



[12] 发明专利申请公开说明书

[21] 申请号 02817388.0

[43] 公开日 2004 年 12 月 1 日

[11] 公开号 CN 1552030A

[22] 申请日 2002.9.6 [21] 申请号 02817388.0

[30] 优先权

[32] 2001. 9. 6 [33] US [31] 60/317,719

[32] 2002. 1. 8 [33] US [31] 60/347,050

[32] 2002. 4. 22 [33] US [31] 10/127,888

[86] 国际申请 PCT/US2002/028467 2002.9.6

[87] 国际公布 WO2003/023638 英 2003.3.20

[85] 进入国家阶段日期 2004.3.5

[71] 申请人 阿瓦雅技术公司

地址 美国新泽西州

[72] 发明人 克里斯托弗·高林奇

阿列克斯·M·克鲁姆-海勒

詹姆斯·D·施鲁德尔

姆内勃·敏哈祖丁

[74] 专利代理机构 中国国际贸易促进委员会专利

商标事务所

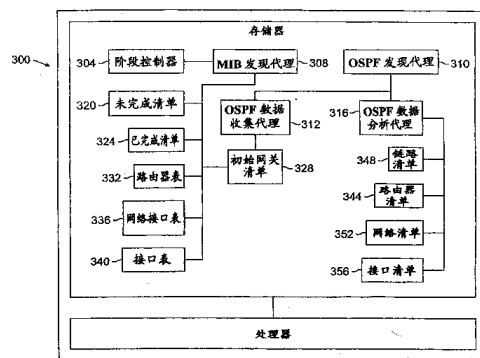
代理人 付建军

权利要求书 6 页 说明书 18 页 附图 16 页

[54] 发明名称 通过分阶段多次发现方法进行拓扑结构发现

[57] 摘要

本发明公开了通过分阶段多次发现方法进行拓扑结构发现。一种用于发现一个分布式处理网络的拓扑结构的系统，包括：第一拓扑结构管理信息库发现代理 308，用于接触第一组路由器以获得一个存储在第一组路由器中的各个路由器内的第一类信息；一个第二拓扑结构开放式最短路径优先发现代理 312 和/或 316，用于接触第二组路由器以获得一个存储在第二组路由器中的各个路由器内的第二类信息；一个阶段控制器 304，用于在第一(308)和第二(312)和/或 316 拓扑结构发现代理之间进行选择。第一组与第二组路由器是不相同的，第一类与第二类信息也是不相同的。在一种配置中，第一类信息由一个网络管理协议定义，而第二类信息则由一个路由选择协议定义。



1. 一种用于发现一个分布式处理网络的拓扑结构的方法，该方法包括：

(a) 从第一组路由器检索存储在第一组路由器中的每个路由器内的第一类信息；

(b) 从第二组路由器检索存储在第二组路由器中的每个路由器内的第二类信息，其中第一和第二组的路由器不同，第一和第二类信息也不同。

2. 根据权利要求 1 的方法，其中第一组路由器至少包括第二组内的一些路由器。

3. 根据权利要求 1 的方法，其中第一类信息包括一个或多个管理信息库内的变量，而且第二类信息包括由路由选择协议定义的一个或多个变量。

4. 根据权利要求 1 的方法，还包括：

在接触步骤 (a) 和 (b) 期间维护并理解未接触路由器和/或路由器接口的未完成清单。

5. 根据权利要求 4 的方法，还包括：

在接触步骤 (a) 和 (b) 期间维护一个被接触路由器和/或路由器接口的已完成清单。

6. 根据权利要求 1 的方法，还包括：

检索一个与一个所选择的路由器和/或接口相关联的下一跳点接口的地址；

确定一个由下一跳点地址所支持的路由选择协议；

为所选择的路由器和/或接口和下一跳点接口的地址中的至少一个指定一个用于指示路由选择协议的标识符。

7. 根据权利要求 1 的方法，还包括：

在接触步骤 (a) 维护一个初始网关清单，以便在步骤 (b) 使用。

8. 根据权利要求1的方法, 还包括:

依据第一类和第二类信息中的至少一个, 识别与分布式处理网络相关联的一个或多个接口对象、路由器对象以及网络对象。

9. 根据权利要求1的方法, 其中在检索步骤(a)第一类信息由一个网络管理协议定义, 在检索步骤(b)第二类信息由一个路由选择协议定义。

10. 一种用于发现一个分布式处理网络的拓扑结构的方法, 包括:

(a) 从第一组路由器检索存储在第一组路由器中的每个路由器内的第一类信息, 第一类信息由一个网络管理协议定义;

(b) 从第二组路由器检索存储在第二组路由器中的每个路由器内的第二类信息, 第二类信息由一个路由选择协议定义;

11. 根据权利要求10的方法, 其中第一组和第二组路由器是不同的, 而且第一和第二类信息也不同。

12. 根据权利要求11的方法, 其中第一组路由器至少包括一些第二组路由器中的路由器。

13. 根据权利要求10的方法, 其中第一类信息包含网络管理信息库内的由一个简单网络管理协议定义的一个或多个变量, 第二类信息包括一个由路由选择协议定义的一个或多个变量。

14. 根据权利要求10的方法, 还包括:

在接触步骤(a)和(b)期间维护一个未接触路由器和/或路由器接口的未完成清单。

15. 根据权利要求14的方法, 还包括:

在接触步骤(a)和(b)期间维护一个已接触路由器和/或路由器接口的已完成清单。

16. 根据权利要求10的方法, 还包括:

检索一个与所选择的路由器和/或接口相关联的下一跳点接口的地址;

确定一个由下一跳点地址支持的路由选择协议;

为所选择的路由器和/或接口和下一跳点接口的地址中的至少一个指定一个用于指示路由选择协议的标识符。

17. 根据权利要求 10 的方法, 还包括:

在接触步骤 (a) 期间维护一个网关清单, 供接触步骤 (b) 使用。

18. 根据权利要求 10 的方法, 还包括:

依据第一类和第二类信息中的至少一个, 识别与分布式处理网络相关联的一个或多个接口对象、路由器对象以及网络对象。

19. 根据权利要求 1 的方法, 还包括:

(c) 通过执行至少以下步骤之一来处理第一组路由器中的至少一个路由器:

检索一个存储在被处理路由器内的一个接口地址清单并为接口地址清单内的每个接口地址重复进行以下步骤:

(1) 选择一个与所选择接口地址连接的下一跳点接口的地址;

(2) 识别与所选择接口地址相关联的路由器对象、接口对象、网络对象和链路对象中的至少一个;

(3) 识别一个与所选择的下一跳点接口相关联的协议;

(4) 为每个与所选择接口地址相关联的下一跳点接口重复步骤 (1) ~ (3)。

20. 一种用于发现一个分布式处理网络的拓扑结构的系统, 包括:

(a) 第一装置, 用于接触第一组路由器以获得一个存储在第一组路由器中的各个路由器内的第一类信息;

(b) 第二装置, 用于接触第二组路由器以获得一个存储在第二组路由器中的各个路由器内的第二类信息, 其中第一组和第二组路由器是不同的, 而且第一和第二类信息也不同。

21. 根据权利要求 20 的系统, 其中第一组路由器至少包括一些第二组路由器中的路由器。

22. 根据权利要求 20 的系统, 其中第一类信息包含网络管理信息

库内的一个或多个变量，第二类信息包括一个或多个由一个路由选择协议定义的变量。

23. 根据权利要求 20 的系统，还包括：

装置，用于维护未接触路由器和/或路由器接口的未完成清单，供第一接触装置使用。

24. 根据权利要求 23 的系统，还包括：

装置，用于维护已接触路由器和/或路由器接口的已完成清单。

25. 根据权利要求 20 的系统，其中第一接触装置包括：

检索装置，用于检索一个与所选择路由器和/或接口相关联的下一跳点接口的地址；

确定装置，用于确定一个由下一跳点地址支持的路由选择协议；

指定装置，用于为所选择的路由器和/或接口和下一跳点接口的地址中的至少一个指定一个用于指示路由选择协议的标识符。

26. 根据权利要求 20 的系统，还包括：

装置，用于维护一个初始网关清单，供第二接触装置使用。

27. 根据权利要求 20 的系统，还包括：

装置，依据至少一个第一和第二类信息，识别一个或多个与分布式处理网络有关的接口对象、路由器对象和网络对象。

28. 根据权利要求 20 的系统，其中第一类信息由一个网络管理协议定义，第二类信息由一个路由选择协议定义。

29. 一种用于发现一个分布式处理网络拓扑结构的系统，包括：

(a) 第一拓扑结构发现代理，配置为可接触第一组路由器以获得存储在第一组路由器中的每个路由器内的第一类信息；

(b) 一个第二拓扑结构发现代理，配置为可接触第二组路由器以获得存储在第二组路由器中的每个路由器内的第二类信息，其中第一组和第二组路由器是不同的，而且第一和第二类信息也不同；

(c) 一个阶段控制器，配置为可在第一和第二拓扑结构发现代理之间进行选择。

30. 根据权利要求 29 的系统，其中第一组路由器至少包括一些第

二组路由器中的路由器。

31. 根据权利要求 29 的系统，其中其第一类信息包含网络管理信息库内的一个或多个变量，第二类信息包括一个或多个由一个路由选择协议定义的变量。

32. 根据权利要求 29 的系统，其中至少配置了第一和第二拓扑结构发现代理和阶段控制器，用于维护一个未接触路由器和/或路由器接口的未完成清单，供第一拓扑结构发现代理使用。

33. 根据权利要求 29 的系统，其中至少配置了一个阶段控制器和第一与第二拓扑结构发现代理，用于维护一个已接触路由器和/或路由器接口的已完成清单。

34. 根据权利要求 29 的系统，其中第一拓扑结构发现代理被配置用于检索一个与所选择路由器和/或接口相关联的下一跳点接口的地址，确定一个由下一跳点地址支持的路由选择协议，并为至少一个所选择的路由器和/或接口以及下一跳点接口的地址指定一个用于指示路由选择协议的标识符。

35. 根据权利要求 29 的系统，其中至少配置了一个阶段控制器和第一与第二拓扑结构发现代理，用于维护一个初始网关清单，供第二接触装置使用。

36. 根据权利要求 29 的系统，其中至少配置了一个阶段控制器和第一与第二拓扑结构发现代理，用于识别与分布式处理网络相关联的一个或多个接口对象、路由器对象以及网络对象。

37. 一种网络模型，包括：

多个接口标识符，每个标识符都与一个相应的路由器相关联；

多个网络标识符；

多个路由选择协议标识符，每个路由选择协议标识符与至少一个路由器和一个接口相关联，而且可以指示一个由至少一个路由器和接口支持的路由选择协议。

38. 根据权利要求 37 的网络拓扑结构模型，其中每个接口标识符都是一个网络上的电子地址。

39. 根据权利要求 37 的网络拓扑结构模型，其中每个接口标识符都是一个网络地址。

40. 根据权利要求 37 的网络拓扑结构模型，其中与路由选择协议标识符相关联的协议至少包括两个距离向量算法、链路状态算法以及它们的组合。

41. 一种用于确定与企业网络有关的拓扑结构的方法，包括：

识别至少一个在企业网中使用的路由选择协议；

根据所识别的路由选择协议在多个数据收集代理中选择一个数据收集代理来收集网络拓扑结构信息，其中多个数据收集代理中的每一个都被配置成支持一个不同的协议。

42. 根据权利要求 41 的方法，其中企业网络使用第一和第二路由选择协议，而第一路由选择协议与第一数据收集代理相关联，第二路由选择协议与第二个不同的路由选择协议相关联，该方法还包括：

利用第一数据收集代理收集第一组网络信息，并利用第二数据收集代理收集第二组不同的网络信息。

43. 根据权利要求 42 的方法，还包括：

在第一网络模型内形成第一组网络信息，并在一个不同的第二网络模型内形成第二组网络信息。

44. 根据权利要求 41 的方法，还包括：

将第一和第二网络模型合并为一个单一网络模型。

通过分阶段多次发现方法进行拓扑结构发现

技术领域

本发明总体上涉及网络，具体而言，本发明涉及确定网络或路由选择拓扑结构的方法。

背景技术

分布式处理网络在我们以信息为基础的社会中变得越来越重要。图 1 描述了一个简单数据网络的网络拓扑结构。网络 100 包含多个路由器 104a-g、一个转接网络 108 和一个桩基网络 112，全部通过链路 116a-i 互相联接。正如后面所要说到的，路由器是一种连接两个或两个以上网络的装置，它为输入的数据确定传输路径或分组传输到适当的网络或节点；转接网是一种不仅仅包含路由器的网络；桩基网络是一种没有被配置为通过网络从一个路由器向另一个路由器传送分组报文的网络；而链路则是两个或两个以上节点之间的通信信道。每个路由器一般都通过一个或几个接口，例如接口 120a-n，与一条链路连接。图 1 的简单网络被一条虚线 124 划分为两个协议区，使该线成为两区的边界线。路由器 104c 处于边界线 124 上，所以称作区域边界路由器，而其它路由器 104a-b 和 104d-g 则不是边界路由器。一个或几个协议区域常常是自治系统。自治系统是受一个行政管理机构控制的网络群。

在一个报文分组交换网络中，通过互连网络传送分组报文的方法取决于路由选择协议。大部分协议都属于距离向量算法和链路状态算法之中的一类。距离向量算法通过计算一份分组报文从源网络传送到目的网络所要经过的路由器跳点数量来确定源节点与目的节点之间的距离。而链路状态算法则利用链路状态通告，即 LSA（包含一个路由器邻居的名称和各种到达的成本尺度），使路由器始终了解网络链路的情况。链路状态算法不像距离向量算法那样存储实际路径分类，而是存储所需信

息，以产生这样的路径。利用距离向量算法的路由器协议例子包括 RIP 和 RIP-2，利用链路状态算法的路由器协议例子包括开放式最短路径优先协议（即 OSPF）、开放系统互联（OSI）的中间系统对中间系统协议（IS-IS）、增强的内部网关路由选择协议（EIGRP）以及 NetWare 链路服务协议（NLSP）。

路由器和其它网络组成部分一般利用网络管理系统进行管理。网络管理系统进行网络维护，识别网络内可能出现的安全问题，确定设备、模块、部件以及插件板的故障部位，确定电路断线部位，监控性能级别（例如误码率，即 BER，和同步损失等）并允许快速和精确地判定网络的使用和通信量级别。用于执行以上任务的网络管理系统的例子包括惠普公司的 OpenViewTM、IBM 公司的 NetviewTM 和数字设备公司的企业管理体系结构，即 EMATM。

为使网络管理系统的运行达到最佳，一般需要精确、详细的网络布局图或 OSI 的网络层拓扑结构图。这些图不仅方便了网络管理系统的运行，而且也可使新连接的主机可以处在对于网络来说正确的位置上并正确配置（以避免影响网络的性能），并使现有主机处在对于新连接主机来说正确的位置上。通常的情况下，网络管理人员不能利用整幅或部分的详细网络拓扑结构图。其原因是不良的记录保持、某些网络规模庞大而复杂，以及缺乏网络的中心管理，例如当网络包含一些自治系统或企业的情况时。

简单网络管理协议，即用于自动发现网络层拓扑结构的 SNMP 算法在许多网络管理工具中得到使用。SNMP 算法可采用多种方法。在一种称作“逐跳法”的方法中，该算法在逐跳的基础上访问标准的路由 SNMP-管理信息库或每个路由器内的 MIB 信息。如此处所用，一个“跳点”被视作没有中间节点的一段路由，而 MIB（管理信息库）则是一组可以按 SNMP 的规定进行管理的一组被管理对象或变量。MIB 中的对象或变量一般由被称作管理信息结构，即 SMI，的一组规则来定义。如后面所述，MIB 信息存储在任何 SNMP 路由器的存储器内。另一种方法是，利用销售商专有算法来产生拓扑结构。这种解决方案的实例有

Cisco System 公司的 CDPTM。这种专有算法一般依靠销售商对标准的 SNMP 管理信息库所作的特别扩充，该管理信息库一般不能在多销售商网络中使用。

当一个网络中的路由器支持不同的路由选择协议，而且（或）不可接触，则 SNMP 网络拓扑结构发现算法一般不能查清网络层的拓扑结构。一个路由器不可接触有各种原因，包括使用了不正确的凭证、接触界面的不良状态以及在路由器中不可访问或不存在的 SNMP 代理等。图 2 对这一问题做出了说明。在图 2 中，路由器 200 支持路由信息协议（RIP）并且是 SNMP 可接触式，路由器 204、208 和 212 支持开放式最短路径优先（OSPF）协议并且是 SNMP 可接触式，路由器 216 支持 OSPF 协议但不是 SNMP 可接触式，而路由器 220 支持 OSPF 和 RIP，是 SNMP 可接触式。如果拓扑结构发现算法最初接触的是路由器 200，它将能用逐跳法发现路由器 200、220 和 216。当算法接触到不是 SNMP 可接触式的路由器 216 时，算法将不能发现路由器 204、208 和 212。这是因为算法将不能访问路由器 216 中的管理信息库（MIB）信息，所以无法了解这些路由器是否存在。

发明内容

通过本发明的各种实施例和附图将解决这些和那些需要。一般说来本发明的体系结构使用多种拓扑结构发现方法来发现网络拓扑结构。不同的发现方法可以是网络管理协议和/或路由选择协议专用的。使用网络管理协议专用的发现算法可以使算法发现变得不可知（即不敏感）。

在一个对企业网或自治系统极为有用的实施例中，一种逐跳发现算法，例如 SNMP 拓扑结构发现法，与一种或多种其它发现算法（例如 OSPF 发现算法、基于专有标准的发现算法以及销售商专有拓扑结构发现算法）相结合，以进行拓扑结构探测。拓扑结构发现法一般设计成对路由选择协议不敏感。这可以通过逐跳发现法成为可能，这时每个路由器都被接触，以发现被接触路由器所了解的网络中其它实体。每个路由器依次被接触，以建立与各个网络实体及其拓扑结构连接有关的数据

库。

无论何时，用逐跳法发现了一个路由器，就向其发出询问以查明是使用了哪些协议来确定其路由表的。如果这些协议有与其相关的专有发现算法，相应的发现算法即运行。任何被发现的其它数据都加入一个网络模型，任何被发现的路由器都可被逐跳算法利用以进行进一步的探测。利用替代发现算法发现的路由器可用来“跳过”不可接触的路由器，例如图 2 中的路由器 216。如图 2 所示，在使用 SNMP 发现算法时又使用 OSPF 拓扑结构发现算法，将会在路由器 204、208、212 和 216 中的任何一个上显露出 OSPF 区域信息。区域信息将鉴别不可接触路由器 216 的存在，但没有其它信息可被推导出来。

在另一个实施例中，体系结构使用了一些发现代理以发现执行各种不同路由选择协议的异构网络。每个发现代理与由一个或多个路由选择协议确定的路由器信息互动。各个发现代理可以在不连续的阶段运行、并行运行或逐个路由器运行。

本发明的体系结构有多种优点。例如，使用不同的发现方法，即便是在网络的组成部分支持不同的协议时也可确定网络组成部分的位置。这种优点可以参照使用了 MIB 和 OSPF 发现方法的体系结构配置来说明。在此处，MIB 被视作各种版本的管理信息库，SNMP 被视作各种版本的简单网络管理协议，而 OSPF 则为各种版本的 OSPF 协议。MIB 发现法的优点是无论存在何种路由选择协议，它都将发现 IP 网络的拓扑结构。该方法的缺点是，MIB 发现代理必需访问网络内的每个路由器以发现整个网络。如果一个路由器不能通过 SNMP 接触，正如前面参照图 2 所述，MIB 代理将只能发现一个网络子集。若仅使用 OSPF，将只能发现网络执行 OSPF 路由选择协议的部分。然而，OSPF 发现方法仅接触少量路由器（例如区域边界路由器）就可以查明一个网络的广大区域的情况，所以不受不可接触的路由器影响。这样，OSPF 发现方法可以发现 MIB 发现法不能发现的路由器，因为 OSPF 无须访问每一个路由器就可以确定一个网络的拓扑结构。特别是，OSPF 可以发现因不可接触的路由器阻塞 MIB 而无法到达的网络部分，正如前面参照图 2

所述。将两种发现方法一起使用，可以使 MIB 发现法发现网络中所有可接触的和不可接触的路由器而不管网络中存在的是什么路由选择协议。不可接触的网络部分常常可以通过使用 OSPF 发现法“跳过”一个不可接触的路由器而随后被 MIB 发现法发现。

这些优点和其它优点将随着本发明的公开而显示出来。

上述实施例和结构并不完全也不彻底，正如后面所述，本发明的其它实施例可能会在后面单独或综合地运用上面提到的一个或多个细节，或加以详细阐述。

附图简要说明

- 图 1 描述了依据现有技术的简单网络拓扑结构；
- 图 2 描述了依据现有技术的另一个简单网络拓扑结构；
- 图 3 是一幅框图，描述了依据本发明一个实施例的计算体系结构；
- 图 4 是一幅流程图，描述了相位控制器的操作；
- 图 5 是一幅流程图，描述了 MIB 发现代理的操作；
- 图 6 是描述了 MIB 发现代理操作的另一幅流程图；
- 图 7 是一幅流程图，描述了 OSPF 发现代理的操作；
- 图 8 是一个路由器表，由 MIB 发现代理输出；
- 图 9 是一个接口表，由 MIB 发现代理输出；
- 图 10 是一个网络接口表，由 MIB 发现代理输出；
- 图 11 是一个初始网关清单，由阶段控制器维护；
- 图 12 是一个未完成清单，由阶段控制器维护；
- 图 13 是一个已完成表清单，由相位控制器维护；
- 图 14 是一个路由器清单，由 OSPF 发现代理输出；
- 图 15 是一个的链路清单，由 OSPF 发现代理输出；
- 图 16 是一个网络清单，由 OSPF 发现代理输出；
- 图 17 是一个接口清单，由 OSPF 发现代理输出的；
- 图 18 是一个典型的 ipAddrTable (IP 地址表) 表；

图 19 是一个典型的 ipRouteEntry (IP 路由条目) 表;

图 20 是一个典型的 ipAddr (IP 地址) 表;

图 21 是一个典型的 ipRoute (IP 路由) 表;

图 22 是一个依据图 18、19 和 21 的 ipAddrTable、ipRouteEntry 和 ipRoute 表中所选择的条目而形成的网络拓扑结构;

图 23 是一幅流程图, 描述了依据本发明另一个实施例的多路由选择协议体系结构;

图 24 描述了 ipAddrEntry 与 ipRouteEntry 表之间的关系。

具体实施方式

为发现一个存在多种路由选择协议的网络的拓扑结构, 使用了一种分阶段的发现方法。需要有三个主要阶段来发现一个网络的拓扑结构。在初始网关发现阶段, 体系结构接触一个种子路由器以启动发现过程。在 MIB 或 MIB2 发现阶段, 体系结构接触企业网中的所有路由器, 以下载所选择的路由器中的 MIB 信息。在 OSPF 发现阶段, 体系结构接触支持 OSPF 的路由器, 以下载 OSPF 路由器中的链路状态通告数据库所发出的链路状态信息。

在开始讨论数据收集和分析代理 208 的操作之前, 了解许多路由选择协议的某些特点是很重要的。在使用某些协议的情况下, 可以依据唯一的路由器标识符来识别一个路由器, 而且路由器亦与唯一的区域标识符相关。一般一个路由器本身并没有 IP 地址。接口是一个属于主机, 例如路由器, 的逻辑装置而不只是一条链路的连接点。一般说来一个接口有 1 或零作为 IP 地址, 而且属于一个网络。接口一般将拥有一个 IP 地址和一个网络掩码。一条链路包括源接口的两个或两个以上绑定以及一个尺度, 即成本。它被尺度表示法模板化, 该尺度专用于路由选择协议, 并代表一个分组报文离开接口的成本。一条链路一般与一个成本尺度和一个路由选择协议标识符有关。一个网络对象代表一个数据网或子网。它有一个地址和一个掩码并代表一个容纳一组主机的地址空间。网络对象可以从其成员接口驱动其地址和/或掩码。

网络拓扑结构发现系统

考虑到这一点，依据本发明的一个实施例，图 3 涉及的是网络拓扑结构发现系统 300。系统 300 配置成与一个计算机网络，例如桩基网络 112，的接入点连接，以便向主机，一般是路由器，发送信息或从主机接收信息。系统 300 包括：一个阶段控制器 304，用于按适当的顺序监控 3 个阶段中的每一个阶段的执行，并确定发现过程何时完成；一个 MIB 发现代理 308，用于在 MIB 发现阶段访问每个可接触路由器中的 MIB 信息并产生一个描述一部分网络拓扑结构的 MIB 输出；一个 OSPF 发现代理 310，该发现代理包括一个 OSPF 数据收集代理 312，配置在 OSPF 发现阶段，通过在各个需要的路由区有选择地接触路由器而有选择地收集与网络拓扑结构有关的信息，还包括一个 OSPF 数据分析代理 316，用于在 OSPF 阶段对 OSPF 数据收集代理 312 所收集的信息进行分析并产生一个描述部分网络拓扑结构的输出。

在拓扑结构发现过程中，系统 300 保持一些列表，包括未完成清单、已完成清单、初始网关清单、区域边界路由器表（未标出）和一个链路状态通告表（未标出）等，以避免重复计算。未完成清单 320（图 12）列出了还未连接的候选主机（接口）地址 1200。在 MIB 和 OSPF 发现阶段，随着新的路由接口地址被发现，它们即被添加到未完成地址清单 320 上。MIB 发现阶段一般将第一个地址从未完成清单中去掉，并探测接口地址所连接的路由器。已完成清单 324（图 13）列出了主机（接口）地址 1300 已经连接，不必再接触。在未完成地址清单上的一个路由器接口被处理后，相应的路由器接口地址就从未完成清单转移到已完成清单。初始网关清单 328（图 11）列出了 OSPF 路由器 1100（依据下面所要讨论的路由器表中指向相关路由器条目的指针识别）和相应的接口 1104（依据下面所要讨论的接口表中指向相关接口条目的指针来识别）在 MIB 发现代理 308 操作期间被发现。这些路由器都在 OSPF 发现阶段被搜索。区域边界路由器清单列出了在 OSPF 发现代理 310 操作期间被发现的 OSPF 区域边界路由器，并包括一个用于每一个路由器

条目的表示路由器是否连接的指示信号，如果已连接，则包括结果。最后，链路状态通告表由 OSPF 数据收集代理 312 输出，而且是一个链路状态通告的列表，即 LSA，从接触的 OSPF 路由器中的链路状态数据库获得。正如后面所述，链路状态数据库，按照 OSPF 协议的定义，是一个链路列表，每条链路用端点和与链路相关的成本尺度来定义。每个处于一个路由区内的区域边界路由器都有一个它所在边界的所有区域（即与边界路由器有关的区域）的完整数据库副本。然而，在一个路由区内的非边界路由器则一般只有一个它所在区域的完整数据库副本，并且不与不同路由区中的路由器拥有相同的链路状态数据库。

系统 300 提供一些输出表，分别是：路由器输出表 332、网络接口输出表 336 和接口表 340，由 MIB 发现代理 308 在结束了 MIB 发现阶段后输出；路由器清单 344、链路清单 348、网络清单 352 和接口清单 356，由 OSPF 发现代理 310 在结束了 OSPF 发现阶段后输出。现在来看图 8。对于每个被发现的路由器 800 来说，路由器表 332 包括相应的协议标识符 804 和一个或多个相应地被连接的接口 808（一般依据指向接口表 340 中相应条目的指针来识别）。现在来看图 9。对于每个接口 900 来说，接口表 340 包括一个相应的 IP 地址 904，一个或多个相应的属性 908（例如接口是否可接触、速度、接口类型、成本尺度、上行/下行状态等），拥有接口的对应的路由器 912（该接口一般由一个指向路由器表 332 中相应条目的指针识别）。现在来看图 10。网络接口表 336 为每个网络地址 1000（或其它类型的网络标识符）列出了一个或多个与网络连接相应的接口 1004（一般由一个指向接口表 340 中相应条目的指针识别）。现在来看图 14。路由器清单 344 包括企业网或自治系统中指定的和连接的路由器 1400 的综合列表。在接口清单内，路由器依据路由器标识符 1404（和/或路由器的 IP 地址之一，此处未标出）、相关的区域标识符 1408、和/或一个或多个指示相关接口的指针 1412 来识别。正如后面所述，一个区域边界路由器将拥有多个区域标识符，而非区域边界路由器则只有一个标识符，而且一个路由器可以有一个或多个相关的接口。现在来看图 15。链路清单 348 列出了被发现的链路。链路

1500 可以有两个路由器或一个路由器和一个网络（桩基网络或转接网）作为端点 1504。路由器可以依据路由器标识符和/或接口的 IP 地址来识别，网络依据掩码和/或一个或多个 IP 地址识别。现在来看图 16。网络清单 352 是一个网络 1600（桩基网络或转接网）的列表。网络 1600 是依据一个网络地址和掩码和/或一个或多个与网络连接的 IP 地址被识别的。每个网络有一组相关的路由器接口 1604（一般由一个指向接口清单内一个相应路由器接口的指针来指示），以及一个相关的指定路由器 1608（一般由一个指向路由器清单内相应路由器的指针来指示）。最后，看图 17。接口清单 356 列出了依据 IP 地址、接口编号和/或网络掩码识别的接口 1700。每个接口 1700 都与一个路由器 1704 相关。相关路由器 1704 一般由一个指向路由器清单 344 内相应路由器的指针来指示。

这些表格一起提供了网络路由拓扑结构和处于其中的网络成分的属性。正如后面所述，“路由拓扑结构”是由一个特定路由选择协议所描述的逻辑网络拓扑结构。在路由器、接口和网络接口表以及路由器、链路、网络 and 接口清单的基础上，可以自动或手动产生路由拓扑结构模型或拓扑结构图。如果使用了一个以上的路由选择协议，可能会有一个以上互不相同的路由拓扑结构。正如下面所述，路由拓扑结构可以与物理网络拓扑结构有很大的不同。

阶段控制器的操作

现在，参照图 4 对阶段控制器 304 加以说明。

在步骤 400 上创建了阶段控制器 304 后，阶段控制器 304 即在步骤 404 上进行初始网关探测。阶段控制器一般使用一个或多个种子 IP 地址以接触一个主机路由器。最初被阶段控制器 304 接触的路由器一般被称作网关路由器。在一个优选实施例中，只使用一个种子 IP 地址。如果用户没有配置阶段控制器以便将一个特定路由器作为初始网关使用，则种子 IP 地址可以自动确定。确定该地址的方法与平台有关。对于所有的平台来说，网关从具有有效网关领域的第一个路由表条目取

得。用于接触网关路由器的简单网络管理协议 (SNMP) 方法可以成为路由选择协议专用方法。例如, RFC1850 提供了利用 OSPF 协议接触一个路由器的规定。阶段控制器 304 试图在步骤 408 上接触初始网关路由器。

在判定框 412 内, 阶段控制器 304 确定, 是否用 SNMP 方法成功接触了网关路由器。如果接触尝试未获成功, 则在步骤 416 上的阶段控制器结束操作并将错误通知用户, 并请求一个新的种子地址。当接触尝试获得了成功, 则在步骤 420 上的阶段控制器访问初始网关路由器内的 MIB 信息, 并在步骤 424 上初始化未完成和已完成清单, 并将网关路由器的可接触接口的接口地址添加到未完成清单内, 以便在后面 MIB 发现阶段使用。

然后, 在步骤 428 上的阶段控制器调用 MIB 发现代理 308, 以致该代理进行 MIB 发现阶段的操作。下面将参照图 5 和图 6 对 MIB 发现代理的操作加以阐述。

当 MIB 发现阶段结束后, 在步骤 423 上的阶段控制器调用 OSPF 发现代理 310 (或直接首先调用 OSPF 数据收集代理 312, 然后调用 OSPF 数据分析代理 316) 进行 OSPF 发现阶段的操作。下面将参照图 7 对这些代理的操作进行阐述。阶段控制器可使用一种观察器模式来监控 MIB 发现代理和 OSPF 发现代理的操作进展情况 (或直接调用 OSPF 数据收集代理和 OSPF 数据分析代理)。

当 OSPF 发现阶段结束后, 阶段控制器在判定框 436 中确定未完成清单 320 是否已经清空。如果候选条目 1200 仍然在未完成清单 320 内, 阶段控制器即重复步骤 428 和 432。如果没有候选条目 1200 在未完成清单 320 内, 阶段控制器即进行步骤 440 并结束操作。

MIB 发现代理的操作

现在, 参照图 5 和图 6 阐述 MIB 发现代理 308 的操作。一般来说, MIB 发现代理 308 重复地从未完成清单 320 内取得下一个接口地址, 接触与接口地址相对应的路由器, 处理路由器的路由表并将接口地

址从未完成清单 320 转移到已完成清单 324。为了避免不必要的网络通信量和通信资源的浪费，MIB 发现代理 308 进行了优选配置，从而在路由器的一个接口被成功接触时，这个路由器的其它接口随后就不会被 MIB 发现代理接触。

MIB 发现代理 308 在步骤 500 上建立。在判定框 504 内，MIB 发现代理 308 确定未完成清单 320 是否已经清空。如果未完成清单 320 已经空了，发现代理 308 即停止操作，阶段控制器则建立 OSPF 数据收集代理 312。如果未完成清单 320 没有清空，发现代理 308 即进行步骤 508，并从未完成清单 320 上获得下一个接口地址。

在步骤 512 上，发现代理 308 接触接口地址，并在判定框 516 内确定地址是否可接触。如果地址是不可接触的，该代理即进行步骤 520，将接口地址从未完成清单 320 转移到已完成清单 324，并返回到判定框 504 以进入循环的下一轮。如果地址被成功接触，在步骤 524 的代理 308 即按下面参照图 6 进一步阐述的方法对路由器进行处理。

现在，参照图 6 描述 MIB 发现代理处理路由器的过程。

在步骤 600 上，发现代理 308 设置了一个旗标，该旗标与表示地址被成功接触的接口地址相关联。该旗标可以包含在未完成清单内。

在步骤 604 上，代理 308 检索与路由器（对应于被接触接口地址）连接的接口地址清单。该信息包含在路由器的 SNMP 表内。由该代理检索的 SNMP 表是由因特网工程任务组（IETF）的 RFC1213 标准所定义的 ipAddr 表、ipAddrTable 表、ipRouteTable 表以及 ipRouteEntry 表（“表”）。正如后面所述，ipAddr 表列出了路由器接口的 IP 地址。ipRouteTable 表可用于确定当前路由器上的每一个接口如何与网络内另一个路由器的一个接口（亦称作下一跳点接口）连接。该表也描述存在于两个路由器接口之间链路上的路由选择协议，例如 OSPF。

在步骤 608 上，该代理检索对应于路由器（正被处理中）的下一个接口地址以便进行处理。在对该地址进行处理之前，代理在判定框 612 内确定被检索地址是否处在未完成清单 320 上。若被检索地址是处

在未完成清单 320 上，则该地址在步骤 616 上被从未完成清单 320 转移到已完成清单 324。然后，或者当被检索地址未处于未完成清单 320 上时，代理 308 即进行步骤 620 的操作。

在步骤 620 上，代理 308 检索与被处理路由器的当前接口连接的下一跳点接口的地址。根据与下一跳点接口相关的信息，一个或多个条目被添加到网络接口表 336、接口表 340 和/或路由表 332 中。如果这些表预先未准备好，则在添加条目之前先进行这些表的初始化。

用于在步骤 624 上处理这些表内下一跳点接口的地址的方法将为本专业熟悉 SNMP 的技术人员所了解。如下所述，`ipAddrTable::ipAdEntIfIndex` 起到 `ipAddrTable` 表内接口地址规范键码的作用。如图 24 所示，`ipRouteEntry` 表与 `ipAddrTable` 表相结合，将 `ipRouteEntry::ipRouteIfIndex` 作为外键码使用（在 `ipRouteEntry::Instance` 上）。这种关系代表了在 `ipAddrTable` 与 `ipRouteEntry` 表之间一对多的关系。对于每一个 `ipAddrEntry` 来说，有 0 或更多的 `ipRouteEntry` 来说明一个接口连接的是哪个网络以及哪个接口被用来触及该网络（利用 `ipRouteEntry::ipRouteNextHop`）。如果没有 `ipRouteEntry`，则接口不被使用。

这些表可用于把运输网络与桩基网络区分开。若分组报文可以直接从接触的接口传递到网络（或接口直接连接到网络（这由 IP 路由类型的域标志“直接”确认）），则该网络被假定为桩基网络。若分组报文必须从接触的接口通过一个网络到达另一个网络（即接口间接地连接到列在 `ipAddrTable` 表内的端点网络（这由 IP 路由类型的域标志“间接”确认）），中间的网络即是运输网络。

对未编号的接口也必须加以考虑。对于一个未编号的接口来说，它必须满足以下条件：出现在 `ipTable` 内而不是 `ipAddrTable` 内；在 `ipTable` 内有一个“Up”状态；不是以太网接口。未编号接口将利用点对点链路的方法始终与另一个路由器连接，并与下一个路由器上的未编号接口连接。在这种情况下，当前路由器上的 `ipRouteTable` 将有一个用于未编号接口的下一跳点地址，作为下一个路由器上的一个任意（编号的）

接口地址。

图 18 ~ 21 说明了 ipAddrTable、ipRouteEntry 和 ipRouteTable 表。图 18 是一个 ipAddrTable 表，在每个 1800 实例 (instance) 中包括 ipAdEntAddr(IDX) 1804、ipAdEntIfIndex 1806 以及 ipAdEntNetMask 1808。如后面所述，一个“实例” (“instance”) 就是接口 IP 地址的一次出现。图 19 是 ipRouteEntry 表，在每个 1900 实例中包括 ipRouteDest(IDX) 1904、ipRouteIfIndex 1908、ipRouteNextHop 1912、ipRouteType 1916、ipRouteProto 1920 (即由对象接口与下一跳点地址 1920 之间的连接所支持的路由选择协议) 以及 ipRouteMask 1924。图 20 是 ipAddrTable 表，在每个实例 2000 中所包含的是与图 18 ipAddrTable 表中相同的变量。最后，图 21 是 ipRouteTable 表，每个实例 2100 所包含的是与 ipRouteTable 表相同的变量。

为了说明以上表格的使用，现在将依据图 18 ~ 21 的表格进行阐述。接口地址 192.168.34.1，即 ipAddrEntry 表的实例 1812 (图 18)，有一个索引 4 和三个在 ipRouteEntry 表 (图 19) 内的联接条目 1902、1928 和 1929。请看条目 1928，网络 192.168.34.0 可以从接口 192.168.34.1 到达。这是合乎逻辑的，因为接口地址是以 34.1 结尾的 C 级地址，并被 ipRouteType 域值“直接 (3)”所确认。因此该接口被直接连接到 192.168.34.0 网络，并且无进一步的信息假设该网络是一个桩基网络。第三个条目 1929 表明网络 192.168.35.0 可以通过接口 192.168.34.2 到达，但是如条目 1916 所示，只能按 ipRouteType 域值“间接 (4)”的规定进行间接连接。作为向该网络跳跃第一站的路由器可以按 ipRouteNextHop 域值的规定通过下一跳点接口 192.168.34.2 到达。为了到达网络 192.168.35.0，分组报文必须通过络网 192.168.34.0，络网 192.168.34.0 即被定义为运输网络。由于 34.0 网络以前被假定为桩基网络，所以现在需要将其更新为运输网络。根据对带有接口 192.168.34.2 的路由器上的 ipAddrEntry (见图 20) 和 ipRouteEntry (见图 21) 表的检查，可以看到路由器通过接口 192.168.35.1 (见图 21 最后一个条目) 连接到网络 192.168.35.0。这种连接由 ipAddrEntry 中

的接口 192.168.35.1 (图 20 的条目 2002) 和 ipRouteEntry 中的相应行 (ipRouteIndex=7) (图 21 中的条目 1908) 来确定。ipRouteType 域值 1916 规定这种连接是“直接 (3)”, 说明该接口直接连接到 35.0 网络。192.168.17.2 路由器表 (图 21) 也表明该路由器通过接口 192.168.34.2 直接连接到了网络 17.0 (见图 21 从上面数第三个条目)。

现在返回到图 18。还有其它三个接口可以搜索。它们是接口 192.168.16.100 (原始网关接口)、192.168.19.2 和 192.168.29.2。用于这些接口的图 19 中的联接 ipRouteEntry 表条目表明, 该路由器也有如下连接: (1) 通过接口 192.168.16.100 直接连接网络 192.168.16.0; (2) 通过接口 192.168.19.2 直接连接网络 192.168.19.0; (3) 通过接口 192.168.19.2 间接连接网络 192.168.18.0 (下一跳点接口是新发现的路由器上的 192.168.19.1); (4) 通过接口 192.168.29.2 直接连接网络 192.168.29.0。

根据前面对图 18~21 的分析所得出的网络拓扑结构如图 22 所示。

再回到图 6。代理 308 在步骤 628 上确定当前接口与下一跳点地址的连接是否使用 OSPF。这要根据图 19 和/或 21 中的条目 ipRouteProto 1920 来确定。若这种连接支持 OSPF, 则一个 OSPF 协议标识符在步骤 629 上被添加到路由器表 332 内, 而且接口地址在步骤 630 上被添加到初始网关清单 328 内以便在以后的 OSPF 发现阶段使用。无论是哪种情况, 代理 308 都要进行下一步判定框 632 的操作。

在判定框 632, 代理 308 确定下一跳点地址是在未完成清单 320 内还是在已完成清单 324 内。两种清单内都没有下一跳点地址, 就在步骤 636 上将下一跳点地址添加到未完成清单 320 内。无论是哪种情况, 代理 308 都要进行判定框 640 的操作。

在判定框 640, 代理 308 确定是否有另一个接口地址与被接触的路由器连接, 而该路由尚未被考虑。如果另一个接口地址还未被考虑, 则代理 308 返回步骤 608 并为该地址重复上述步骤。如果没有地址需要考虑, 则代理 308 进行图 5 所示的步骤 520。

OSPF 数据收集和数据分析代理的操作

现在将参照图 7 阐述 OSPF 数据收集和数据分析代理 312 和 316 的操作。一般说来，初始网关清单 328 种的各个接口地址都被用 OSPF 数据收集代理 308 在 OSPF 发现阶段用作初始网关起始点。曾在上一轮 OSPF 探测中被发现的 OSPF 路由器不需要在数据收集代理 312 的下一轮运行中被用作初始网关起始点。

现在来看图 7。步骤 700 上的阶段控制器 304 检索初始网关清单 328 并于步骤 704 在清单 328 内查到下一个路由器（即与路由器连接的第一个接口地址）以便按 OSPF 发现法处理。阶段控制器便在步骤 708 将 OSPF 发现初始网关设置在第一个接口地址上，在步骤 712 建立 OSPF 数据收集代理 312 并在步骤 716 建立 OSPF 数据分析代理 316。数据收集代理 312 和数据分析代理 316 的操作在与本申请同时申请的、序列号为 No. 10/127,967、由 Goringe 等人申请、名为“利用链路状态信息发现 IP 网络拓扑结构”（“USING LINK STATUS INFORMATION TO DISCOVERER IP NETWORK TOPOLOGY”）的美国专利申请中进行了描述，也在由 Goringe 等人于 2001 年 9 月 6 日递交的序列号为 No. 60/317,719 的同名美国临时申请中进行了描述。所有这些都包含在本处供参考。

正如以上申请文件所描述，数据收集代理 312 使用一个种子接口 IP 地址，以便在所选择的区域接触一个路由器，在被接触主机路由器的链路数据库内下载 ospfLsdb（OSPF 链路状态数据库）表，废除关心区域之外的任何链路状态通告，将各个与区域边界路由器相关的接口的 IP 地址添加到区域边界路由器表内，并将任何关心区域的 LSA（链路状态通告）添加到链路状态通告表内。这些步骤为各个区域边界路由器重复进行。

数据分析代理 316 穿过链路状态通告表，废除 1 类和 2 类之外的所有链路状态通告，并利用 1 类和 2 类链路状态通告形成路由器清单 344（图 14，该表包括所有在关心区域被发现的 OSPF 路由器）、链路清

单 348 (图 15)、网络清单 352 (图 16) 和/或接口清单 356 (图 17)。

数据收集和数据分析代理 312 和 316 的操作完成后, 阶段控制器 304 在步骤 720 上将所有被发现的 OSPF 路由器从初始网关清单 328 上清除。如下所述, 数据收集代理 312 可以为每个被发现 OSPF 路由器设置一个旗标。

在步骤 724 上, 阶段控制器对所有被数据收集和分析代理 312 和 316 发现的路由器接口地址进行检索, 以判定是否还需要进行 MIB 发现阶段操作。做出这种判定是为了保证 MIB 发现代理 308 所考虑的是在 OSPF 发现阶段, 而不是 MIB 发现阶段, 所发现的任何路由器接口地址。为达到这一目的, 阶段控制器 304 在步骤 728 上得到下一个被检索到的接口地址, 并在步骤 732 上确定该检索到的接口地址是在未完成清单 320 上还是在已完成清单 324 上。如果该地址不在未完成清单 320 或已完成清单 324 上, 阶段控制器即在步骤 736 上将该地址添加到未完成清单内。在判定框 740 中, 阶段控制器 304 即确定是否还有需要考虑的其它接口地址。如果有, 控制器即返回步骤 728 并重复前面所述步骤。如果没有, 则控制器进行判定框 744 的操作并确定在初始网关清单 328 中还有其它 OSPF 路由器。如果在清单 328 中还有其它 OSPF 路由器, 则控制器返回步骤 704 以进一步进行 OSPF 发现阶段的操作。当在清单 328 中没有其它的路由器时, 控制器即进行图 4 中的步骤 436 的操作。

各种输出表和清单可以多种方式合并, 以形成一个统一的网络拓扑结构模型。在一种“任你行”的方法中, 网络模型是一个知识储藏库。发现算法用一种命题法来处理模型, 即算法将它所发现的现象通知模型, 并让模型自行决定如何应对。若信息接收自一个协议, 该信息表明以前所得到的有关网络和路由器等的信息是不完全的, 更多的信息即被添加到模型中。所有信息都有标记说明是哪个协议提供的, 以帮助识别冲突信息。若来自两个不同协议的信息发生冲突, 则优选来自 MIB2 路由表的信息。信息被标记为受到有关路由选择协议的冲突而且路由选择协议信息被保留。

可以使用一些对本发明的变动和修改方案, 在提供本发明的一些

特点时不提供其它特点的情况也是合理的。

例如在一个供选择实施例中，算法被用于不同于 OSPF 的协议，诸如属于非 SNMP 网络管理协议的公共管理信息协议 (CMIP) 以及 (或) 不依靠对 MIB 信息访问的发现方法。算法可用于发现支持任何距离向量和链路状态路由选择协议的任何路由器。

在另一个供选择实施例中，算法被同时用于多个路由选择协议。图 23 对该实施例做出了说明。在步骤 2400 上 (当网关路由器被接触时)，系统 300 对在目标网络上使用的路由选择协议进行探测。在某些配置中，网关路由器可以使用多于一个的路由选择协议。在步骤 2404、2408 和 2412 上，进行拓扑结构探测时，对各个被发现的路由选择协议使用了不同的算法。在图 23 中，各个 OSPF、RIP、和 EIGRP 路由选择协议都使用了不同的算法。用于 RIP 和 EIGRP 的算法是从前面所阐述的算法修改而来，这些修改对本领域的技术人员是显而易见的。如果使用了一个以上的路由选择协议，就可能有一个以上的路由拓扑结构。本发明的网络模型可以不将不同的路由拓扑结构合并为一个通用模型。在这种情况下，每一种路由拓扑结构都同时体现在单一的数据结构中。

在另一个供选择实施例中，根据路由器所支持的路由选择协议，在逐个路由的基础上调用各种发现代理。例如，在拓扑结构探测期间，一个 OSPF 可以发现首个 OSPF 路由器，然后 RIP 代理的被调用去探测下一个支持 RIP 而不是 OSPF 的路由器。

在另一个供选择的实施例中，MIB 和 OSPF 发现阶段可以按相反的顺序进行，即 OSPF 发现可以在 MIB 发现之前完成。

在另一个供选择的实施例中，MIB 和 OSPF 发现阶段并行地，即同时地进行。

在另一个实施例中，前面所阐述的任何软件模块都可以作为一个应用程序专用集成电路或任何其它类型的逻辑电路，部分地或整个地应用。

本发明在各种实施例中，正如此处所作的实质性说明和描述，包含了关于网络成份、方法、过程、系统和/或设备的描述，从而也包含了

它们的各种实施例、子组合和子集。本专业的技术人员在理解了现在所公开的内容后将懂得如何实现和利用本发明。本发明，在各个实施例中，包括了在没有使用在此处或各种实施例中未曾介绍和/或已经介绍项目的情况下提供设备和方法的内容，这些项目包括在以前的装置或方法中使用过的许多项目，目的是，例如改善性能、提高实现的简易性和/或降低实现成本。

以上对本发明阐述的目的是对其进行描述和说明。前面的介绍无意将本发明局限于此处所公开的形式。虽然本发明的说明包括了对一个或多个实施例以及某些变更和修改的说明，但其它变更和修改也都属于本发明的范围，例如，在本专业技术人员理解了本发明公开的内容之后，有许多变更和修改都是在他们的技能和知识范围之内的。我们所想要的是获得在允许的程度向已经提出权利要求的项目添加供选择实施例的权利，其中包括供选择的、可互换的和/或等效的结构、功能、范围或步骤，无论这些供选择的、可互换的和/或等效的结构、功能、范围或步骤是否已在此处公布，而且也无须公开任何可申请专利的主题。

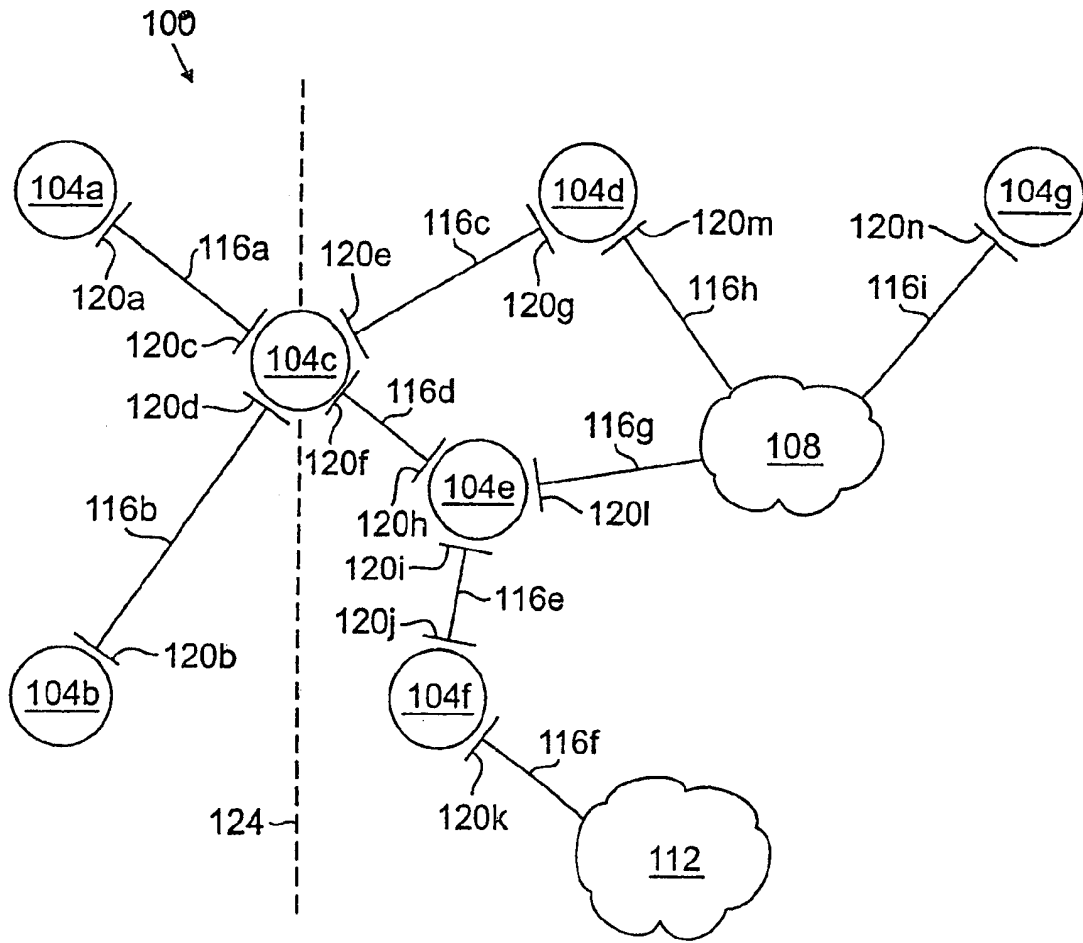


图1
(现有技术)

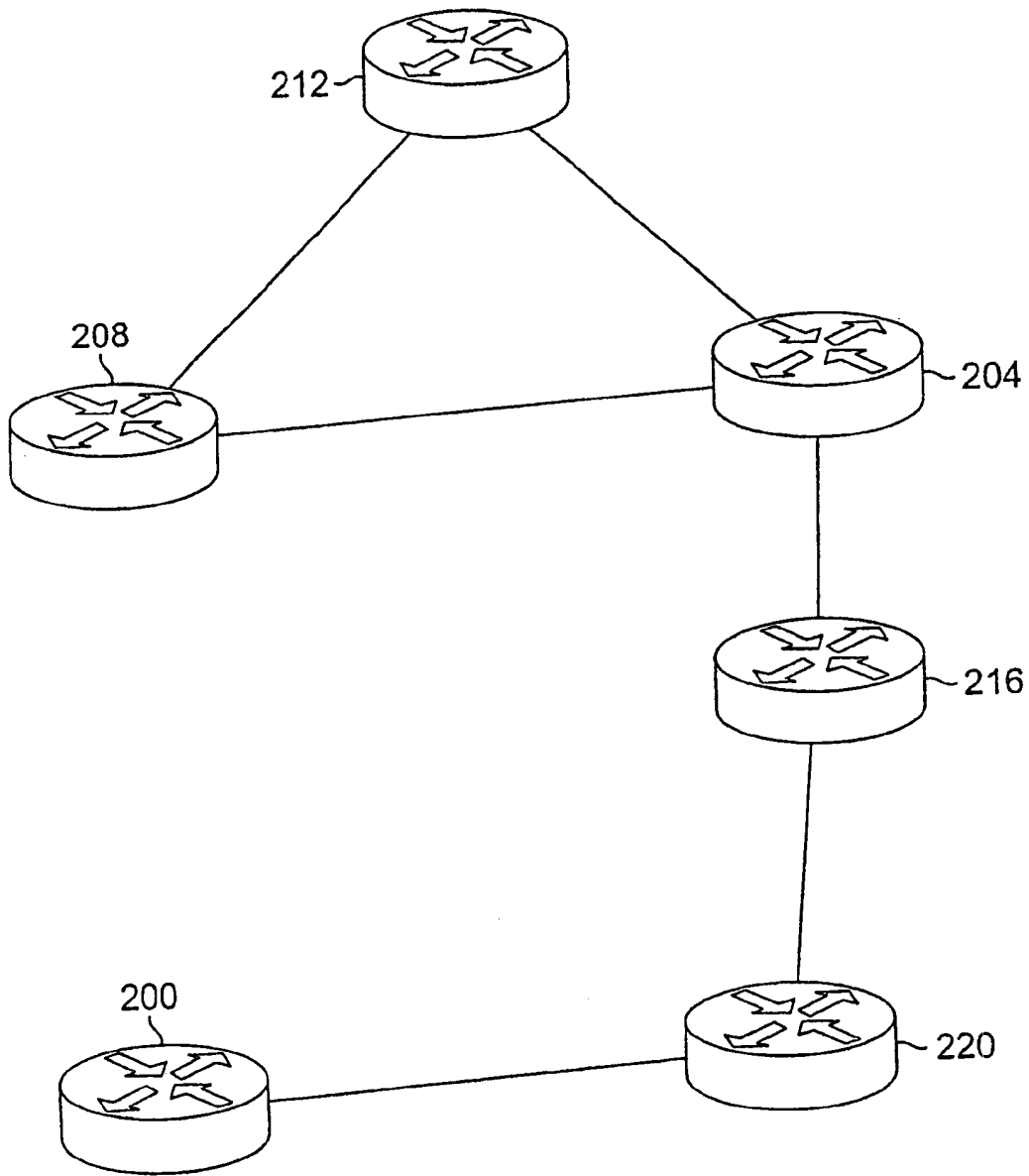
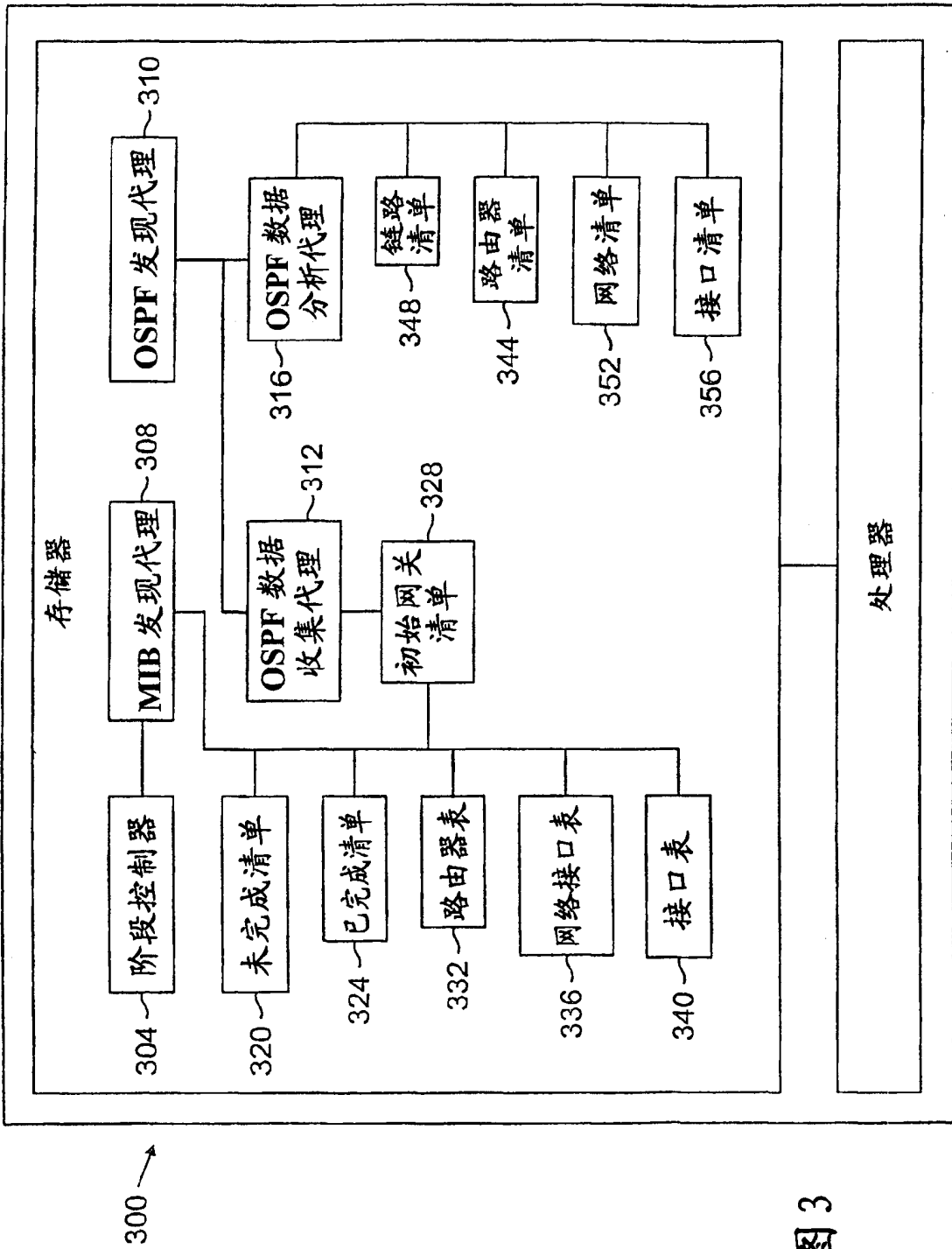


图2
(现有技术)



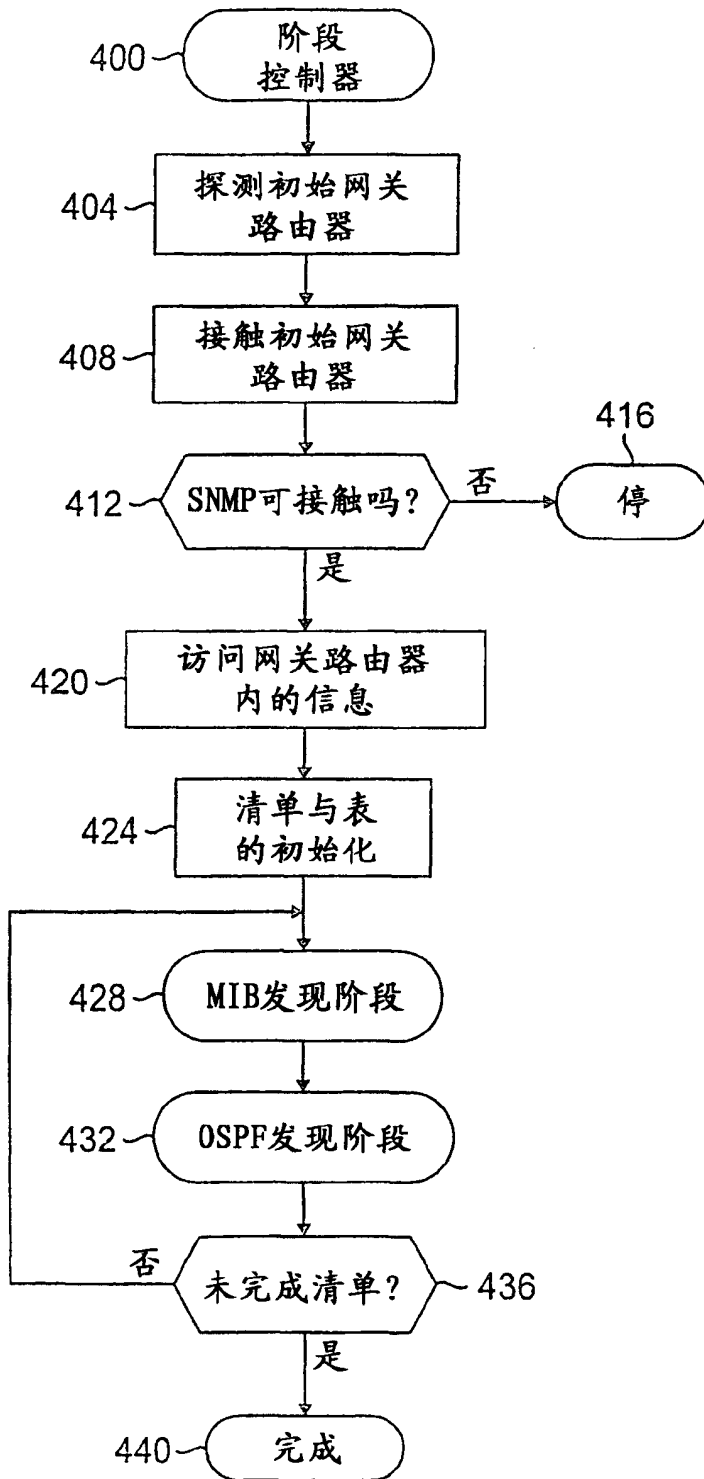


图4

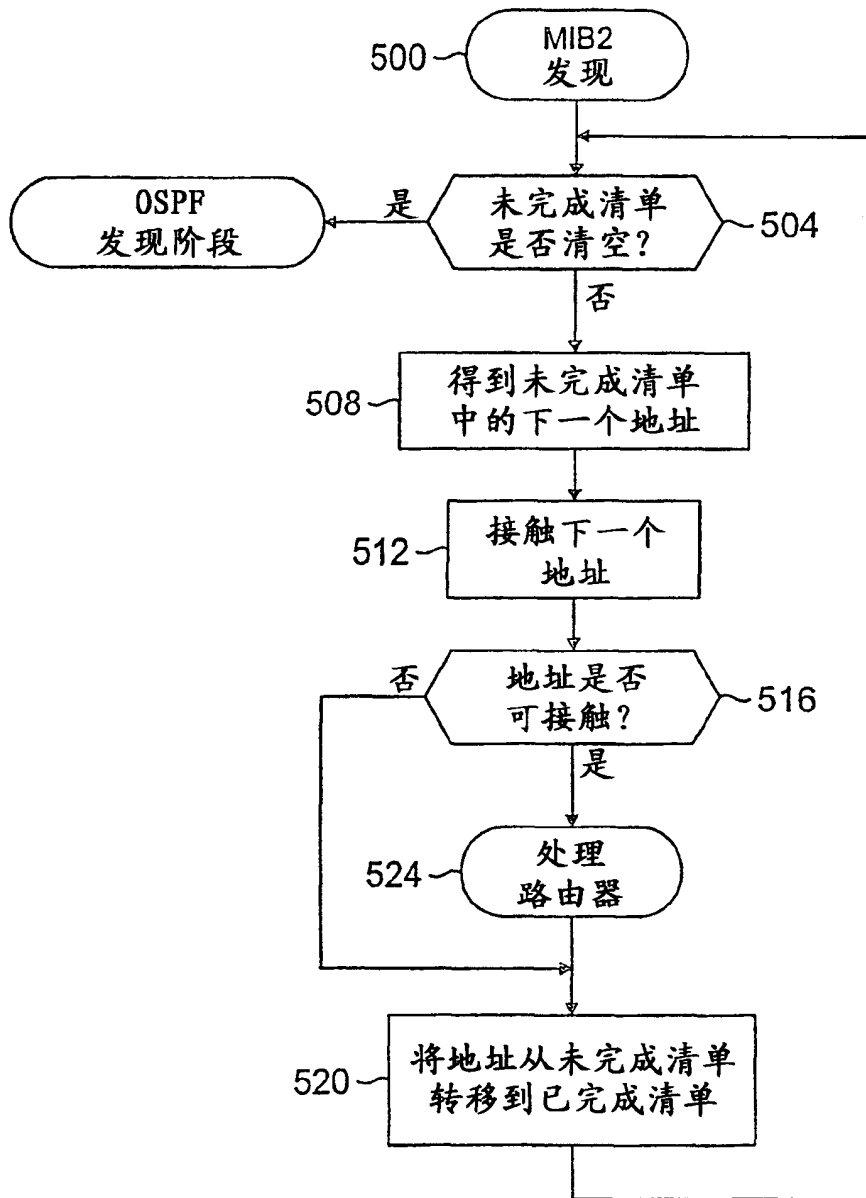


图5

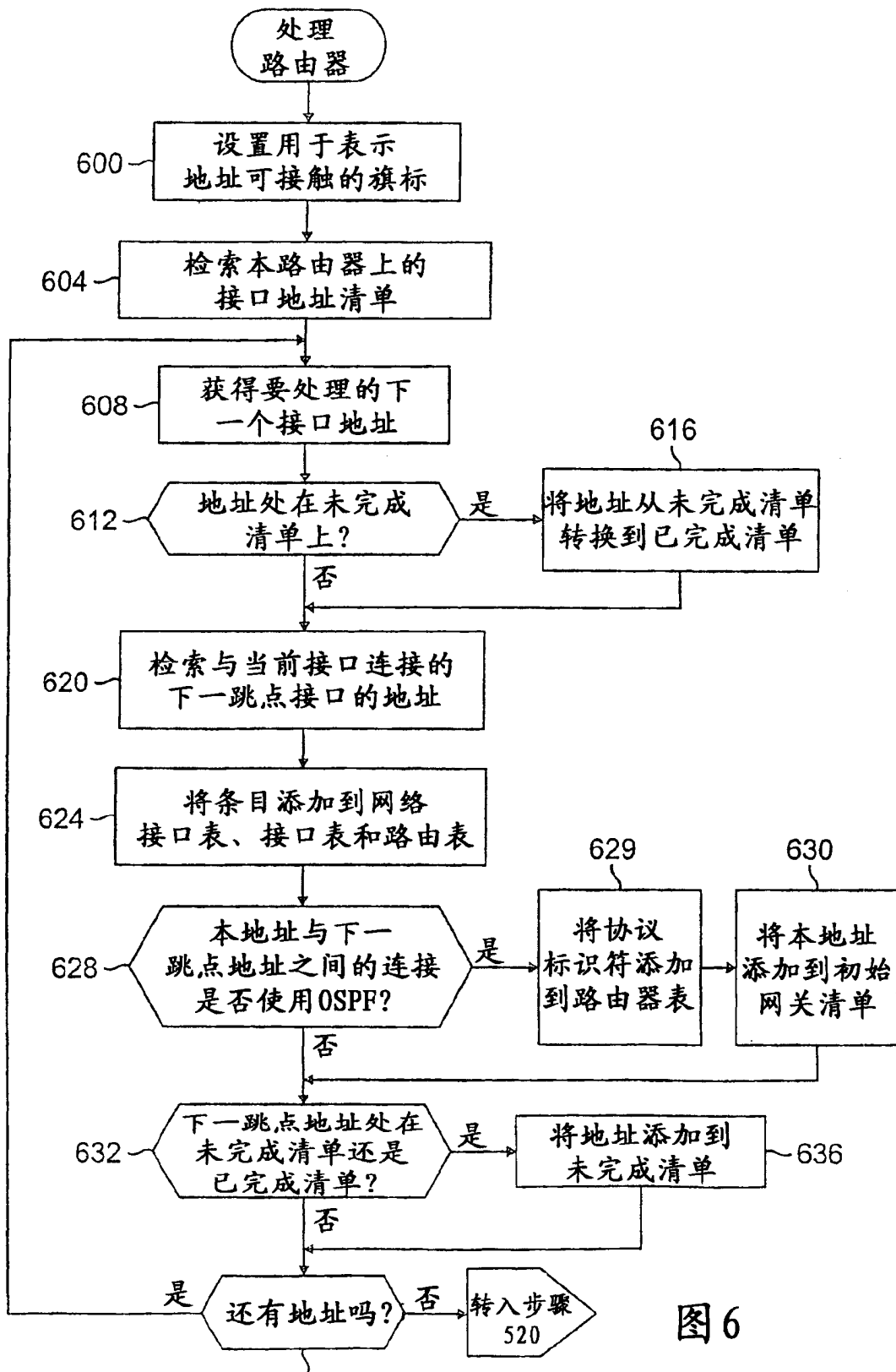


图 6

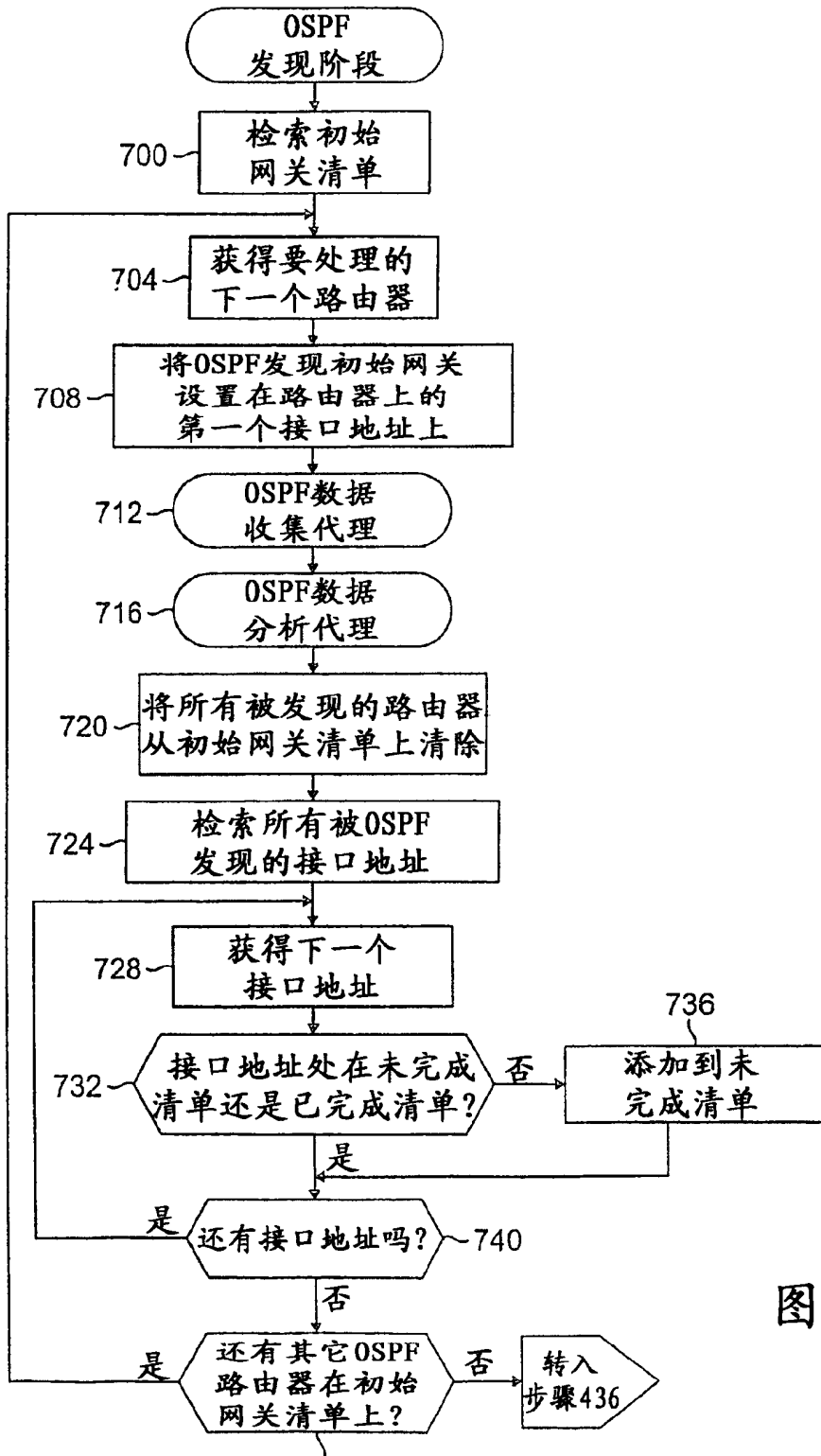


图7

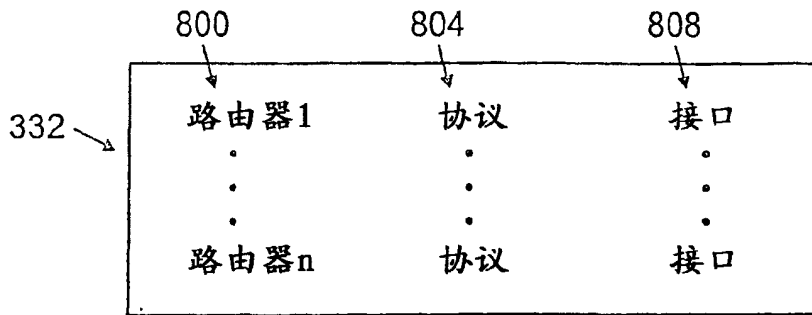


图8

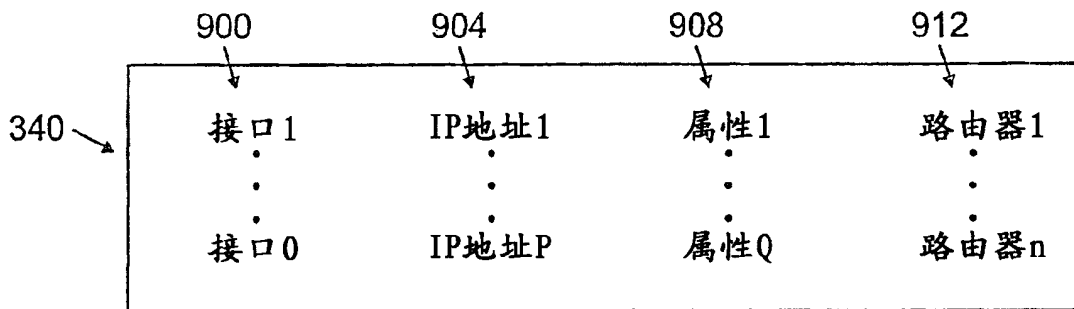


图9

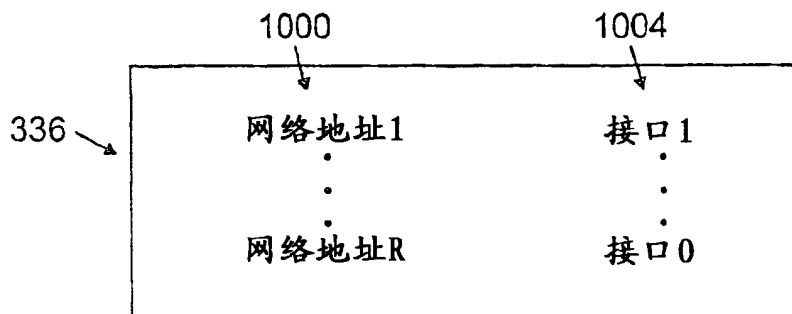


图10

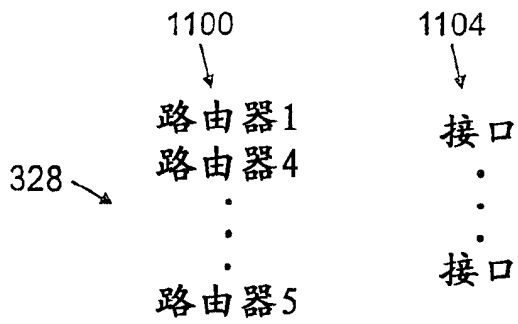


图 11

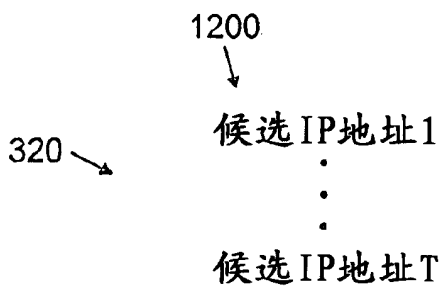


图 12

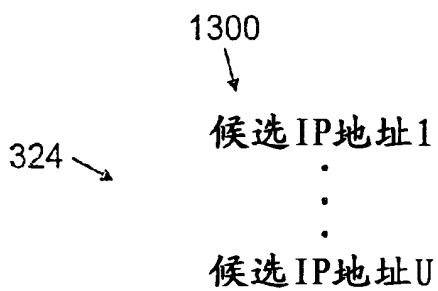


图 13

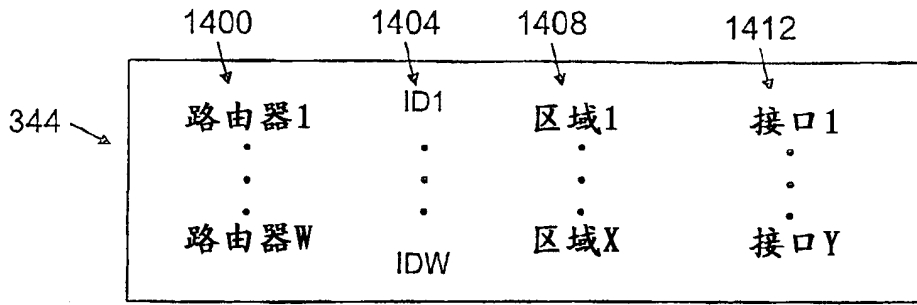


图14

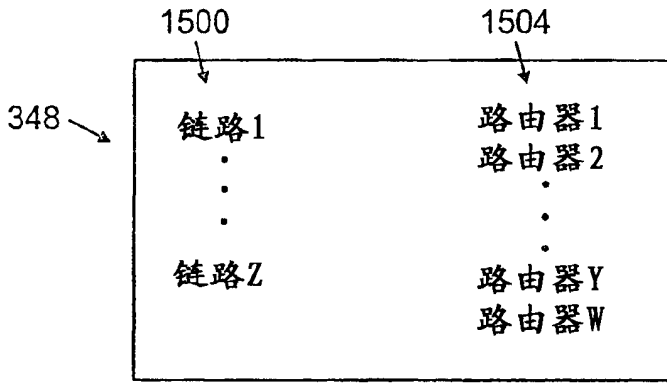


图15

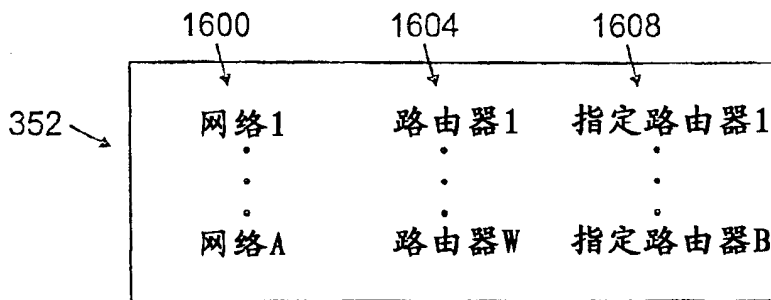


图16

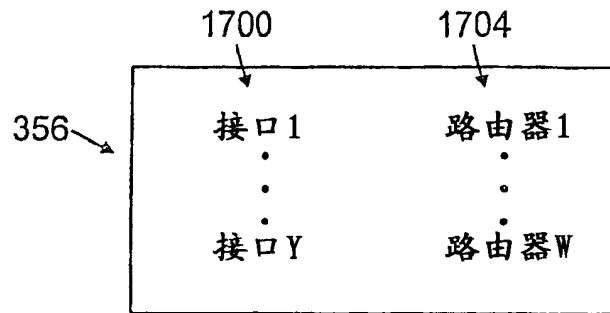


图 17

	1812	1804	1806	1808
⊕	192.168.16.100	192.168.16.100	3	255.255.255.0
⊕	192.168.19.2	192.168.19.2	2	255.255.255.0
⊕	192.168.29.2	192.168.29.2	9	255.255.255.0
⊕	192.168.34.1	192.168.34.1	4	255.255.255.0

图 18

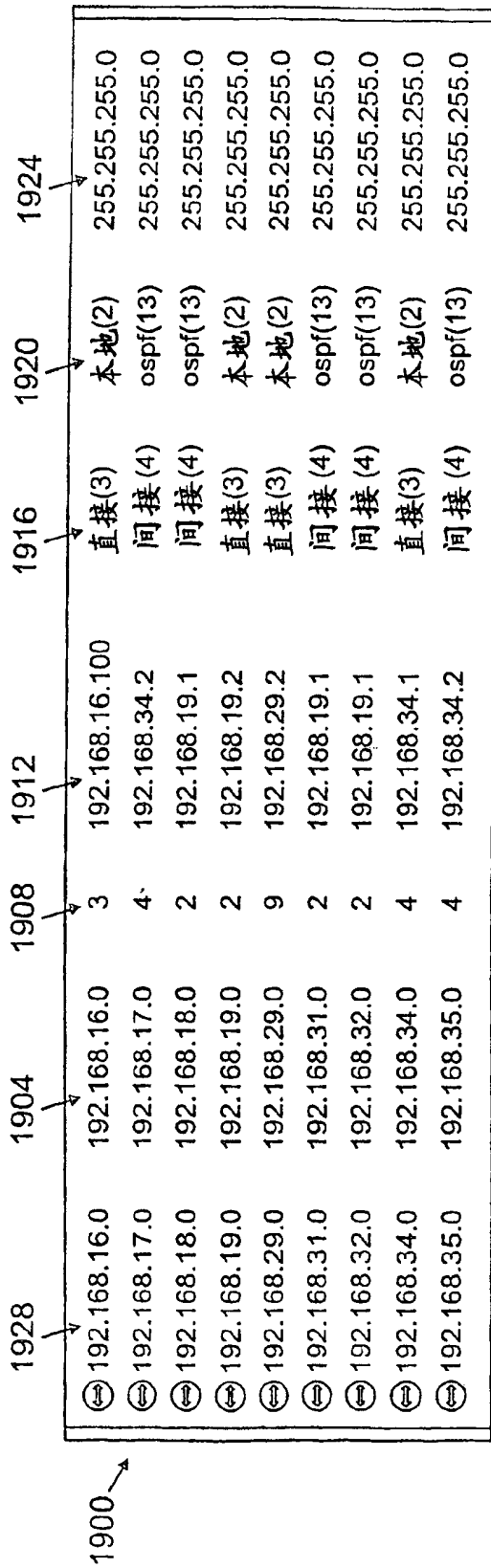


图19

	2000	1804	1806	1808
⊖	192.168.17.2	192.168.17.2	1	255.255.255.0
⊖	192.168.34.2	192.168.34.2	3	255.255.255.0
⊖	192.168.35.1	192.168.35.1	7	255.255.255.0

图 20

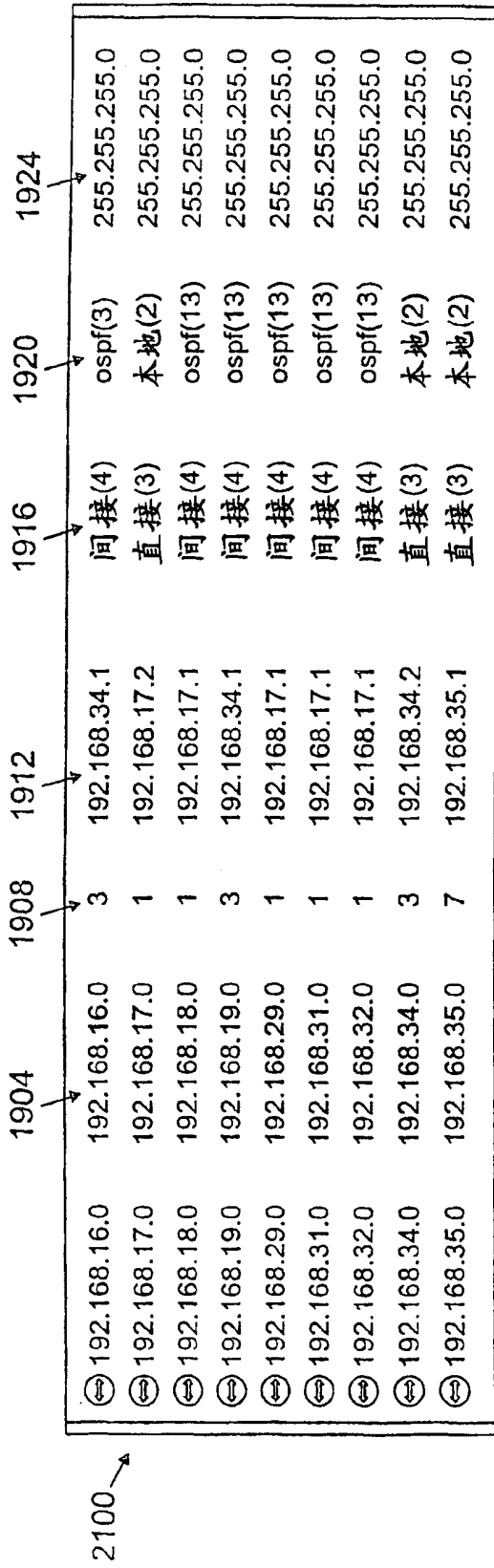


图21

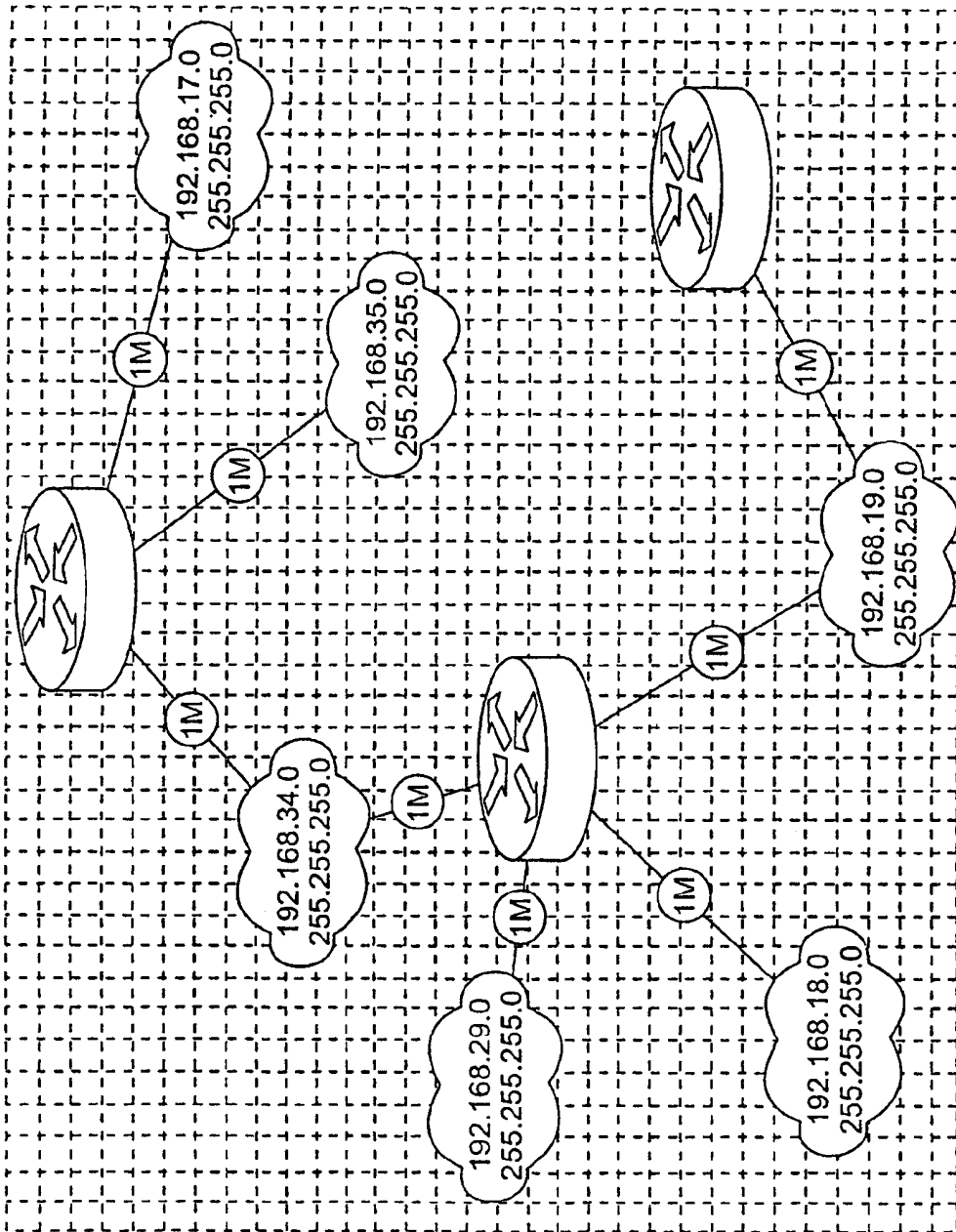


图22

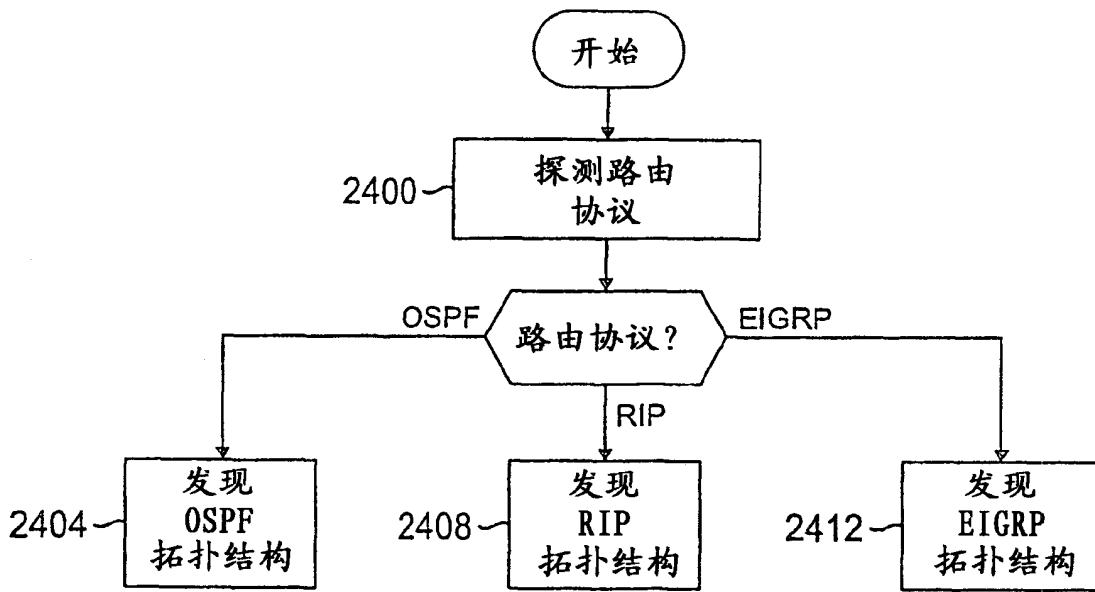


图 23

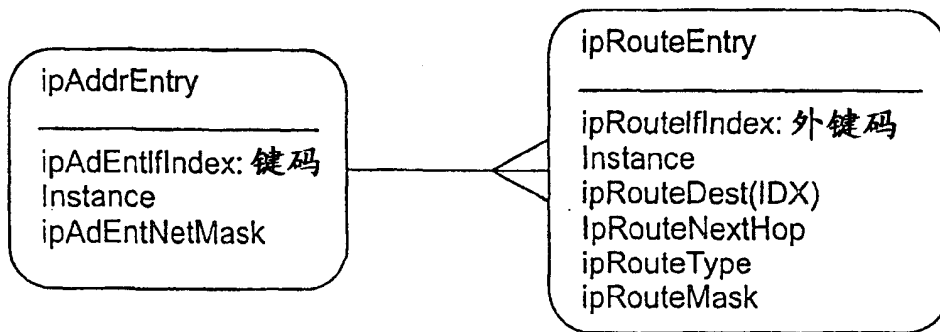


图 24