



# (12) 发明专利申请

(10) 申请公布号 CN 113615132 A

(43) 申请公布日 2021. 11. 05

(21) 申请号 202080021632.X

(51) Int. Cl.

(22) 申请日 2020.03.13

H04L 12/707 (2006.01)

H04L 12/721 (2006.01)

(30) 优先权数据

62/819,062 2019.03.15 US

(85) PCT国际申请进入国家阶段日

2021.09.15

(86) PCT国际申请的申请数据

PCT/US2020/022603 2020.03.13

(87) PCT国际申请的公布数据

WO2020/190707 EN 2020.09.24

(71) 申请人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72) 发明人 陈怀谟

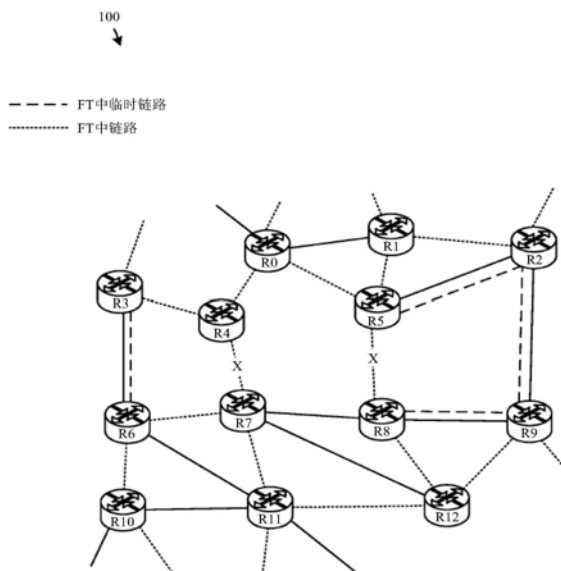
权利要求书4页 说明书13页 附图12页

(54) 发明名称

快速泛洪拓扑保护

(57) 摘要

一种由节点执行的并行泛洪拓扑修复方法，用于修复泛洪拓扑。该并行泛洪拓扑修复方法检测泛洪拓扑上的失效链路和/或失效节点，确定故障链路和/或故障节点是否导致泛洪拓扑分裂，以及响应于泛洪拓扑分裂的确定，通过执行局部泛洪拓扑修复处理来修复泛洪拓扑。



1. 一种并行泛洪拓扑修复方法,其特征在于,包括:  
检测网络故障,其中所述网络故障为故障链路和故障节点中的至少一个;  
确定泛洪拓扑中存在所述网络故障;  
确定所述泛洪拓扑是否被拆分;  
修复所述泛洪拓扑。
2. 一种由用于修复泛洪拓扑的节点执行的并行泛洪拓扑修复方法,其特征在于,所述并行泛洪拓扑修复方法包括:  
检测网络故障,其中所述网络故障为故障链路和故障节点中的至少一个;  
确定所述泛洪拓扑中是否存在所述网络故障;  
当确定所述泛洪拓扑中存在所述网络故障时,确定所述泛洪拓扑是否被拆分;  
当确定所述泛洪拓扑被拆分时,执行本地泛洪拓扑修复过程来修复所述泛洪拓扑。
3. 根据权利要求1或2所述的并行泛洪拓扑修复方法,其特征在于,所述用于修复所述泛洪拓扑的本地泛洪拓扑修复过程包括:  
计算所述网络故障的备用路径来修复所述泛洪拓扑;  
确定所述备用路径是否经过所述节点;  
当确定所述备用路径经过所述节点时,确定所述泛洪拓扑中是否存在所述备用路径上所述节点的本地链路;  
当确定所述泛洪拓扑中不存在所述备用路径上所述节点的所述本地链路时,将所述备用路径上所述节点的所述本地链路暂时添加到所述泛洪拓扑中,直到构建成新的泛洪拓扑。
4. 根据权利要求1至3中任一项所述的并行泛洪拓扑修复方法,其特征在于,通过链路中断事件在本地检测到所述故障链路。
5. 根据权利要求1至3中任一项所述的并行泛洪拓扑修复方法,其特征在于,通过识别所述节点的链路状态数据库(link state database,简称LSDB)中不处于接收链路状态的链路,检测到所述故障链路。
6. 根据权利要求1至5中任一项所述的并行泛洪拓扑修复方法,其特征在于,通过确定所述故障节点附接的每条链路均发生故障,检测到所述故障节点。
7. 根据权利要求1至6中任一项所述的并行泛洪拓扑修复方法,其特征在于,当所述LSDB中存在所述故障链路的标志指示所述泛洪拓扑中存在所述故障链路时,所述泛洪拓扑中存在所述故障链路。
8. 根据权利要求1至6中任一项所述的并行泛洪拓扑修复方法,其特征在于,当所述故障链路处于用于所述泛洪拓扑的所述LSDB中的链路状态时,所述泛洪拓扑中存在所述故障链路。
9. 根据权利要求1至6中任一项所述的并行泛洪拓扑修复方法,其特征在于,当所述泛洪拓扑中存在连接到所述故障节点的链路时,所述泛洪拓扑中存在所述故障节点。
10. 根据权利要求1至6中任一项所述的并行泛洪拓扑修复方法,其特征在于,当所述故障节点处于用于所述泛洪拓扑的所述LSDB中的链路状态时,所述泛洪拓扑中存在所述故障节点。
11. 根据权利要求1至10中任一项所述的并行泛洪拓扑修复方法,其特征在于,所述确

定所述泛洪拓扑是否被所述故障链路拆分包括：确定所述泛洪拓扑中所述故障链路连接的第一节点和第二节点之间是否存在路径；当所述泛洪拓扑中所述故障链路连接的所述第一节点和所述第二节点之间不存在路径时，确定所述泛洪拓扑被所述故障链路拆分；当所述泛洪拓扑中所述故障链路连接的所述第一节点和所述第二节点之间存在路径时，确定所述泛洪拓扑没有被所述故障链路拆分。

12. 根据权利要求1至10中任一项所述的并行泛洪拓扑修复方法，其特征在于，所述确定所述泛洪拓扑是否被所述故障链路拆分包括：

通过遍历所述泛洪拓扑，确定所述泛洪拓扑中节点的第一数量；

将所述泛洪拓扑中节点的所述第一数量与所述泛洪拓扑中在所述LSDB中维护的节点的第二数量进行比较；

当所述泛洪拓扑中节点的所述第一数量小于所述泛洪拓扑中节点的所述第二数量时，确定所述泛洪拓扑被拆分。

13. 根据权利要求1至10中任一项所述的并行泛洪拓扑修复方法，其特征在于，确定所述泛洪拓扑是否被所述故障节点拆分包括：

确定所述节点与所述故障节点之间是否有直连链路；

确定所述泛洪拓扑中所述节点与和所述故障节点直接连接的每个其他节点之间是否存在路径；

当所述泛洪拓扑中所述节点与和所述故障节点直接连接的每个其他节点之间不存在路径时，确定所述泛洪拓扑被所述故障节点拆分。

14. 根据权利要求1至10中任一项所述的并行泛洪拓扑修复方法，其特征在于，确定所述泛洪拓扑是否被所述故障节点拆分包括：

遍历所述泛洪拓扑，从而确定所述泛洪拓扑中节点的第一数量；

将所述泛洪拓扑中节点的所述第一数量与所述泛洪拓扑中在所述LSDB中维护的节点的第二数量进行比较；

当所述泛洪拓扑中节点的所述第一数量小于所述泛洪拓扑中节点的所述第二数量时，确定所述泛洪拓扑被拆分。

15. 根据权利要求3至14中任一项所述的并行泛洪拓扑修复方法，其特征在于，所述计算所述故障链路的备用路径来修复所述泛洪拓扑包括：计算所述故障链路连接的第一节点和第二节点之间的所述备用路径。

16. 根据权利要求3至14中任一项所述的并行泛洪拓扑修复方法，其特征在于，计算所述故障节点的所述备用路径来修复所述泛洪拓扑包括：计算和所述故障节点直接连接的第一节点与和所述故障节点直接连接的每个其他节点之间的所述备用路径。

17. 根据权利要求1至14中任一项所述的并行泛洪拓扑修复方法，其特征在于，对于导致所述泛洪拓扑被拆分的每个网络故障，重复所述本地泛洪拓扑修复过程。

18. 一种由用于修复泛洪拓扑的附接到网络故障的节点执行的顺序泛洪拓扑修复方法，其特征在于，所述顺序泛洪拓扑修复方法包括：

检测网络故障，其中所述网络故障为故障链路和故障节点中的至少一个；

确定所述泛洪拓扑中是否存在所述网络故障；

当确定所述泛洪拓扑中存在所述网络故障时，确定所述泛洪拓扑是否被拆分；

当确定所述泛洪拓扑被拆分时,执行顺序泛洪拓扑修复过程来修复所述泛洪拓扑。

19. 根据权利要求18所述的方法,其特征在于,所述顺序泛洪拓扑修复过程包括:

计算从所述节点到与所述网络故障连接的每个其他节点的第一备用路径;

确定所述泛洪拓扑中是否存在所述备用路径上所述节点的本地链路;

当确定所述泛洪拓扑中不存在所述备用路径上所述节点的所述本地链路时,将所述备用路径上所述节点的所述本地链路暂时添加到所述泛洪拓扑中,直到构建成新的泛洪拓扑;

沿着所述备用路径向下一跳节点发送泛洪拓扑修复请求。

20. 一种由用于修复泛洪拓扑的节点执行的顺序泛洪拓扑修复方法,其特征在于,所述顺序泛洪拓扑修复方法包括:

接收泛洪拓扑修复请求;

确定所述泛洪拓扑中是否存在备用路径上所述节点的本地链路;

当确定所述泛洪拓扑中不存在所述备用路径上所述节点的所述本地链路时,将所述备用路径上所述节点的所述本地链路暂时添加到所述泛洪拓扑中,直到构建成新的泛洪拓扑;

沿着所述备用路径向下一跳节点发送所述泛洪拓扑修复请求。

21. 根据权利要求20所述的方法,其特征在于,还包括:从所述泛洪拓扑修复请求中提取所述备用路径。

22. 根据权利要求20所述的方法,其特征在于,还包括:当所述泛洪拓扑修复请求中不包括所述备用路径时,计算从所述节点到所述备用路径的端节点的备用路径。

23. 根据权利要求18至22中任一项所述的方法,其特征在于,所述泛洪拓扑修复请求为扩展Hello报文。

24. 根据权利要求23所述的方法,其特征在于,所述扩展Hello报文是开放式最短路径优先(Open Shortest Path First,简称OSPF)Hello报文,其中包括所述OSPF Hello报文中的扩展选项和标志类型长度值(Extended Options and Flags Type-Length-Value,简称EOF-TLV)的扩展选项和标志字段中的备用路径比特(backup path bit,简称BP-bit)。

25. 根据权利要求24所述的方法,其特征在于,所述BP比特设置为1,以指示连接发送节点和接收节点的备用路径上的链路被所述发送节点暂时添加到所述泛洪拓扑中。

26. 根据权利要求24和25中任一项所述的方法,其特征在于,所述OSPF Hello报文包括在所述OSPF Hello报文的本地链路信令(link-local signaling,简称LLS)数据块中定义的备用路径端点TLV,所述备用路径端点TLV包括所述备用路径的两个端节点。

27. 根据权利要求18至22中任一项所述的方法,其特征在于,所述泛洪拓扑修复请求为扩展中间系统到中间系统(Intermediate System to Intermediate System,简称IS-IS)Hello报文。

28. 根据权利要求27所述的方法,其特征在于,所述扩展Hello报文是IS-IS Hello报文,其中包括所述IS-IS Hello报文中的EOF-TLV的扩展选项和标志字段中的备用路径比特(backup path bit,简称BP-bit)。

29. 根据权利要求28所述的方法,其特征在于,所述BP比特设置为1,以指示连接发送节点和接收节点的备用路径上链路被所述发送节点暂时添加到所述泛洪拓扑中。

30. 根据权利要求28和29中任一项所述的方法,其特征在于,所述IS-IS Hello报文包括备用路径端点TLV,其中所述备用路径端点TLV包括所述备用路径的两个端节点。

31. 一种网络节点,其特征在于,包括:

存储器,用于存储指令;

处理器,耦合至所述存储器,其中所述处理器用于执行所述指令,以使所述网络节点执行任一项前述方法的权利要求。

## 快速泛洪拓扑保护

[0001] 相关申请案交叉申请

[0002] 本申请要求于2019年3月15日由Huaimo Chen等人递交的发明名称为“快速并行泛洪拓扑保护”的第62/819,062号美国暂时专利申请的优先权,其全部内容通过引用结合在本申请中。

### 技术领域

[0003] 本申请涉及网络通信,更具体地,涉及用于快速泛洪拓扑保护的方法和系统。

### 背景技术

[0004] 路由协议规定了路由器之间的通信方式,并分配信息使这些路由器能够在计算机网络中任意两个节点之间选择路由。内部网关协议 (Interior Gateway Protocol, 简称IGP) 是一种用于在自治系统 (autonomous system, 简称AS) 内的网关 (通常是路由器) 之间交换路由信息的协议。所述路由信息则可以用来路由网络层协议, 例如, 互联网协议 (Internet Protocol, 简称IP) 报文。AS是代表单一管理实体或域的一个或多个网络运营商控制下的连接的IP路由前缀的集合, 向互联网 (例如, 企业局域网系统) 呈现通用且明确定义的路由策略。

[0005] IGP内分不同的类型。第一类IGP称为链路状态路由协议。所述链路状态路由协议由网络中的每个交换节点 (即, 准备转发报文的节点; 在互联网中, 这些被称为路由器) 实现。链路状态路由的基本概念是, 每个节点构建到网络的连接图, 以图形的形式显示哪些节点连接到哪些其他节点。然后, 每个节点独立计算从这个节点到网络中每个可能目的地的最佳逻辑路径或最佳下一跳接口。之后, 每个最佳路径集合将形成每个节点的路由表。链路状态路由协议的示例包括开放式最短路径优先 (Open Shortest Path First, 简称OSPF) 路由协议和中间系统到中间系统 (Intermediate System to Intermediate System, 简称IS-IS或ISIS)。

[0006] OSPF是互联网工程任务组 (Internet Engineering Task Force, 简称IETF) 指定的标准路由协议。OSPF通过链路状态通告 (Link State Advertisement, 简称LSA) 在路由器之间交换路由信息。一个区域内的每个路由器将在这个区域内泛洪第一类LSA (也称为路由器LSA)。LSA封装在OSPF报文和IP报文之后。区域是具有相同区域号的基于OSPF的网络、路由器和链路的逻辑组。属于同一区域的路由器保留整个区域的拓扑数据库。路由器LSA包含路由器所属区域中直连链路的信息 (例如, 具有所述路由器的所有直连链路的列表)。这些直连链路被泛洪给这个区域中所有路由器。如果路由器是区域边界路由器 (area border router, 简称ABR), 则所述路由器为所述路由器连接的所有区域生成第一类LSA, 并将这些LSA发送给相应区域中的所有邻居。

[0007] IS-IS是国际标准化组织 (International Organization for Standardization, 简称ISO) 标准化的路由协议。IS-IS通过链路状态协议数据单元 (Link State Protocol, LSP) 在路由器之间交换路由信息。LSP是网络路由器根据列出路由器邻居的链路状态路由

协议生成的信息报文。LSP报文还可以进一步被定义为确定任何相邻路由器和关联网络的名称、以及到任何相邻路由器和关联网络的成本或距离的特殊数据报。如果发生链路故障，这些信息用于高效确定新邻居是什么，以及有需要的情况下改变链路的成本。

[0008] OSPF和IS-IS的另外一些区别在于：OSPF支持非广播多重访问 (non-broadcast multiple access, 简称NBMA) 网络和点对多点链路，而IS-IS不支持；IS-IS在数据链路层 (L2) 运行，而OSPF在网络层 (L3) 运行；OSPF支持虚拟链路，而IS-IS不支持。

[0009] 无论采用哪种IGP路由协议，随着网络变得越来越大，当所述网络发生链路故障等变化时，IGP路由收敛速度越来越慢。当网络中每个路由器的所有组件 (包括路由信息库 (routing information base, 简称RIB) 和转发信息库 (forwarding information base, 简称FIB) 以及软件和硬件表) 具备最新的路由变化时，IGP会发生融合，使得在最佳出口 (或说下一跳) 接口上路由条目被成功转发。

[0010] 即使底层 (物理) 拓扑未被拆分，一个或多个故障也可以拆分泛洪拓扑 (flooding topology, 简称FT)。需要时间计算并构建新的FT，并将这个新的FT从主节点泛洪到网络中的每个节点。此时，链路状态数据库之间 (link state database, 简称LSDB) 可能不同步，网络收敛速度变慢。

## 发明内容

[0011] 所公开的实施例旨在解决上述问题。在一个实施例中，当几乎相同的时间发生多个故障时，网络区域中的每个节点并行检测链路故障和/或节点故障，检测FT拆分，并修复一个或多个链路故障和/或节点故障。在一个实施例中，为了修复所述一个或多个链路故障和/或节点故障，FT中的每个节点或部分节点并行计算针对导致FT拆分的链路故障和/或节点故障的备用路径。当FT中的链路中断时，备用路径指的是路径或链路序列，只要能为连接拆分的FT的两部分提供连接。当FT中的节点发生故障时，FT可以分为两个或两个以上部分。在这种情况下，需要两条或两条以上备用路径将所有被拆分的部分连接成一条。在一个实施例中，沿着所述备用路径的节点通过剩余FT和备用路径将链路状态 (LSA或LSP) 泛洪到每个活动节点，以修复所述一个或多个链路故障和/或节点故障。公开的实施例提供了在发生多次故障时更快地检测和修复FT拆分的过程。这会导致更快的网络融合，致使丢包更少。

[0012] 如本文所引用的，术语“并行”是指两个或两个以上网络节点或节点无特定顺序独立执行所公开的实施例中描述的过程。所述多个网络节点的独立执行过程可以同时进行，但不要求完全同步。

[0013] 在一个替换实施例中，FT拆分也可以顺序修复。在一个实施例中，只有故障旁边的节点检测导致FT拆分的FT中的链路故障和/或节点故障。例如，在一个实施例中，当附接到故障点 (即，故障链路或故障节点) 的节点首先检测到故障时，所述节点确定FT是否被拆分，然后计算备用路径，并将所述节点的本地链路暂时添加到FT中，直到构建成新的FT，从而自行修复这些故障 (即，修复FT拆分)。然后，所述节点请求相关节点 (即，需要将在备用路径上节点的本地链路暂时添加到FT中的节点，如果在FT中尚未添加这些链路的话) 来修复这些故障。在一个实施例中，顺序进程比并行进程慢，从而使得网络收敛时间更长。

[0014] 第一方面涉及一种用于修复泛洪拓扑的节点执行的并行泛洪拓扑修复方法。所述并行泛洪拓扑修复方法包括：检测网络故障，其中所述网络故障为故障链路和故障节点中

的至少一个；确定所述泛洪拓扑中是否存在所述网络故障；当确定所述泛洪拓扑中存在所述网络故障时，确定所述泛洪拓扑是否被拆分；当确定所述泛洪拓扑被拆分时，执行本地泛洪拓扑修复过程来修复所述泛洪拓扑。

[0015] 根据第一方面，在所述计算机实现方法的第一种实现方式中，所述本地泛洪拓扑修复过程包括：计算所述网络故障的备用路径来修复所述泛洪拓扑；确定所述备用路径是否经过所述节点；当确定所述备用路径经过所述节点时，确定所述泛洪拓扑是否存在所述备用路径上所述节点的本地链路中；当确定所述泛洪拓扑中不存在所述备用路径上的所述节点的所述本地链路时，将所述备用路径上的所述节点的所述本地链路暂时添加到所述泛洪拓扑中，直到构建成新的泛洪拓扑。

[0016] 根据第一方面或第一方面的任一前述实现方式，在所述计算机实现方法的第二种实现方式中，通过链路中断事件在本地检测到所述故障链路。

[0017] 根据第一方面或第一方面的任一前述实现方式，在所述计算机实现方法的第三种实现方式中，识别节点的链路状态数据库(link state database, 简称LSDB)中不处于接收链路状态的链路，检测到所述故障链路。

[0018] 根据第一方面或第一方面的任一前述实现方式，在所述计算机实现方法的第四种实现方式中，确定所述故障节点附接的每条链路均发生故障，检测到所述故障节点。

[0019] 根据第一方面或第一方面的任一前述实现方式，在所述计算机实现方法的第五种实现方式中，当所述LSDB中存在所述故障链路的标志指示所述泛洪拓扑中存在所述故障链路时，所述泛洪拓扑中存在所述故障链路。

[0020] 根据第一方面或第一方面的任一前述实现方式，在所述计算机实现方法的第六种实现方式中，当所述故障链路处于用于所述泛洪拓扑的所述LSDB中的链路状态时，所述泛洪拓扑中存在所述故障链路。

[0021] 根据第一方面或第一方面的任一前述实现方式，在所述计算机实现方法的第七种实现方式中，当所述泛洪拓扑中存在连接所述故障节点的链路时，所述泛洪拓扑中存在所述故障链路。

[0022] 根据第一方面或第一方面的任一前述实现方式，在所述计算机实现方法的第八种实现方式中，当所述故障节点处于用于所述泛洪拓扑的所述LSDB中的链路状态时，所述泛洪拓扑中存在所述故障节点。

[0023] 根据第一方面或第一方面的任一前述实现方式，在所述计算机实现方法的第九种实现方式中，所述确定所述泛洪拓扑是否被所述故障链路拆分包括：确定所述泛洪拓扑中所述故障链路连接的第一节点和第二节点之间是否存在路径；当所述泛洪拓扑中所述故障链路连接的所述第一节点和所述第二节点之间不存在路径时，确定所述泛洪拓扑被所述故障链路拆分；当所述泛洪拓扑中所述故障链路连接的所述第一节点和所述第二节点之间存在路径时，确定所述泛洪拓扑没有被所述故障链路拆分。

[0024] 根据第一方面或第一方面的任一前述实现方式，在所述计算机实现方法的第十种实现方式中，所述确定所述泛洪拓扑是否被所述故障链路拆分包括：遍历所述泛洪拓扑，确定所述泛洪拓扑中节点的第一数量；将所述泛洪拓扑中节点的所述第一数量与所述泛洪拓扑中在所述LSDB中维护的节点的第二数量进行比较；当所述泛洪拓扑中节点的所述第一数量小于所述泛洪拓扑中节点的所述第二数量时，确定所述泛洪拓扑被拆分。



[0025] 根据第一方面或第一方面的任一前述实现方式,在所述计算机实现方法的第十一种实现方式中,确定所述泛洪拓扑是否被所述故障节点拆分包括:确定节点与所述故障节点之间是否有直连链路;确定所述泛洪拓扑中所述节点与和所述故障节点直接连接的每个其他节点之间是否存在路径;当所述泛洪拓扑中所述节点与和所述故障节点直接连接的每个其他节点之间不存在路径时,确定所述泛洪拓扑被所述故障节点拆分。

[0026] 根据第一方面或第一方面的任一前述实现方式,在所述计算机实现方法的第十二种实现方式中,确定所述泛洪拓扑是否被所述故障节点拆分包括:遍历所述泛洪拓扑,确定所述泛洪拓扑中节点的第一数量;将所述泛洪拓扑中节点的所述第一数量与所述泛洪拓扑中在所述LSDB中维护的节点的第二数量进行比较;当所述泛洪拓扑中节点的所述第一数量小于所述泛洪拓扑中节点的所述第二数量时,确定所述泛洪拓扑被拆分。

[0027] 根据第一方面或第一方面的任一前述实现方式,在所述计算机实现方法的第十三种实现方式中,所述计算所述故障链路的备用路径来修复所述泛洪拓扑包括:计算所述故障链路连接的第一节点和第二节点之间的所述备用路径。

[0028] 根据第一方面或第一方面的任一前述实现方式,在所述计算机实现方法的第十四种实现方式中,计算所述故障节点的所述备用路径来修复所述泛洪拓扑包括:计算和所述故障节点直接连接的第一节点与和所述故障节点直接连接的每个其他节点之间的所述备用路径。

[0029] 根据第一方面或第一方面的任一前述实现方式,在所述计算机实现方法的第十五种实现方式中,对于导致所述泛洪拓扑被拆分的每个网络故障,重复本地泛洪拓扑修复过程。

[0030] 第二方面涉及一种由用于修复泛洪拓扑的附接到网络故障的节点执行的顺序泛洪拓扑修复方法。所述顺序泛洪拓扑修复方法包括:检测网络故障,其中所述网络故障为故障链路和故障节点中的至少一个;确定所述泛洪拓扑中是否存在所述网络故障;当确定所述泛洪拓扑中存在所述网络故障时,确定所述泛洪拓扑是否被拆分;当确定所述泛洪拓扑被拆分时,执行顺序泛洪拓扑修复过程来修复所述泛洪拓扑。

[0031] 根据第二方面,在所述计算机实现方法的第一种实现方式中,所述顺序泛洪拓扑修复过程包括:计算从节点到和所述网络故障相连的每个其他节点的第一备用路径;确定所述泛洪拓扑中是否存在所述备用路径上节点的本地链路;当确定所述泛洪拓扑中不存在所述备用路径上的所述节点的所述本地链路时,将所述备用路径上的所述节点的所述本地链路暂时添加到所述泛洪拓扑中,直到构建成新的泛洪拓扑;沿着所述备用路径向下一跳节点发送泛洪拓扑修复请求。

[0032] 第三方面涉及一种由用于修复泛洪拓扑的附接到网络故障的节点执行的顺序泛洪拓扑修复方法。所述顺序泛洪拓扑修复方法包括:接收泛洪拓扑修复请求;确定所述泛洪拓扑中是否存在备用路径上节点的本地链路;当确定所述泛洪拓扑中不存在所述备用路径上所述节点的所述本地链路时,将所述备用路径上所述节点的所述本地链路暂时添加到所述泛洪拓扑中,直到构建成新的泛洪拓扑;沿着所述备用路径向下一跳节点发送所述泛洪拓扑修复请求。

[0033] 根据第三方面,在所述计算机实现方法的第一种实现方式中,所述顺序泛洪拓扑修复方法还包括:从所述泛洪拓扑修复请求中提取所述备用路径。

[0034] 根据第三方面或第三方面的任一前述实现方式,在所述计算机实现方法的第二种实现方式中,所述顺序泛洪拓扑修复方法还包括:当所述泛洪拓扑修复请求中不包括所述备用路径时,计算从所述节点到所述备用路径的端节点的备用路径。

[0035] 根据第三方面或第三方面的任一前述实现方式,在所述计算机实现方法的第三种实现方式中,所述泛洪拓扑修复请求为扩展Hello报文。

[0036] 根据第三方面或第三方面的任一前述实现方式,在所述计算机实现方法的第四种实现方式中,所述扩展Hello报文是开放式最短路径优先(Open Shortest Path First,简称OSPF)Hello报文,其中包括所述OSPF Hello报文中的扩展选项和标志类型长度值(Extended Options and Flags Type-Length-Value,简称EOF-TLV)的扩展选项和标志字段中的备用路径比特(backup path bit,简称BP-bit)。

[0037] 根据第三方面或第三方面的任一前述实现方式,在所述计算机实现方法的第五种实现方式中,所述BP比特设置为1,以指示连接发送节点和接收节点的备用路径上链路被所述发送节点暂时添加到所述泛洪拓扑中。

[0038] 根据第三方面或第三方面的任一前述实现方式,在所述计算机实现方法的第六种实现方式中,所述OSPF Hello报文包括在所述OSPF Hello报文的本地链路信令(link-local signaling,简称LLS)数据块中定义的备用路径端点TLV,所述备用路径端点TLV包括所述备用路径的两个端节点。

[0039] 根据第三方面或第三方面的任一前述实现方式,在所述计算机实现方法的第七种实现方式中,所述泛洪拓扑修复请求为扩展中间系统到中间系统(Intermediate System to Intermediate System,简称IS-IS)Hello报文。

[0040] 根据第三方面或第三方面的任一前述实现方式,在所述计算机实现方法的第八种实现方式中,所述扩展Hello报文是IS-IS Hello报文,其中包括所述IS-IS Hello报文中的EOF-TLV的扩展选项和标志字段中的备用路径比特(backup path bit,简称BP-bit)。

[0041] 根据第三方面或第三方面的任一前述实现方式,在所述计算机实现方法的第九种实现方式中,所述BP比特设置为1,以指示连接发送节点和接收节点的备用路径上的链路被所述发送节点暂时添加到所述泛洪拓扑中。

[0042] 根据第三方面或第三方面的任一前述实现方式,在所述计算机实现方法的第十种实现方式中,所述IS-IS Hello报文包括备用路径端点TLV,其中所述备用路径端点TLV包括所述备用路径的两个端节点。

[0043] 第四方面涉及网络节点,其中包括存储指令的存储器;耦合至所述存储器的处理器,其中所述处理器用于执行所述指令,以使所述网络节点执行任一前述方面或者前述方面的任一实现方式。

[0044] 为表述清楚,在本发明范围内,任何一个前述实施例都可以与任何一个或多个其他前述实施例结合以创建新的实施例。

[0045] 通过以下结合附图和权利要求的详细描述,这些以及其它特征和优点将会被更清楚地理解。

## 附图说明

[0046] 为了更透彻地理解本发明,现参阅结合附图和具体实施方式而描述的以下简要说

明,其中的相同附图标记表示相同部分。

[0047] 图1为本发明实施例提供的网络拓扑和泛洪拓扑(flooding topology,简称FT)的示意图。

[0048] 图2为本发明实施例提供的拆分的FT的示意图。

[0049] 图3为本发明实施例提供的修复的FT的示意图。

[0050] 图4为本发明实施例提供的并行FT修复过程的流程图。

[0051] 图5为本发明实施例提供的本地FT修复过程的流程图。

[0052] 图6为本发明实施例提供的顺序FT修复过程的流程图。

[0053] 图7为本发明实施例提供的顺序FT修复的流程图。

[0054] 图8为本发明实施例提供的扩展选项和标志(Extended Options and Flags,简称EOF)类型长度值(Type-Length-Value,简称TLV)的数据结构示意图。

[0055] 图9为本发明实施例提供的OSPF备用路径端点TLV的数据结构示意图。

[0056] 图10为本发明实施例提供的EOF-TLV的数据结构示意图。

[0057] 图11为本发明实施例提供的IS-IS备用路径端点TLV的数据结构示意图。

[0058] 图12为本发明实施例提供的网元的示意图。

## 具体实施方式

[0059] 首先应理解,尽管下文提供一项或多项实施例的说明性实施方案,但所公开的系统 and/或方法可使用任何数目的技术来实施,无论该技术是当前已知还是现有的。本发明决不应限于下文所说明的说明性实施方案、附图和技术,包括本文所说明并描述的示例性设计和实施方案,而是可在所附权利要求书的范围以及其等效物的完整范围内修改。

[0060] 如上所述,即使底层(物理)拓扑未被拆分,一个或多个故障也可以拆分FT。需要时间计算并构建新的FT,并将这个新的FT从主节点泛洪到网络中的每个节点。此时,LSDB之间可能不同步,网络收敛速度变慢。本发明公开的实施例提供用于快速并行保护FT的各种系统和方法,以期解决上述问题。

[0061] 在一个实施例中,FT中的每个节点或部分节点并行计算针对导致FT拆分的链路故障和/或节点故障的备用路径。沿着所述备用路径的节点通过剩余FT和备用路径将链路状态泛洪到每个活动节点,以修复所述一个或多个链路故障和/或节点故障。剩余FT为从FT中去除了故障链路和故障节点之后得到的FT拓扑。

[0062] 图1为本发明实施例提供的网络拓扑100和泛洪拓扑(flooding topology,简称FT)的示意图。所述网络拓扑100包括路由器R0至R12以及连接所述网络拓扑100中路由器的描绘的链路(实线和虚线)。所述FT是所述(物理)网络拓扑100的子网络。在所示实施例中,所述FT通过连接路由器R0至R12的虚线示出。所述FT与所述网络拓扑100到每个节点(即路由器R0至R12)具有相同的可达性。所述FT用于分发所述网络拓扑100中每个路由器的链路状态信息(即,对于OSPF,在FT中泛洪LSA,对于IS-IS,泛洪LSP)。在图1中,假设R0的标识(identifier,简称ID)小于R1的ID,R1的ID小于R2的ID,以此类推(即R0的ID<R1的ID<R2的ID<……<R12的ID)。在正常运行条件下(即,所述网络中没有故障),每个节点(例如,图1中的路由器R0至R12)具有相同的链路状态数据库(link state database,简称LSDB)、相同的网络拓扑100和相同的FT。所述LSDB是OSPF构建的数据库,基于每个网络节点/路由器在LSA

中分发的信息。所述LSDB在同一区域内的路由器之间同步。

[0063] 当所述网络拓扑100中的所述FT上存在链路故障或节点故障时,每个网络节点或指定网络节点计算所述网络拓扑100中针对故障的备用路径。例如,网络节点可以执行算法以获得路由器R<sub>i</sub>到路由器R<sub>j</sub>的最小跳数路径,以解决FT中链路R<sub>i</sub>—R<sub>j</sub>故障的问题。路由器R<sub>i</sub>和路由器R<sub>j</sub>可以是所述网络拓扑100内所述FT中的任意两个路由器(例如,R<sub>5</sub>和R<sub>8</sub>)。与最短路径优先(Shortest Path First,简称SPF)一样,备用路径中使用的任何链路都必须都是双向的。在一个实施例中,对于所述FT中所述链路R<sub>i</sub>—R<sub>j</sub>的故障,每个节点(R<sub>k</sub>)获取从R<sub>i</sub>到R<sub>j</sub>的所有最小跳数路径来计算链路R<sub>i</sub>—R<sub>j</sub>(假设R<sub>i</sub>的ID<R<sub>j</sub>的ID)的唯一备用路径,其中每个最小跳数路径在FT(HC-FT)值具有跳数。如果存在具有最小HC-FT值的多条路径,则网络节点选择包含沿着从目的路由器R<sub>j</sub>到源路由器R<sub>i</sub>的方向具有较小或最小远程节点ID的链路的路径。然后,如果所述FT中没有节点的本地链路,则备用路径上的节点R<sub>k</sub>将备用路径上所述节点的本地链路暂时添加到FT中,从而启用链路R<sub>i</sub>—R<sub>j</sub>的备用路径,在备用路径上进行暂时的泛洪。

[0064] 图2为本发明实施例提供的拆分的FT的示意图。在所示实施例中,当FT中的链路R<sub>4</sub>至R<sub>7</sub>(R<sub>4</sub>—R<sub>7</sub>)和链路R<sub>5</sub>—R<sub>8</sub>同时发生故障时,所述FT(如图1所示)被拆分。这里,如图2所示,所述FT被拆分为两个分区,FT分区A和FT分区B。FT分区A包括路由器R<sub>0</sub>、R<sub>1</sub>、R<sub>2</sub>、R<sub>3</sub>、R<sub>4</sub>和R<sub>5</sub>。FT分区B包括路由器R<sub>6</sub>、R<sub>7</sub>、R<sub>8</sub>、R<sub>9</sub>、R<sub>10</sub>、R<sub>11</sub>和R<sub>12</sub>。在FT分区A中,每个节点(路由器R<sub>0</sub>、R<sub>1</sub>、R<sub>2</sub>、R<sub>3</sub>和R<sub>5</sub>)从路由器R<sub>4</sub>接收LSA,指示路由器R<sub>4</sub>到路由器R<sub>7</sub>的链路中断。此外,在FT分区A中,每个节点(路由器R<sub>0</sub>、R<sub>1</sub>、R<sub>2</sub>、R<sub>3</sub>和R<sub>4</sub>)从路由器R<sub>5</sub>接收LSA,指示路由器R<sub>5</sub>到路由器R<sub>8</sub>的链路中断。

[0065] 类似地,在FT分区B中,每个节点(路由器R<sub>6</sub>、R<sub>8</sub>、R<sub>9</sub>、R<sub>10</sub>、R<sub>11</sub>和R<sub>12</sub>)从路由器R<sub>7</sub>接收LSA,指示路由器R<sub>7</sub>到路由器R<sub>4</sub>的链路中断。此外,在FT分区B中,每个节点(路由器R<sub>6</sub>、R<sub>7</sub>、R<sub>9</sub>、R<sub>10</sub>、R<sub>11</sub>和R<sub>12</sub>)从路由器R<sub>8</sub>接收LSA,指示路由器R<sub>8</sub>到路由器R<sub>5</sub>的链路中断。

[0066] 图3为本发明实施例提供的修复的FT的示意图。在所示实施例中,计算链路R<sub>4</sub>—R<sub>7</sub>的唯一备用路径和链路R<sub>5</sub>—R<sub>8</sub>的唯一备用路径,然后启用用于暂时泛洪的备用路径,从而修复拆分FT(如图2所述)。对于链路R<sub>4</sub>—R<sub>7</sub>,路由器R<sub>4</sub>、R<sub>3</sub>、R<sub>6</sub>、R<sub>7</sub>和FT中的其他节点计算备用路径。在一个实施例中,分区A中的路由器R<sub>3</sub>和R<sub>4</sub>计算备用路径,分区B中的路由器R<sub>6</sub>和R<sub>7</sub>计算备用路径。在分区A中,存在两条单向链路:从路由器R<sub>7</sub>到路由器R<sub>4</sub>的链路和从路由器R<sub>8</sub>到路由器R<sub>5</sub>的链路。在备用路径计算中,路由器R<sub>3</sub>和R<sub>4</sub>不使用这两条链路。类似地,在分区B中,存在两条单向链路:从路由器R<sub>4</sub>到路由器R<sub>7</sub>的链路和从路由器R<sub>5</sub>到路由器R<sub>8</sub>的链路。在备用路径计算中,路由器R<sub>6</sub>和R<sub>7</sub>不使用这两条链路。在所示实施例中,链路R<sub>4</sub>—R<sub>7</sub>的备用路径为R<sub>4</sub>—R<sub>3</sub>—R<sub>6</sub>—R<sub>7</sub>。如图3所示,为了创建备用路径R<sub>4</sub>—R<sub>3</sub>—R<sub>6</sub>—R<sub>7</sub>,路由器R<sub>3</sub>添加了链路R<sub>3</sub>—R<sub>6</sub>,路由器R<sub>6</sub>将链路R<sub>6</sub>—R<sub>3</sub>暂时添加到所述FT中。然后,路由器R<sub>4</sub>、R<sub>3</sub>、R<sub>6</sub>和R<sub>7</sub>启用备用路径以进行暂时泛洪。如前述图1所示,链路R<sub>4</sub>—R<sub>3</sub>已经在FT中。因此,路由器R<sub>4</sub>和R<sub>3</sub>不会再次将链路R<sub>4</sub>—R<sub>3</sub>添加到FT中。类似地,由于链路R<sub>6</sub>—R<sub>7</sub>已经在FT中,路由器R<sub>6</sub>和R<sub>7</sub>不会再将这条链路添加到FT中。

[0067] 类似地,计算并启用链路R<sub>5</sub>—R<sub>8</sub>的唯一备用路径,以进行暂时泛洪。对于链路R<sub>5</sub>—R<sub>8</sub>,路由器R<sub>5</sub>、R<sub>2</sub>、R<sub>9</sub>、R<sub>8</sub>和FT中的其他节点计算备用路径。在一个实施例中,分区A中的路由器R<sub>2</sub>和R<sub>5</sub>计算备用路径,分区B中的路由器R<sub>8</sub>和R<sub>9</sub>计算备用路径。如上所述,在分区A中,存

在两条单向链路：从路由器R7到路由器R4的链路和从路由器R8到路由器R5的链路。在备用路径计算中，路由器R2和R5不使用这两条链路。在分区B中，存在两条单向链路：从路由器R4到路由器R7的链路和从路由器R5到路由器R8的链路。在备用路径计算中，路由器R8和R9不使用这两条链路。在所示实施例中，链路R5—R8的备用路径为R5—R2—R9—R8。创建备用路径R5—R2—R9—R8，路由器R5暂时将链路R5—R2添加到FT中，路由器R2暂时将链路R2—R5添加到FT中，路由器R2暂时将链路R2—R9添加到FT中，路由器R9暂时将链路R9—R2添加到FT中，路由器R9暂时将链路R9—R8添加到FT中，路由器R8暂时将链路R8—R9添加到FT中。然后，路由器R5、R2、R9和R8启用备用路径R5—R2—R9—R8以进行暂时泛洪。通过暂时添加到FT中的链路，连接分区A和FT分区B，从而修复FT拆分。

[0068] 图4为本发明实施例提供的并行FT修复过程400的流程图。在一个实施例中，并行FT修复过程400可由网络区域中的每个节点（例如，图1中的网络拓扑100中的路由器R0至R12）并行执行，以检测链路故障和/或节点故障、检测FT拆分以及修复一个或多个链路故障/节点故障。在发生多个故障时，通过并行执行所述过程，所述并行FT修复过程400提供快速FT拆分保护。

[0069] 所述并行FT修复过程400从步骤402开始，检测链路故障或节点故障。在一个实施例中，如果所述节点附接到所述故障链路，则通过链路中断事件在本地检测所述故障。否则，将LSDB中对应的链路状态信息与从FT中其他节点接收的在LSA中的链路状态信息进行比较，确定故障链路。所述故障链路指的是LSDB中链路状态信息中的链路，而不是LSA中接收到的链路状态信息中的链路。如果一个节点检测到附接到所述节点的每个链路发生故障，则所述节点本身发生了故障。

[0070] 在步骤404中，所述并行FT修复过程400中，确定在当前FT中的是故障链路还是故障节点。在一个实施例中，如果网络节点的LSDB中存在故障链路的标志指示FT中存在所述链路，则当前FT中存在所述故障链路。在另一个实施例中，如果故障链路处于FT的LSDB中的链路状态，则当前FT中存在故障链路。对于故障节点，在一个实施例中，所述并行FT修复过程400通过确定当前FT中是否存在连接故障节点的链路来确定在当前FT中是否存在故障节点。在一个实施例中，如果LSDB中有一条附接到故障节点的故障链路的标志指示当前FT中存在所述链路，则在当前FT中存在故障节点。在另一个实施例中，如果故障节点处于当前FT的LSDB中的链路状态，则在当前FT中存在故障节点。

[0071] 如果在步骤404中，所述并行FT修复进程400确定在当前FT中不存在故障链路或故障节点，则所述并行FT修复进程400在步骤406中不进行进一步操作，并终止。但是，如果所述并行FT修复过程400确定在当前FT中是否存在故障链路或故障节点，则所述并行FT修复过程400在步骤408确定故障节点或故障链路是否导致FT拆分。如图2中给定示例所示，当FT被拆分为两个或两个以上独立的节点组时，FT被拆分。在一个实施例中，对于当前FT中节点A和节点B之间的故障链路，所述并行FT修复过程400可以通过尝试在当前FT中查找从节点A到节点B（或从节点B到节点A）的路径来确定当前FT是否被拆分。如果在当前FT中找到路径，则所述并行FT修复过程400确定当前FT未被拆分。否则（即，当前FT中没有从节点A到节点B的任何路径），所述并行FT修复过程400确定当前FT被拆分。

[0072] 在另一个实施例中，对于在当前FT中节点A和节点B之间的故障链路，通过遍历当前FT确定所述FT中节点的第一数量，维护网络节点的LSDB中第二数量个活跃（或活动）节

点,所述并行FT修复过程400可以确定当前FT是否被拆分。如果所述第一数量小于所述第二数量,则所述并行FT修复过程400确定所述当前FT被拆分。如果所述第一数量大于所述第二数量,则网络或节点X内部存在问题,并发出或记录警告。

[0073] 在一个实施例中,对于当前FT中的故障节点,对于通过当前FT中的链路连接到故障节点的所有节点,从通过当前FT中链路连接到故障节点一个节点,到通过当前FT中链路连接到故障节点的每个其他节点,试图在当前FT中查找路径,所述并行FT修复过程400确定当前FT是否被拆分。如果在当前FT中查找到路径,则所述并行FT修复过程400确定由于故障节点当前FT未被拆分。否则(即,从通过当前FT中链路连接到故障节点一个节点,到通过当前FT中链路连接到故障节点的每个其他节点,不存在路径),所述并行FT修复过程400确定由于故障节点当前FT被拆分。

[0074] 在另一个实施例中,所述并行FT修复过程400通过遍历当前FT来确定当前FT中节点的第一数量,并维护在网络节点的LSDB中第二数量个活跃节点或活动节点。如果所述第一数量小于所述第二数量,则所述并行FT修复过程400确定由于故障节点当前FT被拆分。如果所述第一数量大于所述第二数量,则网络或节点X内部存在问题,并发出或记录警告。

[0075] 如果所述并行FT修复过程400确定故障节点或故障链路不会造成当前FT被拆分,则在步骤406中,所述并行FT修复进程400不进行进一步操作,并终止。

[0076] 在一个实施例中,如果所述并行FT修复进程400确定故障节点或故障链路导致FT拆分,则在步骤412,所述并行FT修复进程400确定正在执行所述并行FT修复过程400的节点(节点X)与故障节点或故障链路(即故障点)是否离得太远。在一个实施例中,与故障点的距离可以是用户指定的(例如,与故障点相距5跳,或者与故障点相距其他预定数量个跳)。在一个实施例中,当节点X离故障点太远时,在步骤406中,所述并行FT修复过程400不采取动作,并终止。如果所述并行FT修复过程400确定所述节点X离所述故障点不太远,则在步骤414,所述并行FT修复过程400修复FT(例如,如图5中所述)。在一个替换实施例中,所述并行FT修复过程400可以跳过步骤412,在步骤414中不论节点X与故障点的距离是多少修复FT。

[0077] 图5为本发明实施例提供的FT修复过程500的流程图。在一个实施例中,网络区域(例如,图1中网络拓扑100中的路由器R0至R12)中的每个节点(节点X)并行执行所述FT修复过程500。在一个实施例中,在图4的步骤414中执行FT修复过程500。

[0078] 在步骤502中,所述FT修复过程500计算当前FT中每个故障链路的备用路径。例如,为了修复当前FT中节点A和节点B之间的单链路故障,在步骤502,所述FT修复过程500计算节点A和节点B之间的备用路径。在一个实施例中,所述FT修复过程500计算从低ID节点到高ID节点的备用路径。例如,当节点A的ID小于节点B的ID时,所述FT修复过程500计算从节点A到节点B的备用路径。

[0079] 在一个实施例中,如果当前FT中两条或两条以上相邻链路发生故障,则在步骤502,所述FT修复过程500计算当前FT上相邻故障链路的两个端节点之间的备用路径。当当前FT中两条链路共享一个公共节点且当前FT中没有附接到这个公共节点的其他链路时,将这两条链路视为相邻链路。当前FT中相邻两条链路的两个端节点之间存在一条路径(需要说明的是,公共节点是路径的中转节点)。类似地,在当前FT中两条以上相邻链路的两个端节点之间存在一条路径。

[0080] 在一个实施例中,如果当前FT中存在节点故障或当前FT中两个相邻节点发生故

障,则在步骤502中,所述FT修复过程500计算备用路径,从通过当前FT中链路连接到故障节点一个节点,到通过当前FT中链路连接到故障节点的每个其他节点。当当前FT的两个节点直接连接时,则这两个节点被认为是当前FT中的相邻节点。

[0081] 在步骤504中,所述FT修复进程500确定备用路径是否通过节点X(即,执行所述FT修复进程500的节点)。如果所述FT修复进程500确定所述备用路径不通过节点X,则在步骤506中,所述FT修复进程500不采取进一步的操作并终止。

[0082] 如果在步骤504中,所述FT修复进程500确定所述备用路径通过节点X,则在步骤508中,所述FT修复进程500检索所述备用路径上节点X的本地链路(例如,链路L1和L2),并将所述备用路径上节点X的本地链路暂时添加到当前FT中,直到如果当前FT中不存在备用路径上节点X的本地链路构建新的FT。如果当前FT中已经存在备用路径上节点X的本地链路,则不需要将这些本地链路添加到当前FT中,并且所述FT修复过程500终止。由网络区域中每个节点并行执行所述FT修复过程500,修复FT拆分。

[0083] 图6为本发明实施例提供的顺序FT修复过程600的流程图。在一个实施例中,所述顺序FT修复过程600由附接到故障点的节点执行(即,由故障旁边的节点检测故障)。如将进一步描述的,附接到故障点的节点首先检测故障(即故障链路或故障节点),确定FT被拆分,然后修复故障本身,之后请求相关节点来修复故障。这个顺序过程比并行过程(例如,如图4中所述)要慢,并且需要更多的时间进行网络融合。

[0084] 所述顺序FT修复过程600从步骤602开始,附接到链路的节点首先检测链路故障。在一个实施例中,通过链路中断事件在本地检测故障。在步骤604中,所述顺序FT修复过程600确定当前FT中是否存在故障链路。在一个实施例中,如果LSDB中存在故障链路的标志指示FT中存在链路,则当前FT中存在故障链路。在另一个实施例中,如果故障链路处于FT的LSDB中的链路状态,则当前FT中存在所述故障链路。

[0085] 在步骤606中,所述顺序FT修复过程600确定当前FT中是否存在节点故障。在一个实施例中,为了检测节点故障,所述顺序FT修复过程600在步骤606中检测附接到故障链路的另一端节点(节点X)的当前FT中每个其他链路。对于每个其他链路,通过将节点的LSDB中对应的链路状态与接收到的链路状态进行比较,节点确定是否为故障链路。故障链路是处于LSDB中链路状态而不是刚接收到的链路状态的链路。如果附接到节点X的当前FT中所有链路均发生故障,则所述顺序FT修复过程600确定当前FT中节点X发生故障。

[0086] 在步骤608中,所述顺序FT修复过程600确定当前FT是否被拆分。对于当前FT中节点A和节点B之间的故障链路,所述顺序FT修复过程600可以通过尝试在当前FT中查找从节点A到节点B的路径来确定当前FT是否被拆分。如果在当前FT中找到路径,则当前FT未被拆分。否则(即当前FT中没有从节点A到节点B的任何路径),当前FT被拆分。或者,通过遍历当前FT检索所述FT中节点的第一数量,维护LSDB中第二数量个活跃节点或活动节点,所述顺序FT修复过程600可以确定当前FT是否被拆分。如果所述节点的第一数量小于LSDB中所述活跃节点或活动节点的第二数量,则当前FT被拆分。在一个实施例中,如果所述节点的第一数量大于LSDB中所述活跃节点或活动节点的第二数量,则网络中存在问题或者故障节点内存在问题,并发出或记录警告。如果在步骤608中,所述顺序FT修复进程600确定当前FT未被拆分,则所述顺序FT修复进程600在步骤612中不进行进一步操作,并终止。但是,如果在步骤608中,所述顺序FT修复进程600确定当前FT被拆分,则所述顺序FT修复进程600在步骤

610中修复FT(例如,如图7所述)。

[0087] 图7为本发明实施例提供的FT修复过程700的流程图。在一个实施例中,所述FT修复过程700首先由故障旁边的节点执行,然后由每个下一跳节点沿着备用路径执行。所述FT修复过程700从步骤702开始,计算备用路径。如果所述FT修复过程700由故障旁边的节点执行,所述FT修复过程700计算从当前执行节点(节点A)到通过当前FT中链路连接到故障点(节点X)的每个其他节点的备用路径。

[0088] 如果当前FT中不存在本地链路,则所述FT修复过程700在步骤704中将所述备用路径上执行节点的本地链路(例如,链路L1)暂时添加到当前FT中,直到构建成新的FT。在步骤706中,所述FT修复过程700向沿着备用路径的下一跳节点发送扩展Hello报文(例如,图8)中的请求(例如,向链路L1的远端节点发送Hello报文)来请求相关节点修复FT。

[0089] 在接收到所述扩展Hello报文后,所述远程终端节点重复所述FT修复过程700,在步骤702中计算从节点A(即,当前节点)到节点X的备用路径;如果当前FT中不存在本地链路,在步骤704中将所述备用路径上所述节点的所述本地链路(例如,链路L1和L2)暂时添加到当前FT中,直到构建成新的FT;在步骤706中,向沿着所述备用路径的下一跳节点发送扩展Hello报文中的请求(例如,向链路L2的远端节点发送Hello报文)。当目的节点将备用路径上的本地链路添加到当前FT中时,所述FT修复过程700终止。

[0090] 图8为本发明实施例提供的扩展选项和标志(Extended Options and Flags,简称EOF)类型长度值(Type-Length-Value,简称TLV)800的数据结构示意图。所述EOF-TLV 800可以包含在OSPF Hello报文中。例如,在一个实施例中,所述EOF-TLV 800包括在用于发现和维持邻居关系的OSPF Hello报文的本地链路信令(link-local signaling,简称LLS)数据块中。在所示实施例中,所述EOF-TLV 800包括类型字段802(第一类为请求注解(Request For Comments,简称RFC)5613中定义的EOF-TLV)、包含所述EOF-TLV 800的长度的长度字段804以及扩展选项和标志字段806。在一个实施例中,在EOF-TLV 800中扩展选项和标志字段806中定义了称为创建备用路径(BP比特808)的新标志位。在一个实施例中,扩展选项和标志字段806中的比特29用于BP比特808。

[0091] 如图7所示,当节点B通过链路从其相邻节点A接收包含EOF-TLV 800的Hello报文时,如果BP比特808被设置为EOF-TLV 800中的1(1),则指示备用路径上的链路由节点A暂时添加到当前FT中。然后,如果链路不在Hello报文中,节点B还将把链路添加到当前FT中,直到构建成新的FT,计算备用路径,并将Hello报文与关于所述路径的一些信息一起发送到所述路径的下一跳。

[0092] 图9为本发明实施例提供的OSPF备用路径端点TLV 900的数据结构示意图。所述OSPF备用路径端点TLV 900可以携带在OSPF Hello报文的本地链路信令(link-local signaling,简称LLS)数据块中。所述OSPF备用路径端点TLV 900包括类型字段902(类型待定(type to be determined,简称TBD))、长度字段904、源节点标识(identifier,简称ID)字段906、目的节点ID字段908和可选的子TLV字段910。所述源节点ID字段906和所述目的节点ID字段908指定备用路径的两个端节点。所述子TLV 910可被定义为详细包含备用路径。例如,所述子TLV 910可以包含备用路径的每一跳。

[0093] 图10为本发明实施例提供的扩展选项和标志(Extended Options and Flags,简称EOF)TLV 1000的数据结构示意图。在一个实施例中,所述EOF-TLV 1000包括在用于发现



和维护邻居关系的IS-IS Hello报文中。所述EOF-TLV 1000包括类型字段1002 (type to be determined, 简称TBD)、长度字段1004以及扩展选项和标志字段1006。在一个实施例中,在所述EFO-TLV 1000中所述扩展选项和标志字段1006中定义了称为创建备用路径 (BP比特1008) 的新标志位。在一个实施例中,在EOF-TLV TLV 1000中所述扩展选项和标志字段1006中的比特29用于BP比特1008。

[0094] 在一个实施例中,当节点B通过链路从其相邻节点A接收包含EOF-TLV 1000的Hello报文时,如果BP比特1008被设置为Hello报文中的1 (1),则指示备用路径上的链路由节点A暂时添加到当前FT中。然后,如果链路不在Hello报文中,节点B还将把链路暂时添加到当前FT中,直到构建成新的FT,计算备用路径,并将Hello报文与关于所述路径的一些信息一起发送到所述路径的下一跳。

[0095] 图11为本发明实施例提供的IS-IS备用路径端点TLV 1100的数据结构示意图。所述IS-IS备用路径端点TLV 1100可以携带在IS-IS Hello报文中。所述IS-IS备用路径端点TLV 1100包括类型字段1102 (type to be determined, 简称TBD)、长度字段1104、源节点ID字段1106、目的节点ID字段1108和可选的子TLV字段1110。所述源节点ID字段1106和所述目的节点ID字段1108指定备用路径的两个端节点。所述子TLV 1110可被定义为详细包含备用路径。例如,所述子TLV 1110可以包含备用路径的每一跳。

[0096] 图12为本发明实施例提供的网元1200的示意图。所述网元1200可以是任何网络节点,例如但不限于世界各地的服务提供商使用的任何路由器、交换机和控制器。所述网元1200包括用于通过入口端口1210接收数据的接收器单元 (receive, 简称RX) 1220或接收装置。所述网元1200还包括用于通过数据出口端口1250传输的发射器单元 (transmit, 简称TX) 1240或传输装置。

[0097] 所述网元1200包括用于存储指令和各种数据的存储器1260或数据存储装置。所述存储器1260可以是能够存储数据和/或指令的任何类型的存储器组件或存储器组件的组合。例如,所述存储器1260可以包括易失性和/或非易失性存储器,例如只读存储器 (read-only memory, 简称ROM)、随机存取存储器 (random access memory, 简称RAM)、三重内容寻址内存 (ternary content-addressable memory, 简称TCAM) 和/或静态随机存取存储器 (static random-access memory, 简称SRAM)。所述存储器1260还可以包括一个或多个磁盘、磁带驱动器和固态硬盘。在一些实施例中,所述存储器1260可用作溢出数据存储设备,用于在选择程序执行时存储此类程序,并且用于存储程序执行过程中读出的指令和数据。

[0098] 所述网元1200具有一个或多个处理器1230或其他处理装置 (例如,中央处理器 (central processing unit, 简称CPU)) 来处理指令。所述处理器1230可作为一个或更多CPU芯片、核心 (例如,多核处理器)、现场可编程门阵列 (field-programmable gate array, 简称FPGA)、专用集成电路 (application-specific integrated circuit, 简称ASIC) 和数字信号处理器 (digital signal processor, 简称DSP) 实现。所述处理器1230通过系统总线与入口端口1210、RX 1220、TX 1240、出口端口1250和存储器1260通信耦合。所述处理器1230可用于执行所述存储器1260中存储的指令。因此,当所述处理器执行适当指令时,所述处理器1230提供执行与权利要求对应的计算、比较、确定、发起、配置或任何其他操作的装置。在一些实施例中,所述存储器1260可以是与处理器1230集成的存储器。

[0099] 在一个实施例中,所述存储器1260存储快速FT保护模块1270。所述快速FT保护模

块1270包括用于实现所公开实施例的数据和可执行指令。例如,所述快速FT保护模块1270可以包括用于实现图4至7中描述方法的指令。所述快速FT保护模块1270所包括的内容大大改善了所述网元1200的功能,实现了FT的快速修复,除此之外,在FT被拆分时降低了流量损失。

[0100] 虽然本发明多个具体实施例,但应当理解,所公开的系统和方法也可通过其它多种具体形式体现,而不会脱离本发明的精神或范围。本发明的实例应被视为说明性而非限制性的,且本公开并不限于本文中所给出的细节。例如,各种元件或部件可以在另一系统中组合或合并,或者某些特征可以省略或不实施。

[0101] 此外,在不脱离本发明的范围的情况下,各种实施例中描述和说明为离散或单独的技术、系统、子系统和方法可以与其它系统、模块、技术或方法进行组合或集成。展示或论述为彼此耦合或直接耦合或通信的其它项也可以采用电方式、机械方式或其它方式经由某一接口、设备或中间组件间接地耦合或通信。其它变更、替换、更替示例对本领域技术人员而言是显而易见的,均不脱离此处公开的精神和范围。

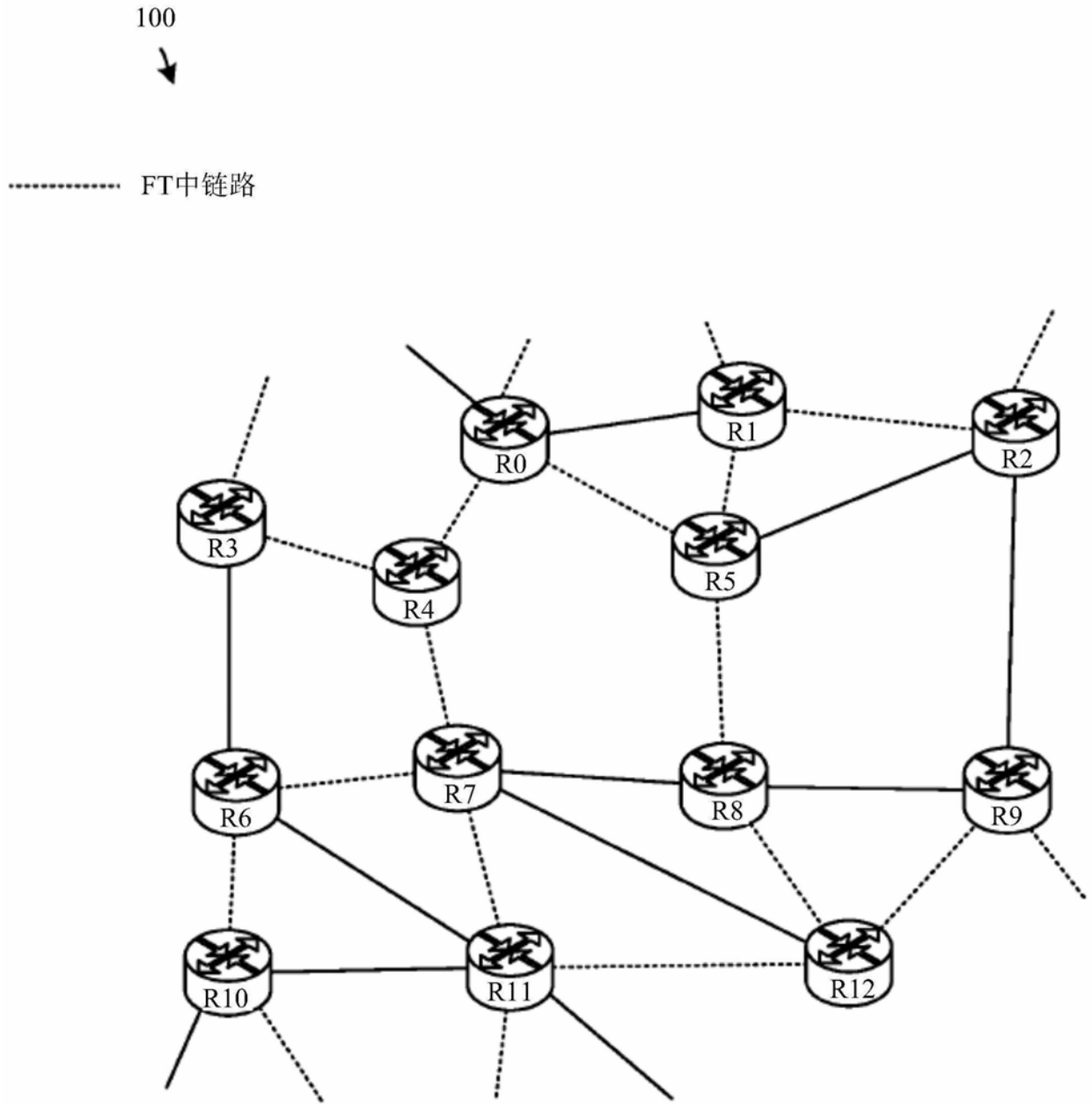


图1

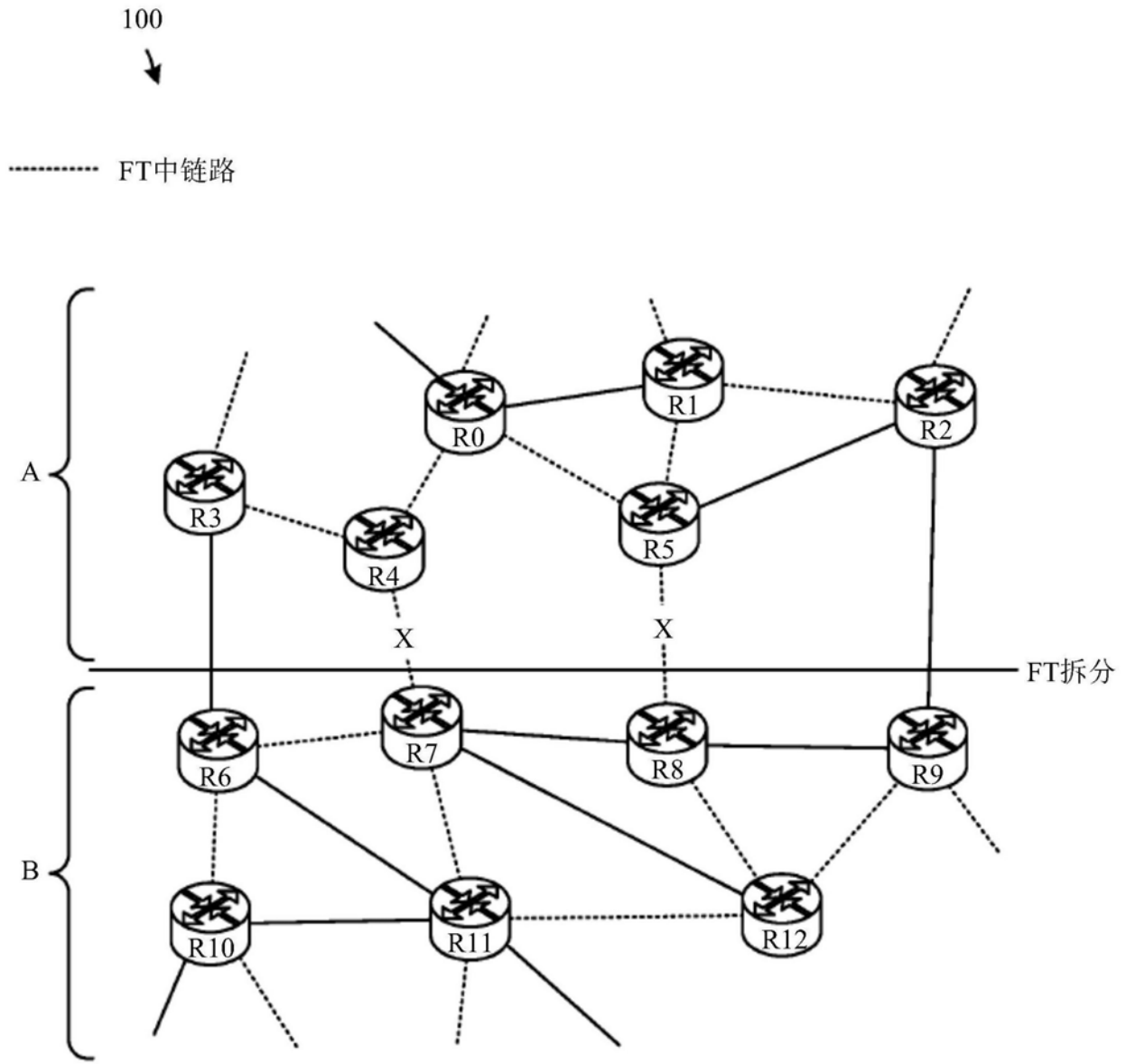


图2

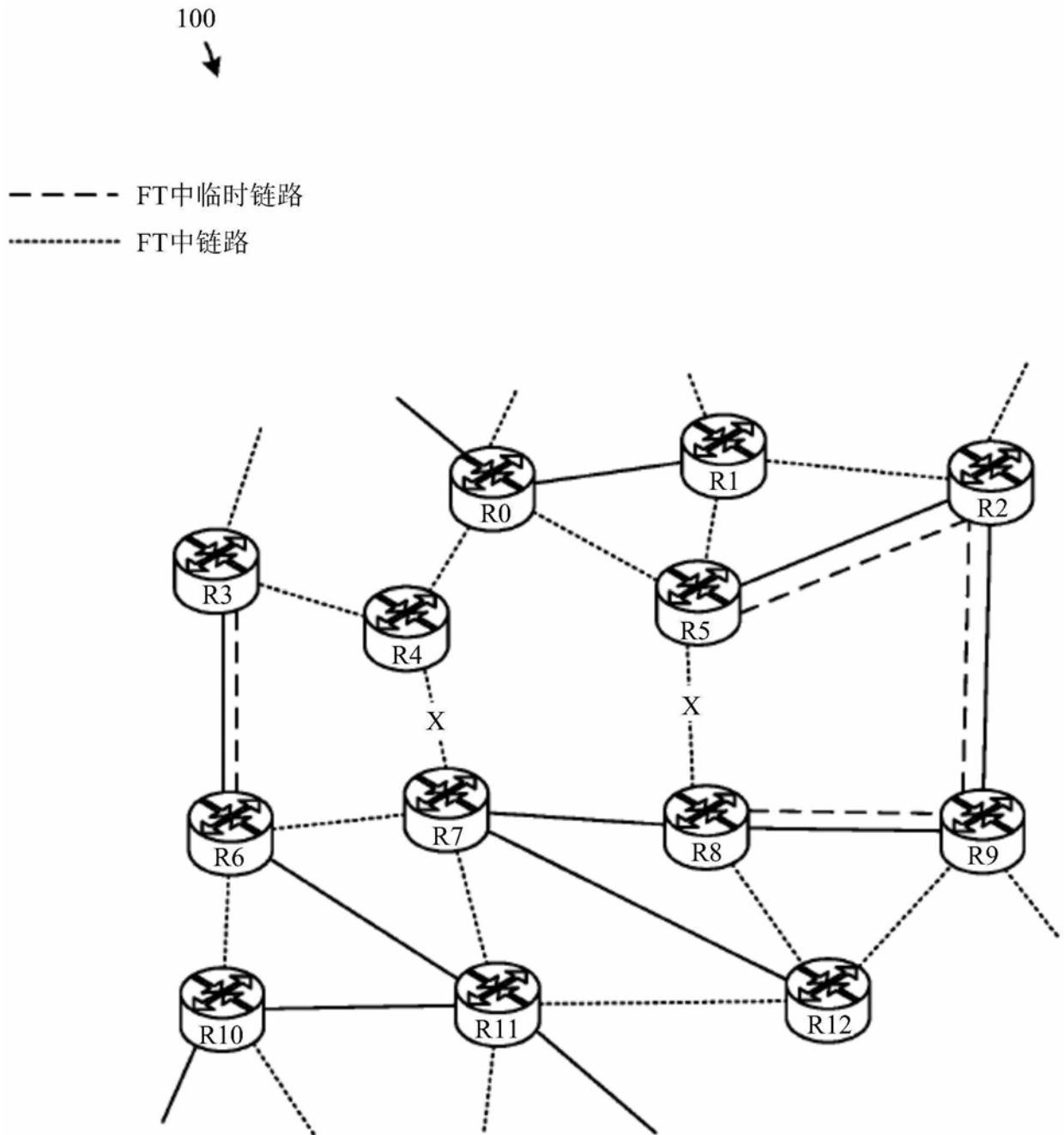


图3

400  
↓

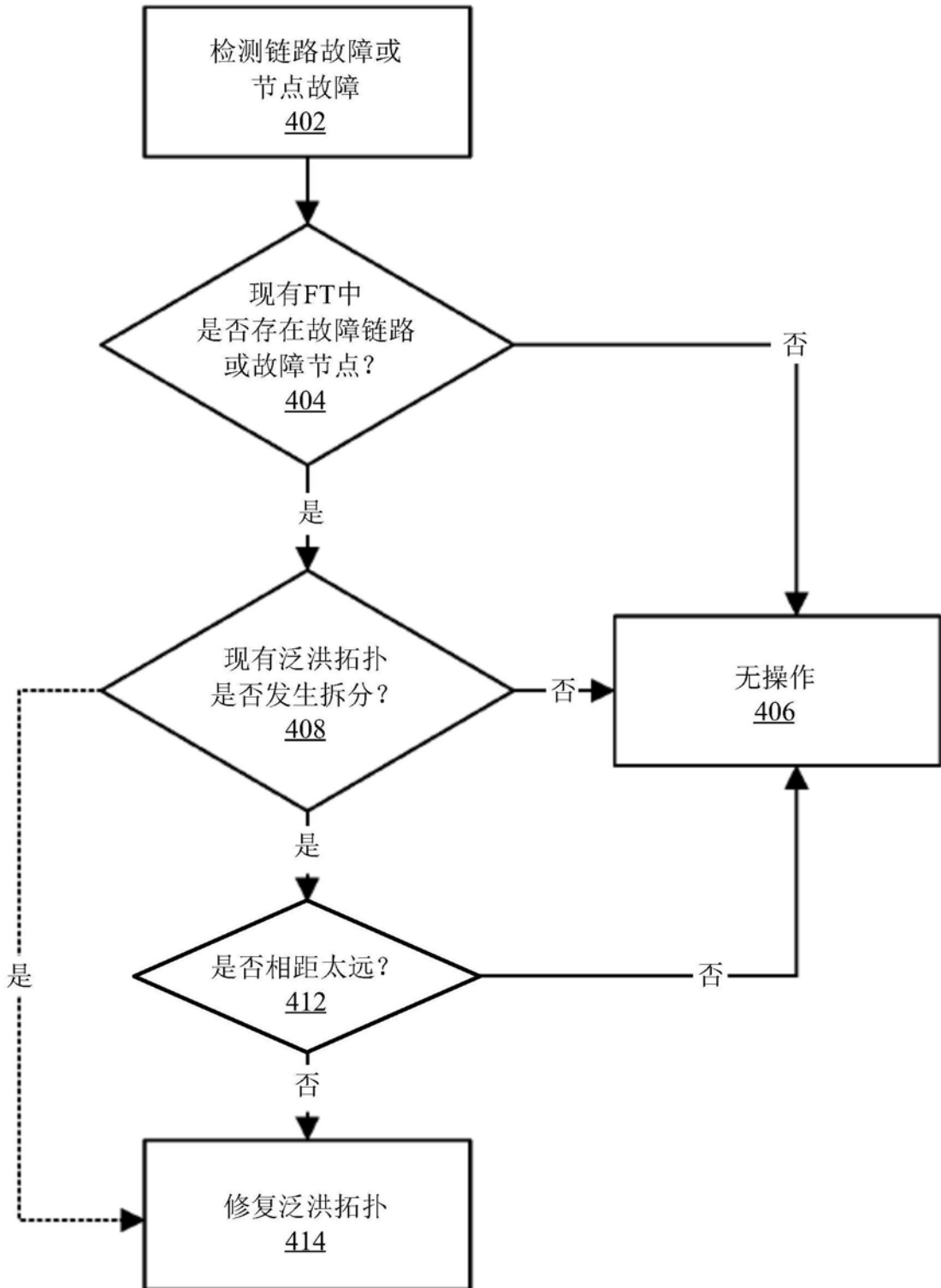


图4

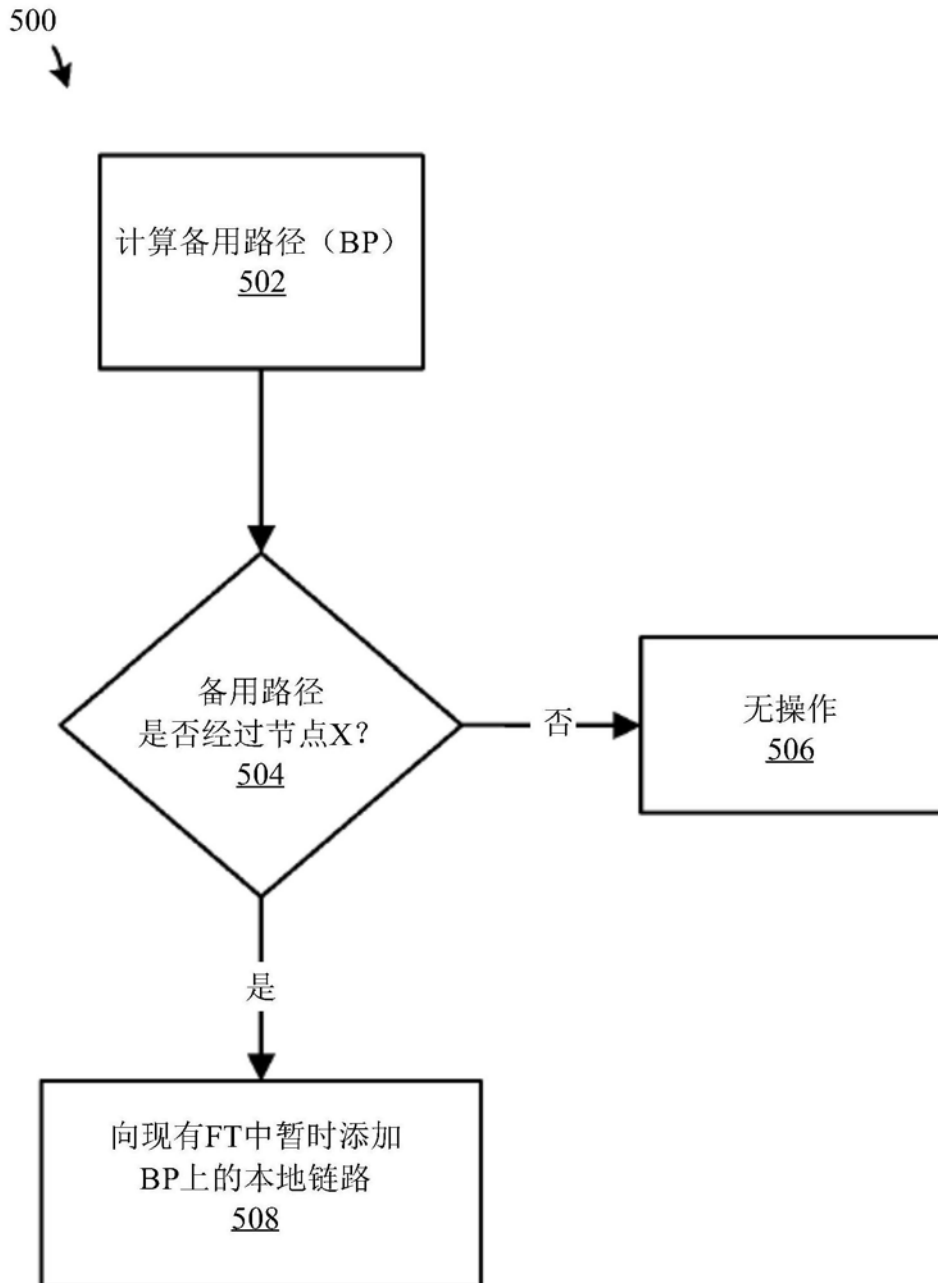


图5

600

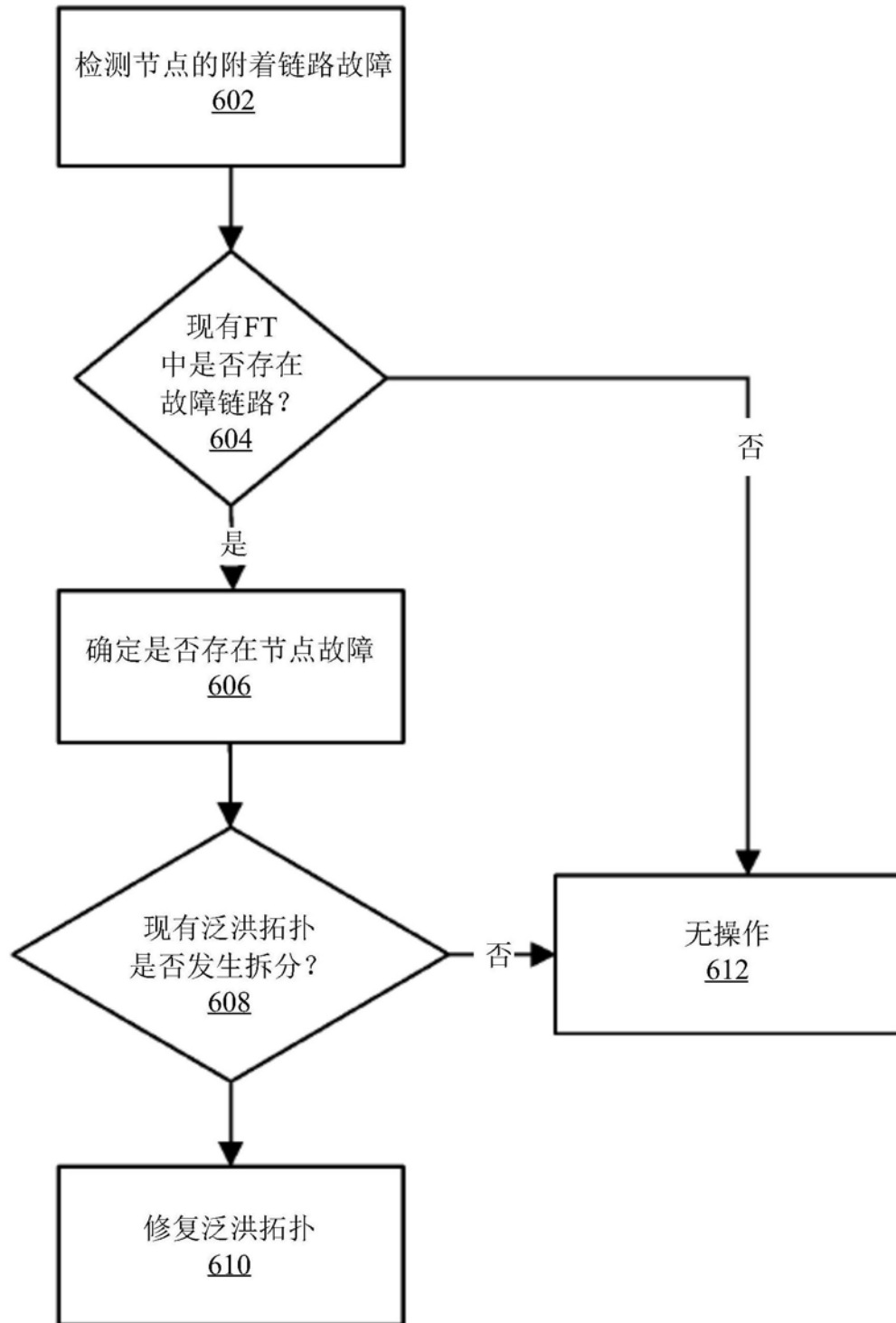


图6



700  
↓

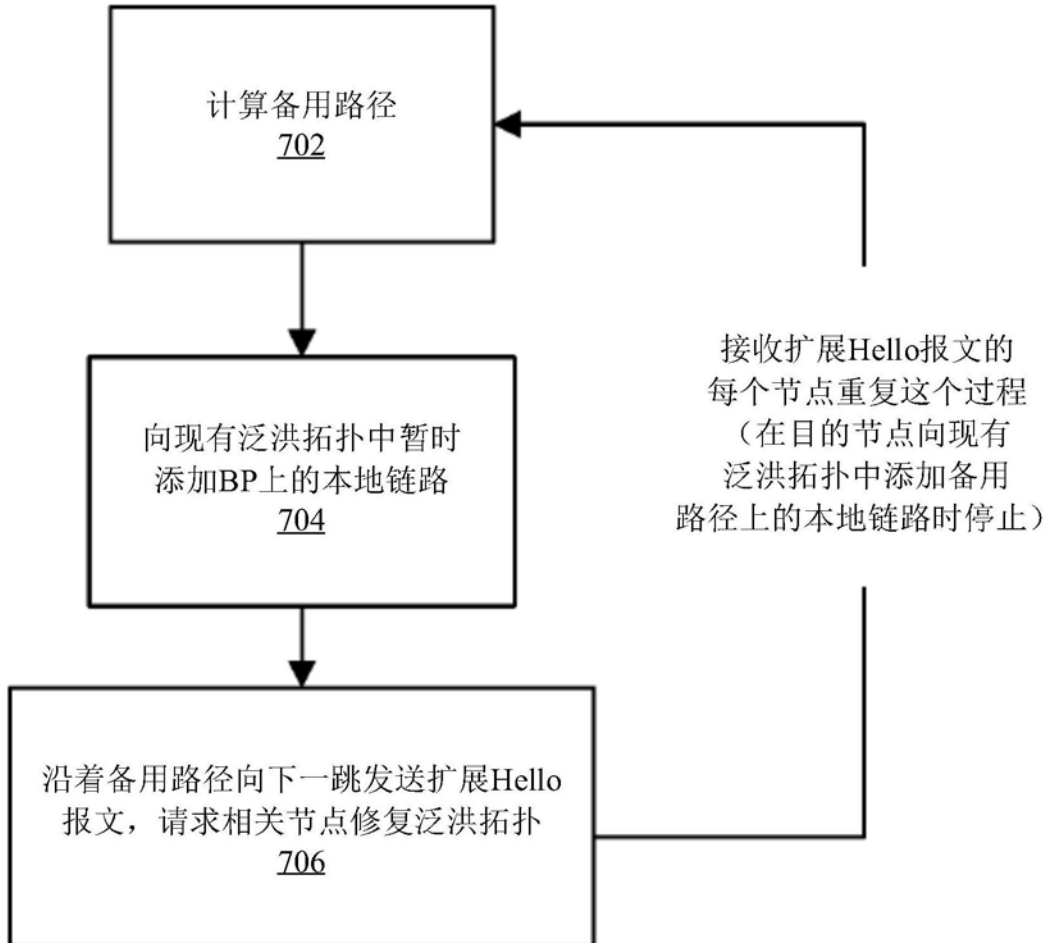


图7

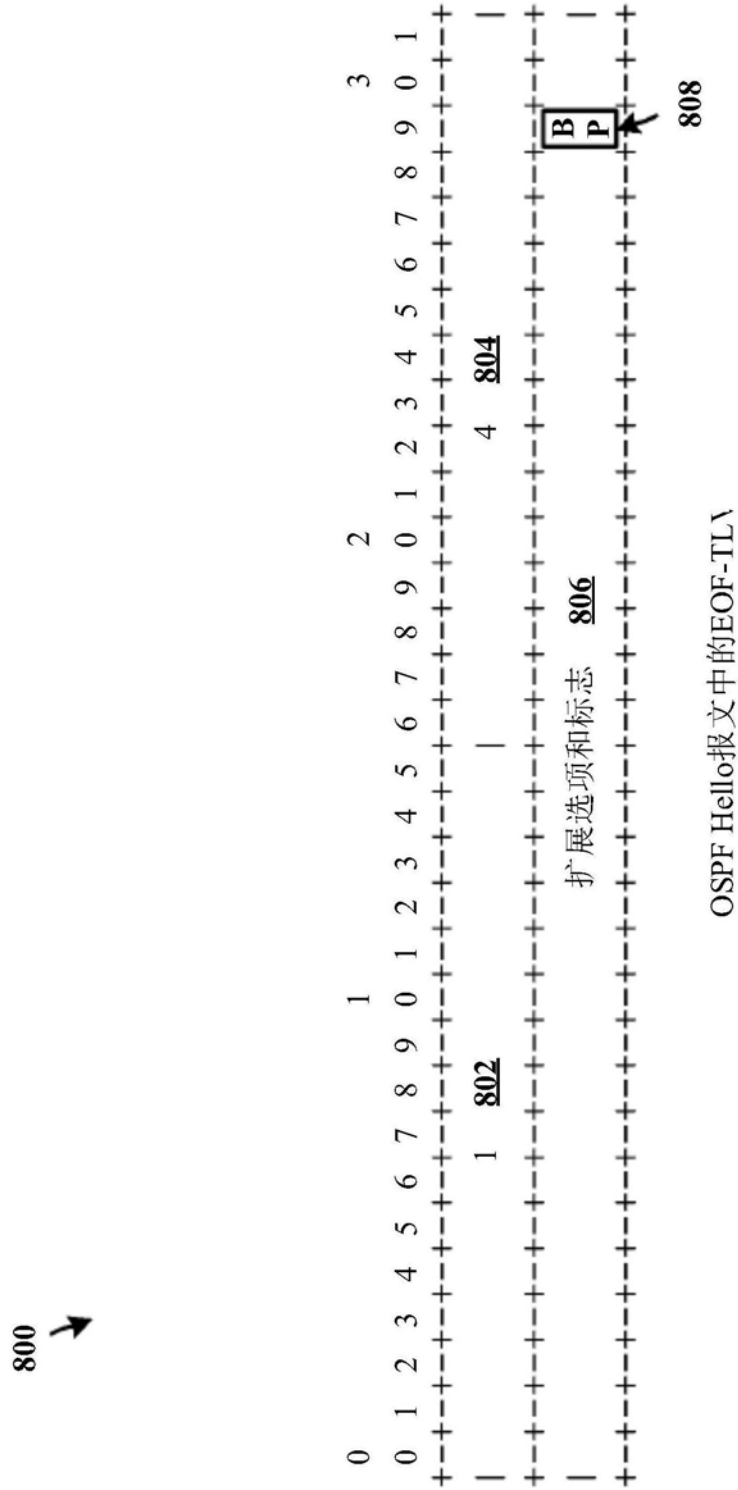


图8

900 →

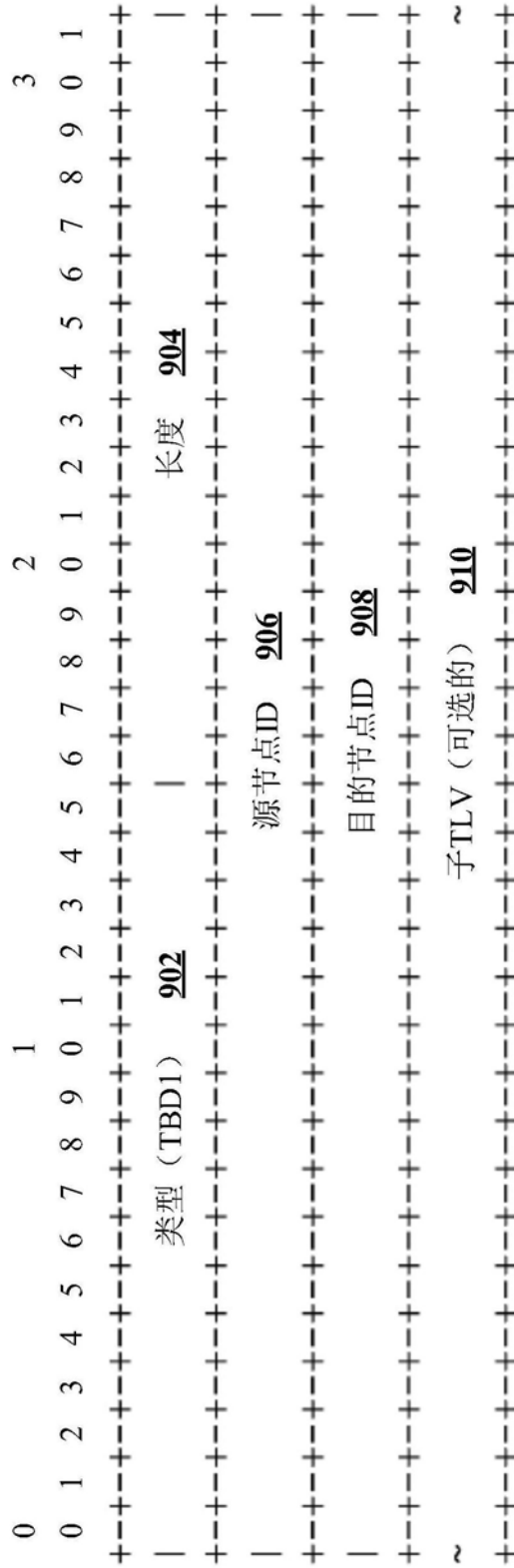


图9

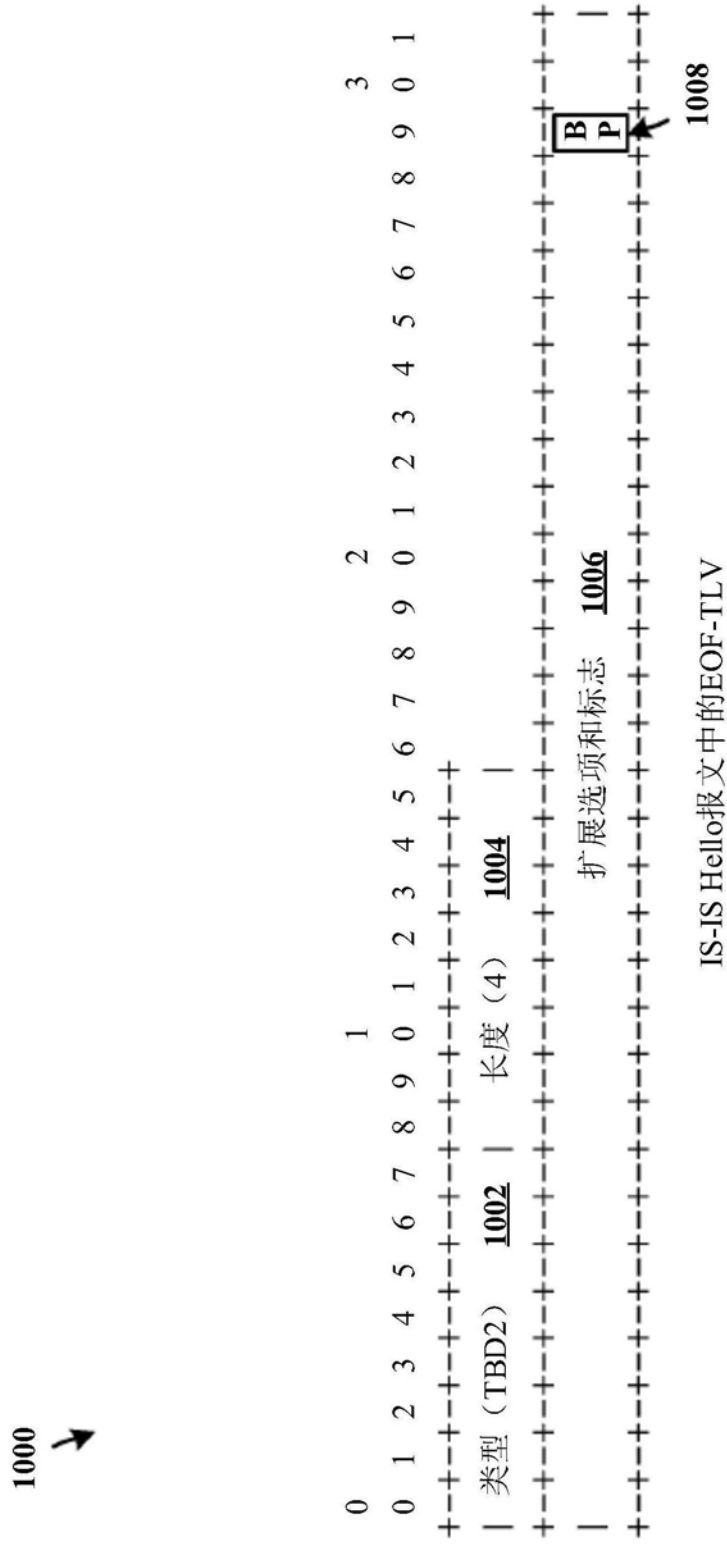


图10

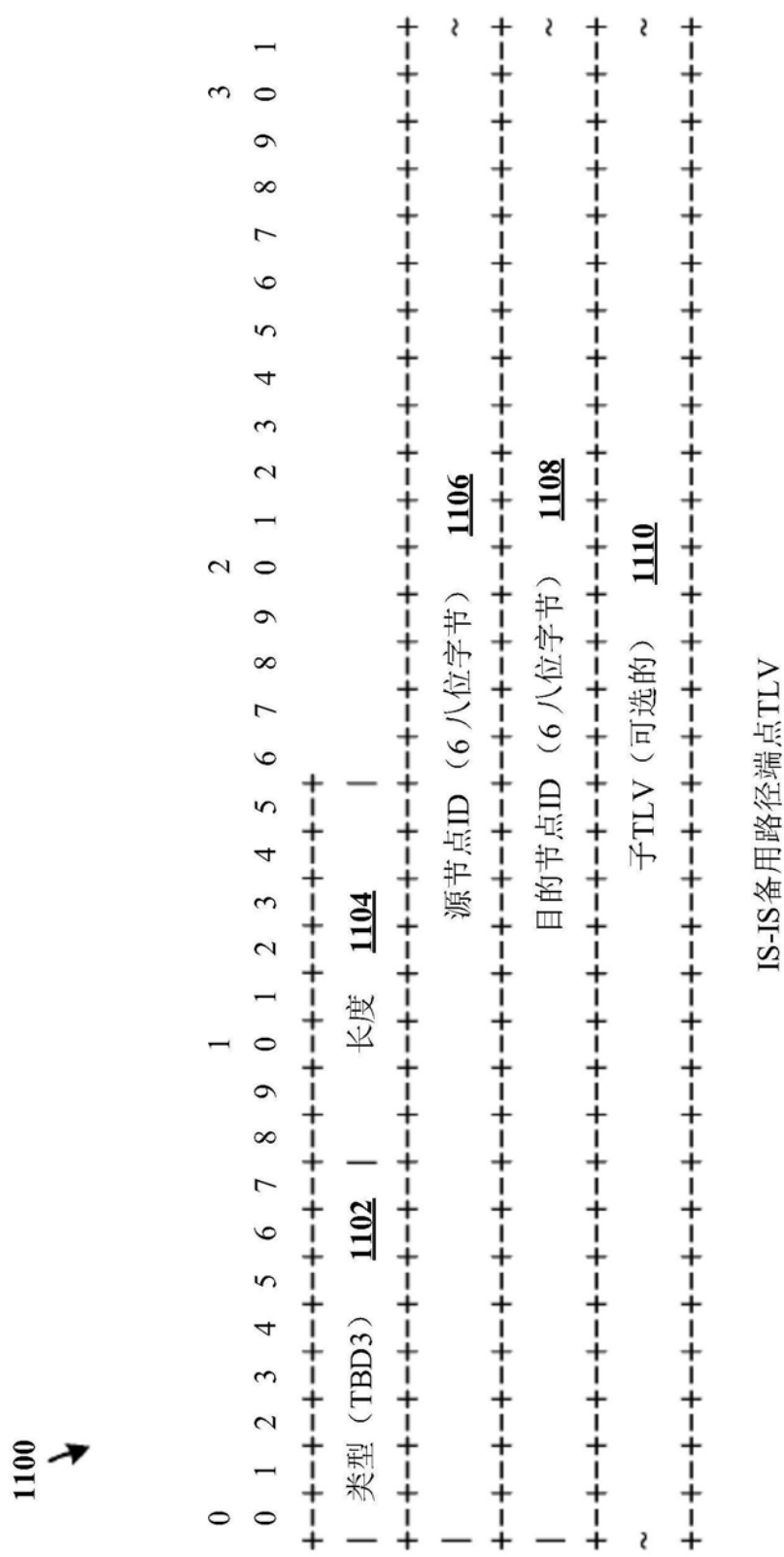


图11

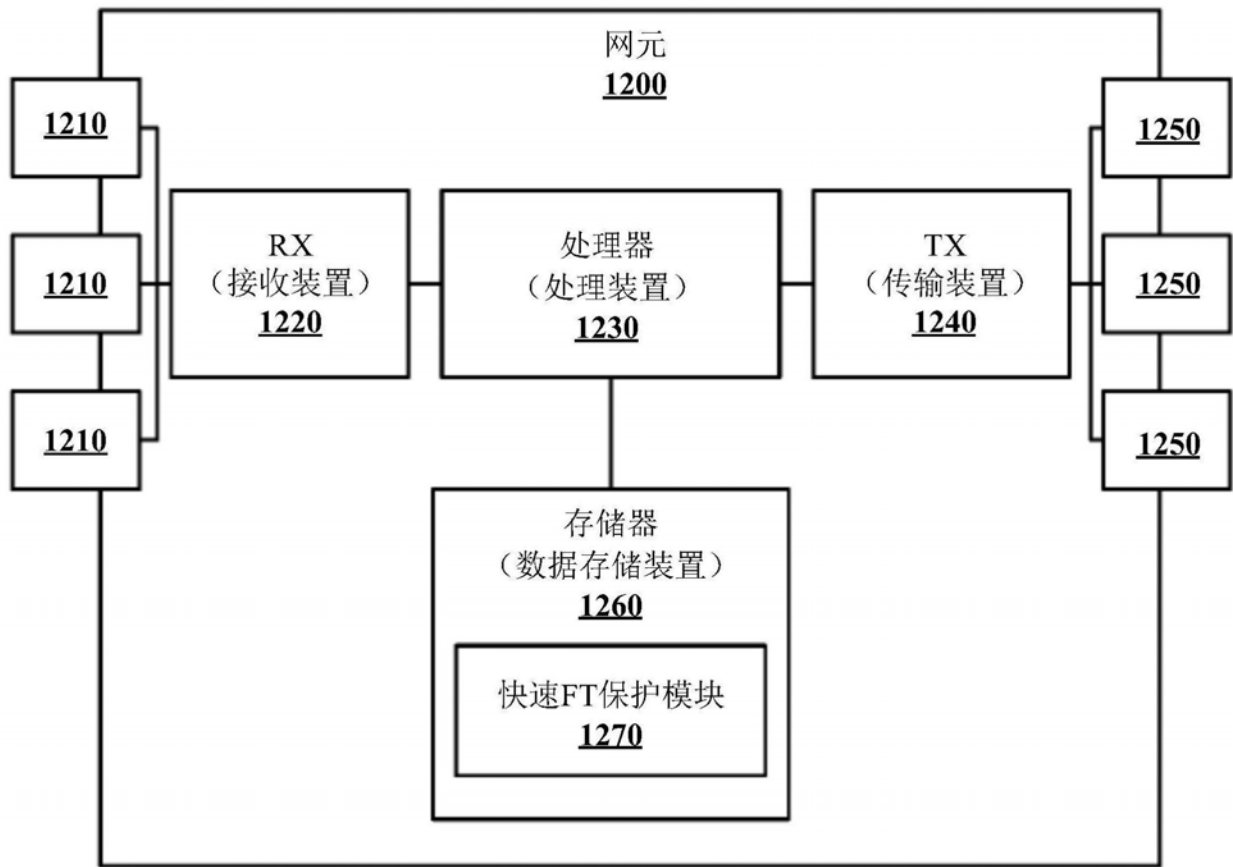


图12