

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G01R 31/08 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200680001652.0

[43] 公开日 2008 年 1 月 2 日

[11] 公开号 CN 101099086A

[22] 申请日 2006.2.21

[21] 申请号 200680001652.0

[30] 优先权

[32] 2005.2.22 [33] US [31] 11/064,275

[86] 国际申请 PCT/US2006/006238 2006.2.21

[87] 国际公布 WO2006/091652 英 2006.8.31

[85] 进入国家阶段日期 2007.6.28

[71] 申请人 思科技术公司

地址 美国加利福尼亚州

[72] 发明人 斯图尔特·弗雷德里克·布赖恩特
伊恩·迈克尔·查尔斯·尚德

[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

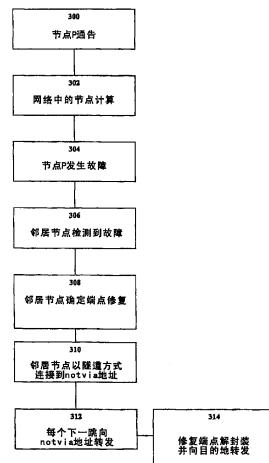
权利要求书 4 页 说明书 22 页 附图 11 页

[54] 发明名称

用于在数据通信网络中构造绕过不可用组件的修复路径的方法和装置

[57] 摘要

公开了一种在数据通信网络中构造绕过不可用组件的修复路径的方法，该数据通信网络具有作为组件的具有网络地址的节点和其间的链路，该方法包括以下步骤：在参与节点处执行从网络中的通知节点接收标识通知节点和通过其可以到达通知节点的组件的通知。该方法还包括以下步骤：从通知中导出网络修复地址，该网络修复地址用在通知中标识的组件不可用的情况下。该方法还包括以下步骤：构造针对修复地址的修复路径。



1. 一种数据分组路由装置，包括：

一个或多个处理器；

可通信地耦合到所述一个或多个处理器的网络接口，其被配置为在网络中在所述一个或多个处理器之间传输一个或多个分组流；

包括一个或多个指令序列的计算机可读介质，所述指令序列当被所述一个或多个处理器执行时，使得所述一个或多个处理器执行以下步骤：

从所述网络中的通知节点接收标识所述通知节点和通过其可以到达所述通知节点的组件的通知；

从所述通知中导出网络修复地址，所述网络修复地址用在在所述通知中标识的组件不可用的情况下；以及

构造针对所述修复地址的修复路径。

2. 如权利要求 1 所述的装置，其中所述通过其可以到达所述通知节点的组件包括相邻节点和相邻链路中的至少一个。

3. 如权利要求 1 所述的装置，其中所述用于构造修复路径的指令包括用于构造忽略了所述不可用组件的修复网络拓扑并标识到所述修复拓扑中的组件的修复路径的指令。

4. 如权利要求 3 所述的装置，其中所述指令还包括用于通过忽略所述不可用组件来构造只针对到其路径受影响的组件的修复拓扑。

5. 如权利要求 3 所述的装置，其中所述指令还包括用于构造排除了不具有修复地址能力的网络组件的修复拓扑的指令。

6. 如权利要求 1 所述的装置，还包括用于经由相应的修复路径发送去往修复地址的网络流量的指令。

7. 如权利要求 1 所述的装置，其中所述参与节点包括到包括修复节点的不可用组件的邻居节点，并且其中所述修复节点包括用于针对目的地地址识别修复地址，以在所述组件不可用的情况下作为用于修复所述目的地地址的封装修复端点的指令。

8. 如权利要求 7 所述的装置，其中所述修复节点还包括用于在要求修

复路径的组件不可用的情况下，将针对所述修复路径的流量封装并发送向所述修复地址的指令。

9. 如权利要求 7 所述的装置，其中所述修复节点还包括用于不标识针对包括修复地址的目的地地址的封裝修复端点的指令。

10. 如权利要求 1 所述的装置，其中所述数据通信网络包括作为组件的多归属前缀，并且其中参与节点还包括用于以下步骤的指令：

确定到多归属前缀的附接点是否可以在不利用修复路径的情况下到达，并且如果可以则封装分组并发送向所述附接点；以及

如果到所述多归属前缀的附接点只能经由修复路径到达，则封装流量并发送向相应的修复地址。

11. 如权利要求 1 所述的装置，其中所述数据通信网络包括包含多个组件的共享风险群组，并且其中参与节点还包括用于以下步骤的指令：

导出用在所述共享风险群组不可用的情况下的网络修复地址；以及构造针对所述修复地址的修复路径。

12. 如权利要求 11 所述的装置，其中所述共享风险群组是 LAN。

13. 如权利要求 6 所述的装置，其中所述指令还包括用于接收被封装并发送向所述修复地址的流量，并在前一组件处对所述分组解封装并发送向所述修复地址的指令。

14. 如权利要求 1 所述的装置，其中修复路径是针对多播流量构造的。

15. 如权利要求 1 所述的装置，利用链路状态路由、MPLS 路由或距离向量路由之一。

16. 如权利要求 1 所述的装置，还包括包含用于执行以下步骤的指令的参与节点：

识别所述参与节点是否处于针对修复地址的修复路径中；以及仅在该情形中构造针对所述修复地址的修复路径。

17. 如权利要求 1 所述的装置，其中通知节点还包括用于执行以下步骤的指令序列：

在通知中标识所述通知节点和通过其可以到达所述通知节点的组件；

在所述通知中标识网络修复地址，所述网络修复地址用在在所述通知中标识的组件不可用的情况下；以及
通告所述通知。

18. 一种包括一个或多个指令序列的计算机可读介质，所述指令序列当被一个或多个处理器执行时，使得所述一个或多个处理器执行以下步骤：

从网络中的通知节点接收标识所述通知节点和通过其可以到达所述通知节点的组件的通知；

从所述通知中导出网络修复地址，所述网络修复地址用在在所述通知中标识的组件不可用的情况下；以及

构造针对所述修复地址的修复路径。

19. 一种方法，包括机器实现的以下步骤：

从网络中的通知节点接收标识所述通知节点和通过其可以到达所述通知节点的组件的通知；

从所述通知中导出网络修复地址，所述网络修复地址用在在所述通知中标识的组件不可用的情况下；以及

构造针对所述修复地址的修复路径。

20. 一种用于在数据通信网络中构造绕过不可用组件的修复路径的装置，所述数据通信网络具有作为组件的具有网络地址的节点和其间的链路，所述装置包括：

用于从所述网络中的通知节点接收标识所述通知节点和通过其可以到达所述通知节点的组件的通知的装置；

用于从所述通知中导出网络修复地址的装置，所述网络修复地址用在在所述通知中标识的组件不可用的情况下；以及

用于构造针对所述修复地址的修复路径的装置。

21. 一种用于向数据通信网络中的组件通知通知节点可达性的装置，包括：

用于在通知中标识所述通知节点和通过其可以到达所述通知节点的组件的装置；以及

用于在所述通知中标识网络修复地址的装置，所述网络修复地址用在在所述通知中标识的组件不可用的情况下。

用于在数据通信网络中构造绕过不可用组件的修复路径的方法和装置

技术领域

本发明一般地涉及数据通信网络。更具体而言，本发明涉及在数据通信网络中构造绕过不可用组件的修复路径。

背景技术

这一部分中描述的方法可以实现，但是不一定是先前已想到或已实现的方法。因此，除非另外指出，否则这一部分中描述的方法不是该申请的权利要求的现有技术，并且也不应当因为被包括在这一部分中而看作是现有技术。

在诸如因特网之类的计算机网络中，数据分组根据各种路由协议之一经由元件的网络从源被发送向目的地，这些元件包括链路（诸如电话线或光线路之类的通信路径）和节点（例如，引导分组沿着连接到其的多条链路中的一条或多条行进的路由器）。

一类路由协议是链路状态协议。链路状态协议依赖于每个节点处的路由算法。网络上的每个节点通过网络向相邻节点通告链路，并提供与每条链路相关联的代价，这一代价可以基于任何适当的度量，例如链路带宽或延迟，并且一般被表达为整数值。链路可以具有不对称代价，即沿一条链路的方向 AB 的代价可能不同于沿方向 BA 的代价。基于链路状态分组（LSP）形式的通告信息，每个节点构造链路状态数据库（LSDB）（这是整个网络拓扑的映射），并根据其基于适当的算法（例如，最短路径优先（SPF）算法）一般地构造到每个可用节点的单条最优路由。结果构造了“生成树”（SPT），其根在节点处并且示出了包括到每个可用目的地节点的中间节点在内的最优路径。SPF 的结果被存储在路由信息库（RIB）中，并且基于这些结果，转发信息库（FIB）或转发表被更新以适当地控制分组的转发。当存在网络改变时，代表改变的 LSP 通过网络被与改变相

邻的每个节点洪泛，每个节点接收 LSP 并将其发送向每个相邻节点。

因此，当去往目的地节点的数据分组到达一个节点时，该节点识别到该目的地的最优路由，并沿着该路由将分组转发到下一跳（“NEXT_HOP”）。下一节点重复这一步骤，依此类推。

将会注意到，在正常的转发中，无论从其接收到分组的节点是那个，每个节点都判定分组应当被转发到的下一节点。在某些情形中，这可能形成“环路”（loop）。具体而言，这可能发生在数据库（以及相应的转发消息）在路由转变期间临时失去同步时，即，由于网络的改变，新的 LSP 被传播，这在 RIB 或 FIB 中产生了环路。例如，如果节点 A 经由节点 B 发送分组到节点 Z（包括根据其 SPF 的最优路由），则可能出现这样的情形，其中节点 B 根据其 SPF 确定到节点 Z 的最佳路由是经由节点 A 并发回分组。这可能持续进行，只要环路仍然存在，尽管通常分组将具有最大跳数，在这之后分组将被丢弃。这种环路可以是两个节点之间的间接环路或围绕节点电路的不直接环路。

对于环路问题已经提出的一种解决方案在 Kevin Miles 等人于 2003 年 1 月 9 日提交的题为 “Method and Apparatus for Constructing a Backup Route in a Data Communications Network” 的未决专利申请 No. 10/340,371（“Miles 等人”）中有所描述，该申请的全部内容通过引用结合于此以用于所有目的，就好像被在这里完全公开一样，并且将在下面更详细地讨论。根据 Miles 等人给出的解决方案，在修复节点检测到相邻组件的故障的情况下，修复节点计算包括除了通过穿越故障组件可达的节点以外根据其协议可达的所有节点的集合的第一节点集合。然后，修复节点计算包括从其可在不穿越故障组件的情况下到达目标节点的所有节点的集合在内的第二节点集合。该方法随后确定在第一节点集合和第二节点集合的交集或其一跳的扩展中是否存在任何中间节点，并将去往目标节点的分组以隧道方式发送到隧道端点（end point），该隧道端点包括第一和第二节点集合的交集中的节点。

Miles 等人描述的解决方案可以参考图 1 进一步理解，图 1 示出了该解决方案被应用于的网络图。网络包括节点 P（标号 10），节点 A、B、C

和 S (标号 12、14、16、18) 经由相应链路 20、22、24、26 附接到节点 P。在节点 P 发生故障的情况下，充当修复节点的节点 S 计算了除了通过故障组件之外的可到达的第一节点集合，这里称为节点 S 的 P 空间， P_S (标号 28)。节点 S 还计算从其可以在不穿越节点 P 的情况下到达目标节点 B 的节点，这里称为节点 B 的 Q 空间， B_Q (标号 30)。作为修复节点的节点 S 随后识别出 P_S 中的节点 E 和 Q_B 中的节点 F (标号 32、34) 经由链路 36 在彼此的一跳内。节点 S 随后以隧道方式经由路径 38 发送分组到节点 E，也就是说，节点 S 封装分组，并附接节点 E 的地址作为目的地头部。为了使分组穿越链路 36，节点 S 还向经封装分组添加“直接转发”指令以使得节点 E 在对分组解封装后转发到节点 F，无论其到节点 B 的下一跳是多少。一旦分组到达节点 F，则根据定义，其将会经由节点 B 的 Q 空间 Q_B 中的路径 40 到达节点 B。

该方法的扩展在 George Swallow 等人于 2004 年 10 月 27 日提交的题为“Method and Apparatus for Forwarding Data in a Data Communications Network”的未决专利申请 No. 10/976,076 (“Swallow 等人”) 中有所描述，该申请的全部内容通过引用结合于此以用于所有目的，就好像被在这里完全公开一样。根据该扩展，修复节点还计算到故障节点的其他邻居的 Q 空间（从而形成扩展 Q 空间），并将分组转发到 P 空间 P_S 和扩展 Q 空间的交集中的节点。再一次参考图 1，例如，作为修复节点的节点 S 识别出节点 E' (标号 42) 可经由路径 44 到达，并且位于 C 的 Q 空间 Q_C (标号 45) 中。分组随后经由路径 44 跨过 Q 空间 Q_C 被转发到节点 F' (标号 46)，节点 F' 也落在 B 的 Q 空间 Q_B 中，并且从节点 F' 分组经由路径 48 被转发到节点 B。

Miles 等人和 Swallow 等人的方法可以针对节点和链路故障实现。在链路故障的情况下，提出了各种解决方案来避免修复节点和经由故障链路连接到修复节点的节点的邻居之间的环路，并且这些方案可以实现冲突修复策略，也就是说，在链路 26 的故障的情况下实现在图 1 的节点 S 与节点 A、B 和 C 之间。

一种这样的解决方案在 Michael Shand 等人于 2003 年 1 月 15 日提交的

题为“Method and Apparatus for Determining a Data Communication Network Repair Strategy”的未决专利申请 No. 10/346,051（“Shand 等人”）中有所描述，该申请的全部内容通过引用结合于此以用于所有目的，就好像被在这里完全公开一样。根据该方法，修复节点沿着修复路径发送探测分组，并且如果返回探测，则环路被识别。又一解决方案在 Michael Shand 等人于 2004 年 5 月 18 日提交的题为“Method and Apparatus for Forwarding Data in a Data Communications Network”的未决专利申请 No. 10/848,669（“Shand 等人 II”）中有所描述，该申请的全部内容通过引用结合于此以用于所有目的，就好像被在这里完全公开一样。根据 Shand 等人 II 描述的方法，修复分组被标记为被修复的以避免环路。

这些系统提供了在组件故障的情况下快速网络恢复，P 和 Q 空间的使用意味着在许多情形中，所选的修复路径不是最短可用修复路径。此外，在某些情形中需要诸如向转发之类的额外的转发机制。在某些情形中需要进一步的额外的封装层，例如在使用扩展 Q 空间或者诸如多归属前缀（MHP）的情况下。多次封装是不合需要的，例如由于对额外的校验和计算的需求。

这些问题中的许多问题之所以存在是因为当链路或路由器发生故障时，只有故障的邻居首先知道故障已发生。根据上述解决方案，该故障由作为故障邻居的路由器修复。这些修复路由器不得不操纵分组去往其目的地，而不管网络中的大多数其他路由器不知道故障的本质和位置这一事实。

又一已知方法可以在万维网上的域“watersprings.org”的目录“pub/id”中的文件“draft-tian-frr-alt-shortest-path-01.txt”处得到的“Fast Re-route using Alternative Shortest Paths”中有所描述。根据该文档中描述的方法，修复节点和其下一跳中的每一个计算替换最短路径作为绕过潜在邻居节点故障的修复路径，这是通过以下方式实现的：在发生这种故障的情况下识别分组的终止点，并通过从链路状态数据库中去除故障节点来计算备用最短路径。然而，该方法需要构造明确的路径，该路径例如使用源路由来在分组自身中对路径编码。

附图说明

本发明在附图中以示例方式而非以限制方式示出，在附图中相似的标号指代类似的元件，其中：

- 图 1 是图示了根据已知方法构造修复路径的方法的网络表示；
- 图 2 是图示了根据本方法构造修复路径的方法的网络表示；
- 图 3 是图示了根据本方法构造修复路径的方法的流程图；
- 图 4 是根据本方法在 LSP 中承载的信息的示意图；
- 图 5 是图示了在对不可用节点的邻居节点处构造的转发表的图；
- 图 6 是图示了在对不可用节点的非邻居节点处构造的转发表的图；
- 图 7 是实现了构造修复路径的方法的节点的生成树图；
- 图 8 是实现了构造修复路径的方法的节点的递增改变的生成树图；
- 图 9 是图示了利用递增 SPF 构造修复路径的方法的流程图；
- 图 10 是图示了本方法的又一实现方式的网络表示；
- 图 11 是图示了针对 MHP 构造修复路径的方法的流程图；
- 图 12 是图示了本方法的又一实现方式的网络表示；
- 图 13 是图示了本方法的又一实现方式的网络表示；以及
- 图 14 是图示了可以在其上实现用于构造修复路径的方法的计算机系统的框图。

具体实施方式

描述了一种用于在数据通信网络中构造绕过不可用组件的修复路径的方法和装置。在下面的描述中，出于说明目的，给出了大量特定细节以提供对本发明的完全理解。然而，本领域技术人员应当清楚，没有这些特定细节也可以实施本发明。在其他实例中，公知的结构和设备以框图形式示出以避免不必要的混淆本发明。

这里根据下面的大纲描述实施例：

- 1.0 一般概述
- 2.0 结构和功能概述

3.0 构造修复路径的方法

4.0 实现机制—硬件概述

5.0 扩展和替换

1.0 一般概述

在本发明中实现了在前述背景技术中给出的需求以及将从下面的描述中变清楚的其他需要和目的，本发明在一个方面中包括一种在数据通信网络中构造绕过不可用组件的修复路径的方法，该数据通信网络具有作为组件的具有网络地址的节点和其间的链路。该方法包括以下步骤：在参与节点处执行从网络中的通知节点接收标识通知节点和通过其可以到达通知节点的组件的通知。该方法还包括以下步骤：从通知中导出网络修复地址，该网络修复地址用在在通知中标识的组件不可用的情况下。该方法还包括以下步骤：构造针对修复地址的修复路径。

在其他方面中，本发明包括被配置为执行前述步骤的计算机装置和计算机可读介质。

2.0 结构和功能概述

总地来说参考图 2 可以理解一种用于构造修复路径的方法，图 2 示出了本方法所应用的说明性网络图。该网络包括主节点 P（标号 200）、源节点 S 以及节点 A、B 和 C（标号 202、204、206、208），节点 S、A、B 和 C 各自经由相应链路 210、212、214、216 连接到节点 P。又一节点 D（标号 218）经由链路 220 连接到节点 B。除了指派给每个节点的标准地址以外，网络中的每个接口还被指派以一个额外的修复地址。在这里用术语称为“notvia 地址（不经由地址）”，将会意识到，这一术语是任意的、描述性的并且非限制性的。Notvia 地址的语义是去往 notvia 地址的分组必须被传递到具有该地址的路由器，而不经由该地址被指派给的接口上的相邻路由器。

例如，从节点 P 通过相应链路 210、212、214、216 到节点 S、A、B、C 的接口可以具有地址 P^a 、 P^b 、 P^c 和 P^s 。类似地，沿相反方向从节点 A、B、C 和 S 分别经由链路 212、214、216、210 到节点 P 的接口具有地址 A^p 、 B^p 、 C^p 、 S^p 。

为了修复故障，修复节点（例如节点 S）封装分组去往在故障远端上的节点接口的 notvia 地址。然后，修复路径上的节点知道它们必须向哪些节点传递分组，并且它们必须避免哪些网络组件。

参考图 2，假定 S 具有去往某一目的地 D 的分组，正常情况下其将经由 P 和 B 发送，并且 S 怀疑 P 已经发生故障，则 S 封装分组去往 B^P 。根据语义，从 S 到 B^P 的路径是从 S 到 B 的最短路径，该路径不经由 P。如果网络包含从 S 到 B 且不经过路由器 P 的路径，则分组将被成功地传递到 B。例如，分组可以沿着路径 222 被转发到节点 X (224)，然后沿着路径 226 被转发到节点 D。由于节点 X 已经为 B^P 计算出修复路径，因此它将正确地转发所封装的分组。当去往 B^P 的分组到达 B 时，B 去除封装，并向其最终目的地（节点 D）转发已修复的分组。

这可以参考图 3 进一步地理解，图 3 是以高级别图示了这里应用的方法的流程图。在框 300 中，节点 P 利用诸如 LSP 之类的通知通告其邻居 A、B、C、S 和其关联的 notvia 地址 P^a 、 P^b 、 P^c 、 P^s 。将会意识到，充当通知节点的所有其他节点也将发出类似的 LSP。结果，不仅可以构造适当的转发表，并且 notvia 地址还可用于每个节点（在每个节点发生故障或者变为不可用节点的情况下），在这种情况下 notvia 地址可以用作修复地址。因此，在框 302 中，所有参与节点不仅针对每个正常（无故障）地址，还针对每个 notvia 地址计算其下一跳。结果，每个节点构造了绕过网络中每个其他节点的修复路径，并针对相应的 notvia 地址进行存储。

在框 304 中节点 P 随后发生故障或者变为不可用的情况下，则在框 306 中邻居节点以任何适当的方式检测到故障或者被通知以故障。在邻居节点随后接收到本来要发送向作为其下一跳的故障组件的分组的情况下，在框 308 中邻居节点作为修复节点，识别其必须将该分组以隧道方式发送到的修复末端点或目标以到达其后续的目的地。在以上给出的示例中，对于目的地是 D 的分组而言，修复节点是节点 S，并且修复端点是节点 B。具体而言，这是由相应的 notvia 地址 B^P 标识的。因此，节点 S 在框 310 中沿着修复路径将分组以隧道方式发送向 B^P 。在框 312 中，每个下一跳将经封装的分组向 notvia 地址 B^P 转发，例如图 2 中的节点 X 正确地转发

分组。由于所有的参与节点都已经利用相同的修复拓扑计算出到 notvia 地址的路径，因此分组被利用正常的 IP 转发加以转发，而不需要对转发代码的扩展。在框 314 中，分组到达修复端点，修复端点对其进行解封装，并向其目的地（在所描述的示例中是目的地 D）转发原始分组，这同样是利用正常的 IP 转发进行的。

因此，只需要一级封装，并且可以修复任何故障（例如即使在网络是高度不对称的时，在这种情况下 Miles 等人描述的 P/Q 空间方法理论上可能失败）。此外，实现了绕过故障的最短可用路径，同样地也不需要诸如导向转发之类的额外的转发机制。

如下面更详细地描述的，参考图 3 描述的方法是极其健壮的，并且可以适用于很宽范围的公共网络配置和需求。例如，该方法可以应用于故障节点或故障链路的修复，并且可以结合用于单个故障点的策略，这种情况即是一个节点只提供到邻居节点或网络的一段的连通性。该方法还可以应用于多归属前缀（MHP）、共享风险组（SRG）、局域网（LAN）以及单播和多播分组的情形。另外，该方法可以利用例如多协议标签交换（MPLS）或距离向量技术实现。另外，该方法可以实现在只有网络的所有节点的子网是被使能构造根据本方法的修复路径的参与节点的情形中。

3.0 构造修复路径的方法

再次参考图 2，将会看出为了允许网络上的每个被使能节点构造针对故障的网络组件（链路或节点）的修复拓扑，每个节点必须通告其 notvia 地址以及存储在其 LSP 中的其他相关信息。参考图 4，图 4 示意性地示出了包含在由节点 P 发布的 LSP 中的信息，将会看出除了通告每个邻居和其关联的度量（例如，与相应链路相关联的代价）以外，还提供了另外的信息。例如，在列 400 中提供邻居信息和在列 402 中提供关联度量的情况下，另外在列 404 中提供了每个邻居的 notvia 地址。Notvia 地址与相应邻居相关联以使得针对邻居 A 的条目有效地指定 P^a 。只要语义由接收到 LSP 的节点识别，则 notvia 地址自身可以采取标准 IP 地址的形式，在这里示意性地示为代表 P^a 等等的 a.a.a.a。将会看出，由于网络中的每一节点都提供了类似的信息，因此每个节点可以导出用于网络上的每一 notvia 地址。

的修复路径。

因此，再次参考参考图 2 描述的示例（其中在节点 P 发生故障的情况下节点 S 将去往节点 D 的分组封装并发送向 $P^{\bar{b}}$ ），每一节点更一般地计算其将在任何可能的节点故障的情况下使用的路径。因此，每个节点使网络中的每一其他路由器发生故障，一次一个，并计算它自己的到该节点的每个邻居的最佳路由。换句话说，同样参考图 2，某些路由器 X 将依次地把每个路由器看作 P，使 P 发生故障，然后计算它自己的到由 P 的邻居通告的每个 notvia P 地址的路由，即，X 计算其到 S^P 、 A^P 、 B^P 和 C^P （在每种情况下都不经由 P）的路由。

因此，参考图 5（图 5 图示了在节点 S 处导出的转发表的相关部分），将会看出对于每个地址（列 500），导出了下一跳（列 502），指定了 notvia 地址（列 504），还实现了相应的修复地址（列 506）。例如，在目的地是节点 B 并且下一跳被计算为节点 P 的情况下，则分组将被以隧道方式发送到的修复地址 B^P 被与相应的修复下一跳存储在一起。在这种情况下，这是沿着从节点 S 到节点 X 的路径 222 的第一跳，在以上参考图 2 描述的示例中指示为沿着从节点 S 出发的链路 230 的节点 Z（标号 228）。在去往节点 D 的分组的情况下，正常的下一跳是节点 P，并且修复地址是 B^P ，因此对于被封装并发送向 B^P 的分组而言，修复下一跳再次是节点 Z。在节点 A 作为目的地地址的情况下，下一跳是节点 P，并且修复地址是 A^P ，这提供了某个修复下一跳 Ω_1 （未示出）。节点 S 的转发表中的修复地址总是去往邻居的邻居，即修复隧道的端点。然而，将会看出，在列 500 中的正常地址是 notvia 地址（例如 C^P ）的情况下，尽管下一跳被提供为沿着从节点 S 出发的链路 234 的节点 Y（标号 232），修复地址和修复下一跳也不被提供，如下更详细地所述。因此，节点 S 将利用正常转发将分组转发到 notvia 地址（当其依赖于另一节点的修复路径时），但是在进入分组已经是去往 notvia 地址的时，将不会鼓励对 notvia 地址的隧道进行修复。

将会意识到，对于去往给定目的地的给定分组而言，在下一跳发生故障的情况下识别正确的修复地址的方法可以以任何适当的方式确定，例如

根据在节点处构造的 SPT 并且例如 Miles 等人描述的。还将会看出，修复地址是根据接收的所有 LSP 导出的，并且不必从这样的节点导出，修复最终绕过该节点得以形成。例如，再次参考图 5，在分组去往节点 B 的情况下，在下一跳节点 P 发生故障时，修复地址是 B^P ，这将从节点 B（而非节点 P）接收到。参考图 6（图 6 示出了在节点 X 处构造的转发表的某些方面），将会看出，构造了类似的表，包括目的地地址字段 600、下一跳字段 602，为了简化修复地址字段和修复下一跳字段未被示出，但是这些将按照与针对上述节点 S 相同的方式被填充。在图示的片段中，将会看出，对于目的地地址 B 和 D 而言，下一跳是某一节点 Ω_2 （未示出）。在图示拓扑中，目的地地址 B^P 的下一跳正好是 Ω_2 ，因为到 B 和 D 的最短路径在修复拓扑中不改变。

相反地，节点 P 的故障可能意味着节点 X 对 B^P 的下一跳也改变。因此，如下更详细地所述的，在某些情形中可以减少在正常拓扑和修复拓扑之间的改变（即，在故障组件被从修复拓扑中切除的情况下）不影响某些节点的情况下的 SPF 计算开销。

为了减少计算开销，存在各种可能性，具体而言，其确保了每个节点不必为每个其他可能节点的故障计算整个 SPF。首先，如果节点可以识别其不处于从另一节点到修复地址的修复路径中，则其不需要计算对于该修复地址的下一跳。将会注意到，以隧道方式被发送向修复地址的分组将只来源于修复绕过故障组件的节点，即，在距离修复地址的故障组件的另一侧上的节点。因此，这可以通过从修复节点向其针对给定修复地址的修复路径中的每个其他节点发出信令来实现，在这种情况下接收到信号的每个节点将计算它自己的针对该修复地址的下一跳。或者，可以构建某种形式的“被发现”（discovered）信令路由。例如，在节点 S 将分组发送向其修复下一跳（例如节点 X）的情况下，如果该修复下一跳还未构造修复路径，则其将丢弃分组，然后计算它自己的修复下一跳。如果在更高级协议的控制下，S 在没有从最终目的地接收到确认时重发分组，则节点 X 现在将把分组转发到其修复下一跳，修复下一跳也将丢弃分组但是同时构造它自己的修复下一跳。该过程继续，直到在将发送确认的时间点处建立了修

复路径为止。

根据减少 SPF 计算开销的第二方式，可以实现递增 SPF。这可以参考图 7 理解，图 7 示出了在节点 X 构造的部分 SPT。X 首先计算基本拓扑（这时所有路由器都是工作着的），并确定它到所有节点地址的正常路径。例如，其对于节点 B 和 D 的下一跳是经由 Ω_2 的，对于节点 P 和 Y 的下一跳是经由节点 S（这里为了简化忽略了节点 Z）的，并且节点 S 对于节点 A 和 C 的下一跳是经由节点 P 的。

然后，节点 X 使路由器 P 故障，并执行递增 SPF。递增 SPF 是本领域技术人员公知的，并且不在这里详细描述，而仅仅参考图 8 出于说明目的进行总结，图 8 示出了递增 SPT。具体而言，将会看出，节点 P 已被从修复拓扑中切除，因此，节点 A 和 C 重新附接到节点 Y，但是节点 B 和 D 不受影响。因此，可以快速计算出与 P^P 相对应的 notvia 地址。节点 X 对于 A^P 和 C^P 的下一跳仍然是节点 S。然而，当节点 S 重新计算其 SPT 时，其将会鼓励节点 Y 作为其对 A^P 和 C^P 的修复下一跳。节点 X 对于节点 B 和 D 的下一跳不受影响，因此对于目的地 B、D 和 B^{P^P} 而言，下一跳是 Ω_2 。

因此，在先前经由 P 到达的所有地址被重新附接时，可以终止递增计算。然后，节点 X 回复基本拓扑，并重复依次使每个路由器故障并据此导出 notvia 地址的过程。该算法相比于一组完全的 SPF 明显廉价。具体而言，尽管路由器仍然必须为 N-1 故障计算修复路径，但是计算努力远小于 N-1 SPF。

该方法可以参考图 9 进一步理解，图 9 是图示了 SPF 的计算的流程图。在框 900 中，节点 X 构造网络中所有节点的基本拓扑。在框 902 中，节点 X 构造相应的 SPT。在框 904 中，节点 X 使节点 P 故障，在框 906 中，节点 X 构造递增 SPT，将节点 P 从针对 P^P 的修复拓扑中切除。在框 908 中，任何分离的 notvia 地址被与必须被重新附接的任何其他地址重新附接在一起，以允许 notvia 地址的重新附接，并且在框 910 中，在所有地址都被重新附接时计算终止。然后在框 912 中，转发表被填充以对于 notvia P 地址的所有下一跳，例如 A^P 、 B^P 、 C^P 等等。在框 914 中，对于网络中的每一其他参与节点重复这一程序。

将会看出，这里描述的方法可以应用于节点或链路故障两种情况下。以上提供的讨论尤其涉及节点故障，其中 notvia 地址或修复路径的构造是基于节点故障和据此导出的 SPT 的。然而，将会意识到，以类似的方式本发明也可以涉及链路故障。

例如，在图 2 的情况下，节点 S 和/或节点 P 可以额外地或可替换地基于链路 210 的故障通告 notvia 地址。在这种情况下，notvia 地址可以再次利用适当的语义进行通告，以使得接收节点识别出地址类型和所需的计算以获得链路修复下一跳。例如，以上参考图 7 到 9 所述的递增 SPT 方法将基于修复拓扑，其中并不去除节点 P（以及由此带来的节点 P 和其他节点之间的所有链路），而是只去除节点 S 和 P 之间的链路 210，并据此计算递增 SPT。

假定链路而不是节点故障对某些目的地只通过故障路由器可达的情形有帮助，则希望进行尝试以修复这些目的地，这是通过假定只发生了链路故障，并利用针对节点故障计算的 notvia 地址尝试链路修复来实现的。

为了执行链路修复，S 封装并发送向 $P^{\bar{s}}$ （即，其指示网络不经由 S 将分组传递到节点修复地址 P）。S 的所有邻居都已经计算了在 S 自身发生故障的情况下到 $P^{\bar{s}}$ 的路径。因此，S 可以将分组给予到其邻居中的任何一个（当然，除了 P）。然而，S 应当优先地在最短可用路径上将经封装分组发送向 P。该路径是通过在链路 SP 故障的情况下运行 SPF 而计算的。

在该链路修复是在存在节点故障的情况下（在这种情况下故障节点的其他邻居可以运行相冲突的修复策略）进行尝试的时，则环路以非常简单的方式得以避免。具体而言，如上参考图 5 所述，确保了对于 notvia 地址不提供修复路径。参考图 2，如果 A 是 P 的邻居，其位于从 S 到 P 的链路修复路径上，并且 P 自身已发生故障，则来自 S 的经修复分组将到达 A，并被封装发送向 $P^{\bar{s}}$ 。A 将检测到 AP 链路已发生故障，并且将正常地尝试修复分组。然而，由于对于 $P^{\bar{s}}$ 未提供修复路径，因此 A 将被强制丢弃分组，从而防止了环路的形成。

该方法在例如单个故障点的情形中是有利的。再次参考图 2，其中在节点 W（标号 240）自身可以提供与网络的一个分段的连通性时，节点 P

只提供到节点 W 的连通性，另外，如果节点 P 发生故障，则没有网络节点可以到达节点 W。因此，在节点 S 检测到在其到节点 P 的接口处的故障时，其可能希望假定只有链路 210 $\underline{\text{发生故障}}$ ，而不是节点 P。因此，S 将去往节点 W 的分组封装并发送向 P^S 。尽管节点 S 理论上可以发送分组到其邻居中的任何一个，但是其最优地使用到 P^S 的最短路径。当分组到达 P 时，P 将剥离封装，并将分组发送向 W。然而，如果 P 已确实发生故障，则分组将到达 P 的邻居（在封装中仍然去往 P^S ），并且由于该邻居没有用于 notvia 地址的修复路径，因此它将丢弃分组，因而避免了环路，如上所述。

还将会看出，在相等代价多路径路由（ECMP）可用或者下游路径路由可用的情况下，这些路由可以据此实现。例如，在下游路径或无环路备用（LFA）存在的情况下，诸如节点 S 之类的修复节点可以在适当的情形中用其来替代 notvia 修复机制。类似地，路由器可以使用 ECMP 修复来替代 notvia 修复。

可以看出，这里描述的方法可以适用于很宽范围的网络配置和情形。例如，在多归属前缀（MHP）（这是一个可经由网络中的多于一个路由器到达的前缀，因而具有“多个归属”）的情况下，可以实现本方法。图 10 示出了包括多归属前缀 Y 的网络。该网络包括根据以上参考图 2 所述的连通性的节点 S、P、B 和 C。另外，节点 S 通过路径 1004 连接到节点 Z 1000，节点 B 通过路径 1006 连接到节点 V 1002。MHP Y（标号 1008）附接在节点 Z、P 和 V 处。在这种情况下，当节点 S 发现节点 P 已发生故障时，其需要将去往 Y（通常可通过 P 到达）的 MHP 分组发送向仍然能够到达 Y 的备用路由。

参考图 11，图 11 是图示了在转发 MHP 分组中涉及到的步骤的流程图，在步骤 1102 中，节点 S 在节点 P 故障的情况下以上文参考图 7 到 9 所述的方式运行其递增 SPF，直到 P 的所有邻居和 MHP Y（其将是第二最近的）的情形被附接。节点 S 可以以任何适当方式识别何时所有的适当节点被重新附接，例如通过记录基本 SPF 期间的所有 MHP 并确保所有都被重新附接到递增 SPF。

在框 1104 中，节点 S 封装分组并发送向被识别为最近备用节点的前缀附接点。在这是在节点 Z (S 可以在不使用节点 P 的情况下到达) 处的情况下，仍然需要封装，因为网络中的其他节点将不会知道节点 P 的故障，并且可能将分组环回 S。因此，在框 1106 中，节点 Z 解封装分组并将其转发到节点 X。将会注意到，节点 Z 将包括转发指令，该转发指令确保了用于 MHP Y 的解封装分组必须被沿着连接链路 1010 转发到节点 Y，以避免在存在经由 P 的较低代价路由的情况下潜在环路。

在备用路由器是节点 V (节点 S 经由节点 P 和节点 B 到达) 的情况下，节点 S 必须首先利用正常的 notvia 修复机制修复到节点 B 的分组。因此，节点 S 将针对节点 Y 的分组封装并发送向 B^P 。当分组到达节点 B 时，节点 B 对其解封装，并看出分组是去往 MHP Y 的。由于节点 B 已经确立节点 P 已发生故障并且将会执行与节点 S 相同的计算，因此其将会认识到其对于 MHP Y 而言的最近附接点是节点 V，从而遵循与在节点 S 处执行的相同计算，并正常对其进行修复，从而对其进行封装并发送向 V。

在替换方法中，前缀 Y 可被认为附接到伪节点 N，伪节点 N 依次又连接到节点 B、V、Z 中的每一个，在这种情况下，相应接口被指派以 notvia 地址，并且实现上述的修复路径方法。将会认识到，“伪节点”的概念是本领域技术人员公知的，并且提供了用于处理 MHP 和 LAN 的有用的拓扑发明，从而这里不需要进行详细描述。

这里描述的方法还适合于用在包含共享风险群组 (SRG) 的网络中，SRG 是被识别为具有同时发生故障 (例如由于公共的物理位置) 的概率的网络组件的群组。图 12 是图示了包括共享风险群组的网络的示意性网络图，并且与图 2 共同的元件用同一标号。另外，该网络包括节点 P' (1200) 和 P'' (1202)，其中 P' 连接在节点 P 和 A 之间，P'' 连接到节点 A 的另一侧。节点 F (1204) 连接到节点 P''，并且节点 G (1206) 连接到节点 F。节点 D 和 E (标号 1208、1210) 分别连接到节点 P' 和 P''。节点 J 和 K (标号 1212、1214) 分别连接到节点 D 和 E。节点 H (标号 1216) 连接到节点 C。

节点 P、P'、P'' 形成了 SRG，因此，如果假定 SRG 的所有成员同时

发生故障，则 notvia 地址的范围指示“不经由整个 SRG”。换句话说， $\bar{A^P} = A\{\overline{p, p', p''}\}$ 。所有路由器计算到将 SRG 连接到网络的其余部分的接口的每个 notvia 地址的 notvia 路由，该计算是通过同时使单个递增 SPF 中的 SRG 的所有成员（P、P'、P''）发生故障而进行的。为了识别 SRG，每个节点保持属于相应 SRG 的网络中的节点的记录。

然后，当与 SRG 成员相邻的路由器进行修复时，其封装分组并发送向具有从 SRG 到目的地的最低代价路由的路由器的 notvia 地址。例如参考图 12，在 S 接收去往节点 H 的分组（正常情况下该分组将经由节点 P 和 C 转发）的情况下，在检测到 SRG 的故障时，节点 S 将封装分组并经由适当的连通性（未示出）发送向节点 C^P 。类似地，节点 S 将封装针对节点 J 的分组并发送向 D^P ，等等。

还将会注意到，除了创建针对 SRG 群组的 notvia 地址以外，个别 notvia 地址也可以针对群组的每个成员形成。例如，这可以用在首先是有益的时候，例如在群组的成员包括单个故障点的情况下。以与用于链路故障保护相同的方式避免了环路，这是因为对于邻居 notvia 地址不存储修复路径。

还将会看出，这里描述的方法可以实现在参考图 13 的局域网中，图 13 是图示了 LAN 的网络图。节点 S (1300) 连接到伪节点 N (1302)，节点 N 自身连接到 LAN 的成员，节点 P (1304)、节点 P' (1306) 和节点 P'' (1308)。节点 P 连接到节点 B (1310)。LAN 可被视作 SRG 的特殊情形，因此针对 LAN 的所有成员（这里称为 L）的故障创建 notvia 地址。然后节点 S 可以提供用于流量的修复路径，对于该流量而言其具有避免了 L 的替换路径。如果 L 能够测试 LAN 的成员（在这种情况下其只能针对 LAN 上已知的故障节点使用 notvia 地址），则可以采用更复杂的方法。由于利用了 SRG 计算，因此只需要单个递增 SPF 来计算用于群组 L 的 notvia 修复路径。

当 S 发现其丢失了到 P 的连通性时，其并不确定故障是其自己的到 LAN 的接口，是 LAN 自身，是 P 的 LAN 接口还是节点 P。在不进行进一步的诊断的情况下，S 必须通过 P 和 B 发送的流量，以不经由 P、N

(即, 不经由 P 且不经由 N) 发送向 B, 其保守假设是整个 LAN 和 P 都已发生故障。P 是单个故障点的目的地必须照常利用避免了某一接口的地址发送向 P, 通过该接口 P 可从 S 到达 (即, 不经由 N 到 P), 对于路由器 P' 和 P'' 类似。连接到 LAN 的每个路由器必须照常针对每个邻居通告一个 notvia 地址。另外, LAN 上的每个路由器必须通告不经由伪节点 (N) 的额外地址。另外, 连接到 LAN 的路由器的每个邻居必须通告两个 notvia 地址, 照常地一个不经由该邻居, 而另一个不经由该邻居或伪节点, 实际上如果这一程度的连通性已经通过使用点对点链路而实现, 则将已经通告了其之外的一个地址。如上所述, 为了明确地诊断故障网络组件, S 相关来自 P 的连通性报告和 LAN 上的其他路由器中的一个或多个 (在这种情况下是 P' 和 P'')。如果其仅仅丢失了到 P 的连通性, 则可以推论 LAN 仍然在工作, 并且故障发生在 P 处, 或将 P 连接到 LAN 的接口处。然后, 其将按通常方式修复 B 而不经由 P (并且对于 P 是单个故障点的目的地而言, P 不经由 N)。如果 S 丢失了到 LAN 上的多于一个路由器的连通性, 其可以推断出故障只发生在 LAN 处, 并且可以修复 P、P' 和 P'' 而不经由 N, 这同样是按通常方式进行的。如果 S 发生故障, 则路由器 A 需要能够修复 LAN 上的每一路由器而不经由 S, 这是利用 notvia S 地址的集合按通常方式实现的。提供了较低的故障后连通性但是使用较少地址的替换方法是将 LAN 和其所有的连接路由器看作是单个 SRG。从而, 不经由 LAN 的地址 P (P1) 将要求不经由连接到 LAN 的任何路由器到达 P。在这种情况下, 当 S 检测到 P 已经发生故障时, S 会将经由 P 和 B 到达的流量发送向 B, 而不经由 LAN 或附接到 LAN (即, 附接到 B1) 的任何路由器。任何只能通过 P 到达的目的地都将被寻址到 P, 而不经由 LAN 或附接到 LAN 的任何路由器 (当然除了 P)。

该方法同样可用在进一步的实现方式中。例如在多播分组的情况下, 多播流量按与单播类似的方式加以修复。然而, 多播转发器能够使用多播分组被寻址到的 notvia 地址作为预期接收器接口的指示, 从而正确地运行所需的反向路径转发 (RPF) 检查。

此外, 可以采用这样的技术, 例如在次末端节点处剥离隧道头部以发

送向某一节点，该节点将把未经封装的分组转发到期望修复端点。该方法在 Michael Shand 等人于 2003 年 7 月 15 日提交的题为 “Method and Apparatus for Forwarding a Tunneled Packet in a Data Communications Network” 的未决专利申请 No. 10/620,866 (“Shand 等 III”) 中有所描述，该申请的全部内容通过引用结合于此以用于所有目的，就好像被在这里完全公开一样。

还将会看出，这里描述的方法可以实现在替换的路由协议或转发范例中。

一种这样的范例是 MPLS（多协议标签交换）。MPLS 是一种对于本领域技术人员来说公知的协议，并且在文档 “Multi Protocol Label Switching Architecture” 中有所描述，该文档可以在万维网上的域 “ietf.org” 的目录 “rfc” 中的文件 “rfc3031.txt” 处得到。根据 MPLS，建立用于源-目的地对的完整路径，并且用于在路径中的相邻路由器之间转发分组以及头部或“标签”所需的值被预先挂到分组上。标签被用于引导分组去往正确的接口和下一跳。标签在 IP 或其他头部之前，从而允许更小的外出头部。

用于源-目的地对的路径（称为标签交换路径，LSP）可以根据各种不同的方法建立。一种这样的方法是标签分发协议（LDP），其中路径中的每个路由器将其标签发送到路径上根据其 IP 路由表确定的下一跳路由器。或者，可以调用资源保留协议（RSVP），在这种情况下，例如网络管理员可以管理路径，从而提供了严格的源路由。具体而言，LDP 将按通常方式分发用于 notvia 地址的标签。因此，notvia 修复机制可以用于提供 MPLS 网络中的修复路径，这是通过首先推入修复点用来转发分组的标签，然后推入与实现修复所需要的 notvia 地址相对应的标签而实现的。例如，参考图 2，在节点 S 有去往节点 D 的分组（节点 D 正常情况下经由节点 P 和 B 到达）的情况下，在节点 P 已发生故障时，节点 S 首先推入 B 针对 D 的标签，然后推入其到 B^P （节点 X）的下一跳到达 B^P 所需的标签，即 $[B_d, N_{B_p}]$ 。再一次 notvia 地址已被传播并且被认识，因此该方法被简单地实现。将会注意到，节点 S 需要节点 B 针对 D 的标签， B_d 。这可

以按任何适当的方式实现，例如有向 LDP 会话。为了获得所有适当的标签，节点 S 可以与节点 P 的邻居中的每一个（在节点修复的情况下）或所有的出口路由器（在 SRG 的情况下）执行有向 LDP 会话。在链路修复的情况下，节点 S 将已经具有节点 P 针对 D 的标签， P_D ，因为 P 是邻居节点。当然，可以采用任何适当的用于获得修复点针对目的地节点的标签的方式，例如 Swallow 等人描述的任何适当的方法。

或者，该方法同样可以利用诸如距离或路径向量之类的路由向量协议实现，其中通告节点向所有其他节点通告其到目的地的完整路由或向量。在这种情况下，节点 P 通过忽略针对其每个邻居 A^P 、 B^P 、 C^P 、 S^P 的向量，来有效地通告其 notvia 地址。因此，如果 P 发生故障，则在任何节点处没有任何关于在任何情况下如何经由 P 到达 B 的信息，因此其有效地具有经由另一节点选择的 notvia 路由。因此，所有节点将计算不穿越节点 P 的路由。

还将会看出，所述方法可以跨网络实现，其中所有路由器都能够计算 notvia 地址并因而充当参与节点，其中只有路由器的一部分被启用或者其中某些路由器被部分启用。在所有路由器都被启用并且具有修复地址能力的情况下，很清楚该方法可以非常简单地按上述方式实现。在某些路由器被启用而其他路由器不被启用的情况下，当计算 notvia 路径时，未被启用的节点被从基本拓扑中去除。因此，所计算的到 notvia 地址的任何一条路由都不会尝试穿越未被启用的节点，从而未被启用的节点将不会接收到它们未被配备来进行处理的 notvia 地址。路由器可能被部分启用，例如以使得它们能够导出它们自己的修复路径，并因而接收分组并正确地转发分组到 notvia 地址，而不是它们自己充当执行上述的封装和转发步骤的邻居路由器。在这种情况下，部分启用的路由器可以被包括在修复拓扑中，但是它们自身不能鼓励修复。

这里描述的方法被实现的方式（其中软件、固件、硬件或其任意组合并且具有任何适当的代码改变）对于本领域技术人员来说是清楚的，这里不需要进行详细描述。例如，对诸如内部网关协议（IGP）之类的通信协议的扩展可能需要扩展。具体而言，直接连接到受保护的网络组件的每个

被启用的路由器将通告针对该组件的 notvia 地址，通告方式使得受保护组件和 notvia 地址之间的关联可以由网络中的其他路由器确定，如上所述。此外，被启用的路由器可以通告其用于计算并实现 IGP 中的 notvia 路由的能力以及它们支持的封装类型，例如 IP in IP、GRE、L2TPv3，所有这些都是本领域技术人员公知的。

被指派为 notvia 地址的地址可以是例如从网络的私有地址空间取得的任何适当的地址。

任何适当的封装都可用于执行 notvia 修复，例如 IP in IP、GRE 或 L2TPv3。类似地，可以采用任何备用封装方案，只要封装路由器和被封装分组作为修复点寻址到的路由器具有处理所选的封装类型的公共能力。

作为上述方法的结果，应用了一种简单的程序，根据该程序，只需要单级封装，并且由于所有节点都有效地计算最短修复路径，因此其提供了针对任何故障的最短修复路径。此外，将会看出，解封装分组可以“返回（backtrack）”，即被转发到其已经经由隧道到达但是仍然利用正常转发加以转发的一个或多个节点，而不向目的地环回，这是因为修复点离目的地更近。

4.0 实现机制—硬件概述

图 14 是图示了在其上可以实现本方法的计算机系统 140 的框图。该方法是利用运行在诸如路由器设备之类的网络元件上的一个或多个计算机程序实现的。从而，在该实施例中，计算机系统 140 是路由器。

计算机系统 140 包括用于传输信息的总线 142 或其他通信机构，以及与总线 142 相耦合用于处理信息的处理器 144。计算机系统 140 还包括主存储器 146，例如随机访问存储器（RAM）、闪存或其他动态存储设备，其耦合到总线 142，用于存储信息和由处理器 144 执行的指令。主存储器 146 还可以用于存储在处理器 144 执行指令的执行期间的临时变量或其他中间信息。计算机系统 140 还包括只读存储器（ROM）148 或其他静态存储设备，其耦合到总线 142，用于存储静态信息和处理器 144 的指令。提供了存储设备 150，例如磁盘、闪存或光盘，其耦合到总线 142，用于存储信息和指令。

通信接口 158 可以耦合到总线 142，用于向处理器 144 传输信息和命令选择。接口 158 是传统的串行接口，例如 RS-232 或 RS-422 接口。外部终端 152 或其他计算机系统连接到计算机系统 140，并且利用接口 158 向其提供命令。运行在计算机系统 140 中的固件或软件提供终端接口或基于字符的命令接口，以使得可以向计算机系统给予外部命令。

交换系统 156 耦合到总线 142，并且具有输入接口和到外部网络元件的相应输出接口（本地指定为 159）。外部网络元件可以包括多个额外路由器 160 或耦合到一个或多个主机或路由器的本地网络，或者具有一个或多个服务器的全球网络（例如因特网）。交换系统 156 根据预定义的协议和惯例将到达输入接口的信息流量交换到输出接口 159。例如，交换系统 156 与处理器 144 协同操作可以确定到达输入接口的数据分组的目的地，并利用输出接口将其发送向正确目的地。目的地可以包括主机、服务器、其他末端站、或本地网络或因特网中的其他路由和交换设备。

计算机系统 140 实现为充当转发数据的上述方法中的参与节点、修复节点或通知节点的路由器。该实现方式由计算机系统 140 响应于处理器 144 执行包含在主存储器 146 中的一条或多条指令的一个或多个序列而提供。这些指令可以从另一计算机可读介质（例如存储设备 150）读取到主存储器 146 中。包含在主存储器 146 中的指令序列的执行使得处理器 144 执行这里描述的进程步骤。多处理配置中的一个或多个处理器也可以用于执行包含在主存储器 146 中的指令序列。在替换实施例中，硬连线电路可以用于替代软件指令或者与软件指令相组合以实现本方法。从而，实施例并不限于硬件电路和软件的任何特定组合。

这里所用的术语“计算机可读介质”指参与向处理器 144 提供指令以供执行的任何介质。这种介质可以采取很多形式，包括但不限于非易失性介质、易失性介质和传输介质。非易失性介质包括例如光盘或磁盘，例如存储设备 150。易失性介质包括动态存储器，例如主存储器 146。传输介质包括同轴电缆、铜线和光纤，包括包含总线 142 的线路。传输介质还可以采取声波或电磁波的形式，例如在无线电波和红外数据通信期间生成的声波或电磁波。

计算机可读介质的公共形式包括例如软盘、柔性盘、硬盘、磁带或任何其他磁介质、CD-ROM、任何其他光介质、穿孔卡、纸带、任何其他具有孔图案的物理介质、RAM、PROM 和 EPROM、FLASH-EPROM、任何其他存储器芯片或盒、下文中描述的载波、或计算机可以读取的任何其他介质。

各种形式的计算机可读介质都可以用于运送一条或多条指令的一个或多个序列到处理器 144 以供执行。例如，该指令可以首先承载在远程计算机的磁盘上。远程计算机可以将指令加载到其动态存储器中，并利用调制解调器通过电话线发送指令。计算机系统 140 本地的调制解调器可以接收电话线上的数据，并使用红外发送器来将数据转换为红外信号。耦合到总线 142 的红外检测器可以接收在红外信号中承载的数据，并将数据放到总线 142 上。总线 142 将数据运送到主存储器 146，处理器 144 从主存储器 146 取得指令并执行指令。主存储器 146 接收的指令可以可选地在处理器 144 的执行之前或之后存储在存储设备 150 上。

接口 159 还提供到连接到本地网络的网络链路的双向数据通信耦合。例如，接口 159 可以是集成业务数字网络（ISDN）卡或调制解调器，以提供到相应类型电话线的数据通信连接。又例如，接口 159 可以是局域网（LAN）卡以提供到兼容 LAN 的数据通信连接。还可以实现无线链路。在任何这种实现方式中，接口 159 发送并接收电的、电磁的或光信号，这些信号承载了代表各类信息的数字数据流。

网络链路一般通过一个或多个网络提供到其他数据设备的数据通信。例如，网络链路可以通过本地网络提供到主机计算机或由因特网服务提供商（ISP）操作的数据设备的连接。ISP 又通过全球分组数据通信网络（现在通常称为“因特网”）提供数据通信服务。本地网络和因特网都使用承载数字数据流的电的、电磁的或光信号。经过各种网络的信号以及网络链路上的并且经过接口 159 的信号（这些信号承载去往和来自计算机系统 140 的数字数据）是传输信息的载波的示例性形式。

计算机系统 140 可以通过（一个或多个）网络、网络链路和接口 159 发送消息并接收数据，包括程序代码。在因特网示例中，服务器可能通过

因特网、ISP、本地网络和通信接口 158 发送应用程序所请求的代码。一个这样下载的程序提供了这里描述的方法。

处理器 144 可以在接收时执行所接收的代码和/或存储在存储设备 150 或其他非易失性存储装置中以供以后执行。以这种方式，计算机系统 140 可以获得载波形式的应用代码。

5.0 扩展和替换

在前述说明书中，已经参考本发明的特定实施例描述了本发明。但是，应当清楚，可以对本发明进行各种修改和改变，而不脱离本发明的更宽广的精神和范围。因此，说明书和附图应被当作是说明性的，而非限制性的。

任何适当的路由协议和机制以及转发范例都可用来实现本发明。给出的方法步骤可以根据示例和实施例按任何适当的顺序和方面执行，所描述的这些示例和实施例在适当时可以被并列或交换。例如，该方法可以利用链路状态协议和任何转发范例（例如 MPLS）实现，链路状态协议例如是中间系统-中间系统（IS-IS）或开放最短路径优先（OSPF）或路由向量协议。该方法可以应用自任何拓扑的任何网络中，并且可以用于网络中的任何组件改变，例如链路或节点故障，或者可以应用于由管理员执行的网络组件的引入和去除。

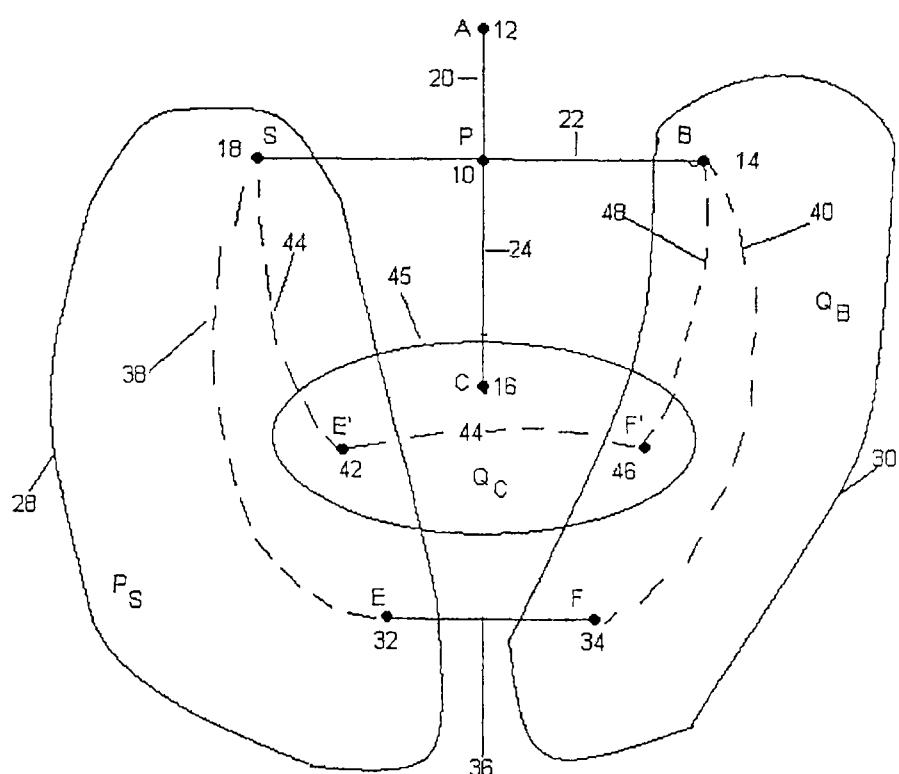


图1

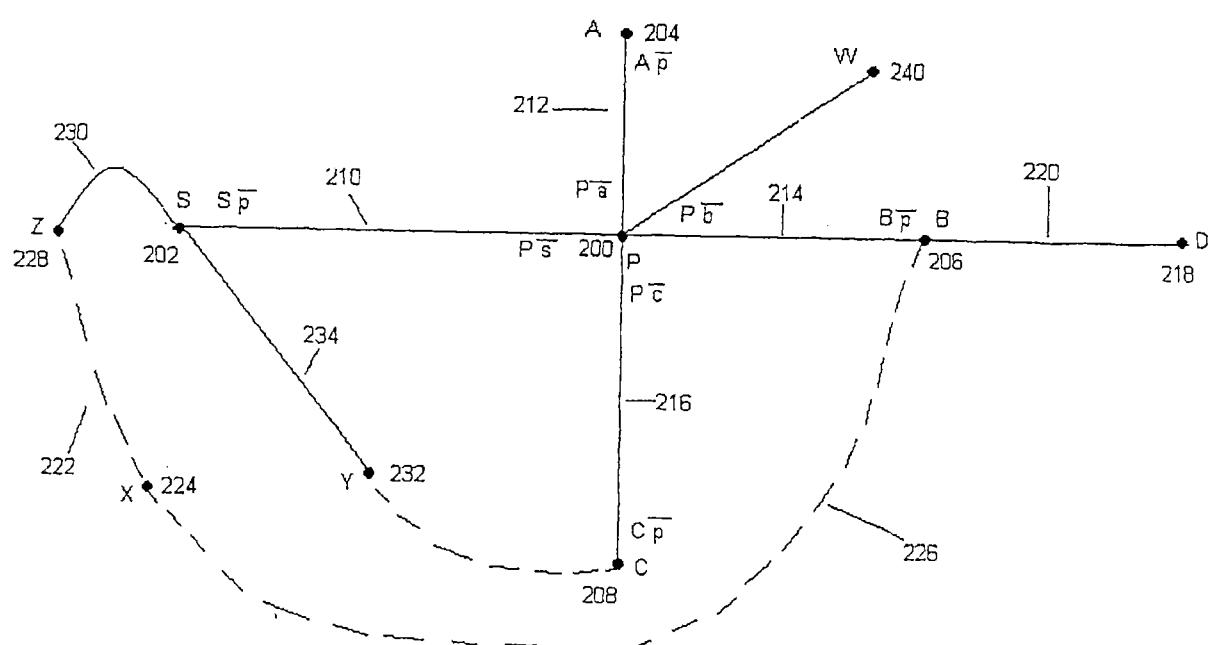


图2

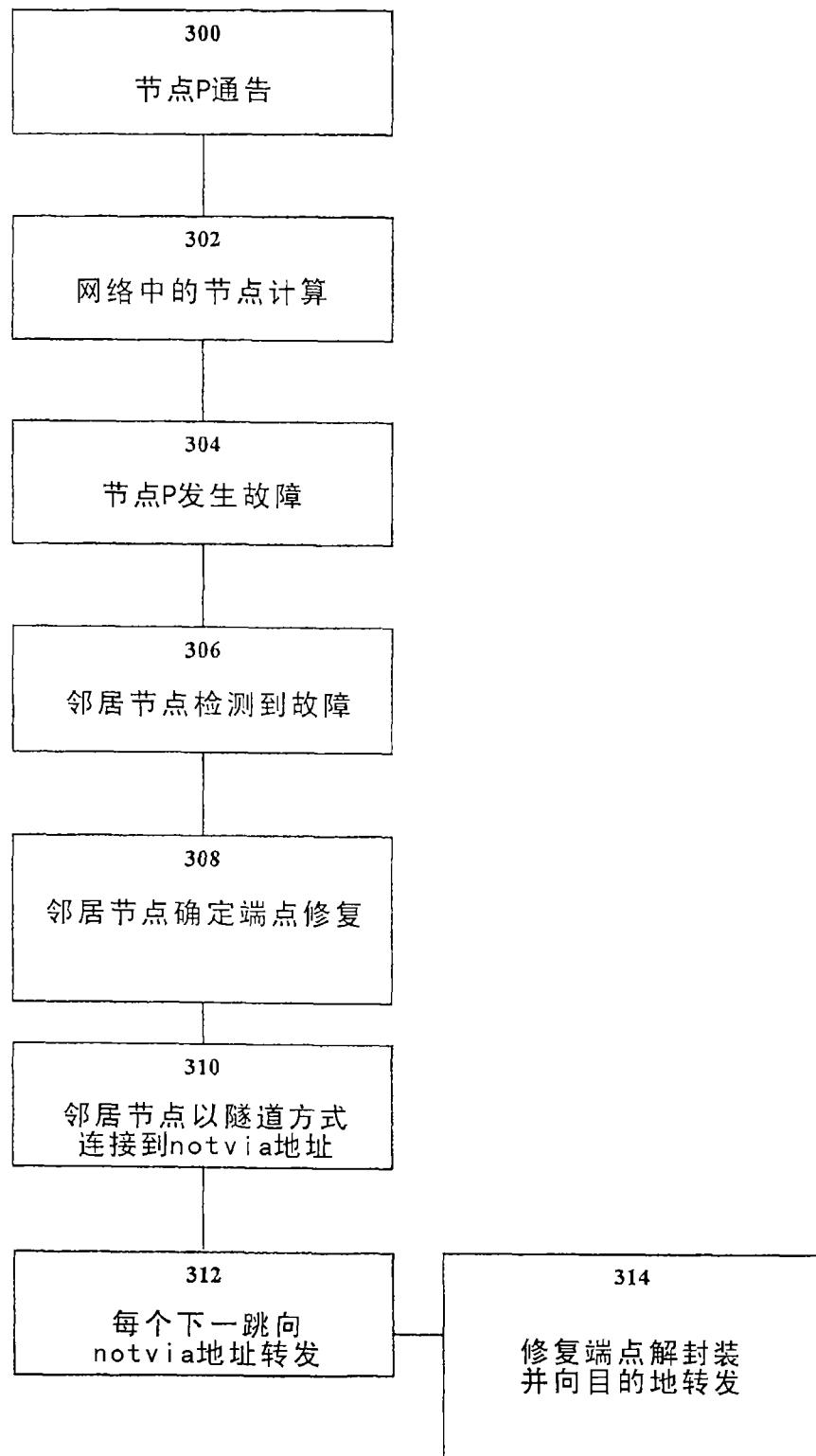


图3

邻居	度量	NOTVIA地址
A	∞	a.a.a.a.(P \bar{a})
B	β	b.b.b.b (P \bar{b})
C	γ	c.c.c.c (P \bar{c})
S	ζ	d.d.d.d (P \bar{d})

图4

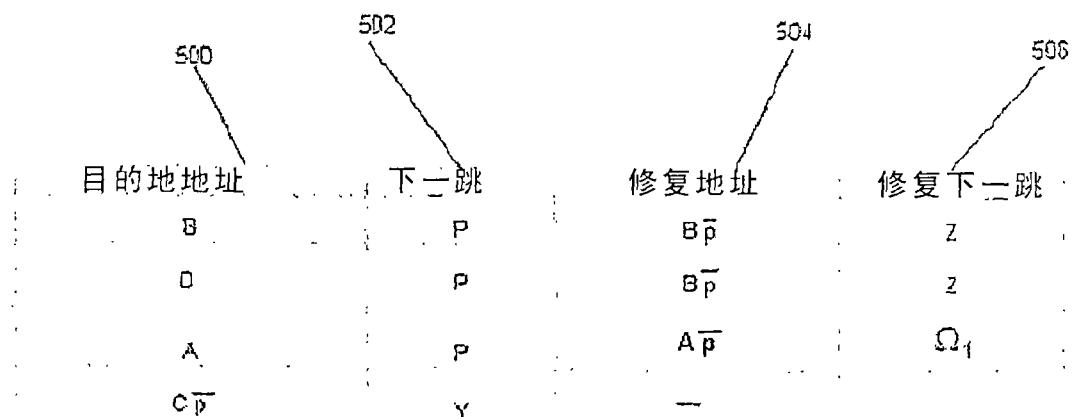


图5

目的地地址	下一跳
D	Q ₂
B	Q ₃
6P	Q ₂

图6

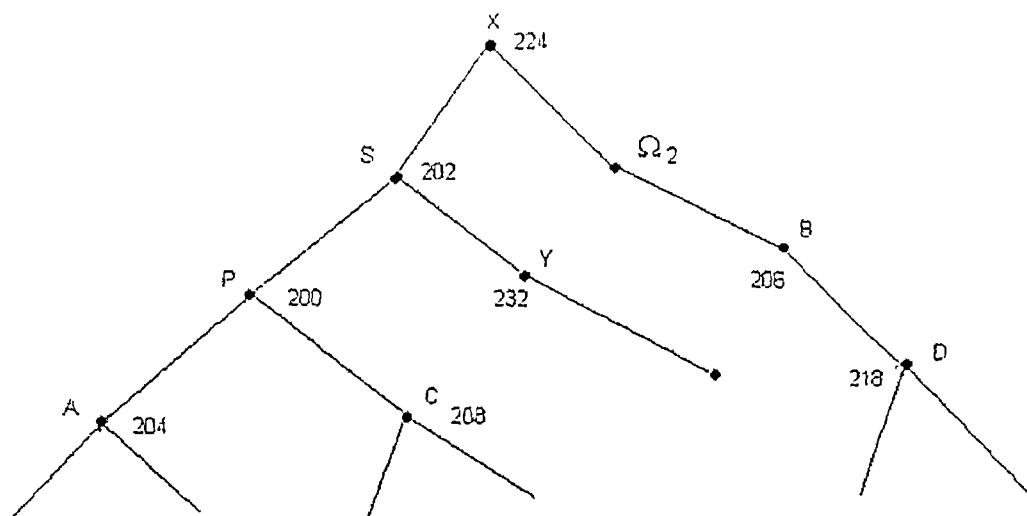


图7

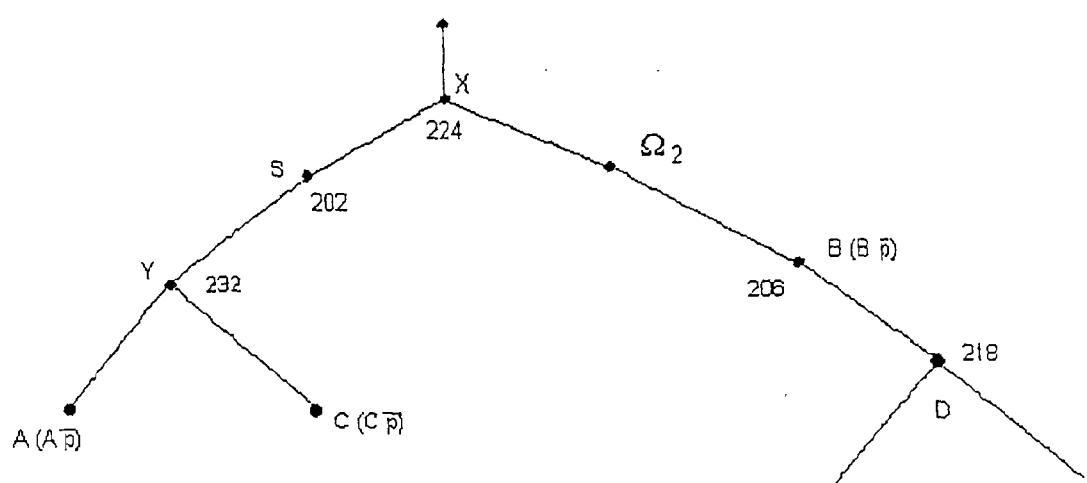


图8

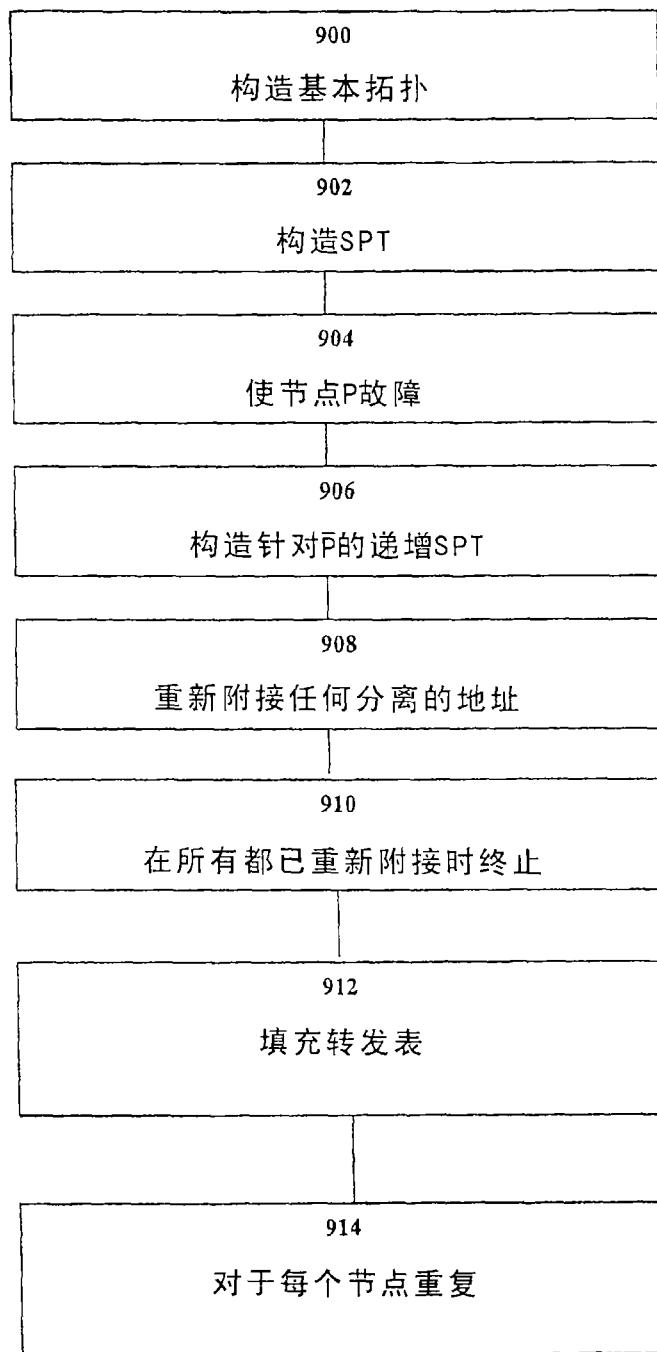


图9

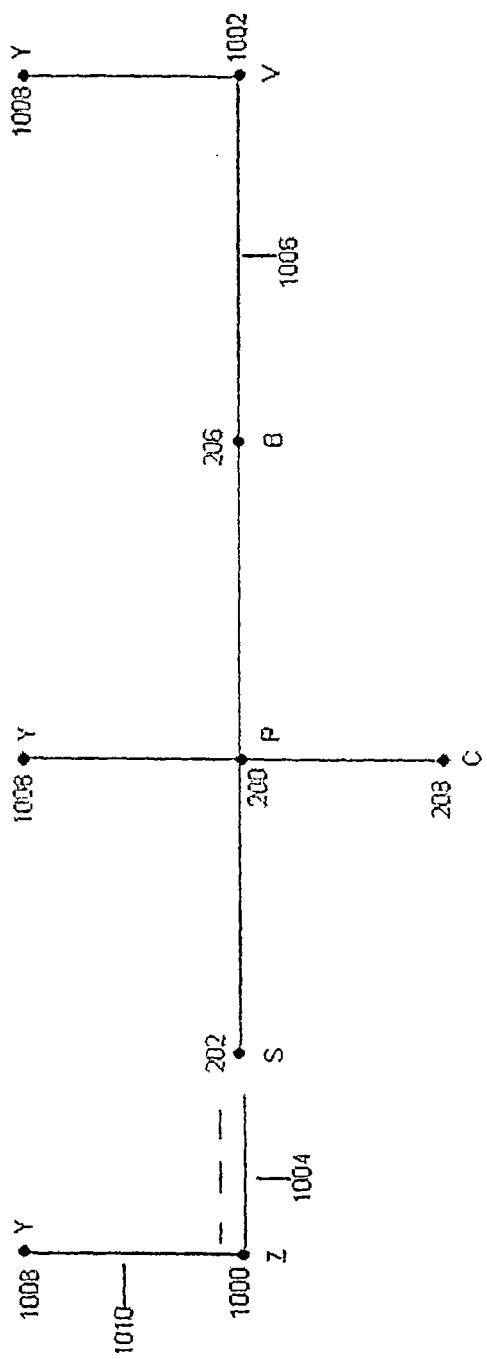


图10



图 11

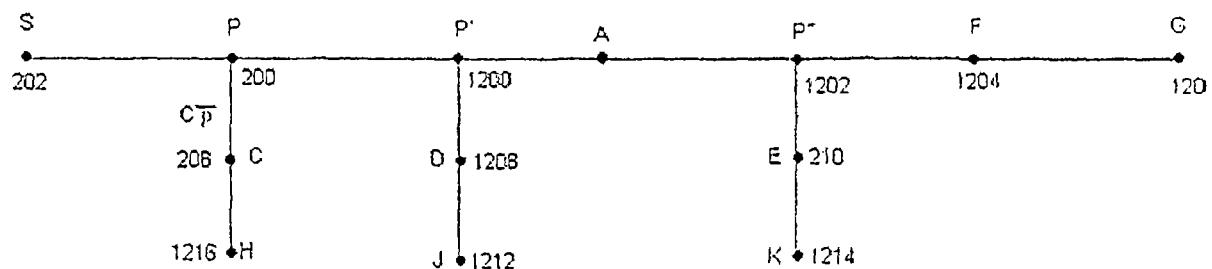


图 12

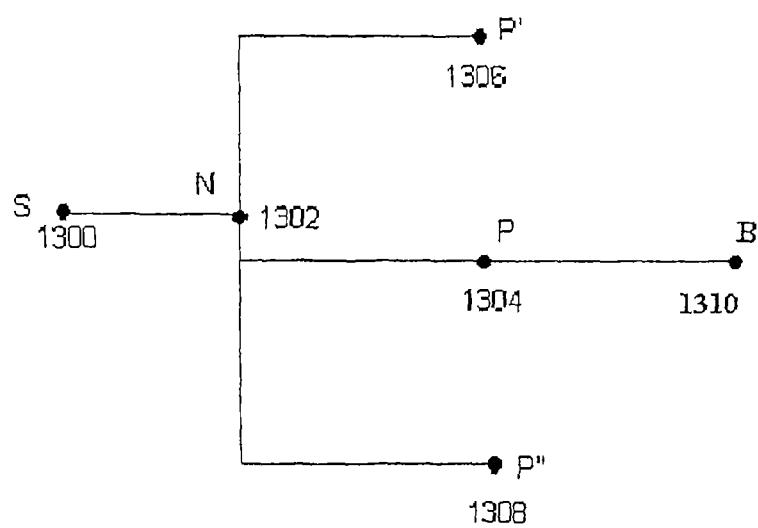


图 13

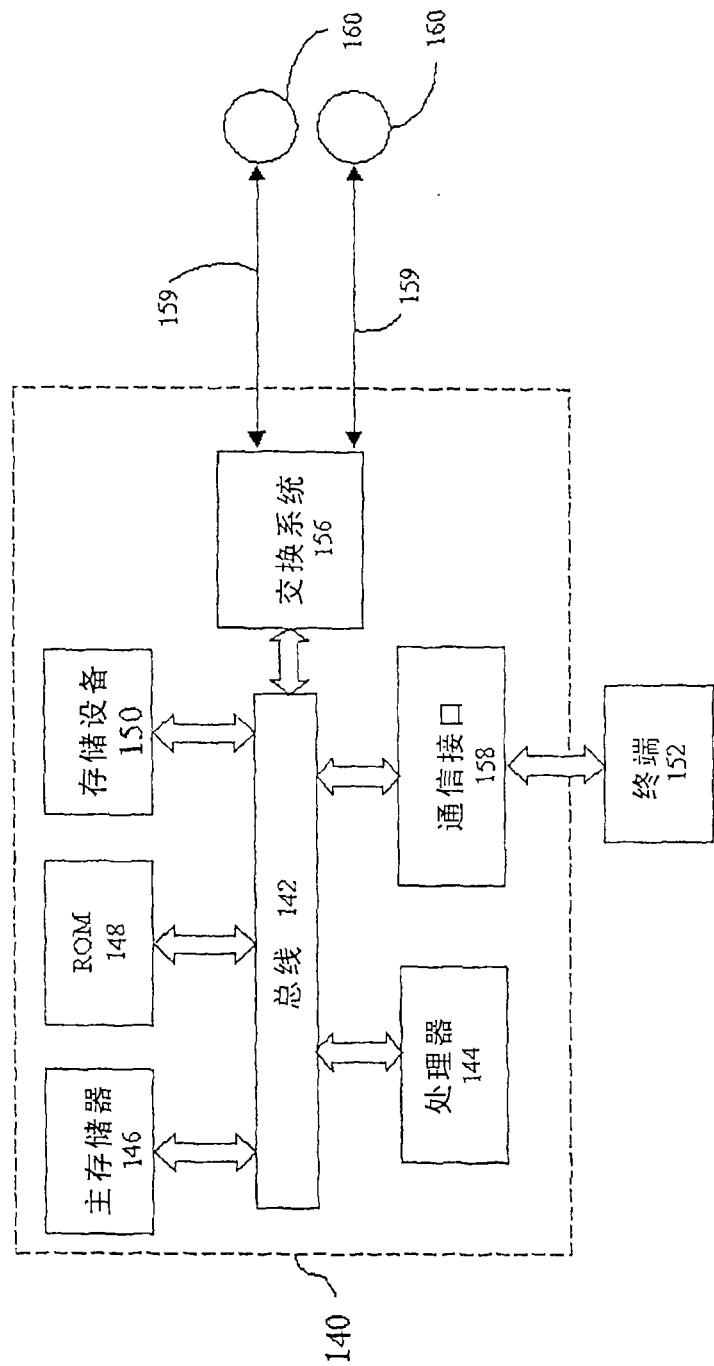


图 14