



(12) 发明专利申请

(10) 申请公布号 CN 114073050 A

(43) 申请公布日 2022. 02. 18

(21) 申请号 202080047511.2

(51) Int. Cl.

(22) 申请日 2020.07.02

H04L 45/02 (2022.01)

H04L 45/16 (2022.01)

(30) 优先权数据

62/870,893 2019.07.05 US

(85) PCT国际申请进入国家阶段日

2021.12.28

(86) PCT国际申请的申请数据

PCT/US2020/040613 2020.07.02

(87) PCT国际申请的公布数据

WO2021/007092 EN 2021.01.14

(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

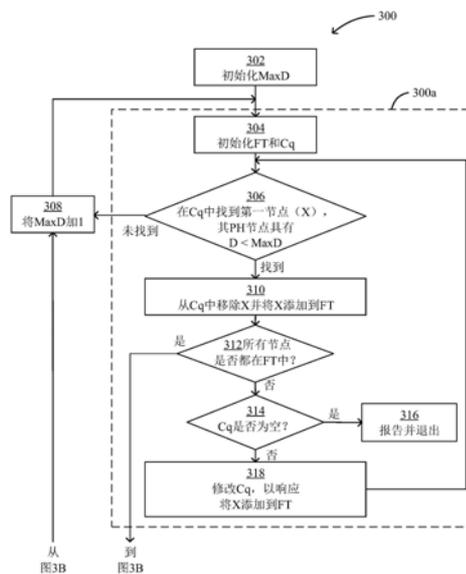
权利要求书10页 说明书21页 附图36页

(54) 发明名称

用于计算泛洪拓扑的系统和方法

(57) 摘要

本发明提出了用于计算网络泛洪拓扑(flooding topology, FT)的方法。所述方法包括用于计算包括所述网络中所有节点的FT的过程和用于确保所述FT中的所有节点在所述FT中都至少具有两条链路的过程。一些方法使所述FT中所述节点的链路数最小化。一些方法还将所述FT中的一些节点限制为最大链路数。一些方法计算所述FT中最大链路数等于所述网络中链路数的节点的第一FT,再计算所述网络中剩余节点的第二FT,然后组合这两个FT以计算所述网络的完整FT。



1. 一种用于计算网络设备中的泛洪拓扑(flooding topology, FT)的方法,其特征在于,所述方法包括:

从网络中选择节点R0;

通过所述节点R0初始化所述FT;

初始化候选队列(candidate queue, Cq);

实现所述方法的第一循环,包括:

从所述Cq中移除第一节点并将所述第一节点附加到所述FT;

确定所述FT是否包括所述网络中的所有节点;

当所述FT包括所述网络中的所有节点时,终止所述第一循环;

当所述FT不包括所述网络中的所有节点时,将在所述网络中耦合到所述第一节点并且不在所述FT中的那些节点附加到所述Cq;

向在所述FT中仅具有一条链路的所述FT中的任何节点添加链路。

2. 根据权利要求1所述的方法,其特征在于,所述方法的第一循环还包括:

当所述FT不包括所述网络中的所有节点时,确定所述Cq是否为空;

当所述Cq为空时:

向所述网络的管理员发送第一消息;

向所述网络设备中的组件发送第二消息,所述第一消息和所述第二消息指示所述网络中存在未连接到所述网络的其它节点的节点;

终止权利要求1所述的方法。

3. 根据权利要求1或2所述的方法,其特征在于,向在所述FT中仅具有一条链路的所述FT中的任何节点添加所述链路包括:

实现所述方法的第二循环,包括:

确定所述FT是否包括在所述FT中仅具有一条链路的节点Q;

当所述FT不包括所述节点Q时,终止所述第二循环,其中,所述FT中的所有节点在所述FT中都具有多条链路;

当所述FT确实包括所述节点Q时:

创建在所述网络中耦合到所述节点Q的所有节点的集合;

选择所述集合中的节点L,所述节点L(i)在所述FT中具有最小链路数,并且

(i) 具有最低nodeID(当两个或多个节点在所述FT中具有相同的最小链路数时);

将所述FT中的所述节点Q的度递增1;

将所述FT中的所述节点L的度递增1;

将Q与L之间的链路添加到所述FT。

4. 根据权利要求1至3中任一项所述的方法,其特征在于,所述节点R0在所述网络中具有最低nodeID。

5. 根据权利要求1至4中任一项所述的方法,其特征在于,通过从最低nodeID到最高nodeID排序的节点来初始化所述Cq。

6. 根据权利要求1至5中任一项所述的方法,其特征在于,附加到所述Cq的节点按从最低nodeID到最高nodeID的顺序排序。

7. 一种用于计算网络设备中的泛洪拓扑(flooding topology, FT)的方法,其特征在

于,所述方法包括:

设置最大度 (maximum degree, MaxD) 的初始值;

实现所述方法的第一循环,包括:

从网络中选择节点R0;

通过所述节点R0初始化所述FT;

将来自所述网络的节点添加到所述FT;

确定所述FT是否包括所述网络中的所有节点;

递增所述MaxD的所述值并重复所述第一循环的步骤,直到所述FT包括所述网络中的所有节点;

当所述FT包括所述网络中的所有节点时,实现所述方法的第二循环,包括:

向在所述FT中仅具有一条链路并且在所述网络中耦合到链路数小于MaxD的节点的所述FT中的任何节点添加链路;

确定所述FT中的所有节点在所述FT中是否都具有多条链路;

递增所述MaxD的所述值并重复所述第一循环和所述第二循环,直到所述FT中的所有节点在所述FT中都具有多条链路。

8. 根据权利要求7所述的方法,其特征在于,将来自所述网络的所述节点添加到所述FT包括:

初始化候选队列 (candidate queue, Cq) 以包括在所述网络中耦合到所述节点R0的所有节点,这些节点在所述Cq中按每个节点的nodeID排序;

实现所述方法的第三循环,包括:

确定所述Cq是否包括节点X,所述节点X是所述Cq中的第一节点,所述Cq在所述节点X的上一跳 (Previous Hop, PH) 中具有节点Y,所述节点Y具有小于MaxD的度;

当所述Cq不包括所述节点X时,终止所述第三循环,其中,所述FT不包括所述网络中的所有节点;

从所述Cq中移除所述节点X,并将所述节点X添加到所述FT;

确定所述FT是否包括所述网络中的所有节点;

当所述FT不包括所述网络中的所有节点时,修改所述Cq;

当所述FT包括所述网络中的所有节点时,终止所述第三循环。

9. 根据权利要求7或8所述的方法,其特征在于,当所述FT不包括所述网络中的所有节点时修改所述Cq包括:

创建在所述网络中耦合到所述节点X且不在所述FT中的所有节点的队列,所述队列按所述队列中的所述节点的nodeID排序;

对于所述队列中的每个节点Z:

确定所述节点Z是否在所述Cq中;

当所述节点Z在所述Cq中时,将所述节点X添加到所述Cq中的所述节点Z的所述PH;

当所述节点Z不在所述Cq中时,将所述节点Z附加到所述Cq。

10. 根据权利要求7至9中任一项所述的方法,其特征在于,向在所述FT中仅具有一条链路并且在所述网络中耦合到链路数小于MaxD的所述节点的所述FT中的任何节点添加所述链路包括:

实现所述方法的第四循环,包括:

确定所述FT是否包括在所述FT中仅具有一条链路的节点Q;

当所述FT不包括所述节点Q时,终止所述第四循环,其中,所述FT中的所有节点在所述FT中都具有多条链路;

当所述FT确实包括所述节点Q时,确定所述FT是否包括在所述网络中耦合到Q的节点L,所述节点L (i) 在所述FT中具有最小链路数, (ii) 具有最低nodeID(当两个或多个节点在所述FT中具有相同的最小链路数时), 并且 (iii) 具有小于MaxD的度;

当所述FT确实包括所述节点L时:

将所述FT中的所述节点Q的度递增1;

将所述FT中的所述节点L的度递增1;

将所述节点L添加到所述FT中的所述节点Q的所述PH;当所述FT不包括所述节点L时,终止所述第四循环,其中,所述FT中的所有节点在所述FT中都不具有多条链路。

11. 根据权利要求7至10中任一项所述的方法,其特征在于,所述节点R0在所述网络中具有最低nodeID。

12. 根据权利要求7至11中任一项所述的方法,其特征在于,通过从最低nodeID到最高nodeID排序的节点初始化所述Cq。

13. 根据权利要求7至12中任一项所述的方法,其特征在于,附加到所述Cq的节点按从最低nodeID到最高nodeID的顺序排序。

14. 一种用于计算网络设备中的泛洪拓扑(flooding topology, FT)的方法,其特征在于,所述方法包括:

设置最大度(maximum degree, MaxD)的初始值,所述初始值等于先前计算的所述网络FT的MaxD的存储值;

实现所述方法的第一循环,包括:

从所述网络中选择节点R0;

通过所述节点R0初始化所述FT;

将来自所述网络的节点添加到所述FT;

确定所述FT是否包括所述网络中的所有节点;

当所述FT包括所述网络中的所有节点时,向在所述FT中仅具有一条链路并且在所述网络中耦合到链路数小于MaxD的节点的所述FT中的任何节点添加链路;

确定 (i) 所述FT是否包括所述网络中的所有节点,以及 (ii) 所述FT中的所有节点在所述FT中是否都具有多条链路;

当 (i) 所述FT包括所述网络中的所有节点且 (ii) 所述FT中的所有节点在所述FT中都具有多条链路时,将MaxD的所述值存储为成功的MaxD并且将所述FT存储为成功的FT;

递减所述MaxD的所述值并重复所述第一循环,直到 (i) 所述FT不包括所述网络中的所有节点或者 (ii) 并非所述FT中的所有节点在所述FT中都具有多条链路;

将所述MaxD设置为所述成功MaxD的存储值;

将所述FT设置为所述已存储的成功FT。

15. 根据权利要求14所述的方法,其特征在于,将来自所述网络的所述节点添加到所述FT包括:

初始化候选队列(candidate queue,Cq)以包括在所述网络中耦合到所述节点R0的所有节点,这些节点在所述Cq中按每个节点的nodeID排序;

实现所述方法的第二循环,包括:

确定所述Cq是否包括节点X,所述节点X是所述Cq中的第一节点,所述Cq在所述节点X的上一跳(Previous Hop,PH)中具有节点Y,所述节点Y具有小于MaxD的度;

当所述Cq不包括所述节点X时,终止所述第二循环,其中,所述FT不包括所述网络中的所有节点;

从所述Cq中移除所述节点X,并将所述节点X添加到所述FT;

确定所述FT是否包括所述网络中的所有节点;

当所述FT不包括所述网络中的所有节点时,修改所述Cq;

当所述FT包括所述网络中的所有节点时,终止所述第二循环。

16. 根据权利要求14或15所述的方法,其特征在于,当所述FT不包括所述网络中的所有节点时修改所述Cq包括:

创建在所述网络中耦合到所述节点X且不在所述FT中的所有节点的队列,所述队列按所述队列中的所述节点的nodeID排序;

对于所述队列中的每个节点Z:

确定所述节点Z是否在所述Cq中;

当所述节点Z在所述Cq中时,将所述节点X添加到所述Cq中的所述节点Z的所述PH;

当所述节点Z不在所述Cq中时,将所述节点Z附加到所述Cq。

17. 根据权利要求14至16中任一项所述的方法,其特征在于,向在所述FT中仅具有一条链路并且在所述网络中耦合到链路数小于MaxD的节点的所述FT中的任何节点添加链路包括:

实现所述方法的第三循环,包括:

确定所述FT是否包括在所述FT中仅具有一条链路的节点Q;

当所述FT不包括所述节点Q时,终止所述第三循环,其中,所述FT中的所有节点在所述FT中都具有多条链路;

当所述FT确实包括所述节点Q时,确定所述FT是否包括在所述网络中耦合到Q的节点L,所述节点L (i) 在所述FT中具有最小链路数, (ii) 具有最低nodeID(当两个或多个节点在所述FT中具有相同的最小链路数时),并且 (iii) 具有小于MaxD的度;

当所述FT确实包括所述节点L时:

将所述FT中的所述节点Q的度递增1;

将所述FT中的所述节点L的度递增1;

将所述节点L添加到所述FT中的所述节点Q的所述PH;当所述FT不包括所述节点L时,终止所述第三循环,其中,所述FT中的所有节点在所述FT中都不具有多条链路。

18. 根据权利要求14至17中任一项所述的方法,其特征在于,所述节点R0在所述网络中具有最低nodeID。

19. 根据权利要求14至18中任一项所述的方法,其特征在于,通过从最低nodeID到最高nodeID排序的节点初始化所述Cq。

20. 根据权利要求14至19中任一项所述的方法,其特征在于,附加到所述Cq的节点按从

最低nodeID到最高nodeID的顺序排序。

21. 一种用于计算网络设备中的泛洪拓扑 (flooding topology, FT) 的方法, 其特征在于, 所述方法包括:

设置最大度 (maximum degree, MaxD) 的初始值;

实现所述方法的第一循环, 包括:

从网络中选择节点R0;

通过所述节点R0初始化所述FT;

将来自所述网络的节点添加到所述FT;

确定所述FT是否包括所述网络中的所有节点;

递增所述MaxD的所述值并重复所述第一循环的步骤, 直到所述FT包括所述网络中的所有节点;

当所述FT包括所述网络中的所有节点时, 实现所述方法的第二循环, 包括:

向在所述FT中仅具有一条链路的所述FT中的任何节点添加链路;

确定所述FT中的所有节点在所述FT中是否都具有多条链路;

当所述FT中的所有节点在所述FT中都具有多条链路时, 终止所述第二循环;

在所述FT中找到所述FT中具有最大链路数的节点M;

将所述MaxD的所述值设置为所述节点M在所述FT中的链路数。

22. 根据权利要求21所述的方法, 其特征在于, 将来自所述网络的所述节点添加到所述FT包括:

初始化候选队列 (candidate queue, Cq) 以包括在所述网络中耦合到所述节点R0的所有节点, 这些节点在所述Cq中按每个节点的nodeID排序;

实现所述方法的第三循环, 包括:

确定所述Cq是否包括节点X, 所述节点X是所述Cq中的第一节点, 所述Cq在所述节点X的上一跳 (Previous Hop, PH) 中具有节点Y, 所述节点Y具有小于MaxD的度;

当所述Cq不包括所述节点X时, 终止所述第三循环, 其中, 所述FT不包括所述网络中的所有节点;

从所述Cq中移除所述节点X, 并将所述节点X添加到所述FT;

确定所述FT是否包括所述网络中的所有节点;

当所述FT不包括所述网络中的所有节点时, 修改所述Cq;

当所述FT包括所述网络中的所有节点时, 终止所述第三循环。

23. 根据权利要求21或22所述的方法, 其特征在于, 当所述FT不包括所述网络中的所有节点时修改所述Cq包括:

创建在所述网络中耦合到所述节点X且不在所述FT中的所有节点的队列, 所述队列按所述队列中的所述节点的nodeID排序;

对于所述队列中的每个节点Z:

确定所述节点Z是否在所述Cq中;

当所述节点Z在所述Cq中时, 将所述节点X添加到所述Cq中的所述节点Z的所述PH;

当所述节点Z不在所述Cq中时, 将所述节点Z附加到所述Cq。

24. 根据权利要求21至23中任一项所述的方法, 其特征在于, 向在所述FT中仅具有一条

链路的所述FT中的任何节点添加所述链路包括：

实现所述方法的第四循环，包括：

确定所述FT是否包括在所述FT中仅具有一条链路的节点Q；

当所述FT不包括所述节点Q时，终止所述第四循环，其中，所述FT中的所有节点在所述FT中都具有多条链路；

当所述FT确实包括所述节点Q时，确定所述FT是否包括在所述网络中耦合到Q的节点L，所述节点L (i) 在所述FT中具有最小链路数并且 (ii) 具有最低nodeID (当两个或多个节点在所述FT中具有相同的最小链路数时)；

当所述FT确实包括所述节点L时：

将所述FT中的所述节点Q的度递增1；

将所述FT中的所述节点L的度递增1；

将所述节点L添加到所述FT中的所述节点Q的所述PH。

25. 根据权利要求21至24中任一项所述的方法，其特征在于，所述节点R0在所述网络中具有最低nodeID。

26. 根据权利要求21至25中任一项所述的方法，其特征在于，通过从最低nodeID到最高nodeID排序的节点初始化所述Cq。

27. 根据权利要求21至26中任一项所述的方法，其特征在于，附加到所述Cq的节点按从最低nodeID到最高nodeID的顺序排序。

28. 一种用于计算网络设备中的泛洪拓扑 (flooding topology, FT) 的方法，其特征在于，所述方法包括：

设置最大度 (maximum degree, MaxD) 的初始值；

获取用于识别网络中节点子集的信息，所述网络具有对所述节点子集中的节点的所述FT中链路数的约束 (ConMaxD)；

实现所述方法的第一循环，包括：

从所述网络中选择节点R0；

通过所述节点R0初始化所述FT；

将来自所述网络的节点添加到所述FT；

确定所述FT是否包括所述网络中的所有节点；

递增所述MaxD的所述值并重复所述第一循环的步骤，直到所述FT包括所述网络中的所有节点；

当所述FT包括所述网络中的所有节点时，实现所述方法的第二循环，包括：

向所述FT中的任何节点Q添加链路，所述节点Q在所述FT中仅具有一条链路并且在所述网络中耦合到节点L，所述节点L是在所述FT中链路数小于所述ConMaxD的节点子集中的节点或是不在所述FT中链路数小于所述MaxD的节点子集中的节点；

确定所述FT中的所有节点在所述FT中是否都具有多条链路；

递增所述MaxD的所述值并重复所述第一循环和所述第二循环，直到所述FT中的所有节点在所述FT中都具有多条链路。

29. 根据权利要求28所述的方法，其特征在于，将来自所述网络的所述节点添加到所述FT包括：

初始化候选队列(candidate queue,Cq)以包括在所述网络中耦合到所述节点R0的所有节点,这些节点在所述Cq中按每个节点的nodeID排序;

实现所述方法的第三循环,包括:

确定所述Cq是否包括节点X,所述节点X是所述Cq中的第一节点,所述Cq(i)在所述节点X的上一跳(Previous Hop,PH)中具有节点Y,所述节点Y具有小于所述ConMaxD的度(当所述节点X是所述节点子集中的节点时),或者(ii)在所述节点X的上一跳(Previous Hop,PH)中具有节点Y,所述节点Y具有小于所述MaxD的度(当所述节点X不是所述节点子集中的节点时);

当所述Cq不包括所述节点X时,终止所述第三循环,其中,所述FT不包括所述网络中的所有节点;

从所述Cq中移除所述节点X,并将所述节点X添加到所述FT;

确定所述FT是否包括所述网络中的所有节点;

当所述FT不包括所述网络中的所有节点时,修改所述Cq;

当所述FT包括所述网络中的所有节点时,终止所述第三循环。

30.根据权利要求28或29所述的方法,其特征在于,当所述FT不包括所述网络中的所有节点时修改所述Cq包括:

创建在所述网络中耦合到所述节点X且不在所述FT中的所有节点的队列,所述队列按所述队列中的所述节点的nodeID排序;

对于所述队列中的每个节点Z:

确定所述节点Z是否在所述Cq中;

当所述节点Z在所述Cq中时,将所述节点X添加到所述Cq中的所述节点Z的所述PH;

当所述节点Z不在所述Cq中时,将所述节点Z附加到所述Cq。

31.根据权利要求28至30中任一项所述的方法,其特征在于,向在所述FT中仅具有一条链路并且在所述网络中耦合到链路数小于MaxD的节点的所述FT中的任何节点添加链路包括:

实现所述方法的第四循环,包括:

确定所述FT是否包括在所述FT中仅具有一条链路的节点Q;

当所述FT不包括所述节点Q时,终止所述第四循环,其中,所述FT中的所有节点在所述FT中都具有多条链路;

当所述FT确实包括所述节点Q时,确定所述FT是否包括在所述网络中耦合到Q的节点L,所述节点L(i)在所述FT中具有最小链路数,(ii)具有最低nodeID(当两个或多个节点在所述FT中具有相同的最小链路数时),并且(iii)(a)具有小于所述ConMaxD的度(当所述节点L是所述节点子集中的节点时),或者(b)具有小于所述MaxD的度(当所述节点L不是所述节点子集中的节点时);

当所述FT确实包括所述节点L时:

将所述FT中的所述节点Q的度递增1;

将所述FT中的所述节点L的度递增1;

将所述节点L添加到所述FT中的所述节点Q的所述PH;当所述FT不包括所述节点L时,终止所述第四循环,其中,所述FT中的所有节点在所述FT中都不具有多条链路。

32. 根据权利要求28至31中任一项所述的方法,其特征在于,所述节点R0在所述网络中具有最低nodeID。

33. 根据权利要求28至32中任一项所述的方法,其特征在于,通过从最低nodeID到最高nodeID排序的节点初始化所述C_q。

34. 根据权利要求28至33中任一项所述的方法,其特征在于,附加到所述C_q的节点按从最低nodeID到最高nodeID的顺序排序。

35. 一种用于计算网络设备中的完整泛洪拓扑(flooding topology, FT)的方法,其特征在于,所述方法包括:

获取用于识别网络中节点子集的信息,所述网络具有对所述节点子集中的节点的所述FT中链路数的约束(ConMaxD);

计算包括所述网络中的所有确定性节点的第一泛洪拓扑(FT0),所述确定性节点是所述子集中的节点,与所述网络中其它节点的链路数等于ConMaxD;

形成网络拓扑(NT1),所述网络拓扑包括所述网络中不在所述FT0中的所有节点;

为所述NT1中的所述节点计算第二泛洪拓扑(FT1);

通过将所述FT0附加到所述FT1来计算所述完整FT。

36. 根据权利要求35所述的方法,其特征在于,计算包括所述网络中的所有确定性节点的所述FT0包括:

初始化第一候选队列(C_{q0})以包括所述网络中的所有节点;

实现所述方法的第一循环,包括:

确定所述C_{q0}是否包括节点X,所述节点X是所述C_{q0}中作为确定性节点的第一节点;

当所述C_{q0}不包括节点X时,终止所述第一循环;

当所述C_{q0}确实包括节点X时:

从所述C_{q0}中移除所述节点X;

将所述节点X添加到所述FT0中,所述FT0的度等于ConMaxD且上一跳(Previous Hop, PH)包含所述网络中与所述网络中的所述节点X链接的节点;

递增C_{q0}中所述网络中与所述网络中的所述节点X链接的每个节点的度。

37. 根据权利要求35或36所述的方法,其特征在于,为所述NT1中的所述节点计算所述FT1包括:

确定所述NT1中的任何节点是否在所述节点子集中;

当所述NT1中没有节点在所述节点子集中时,为所述NT1计算最小化的FT1;

当所述NT1中的节点在所述节点子集中时,为所述NT1计算受约束且最小化的FT1。

38. 根据权利要求35至37中任一项所述的方法,其特征在于,为所述NT1计算所述最小化的FT1包括:

在所述C_{q0}中找到具有所述C_{q0}中最高节点度值的节点H;

将最大度(maximum degree, MaxD)的初始值设置为等于1加上所述节点H的度值;

实现所述方法的第二循环,包括:

从所述NT1中选择节点R0;

通过所述节点R0初始化所述FT1;

基于所述MaxD将来自所述网络的节点添加到所述FT1;

确定所述FT1是否包括所述NT1中的所有节点；

递增所述MaxD的所述值并重复所述第二循环的步骤，直到所述FT1包括所述NT1中的所有节点；

当所述FT1包括所述NT1中的所有节点时，实现所述方法的第三循环，包括：

向在所述FT1中仅具有一条链路并且在所述NT1中耦合到链路数小于MaxD的节点的所述FT1中的任何节点添加链路；

确定所述FT1中的所有节点在所述FT1中是否都具有多条链路；

递增所述MaxD的所述值并重复所述第二循环和所述第三循环，直到所述FT1中的所有节点在所述FT1中都具有多条链路。

39. 根据权利要求35至38中任一项所述的方法，其特征在于，为所述NT1计算所述受约束且最小化的FT1包括：

在所述Cq0中找到具有所述Cq0中最高节点度值的节点H；

将最大度 (maximum degree, MaxD) 的初始值设置为等于1加上所述节点H的度值；

实现所述方法的第四循环，包括：

从所述NT1中选择节点R0；

通过所述节点R0初始化所述FT1；

基于所述MaxD和所述ConMaxD将来自所述网络的节点添加到所述FT1；

确定所述FT1是否包括所述NT1中的所有节点；

递增所述MaxD的所述值并重复所述第二循环的步骤，直到所述FT1包括所述NT1中的所有节点；

当所述FT1包括所述NT1中的所有节点时，实现所述方法的第五循环，包括：

向所述FT1中的任何节点Q添加链路，所述节点Q在所述FT1中仅具有一条链路并且在所述网络中耦合到节点L，所述节点L是在所述FT1中链路数小于所述ConMaxD的节点子集中的节点或是在不在所述FT1中链路数小于所述MaxD的节点子集中的节点；

确定所述FT1中的所有节点在所述FT1中是否都具有多条链路；

递增所述MaxD的所述值并重复所述第一循环和所述第二循环，直到所述FT1中的所有节点在所述FT1中都具有多条链路。

40. 一种网络设备，其特征在于，所述网络设备包括：

接收器单元，用于接收数据；

发送器单元，用于发送数据；

处理器，耦合到所述接收器单元或所述发送器中的至少一个；

存储器，耦合到所述处理器且用于存储指令，所述处理器用于执行所述存储器中存储的所述指令，以执行权利要求1至39中任一项所述的方法。

41. 一种用于使用权利要求1至13和21至39中任一项所述的方法来计算泛洪拓扑 (flooding topology, FT) 的装置，其特征在于，所述装置包括：

用于计算FT的装置，所述FT包括网络中的所有节点；

用于确保所述FT中的所有节点在所述FT中都至少具有两条链路的装置。

42. 一种用于使用权利要求14至20中任一项所述的方法来计算完整泛洪拓扑 (flooding topology, FT) 的装置，其特征在于，所述装置包括：

用于计算FT0的装置,所述FT0包括所述网络中的所有确定性节点;
具有以下用途的装置:
计算FT1,所述FT1包括所述网络中不在所述FT0中的所有节点;
将所述FT0附加到所述FT1,以计算所述完整FT。

用于计算泛洪拓扑的系统和方法

[0001] 相关申请案交叉引用

[0002] 本申请要求于2019年7月5日由Huaimo Chen递交的申请号为62/870,893、发明名称为“用于计算泛洪拓扑的系统和方法(System and Method for Computing Flooding Topology)”的美国临时专利申请案的优先权的权益,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,涉及一种用于计算泛洪拓扑的系统和方法。

背景技术

[0004] 路由协议指定路由器相互通信的方式,分发信息使路由器能够选择计算机网络上任意两个节点之间的路由。内部网关协议(interior gateway protocol,IGP)是一种用于在自治系统(autonomous system,AS)内的网关(通常为路由器)之间交换路由信息的协议。然后,该路由信息可用于路由网络层协议,如互联网协议(Internet Protocol,IP)包。AS是代表单个管理实体或域的一个或多个网络运营商控制下的连接IP路由前缀的集合,该管理实体或域向互联网(例如,企业局域网系统)呈现通用的、明确定义的路由策略。

[0005] 对于IGP,有不同的类型。第1类IGP称为链路状态路由协议。链路状态路由协议由网络中的每个交换节点(即,准备转发包的节点;在互联网中,这些节点称为路由器)执行。链路状态路由的基本概念是每个节点以图形的形式构建到网络的连接图,显示哪些节点连接到哪些其它节点。然后,每个节点独立计算从它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。然后每个最佳路径的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,OSPF)路由协议和中间系统到中间系统(Intermediate System to Intermediate System,IS-IS或ISIS)协议。

[0006] OSPF是互联网工程任务组(Internet Engineering Task Force,IETF)指定的标准路由协议。OSPF使用链路状态通告(Link State Advertisement,LSA)在路由器之间交换路由信息。区域内的每个路由器将泛洪该区域内的第1类LSA(也称为路由器LSA)。LSA封装在OSPF包之后,然后是IP包。区域是具有相同区域编号的基于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包含关于路由器所属区域内的直连链路的信息(例如,具有通告路由器的所有直连链路的列表)。该列表将泛洪到该区域内的所有路由器。如果路由器是区域边界路由器(Area Border Router,ABR),它将为所连接的所有区域生成第1类LSA,并将这些LSA发送到相应区域内的所有邻居。

[0007] IS-IS是由国际标准化组织(International Organization for Standardization,ISO)制定的标准化路由协议。IS-IS使用链路状态协议数据单元(Link State Protocol Data Unit,LSP)在路由器之间交换路由信息。LSP是网络路由器在链路状态路由协议中生成的信息包,该协议列出了路由器的邻居。LSP包还可以进一步定义为专用数据报,用于确定任何相邻路由器和关联网络的名称、成本或距离。它们用于有效地确定新

邻居是什么,是否发生链路故障,以及在需要时更改链路的成本。

[0008] OSPF与IS-IS之间的一些额外区别在于,OSPF支持非广播多址网络(non-broadcast multiple access network,NBMA)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] 无论使用哪种IGP路由协议,随着网络规模越来越大,当网络发生链路故障等变化时,IGP路由收敛速度会变慢。当网络中每个路由器的所有组件(包括路由信息库(Routing Information Base,RIB)和转发信息库(Forwarding Information Base,FIB)以及软件和硬件表)都获得最近的路由更改时,就会发生IGP收敛,以便在最佳出口(或“下一跳”)接口上成功完成路由表项的转发。

[0010] 网络中节点之间的链路在存储在路由器的链路状态数据库(link state database,LSDB)中的网络拓扑(network topology,NT)中进行描述。网络中的每个路由器都有一个LSDB,它表示路由器所属区域的网络拓扑。在每个路由器上运行的算法将使用路由器的LSDB中的网络拓扑。

[0011] 即使底层(物理)拓扑未拆分,一个或多个故障也可能会拆分泛洪拓扑(flooding topology,FT)。计算和构建新FT并将新FT从主节点(leader node)泛洪到网络中的每个节点都需要时间。在此期间,链路状态数据库(link state database,LSDB)可能不同步并且网络收敛速度会减慢。

发明内容

[0012] 第一方面涉及一种用于计算网络设备中的泛洪拓扑(flooding topology,FT)的方法。所述方法包括:从网络中选择节点R0;通过所述节点R0初始化所述FT;初始化候选队列(candidate queue,Cq);实现所述方法的第一循环。所述方法的所述第一循环包括:从所述Cq中移除第一节点并将所述第一节点附加到所述FT;确定所述FT是否包括所述网络中的所有节点;当所述FT包括所述网络中的所有节点时,终止所述第一循环;当所述FT不包括所述网络中的所有节点时,将在所述网络中耦合到所述第一节点并且不在所述FT中的那些节点附加到所述Cq。所述方法还包括:向在所述FT中仅具有一条链路的所述FT中的任何节点添加链路。

[0013] 可选地,在上述任一方面,根据所述方面的另一种实现方式,所述方法的所述第一循环还包括:当所述FT不包括所述网络中的所有节点时,确定所述Cq是否为空;当所述Cq为空时:向所述网络的管理员发送第一消息;向所述网络设备中的组件发送第二消息,所述第一消息和所述第二消息指示所述网络中存在未连接到所述网络的其它节点的节点;终止权利要求1所述的方法。

[0014] 可选地,在上述任一方面,根据所述方面的另一种实现方式,向在所述FT中仅具有一条链路的所述FT中的任何节点添加所述链路包括:实现所述方法的第二循环。所述方法的所述第二循环包括:确定所述FT是否包括在所述FT中仅具有一条链路的节点Q;当所述FT不包括所述节点Q时,终止所述第二循环,其中,所述FT中的所有节点在所述FT中都具有多条链路;当所述FT确实包括所述节点Q时:创建在所述网络中耦合到所述节点Q的所有节点的集合;选择所述集合中的节点L,所述节点L(i)在所述FT中具有最小链路数,并且(ii)具有最低nodeID(当两个或多个节点在所述FT中具有相同的最小链路数时);将所述FT中的所

述节点Q的度递增1;将所述FT中的所述节点L的度递增1;将Q与L之间的链路添加到所述FT。

[0015] 可选地,在上述任一方面,根据所述方面的另一种实现方式,所述节点R0在所述网络中具有最低nodeID。

[0016] 可选地,在上述任一方面,根据所述方面的另一种实现方式,通过从最低nodeID到最高nodeID排序的节点初始化所述Cq。

[0017] 可选地,在上述任一方面,根据所述方面的另一种实现方式,附加到所述Cq的节点按从最低nodeID到最高nodeID的顺序排序。

[0018] 第二方面涉及一种用于计算网络设备中的泛洪拓扑(flooding topology,FT)的方法。所述方法包括:设置最大度(maximum degree,MaxD)的初始值;实现所述方法的第一循环。所述方法的所述第一循环包括:从网络中选择节点R0;通过所述节点R0初始化所述FT;将来自所述网络的节点添加到所述FT;确定所述FT是否包括所述网络中的所有节点;递增所述MaxD的所述值并重复所述第一循环的步骤,直到所述FT包括所述网络中的所有节点。所述方法还包括:当所述FT包括所述网络中的所有节点时,实现所述方法的第二循环。所述方法的所述第二循环包括:向在所述FT中仅具有一条链路并且在所述网络中耦合到链路数小于MaxD的节点的所述FT中的任何节点添加链路;确定所述FT中的所有节点在所述FT中是否都具有多条链路;递增所述MaxD的所述值并重复所述第一循环和所述第二循环,直到所述FT中的所有节点在所述FT中都具有多条链路。

[0019] 可选地,在上述任一方面,根据所述方面的另一种实现方式,将来自所述网络的所述节点添加到所述FT包括:初始化候选队列(candidate queue,Cq)以包括在所述网络中耦合到所述节点R0的所有节点,这些节点在所述Cq中按每个节点的nodeID排序;实现所述方法的第三循环。所述方法的所述第三循环包括:确定所述Cq是否包括节点X,所述节点X是所述Cq中的第一节点,所述Cq在所述节点X的上一跳(Previous Hop,PH)中具有节点Y,所述节点Y具有小于MaxD的度;当所述Cq不包括所述节点X时,终止所述第三循环,其中,所述FT不包括所述网络中的所有节点;从所述Cq中移除所述节点X,并将所述节点X添加到所述FT;确定所述FT是否包括所述网络中的所有节点;当所述FT不包括所述网络中的所有节点时,修改所述Cq;当所述FT包括所述网络中的所有节点时,终止所述第三循环。

[0020] 可选地,在上述任一方面,根据所述方面的另一种实现方式,当所述FT不包括所述网络中的所有节点时修改所述Cq包括:创建在所述网络中耦合到所述节点X且不在所述FT中的所有节点的队列,所述队列按所述队列中的所述节点的nodeID排序;对于所述队列中的每个节点Z:确定所述节点Z是否在所述Cq中;当所述节点Z在所述Cq中时,将所述节点X添加到所述Cq中的所述节点Z的所述PH;当所述节点Z不在所述Cq中时,将所述节点Z附加到所述Cq。

[0021] 可选地,在上述任一方面,根据所述方面的另一种实现方式,向在所述FT中仅具有一条链路并且在所述网络中耦合到链路数小于MaxD的所述节点的所述FT中的任何节点添加所述链路包括:实现所述方法的第四循环。所述方法的所述第四循环包括:确定所述FT是否包括在所述FT中仅具有一条链路的节点Q;当所述FT不包括所述节点Q时,终止所述第四循环,其中,所述FT中的所有节点在所述FT中都具有多条链路;当所述FT确实包括所述节点Q时,确定所述FT是否包括在所述网络中耦合到Q的节点L,所述节点L(i)在所述FT中具有最小链路数,(ii)具有最低nodeID(当两个或多个节点在所述FT中具有相同的最小链路数

时),并且(iii)具有小于MaxD的度;当所述FT确实包括所述节点L时:将所述FT中的所述节点Q的度递增1;将所述FT中的所述节点L的度递增1;将所述节点L添加到所述FT中的所述节点Q的所述PH;当所述FT不包括所述节点L时,终止所述第四循环,其中,所述FT中的所有节点在所述FT中都不具有多条链路。

[0022] 可选地,在上述任一方面,根据所述方面的另一种实现方式,所述节点R0在所述网络中具有最低nodeID。

[0023] 可选地,在上述任一方面,根据所述方面的另一种实现方式,通过从最低nodeID到最高nodeID排序的节点初始化所述Cq。

[0024] 可选地,在上述任一方面,根据所述方面的另一种实现方式,附加到所述Cq的节点按从最低nodeID到最高nodeID的顺序排序。

[0025] 第三方面涉及一种用于计算网络设备中的泛洪拓扑(flooding topology,FT)的方法。所述方法包括:设置最大度(maximum degree,MaxD)的初始值,所述初始值等于先前计算的所述网络FT的MaxD的存储值;实现所述方法的第一循环。所述方法的所述第一循环包括:从所述网络中选择节点R0;通过所述节点R0初始化所述FT;将来自所述网络的节点添加到所述FT;确定所述FT是否包括所述网络中的所有节点;当所述FT包括所述网络中的所有节点时,向在所述FT中仅具有一条链路并且在所述网络中耦合到链路数小于MaxD的节点的所述FT中的任何节点添加链路;确定(i)所述FT是否包括所述网络中的所有节点,以及(ii)所述FT中的所有节点在所述FT中是否都具有多条链路;当(i)所述FT包括所述网络中的所有节点且(ii)所述FT中的所有节点在所述FT中都具有多条链路时,将MaxD的所述值存储为成功的MaxD并且将所述FT存储为成功的FT;递减所述MaxD的所述值并重复所述第一循环,直到(i)所述FT不包括所述网络中的所有节点或者(ii)并非所述FT中的所有节点在所述FT中都具有多条链路。所述方法还包括:将所述MaxD设置为所述成功MaxD的存储值;将所述FT设置为所述已存储的成功FT。

[0026] 可选地,在上述任一方面,根据所述方面的另一种实现方式,将来自所述网络的所述节点添加到所述FT包括:初始化候选队列(candidate queue,Cq)以包括在所述网络中耦合到所述节点R0的所有节点,这些节点在所述Cq中按每个节点的nodeID排序;实现所述方法的第二循环。所述方法的所述第二循环包括:确定所述Cq是否包括节点X,所述节点X是所述Cq中的第一节点,所述Cq在所述节点X的上一跳(Previous Hop,PH)中具有节点Y,所述节点Y具有小于MaxD的度;当所述Cq不包括所述节点X时,终止所述第二循环,其中,所述FT不包括所述网络中的所有节点;从所述Cq中移除所述节点X,并将所述节点X添加到所述FT;确定所述FT是否包括所述网络中的所有节点;当所述FT不包括所述网络中的所有节点时,修改所述Cq;当所述FT包括所述网络中的所有节点时,终止所述第二循环。

[0027] 可选地,在上述任一方面,根据所述方面的另一种实现方式,当所述FT不包括所述网络中的所有节点时修改所述Cq包括:创建在所述网络中耦合到所述节点X且不在所述FT中的所有节点的队列,所述队列按所述队列中的所述节点的nodeID排序;对于所述队列中的每个节点Z:确定所述节点Z是否在所述Cq中;当所述节点Z在所述Cq中时,将所述节点X添加到所述Cq中的所述节点Z的所述PH;当所述节点Z不在所述Cq中时,将所述节点Z附加到所述Cq。

[0028] 可选地,在上述任一方面,根据所述方面的另一种实现方式,向在所述FT中仅具有

一条链路并且在所述网络中耦合到链路数小于MaxD的节点的所述FT中的任何节点添加链路包括:实现所述方法的第三循环。所述方法的所述第三循环包括:确定所述FT是否包括在所述FT中仅具有一条链路的节点Q;当所述FT不包括所述节点Q时,终止所述第三循环,其中,所述FT中的所有节点在所述FT中都具有多条链路;当所述FT确实包括所述节点Q时,确定所述FT是否包括在所述网络中耦合到Q的节点L,所述节点L (i) 在所述FT中具有最小链路数,(ii) 具有最低nodeID(当两个或多个节点在所述FT中具有相同的最小链路数时),并且(iii) 具有小于MaxD的度;当所述FT确实包括所述节点L时:将所述FT中的所述节点Q的度递增1;将所述FT中的所述节点L的度递增1;将所述节点L添加到所述FT中的所述节点Q的所述PH;当所述FT不包括所述节点L时,终止所述第三循环,其中,所述FT中的所有节点在所述FT中都不具有多条链路。

[0029] 可选地,在上述任一方面,根据所述方面的另一种实现方式,所述节点R0在所述网络中具有最低nodeID。

[0030] 可选地,在上述任一方面,根据所述方面的另一种实现方式,通过从最低nodeID到最高nodeID排序的节点初始化所述Cq。

[0031] 可选地,在上述任一方面,根据所述方面的另一种实现方式,附加到所述Cq的节点按从最低nodeID到最高nodeID的顺序排序。

[0032] 第四方面涉及一种用于计算网络设备中的泛洪拓扑(flooding topology,FT)的方法。所述方法包括:设置最大度(maximum degree,MaxD)的初始值;实现所述方法的第一循环。所述方法的所述第一循环包括:从网络中选择节点R0;通过所述节点R0初始化所述FT;将来自所述网络的节点添加到所述FT;确定所述FT是否包括所述网络中的所有节点;递增所述MaxD的所述值并重复所述第一循环的步骤,直到所述FT包括所述网络中的所有节点。所述方法还包括:当所述FT包括所述网络中的所有节点时,实现所述方法的第二循环。所述方法的所述第二循环包括:向在所述FT中仅具有一条链路的所述FT中的任何节点添加链路;确定所述FT中的所有节点在所述FT中是否都具有多条链路;当所述FT中的所有节点在所述FT中都具有多条链路时,终止所述第二循环。所述方法还包括:在所述FT中找到所述FT中具有最大链路数的节点M;将所述MaxD的所述值设置为所述节点M在所述FT中的链路数。

[0033] 可选地,在上述任一方面,根据所述方面的另一种实现方式,将来自所述网络的所述节点添加到所述FT包括:初始化候选队列(candidate queue,Cq)以包括在所述网络中耦合到所述节点R0的所有节点,这些节点在所述Cq中按每个节点的nodeID排序;实现所述方法的第三循环。所述方法的所述第三循环包括:确定所述Cq是否包括节点X,所述节点X是所述Cq中的第一节点,所述Cq在所述节点X的上一跳(Previous Hop,PH)中具有节点Y,所述节点Y具有小于MaxD的度;当所述Cq不包括所述节点X时,终止所述第三循环,其中,所述FT不包括所述网络中的所有节点;从所述Cq中移除所述节点X,并将所述节点X添加到所述FT;确定所述FT是否包括所述网络中的所有节点;当所述FT不包括所述网络中的所有节点时,修改所述Cq;当所述FT包括所述网络中的所有节点时,终止所述第三循环。

[0034] 可选地,在上述任一方面,根据所述方面的另一种实现方式,当所述FT不包括所述网络中的所有节点时修改所述Cq包括:创建在所述网络中耦合到所述节点X且不在所述FT中的所有节点的队列,所述队列按所述队列中的所述节点的nodeID排序;对于所述队列中

的每个节点Z:确定所述节点Z是否在所述Cq中;当所述节点Z在所述Cq中时,将所述节点X添加到所述Cq中的所述节点Z的所述PH;当所述节点Z不在所述Cq中时,将所述节点Z附加到所述Cq。

[0035] 可选地,在上述任一方面,根据所述方面的另一种实现方式,向在所述FT中仅具有一条链路的所述FT中的任何节点添加所述链路包括:实现所述方法的第四循环。所述方法的所述第四循环包括:确定所述FT是否包括在所述FT中仅具有一条链路的节点Q;当所述FT不包括所述节点Q时,终止所述第四循环,其中,所述FT中的所有节点在所述FT中都具有多条链路;当所述FT确实包括所述节点Q时,确定所述FT是否包括在所述网络中耦合到Q的节点L,所述节点L (i) 在所述FT中具有最小链路数并且 (ii) 具有最低nodeID(当两个或多个节点在所述FT中具有相同的最小链路数时);当所述FT确实包括所述节点L时:将所述FT中的所述节点Q的度递增1;将所述FT中的所述节点L的度递增1;将所述节点L添加到所述FT中的所述节点Q的所述PH。

[0036] 可选地,在上述任一方面,根据所述方面的另一种实现方式,所述节点R0在所述网络中具有最低nodeID。

[0037] 可选地,在上述任一方面,根据所述方面的另一种实现方式,通过从最低nodeID到最高nodeID排序的节点初始化所述Cq。

[0038] 可选地,在上述任一方面,根据所述方面的另一种实现方式,附加到所述Cq的节点按从最低nodeID到最高nodeID的顺序排序。

[0039] 第五方面涉及一种用于计算网络设备中的泛洪拓扑(flooding topology,FT)的方法。所述方法包括:设置最大度(maximum degree,MaxD)的初始值;获取用于识别网络中节点子集的信息,所述网络具有对所述节点子集中的节点的所述FT中链路数的约束(ConMaxD);实现所述方法的第一循环。所述方法的所述第一循环包括:从所述网络中选择节点R0;通过所述节点R0初始化所述FT;将来自所述网络的节点添加到所述FT;确定所述FT是否包括所述网络中的所有节点;递增所述MaxD的所述值并重复所述第一循环的步骤,直到所述FT包括所述网络中的所有节点。所述方法还包括:当所述FT包括所述网络中的所有节点时,实现所述方法的第二循环。所述方法的所述第二循环包括:向所述FT中的任何节点Q添加链路,所述节点Q在所述FT中仅具有一条链路并且在所述网络中耦合到节点L,所述节点L是在所述FT中链路数小于所述ConMaxD的节点子集中的节点或是在不在所述FT中链路数小于所述MaxD的节点子集中的节点;确定所述FT中的所有节点在所述FT中是否都具有多条链路;递增所述MaxD的所述值并重复所述第一循环和所述第二循环,直到所述FT中的所有节点在所述FT中都具有多条链路。

[0040] 可选地,在上述任一方面,根据所述方面的另一种实现方式,将来自所述网络的所述节点添加到所述FT包括:初始化候选队列(candidate queue,Cq)以包括在所述网络中耦合到所述节点R0的所有节点,这些节点在所述Cq中按每个节点的nodeID排序;实现所述方法的第三循环。所述方法的所述第三循环包括:确定所述Cq是否包括节点X,所述节点X是所述Cq中的第一节点,所述Cq (i) 在所述节点X的上一跳(Previous Hop,PH)中具有节点Y,所述节点Y具有小于所述ConMaxD的度(当所述节点X是所述节点子集中的节点时),或者(ii) 在所述节点X的上一跳(Previous Hop,PH)中具有节点Y,所述节点Y具有小于所述MaxD的度(当所述节点X不是所述节点子集中的节点时);当所述Cq不包括所述节点X时,终止所述第

三循环,其中,所述FT不包括所述网络中的所有节点;从所述Cq中移除所述节点X,并将所述节点X添加到所述FT;确定所述FT是否包括所述网络中的所有节点;当所述FT不包括所述网络中的所有节点时,修改所述Cq;当所述FT包括所述网络中的所有节点时,终止所述第三循环。

[0041] 可选地,在上述任一方面,根据所述方面的另一种实现方式,当所述FT不包括所述网络中的所有节点时修改所述Cq包括:创建在所述网络中耦合到所述节点X且不在所述FT中的所有节点的队列,所述队列按所述队列中的所述节点的nodeID排序;对于所述队列中的每个节点Z:确定所述节点Z是否在所述Cq中;当所述节点Z在所述Cq中时,将所述节点X添加到所述Cq中的所述节点Z的所述PH;当所述节点Z不在所述Cq中时,将所述节点Z附加到所述Cq。

[0042] 可选地,在上述任一方面,根据所述方面的另一种实现方式,向在所述FT中仅具有一条链路并且在所述网络中耦合到链路数小于MaxD的节点的所述FT中的任何节点添加链路包括:实现所述方法的第四循环。所述方法的所述第四循环包括:确定所述FT是否包括在所述FT中仅具有一条链路的节点Q;当所述FT不包括所述节点Q时,终止所述第四循环,其中,所述FT中的所有节点在所述FT中都具有多条链路;当所述FT确实包括所述节点Q时,确定所述FT是否包括在所述网络中耦合到Q的节点L,所述节点L (i) 在所述FT中具有最小链路数, (ii) 具有最低nodeID(当两个或多个节点在所述FT中具有相同的最小链路数时),并且 (iii) (a) 具有小于所述ConMaxD的度(当所述节点L是所述节点子集中的节点时),或者 (b) 具有小于所述MaxD的度(当所述节点L不是所述节点子集中的节点时);当所述FT确实包括所述节点L时:将所述FT中的所述节点Q的度递增1;将所述FT中的所述节点L的度递增1;将所述节点L添加到所述FT中的所述节点Q的所述PH;当所述FT不包括所述节点L时,终止所述第四循环,其中,所述FT中的所有节点在所述FT中都不具有多条链路。

[0043] 可选地,在上述任一方面,根据所述方面的另一种实现方式,所述节点R0在所述网络中具有最低nodeID。

[0044] 可选地,在上述任一方面,根据所述方面的另一种实现方式,通过从最低nodeID到最高nodeID排序的节点初始化所述Cq。

[0045] 可选地,在上述任一方面,根据所述方面的另一种实现方式,附加到所述Cq的节点按从最低nodeID到最高nodeID的顺序排序。

[0046] 第六方面涉及一种用于计算网络设备中的完整泛洪拓扑(flooding topology, FT)的方法。所述方法包括:获取用于识别网络中节点子集的信息,所述网络具有对所述节点子集中的节点的所述FT中链路数的约束(ConMaxD);计算包括所述网络中的所有确定性节点的第一泛洪拓扑(FT0),所述确定性节点是所述子集中的节点,与所述网络中其它节点的链路数等于ConMaxD;形成网络拓扑(NT1),所述网络拓扑包括所述网络中不在所述FT0中的所有节点;为所述NT1中的所述节点计算第二泛洪拓扑(FT1);通过将所述FT0附加到所述FT1来计算所述完整FT。

[0047] 可选地,在上述任一方面,根据所述方面的另一种实现方式,计算包括所述网络中的所有确定性节点的所述FT0包括:初始化第一候选队列(Cq0)以包括所述网络中的所有节点;实现所述方法的第一循环。所述方法的所述第一循环包括:确定所述Cq0是否包括节点X,所述节点X是所述Cq0中作为确定性节点的第一节点;当所述Cq0不包括节点X时,终止所

述第一循环;当所述Cq0确实包括节点X时:从所述Cq0中移除所述节点X;将所述节点X添加到所述FT0,所述FT0的度等于ConMaxD且上一跳(Previous Hop,PH)包含所述网络中与所述网络中的所述节点X链接的节点;递增Cq0中所述网络中与所述网络中的所述节点X链接的每个节点的度。

[0048] 可选地,在上述任一方面,根据所述方面的另一种实现方式,为所述NT1中的所述节点计算所述FT1包括:确定所述NT1中的任何节点是否在所述节点子集中;当所述NT1中没有节点在所述节点子集中时,为所述NT1计算最小化的FT1;当所述NT1中的节点在所述节点子集中时,为所述NT1计算受约束且最小化的FT1。

[0049] 可选地,在上述任一方面,根据所述方面的另一种实现方式,为所述NT1计算所述最小化的FT1包括:在所述Cq0中找到具有所述Cq0中最高节点度值的节点H;将最大度(maximum degree,MaxD)的初始值设置为等于1加上所述节点H的度值;实现所述方法的第二循环。所述方法的所述第二循环包括:从所述NT1中选择节点R0;通过所述节点R0初始化所述FT1;基于所述MaxD将来自所述网络的节点添加到所述FT1;确定所述FT1是否包括所述NT1中的所有节点;递增所述MaxD的所述值并重复所述第二循环的步骤,直到所述FT1包括所述NT1中的所有节点。所述方法还包括:当所述FT1包括所述NT1中的所有节点时,实现所述方法的第三循环。所述方法的所述第三循环包括:向在所述FT1中仅具有一条链路并且在所述NT1中耦合到链路数小于MaxD的节点的所述FT1中的任何节点添加链路;确定所述FT1中的所有节点在所述FT1中是否都具有多条链路;递增所述MaxD的所述值并重复所述第二循环和所述第三循环,直到所述FT1中的所有节点在所述FT1中都具有多条链路。

[0050] 可选地,在上述任一方面,根据所述方面的另一种实现方式,为所述NT1计算所述受约束且最小化的FT1包括:在所述Cq0中找到具有所述Cq0中最高节点度值的节点H;将最大度(maximum degree,MaxD)的初始值设置为等于1加上所述节点H的度值;实现所述方法的第四循环。所述方法的所述第四循环包括:从所述NT1中选择节点R0;通过所述节点R0初始化所述FT1;基于所述MaxD和所述ConMaxD将来自所述网络的节点添加到所述FT1;确定所述FT1是否包括所述NT1中的所有节点;递增所述MaxD的所述值并重复所述第二循环的步骤,直到所述FT1包括所述NT1中的所有节点。所述方法还包括:当所述FT1包括所述NT1中的所有节点时,实现所述方法的第五循环。所述方法的所述第五循环包括:向所述FT1中的任何节点Q添加链路,所述节点Q在所述FT1中仅具有一条链路并且在所述网络中耦合到节点L,所述节点L是在所述FT1中链路数小于所述ConMaxD的节点子集中的节点或是在不在所述FT1中链路数小于所述MaxD的节点子集中的节点;确定所述FT1中的所有节点在所述FT1中是否都具有多条链路;递增所述MaxD的所述值并重复所述第一循环和所述第二循环,直到所述FT1中的所有节点在所述FT1中都具有多条链路。

[0051] 为表述清楚,在不脱离本发明范围的前提下,任何一个前述实施例都可以与任何一个或多个其它前述实施例结合以创建新的实施例。

[0052] 结合附图和权利要求,通过以下详细描述,本发明的这些及其它特征和优点将变得更加显而易见。

附图说明

[0053] 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。

[0054] 图1示出了本发明实施例提供的用于计算泛洪拓扑(flooding topology,FT)的第一种方法的流程图;

[0055] 图2A-F示出了图1所示方法的迭代的数据结构和网络图;

[0056] 图2G示出了通过图1所示方法计算的FT以及该FT的网络图;

[0057] 图3A和3B分别示出了本发明实施例提供的用于计算FT的第二种方法的第一子过程和第二子过程的流程图;

[0058] 图3C示出了作为图3B所示子过程的替代版本的子过程的流程图;

[0059] 图4A-E示出了图3A和3B所示方法的迭代的数据结构和网络图;

[0060] 图4F示出了通过图3A和3B所示方法计算的FT以及该FT的网络图;

[0061] 图5A和5B分别示出了本发明实施例提供的用于计算FT的第三种方法的第一子过程和第二子过程的流程图;

[0062] 图6A-E示出了图5A和5B所示方法的第一次迭代的数据结构和网络图;

[0063] 图7A-C示出了图5A和5B所示方法的第二次迭代的数据结构和网络图;

[0064] 图7D示出了通过图5A和5B所示方法的第二次迭代计算的FT以及该FT的网络图;

[0065] 图8示出了本发明实施例提供的用于计算FT的第四种方法的流程图;

[0066] 图9示出了图8所示方法的第一部分的操作的数据结构和网络图;

[0067] 图10A-B和11A-B示出了图8所示方法的第二部分的操作的数据结构和网络图;

[0068] 图12示出了通过图8所示方法计算的FT以及该FT的网络图;

[0069] 图13是本发明实施例提供的网元的示意图;

[0070] 图14示出了用于实现本文所述的用于计算FT的一种或多种方法的第一装置;

[0071] 图15示出了用于实现本文所述的用于计算FT的一种或多种方法的第二装置;

[0072] 图16A和16B示出了本发明实施例提供的用于计算FT的第五种方法的流程图。

具体实施方式

[0073] 首先应理解,尽管下文提供一项或多项实施例的说明性实现方式,但所公开的系统 and/或方法可使用任何数量的技术来实施,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现方式、附图和技术,包括本文所说明并描述的示例性设计和实现方式,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。

[0074] 可以为给定的网络拓扑(Network Topology,NT) 构建多个泛洪拓扑。例如,连接NT中所有节点的链是一种泛洪拓扑。连接所有节点的圆是另一种泛洪拓扑。连接所有节点的树是一种泛洪拓扑。此外,树以及树的一些叶子和树的分支节点之间的连接是一种泛洪拓扑。

[0075] 网络的FT本质上是动态的:当网络中的节点和链路的基本拓扑(整个拓扑图)改变时,FT(其是NT的子图)会被重新计算(或重构),以确保在新NT上可达的任何节点在重新计算的FT上也可达。

[0076] 对于某些网络,例如密集数据中心(Data Center,DC)网络,某些链路状态(Link State,LS)泛洪机制效率低下,并且可能存在其它缺点。使用此类LS泛洪机制引起的额外LS泛洪会消耗网络带宽。处理额外LS泛洪,包括接收、缓冲和解码额外LS,会浪费内存空间和

处理器时间。这可能会导致可扩展性问题,并对网络收敛产生不利影响。

[0077] 如上所述,在计算和构建新FT并将新FT从主节点(leader node)泛洪到网络中的每个节点的过程中,LSDB可能不同步并且网络收敛速度会减慢。本发明公开的实施例试图通过提供用于由网络的多个节点同时进行确定性计算FT的各种系统和方法来解决上述问题。

[0078] 本发明公开的实施例根据存储在路由器的LSDB中的NT来计算FT。FT是NT的链路的子集,使得与子网络中的所有节点的可达性与真实网络中的可达性相同,FT的度和FT中的链路数被最小化,并且当FT中的一条链路发生故障时,与所有节点的可达性仍然与真实网络中的可达性相同。

[0079] 对FT求值时,可以考虑几个参数:

[0080] 度:FT上节点的度是FT上该节点与其它节点的连接数。FT的度是FT上节点的度中的最大度。

[0081] 链路数:FT上的链路数是减少LS泛洪量的一个因素。通常,链路数越少,LS泛洪量越小。

[0082] 直径:FT的直径是FT上两个最远节点之间的最短距离。直径是减少网络收敛时间的一个因素。通常,直径越小,收敛时间越短。

[0083] 冗余:FT中的冗余提供对FT上一个或多个链路和/或节点故障的容限。如果FT因某些故障而拆分,则认为它不能容忍这些故障。通常,FT上的链路数越多,FT的冗余程度就越高,而且FT对故障的容忍度也就越高。

[0084] 在本发明的实施例中,在所有节点上运行的过程快速计算相同的FT,并且该FT满足上述条件。本发明提供了四种这样的过程。通过在区域中的每个节点中使用相同的过程来计算泛洪拓扑并且使用计算的FT来泛洪链路状态,减少了网络中的泛洪流量。这种同时计算减少了收敛时间,从而产生更稳定和优化的路由环境。

[0085] 图1示出了本发明实施例提供的用于在广度优先过程中为NT计算FT的第一种方法100的流程图。方法100包括用于计算包括网络中所有节点的FT的第一子过程100a和用于确保FT中的所有节点在FT中都至少具有两条链路的第二子过程100b。

[0086] 在方法100中,FT由节点描述符的有序集合(nodeID,D,PH)表示,其中,nodeID是NT中节点的ID,D是FT中节点的度,PH是FT中该节点所链接的上一跳节点的nodeID。在方法100的描述中,包含FT的集合数据结构称为FT。当方法100通过子过程100a和100b的迭代进行时,通过在FT中添加和修改节点描述符来构建FT。

[0087] 在步骤102中,通过初始化FT和候选队列(candidate queue,Cq)开始向FT添加节点描述符的子过程100a。选择网络中具有最低nodeID的根节点R0,并将FT初始化为值{(R0,0,null)}。值“null”表示R0在FT中没有PH。使用以下步骤初始化Cq。首先,将NT中耦合到R0的所有节点收集在一个集合{Ra,Rb,...Rn}中。集合{Ra,Rb,...Rn}按从最低nodeID到最高nodeID的顺序排序。其次,将{Ra,Rb,...Rn}中每个节点Rx的节点描述符(Rx,0,R0)添加到Cq。因此,Cq被初始化为{(Ra,0,R0),(Rb,0,R0),..., (Rn,0,R0)}。

[0088] 在步骤104中,从Cq中移除Cq中的节点X的第一节点描述符,将X的节点描述符中的值D设置为1,并将X的节点描述符附加到FT的末尾。此外,将X的PH的值D加1。

[0089] 在步骤106中,确定NT中的所有节点是否都包括在FT中。如果是,则方法100继续执

行子过程100b的步骤114。如果NT中的所有节点都不包括在FT中，则方法100继续执行步骤108。

[0090] 在步骤108中，确定C_q是否为空。如果是，则方法100在未连接到NT的其它节点的节点“孤岛”上运行；在步骤110中，方法100将报告存在节点孤岛的消息发送至网络管理员和网络中的路由器组件，并且方法100结束。如果C_q不为空，则方法100继续执行步骤112。

[0091] 在步骤112中，将在NT中耦合到X（步骤104中）且不在FT中的所有节点收集在集合{R_a, R_b, ... R_n}中。集合{R_a, R_b, ... R_n}按从最低nodeID到最高nodeID的顺序排序。其次，将{R_a, R_b, ... R_n}中每个节点R_x的节点描述符(R_x, 0, X)附加到C_q的末尾。然后，方法100进入步骤104。

[0092] 在方法100中，选择R₀作为具有最低nodeID的网络节点，在步骤102中，通过从最低nodeID到最高nodeID排序的节点初始化C_q；在步骤112中，添加到C_q的节点在添加到C_q之前按从最低nodeID到最高nodeID的顺序排序。在其它实施例中，选择R₀作为具有最高nodeID的网络节点，在步骤102中，通过从最高nodeID到最低nodeID排序的节点初始化C_q；在步骤112中，添加到C_q的节点在添加到C_q之前按从最高nodeID到最低nodeID的顺序排序。在其它实施例中，可以使用用于在添加到C_q之前选择R₀和排序节点的其它合适条件来提供用于在方法100中计算FT的广度优先过程。

[0093] 在步骤114中，用于确保FT中的所有节点在FT中都至少具有两条链路的子过程100b开始。在步骤114中，如果两个或多个节点具有D=1，则找到FT中具有D=1和最低nodeID的第一节点(Q)。如果确定FT中不存在此类节点Q，则FT是完整FT；在步骤116中，方法100将FT发送到网络中的路由器组件，并且方法100结束。

[0094] 如果在步骤114中确定存在具有D=1的节点Q，则在步骤118中，将在NT中耦合到Q的所有节点收集在集合{R_a, R_b, ... R_n}中。接下来，如果两个或多个节点具有相同的最低D值，则在{R_a, R_b, ... R_n}中找到具有最低D值和最低nodeID的节点(L)，并且在FT中，节点Q和L的D值都递增1。最后，通过将节点描述符(Q, d, L)添加到FT，在FT中添加Q与L之间的链路，其中，d等于节点Q的递增D值，并且方法100进入步骤114。

[0095] 图2A-F示出了图1所示方法100的迭代的数据结构和网络图。在图2A中，网络图200a以图形方式表示在通过图1所示子过程100a的步骤进行第一次迭代之后NT和FT的状态。虚线202表示NT中节点R₃到R₆之间的链路。其它类似虚线表示NT中的节点之间的其它链路。粗线204表示已将R₁添加到FT，其中R₀作为其PH。数据结构210显示迭代以C_q = {(R₁, 0, R₀), (R₂, 0, R₀), (R₃, 0, R₀)}开始。在步骤104中，从C_q中移除R₁的节点描述符(或简称为R₁)并将其添加到FT。网络中的所有节点都不在FT中且C_q不为空，因此在步骤112中，将节点R₉添加到C_q(表示为C_q')，并且方法100进入步骤104以进行第二次迭代。

[0096] 图2B示出了子过程100a的步骤的第二次迭代的结果。在网络图200b中，实线206表示R₁链接到FT中的R₀，粗线表示已将R₂添加到FT，其中R₀作为其PH。数据结构显示已从C_q中移除R₂并将其添加到FT，并且已将R₄和R₅添加到C_q'。

[0097] 图2C示出了子过程100a的步骤的第九次(最后一次)迭代的结果。在网络图200i中，实线表示在先前的迭代中已将节点R₁-R₇和R₉添加到FT，粗线表示在该迭代中已将R₈添加到FT，其中R₉作为其PH。数据结构显示已从C_q中移除R₈并将其添加到FT。步骤106确定所有节点R₀-R₉都在FT中，方法100继续执行步骤114。

[0098] 图2D示出了通过子过程100b的步骤进行的第一次迭代的结果。数据结构220显示,在步骤114中,确定节点R4在FT中具有 $D=1$ 和最低nodeID;在步骤118中,确定R5是耦合到R4且具有最低D值和最低nodeID的节点,并且R4和R5的D值都递增1(数据结构FT'),并将R4与R5之间的链路的节点描述符(R4,2,R5)添加到FT'。网络图200j显示了由此在FT中建立的节点R4与R5之间的链路。

[0099] 图2E示出了通过子过程100b的步骤进行的第二次迭代的结果。在步骤114中,确定节点R6具有 $D=1$ 和最低nodeID;在步骤118中,确定R7是耦合到R6且具有最低D值和最低nodeID的节点,并且R6和R7的D值都递增1,并将R6与R7之间的链路的节点描述符(R6,2,R7)添加到FT'。网络图200k显示了由此在FT中建立的节点R6与R7之间的链路。

[0100] 图2F示出了通过子过程100b的步骤进行的第三次迭代的结果。在步骤114中,确定节点R8具有 $D=1$;在步骤118中,确定R4是耦合到R8且具有最低D值和最低nodeID的节点,并且R4和R8的D值都递增1,并将R8与R4之间的链路的节点描述符(R8,2,R4)添加到FT'。网络图200l显示了由此在FT中建立的节点R8与R4之间的链路。

[0101] 在步骤114的下一迭代中,未在FT中找到具有 $D=1$ 的节点;在步骤116中,方法100将FT发送到网络中的路由器组件并终止。图2G示出了通过图1所示方法100计算的FT 230以及FT 230的网络图240。

[0102] 图3A和3B分别示出了本发明实施例提供的用于计算FT的第二种方法300的第一子过程和第二子过程的流程图。方法300包括用于计算包括网络中所有节点的FT的子过程300a(如图3A所示)和用于确保FT中的所有节点在FT中都至少具有两条链路的子过程300b(如图3B所示)。图3C示出了作为图3B所示子过程300b的替代版本的子过程300c的流程图。

[0103] 在方法300中,FT由节点描述符的有序集合(nodeID,D,PH)表示,其中,nodeID是NT中节点的ID,D是FT中节点的度,PH是FT中该节点所链接的一个或多个上一跳的nodeID的集合。在方法300的描述中,包含FT的集合数据结构称为FT。当方法300通过子过程300a和300b的迭代进行时,通过在FT中添加和修改节点描述符来构建FT。

[0104] 在步骤302中,选择FT中任何节点的度的上限,并将变量MaxD(最大度)的值初始化为该上限。在若干实施例中,方法300计算具有最小化MaxD的FT。在本发明的一些实施例中,MaxD被设置为较小的数字,并运行方法300。在图3A和3B所示的实施例中,如果子过程300a或子过程300b在FT中没有节点超过MaxD的情况下无法完成,则方法300将MaxD递增1并重新开始,重复此类操作,直到找到使方法300成功完成的MaxD值。

[0105] 在步骤304中,选择网络中具有最低nodeID的根节点R0,并将FT初始化为{(R0,0,{})}。使用以下步骤初始化Cq。首先,将NT中耦合到R0的所有节点收集在一个集合{Ra,Rb,...Rn}中。集合{Ra,Rb,...Rn}按从最低nodeID到最高nodeID的顺序排序。其次,将{Ra,Rb,...Rn}中每个节点Rx的节点描述符(Rx,0,{R0})添加到Cq。因此,Cq被初始化为{(Ra,0,{R0}),(Rb,0,{R0}),..., (Rn,0,{R0})}。

[0106] 在步骤306中,找到Cq中的第一节点描述符(例如,简称为节点X)在其PH中具有D小于MaxD的节点(例如,简称为节点Y)。如果找到此类节点,则方法300继续执行步骤310。如果未找到此类节点,则方法300继续执行步骤308,其中MaxD递增1,并且方法300进入步骤304以使用较高的MaxD值重新开始。

[0107] 在一些实施例中,存储在Cq中的节点描述符包括第四个值W,其指示已在步骤306

中传递节点,因为PH中没有D小于MaxD的节点。尚未在步骤306中求值的节点具有 $W=0$ 。在这些实施例中,当新的上一跳被添加到节点的PH时(如下所述),在步骤318中,将 C_q 中的节点的值 W 重置为0。这些实施例可以比图3A和3B所示的实施例更有效地运行。

[0108] 在步骤310中,从 C_q 中移除节点描述符 X ,将 X 中的值 D 设置为1,并且将 X 的PH减小到仅包括在步骤306中找到的节点 Y 的nodeID。进行此类修改后,将 X 添加到FT的末尾。此外,将节点 Y 的值 D 加1。

[0109] 在步骤312中,确定NT中的所有节点是否都包括在FT中。如果是,则子过程300a已经成功完成并且方法300继续执行子过程300b的步骤320(如图3B所示)。如果NT中的所有节点都不包括在FT中,则方法300继续执行步骤314。

[0110] 在步骤314中,确定 C_q 是否为空。如果是,则方法300在未连接到NT的其它节点的节点“孤岛”上运行;在步骤316中,方法300将报告存在节点孤岛的消息发送至网络管理员和网络中的路由器组件,并且方法300结束。如果 C_q 不为空,则方法300继续执行步骤318。

[0111] 在步骤318中,修改 C_q ,以响应将 X 添加(在步骤310中描述)到FT。将在NT中耦合到 X 且不在FT中的所有节点收集在按从最低nodeID到最高nodeID排序的队列 $\{R_a, R_b, \dots R_n\}$ 中。对于 $\{R_a, R_b, \dots R_n\}$ 中的每个节点 Z ,如果 Z 在 C_q 中,则将 X 添加到 Z 的PH的末尾。在其它实施例中, Z 的PH中的节点保持所需顺序,例如, Z 的PH中的节点可以按其 D 值从最低值到最高值来排序。在这些实施例中,如果 Z 的PH中的两个或多个节点具有相等的 D 值,则它们在 Z 的PH中按从最低nodeID到最高nodeID的顺序排序。

[0112] 如果 Z 不在 C_q 中,则将节点描述符 $(Z, 0, \{X\})$ 附加到 C_q 的末尾。然后,方法300进入步骤306。在 C_q 中存储的节点描述符包括第四个值 W 的实施例中,在进入步骤306之前,在步骤318中添加到 C_q 的新节点描述符被赋予 $W=0$ 的初始值,并且对于 X 已添加到其PH的任何节点描述符, W 被重置为0。

[0113] 如针对方法100所描述的,在方法300的其它实施例中,选择 R_0 作为具有最高nodeID的网络节点;在步骤304中,通过从最高nodeID到最低nodeID排序的节点初始化 C_q ;在步骤318中,添加到 C_q 的节点在添加到 C_q 之前按从最高nodeID到最低nodeID的顺序排序。在其它实施例中,可以使用用于在添加到 C_q 之前选择 R_0 和排序节点的其它合适条件来提供用于在方法300中计算FT的广度优先过程。

[0114] 在步骤320中,用于确保FT中的所有节点在FT中都至少具有两条链路的子过程300b开始。在步骤320中,如果两个或多个节点具有 $D=1$,则找到FT中具有 $D=1$ 和最低nodeID的第一节点(Q)。如果确定FT中不存在此类节点 Q ,则子过程300b已经成功完成并且FT是完整FT。在步骤322中,方法300将FT发送到网络中的路由器组件,并且方法300结束。

[0115] 如果在步骤320中确定存在具有 $D=1$ 的节点 Q ,则在步骤324中,将在NT中耦合到 Q 的所有节点收集在集合 $\{R_a, R_b, \dots R_n\}$ 中。同样,在步骤324中,在 $\{R_a, R_b, \dots R_n\}$ 中找到节点(L),其中, L 具有(i)最低 D 值,(ii)最低nodeID(如果两个或多个节点具有相同的最低 D 值),且(iii) $D < \text{MaxD}$ 。如果未找到节点 L ,则方法300进入步骤308以递增 MaxD 并重新开始。如果找到节点 L ,则在步骤326中,通过将FT中的节点 Q 和 L 的 D 值递增1并将节点 L 添加到节点 Q 的PH,来将节点 L 链接到节点 Q 。然后,方法300进入步骤320。

[0116] 图3C示出了作为图3B所示子过程300b的替代版本的子过程300c的流程图。在本发明的另一实施例中,方法300包括用于计算包括网络中所有节点的FT的子过程300a(如结合

图3A所述)和用于确保FT中的所有节点在FT中都至少具有两条链路的子过程300c。子过程300c从步骤330开始,该步骤与针对子过程300b描述的步骤320相同。如果在步骤330中确定FT中没有节点具有 $D=1$,则子过程300c已经成功完成,并且步骤330进入步骤332。完成子过程300c后,FT可以包括一些具有 D 大于 $MaxD$ 的节点。在步骤332中,如果在FT中找到此类节点,则将 $MaxD$ 的值增加到在FT中找到的最大 D 值。对 $MaxD$ 的值进行此类更改(如有)之后,在步骤332中,方法300将FT发送到网络中的路由器组件,并且方法300结束。

[0117] 如果在步骤330中确定存在具有 $D=1$ 的节点 Q ,则在步骤334中,将在NT中耦合到 Q 的所有节点收集在集合 $\{Ra, Rb, \dots, Rn\}$ 中。同样,在步骤334中,在 $\{Ra, Rb, \dots, Rn\}$ 中找到节点 L ,其中, L 具有(i)最低 D 值和(ii)最低nodeID(如果两个或多个节点具有相同的最低 D 值)。然后通过将FT中的节点 Q 和 L 的 D 值递增1并将节点 L 添加到节点 Q 的PH,来将节点 L 链接到节点 Q 。然后,方法300进入步骤330。

[0118] 图4A-E示出了图3A和3B所示方法的迭代的数据结构和网络图。在图4A-E中, $MaxD=3$ 。

[0119] 在图4A中,网络图400a以图形方式表示在通过图3A所示子过程300a的步骤进行第一次迭代之后NT和FT的状态。虚线402表示NT中节点 $R3$ 到 $R4$ 之间的链路。其它类似虚线表示NT中的节点之间的其它链路。粗线404表示已将 $R1$ 添加到FT,其中 $R0$ 作为其PH。数据结构410显示迭代以 $Cq = \{(R1, 0, \{R0\}), (R2, 0, \{R0\}), (R3, 0, \{R0\}), (R4, 0, \{R0\})\}$ 开始。在步骤310中,从 Cq 中移除 $R1$ 并将其添加到FT。网络中的所有节点都不在FT中且 Cq 不为空,因此在步骤318中,确定链接到 $R1$ 并且不在FT中的节点是 $R2$ 、 $R3$ 和 $R4$,因此将 $R1$ 添加到 Cq 中的节点 $R2$ 、 $R3$ 和 $R4$ 中的每一个的PH。步骤318的结果显示在 Cq' 中。然后,方法300进入步骤306以进行第二次迭代。

[0120] 图4B示出了子过程300a的步骤的第二次迭代的结果。在网络图400b中,实线406表示 $R1$ 链接到FT中的 $R0$,粗线表示已将 $R2$ 添加到FT,其中 $R0$ 作为其PH。数据结构显示已从 Cq 中移除 $R2$,已将 $R2$ 与 $R0$ 之间的链路添加到FT,并且已将 $R2$ 添加到 Cq' 中的 $R3$ 和 $R4$ 中的每一个的PH。

[0121] 图4C示出了子过程300a的步骤的第四次(最后一次)迭代的结果。在网络图400d中,实线表示在先前的迭代中已将节点 $R1$ - $R3$ 添加到FT,粗线表示在该迭代中已将 $R4$ 和 $R4$ 与 $R1$ 之间的链路添加到FT,其中 $R1$ 作为其PH。数据结构显示已从 Cq 中移除 $R4$,并且已将 $R4$ 与 $R1$ 之间的链路添加到FT。步骤312确定所有节点 $R0$ - $R4$ 都在FT中,方法300继续执行步骤320。

[0122] 图4D示出了通过图3B所示子过程300b的步骤进行的第一次迭代的结果。数据结构420显示,在步骤320中,即使 $R3$ 和 $R4$ 中的每一个都具有 $D=1$,也确定节点 $R2$ 具有 $D=1$ (数据结构FT)和最低nodeID;在步骤326中,确定 $R3$ 是耦合到 $R2$ 并具有最低 D 值、最低nodeID且 $D < MaxD$ 的节点。在步骤328中,将 $R2$ 和 $R3$ 的 D 值递增1(数据结构FT'),并将 $R3$ 添加到 $R2$ 的PH中。网络图400e显示了由此在FT中建立的节点 $R2$ 与 $R3$ 之间的链路。

[0123] 图4E示出了通过子过程300b的步骤进行的第二次迭代的结果。在步骤320中,确定节点 $R4$ 具有 $D=1$;在步骤326中,确定 $R2$ 是耦合到 $R4$ 并具有最低 D 值、最低nodeID且 $D < MaxD$ 的节点。在步骤328中,将 $R2$ 和 $R4$ 的 D 值递增1,并将 $R2$ 添加到 $R4$ 的PH中。网络图400f显示了由此在FT中建立的节点 $R2$ 与 $R4$ 之间的链路。

[0124] 在步骤320的下一迭代中,未在FT中找到具有 $D=1$ 的节点;在步骤322中,方法

300将FT发送到网络中的路由器组件并终止。图4F示出了通过图3A和3B所示方法300计算的FT 430以及FT 430的网络图440。

[0125] 图5A和5B分别示出了本发明实施例提供的用于计算FT的第三种方法500的第一子过程和第二子过程的流程图。方法500包括用于计算包括网络中所有节点的FT的第一子过程500a(如图5A所示)和用于确保FT中的所有节点在FT中都至少具有两条链路的第二子过程500b(如图5B所示)。

[0126] 在方法500中,FT由节点描述符的有序集合 $(nodeID, D, PH)$ 表示,其中, $nodeID$ 是NT中节点的ID, D 是FT中节点的度, PH 是FT中该节点所链接的一个或多个上一跳的 $nodeID$ 的集合。在方法500的描述中,包含FT的集合数据结构称为FT。当方法500通过子过程500a和500b的迭代进行时,通过在FT中添加和修改节点描述符来构建FT。

[0127] 在步骤502中,初始化变量 $MaxD$ (最大度),该变量表示FT中NT的某些节点的度的上限。同样,在步骤502中,获取(最初从网络管理员处获取)存储信息,该信息用于识别在FT中对其最大度具有约束的NT的节点子集。将值 $ConMaxD$ (受约束最大度) 设置为该约束值。 $MaxD$ 是方法500尝试对不在子集中的节点最小化的值。不在NT的节点子集中的节点在本文中可称为第一类型的节点。在NT的节点子集中的节点在本文中可称为第二类型的节点。 $ConMaxD$ 是由于网络原因方法500不能超过的值。例如,叶脊拓扑网络中的叶子节点可被赋值为 $ConMaxD=2$ 。

[0128] 在若干实施例中,方法500计算具有受约束且最小化 $MaxD$ 的FT。在本发明的一些实施例中, $MaxD$ 被设置为较小的数字,并运行方法500。如果子过程500a或子过程500b在FT中没有节点超过 $MaxD$ 的情况下无法完成,则方法500将 $MaxD$ 递增1并重新开始,重复此类操作,直到找到使方法500成功完成的 $MaxD$ 值。

[0129] 在其它实施例中,一旦方法500成功完成,便存储 $MaxD$ 的成功值,并且如果需要改变NT并重新计算FT,则方法500通过将 $MaxD$ 初始化为 $MaxD$ 的存储值来开始。在一些此类实施例中,如果方法500成功完成,则方法500返回到步骤502并连续将 $MaxD$ 初始化为连续较小的值,直到方法500最终未能完成,而不是结束于步骤522(如下所述)。在这些实施例中,一旦方法500未能完成,便存储紧接在前面的 $MaxD$ 值,并且使用先前的 $MaxD$ 计算的FT被用作针对网络计算的FT。

[0130] 在步骤504中,选择网络中具有最低 $nodeID$ 的根节点 $R0$,并将FT初始化为 $\{(R0, 0, \{\})\}$ 。使用以下步骤初始化 Cq 。首先,将NT中耦合到 $R0$ 的所有节点收集在一个集合 $\{Ra, Rb, \dots, Rn\}$ 中。集合 $\{Ra, Rb, \dots, Rn\}$ 按从最低 $nodeID$ 到最高 $nodeID$ 的顺序排序。其次,将 $\{Ra, Rb, \dots, Rn\}$ 中每个节点 Rx 的节点描述符 $(Rx, 0, \{R0\})$ 添加到 Cq 。因此, Cq 被初始化为 $\{(Ra, 0, \{R0\}), (Rb, 0, \{R0\}), \dots, (Rn, 0, \{R0\})\}$ 。

[0131] 在步骤506中,找到有资格包含在FT中的 Cq 中的第一节点(例如,简称为节点X)。如果节点是第一类型的网络节点并且在其 D 小于 $MaxD$ 的 PH 中具有节点(例如,简称为节点Y),或者节点是第二类型的网络节点并且在其 D 小于 $ConMaxD$ 的 PH 中具有节点Y,则节点X有资格包含在FT中。如果确定 Cq 包括节点X,则方法500继续执行步骤510。如果确定 Cq 不包括节点X,则方法500继续执行步骤508,其中 $MaxD$ 递增1,并且方法500进入步骤504以使用较高的 $MaxD$ 值重新开始。

[0132] 在一些实施例中,存储在 Cq 中的节点描述符包括第四个值 W ,其指示已在步骤506

中对节点求值并传递该节点,因为该节点不是合格节点。尚未在步骤506中求值的节点具有 $W=0$ 。已求值并传递的节点具有 $W=1$ 。在这些实施例中,在步骤518(如下所述)中,将具有添加到其 C_q 中的PH的新的上一跳的节点的值 W 重置为0。这些实施例可以比图5A和5B所示的实施例更有效地运行。

[0133] 在步骤510中,从 C_q 中移除节点 X ,将 X 的节点描述符中的值 D 设置为1,并且将 X 的PH减小到仅包括在步骤506中找到的节点 Y 的nodeID。进行此类修改后,将 X 的节点描述符添加到FT的末尾。此外,将 X 的PH中唯一节点的值 D 加1。

[0134] 在步骤512中,确定NT中的所有节点是否都包括在FT中。如果确定NT中的所有节点都包括在FT中,则方法500继续执行子过程500b的步骤520(如图5B所示)。如果确定NT中的所有节点都不包括在FT中,则方法500继续执行步骤514。

[0135] 在步骤514中,确定 C_q 是否为空。如果是,则方法500在未连接到NT的其它节点的节点“孤岛”上运行;在步骤516中,方法500将报告存在节点孤岛的消息发送至网络管理员和网络中的路由器组件,并且方法500结束。如果 C_q 不为空,则方法500继续执行步骤518。

[0136] 在步骤518中,将在NT中耦合到 X (在步骤506中描述)且不在FT中的所有节点收集在集合 $\{R_a, R_b, \dots, R_n\}$ 中。集合 $\{R_a, R_b, \dots, R_n\}$ 按从最低nodeID到最高nodeID的顺序排序。对于 $\{R_a, R_b, \dots, R_n\}$ 中的每个节点 Z ,如果 Z 在 C_q 中,则将 X 添加到 Z 的PH。如果 Z 不在 C_q 中,则将节点描述符 $(Z, 0, \{X\})$ 附加到 C_q 的末尾。然后,方法500进入步骤506。在 C_q 中存储的节点描述符包括第四个值 W 的实施例中,在进入步骤506之前,在步骤518中添加到 C_q 的新节点描述符被赋予 $W=0$ 的初始值,并且对于 C_q 中具有 $W=1$ 且 X 已添加到其PH的任何节点描述符, W 被重置为0。

[0137] 如针对方法100和方法300所描述的,在方法500的其它实施例中,选择 R_0 作为具有最高nodeID的网络节点;在步骤504中,通过从最高nodeID到最低nodeID排序的节点初始化 C_q ;在步骤518中,添加到 C_q 的节点在添加到 C_q 之前按从最高nodeID到最低nodeID的顺序排序。在其它实施例中,可以使用用于在添加到 C_q 之前选择 R_0 和排序节点的其它合适条件来提供用于在方法500中计算FT的广度优先过程。

[0138] 在步骤520中,用于确保FT中的所有节点在FT中都至少具有两条链路的子过程500b开始。在步骤520中,如果两个或多个节点具有 $D=1$,则找到FT中具有 $D=1$ 和最低nodeID的第一节点(Q)。如果确定FT中不存在节点 Q ,则FT是完整FT;在步骤522中,方法500将FT发送到网络中的路由器组件,并且方法500结束。

[0139] 如果在步骤520中确定存在具有 $D=1$ 的节点 Q ,则在步骤524中,将在NT中耦合到 Q 的所有节点收集在集合 $\{R_a, R_b, \dots, R_n\}$ 中。在步骤526中,在 $\{R_a, R_b, \dots, R_n\}$ 中找到节点(L),其中, L 具有(i)最低 D 值,(ii)最低nodeID(如果两个或多个节点具有相同的最低 D 值),并且(iii)如果 L 是第一类型的网络节点且其 D 小于 $MaxD$,或者 L 是第二类型的网络节点且其 D 小于 $ConMaxD$ 。如果未找到此类节点 L ,则方法500进入步骤508以递增 $MaxD$ 并重新开始。如果找到节点 L ,则在步骤528中,在FT中将节点 Q 和 L 的 D 值递增1,通过将 L 加到 Q 的PH,来在FT中添加 Q 与 L 之间的链路,并且方法500进入步骤520。

[0140] 图6A-E示出了图5A和5B所示方法500的第一次迭代的数据结构和网络图。在图5A和5B所示方法500的第一次迭代中, $MaxD$ 的值等于3。在图6A-E中,约束 $ConMaxD$ 设置为等于2。

[0141] 在图6A中,网络图600a以图形方式表示在通过图5A所示子过程500a的步骤进行第一次迭代之后NT和FT的状态。虚线602表示NT中节点R1到R5之间的链路。其它类似虚线表示NT中的节点之间的其它链路。粗线604表示已将R2添加到FT,其中R0作为其PH。数据结构610显示迭代以 $C_q = \{(R2, 0, \{R0\}), (R3, 0, \{R0\}), (R4, 0, \{R0\}), (R5, 0, \{R0\})\}$ 开始。在步骤510中,从 C_q 中移除R2并将其添加到FT。网络中的所有节点都不在FT中且 C_q 现在不为空,因此在步骤518中,确定节点R1和R6-R10链接到R2并且不在FT中,因此将PH为 $\{R2\}$ 的R1和R6-R10的节点描述符添加到 C_q 。由于R2未链接到R3-R5中的任何一个,因此不会将R2添加到 C_q 中的节点R3-R5的PH。步骤518的结果显示在数据结构 C_q' 中。然后,方法500进入步骤506以进行第二次迭代。

[0142] 图6B示出了子过程500a的步骤的第二次迭代的结果。在网络图600b中,实线606表示R2链接到FT中的R0,粗线表示已将R3添加到FT,其中R0作为其PH。数据结构显示已从 C_q 中移除R3并将其添加到FT,并且已将R3添加到 C_q' 中的R1和R6-R10中的每一个的PH。

[0143] 图6C示出了子过程500a的步骤的第十次(最后一次)迭代的结果。在网络图600j中,实线表示在先前的迭代中已将节点R1-R9添加到FT,粗线表示在该迭代中已将R10添加到FT,其中R4作为其PH。数据结构显示已从 C_q 中移除R10并将其添加到FT。步骤512确定所有节点R0-R10都在FT中,方法500继续执行步骤520。

[0144] 图6D示出了通过图5B所示子过程500b的步骤进行的第一次迭代的结果。数据结构620显示,在步骤520中,确定节点R5具有 $D=1$ (数据结构FT)和最低nodeID;在步骤526中,确定R6是耦合到R5并具有最低D值、最低nodeID、 $D < \text{MaxD}$ 并且 $D < \text{ConMaxD}$ 的节点。在步骤528中,将R5和R6的D值递增1(数据结构FT'),并通过将R6添加到R5的PH,来将R5与R6之间的链路添加到FT'中。网络图600k显示了由此在FT中建立的节点R5与R6之间的链路。

[0145] 图6E示出了通过子过程500b的步骤进行的第三次迭代的结果的网络图600m。在步骤520中,确定节点R8具有 $D=1$ 和最低nodeID,但是在步骤526中,确定不存在耦合到R8的合格节点,因为耦合到R8的节点R2、R4和R5中的每一个已经具有 $D = \text{MaxD}$ 。因此,方法500进入步骤508以递增 MaxD 的值,并通过步骤504重新开始方法500。

[0146] 图7A-C示出了图5A和5B所示方法500的第二次迭代的数据结构和网络图。在图5A和5B所示方法500的第二次迭代中, MaxD 的值等于4。在图7A-C中,约束 ConMaxD 仍然等于2。

[0147] 图7A示出了子过程500a的步骤的第十次(最后一次)迭代($\text{MaxD}=4$)的结果。在网络图700j中,实线表示在先前的迭代中已将节点R1-R9添加到FT(有些具有不同于图6C中的PH),粗线表示在该迭代中已将R10添加到FT,其中R3作为其PH。数据结构710显示已从 C_q 中移除R10并将其添加到FT。步骤512确定所有节点R0-R10都在FT中,方法500继续执行步骤520。

[0148] 图7B示出了通过图5B所示子过程500b的步骤进行的第一次迭代($\text{MaxD}=4$)的结果。数据结构720显示,在步骤520中,确定节点R1具有 $D=1$ (数据结构FT)和最低nodeID(R4至R10中的每一个具有 $D=1$ 但nodeID比R1高);在步骤526中,确定R4是耦合到R1并具有最低D值、最低nodeID(耦合到R1的R5具有 $D=1$ 但nodeID比R4高)且 $D < \text{MaxD}$ 的节点。在步骤528中,将R1和R4的D值递增1(数据结构FT'),并通过将R4添加到R1的PH,来将R1与R4之间的链路添加到FT'中。网络图700k显示了由此在FT中建立的节点R1与R4之间的链路。

[0149] 图7C示出了通过子过程500b的步骤进行的第六次迭代($\text{MaxD}=4$)的结果。第二次

到第五次迭代为节点R5-R9添加了冗余链路。在该迭代中,在步骤520中,确定节点R10具有 $D=1$;在步骤526中,确定R5是耦合到R10并在FT中具有最低D值且 $D<MaxD$ 的节点。在步骤528中,将R5和R10的D值递增1(数据结构FT'),通过将R5添加到R10的PH,来将R10与R5之间的链路添加到FT'。网络图700p显示了由此在FT中建立的节点R10与R5之间的链路。

[0150] 在步骤520的下一迭代中,未在FT中找到具有 $D=1$ 的节点;在步骤522中,方法500将FT发送到网络中的路由器组件并终止。图7D示出了通过图5A和5B所示方法500($MaxD$ 等于4)计算的FT 730以及该FT的网络图740。

[0151] 图8示出了本发明实施例提供的用于计算FT的第四种方法800的流程图。当NT包括确定性节点时,使用第四种方法800。如果一个节点的D被约束为 $ConMaxD$,并且其在NT中的链路数等于 $ConMaxD$,则该节点是确定性节点。

[0152] 方法800的子过程800a通过首先计算FT0来计算泛洪拓扑,FT0包括NT中的所有确定性节点。一旦在子过程800a中将所有确定性节点添加到FT0,方法800的子过程800b便计算FT1,FT1包括NT中不在FT0中的所有节点。最后,方法800通过将FT0附加到FT1的末尾来计算完整FT。

[0153] 在步骤802中,获取(最初从网络管理员处获取)存储信息,该信息用于识别在FT中对其最大度具有约束的NT的节点子集。将值 $ConMaxD$ (受约束最大度)设置为该约束值。同样,在步骤802中,将 $Cq0$ 初始化为集合 $\{(R0, 0, \{ \}), (R1, 0, \{ \}), \dots (Rn, 0, \{ \})\}$,其中 $R0$ 、 $R1$ 、 \dots 、 Rn 是NT中的所有节点。在步骤804中,找到 $Cq0$ 中的第一确定性节点(X),其中连接到节点X的链路数等于 $ConMaxD$,并且节点X在子集中。如果未找到此类节点,则方法800继续执行步骤810。如果找到节点X,则方法800继续执行步骤806。

[0154] 在步骤806中,从 $Cq0$ 中移除X的节点描述符,并将节点 Ra 、 Rb 、 \dots 、 Rm 添加到X的PH,其中 Ra 、 Rb 、 \dots 、 Rm 是NT中链接到X的节点。最后,将X的节点描述符添加到FT0的末尾。在步骤808中,将节点 Ra 、 Rb 、 \dots 、 Rm 中的每个节点的 $Cq0$ 中的D值递增1,并且方法800进入步骤804。

[0155] 在步骤810中,通过从NT中移除FT0中的所有节点及其到其它节点的链路,来创建临时网络拓扑NT1。在步骤812中,确定NT1中的任何节点是否被限制为 $ConMaxD$ 。如果确定NT1中没有限制为 $ConMaxD$ 的节点,则方法800继续执行步骤814。如果确定NT1中确实存在限制为 $ConMaxD$ 的节点,则方法800继续执行步骤816。

[0156] 在步骤814中,方法800对NT1执行方法300以计算FT1。在方法300的步骤302中,将 $MaxD$ 初始化为1加上 $Cq0$ 中D的最高值。在方法300的步骤304中,如方法300的描述中所讨论的,从NT1中选择根节点 $R0$ 作为具有最低或最高nodeID的节点。在方法300的步骤310中将节点添加到FT时,将添加的节点的初始D值设置为等于 $Cq0$ 中节点的D值。在方法300的步骤322中,将通过方法300计算的FT1返回到方法800的步骤814,而不是将FT1发送到网络中的路由器组件。

[0157] 在步骤816中,方法800对NT1执行方法500以计算FT1。在方法500的步骤502中,将 $MaxD$ 初始化为1加上 $Cq0$ 中D的最高值,并且对于NT1中受 $ConMaxD$ 限制的任何节点, $ConMaxD$ 保持不变。在方法500的步骤504中,如方法500的描述中所讨论的,从NT1中选择根节点 $R0$ 作为具有最低或最高nodeID的节点。在方法500的步骤510中将节点添加到FT时,将添加的节点的初始D值设置为等于 $Cq0$ 中节点的D值。在方法500的步骤522中,将通过方法500计算的

FT1返回到方法800的步骤816,而不是将FT1发送到网络中的路由器组件。

[0158] 在步骤818中,将FT0附加到FT1的末尾以形成NT的完整FT。在步骤820中,方法800将完整FT发送到网络中的路由器组件,并且方法800结束。

[0159] 图9示出了图8所示方法800的第一部分的操作的数据结构910和网络图900。图9示出了子过程800a在网络900上的操作示例。在步骤802中,将Cq0初始化为 $\{(R0,0,\{\}), (R1,0,\{\}), (R2,0,\{\}), (R3,0,\{\}), (R4,0,\{\}), (R5,0,\{\}), (R6,0,\{\})\}$ 。节点R4-R6的D限制为ConMaxD,其中ConMaxD等于2。因为节点R4-R6中的每一个都只有两条链路,因此它们是确定性节点。

[0160] 在步骤804、806和808的三次迭代之后,节点R4、R5和R6已从Cq0中移除并添加到FT0,其现在等于 $\{(R4,2,\{R2,R3\}), (R5,2,\{R2,R3\}), (R6,2,\{R2,R3\})\}$ 。当步骤804在Cq0中找不到剩余确定性节点时,Cq0的最终值为 $\{(R0,0,\{\}), (R1,0,\{\}), (R2,3,\{\}), (R3,3,\{\})\}$ (数据结构Cq0')。请注意,节点R2和R3中的每一个在Cq0中具有D=3,因为每个节点在FT0中都链接到所有三个节点R4、R5和R6。

[0161] 图10A-B和11A-B示出了图8所示方法800的第二部分的操作的数据结构和网络图。在步骤810中,从NT中移除节点R4、R5和R6及其与其它节点的链路以创建NT1,这在图10A中显示为网络图1000a。步骤812确定NT1中的剩余节点(R0-R3)都未限制为ConMaxD,因此方法800继续执行步骤814,对NT1执行方法300以计算FT1。图10A的数据结构1002指示,由于Cq0中D的最高值是3,因此在方法300的步骤302中将MaxD初始化为MaxD=4。此外,在方法300的步骤310中将节点R0-R3添加到FT时,R0和R1将具有初始值D=1,R2和R3将具有初始值D=3。

[0162] 图10B在数据结构1020和网络图1000c中示出了执行方法300的子过程300a的结果是 $FT1 = \{(R0,3,\{\}), (R1,1,\{R0\}), (R2,4,\{R0\}), (R3,4,\{R0\})\}$ 。然而,在第一次通过子过程300b时,节点R1被识别为具有D=1,但无法在FT1中链接到R2或R3,因为它们都具有D=MaxD,如数据结构1020所示。

[0163] 因此,方法300进入步骤308,将MaxD递增到值5,并重复子过程300a。结果与图10B所示的FT 1000c相同。然而,图11A在数据结构1120和网络图1100a中示出方法300的步骤328现在可以将R1链接到R2,因为FT1中R2的D值小于MaxD。图11B在数据结构1130中示出结果为 $FT1 = \{(R0,3,\{\}), (R1,2,\{R0,R2\}), (R2,5,\{R0\}), (R3,4,\{R0\})\}$,如网络图1140所示。将通过方法300计算的FT1返回到方法800的步骤818。

[0164] 图12示出了通过图8所示方法800计算的完整FT 1210以及该完整FT的网络图1200。数据结构1210显示NT的完整FT为 $FT = \{(R0,3,\{\}), (R1,2,\{R0,R2\}), (R2,5,\{R0\}), (R3,4,\{R0\}), (R4,2,\{R2,R3\}), (R5,2,\{R2,R3\}), (R6,2,\{R2,R3\})\}$,如网络图1200所示,其中包括NT中的所有节点和链路。

[0165] 图13示出了本发明实施例提供的网元1300的示意图。网元1300可以是世界各地的服务提供商使用的任何网络节点,例如但不限于,任何路由器、交换机和控制器。网元1300也可以称为网络设备。网元1300包括用于通过入端口1310接收数据的接收器单元(receiver unit,RX) 1320或接收装置。网元1300还包括用于通过出端口1350发送数据的发送器单元(transmitter unit,TX) 1340或发送装置。

[0166] 网元1300包括用于存储指令和各种数据的存储器1360或数据存储装置。存储器1360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其组合。例如,存储器1360

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,ROM)、随机存取存储器(random access memory,RAM)、三态内容寻址存储器(ternary content-addressable memory,TCAM)和/或静态随机存取存储器(static random-access memory,SRAM)。存储器1360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,存储器1360可以用作溢出数据存储设备,以在选择执行程序时存储这些程序,并存储在程序执行期间读取的指令和数据。

[0167] 网元1300具有一个或多个处理器1330或其它处理装置(例如,中央处理器(central processing unit,CPU))来处理指令。处理器1330可以实现为一个或多个CPU芯片、内核(例如,作为多核处理器)、现场可编程门阵列(field-programmable gate array,FPGA)、专用集成电路(application specific integrated circuit,ASIC)和数字信号处理器(digital signal processor,DSP)。处理器1330通过系统总线与入端口1310、RX 1320、TX 1340、出端口1350和存储器1360通信耦合。处理器1330可用于执行存储在存储器1360中的指令。因此,当相应的指令由处理器执行时,处理器1330提供用于执行与权利要求对应的任何计算、比较、确定、启动、配置或任何其它操作的装置。在一些实施例中,存储器1360可以是与处理器1330集成的存储器。

[0168] 在一个实施例中,存储器1360存储FT计算模块1370。FT计算模块1370包括用于实现所公开实施例的数据和可执行指令。例如,FT计算模块1370可以包括用于实现图1、3A-C、5A-B、8和16A-B中描述的方法的指令。通过包含FT计算模块1370,可以允许网络的多个节点同时进行FT的确定性计算,从而显著改进网元1300的功能。包括实现所公开的用于计算FT的第一方法的FT计算模块1370的网元1300是与实现所公开的用于计算FT的第二过程的FT计算模块1370的网元1300不同的系统。

[0169] 图14示出了第一装置1400,该第一装置用于实现本文所述的用于计算网络的FT的一种或多种方法,例如,图1的过程100、图3A-B的过程300、图5A-B的过程500以及图16A-B的过程1600。第一装置1400可以在网元1300中实现。装置1400包括:装置1402,用于计算FT,所述FT包括网络中的所有节点,如结合图1的子过程100a、图3A的子过程300a、图5A的子过程500a以及图16A的子过程1600a所述;以及装置1404,用于确保FT中的所有节点在FT中都至少具有两条链路,如结合图1的子过程100b、图3B的子过程300b、图5B的子过程500b和图16B的子过程1600b所述。

[0170] 图15示出了第二装置1500,该第二装置用于实现本文所述的用于计算完整FT的一种或多种方法,例如,图8的过程800。第二装置1500可以在网元1300中实现。装置1500包括:装置1502,用于计算FT0,FT0包括网络中的所有确定性节点,如结合图8的子过程800a所述;以及装置1504,用于通过计算FT1来计算完整FT,FT1包括网络中不在FT0中的所有节点,并且将FT0附加到FT1以计算完整FT,如结合图8的子过程800b所述。

[0171] 图16A和16B示出了本发明实施例提供的用于计算FT的第五种方法1600的流程图。方法1600类似于结合图3A和3B所述的方法300。方法300从MaxD的估计值开始,并根据需要增加该值以成功计算网络的FT。相比之下,方法1600从先前成功计算网络(或先前版本的网络)的FT的MaxD的存储值开始,并尝试减小MaxD的值以找到将成功计算网络的FT的最小MaxD。

[0172] 方法1600包括用于计算包括网络中所有节点的FT的子过程1600a(如图16A所示)

和用于确保FT中的所有节点在FT中都至少具有两条链路的子过程1600b(如图16B所示)。除了下面描述的差异之外,方法1600中的步骤与方法300中相同编号的步骤的执行方式相同。

[0173] 在步骤1602中,将变量MaxD设置为初始值,该初始值等于来自先前为网络计算的FT的MaxD的存储值。在步骤1620中,当确定FT中的所有节点在FT中具有多条链路时(先前已在步骤1612中确定FT包括网络中的所有节点),方法1600已经成功地为当前MaxD值计算FT,并进入步骤1640。在步骤1640中,如果确定MaxD增加,即FT的先前计算不成功,则方法1600进入步骤1645;否则,方法1600进入步骤1652。在步骤1645中,存储MaxD的值,然后将FT发送到网络中的路由器组件,并且方法1600结束。在步骤1652中,将MaxD的成功值与成功计算的FT一起存储,MaxD的值递减,并且方法1600进入步骤1604,以利用MaxD的新值重新运行子过程1600a和1600b。

[0174] 如果在步骤1606中确定不能向FT中添加更多节点或在步骤1624中确定不能向节点添加第二链路,则MaxD的当前值太低而不能成功计算FT,并且方法1600进入步骤1642。在步骤1642中,如果确定存储了MaxD的值和先前计算的FT,则方法1600进入步骤1645,否则方法1600进入步骤1648。在步骤1645中,将MaxD的值设置为先前存储的MaxD的成功值,并且将FT设置为先前存储的成功计算的FT。然后将FT发送到网络中的路由器组件,并且方法1600结束。在步骤1648中,将MaxD的值增加1,然后方法1600进入步骤1604,以利用MaxD的新值重新运行子过程1600a和1600b。

[0175] 虽然本发明中已提供若干实施例,但应理解,在不脱离本发明的精神或范围的情况下,本发明所公开的系统和方法可以用许多其它特定形式来体现。本发明的示例应被视为说明性而非限制性的,且本发明并不限于本文中所给出的细节。例如,各种元件或组件可以在另一系统中组合或整合,或者某些特征可以省略或不实施。

[0176] 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 method 可以与其它系统、模块、技术或方法进行组合或合并。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,均不脱离本文公开的精神和范围。

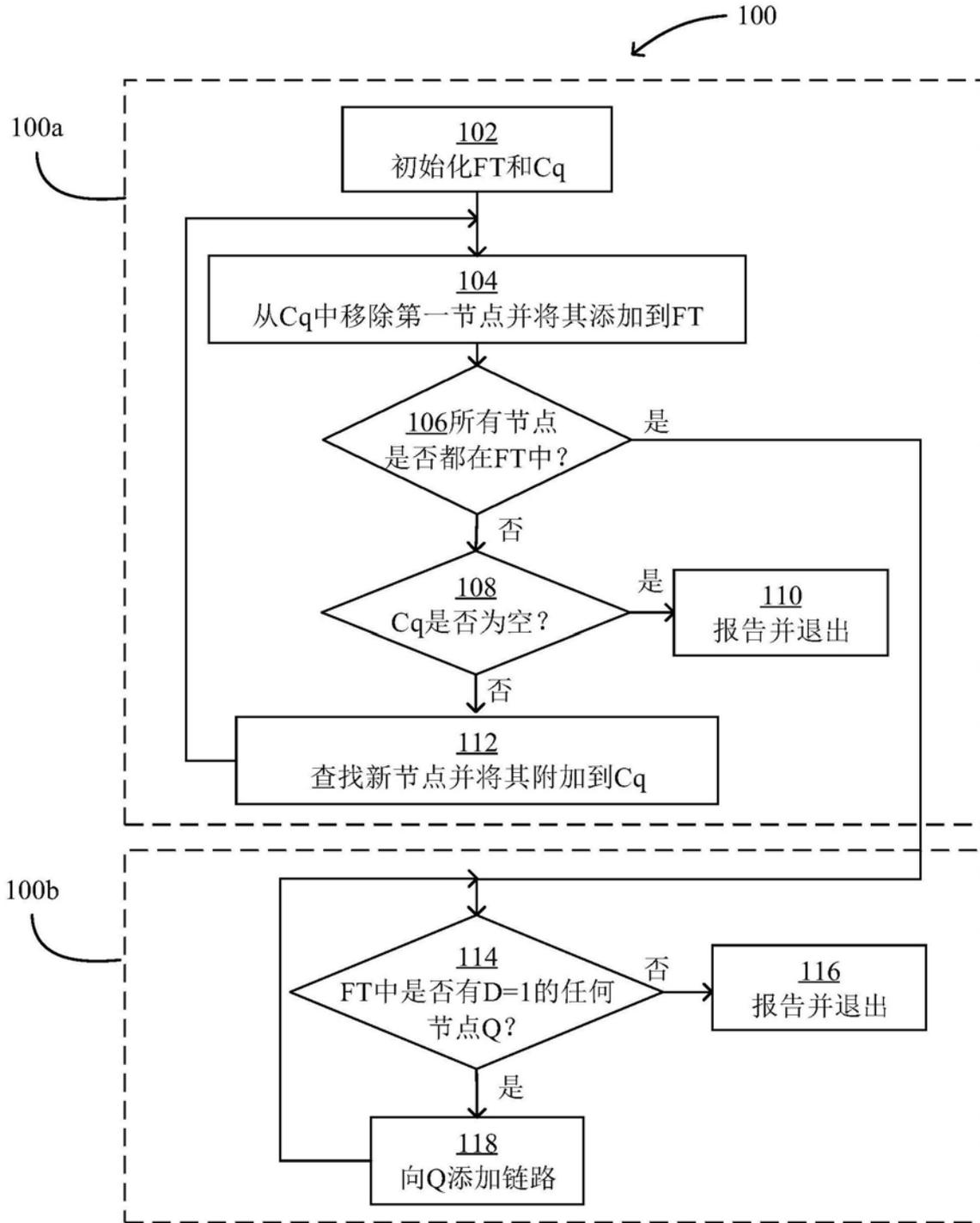


图1

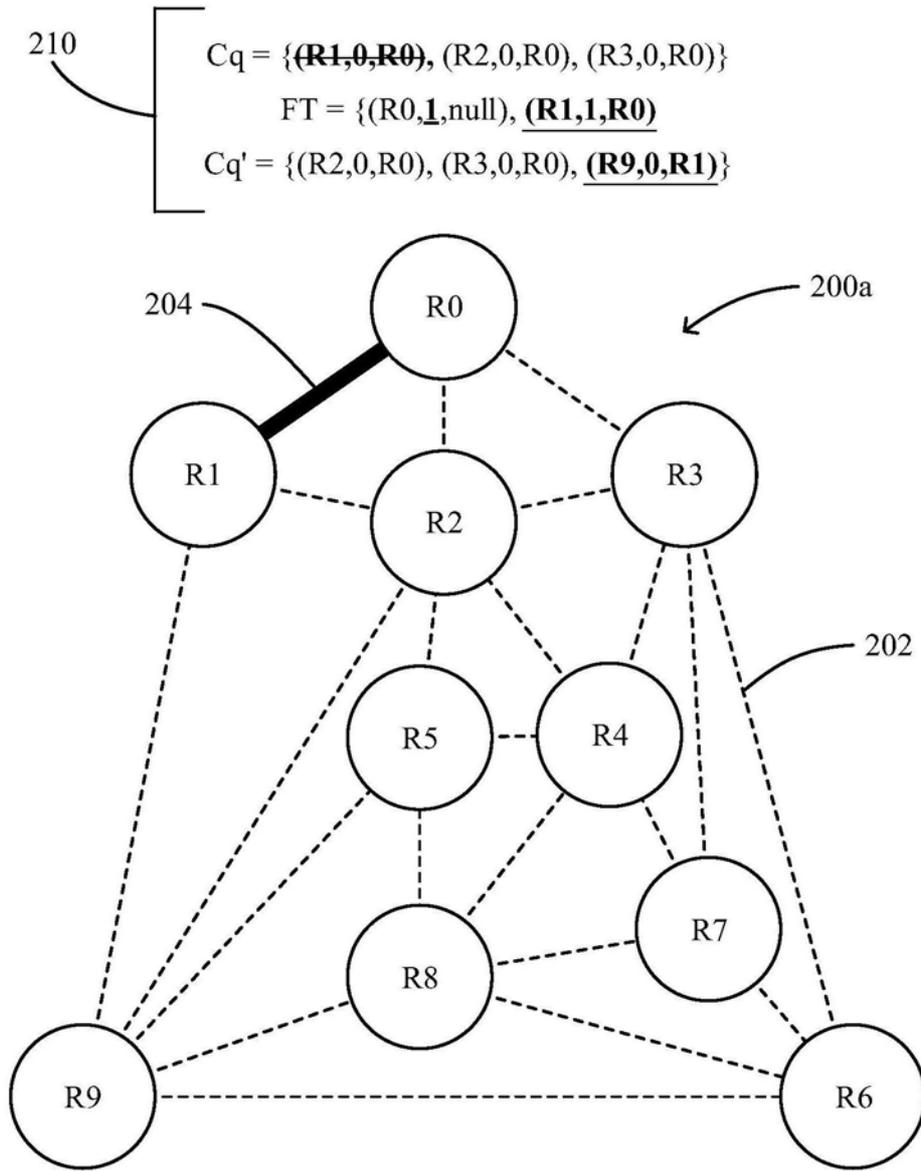


图2A

$$Cq = \{(\underline{R2},0,R0), (R3,0,R0), (R9,0,R1)\}$$

$$FT = \{(R0,2,null), (R1,1,R0), (\underline{R2},1,R0)\}$$

$$Cq' = \{(R3,0,R0), (R9,0,R1), (\underline{R4},0,R2), (\underline{R5},0,R2)\}$$

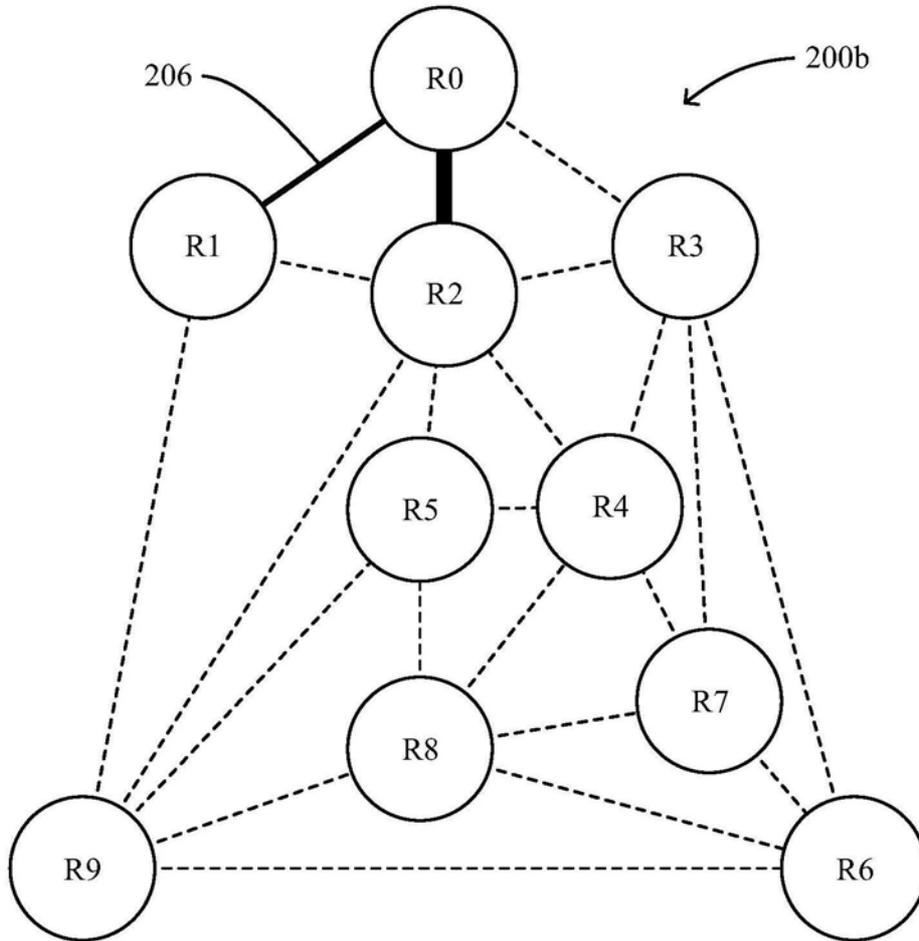


图2B

$$Cq = \{(\mathbf{R8,0,R9})\}$$

$$FT = \{(R0,3,null), (R1,2,R0), (R2,3,R0), (R3,3,R0), (R9,2,R1), (R4,1,R2), (R5,1,R2), (R6,1,R3), (R7,1,R3), (\mathbf{R8,1,R9})\}$$

$$Cq' = \{ \}$$

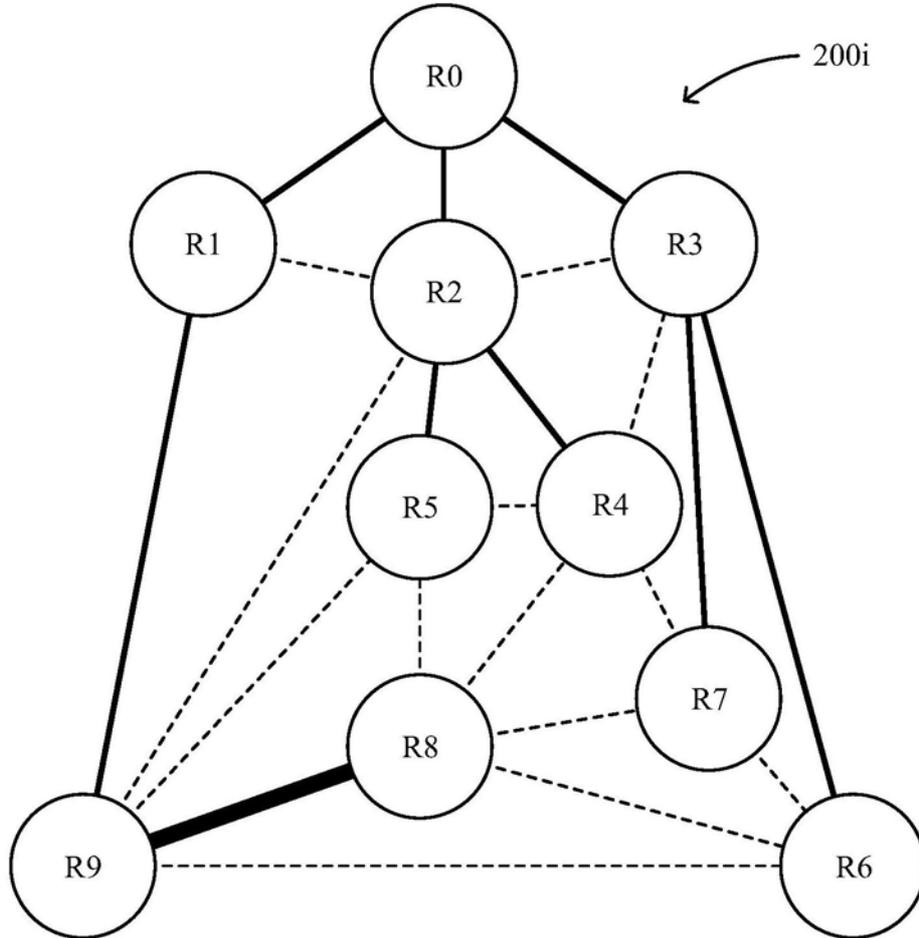


图2C

220 $FT = \{(R0,3,null), (R1,2,R0), (R2,3,R0), (R3,3,R0), (R9,2,R1), (R4,1,R2), (R5,1,R2), (R6,1,R3), (R7,1,R3), (R8,1,R9)\}$
 $FT' = \{(R0,3,null), (R1,2,R0), (R2,3,R0), (R3,3,R0), (R9,2,R1), (R4,2,R2), (R5,2,R2), (R6,1,R3), (R7,1,R3), (R8,1,R9), (R4,2,R5)\}$

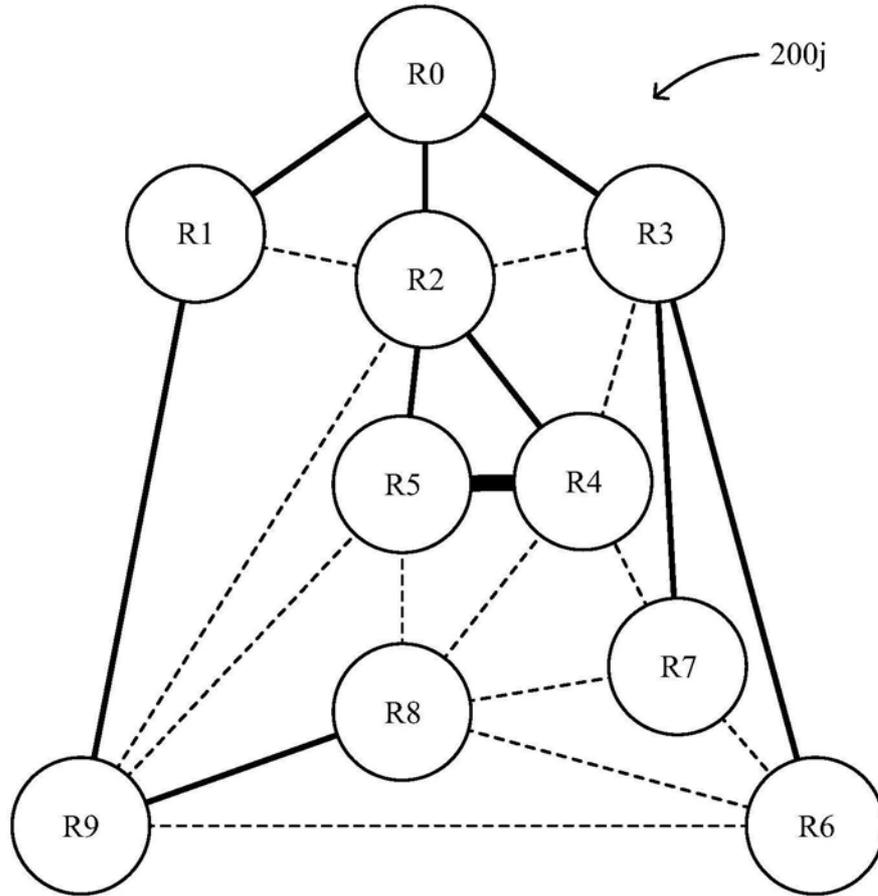


图2D

FT = {(R0,3,null), (R1,2,R0), (R2,3,R0), (R3,3,R0), (R9,2,R1), (R4,2,R2),
(R5,2,R2), (R6,1,R3), (R7,1,R3), (R8,1,R9), (R4,2,R5)}
FT' = {(R0,3,null), (R1,2,R0), (R2,3,R0), (R3,3,R0), (R9,2,R1), (R4,2,R2),
(R5,2,R2), (R6,2,R3), (R7,2,R3), (R8,1,R9), (R4,2,R5), **(R6,2,R7)**}

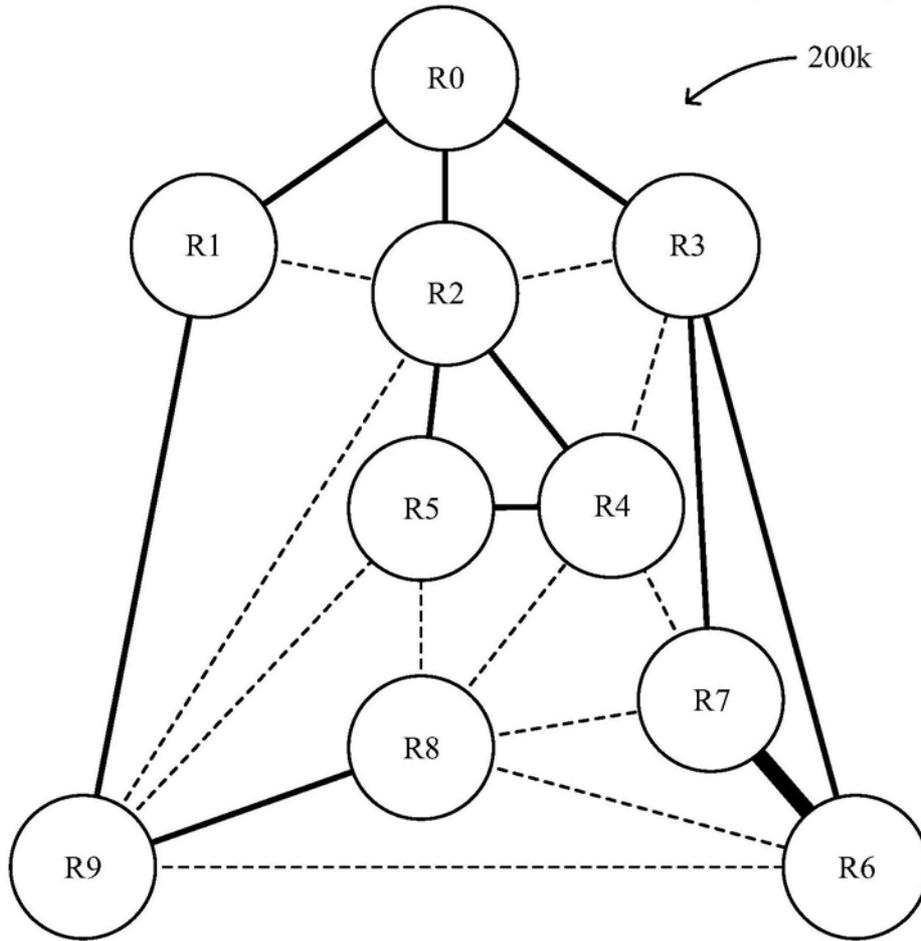


图2E

FT = {(R0,3,null), (R1,2,R0), (R2,3,R0), (R3,3,R0), (R9,2,R1), (R4,2,R2),
(R5,2,R2), (R6,2,R3), (R7,2,R3), (R8,1,R9), (R4,2,R5), (R6,2,R7)}
FT' = {(R0,3,null), (R1,2,R0), (R2,3,R0), (R3,3,R0), (R9,2,R1), (R4,3,R2),
(R5,2,R2), (R6,2,R3), (R7,2,R3), (R8,2,R9), (R4,2,R5), (R6,2,R7), **(R8,2,R4)**}

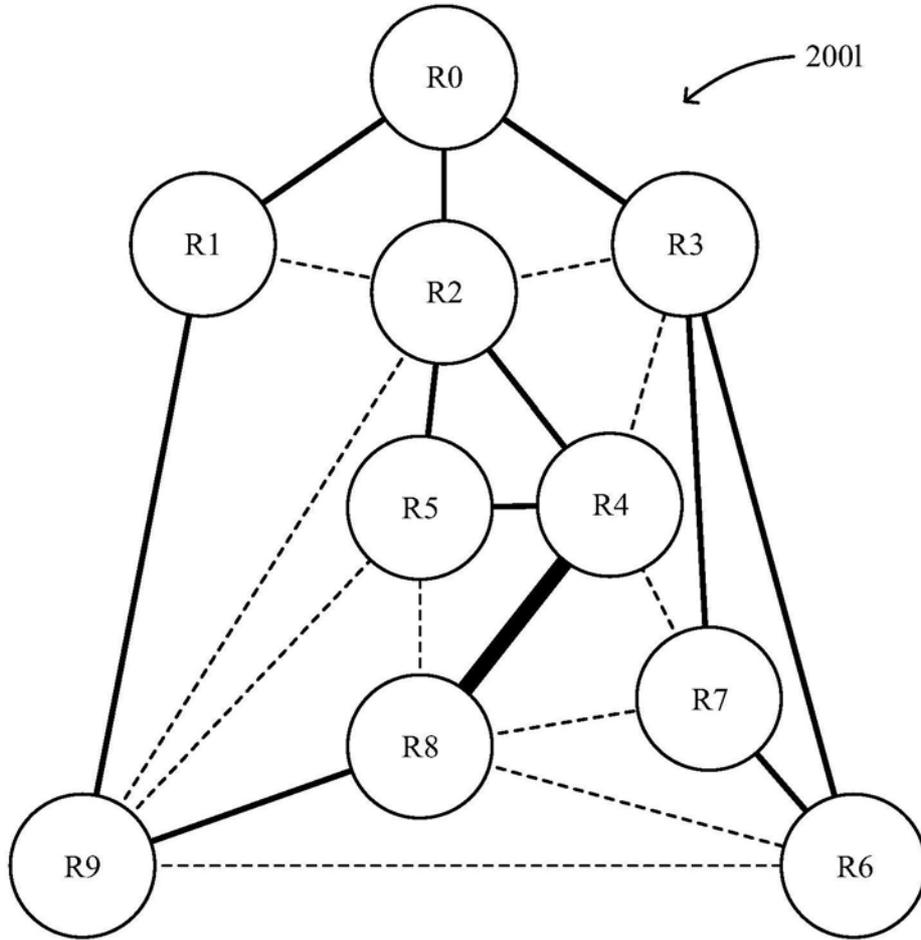
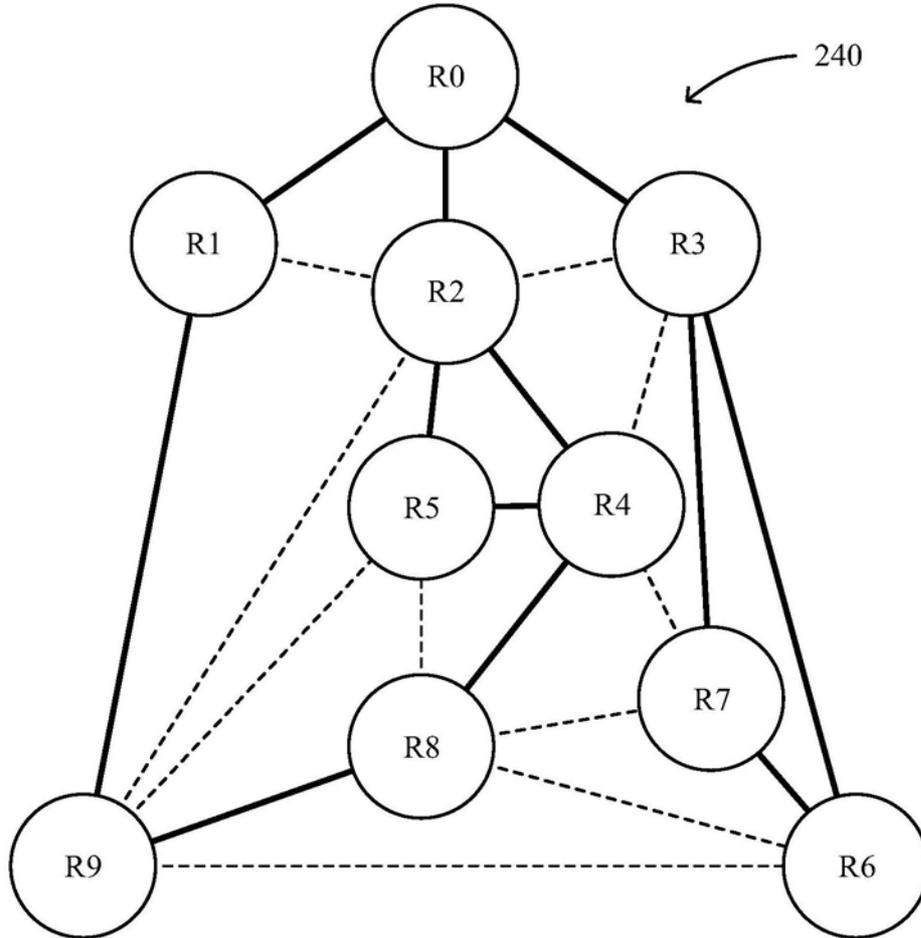


图2F

230



$FT = \{(R0,3,null), (R1,2,R0), (R2,3,R0), (R3,3,R0), (R9,2,R1), (R4,3,R2), (R5,2,R2), (R6,2,R3), (R7,2,R3), (R8,2,R9), (R4,2,R5), (R6,2,R7), (R8,2,R4)\}$



240

图2G

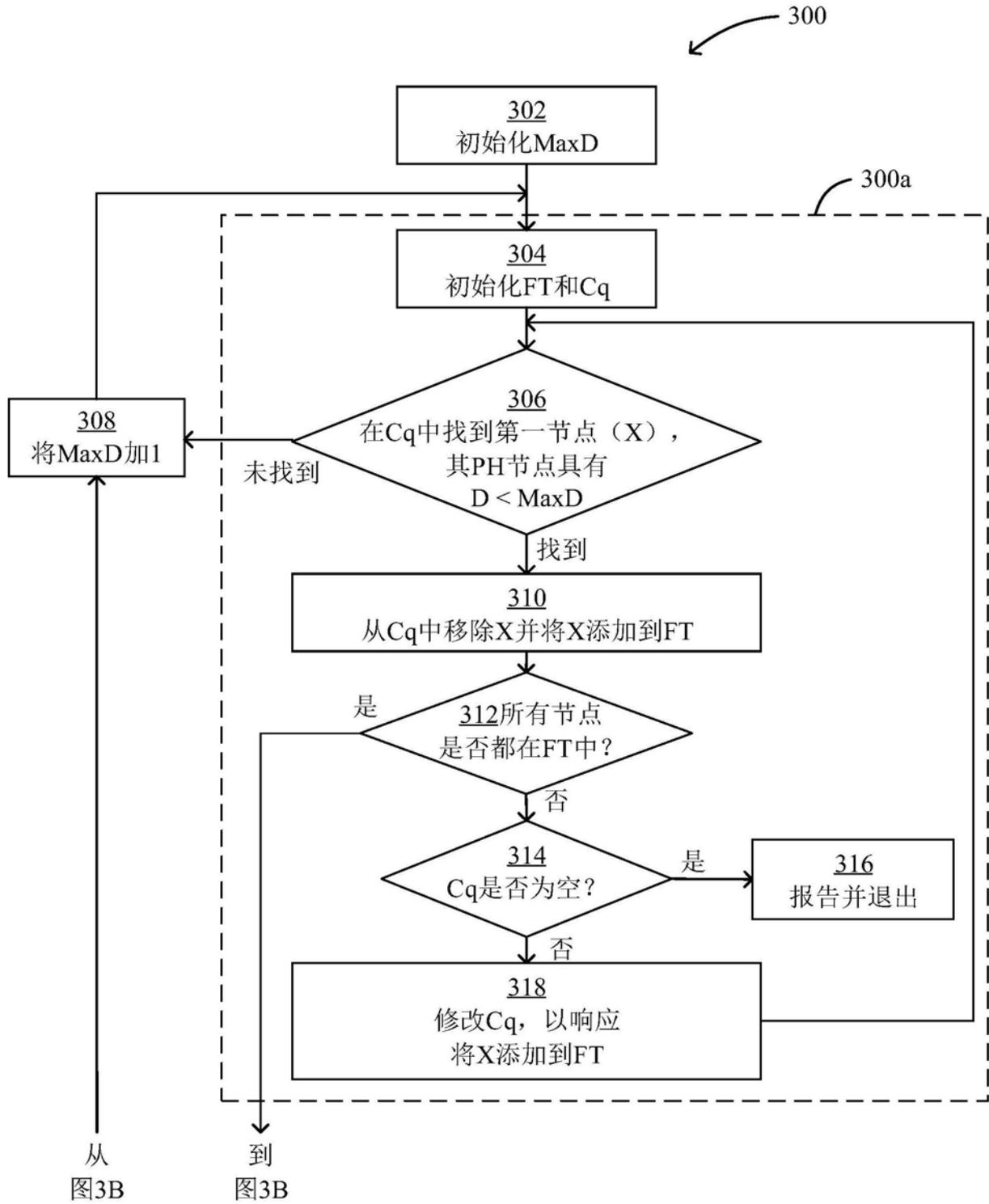


图3A

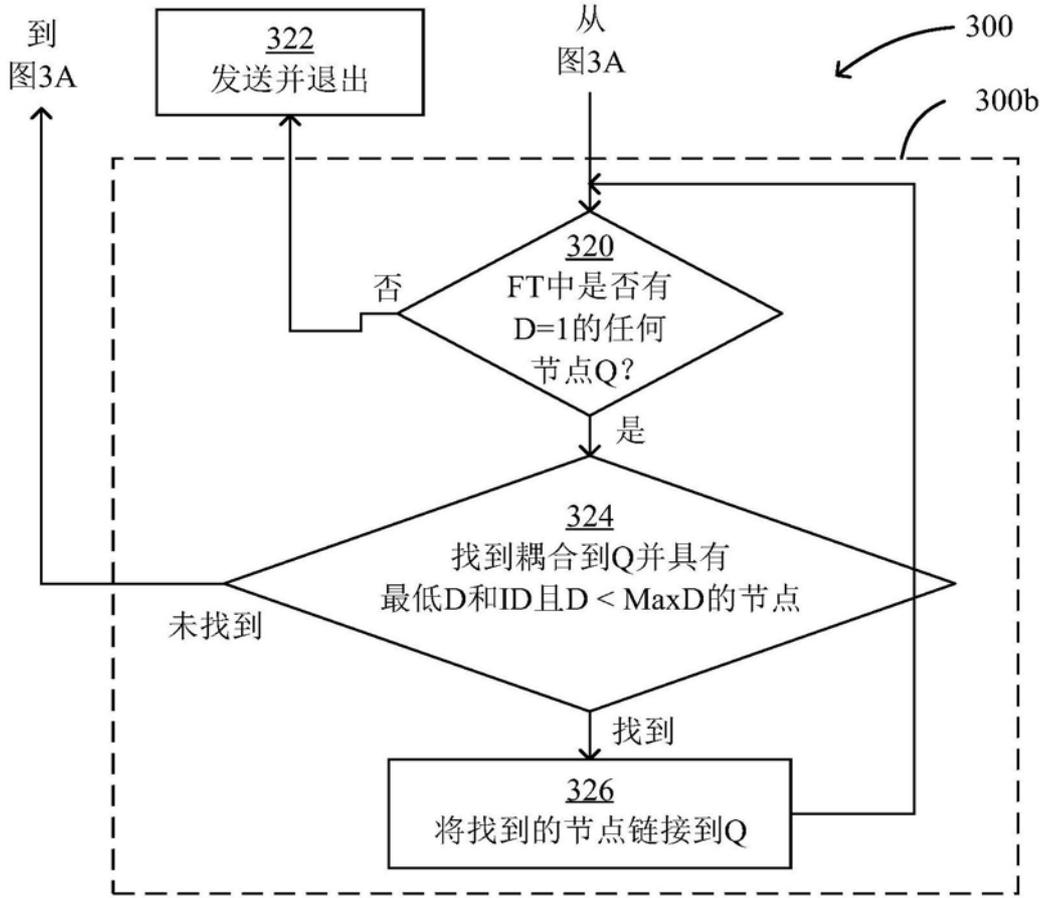


图3B

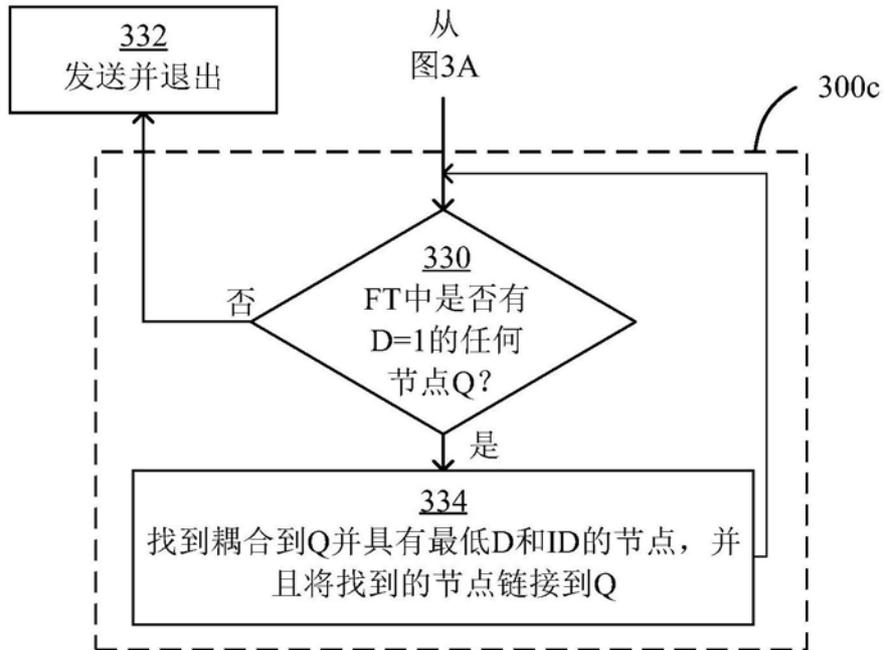


图3C

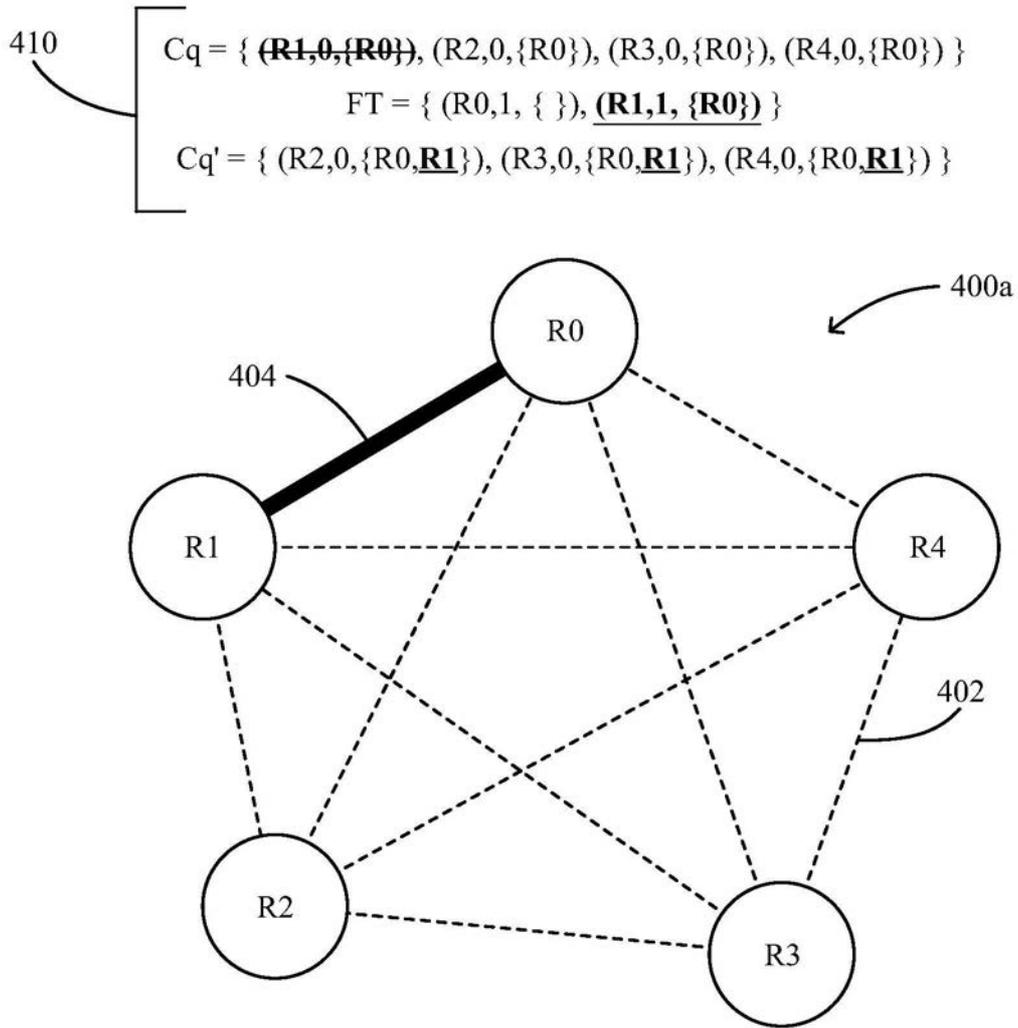


图4A

$$Cq = \{ (\mathbf{R2},0,\{\mathbf{R0},\mathbf{R1}\}), (R3,0,\{R0,R1\}), (R4,0,\{R0,R1\}) \}$$

$$FT = \{ (R0,\mathbf{2},\{ \}), (R1,1,\{R0\}), (\mathbf{R2},1,\{\mathbf{R0}\}) \}$$

$$Cq' = \{ (R3,0,\{R0,R1,\mathbf{R2}\}), (R4,0,\{R0,R1,\mathbf{R2}\}) \}$$

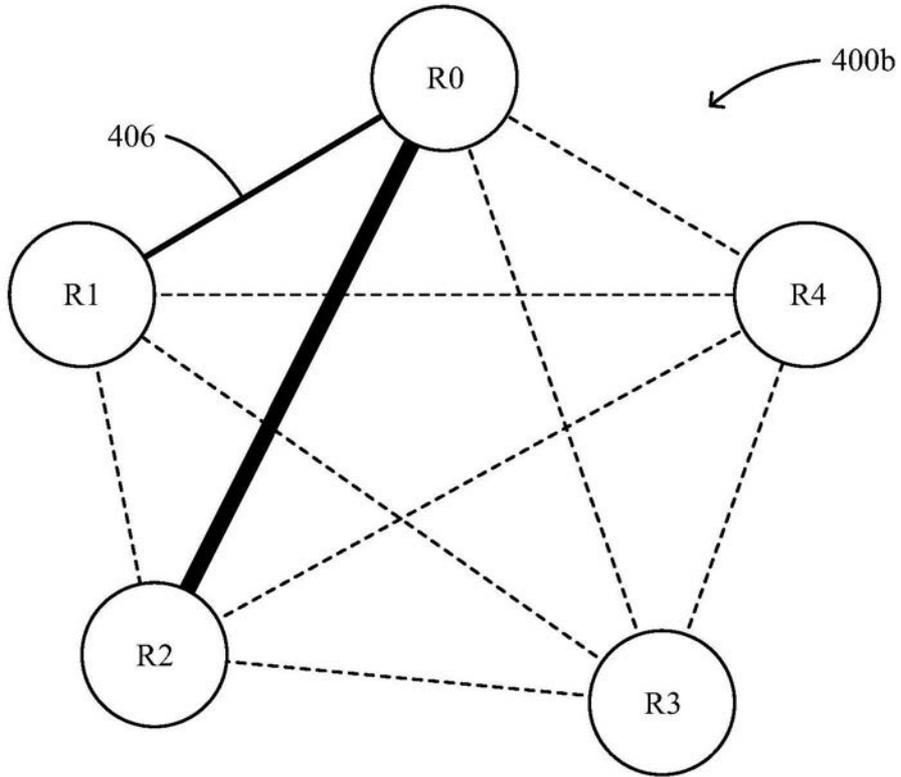


图4B

$$Cq = \{ (\mathbf{R4,0},\{\mathbf{R0,R1,R2}\}) \}$$

$$FT = \{ (R0,3,\{\}), (R1,2,\{R0\}), (R2,1,\{R0\}), (R3,1,\{R0\}), (\mathbf{R4,1,\{R1\}}) \}$$

$$Cq' = \{ \}$$

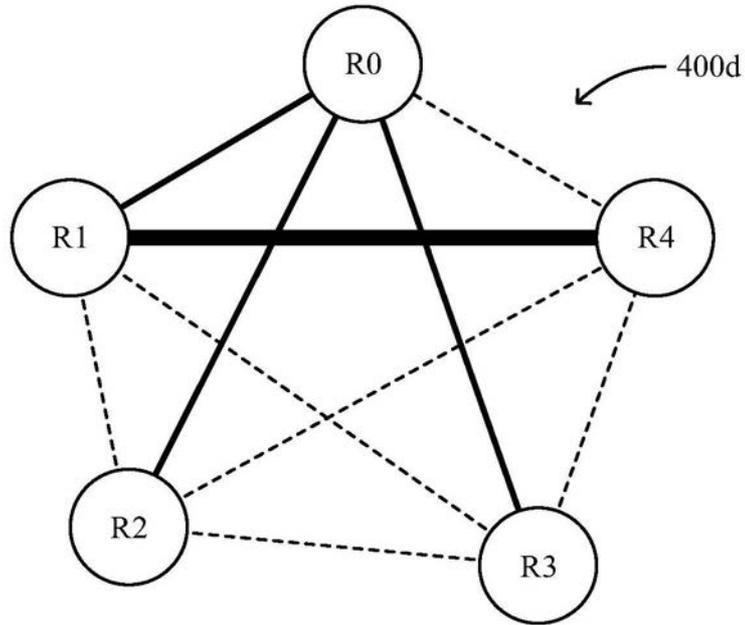


图4C

420 [FT = { (R0,3,{ }), (R1,2,{R0}), (R2,1,{R0}), (R3,1,{R0}), (R4,1,{R1}) }
FT' = { (R0,3,{ }), (R1,2,{R0}), (R2,2,{R0, **R3**}), (R3,2,{R0}), (R4,1,{R1}) }]

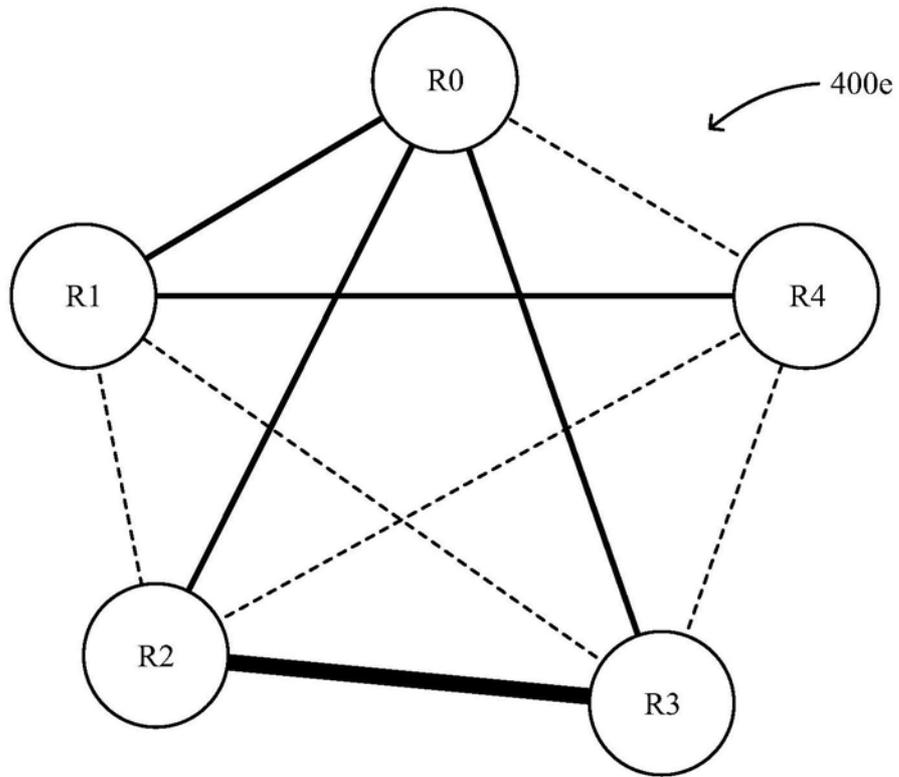


图4D

$FT = \{ (R0,3,\{ \}), (R1,2,\{R0\}), (R2,2,\{R0, R3\}), (R3,2,\{R0\}), (R4,1,\{R1\}) \}$
 $FT' = \{ (R0,3,\{ \}), (R1,2,\{R0\}), (R2,3,\{R0, R3\}), (R3,2,\{R0\}), (R4,2,\{R1, \mathbf{R2}\}) \}$

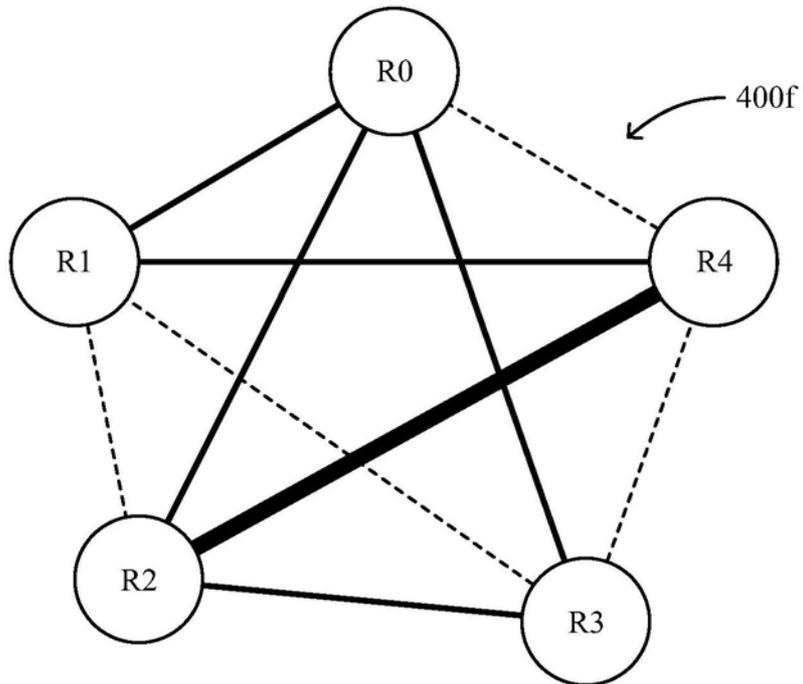


图4E

430

$$FT = \{ (R0,3,\{ \}), (R1,2,\{R0\}), (R2,3,\{R0, R3\}), (R3,2,\{R0\}), (R4,2,\{R1, R2\}) \}$$

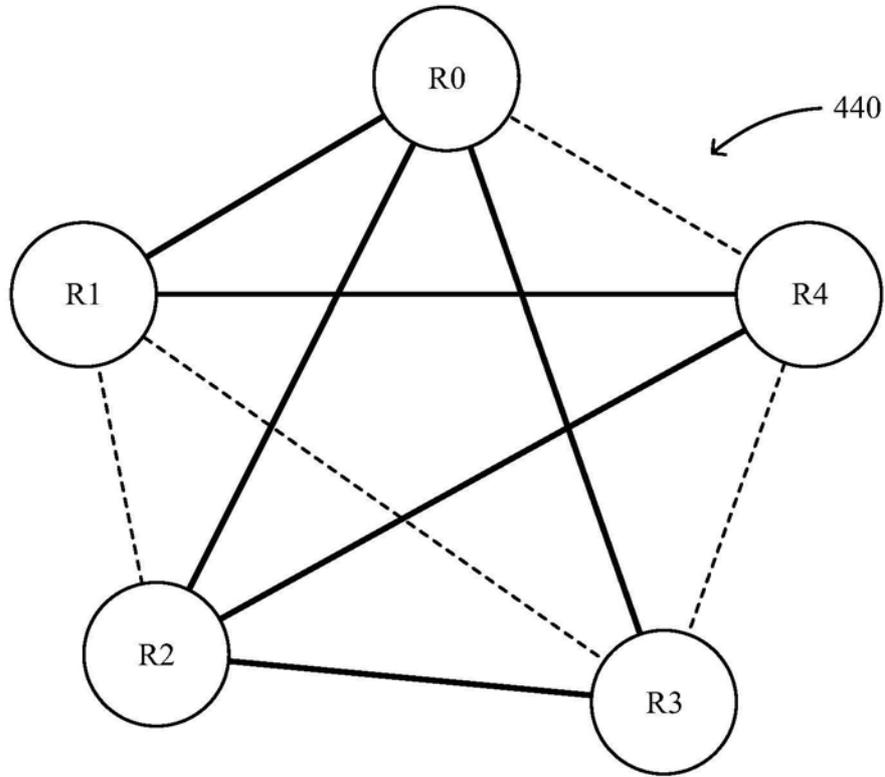


图4F

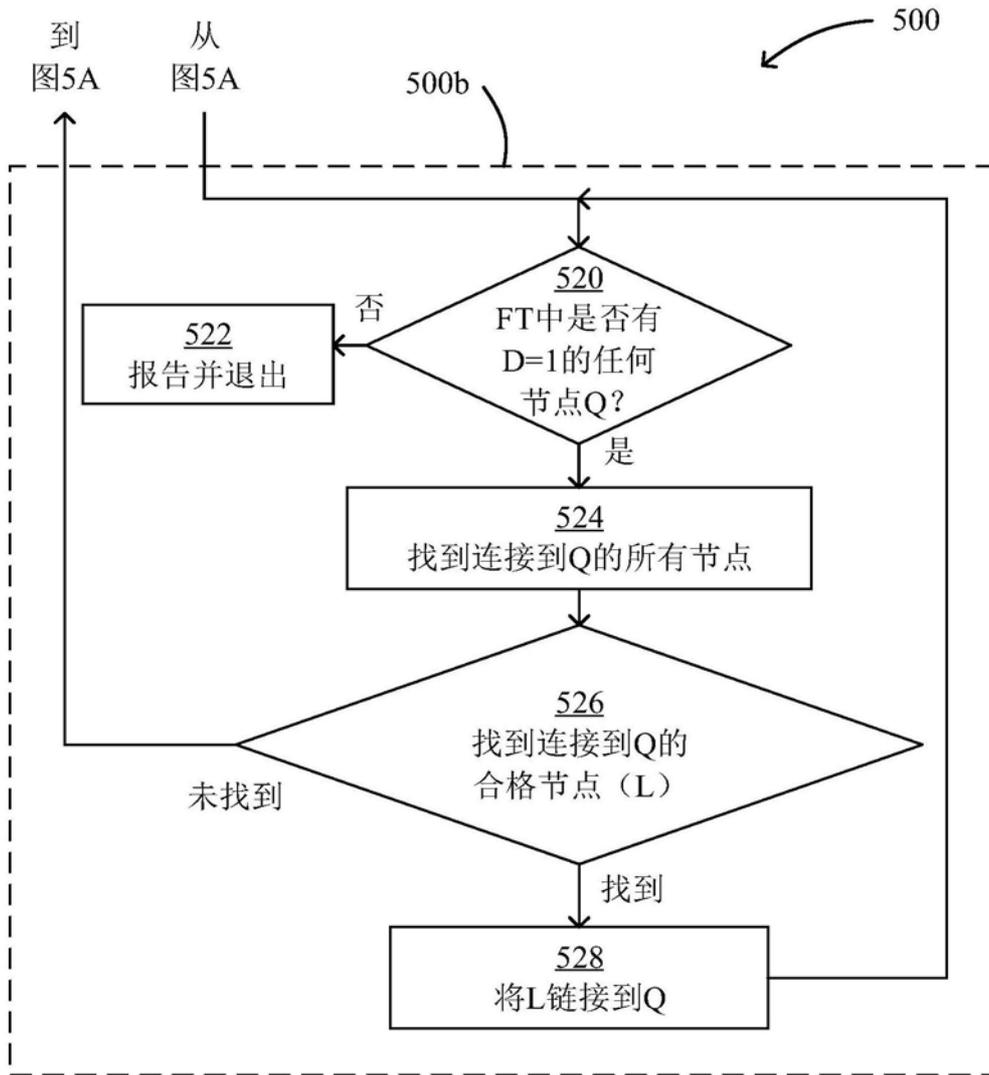


图5B

610

MaxD = 3

$C_q = \{ (\mathbf{R2},0,\{\mathbf{R0}\}), (R3,0,\{R0\}), (R4,0,\{R0\}), (R5,0,\{R0\}) \}$

$FT = \{ (R0,\mathbf{1},\{ \}), (\mathbf{R2},1,\{\mathbf{R0}\}) \}$

$C_{q'} = \{ (R3,0,\{R0\}), (R4,0,\{R0\}), (R5,0,\{R0\}), (\mathbf{R1},0,\{\mathbf{R2}\}), (\mathbf{R6},0,\{\mathbf{R2}\}), (\mathbf{R7},0,\{\mathbf{R2}\}), (\mathbf{R8},0,\{\mathbf{R2}\}), (\mathbf{R9},0,\{\mathbf{R2}\}), (\mathbf{R10},0,\{\mathbf{R2}\}) \}$

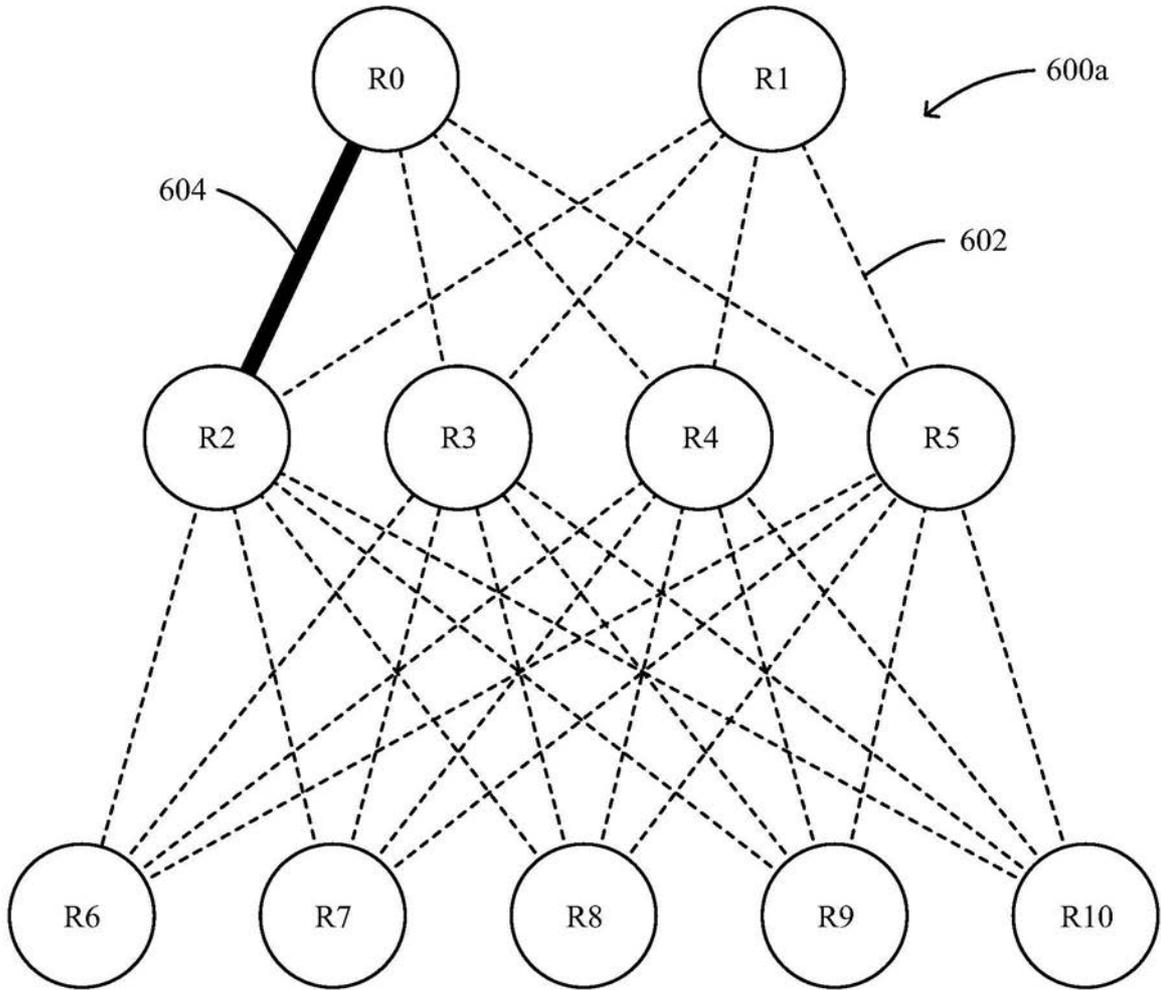


图6A

MaxD = 3

$C_q = \{ (\mathbf{R3,0}, \{\mathbf{R0}\}), (R4,0, \{R0\}), (R5,0, \{R0\}), (R1,0, \{R2\}), (R6,0, \{R2\}), (R7,0, \{R2\}), (R8,0, \{R2\}), (R9,0, \{R2\}), (R10,0, \{R2\}) \}$

$FT = \{ (R0, \mathbf{2}, \{ \}), (R2,1, \{R0\}), (\mathbf{R3,1}, \{R0\}) \}$

$C_{q'} = \{ (R4,0, \{R0\}), (R5,0, \{R0\}), (R1,0, \{R2, \mathbf{R3}\}), (R6,0, \{R2, \mathbf{R3}\}), (R7,0, \{R2, \mathbf{R3}\}), (R8,0, \{R2, \mathbf{R3}\}), (R9,0, \{R2, \mathbf{R3}\}), (R10,0, \{R2, \mathbf{R3}\}) \}$

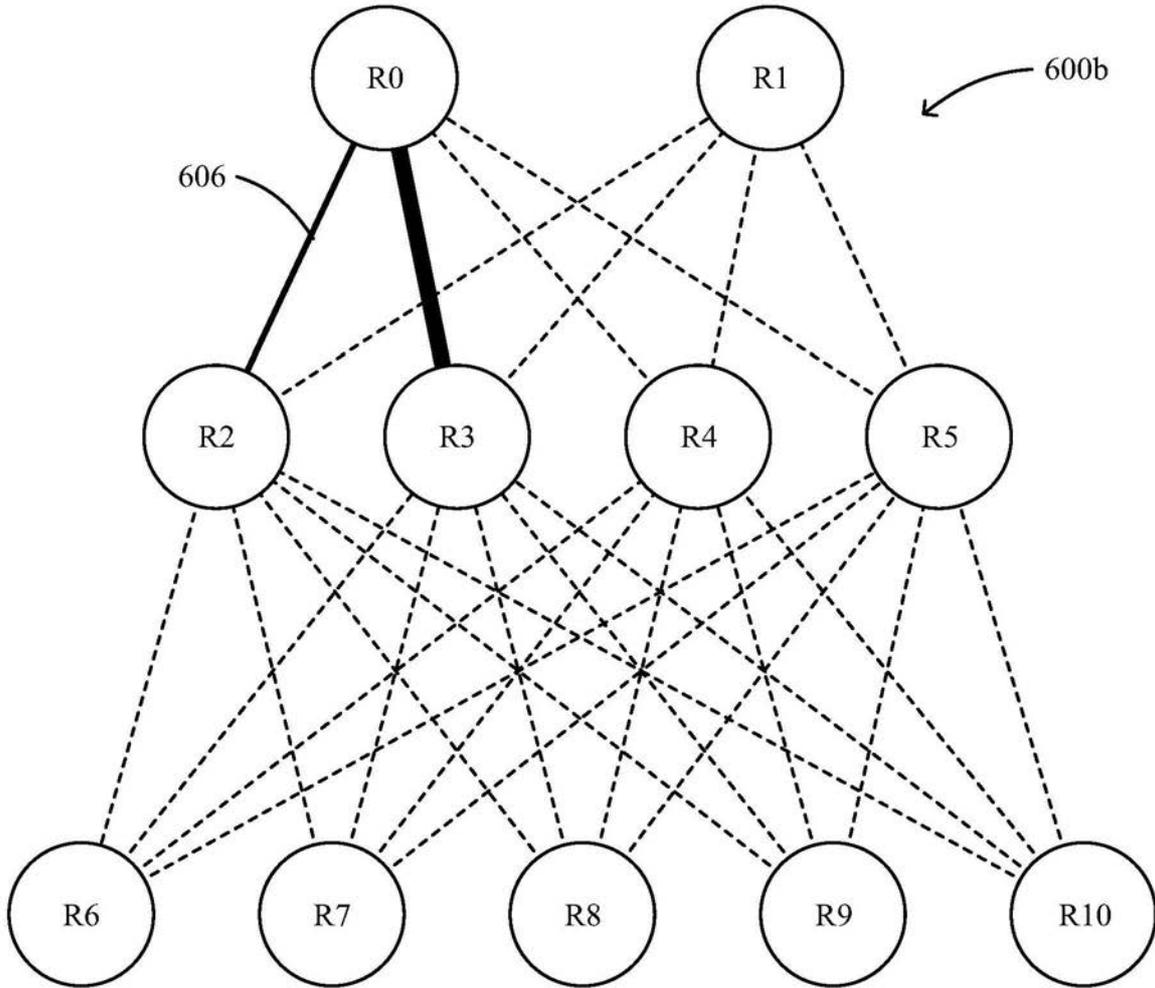


图6B

MaxD = 3
Cq = { (**R10,0**,{R2,R3,R4,R5}) }
FT = { (R0,3, { }), (R2,3, {R0}), (R3,3, {R0}), (R4,3, {R0}), (R1,2, {R2}),
(R5,1, {R1}), (R6,1, {R2}), (R7,1, {R3}), (R8,1,{R3}), (R9,1, {R4}), (**R10,1,{R4}**) }

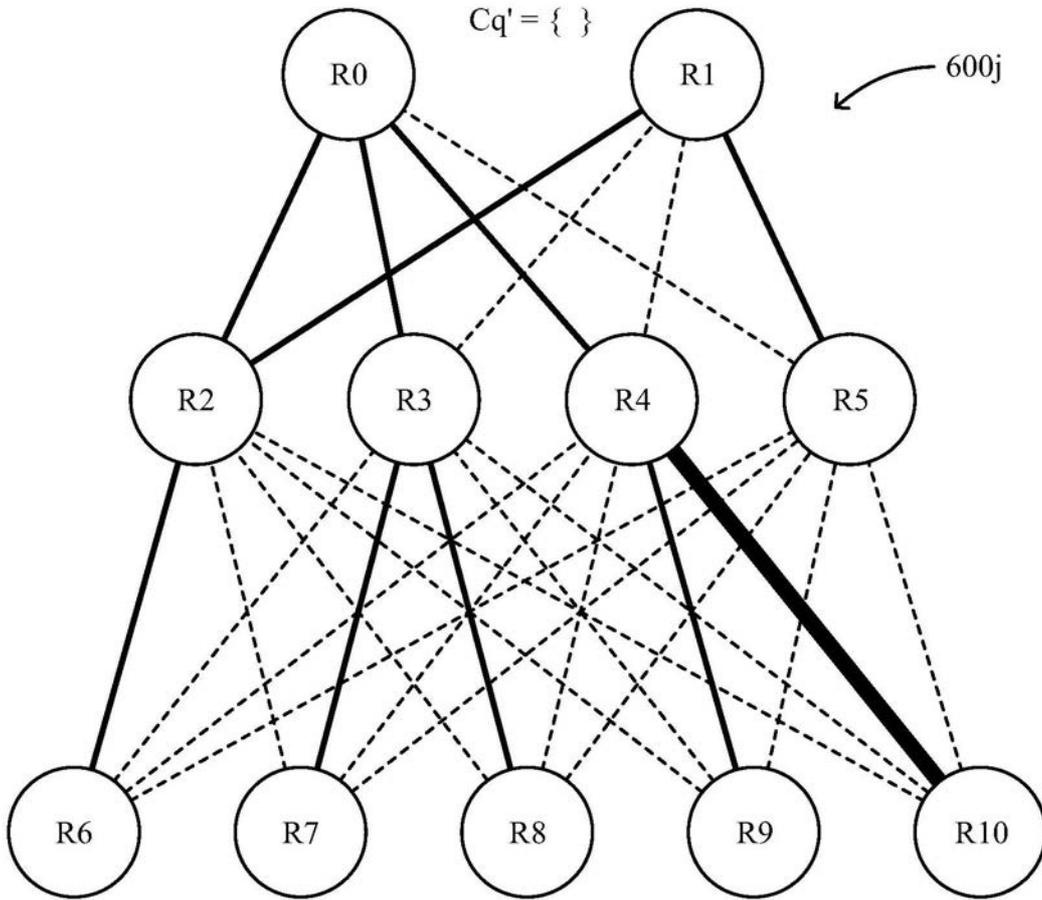


图6C

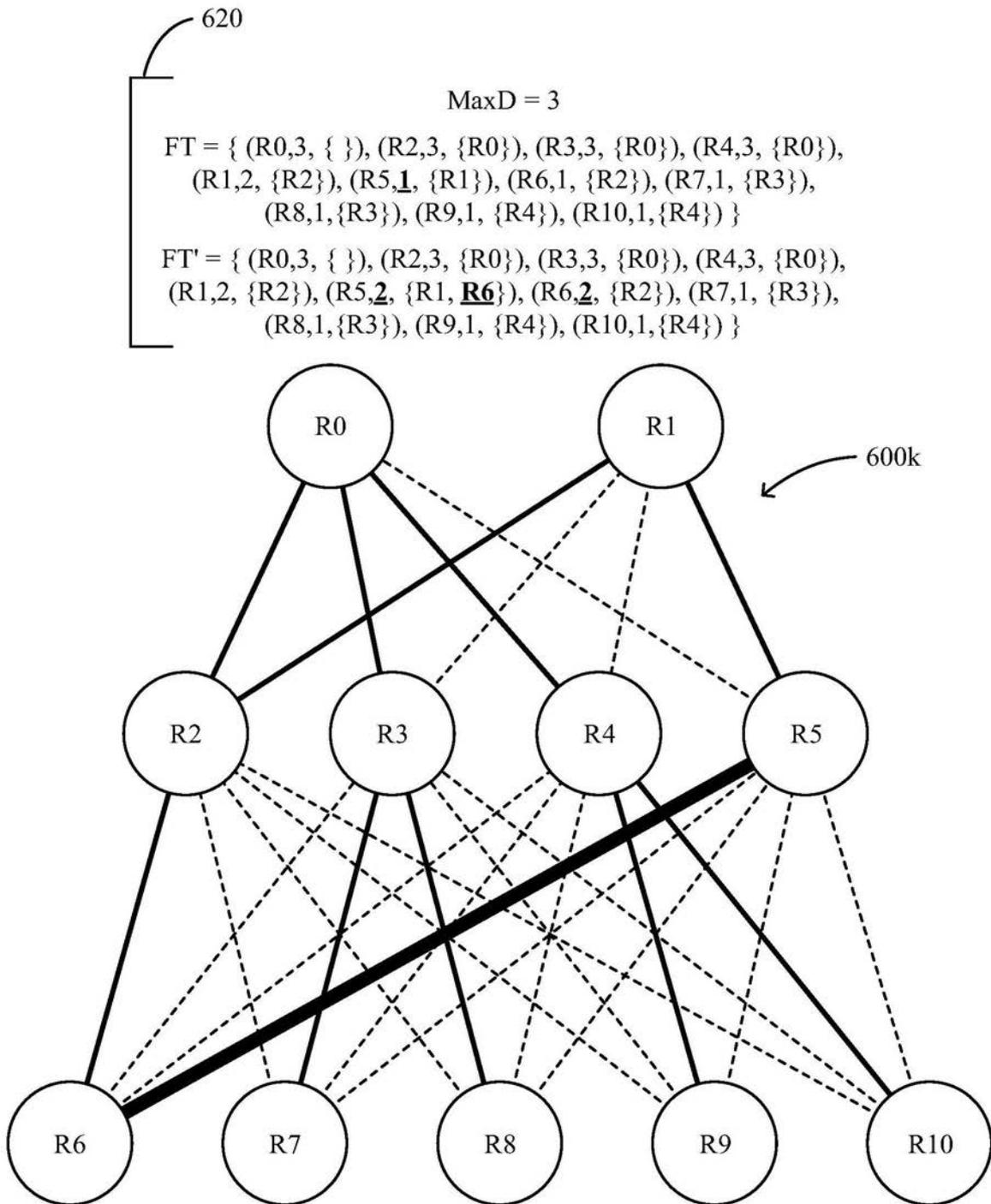


图6D

MaxD = 3

FT = { (R0,3, { }), (R2,3, {R0}), (R3,3, {R0}), (R4,3, {R0}),
(R1,2, {R2}), (R5,3, {R1, R6}), (R6,2, {R2}), (R7,2, {R3, R5}),
(R8,1,{R3}), (R9,1, {R4}), (R10,1,{R4}) }

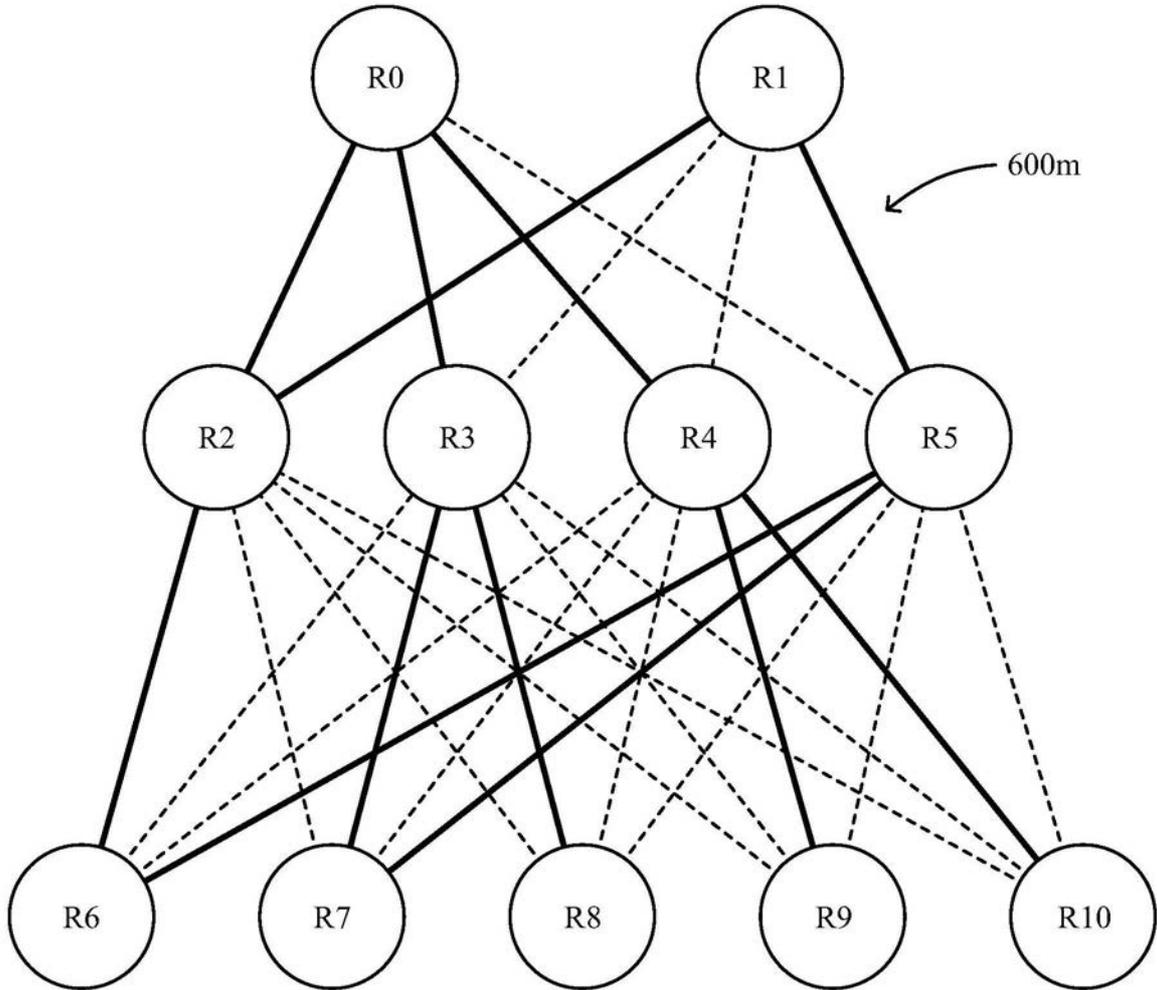


图6E

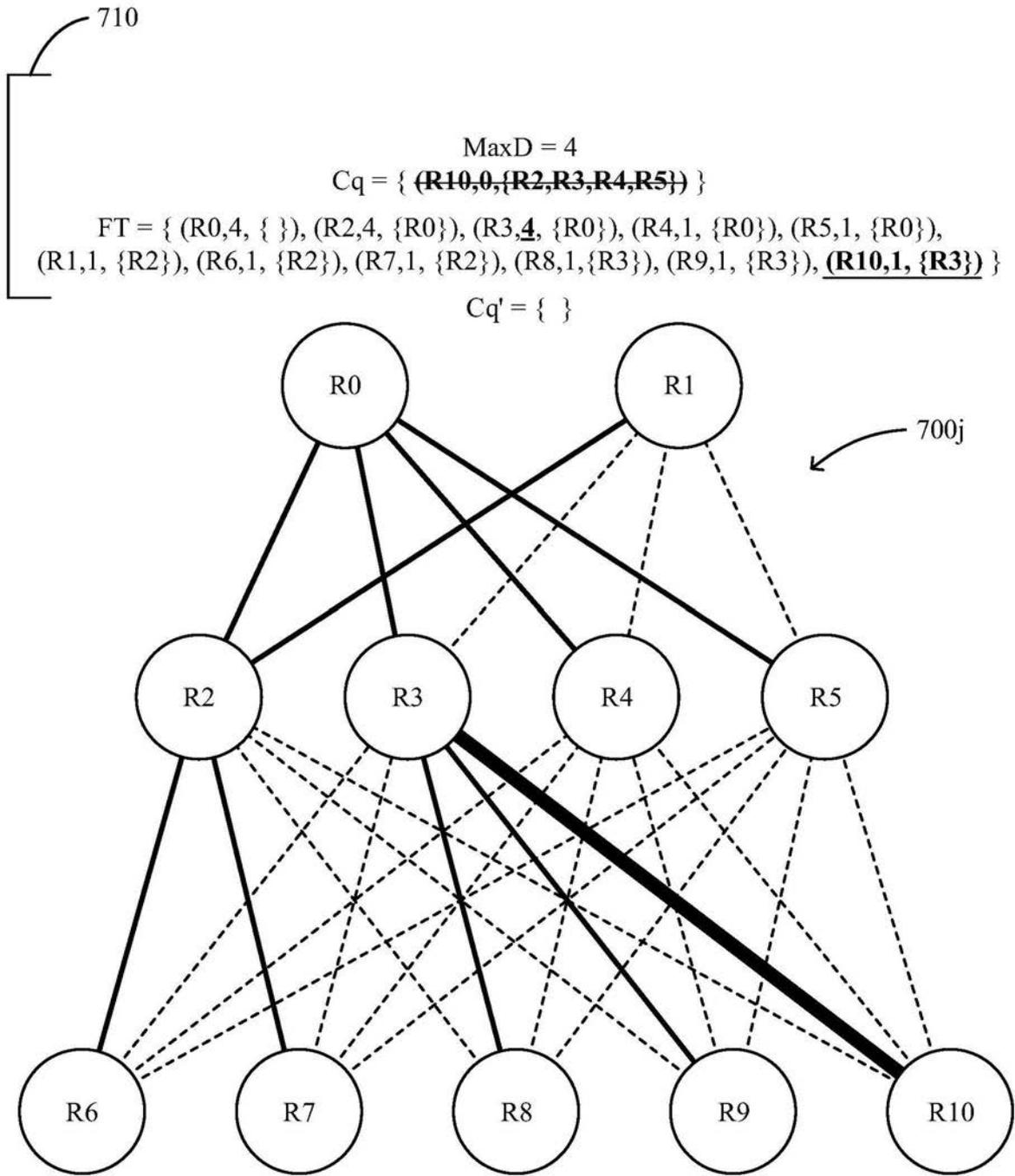


图7A

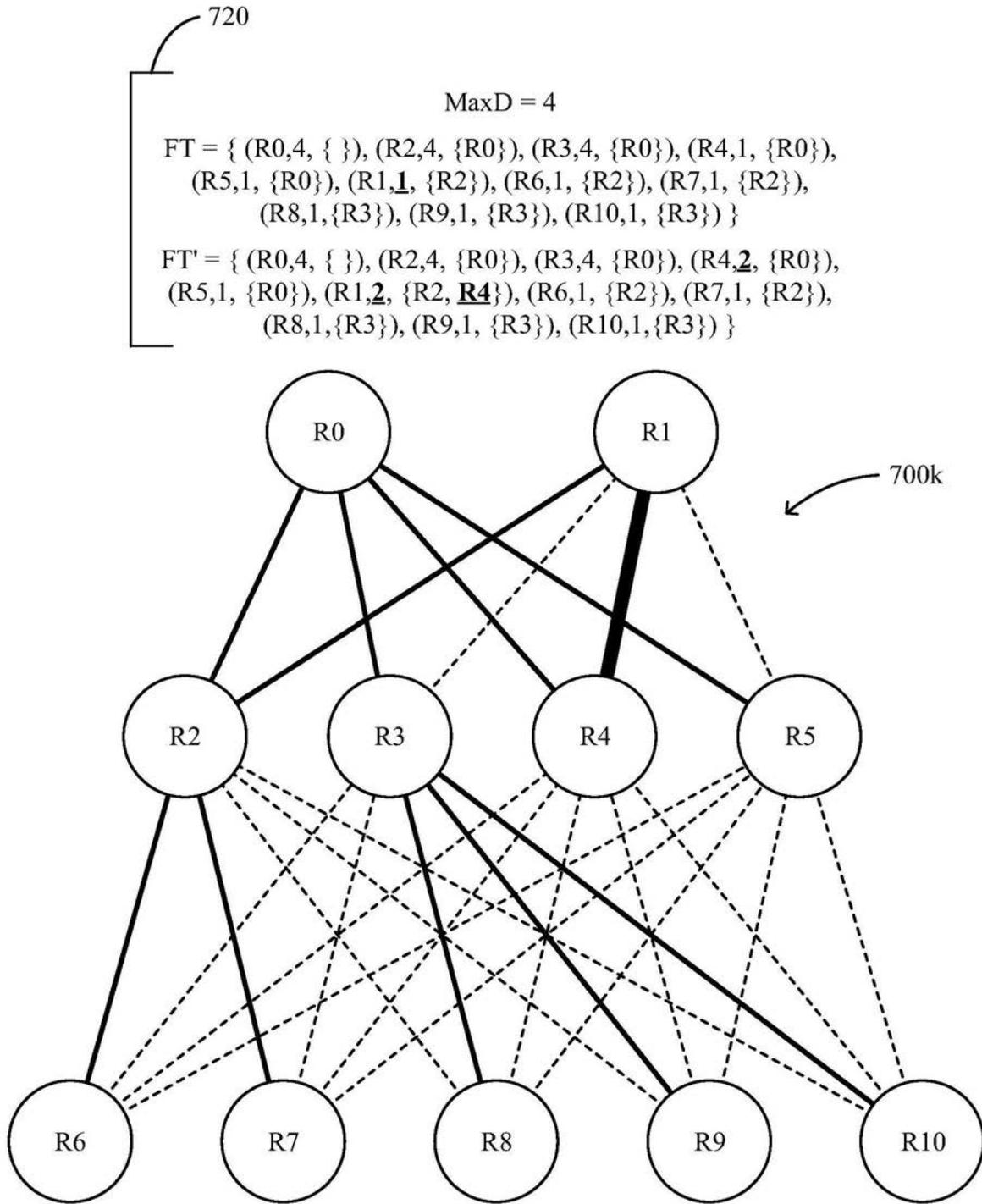


图7B

MaxD = 4

FT = { (R0,4, { }), (R2,4, {R0}), (R3,4, {R0}), (R4,4, {R0}),
(R5,3, {R0, R6}), (R1,2, {R2,R4}), (R6,2, {R2}), (R7,2, {R2, R4}),
(R8,2, {R3, R5}), (R9,2, {R3, R4}), (R10,1, {R3}) }

FT' = { (R0,4, { }), (R2,4, {R0}), (R3,4, {R0}), (R4,4, {R0}),
(R5,4, {R0, R6}), (R1,2, {R2,R4}), (R6,2, {R2}), (R7,2, {R2, R4}),
(R8,2, {R3, R5}), (R9,2, {R3, R4}), (R10,2, {R3, R5}) }

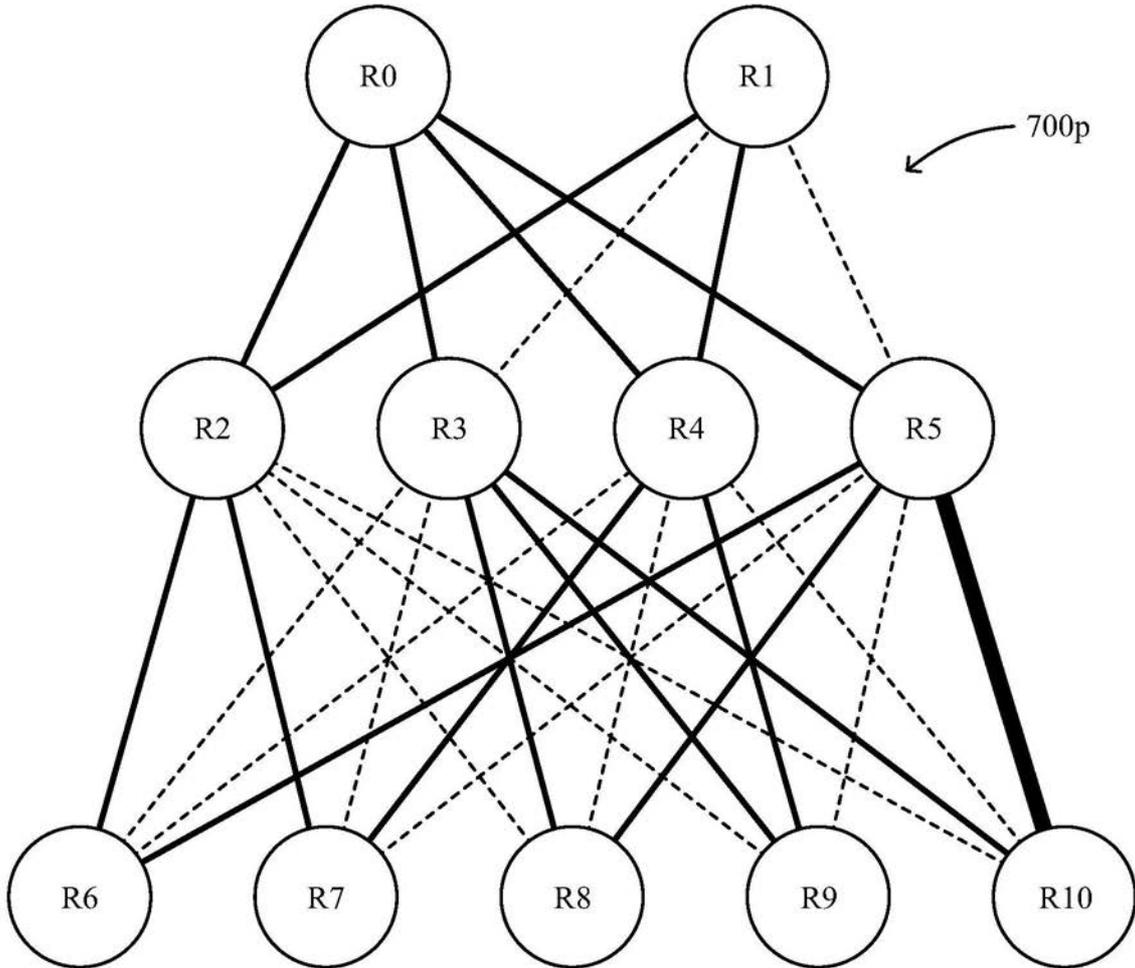


图7C

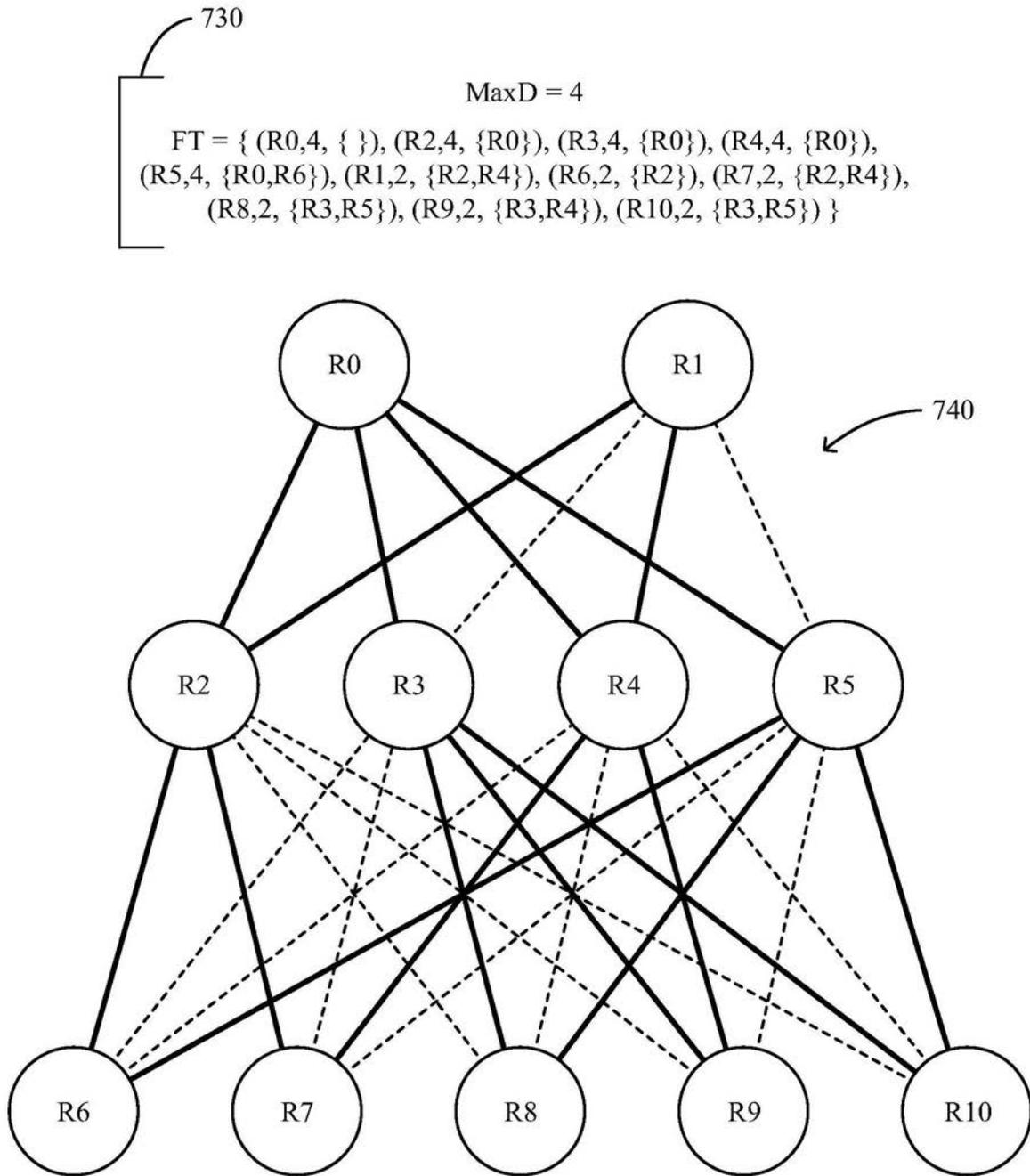


图7D

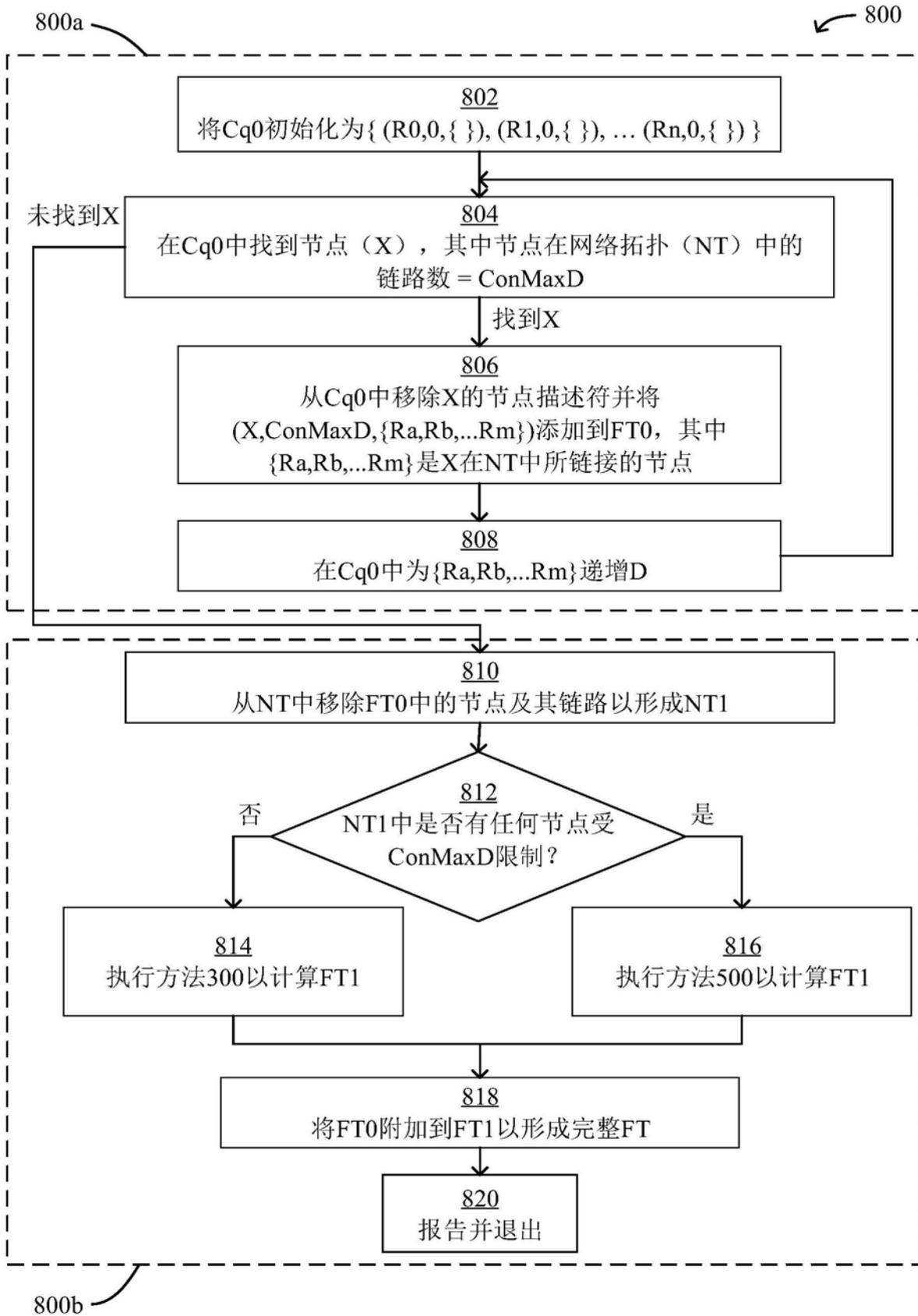


图8

910

ConMaxD = 2 (对于节点R4-R6)

$Cq0 = \{ (R0,0,\{\}), (R1,0,\{\}), (R2,0,\{\}), (R3,0,\{\}), (R4,0,\{\}), (R5,0,\{\}), (R6,0,\{\}) \}$

$FT0 = \{ (R4,2,\{R2,R3\}), (R5,2,\{R2,R3\}), (R6,2,\{R2,R3\}) \}$

$Cq0' = \{ (R0,0,\{\}), (R1,0,\{\}), (R2,3,\{\}), (R3,3,\{\}) \}$

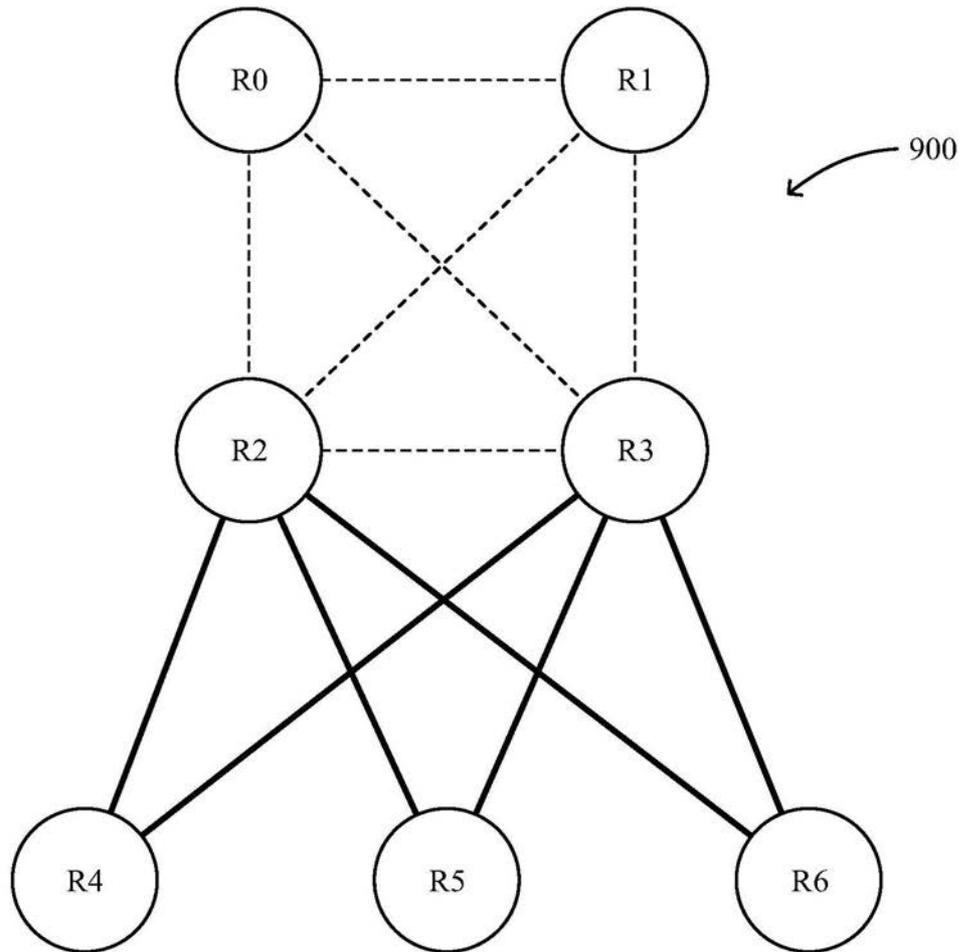


图9

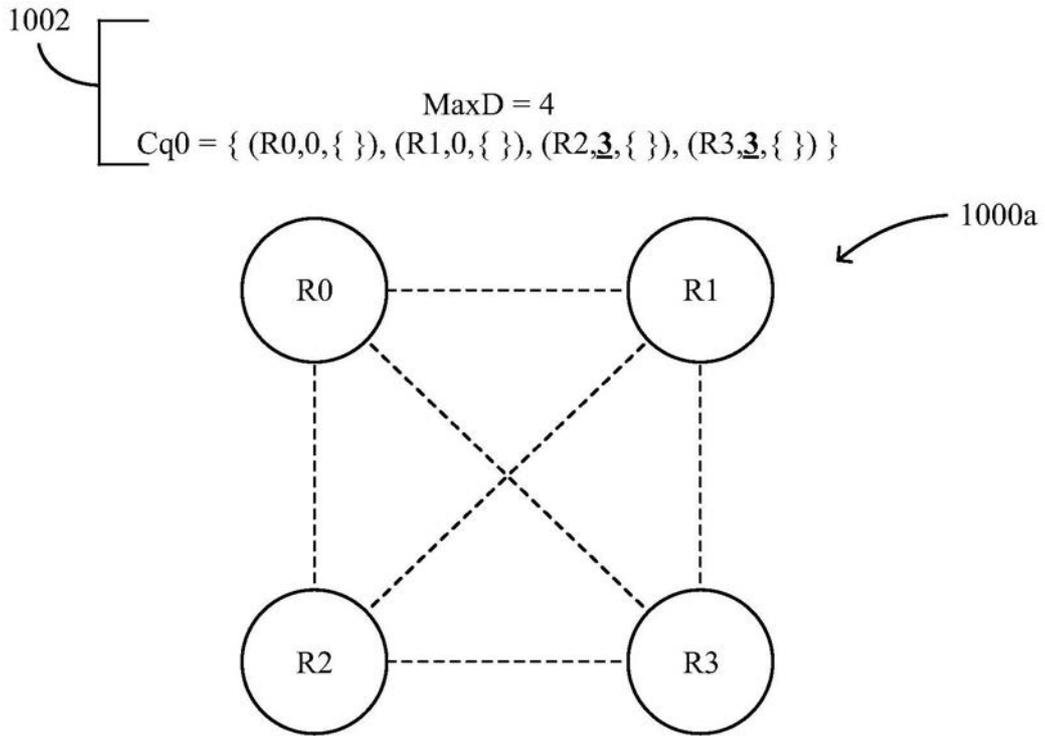


图10A

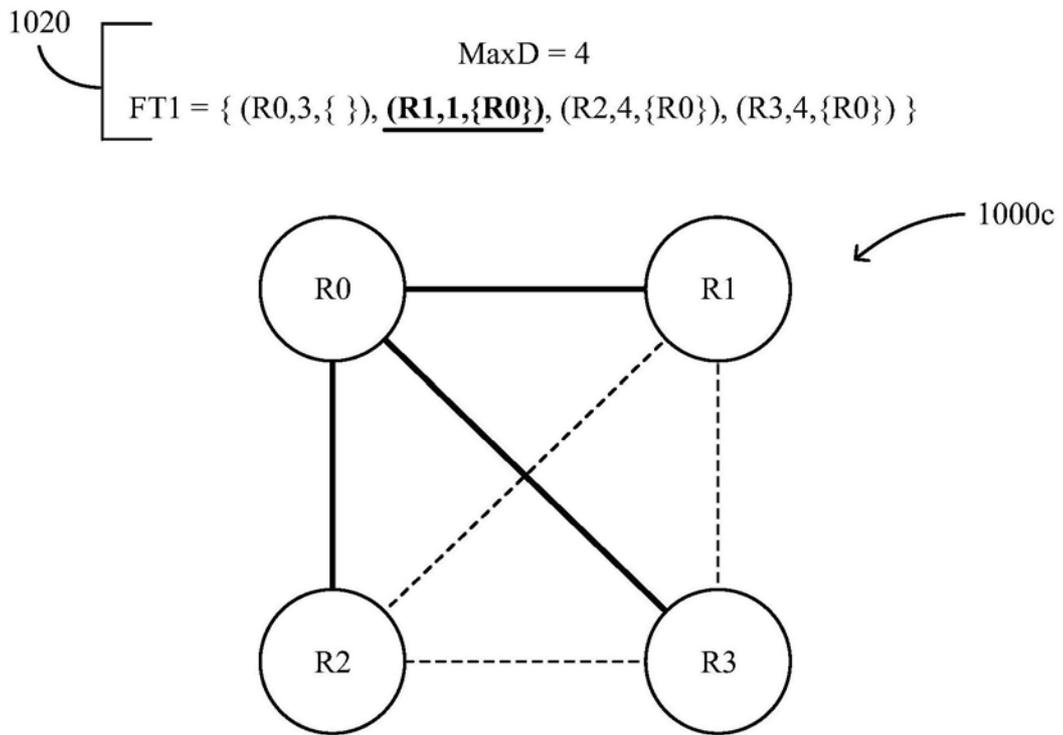


图10B

1120 {
 MaxD = 5
 FT1 = { (R0,3,{ }), (**R1,1,{R0}**), (R2,4,{R0}), (R3,4,{R0}) }
 FT1' = { (R0,3,{ }), (R1,**2**,{R0, **R2**}), (R2,**5**,{R0}), (R3,4,{R0}) }

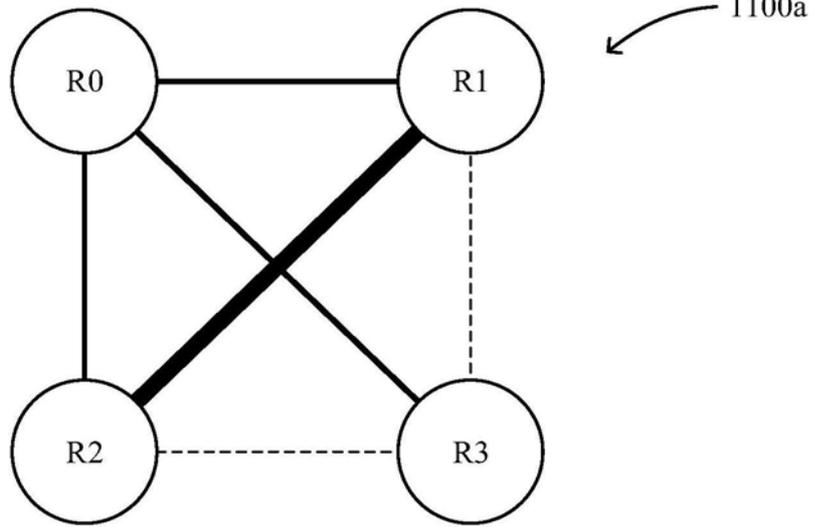


图11A

1130 {
 MaxD = 5
 FT = { (R0,3,{ }), (R1,2,{R0, R2}), (R2,5,{R0}), (R3,4,{R0}) }

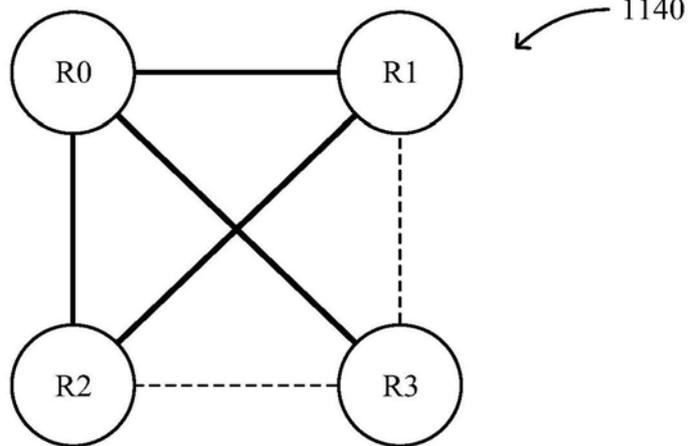


图11B

1210



$FT = \{ (R0,3,\{ \}), (R1,2,\{R0, R2\}), (R2,5,\{R0\}), (R3,4,\{R0\}), (R4,2,\{R2,R3\}), (R5,2,\{R2,R3\}), (R6,2,\{R2,R3\}) \}$

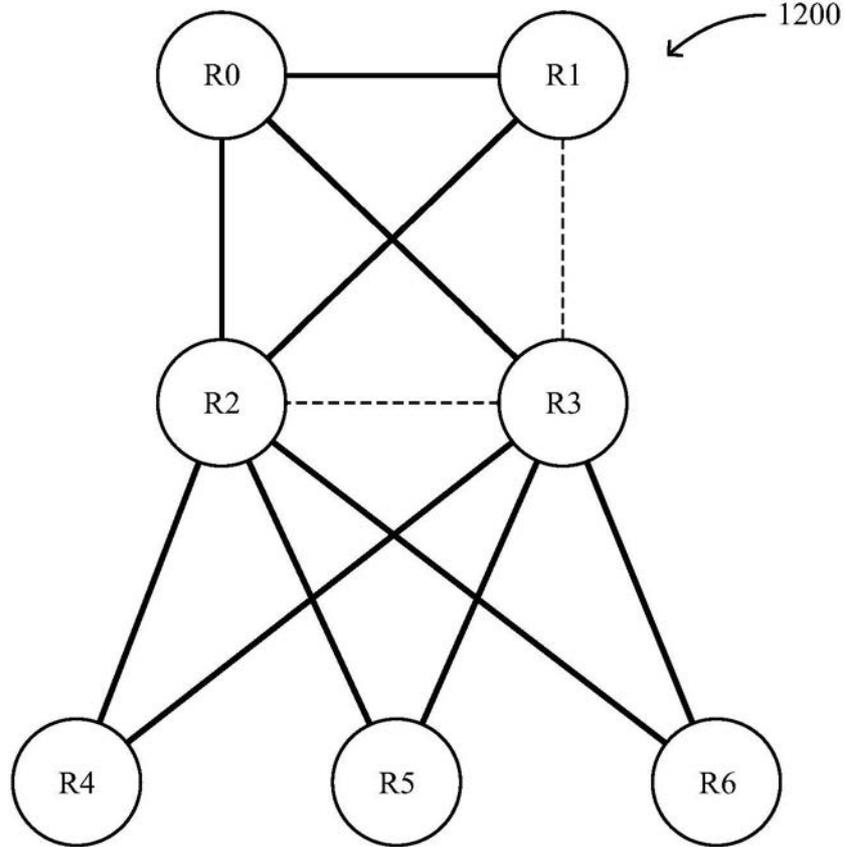


图12

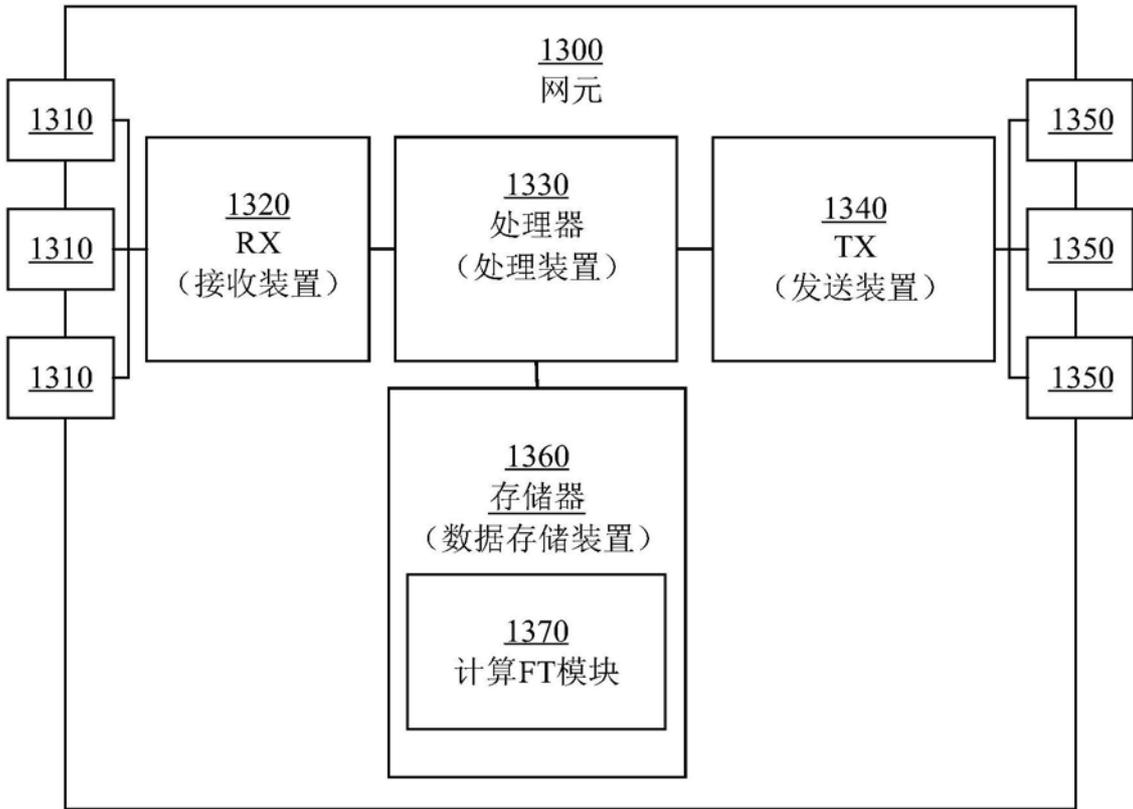


图13

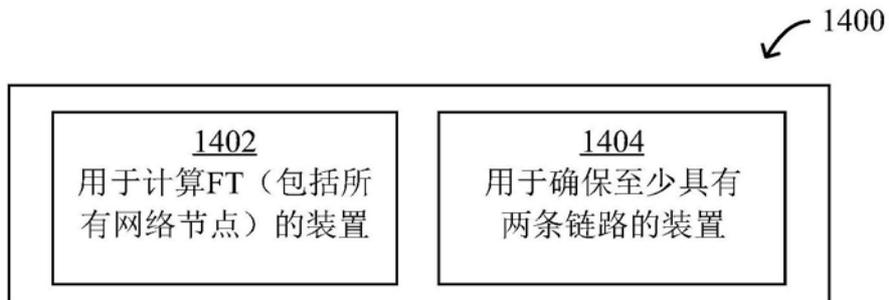


图14



图15

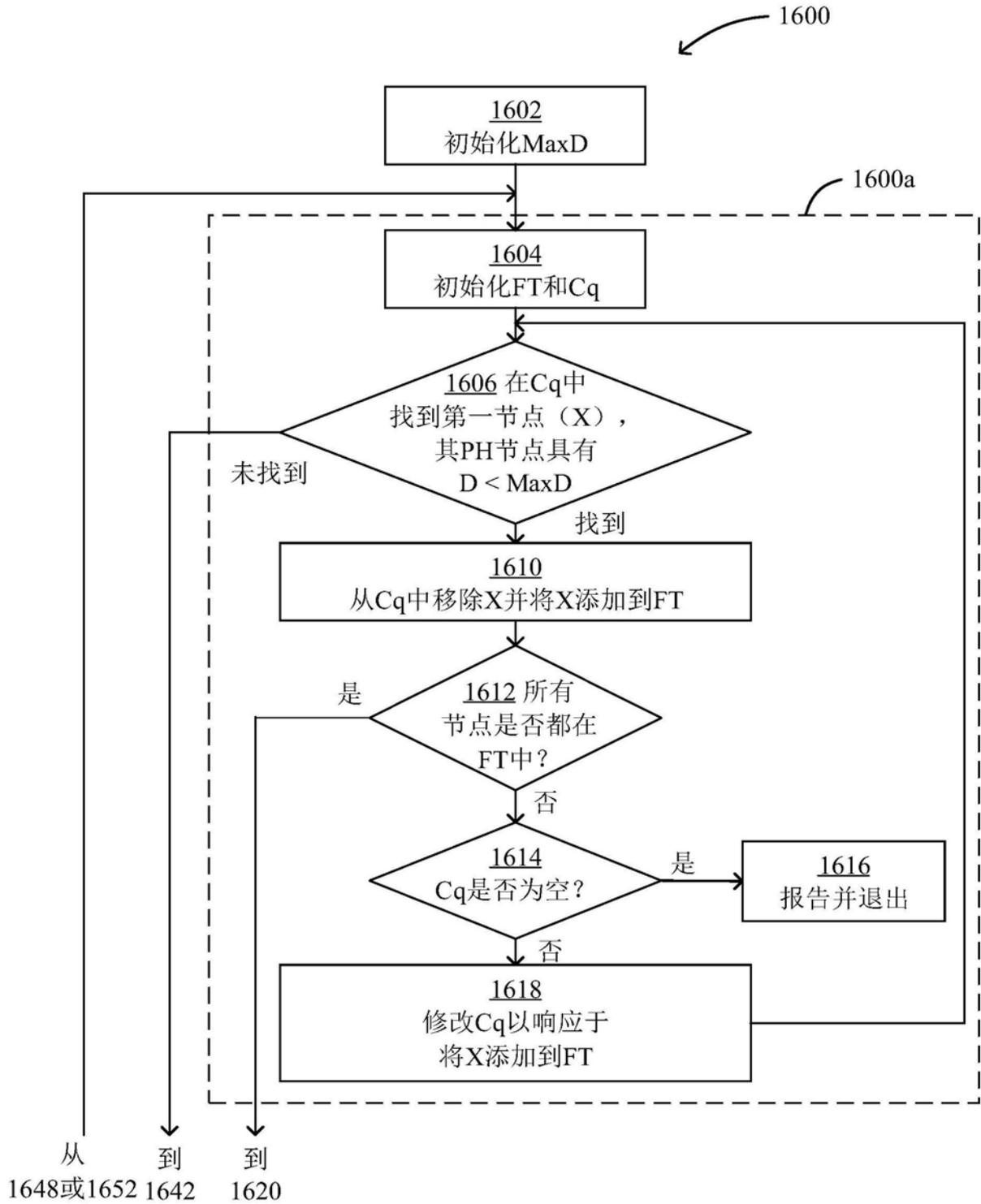


图16A

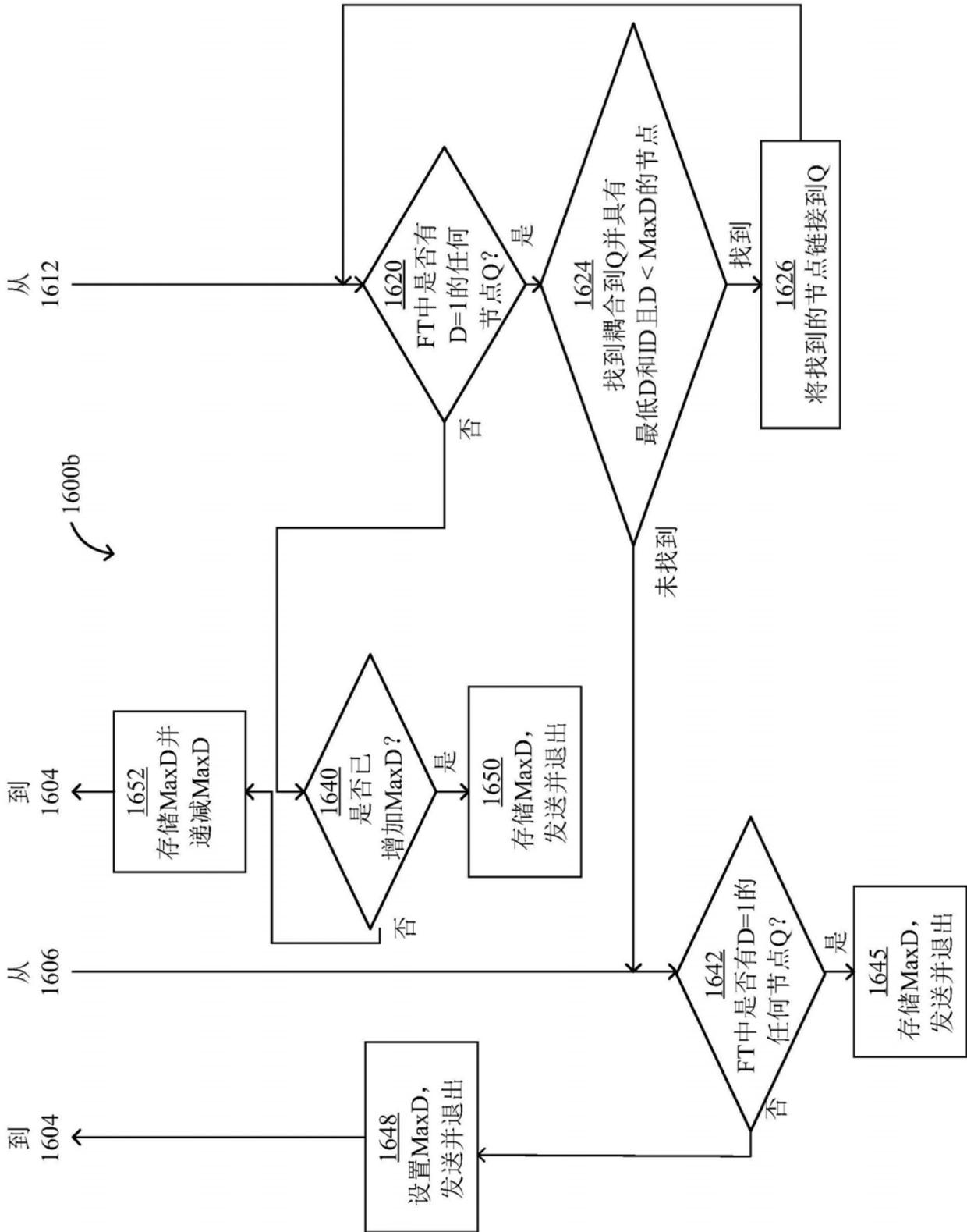


图16B