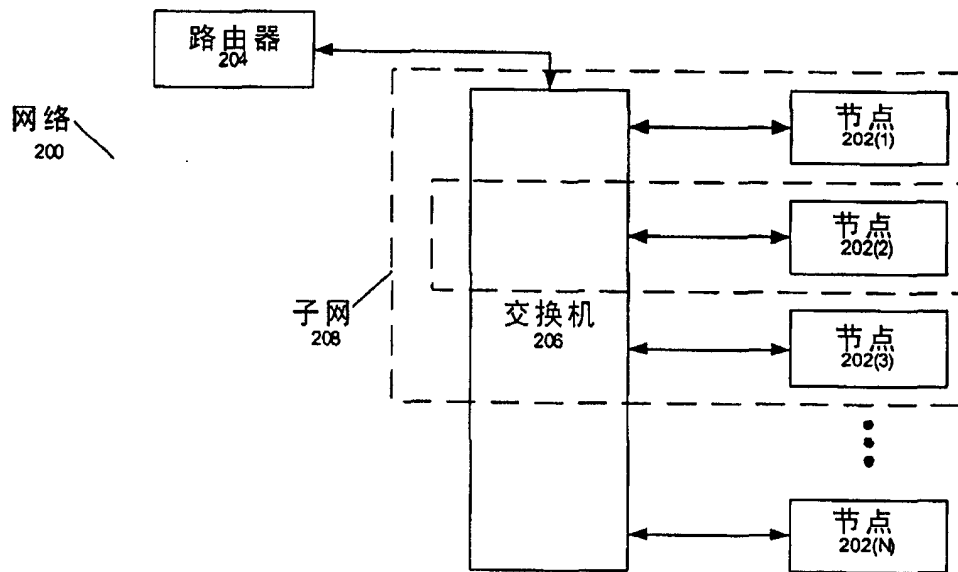


图2A
(现有技术)

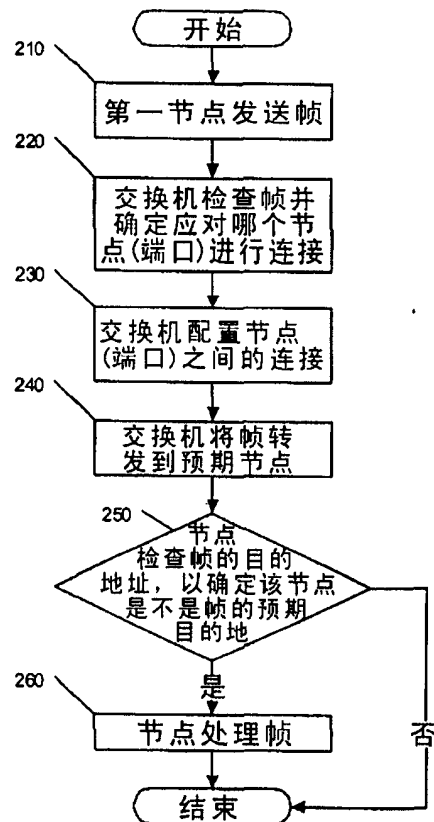


图2B
(现有技术)

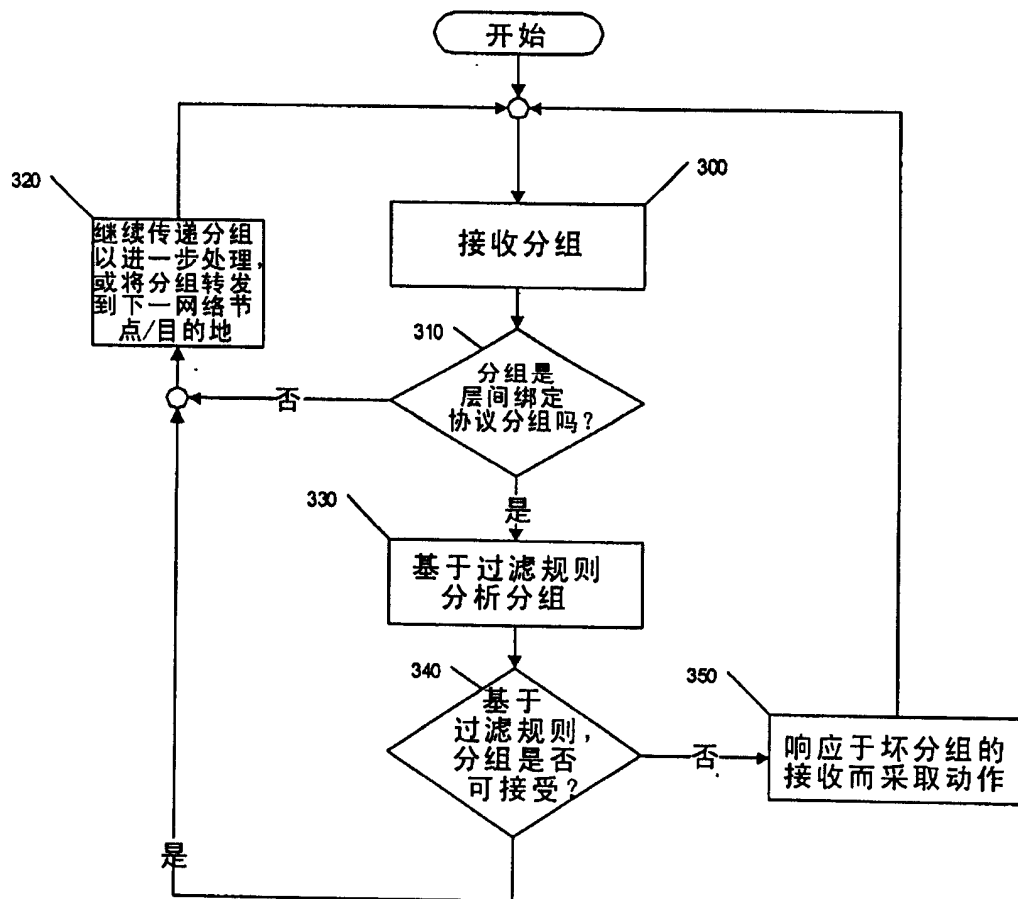


图3

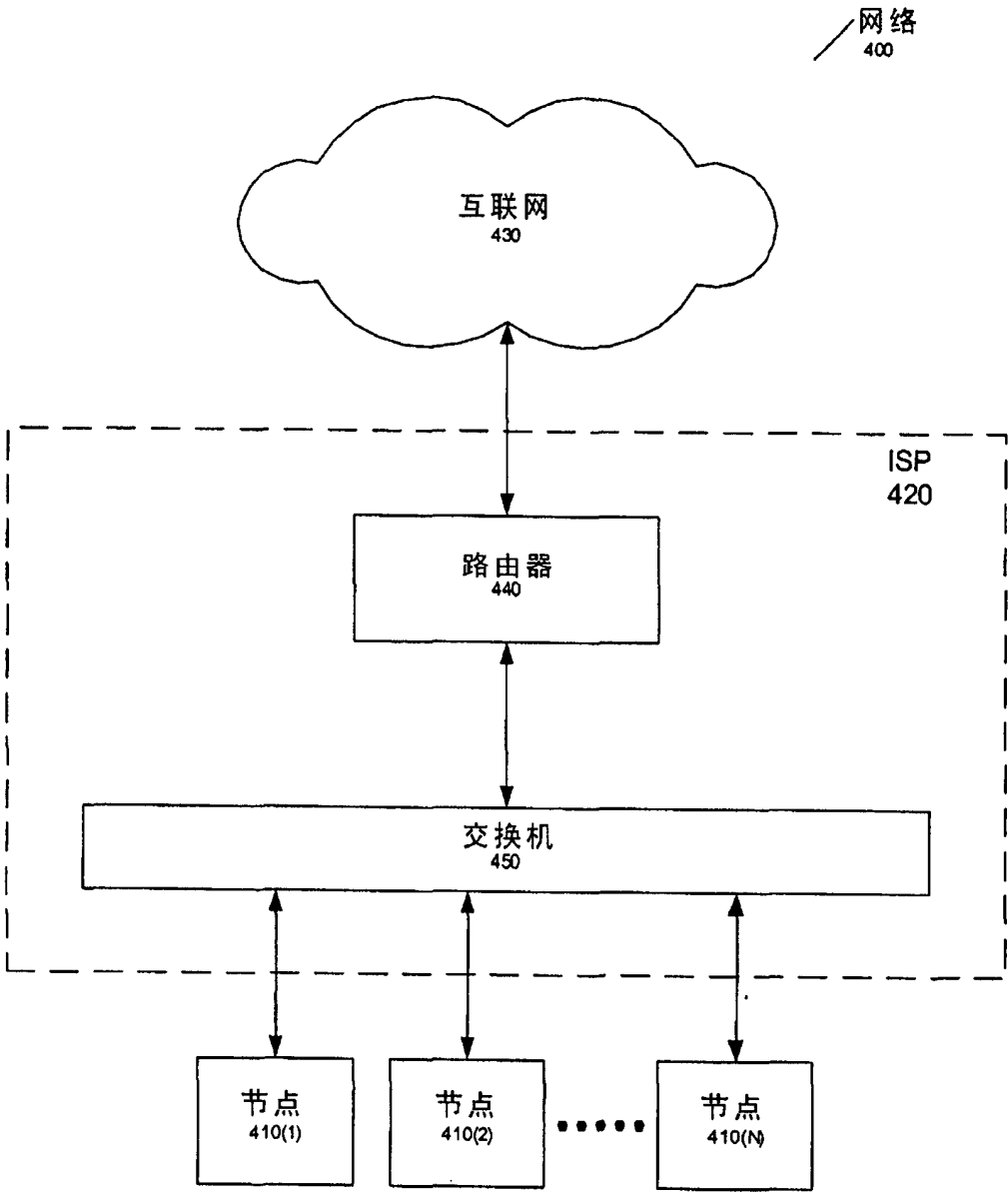


图4A

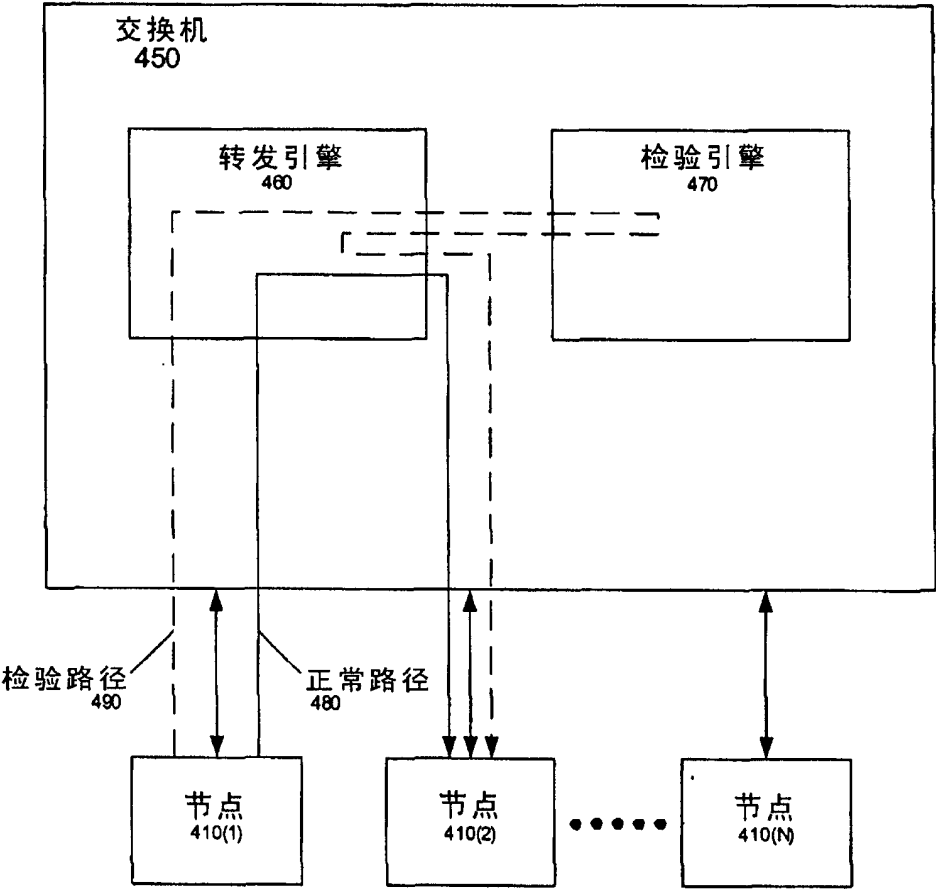


图4B

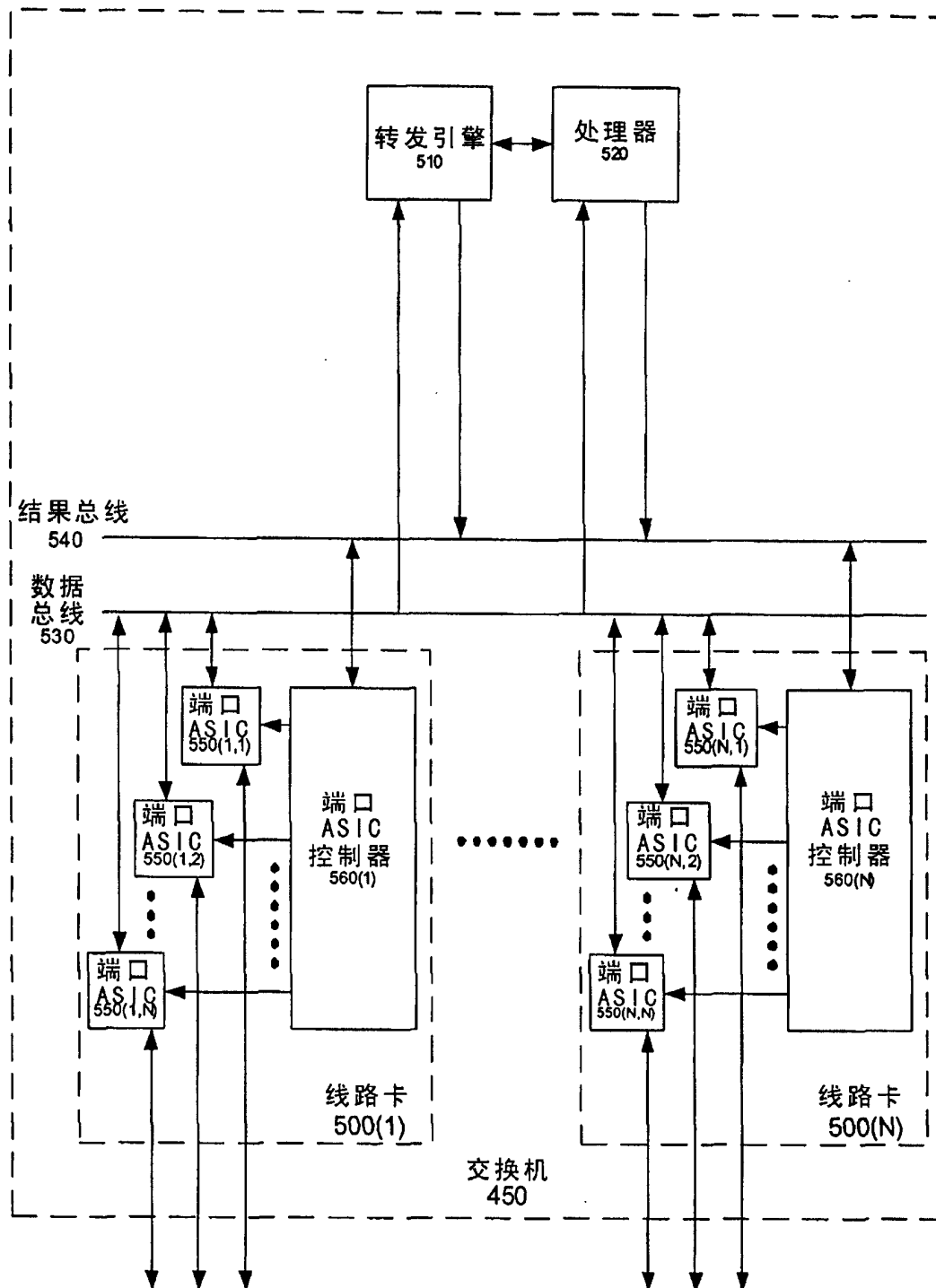


图5

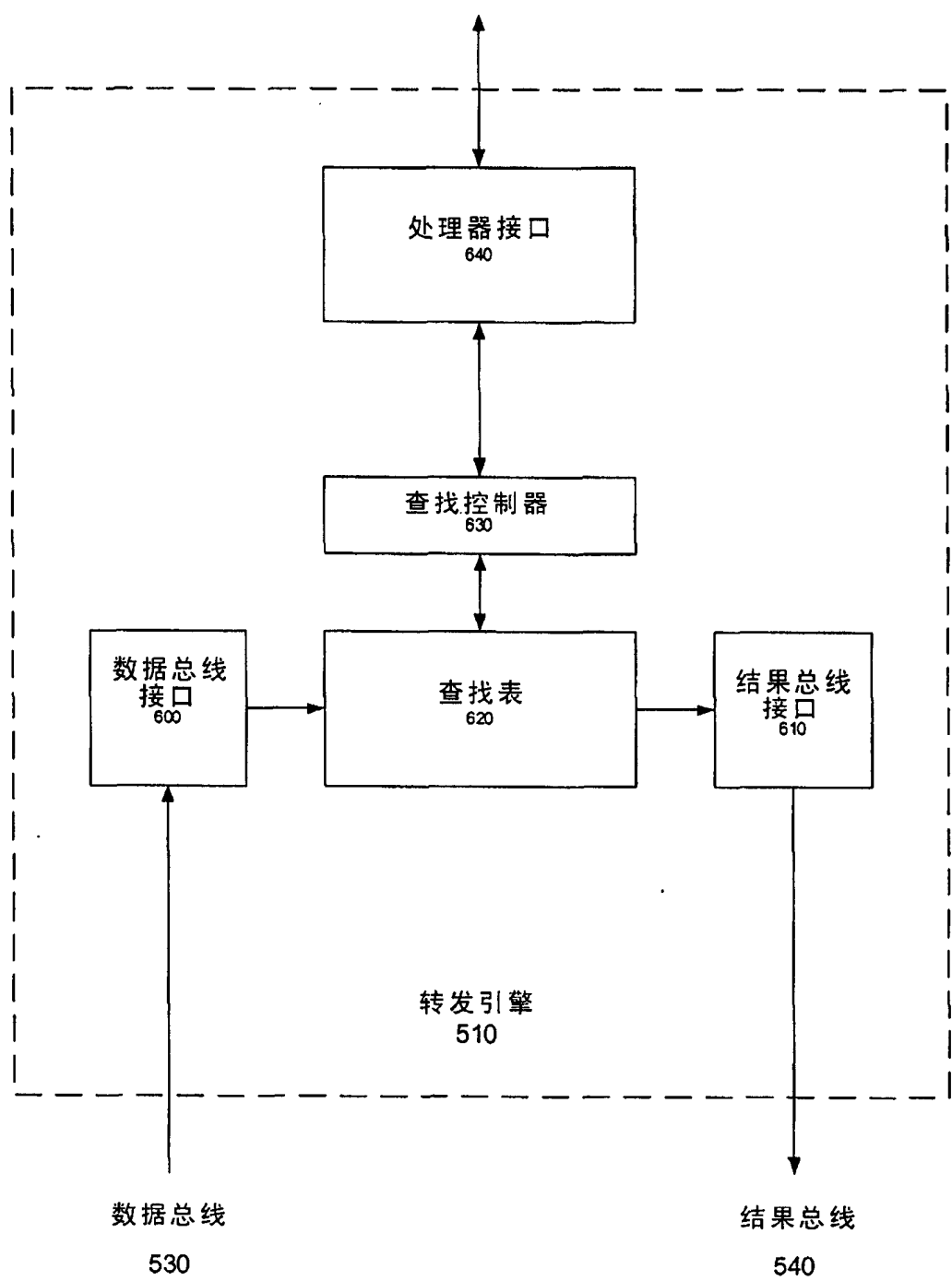


图6

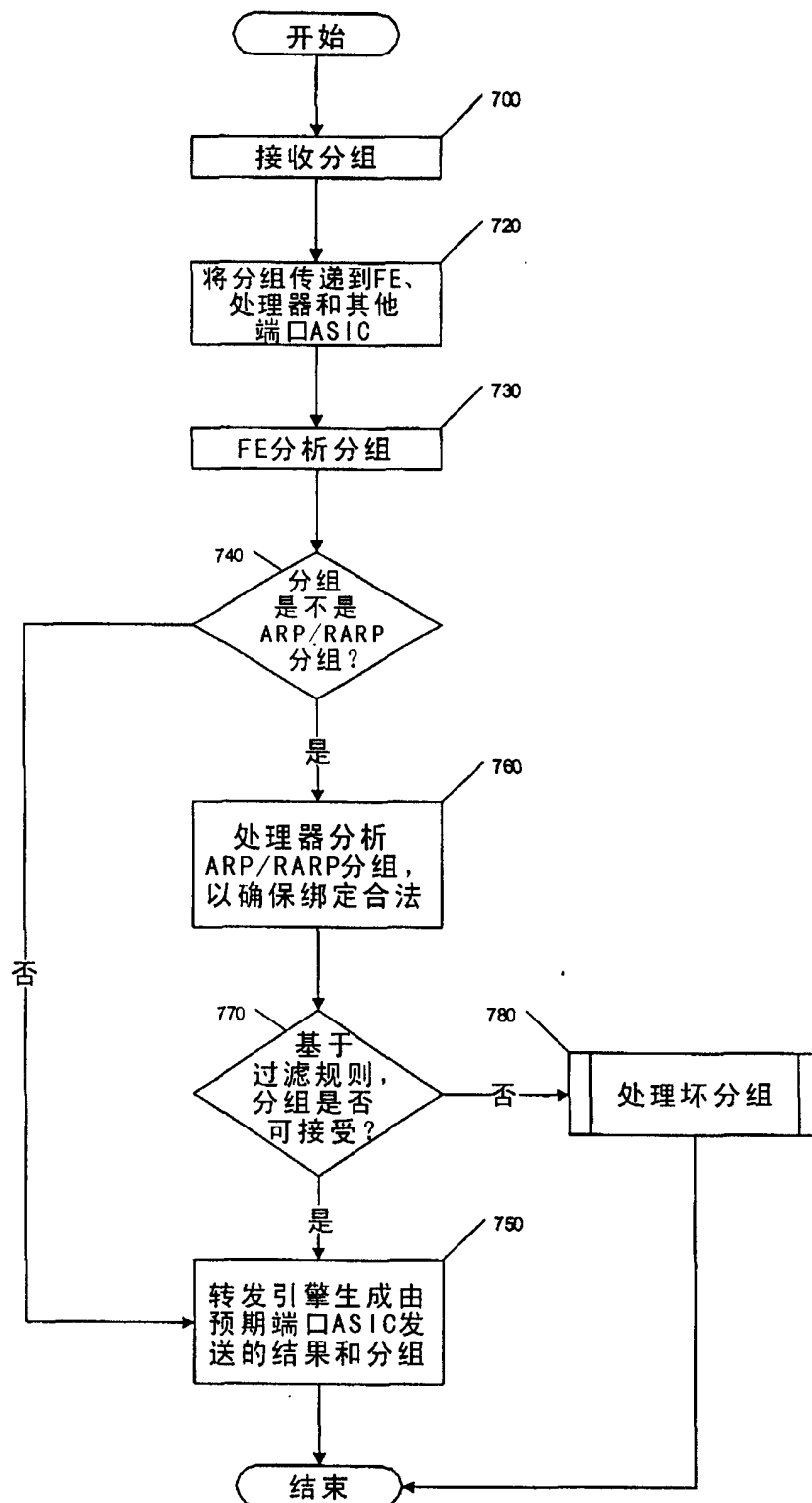


图7

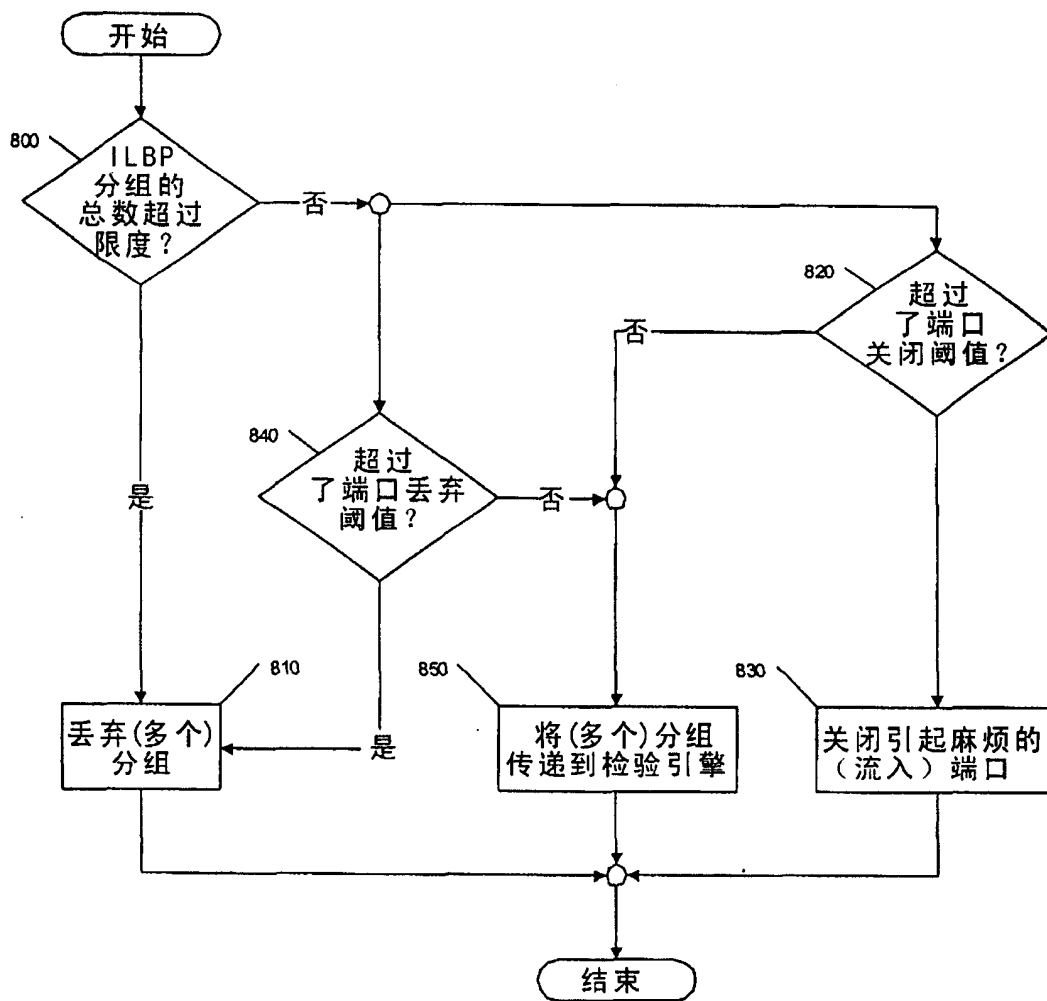


图8