



(12) 发明专利申请

(10) 申请公布号 CN 103973566 A

(43) 申请公布日 2014. 08. 06

(21) 申请号 201410174422. 0

(22) 申请日 2008. 12. 11

(30) 优先权数据

11/964478 2007. 12. 26 US

(62) 分案原申请数据

200880127674. 0 2008. 12. 11

(71) 申请人 北方电讯网络有限公司

地址 加拿大魁北克省

(72) 发明人 J. 恰鲍特 D. 艾伦 N. 布拉格

P. 阿什伍德史密斯

(74) 专利代理机构 中国专利代理(香港)有限公司

司 72001

代理人 张懿 刘春元

(51) Int. Cl.

H04L 12/721 (2013. 01)

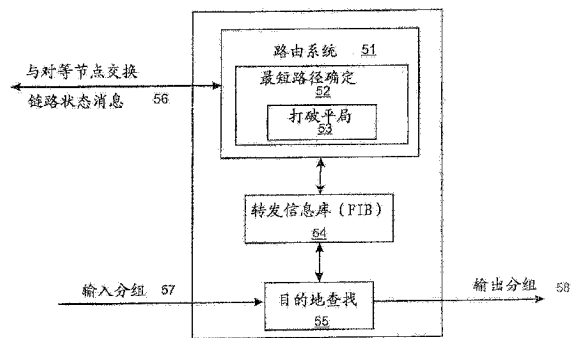
权利要求书7页 说明书15页 附图6页

(54) 发明名称

最短路径确定中的打破平局

(57) 摘要

通过比较多个端到端路径中的每一个的、节点标识符的经排序的集合来实现等开销最短(最低开销)路径之间一致的打破平局决定。可替换地,随着最短路径树被建立,通过使用该树的分叉支路的节点标识符对等开销的路径进行选择,可以在传输过程中实现相同的结果。两种变形都允许对等开销的路径进行一致的选择而不管所述最短路径在网络中的什么地方被计算。这确保了任何两个节点之间的业务流在正方向和反方向上都始终沿着相同的路径通过网络。



1. 一种确定转发信息供在分组转发网络的转发节点处转发分组时使用的方法,所述网络的每个节点具有唯一的节点标识符,所述方法包括:

确定所述网络的第一节点和第二节点之间的最短路径;

确定多个最短路径具有相等的开销的时间;

形成定义所述路径中的节点的集合的节点标识符集合;

使用第一排序标准将每个节点标识符集合排序并且按该顺序串联所述节点标识符以形成路径标识符,其中所述第一排序标准独立于与所述节点标识符对应的节点在所述路径中出现的顺序;以及

通过比较所述路径标识符而选择所述多个等开销的路径中的至少一个等开销的路径。

2. 根据权利要求1所述的方法,其中确定所述多个最短路径具有相等的开销的时间包括确定所述多个最短路径按照至少一个链路度量在彼此的理想偏移以内具有相等的开销的时间。

3. 根据权利要求1所述的方法,其中所述第一排序标准是递增的字典式顺序和递减的字典式顺序中的一个。

4. 根据权利要求1所述的方法,其还包括使用第二排序标准将所述多个路径标识符排序成有序列表;并且其中选择所述多个等开销的路径中的至少一个等开销的路径包括选择在所述路径标识符的有序列表中第一个或最后一个出现的等开销的路径。

5. 根据权利要求4所述的方法,其中所述第二排序标准是递增的字典式顺序和递减的字典式顺序中的一个。

6. 根据权利要求4所述的方法,其中通过比较所述路径标识符而选择所述多个等开销的路径中的至少一个等开销的路径包括以下各项中的一个:

通过使用两个不同的第一排序标准来形成两个路径标识符集合并且使用共同的第二排序标准以从所述两个集合中的每一个各选择一个的方式选择两个不同的路径标识符而选择所述等开销的路径中的两个;

通过使用共同的第一排序标准来形成路径标识符并且使用两个不同的第二排序标准来选择两个不同的路径标识符而选择所述等开销路径中的两个。

7. 根据权利要求4所述的方法,其还包括通过以下步骤来选择所述等开销的路径中的四个:

使用两个不同的第一排序标准来形成路径标识符并且使用共同的第二排序标准来创建路径标识符的两个相应的有序列表,路径标识符的每个有序列表与所述两个不同的第一排序标准中的相应的一个对应;以及

选择与在路径标识符的所述两个有序列表中的每一个中第一个和最后一个出现的路径标识符对应的等开销的路径。

8. 一种供在分组转发网络中使用的转发节点,其包括:

处理器;以及

存储可由处理器执行以执行以下操作的指令的处理器可读介质:

确定所述网络的第一节点和第二节点之间的最短路径;

确定多个最短路径具有相等的开销的时间;

为每个等开销的路径形成定义所述路径中的节点的集合的节点标识符集合;

使用第一排序标准将每个节点标识符集合排序并且按该顺序串联所述节点标识符以形成路径标识符,其中所述第一排序标准独立于与所述节点标识符对应的节点在所述路径中出现的顺序;以及

通过比较所述路径标识符而选择所述多个等开销的路径中的至少一个等开销的路径。

9. 根据权利要求 8 所述的转发节点,其中确定所述多个最短路径具有相等的开销的时间包括确定所述多个最短路径按照至少一个链路度量在彼此的理想偏移以内具有相等的开销的时间。

10. 根据权利要求 8 所述的转发节点,其中所述第一排序标准是递增的字典式顺序和递减的字典式顺序中的一个。

11. 根据权利要求 8 所述的转发节点,其还包括可执行以使用第二排序标准将所述多个路径标识符排序成有序列表的指令;并且其中可执行以选择所述多个等开销的路径中的至少一个等开销的路径的指令包括可执行以选择在所述路径标识符的有序列表的一端处出现的等开销的路径的指令。

12. 根据权利要求 11 所述的转发节点,其中所述第二排序标准是递增的字典式顺序和递减的字典式顺序中的一个。

13. 根据权利要求 11 所述的转发节点,其中可执行以通过比较所述路径标识符而选择所述多个等开销的路径中的至少一个等开销的路径的指令包括以下各项中的一个:

可执行以通过使用两个不同的第一排序标准来形成两个路径标识符集合并且使用共同的第二排序标准以从所述两个集合中的每一个各选择一个的方式选择两个不同的路径标识符而选择所述等开销的路径中的两个的指令;

可执行以通过使用共同的第一排序标准来形成路径标识符并且使用两个不同的第二排序标准来选择两个不同的路径标识符而选择所述等开销路径中的两个的指令。

14. 根据权利要求 11 所述的转发节点,其中所述指令还包括可执行以通过以下步骤来选择所述等开销的路径中的四个的指令:

使用两个不同的第一排序标准和共同的第二排序标准来创建路径标识符的两个相应的有序列表,路径标识符的每个有序列表与所述两个不同的第一排序标准中的相应的一个对应;以及

选择与在路径标识符的所述两个有序列表中的每一个中第一个和最后一个出现的路径标识符对应的等开销的路径。

15. 一种确定转发信息供在分组转发网络的转发节点处转发分组时使用的方法,所述网络的每个节点具有唯一的节点标识符,所述方法包括:

确定所述网络的第一节点和第二节点之间的最短路径;

在确定所述最短路径的同时,确定多个路径具有等开销的时间,每个等开销的路径包括从所述等开销的路径所共有的分叉节点分出的支路;

使用第一选择标准在每个分叉支路中选择节点标识符以形成相应的支路标识符,所述第一选择标准独立于与所述节点标识符对应的节点在所述支路中出现的顺序;以及

通过比较所述支路标识符而选择所述多个支路中的至少一个支路。

16. 根据权利要求 15 所述的方法,其中确定第一节点和第二节点之间的最短路径的步骤通过迭代地形成最短路径树来执行。

17. 根据权利要求 15 所述的方法,其中选择节点标识符包括在每个分叉支路中选择最佳地达到所述第一选择标准的相应节点标识符。

18. 根据权利要求 17 所述的方法,其中:

所述分叉支路在汇聚节点处汇聚;并且

对于每个分叉支路,选择最佳地达到所述第一选择标准的相应节点标识符包括:

记录在该支路上与所述汇聚节点相邻的相应节点的节点标识符;

沿所述支路从所述汇聚节点朝所述分叉节点一个节点接一个节点地进行处理,直到到达与所述分叉节点相邻的相应节点,同时将每个节点标识符与先前记录的节点标识符比较并且记录那两个节点标识符中达到所述第一选择标准的节点标识符;以及

在到达与所述分叉节点相邻的相应节点时,将所记录的节点标识符记录为用于所述支路的相应的支路标识符。

19. 根据权利要求 15 所述的方法,其中:

在第一节点和第二节点之间确定最短路径的步骤以及确定两个路径具有等开销的时间的步骤使用 Dijkstra 算法来执行;并且

通过比较所述两个路径所共有的节点的标识符而选择所述等开销的路径中的一个的步骤被集成到所述 Dijkstra 算法中。

20. 一种供在分组转发网络中使用的转发节点,其包括:

处理器;以及

存储可由处理器执行以执行以下操作的指令的处理器可读介质:

确定所述网络的第一节点和第二节点之间的最短路径;

在形成所述最短路径树的同时,确定多个路径具有等开销的时间,每个等开销的路径包括从所述等开销的路径所共有的分叉节点分出的支路;

使用第一选择标准在每个分叉支路中选择节点标识符以形成支路标识符,所述第一选择标准独立于与所述节点标识符对应的节点在所述支路中出现的顺序;以及

通过比较所述支路标识符而选择所述多个支路中的至少一个支路。

21. 根据权利要求 20 所述的转发节点,其中可执行以确定第一节点和第二节点之间的最短路径的指令可执行以迭代地形成最短路径树。

22. 根据权利要求 20 所述的转发节点,其中所述指令还包括可执行以在每个分叉支路中选择最佳地达到所述第一选择标准的相应节点标识符。

23. 根据权利要求 22 所述的转发节点,其中:

所述分叉支路在汇聚节点处汇聚;并且

对于每个分叉支路,所述转发节点还包括可执行以执行以下操作的指令:

记录在该支路上与所述汇聚节点相邻的相应节点的节点标识符;

沿所述支路从所述汇聚节点朝所述分叉节点一个节点接一个节点地进行处理,直到到达与所述分叉节点相邻的相应节点,同时将每个节点标识符与先前记录的节点标识符比较并且记录那两个节点标识符中达到所述第一选择标准的节点标识符;以及

在到达与所述分叉节点相邻的相应节点时,将所记录的节点标识符记录为用于所述支路的相应的支路标识符。

24. 根据权利要求 20 所述的转发节点,其中:

可执行以在第一节点和第二节点之间确定最短路径以及确定两个路径具有等开销的时间的指令实现 Dijkstra 算法 ; 并且

可执行以通过比较所述两个路径所共有的节点的标识符而选择所述等开销的路径中的一个的指令被集成到所述 Dijkstra 算法中。

25. 一种确定转发信息供在分组转发网络的转发节点处转发分组时使用的方法, 所述网络的每个节点具有唯一的节点标识符, 所述方法包括 :

由所述转发节点确定所述网络的第一节点和第二节点之间的最短路径 ;

由所述转发节点在确定所述最短路径的同时确定多个路径具有相等的开销的时间 ; 以及

由所述转发节点选择所述等开销的路径中的一个, 被选择的等开销的路径具有以下特性 : 如果在每个等开销的路径中分配给节点的节点标识符根据第一排序标准被排序并且按该顺序被串联以得到用于每个等开销的路径的路径标识符, 其中所述第一排序标准独立于与所述节点标识符对应的节点在所述路径中出现的顺序, 并且结果得到的路径标识符根据第二排序标准被排序以得到路径标识符的有序列表, 则所述被选择的等开销的路径在所述路径标识符的有序列表中是第一个或最后一个。

26. 根据权利要求 25 所述的方法, 其中所述第一排序标准是字典式排序。

27. 根据权利要求 25 所述的方法, 其中所述第二排序标准是字典式排序。

28. 根据权利要求 25 所述的方法, 其中所述确定所述网络的第一节点和第二节点之间的最短路径的步骤包括迭代地形成最短路径树。

29. 根据权利要求 25 所述的方法, 其包括以下各项中的一个 :

选择所述等开销的路径中具有以下特性的两个等开销的路径 : 如果在每个等开销的路径中分配给节点的节点标识符根据两个不同的第一排序标准被排序并且按该顺序被串联以得到两个路径标识符集合, 每个路径标识符集合对于每个等开销的路径具有相应的路径标识符, 其中每个第一排序标准独立于与所述节点标识符对应的节点在所述路径中出现的顺序, 并且结果得到的路径标识符根据共同的第二排序标准被排序以得到路径标识符的两个有序列表, 所述两个被选择的等开销的路径中的每一个具有处于所述路径标识符的有序列表中的一个的一个末端的相应标识符 ; 以及

选择所述等开销的路径中具有以下特性的两个等开销的路径 : 如果在每个等开销的路径中分配给节点的节点标识符根据第一排序标准被排序并且按该顺序被串联以得到路径标识符集合, 所述路径标识符集合对于每个等开销的路径具有相应的路径标识符, 其中所述第一排序标准独立于与所述节点标识符对应的节点在所述路径中出现的顺序, 并且结果得到的路径标识符根据两个不同的第二排序标准被排序以得到路径标识符的两个有序列表, 所述两个被选择的等开销的路径中的每一个具有处于所述路径标识符的有序列表中的一个的一个末端的相应标识符。

30. 根据权利要求 25 所述的方法, 其还包括选择所述等开销的路径中具有以下特性的四个等开销的路径 : 如果在每个等开销的路径中分配给节点的节点标识符根据两个不同的第一排序标准被排序并且按该顺序被串联以得到两个路径标识符集合, 每个路径标识符集合对于每个等开销的路径具有相应的路径标识符, 其中每个第一排序标准独立于与所述节点标识符对应的节点在所述路径中出现的顺序, 并且结果得到的路径标识符根据共同的第

二排序标准被排序以得到路径标识符的两个有序列表,所述四个被选择的等开销的路径中的每一个具有处于所述路径标识符的有序列表中的一个的一个末端的相应标识符。

31. 一种用于分组转发网络的转发节点,在所述分组转发网络中多个节点中的每一个具有唯一的节点标识符,所述转发节点包括:

处理器;以及

存储可由所述处理器执行以执行以下操作的指令的处理器可读介质:

确定所述网络的第一节点和第二节点之间的最短路径;

在确定所述最短路径的同时确定多个路径具有相等的开销的时间;以及

选择所述等开销的路径中的一个,被选择的等开销的路径具有以下特性:如果在每个等开销的路径中分配给节点的节点标识符根据第一排序标准被排序并且按该顺序被串联以得到用于每个等开销的路径的路径标识符,其中所述第一排序标准独立于与所述节点标识符对应的节点在所述路径中出现的顺序,并且结果得到的路径标识符根据第二排序标准被排序以得到路径标识符的有序列表,则所述被选择的等开销的路径在所述路径标识符的有序列表中是第一个或最后一个。

32. 根据权利要求 31 所述的转发节点,其中所述第一排序标准是字典式排序。

33. 根据权利要求 31 所述的转发节点,其中所述第二排序标准是字典式排序。

34. 根据权利要求 31 所述的转发节点,其中可由所述处理器执行以确定所述网络的第一节点和第二节点之间的最短路径的指令包括可执行以迭代地形成最短路径树的指令。

35. 根据权利要求 31 所述的转发节点,其中所述指令包括以下各项中的一个:

可执行以选择所述等开销的路径中具有以下特性的两个等开销的路径的指令:如果在每个等开销的路径中分配给节点的节点标识符根据两个不同的第一排序标准被排序并且按该顺序被串联以得到两个路径标识符集合,每个路径标识符集合对于每个等开销的路径具有相应的路径标识符,其中每个第一排序标准独立于与所述节点标识符对应的节点在所述路径中出现的顺序,并且结果得到的路径标识符根据共同的第二排序标准被排序以得到路径标识符的两个有序列表,所述两个被选择的等开销的路径中的每一个具有处于所述路径标识符的有序列表中的一个的一个末端的相应标识符;以及

可执行以选择所述等开销的路径中具有以下特性的两个等开销的路径的指令:如果在每个等开销的路径中分配给节点的节点标识符根据第一排序标准被排序并且按该顺序被串联以得到路径标识符集合,所述路径标识符集合对于每个等开销的路径具有相应的路径标识符,其中所述第一排序标准独立于与所述节点标识符对应的节点在所述路径中出现的顺序,并且结果得到的路径标识符根据两个不同的第二排序标准被排序以得到路径标识符的两个有序列表,所述两个被选择的等开销的路径中的每一个具有处于所述路径标识符的有序列表中的一个的一个末端的相应标识符。

36. 根据权利要求 31 所述的转发节点,其中所述指令包括可执行以选择所述等开销的路径中具有以下特性的四个等开销的路径的指令:如果在每个等开销的路径中分配给节点的节点标识符根据两个不同的第一排序标准被排序并且按该顺序被串联以得到两个路径标识符集合,每个路径标识符集合对于每个等开销的路径具有相应的路径标识符,其中每个第一排序标准独立于与所述节点标识符对应的节点在所述路径中出现的顺序,并且结果得到的路径标识符根据共同的第二排序标准被排序以得到路径标识符的两个有序列表,所

述四个被选择的等开销的路径中的每一个具有处于所述路径标识符的有序列表中的一个的一个末端的相应标识符。

37. 一种确定转发信息供在分组转发网络的转发节点处转发分组时使用的方法,所述网络的每个节点具有唯一的节点标识符,所述方法包括:

由所述转发节点确定所述网络的第一节点和第二节点之间的最短路径;

由所述转发节点确定两个路径具有相等的开销的时间;以及

由所述转发节点选择所述等开销的路径中的一个,被选择的等开销的路径具有以下特性:其包括不是为所述等开销的路径共有的节点,该节点或者具有不是为所述等开销的路径共有的所有节点中的最大节点标识符,或者具有不是为所述等开销的路径共有的所有节点中的最小节点标识符。

38. 根据权利要求 37 所述的方法,其中所述选择所述等开销的路径中的一个的步骤包括选择包括在所述两个路径中的所有节点标识符的集合之中不是为所述两个路径共有的最小节点标识符。

39. 根据权利要求 37 所述的方法,其中所述选择所述等开销的路径中的一个的步骤包括选择包括在所述两个路径中的所有节点标识符的集合之中不是为所述两个路径共有的最大节点标识符。

40. 根据权利要求 37 所述的方法,其中所述确定所述网络的第一节点和第二节点之间的最短路径的步骤通过迭代地形成最短路径树来执行。

41. 根据权利要求 37 所述的方法,其中所述选择所述等开销的路径中的一个的步骤通过仅考虑所述等开销的路径的分叉支路上的节点而局部地执行。

42. 根据权利要求 37 所述的方法,其中所述选择所述等开销的路径中的一个的步骤包括通过以下步骤在具有第一支路的第一等开销的路径和具有从所述第一支路分出的第二支路的第二等开销的路径之间选择:

通过根据第一标准选择所述第一支路上的节点的节点标识符为所述第一支路定义第一支路标识符;

通过根据所述第一标准选择所述第二支路上的节点的节点标识符为所述第二支路定义第二支路标识符;以及

通过对所述第一和第二支路标识符应用第二标准来选择所述第一和第二路径中的一个。

43. 根据权利要求 42 所述的方法,其中定义第一和第二支路标识符以及选择第一和第二路径中的一个的步骤对于两个等开销的路径在其处汇聚的每个节点被执行。

44. 根据权利要求 42 所述的方法,其还包括在向所述第一和第二支路在其处分叉的节点回溯的同时记录在所述分叉支路中的每一个中根据所述第一标准被选择的节点标识符。

45. 根据权利要求 44 所述的方法,其还包括在每个逆向的步骤处,将所记录的节点标识符与在该步骤处所遇到的新的节点标识符比较并且记录达到所述第一选择标准的节点标识符

46. 根据权利要求 37 所述的方法,其中:

所述确定第一节点和第二节点之间的最短路径以及确定两个路径具有相等的开销的时间的步骤使用 Dijkstra 算法来执行;并且

所述选择所述等开销的路径中的一个的步骤被集成到所述 Dijkstra 算法中。

47. 根据权利要求 37 所述的方法,其中:

不止两个路径是等开销的路径;并且

所述选择所述等开销的路径中的一个的步骤包括对于一连串的、所述等开销的路径的对,比较不是为所述等开销的路径的对共有的节点的节点标识符。

48. 一种供在分组转发网络中使用的转发节点,其包括:

处理器;以及

存储可由所述处理器执行以执行以下操作的指令的处理器可读介质:

确定所述网络的第一节点和第二节点之间的最短路径;

在形成所述最短路径树的同时确定两个路径具有相等的开销的时间;以及

选择所述等开销的路径中的一个,被选择的等开销的路径具有以下特性:其包括不是为所述等开销的路径共有的节点,该节点或者具有不是为所述等开销的路径共有的所有节点中的最大节点标识符,或者具有不是为所述等开销的路径共有的所有节点中的最小节点标识符。

49. 根据权利要求 48 所述的转发节点,其中可执行以选择所述等开销的路径中的一个的指令包括可执行以选择包括在所述两个路径中的所有节点标识符的集合之中不是为所述两个路径共有的最小节点标识符的指令。

50. 根据权利要求 48 所述的转发节点,其中可执行以选择所述等开销的路径中的一个的指令包括可执行以选择包括在所述两个路径中的所有节点标识符的集合之中不是为所述两个路径共有的最大节点标识符的指令。

51. 根据权利要求 48 所述的转发节点,其中可执行以确定第一节点和第二节点之间的最短路径的指令包括可执行以迭代地形成最短路径树的指令。

52. 根据权利要求 48 所述的转发节点,其中:

不止两个路径是等开销的路径;并且

可执行以选择所述等开销的路径中的一个的指令包括对于一连串的、所述等开销的路径的对,可执行以比较不是为所述等开销的路径的对共有的节点的节点标识符的指令。

最短路径确定中的打破平局

[0001] 本申请是申请日为 2008 年 12 月 11 日、申请号为 200880127674.0、发明名称为“最短路径确定中的打破平局”的专利申请的分案申请。

技术领域

[0002] 本发明涉及在诸如以太网的分组转发通信网络中在多种可能性中一致地选择路径,诸如等开销最短路径。

背景技术

[0003] 在分组转发通信网络中,节点可以得知有关网络的拓扑结构的情况,并且可以基于它获取的关于该拓扑结构的信息决定它将如何向其它网络节点中的每一个传递业务(traffic)。选择路径的主要根据是路径开销,其可以按照节点之间的跳跃(hop)的数量或者通过诸如连接节点的链路的带宽等某个其它度量来规定,或者通过这两者来规定。开放式最短路径优先(OSPF)和中间系统到中间系统(IS-IS)是被广泛使用的链路状态协议,它们基于每个节点的对路径开销的通告来建立最短路径。这些协议通常不尝试在多个等开销的路径之间打破平局(tie-break)。相反,它们通常跨若干等开销的路径传播业务。传播算法没有被规定并且可以从路由器到路由器不同。可替换地,它们可以对单个路径进行局部选择而不考虑与由其它路由器进行的选择的一致性。因此,无论发生何种情况,不能保证流的反方向(reverse direction)使用正方向(forward direction)所使用的路径。

[0004] 诸如组播开放式最短路径优先协议(MOSPF)的组播(multicast)路由协议依赖于网络中的每个路由器构造相同的最短路径树。由于这个原因,MOSPF实现基于链路类型、LAN相比点对点(LAN vs. point-to-point)以及路由器标识符的打破平局方案以确保产生一样的树。但是,把打破平局决定建立在具有最大的标识符的父代(parent)之上意味着一般而言反向的流所使用的路径将不会与正向的流所使用的路径相同。

[0005] 生成树协议(生成树协议(STP)、快速生成树协议(RSTP)、多生成树协议(MSTP))是在任意的拓扑结构中创建无回路的生成树的方法。生成树协议由网络中的每个节点执行。所有生成树协议都使用基于(桥标识符,端口标识符)的局部打破平局决定在等开销的路径之间进行选择。在生成树中,首选选择根节点,然后相对于那个根通过所有节点来构造树。因此,尽管所有路径对于离开和返回业务是对称的(根据定义,简单树(simple tree)使得这成为仅有的可能的构造),但是选择过程是慢的并且简单树的结构不能使用任何多余的容量。类似地,Radia Perlman的Rbridges提议使用父节点的标识符作为决胜局(tie-breaker)。

[0006] Mick Seaman在他给IEEE802.1工作组的“最短路径桥接”提议(<http://www.ieee802.org/1/files/public/docs2005/new-seaman-shortest-path-0305-02.pdf>)中描述了对快速生成树协议的简单的协议增强,其通过增加“截断矢量(cut vector)”来强制执行一致的打破平局决定。该提议使用每一节点的VID来标识每一节点的生成树。为了把需要由桥传送的所有信息放进单个合法的以太网帧,这种技术目前将以太网的大小限制为

32 个桥。

[0007] 图 1 示出,即使对于普通的网络的例子,基于父节点标识符的打破平局方法如何无法产生对称路径。在这个例子中,链路被认为具有等开销并且因此路径开销的确定仅考虑跳跃的数量。首先考虑计算从 A 到 B 的路径。当计算到达节点 2 时,将会发现等开销的路径的存在。有第一路径 (A-1-3-6) 和第二路径 (A-1-4-5)。如果打破平局算法基于具有最小的标识符的父节点来选择路径,则它将选择第二路径 (A-1-4-5),因为节点标识符 5 小于节点标识符 6。但是,现在考虑计算从 B 到 A 的路径。当计算到达节点 1 时,将会发现等开销的路径的存在。有第一路径 (B-2-6-3) 和第二路径 (B-2-5-4)。使用相同的打破平局标准,该打破平局算法选择第一路径 (B-2-6-3),因为节点标识符 3 小于节点标识符 4。因此,可以看到的是由节点 A 和 B 进行的最短路径计算提供不一致的结果。

[0008] 在诸如给 IEEE802.1aq 的提议“供应商链路状态桥接 (PLSB)”的一些新兴的协议中有为单播和未知 / 组播业务两者保持跨网络转发的一致性以及在流的正方向和反方向两者上都使用共同的路径的要求。因此,重要的是当在等开销的路径之间打破平局时节点可以一致地得出相同的决定。此外,理想的是节点可以用最少量的处理努力来执行打破平局。

发明内容

[0009] 本发明的第一方面提供确定转发信息供在分组转发网络的第一节点处转发分组时使用的方法。该方法确定所述第一节点和所述网络的第二节点之间的最短路径并且确定多个最短路径具有大体上相等的开销的时间。该方法为每个大体上等开销的路径形成节点标识符的集合,所述节点标识符定义路径中的节点的集合并且然后使用第一排序标准将节点标识符的每个集合排序以形成路径标识符。所述第一排序标准独立于节点标识符在所述路径中出现的顺序。该方法然后通过比较所述路径标识符而在所述多个等开销的路径之间进行选择。所述网络的每个节点具有唯一的节点标识符。

[0010] 有利地,所述第一排序标准是递增的字典式顺序 (lexicographic order) 或者是递减的字典式顺序,但也可以使用创建节点的全序集合 (totally ordered set) 的任何排序标准。

[0011] 优选地,所述方法还包括使用第二排序标准将所述多个路径标识符排序成有序列表。类似地,所述第二排序标准可以是递增的字典式顺序、递减的字典式顺序或者创建路径标识符的全序集合的任何排序标准。

[0012] 本发明的另一个方面提供确定转发信息供在分组转发网络的第一节点处转发分组时使用的方法。该方法包括通过迭代地形成最短路径树来确定所述第一节点和所述网络的第二节点之间的最短路径并且在形成所述最短路径树的同时确定多个路径具有等开销的时间,每个等开销的路径包括从所述等开销的路径所共有的分叉节点 (divergency node) 分出的支路。该方法使用第一选择标准在每个分叉支路中标识节点标识符以形成支路标识符并且通过比较所述支路标识符而在所述多个支路之间进行选择。

[0013] 有利地,该方法使用全排序 (total ordering) 标准来比较和选择每个支路中的节点标识符,诸如字典式顺序。

[0014] 有利地,该方法在向所述分叉节点回溯 (backtrack) 的同时记录所述分叉支路的

每一个中达到所述第一选择标准的节点标识符。这具有进一步简化计算和减少存储需求的优势。

[0015] 本发明的两个方面可以被用于通过使用不同的第一排序 / 选择标准和共同的第二排序 / 选择标准或者通过使用共同的第一排序标准 / 选择和不同的第二排序 / 选择标准来选择两个等开销的路径。可以通过在节点处一致地应用所述第一和第二排序 / 选择标准并且选择处于所述有序列表中的特定位置的标识符而以相似的方式选择三个或四个等开销的路径。

[0016] 本发明可以被用作决胜局 (tie-breaker) 以通过比较多个端到端路径中的每一个的、节点标识符的经排序的集合而在等开销的路径之间进行选择。可替换地, 已被发现的是随着最短路径树被建立, 局部于 (local to) 选择决定需要被做出的地方通过使用该树的分叉支路的节点标识符对等开销的路径进行选择可以在传输过程中 (on-the-fly) 实现相同的结果。这具有减少计算量并且减少需要被存储的数据的量的优势。支路可以在成对的基础上被比较以进一步减少计算量。随着网络的大小和复杂度增加, 这变得尤其重要。本发明的两种变形都具有允许对等开销的路径进行一致的选择而不管所述最短路径在网络中的什么地方被计算的重要特性。这确保了任何两个节点之间业务流在正方向和反方向上都始终沿着相同的路径通过网络。

[0017] 本发明不是打算要被限制于确定最短路径的任何特定方式, 包括: Dijkstra 算法、Folyd 算法或者可以被使用的任何其它合适的替代。

[0018] 本发明可以被用作具有完全相同的值的等开销的路径之间或者处于彼此的理想偏移以内的路径之间的决胜局, 这两种情况下均就链路度量或者跳跃的数量而言。这可能是实际情况中所希望的, 以在合格路径的集合之间增加多样性。例如, 以通常被要求用于在任何两个端点之间实现完全相等的开销的对称方式部署节点和链路可能不总是划算的。通过将该约束放宽为要求不同路线上的跳数 (hop count) 彼此相差一跳以内, 适度的非对称仍然可以产生合格的路线, 并且无回路的 (loop-free) 拓扑结构仍然被保证, 因为两次跳跃的差距是实现成环路径 (looping path) 所必要的最小值。

[0019] 将理解的是术语“最短路径”不被限制于仅基于距离来确定路径, 并且旨在包括可以被用于规定链路的“开销 (cost)”的任何度量或者度量的组合。度量的非穷举的列表为: 距离, 跳跃的数量、容量 (capacity)、速度、使用率 (usage)、可用性。

[0020] 等开销最短路径的选择不受不在被选择的路径上的网络的部分 (诸如出故障的节点或者链路) 的移除影响, 在这个意义上来说该方法是稳定的。

[0021] 有利地, 所述网络是以太网, 但是本发明可以被应用于其它类型的分组转发网络, 尤其是对于对称的业务路由路径有要求的那些分组转发网络。

[0022] 在此所描述的功能可以用软件、硬件或者这些的组合来实现。本发明可以通过合适地被编程的计算机或者任何形式的处理设备来实现。因此, 本发明的另一个方面提供用于实现所描述的方法中的任何一种的软件。该软件可以被存储在电子存储装置、硬盘、光盘或者其它机器可读存储介质上。该软件可以作为计算机程序产品在机器可读载体上被交付或者它可以通过网络连接被下载到节点。

[0023] 本发明的另外的方面提供包括处理器的网络节点, 所述处理器被配置用于执行所描述的方法中的任何一种。

[0024] 本发明的另外的方面提供节点的网络,所述节点各自一致地应用所描述的方法在等开销的路径之间进行选择。

附图说明

- [0025] 将参考附图仅通过示例的方式来描述本发明的实施例,其中:
- [0026] 图 1 示出具有等开销的路径的网络拓扑结构;
- [0027] 图 2 示出可以在其中实现本发明的分组转发网络的例子;
- [0028] 图 3 示意性地示出在图 2 的桥接节点中的一个处的设备。
- [0029] 图 4 示出打破平局决定的局部性 (locality);
- [0030] 图 5 到 7 示出网络拓扑结构的例子以示意最短路径的计算;
- [0031] 图 8 示出网络拓扑的另外的例子以示意最短路径的计算;
- [0032] 图 9 到 11 示出图 8 所示的网络拓扑结构的最短路径计算的打破平局步骤;
- [0033] 图 12 示出双归宿 (dual-home) 到网状网络上的节点的例子。
- [0034] 图 13A 和 13B 示意本发明的打破平局方法的特性。

具体实施方式

[0035] 图 2 示出可以在其中实现本发明的受链路状态协议控制的以太网 10 的例子并且图 3 示意性地示出在节点 41-48 中的一个处的设备。形成网状网络的节点 (也被称作桥或者桥接节点) 41-48 彼此交换链路状态通告 (advertisement) 56。这经由链路状态路由系统的被充分理解的机制来实现。路由系统模块 51 使用链路状态路由协议与网络中的对等节点交换关于网络拓扑结构的信息 56。信息的这种交换允许节点生成对网络拓扑结构同步的观察。在每个节点处,最短路径确定模块 52 计算最短路径树,该最短路径树确定到每个其它节点的最短路径。由模块 52 确定的最短路径被用于向转发信息库 54 填充用于引导业务通过网络的条目。如将在下文中更详细地描述的,当模块 52 遇到多个等开销的路径时情况将出现。打破平局模块 53 以一致的方式选择等开销的路径中的一个 (或多个)。在正常的操作中,分组在节点处被接收 57 并且目的地查找模块 55 使用 FIB54 来确定所接收的分组应当通过其被转发 58 的端口 (或者在组播分配的情况中为多个端口)。如果在 FIB54 中没有有效的条目,则分组可以随后被丢弃。将理解的是图 3 所示的模块仅用于示意的目的并且可以通过合并或者在节点的模块之间分配功能来实现,这将被本领域的技术人员理解。

[0036] 可以使用各种最短路径算法来确定给定的节点是否在给定的一对桥之间的最短路径上。可以在节点 41-48 中实现所有点对间最短路径算法 (all-pairs shortest path algorithm) 来计算节点对之间的最短路径,诸如 Floyd 算法 [R. Floyd:算法 97 (最短路径) (Algorithm 97 (shortest path)), ACM 通信 (Communications of the ACM), 7:345, 1962] 或者 Dijkstra 的单源最短路径算法 [E. W. Dijkstra:对与图论相关的两个问题的记录 (A note on two problems in connexion with graphs), 数学学报 (Numerical Mathematics), 1:269-271, 1959]。应当理解的是还可以使用任何合适的最短路径算法。最短路径算法所使用的链路度量可以是静态的或者可以动态地被修改以将业务工程信息 (traffic engineering information) 考虑在内。例如,链路度量可以包括对诸如容量、速度、使用率和可用性等开销的测量。

[0037] 作为对问题的引入, 首先将描述对可以在等开销的路径之间做出一致决定的打破平局算法的要求。要求的列表在下面的表 1 中列出:

[0038]

#	要求	描述
1	完备性	打破平局算法必须始终能够在两个路径之间选定
2	交换性	$\text{tiebreak}(a, b) = \text{tiebreak}(b, a)$
3	结合性	$\text{tiebreak}(a, \text{tiebreak}(b, c)) = \text{tiebreak}(\text{tiebreak}(a, b), c)$
4	对称性	$\text{tiebreak}(\text{reverse}(a), \text{reverse}(b)) = \text{reverse}(\text{tiebreak}(a, b))$
5	局部性	$\text{tiebreak}(\text{concat}(a, c), \text{concat}(b, c)) = \text{concat}(\text{tiebreak}(a, b), c)$

[0039] 表 1

[0040] 打破平局算法的要素是始终“起作用 (work)”。无论该算法被提供以怎样的路径集合, 该算法应当始终能够选定一个并且仅选定一个路径。首要地, 打破平局算法应当是完备的 (1)。对于一致的打破平局, 该算法必须产生相同的结果而不管等开销的路径被发现以及打破平局被执行的顺序。也就是说, 打破平局算法应当是可交换 (2) 以及可结合 (3) 的。在三个路径之间打破平局必须产生相同的结果而不管路径对被考虑的顺序的要求 (3) 不是很明显, 但是它对于一致的结果绝对是必要的, 因为取决于该计算通过网络的方向按不同的顺序发现等开销的路径。打破平局算法必须是对称的 (4), 即打破平局算法必须产生相同的结果而不管路径的方向: 两个节点 A 和 B 之间的最短路径必须是 B 和 A 之间的最短路径的反向。最后, 局部性是最短路径的非常重要的特性 (5), 该特性被路由系统充分利用。局部特性简单地来说就是: 最短路径的子路径也是最短路径。最短路径的这个看似普通的特性在使用基于目的地的转发的分组网络中具有重要的应用。在这些网络中, 在沿路径的中间节点处的转发决定唯一地基于分组的目的地地址, 而不是它的源地址。因此, 为了生成它的转发信息, 节点仅需要计算从它本身到所有其它节点的最短路径并且所产生的转发信息的量随网络中的节点数量线性地而不是二次地 (quadratically) 增长。为了允许基于目的地的转发, 打破平局算法因此必须保持最短路径的局部特性: 由打破平局算法选择的最短路径的子路径必须是由该打破平局算法选择的最短路径。

[0041] 对计算效率的考虑对打破平局算法提出了另一看似不同的要求: 等开销的路径一被发现该算法就应当能够做出打破平局决定。图 4 示意了这一点。中间节点 I 由两个等开销的路径 p 和 q 连接到节点 A, 并且由另一对等开销的路径 r 和 s 连接到节点 B。因此在节点 A 和 B 之间有四个等开销的路径: p+r、p+s、q+r、q+s, 它们全部经过节点 I。随着对从 A 到 B 的最短路径的计算的进行, 将首先发现 A 和 I 之间的等开销的子路径的存在。为了避免不得不推进 (carry forward) 这两个路径的信息, 第二等开销的最短子路径的存在一被发现该打破平局算法就应当能够在它们之间选定。在中间节点处做出的打破平局决定将最终影响计算的结果。通过排除节点 A 和 I 之间的两个子路径 p 和 q 中的一个, 该算法根据进一步的考虑移除了节点 A 和 B 之间的四个最短路径中的两个。相似地, 在反方向上, 该打破平

局算法将在做出最后确定之前在子路径 r 和 s 之间选定。这些局部的决定必须彼此一致，而且特别地，如果路径要以相同的方式被延伸，则在两个等开销的路径之间的抉择应当保持相同。例如，在图 3 所画的情况中，打破平局算法应当验证以下四个恒等式 (identity)：

[0042] $\text{tiebreak}(\text{concat}(p, r), \text{concat}(q, r)) = \text{concat}(\text{tiebreak}(p, q), r)$

[0043] $\text{tiebreak}(\text{concat}(p, s), \text{concat}(q, s)) = \text{concat}(\text{tiebreak}(p, q), s)$

[0044] $\text{concat}(p, \text{tiebreak}(r, s)) = \text{tiebreak}(\text{concat}(p, r), \text{concat}(p, s))$

[0045] $\text{concat}(q, \text{tiebreak}(r, s)) = \text{tiebreak}(\text{concat}(q, r), \text{concat}(q, s))$

[0046] 事实证明，对称性 (4) 和局部性 (5) 这两个条件都是必要的并且足以保证打破平局算法将做出一致的局部决定，这一事实可以被充分利用以在有等开销的最短路径的情况下产生单源最短路径算法的非常有效的实现。

[0047] 表 1 所列出的要求列表并非意在穷举，并且有最短路径的其它特性可以被包括在表 1 中。例如，如果不是最短路径的部分的链路从图中被移除，则该最短路径不受影响。同样地，如果不是被选择的路径的部分的链路从图中被移除并且即使该链路是等开销的路径中被打破平局算法否决的一些路径的部分，该算法在多个等开销的路径之间的选择不受影响。

[0048] 现在将描述一致的打破平局算法的第一实施例。这个算法通过为每个路径形成路径标识符开始。路径标识符是通过网络的路径所遍历的每个节点的标识符的有序列表。按字典式顺序将节点标识符排序。路径标识符是经排序的节点标识符的由此产生的串联 (concatenation)。图 5 示出网络的例子，该网络具有端节点 A、B 和中间节点 0-9。节点 A 和 B 之间的第一路径 (沿图 5 的最上面) 遍历具有节点标识符 A-0-5-6-1-4-8-B 的节点。在按升序的字典式顺序将节点标识符的列表排序之后，该路径可以由路径标识符 014568AB 来表示。这种构造确保了路径及其反向将具有相同的路径标识符。此外，由于算法仅处理最短路径或者接近最短的路径，只有两个路径 - 直接路径和对应的反向路径 - 能够共享标识符。最后，所述打破平局算法仅选择具有最小 (或最大的) 路径标识符的路径。该算法可以被总结为：

[0049] 1) 根据第一排序标准将路径中的节点的标识符的集合排序，该第一排序标准实现节点标识符的集合的全排序。优选的第一排序标准是递增或者递减的字典式顺序；

[0050] 2) 串联经排序的节点标识符的集合来创建路径标识符；

[0051] 3) 根据第二排序标准将路径标识符排序，该第二排序标准实现路径标识符的集合的全排序。优选的第二排序标准是递增或者递减的字典式顺序；

[0052] 4) 选择其路径标识符在路径标识符的经排序的集合的一端 (第一个或最后一个) 处出现的路径。有利地，该步骤选择在路径标识符的经排序的集合中第一个出现的路径标识符。

[0053] 执行这个算法的网络中的每个节点一致地使用相同的排序标准并且选择在路径标识符的集合中相同的约定 (agreed) 位置处的路径以便选择相同的路径。

[0054] 术语“字典式顺序，”意思是按标识符的大小的顺序来排列节点标识符的集合。所以，如果节点标识符是字母，则节点标识符的集合按字母顺序 A、B、C、D... 等被排列；如果节点标识符是数字，则节点标识符的集合按数字顺序被排列。明显地，这种方案可以适用于标注节点的任何方式以及标识符类型的任何组合。例如，数字和字母的混合可以通过相对

于字母约定数字的顺序（例如首先将数字排序，然后将字母排序）来排序。可替换地，每个字符可以被给予它的美国信息交换标准代码（ASCII）的码并且可以按递增（递减）的顺序将该 ASCII 码排序。每个节点使用相同的协定（convention）以用相同的方式将路径的节点标识符排序。这个算法将产生一致的结果，因为在路径（严格来说是由路径和它的反向构成的对）和它的标识符之间有一对一的映射，并且有对路径标识符的全排序。

[0055] 再次参考图 5，在排序之后，由路径标识符 014568AB 来表示节点 A 和 B 之间最上面的路径。类似地，节点 A 和 B 之间的第二路径遍历节点 A-0-7-9-1-4-8-B 并且在排序之后这可以由路径标识符 014789AB 来表示。最后，节点 A 和 B 之间的第三路径（沿图 5 的最下面）遍历节点 A-0-7-9-2-3-8-B 并且在排序之后这可以由路径标识符 023789AB 来表示。在约定的方向上，所述打破平局算法比较经排序的路径标识符的每个元素。在这个例子中，将被使用的协定是当路径标识符在特定方向上（例如从左到右）被比较时，每个节点选择经排序的路径标识符中最小的那个。对于这三个等开销的路径，经排序的路径标识符为：

[0056] 014568AB

[0057] 014789AB

[0058] 023789AB

[0059] 从标识符左侧的元素开始，所有三个路径标识符都以‘0’开始。下一个元素为“1，或‘2’，所以只需要再进一步考虑最上面两个标识符。到达第四个元素，“0145...”小于“0147...”，因此最上面的路径被选择。IS-IS 和以太网中实际的节点标识符由六个 8 位的字节组成并且通常被写成十六进制的串，诸如 00-e0-7b-c1-a8-c2。还可以使用节点的别名，只要它们一致地被使用。

[0060] 图 6 示出简单的网络拓扑结构以示意不同排序标准的效果。通过具有节点标识符 1-8 的四个等开销的路径连接两个节点 X、Y。现在将描述四个可能的选项：

[0061] ●按升序将节点 ID 排序；按升序将路径 ID 排序；选择第一个（最小的）路径 ID。如果每个路径中的节点标识符按大小的升序被排序（例如具有节点 1,7 的最上面的路径变成 17），则那样给出路径标识符 17, 28, 35, 46。按大小的升序排列这些路径标识符并且选择有序列表中的第一个路径标识符获得选择第一个（最上面的）路径的结果，该路径具有节点 1 和 7。

[0062] ●按升序将节点 ID 排序；按升序将路径 ID 排序；选择最后一个（最大的）路径 ID。这个选项获得选择最后一个（最下面的）路径的结果，该路径具有节点 4 和 6。

[0063] ●按降序将节点 ID 排序；按升序将路径 ID 排序；选择第一个（最小的）路径 ID。按大小的降序将每个路径中的节点标识符排序给出路径标识符 (71, 82, 53, 64)。按大小的升序排列这些路径标识符给出 (53, 64, 71, 82) 并且选择有序的列表中的第一个（最小的）路径标识符获得选择第三个路径的结果，该路径具有节点 3 和 5。

[0064] ●按降序将节点 ID 排序；按升序将路径 ID 排序；选择最后一个（最大的）路径 ID。这个选项获得选择第二个路径的结果，该路径具有节点 8 和 2。

[0065] 如将在下文中更详细地描述的，存在一些情况，其中所希望的是节点应用多个不同的排序和 / 或选择标准来选择多个等开销的路径。

[0066] 到目前为止，本说明书假设该算法是非局部的（non-local）并且打破平局在所有等开销的路径已被找到之后被执行。但是，已经发现这个算法的局部形式可以通过仅考虑

分叉支路上的节点而产生相同的结果。实际上,打破平局的结果仅取决于分叉支路中最小的节点标识符的相对位置。一致的打破平局算法的第二实施例可以被总结为:

[0067] 1) 在第一路径的分叉支路中找到达到第一选择标准的节点标识符。这可以被认为是该第一路径的支路标识符;

[0068] 2) 在第二路径的分叉支路中找到达到第一选择标准的节点标识符。这可以被认为是该第二路径的支路标识符;

[0069] 3) 使用第二选择标准来选择所述路径中的一个,其在通过步骤(1)和(2)所选择的支路标识符上运算(operate)。

[0070] 对于第一选择标准优选的选项是找到这样的节点标识符,即当使用诸如字典式顺序(递增或者递减的字典式顺序)的全排序方案来排列节点标识符时,该节点标识符是第一个(或者最后一个)。如将在下文中所解释的,该方案不一定要汇集支路中的节点标识符的总的集合,然后将该集合排序。相反,该方案可以利用对字典式顺序的认识迭代地比较节点标识符对。类似地,对于第二选择标准优选的选项是找到这样的支路标识符,即当使用诸如字典式顺序(递增或者递减的字典式顺序)的全排序方案来排列支路标识符时,该支路标识符是第一个(或者最后一个)。

[0071] 再次参考图 6 的拓扑结构,节点 X 和 Y 之间的四个等开销的路径可以表示来自父节点 X 的四个等开销的分叉支路。所述打破平局算法需要选择这四个支路中的一个。有四个可能的选项:

[0072] ● 标识每个支路中最小的节点 ID。这产生 (1,2,3,4) 作为支路标识符。然后,标识支路标识符中最小的那个。这获得选择第一个(最上面的)路径的结果,该路径具有节点 1 和 7。

[0073] ● 标识每个支路中最小的节点 ID。然后,标识支路标识符中最大的那个。这个选项获得选择最后一个(最下面的)路径的结果,该路径具有节点 4 和 6。

[0074] ● 标识每个支路中最大的节点 ID。这产生 (5,6,7,8) 作为支路标识符。然后,标识支路标识符中最小的那个。这获得选择具有节点 3 和 5 的路径的结果。

[0075] ● 标识每个支路中最大的节点 ID。然后,标识支路标识符中最大的那个。这个选项获得选择具有节点 2 和 8 的路径的结果。

[0076] 如将在下文中更详细地描述的,存在一些情况,其中所希望的是节点应用多个不同的排序和/或选择标准来选择多个等开销的路径。

[0077] 这个算法可以通过简单的比较而非常容易且有效地被实现。图 7 示出另一网络拓扑结构。所述方法的局部形式将在节点 13 处开始,并且进而查找从节点 15 引出的两个分叉支路。该方法探索直到节点 16 为止的两个独立的路径,这两个路径在节点 16 处再次汇聚。在这点上,该方法检查两个支路中的每一个的节点标识符。对于第一支路,节点标识符为:10,14,17,21 并且对于第二支路,节点标识符为:11,12,19,20。具有最小的标识符(10)的支路是最上面的路径的部分。该方法可以仅从节点 16 向节点 15 回溯,保持对在每个支路中找到的最小的节点标识符的跟踪。在每个逆向的步骤处,该方法将到目前为止所找到的最小的节点标识符与在该步骤处所遇到的新的节点标识符比较。最小的节点标识符被存储。当该方法已经回溯至节点 15 时,两个最小的值(在最上面的支路中为 10,在较低的支路中为 11)可以仅被相互比较以找到具有最小的节点标识符的支路。因此,最上面的支路

被选择,其形成最上面的路径的部分。当执行这种打破平局时,这两个分叉支路所共有的路径的部分被忽略。

[0078] 用于在网络中查找最短开销路径 (shortest cost path) 的最常用的算法之一是 Dijkstra 算法 [Dijkstra59]。它解决的问题是在路径的长度被定义为正的 (positive) 逐跳 (hop-by-hop) 链路开销的总和的情况下查找从图中的一个点 (源节点或根节点) 到所有可能的目的地的最短路径。这个问题有时候被称为单源最短路径问题。对于图 $G = (N, L)$, 其中 N 是节点的集合并且 L 是连接它们的链路的集合, Dijkstra 算法使用通常被称为 TENT 的优先级队列以按到源节点的距离递增的顺序来访问节点。实现 Dijkstra 算法所需要的其它数据结构为:

[0079] 距离 (Distance): 从源节点到每个节点的最短距离的最佳估计的数组。

[0080] 父代 (Parent): 每个节点的前驱的数组。

[0081] 下面的文本描述已知的 Dijkstra 算法, 并且描述其可以如何被修改以在多个等开销的路径被发现时执行打破平局。在此描述 Dijkstra 算法是因为它是最常用的最短路径查找算法之一。但是, 将理解的是其它算法同样可以被使用。初始化阶段将每个节点的距离设置为无穷大, 除了源节点本身。源节点的距离被设置为零并且它的父代被设置为空 (Null), 因为它是树的根。在计算开始时, 优先级队列仅包含源节点。随着算法的进行, 节点在从源节点到它们的路径被找到时被加入优先级队列。在节点和源节点之间的最短路径已被找到之后, 按到源节点的距离递增的顺序将这些节点从优先级队列中取出。该算法在可从源节点到达的所有节点都已循环通过优先级队列时停止。当优先级队列 TENT 不为空时, 该算法执行以下步骤:

[0082] 1) 在 TENT 中查找与源节点最接近的节点 N 并将其从 TENT 中移除。

[0083] 2) 对于连接到 N 的每个节点, 如果通过使 N 成为节点的父代将减

[0084] 小它到源的距离, 则将该节点的父代改为 N , 将该节点的距离设

[0085] 置为新的距离, 并且将该节点加入 TENT。

[0086] 当该算法完成时, 距离 (节点) 包含从源节点到节点的最短距离 (或者如果不能从源节点到达该节点则距离为无穷远) 并且父代 (节点) 包含生成树中节点的前驱 (除了源节点以及不能从源节点到达的节点)。只有当改变父代实际上减小了节点的距离时才更新节点的父代。这意味着, 如果在源节点和一些其它节点之间存在多个等开销最短路径, 则只有在算法的执行期间所遇到的第一个节点将被考虑。

[0087] 上面的步骤是 Dijkstra 算法的常规步骤。在这点上, Dijkstra 被修改以增加一致的打破平局步骤。上面的步骤 2 如下被修改:

[0088] 2) 为连接到节点 N 的每个节点进行以下操作:

[0089] 2a) 如果通过使 N 成为节点的父代将减小节点到源节点的距离, 则将该节点的父代改为 N , 将该节点的距离设置为新的距离, 并且将该节点加入 TENT。

[0090] 2b) 如果在使 N 成为节点的父代之后该节点到源节点的距离将保持相同, 则调用打破平局算法来确定该节点的父代是否应当被改变。在到达两个分叉支路的汇聚点时调用打破平局算法。例如, 考虑图 7 所示的拓扑结构, 如果 Dijkstra 算法从节点 13 开始, 从节点 15 引出的分叉支路被发现 (具有节点 10、14、17、21 的上部支路和具有节点 11、12、19、20 的下部支路), 并且这些分叉支路在节点 16 处汇聚。在节点 16 处, 打破平局算法将被调用

以在这两个支路之间进行选择。

[0091] 下面的伪代码示出了使用 TENT 集合的优先级队列实现对具有一致的打破平局的被修改的 Dijkstra 算法的实现。入列 (enqueue) 操作带有两个自变量, 即队列和节点, 并且根据节点到源节点的距离将它放在恰当的队列位置上。出列 (dequeue) 操作从队列中移除在队列最前面的节点, 即具有到源节点的最小距离的节点。

[0092]

```
for each Node in Network do Distance(Node) = Infinity;
Empty(Tent);
Distance(Source) = 0;
Parent(Source) = Null;
Node = Source;
do
  for each Link in OutgoingLinks(Node) do
    newDistance = Distance(Node) + Cost(Link);
    Child = EndNode(Link);
    if (newDistance < Distance(Child) do
      Distance(Child) = newDistance;
      Parent(Child) = Node;
      Enqueue(Tent, Child);
    else if (newDistance == Distance(Child) do
      Parent(Child) = TieBreak(Node, Parent(Child));
  while (Node = Dequeue(Tent));
```

[0093] 该打破平局算法通过回溯两个等开销的路径来运算, 分别从节点的当前父代以及新的候选父代开始, 一路返回至分叉点。两个分叉路径可能具有不同数量的跳跃的事实使情况稍微变复杂, 因为两个路径必须被回溯未知的、不等数量的跳跃。这个问题可以通过始终首先回溯两个路径中较长的那个或者当它们具有等开销时同时回溯两者来解决。可替换地, 这种困难可以通过确保当且仅当两个路径具有相同数量的跳跃时它们才将被认为具有等开销而完全被消除。这通过将跳数并入路径开销或者通过将跳数用作第一级的决胜局 (first order tie-breaker) 而容易地被实现。

[0094] 下面的伪代码示出了打破平局算法的实现, 其假定两个路径具有相同数量的跳跃 (并且因此它们的分叉支路也具有相同数量的跳跃)。打破平局函数带有处于两个相等的路径的末端的两个节点并且返回它们中的一个来指示它选择了这两个路径中的哪一个。

[0095]

```

old = oldParent;
new = newParent;
oldMinId = SysId(old);
newMinId = SysId(new);

while ((old=Parent(old)) != (new=Parent(new))) do
    tmp = SysId(old);
    if (tmp < oldMinId) do oldMinId = tmp;
    tmp = SysId(new);
    if (tmp < newMinId) do newMinId = tmp;

if (newMinId < oldMinId) return newParent;
else return oldParent;

```

[0096] 算法需要被执行的频率取决于应用。PLSB 基本上需要计算所有点对间最短路径（有时是其子集）。在这种情况下，Dijkstra 算法需要为网络中的所有节点（准确地说是除了一个节点以外的所有节点）被运行。Floyd 算法计算所有点对间最短路径，所以它仅需要被运行一次。其它应用可能仅要求计算较少数量的路径（例如，如果只要求一个最短路径，则 Dijkstra 算法将必须仅被运行一次，其中以路径的端点中的一个作为源）。

[0097] 图 8 示出网络的例子，该网络具有通过链路互连的节点 A-H、J。对于每一链路，与该链路相关联的度量被示出为链路上的整数值。在这个网络中在节点 A 和节点 B 之间有六个不同的、等开销的、最短路径。这些路径在下表中以它们各自的长度和路径标识符被示出：

[0098]

路径	AGDHB	AGCHB	AGCJB	AFCHB	AFCJB	AFEJB
长度	10	10	10	10	10	10
标识符	ABDGH	ABCGH	ABCGJ	ABCFH	ABCFJ	ABEFJ

[0099] 所有这六个路径具有相同的长度 10。打破平局算法的非局部形式将选择具有最小路径标识符 (ABCFH) 的那个路径，即路径 AFCHB。这节的剩余部分示出打破平局算法的局部形式如何通过随着等开销的路径和子路径在 Dijkstra 算法的执行期间被发现仅做出局部的打破平局决定而达到相同的结果。Dijkstra 算法为网络中的节点初始化距离和父代（前驱）的表。所有距离初始地被设置为无穷大，除了源节点，它的距离被设置为零。父代在这个阶段未被定义：

[0100]

节点	A	B	C	D	E	F	G	H	J
距离	0	∞	∞	∞	∞	∞	∞	∞	∞
父代	-	-	-	-	-	-	-	-	-

[0101] Dijkstra 算法还将它的优先级队列初始化为仅包含源节点 A :TENT = [(A,0)]。

[0102] Dijkstra 循环的第一次迭代选择 TENT 中的第一个并且仅有的一个节点,即节点 A。然后对于节点 A 的邻节点(即节点 F 和 G)中的每一个,其更新它们到源的距离并且使节点 A 成为它们的父代。最后,这两个节点被加入 TENT 优先级队列。

[0103] 在 Dijkstra 算法的这个第一次迭代期间,距离和父代的表变为:

[0104]

节点	A	B	C	D	E	F	G	H	J
距离	0	∞	∞	∞	∞	2	1	∞	∞
父代	-	-	-	-	-	A	A	-	-

[0105] 在第一次迭代结束时,优先级队列为:TENT = [(G,1), (F,2)]。

[0106] Dijkstra 循环的第二次迭代从优先级队列中移除具有最小距离的节点即节点 G。它更新了 G 的邻节点中还没有被处理的两个节点,即节点 C 和 D,并且将它们加入优先级队列:

[0107]

节点	A	B	C	D	E	F	G	H	J
距离	0	∞	5	4	∞	2	1	∞	∞
父代	-	-	G	G	-	A	A	-	-

[0108] 在第二次迭代结束时,优先级队列为:TENT = [(F,2), (D,4), (C,5)]。Dijkstra 循环的第三次迭代将节点 F 从优先级队列中移除。其更新节点 F 的两个邻节点,即节点 C 和 E,并且将节点 E 加入优先级队列(节点 C 已经在该队列中)。节点 C 的距离没有改变,但是在节点 A 和节点 C 之间有新的候选的相等路径,该路径经过节点 F。因此,必须调用所述打破平局算法以在经过节点 F 的这个新的路径和经过节点 G 的旧的路径之间选定。这在图 9 中示出。以节点 C 的新的候选父代(节点 F)和它的旧的父代(节点 G)来调用所述打破平局算法。oldMin 被设置为旧的父代的标识符 G 并且 newMin 被设置为新的父代的标识符 F。由于节点 F 和 G 共享相同的父代(节点 A),所以不执行回溯循环。所述打破平局仅比较 oldMin 和 newMin,并且由于 newMin = F < G = oldMin,所以 F 被选择为节点 C 的新的父代:

[0109]

节点	A	B	C	D	E	F	G	H	J
距离	0	∞	5	4	4	2	1	∞	∞
父代	-	-	F	G	F	A	A	-	-

[0110] 在第三次迭代结束时,优先级队列为:TENT = [(D,4), (E,4), (C,5)]。Dijkstra 循环的第四次迭代从优先级队列中移除具有距离 4 的两个节点中的一个,例如节点 D。D 的

两个邻节点中只有一个节点,即节点 H 被更新并且被加入优先级队列:

[0111]

节点	A	B	C	D	E	F	G	H	J
距离	0	∞	5	4	4	2	1	6	∞
父代	-	-	F	G	F	A	A	D	-

[0112] 在第四次迭代结束时,优先级队列为:TENT = [(E,4), (C,5), (H,6)]。Dijkstra 循环的第五次迭代将节点 E 从优先级队列中移除。E 的两个邻节点中只有一个节点,即节点 J 被更新并且被加入优先级队列。

[0113]

节点	A	B	C	D	E	F	G	H	J
距离	0	∞	5	4	4	2	1	6	6
父代	-	-	F	G	F	A	A	D	E

[0114] 在第五次迭代结束时,优先级队列为:TENT = [(C,5), (H,6), (J,6)]。Dijkstra 循环的第六次迭代将节点 C 从优先级队列中移除。C 的两个邻节点(节点 J 和 H)具有经过节点 C 到节点 A 的等开销的路径。所述打破平局算法因此必须分别为节点 J 和 H 被调用两次。对于节点 J,以新的潜在的父代(节点 C)和旧的父代(节点 E)来调用所述打破平局算法。oldMin 被设置为旧的父代的标识符 E 并且 newMin 被设置为新的父代的标识符 C。由于这两个节点 E 和 C 共享相同的父代(节点 F),所以不执行回溯循环。所述打破平局仅比较 oldMin 和 newMin 并且由于 newMin = C < E = oldMin,所以选择新的父代。因此,节点 J 的父代由节点 C 代替。这在图 10 中示出。

[0115] 对于节点 H,以新的潜在的父代(节点 C)和旧的父代(节点 D)来调用所述打破平局算法。oldMin 被设置为旧的父代的标识符 D 并且 newMin 被设置为新的父代的标识符 C。由于这两个节点具有不同的父代,所以这两个路径必须另外被回溯一次跳跃。D 的父代是 G 并且由于 G > oldmin (= D),所以 oldMin 不改变。C 的父代是 F 并且由于 F > newMin (= C),所以 newMin 也不改变。由于 F 和 G 共享相同的父代,即节点 A,所以回溯循环停止。所述打破平局算法然后比较 oldMin 和 newMin,并且由于 newMin = C < D = oldMin,节点 C 被选择变为节点 H 的新的父代。这在图 11 中示出。

[0116]

节点	A	B	C	D	E	F	G	H	J
距离	0	∞	5	4	4	2	1	6	6
父代	-	-	F	G	F	A	A	C	C

[0117] 在第六次迭代结束时,优先级队列为:TENT = [(H,6), (J,6)]。

[0118] Dijkstra 循环的第七次迭代从优先级队列中移除具有距离 6 的两个节点中的一

个,例如节点 H。H 的邻节点中只有一个,即节点 B 被更新并且被加入优先级队列:

[0119]

节点	A	B	C	D	E	F	G	H	J
距离	0	10	5	4	4	2	1	6	6
父代	-	H	F	G	F	A	A	C	C

[0120] 在第七次迭代结束时,优先级队列为:TENT = [(J,6), (B,10)]。

[0121] Dijkstra 循环的第八次迭代将节点 J 从优先级队列中移除。在 J 的邻节点中只有节点 B 需要被更新。它的距离不改变但是在节点 A 和节点 B 之间有新的候选的相等路径,该路径经过节点 J。

[0122] 以节点 B 的新的潜在的父代(节点 J)和旧的父代(节点 H)来调用所述打破平局算法。oldMin 被设置为旧的父代的标识符 H 并且 newMin 被设置为新的父代的标识符 J。由于这两个节点 H 和 J 共享相同的父代(节点 C),所以不执行回溯循环。所述打破平局仅比较 oldMin 和 newMin,并且由于 oldMin = H < J = newMin,所以旧的父代被选择并且节点 B 的父代保持相同。

[0123]

节点	A	B	C	D	E	F	G	H	J
距离	0	10	5	4	4	2	1	6	6
父代	-	H	F	G	F	A	A	C	C

[0124] 在第八次迭代结束时,优先级队列为:TENT = [(B,10)]。

[0125] 最后,Dijkstra 循环的最后一次迭代将节点 B 从队列中移除并且由于节点 B 的邻节点中没有可以被更新(节点 B 是离源节点 A 最远的节点),所以该算法终止。

[0126] 从节点 A 到节点 B 的最短路径的反向可以直接从父代的表中读出,从节点 B 开始并且沿父代前进直到到达节点 A :BHCFA。通过局部打破平局算法所选择的从节点 A 到节点 B 的最短路径因此是它的反向路径 :AFCHB。

[0127] 尽管在节点 A 和 B 之间有 6 个等开销的路径,但是在 Dijkstra 算法的执行期间所述局部打破平局仅被调用总计 4 次。在它的第一次调用时,该打破平局算法必须在子路径 AFC 和 AGC 之间选定。它选择了子路径 AFC,由此根据进一步的考虑将两个路径 AGCJB 和 AGCHB 排除。在它的第二次调用时,该打破平局算法必须在子路径 AFCJ 和 AFEJ 之间选定。它选择了子路径 AFCJ,由此根据进一步的考虑将第三个路径 AFEJB 排除。在它的第三次调用时,该打破平局算法必须在子路径 AGDH 和 AGCH 之间选定。它选择了子路径 AGCH,由此根据进一步的考虑将第四个路径 AGDHB 排除。最后,在它的第四次调用时,该打破平局算法必须在路径 AFCHB 和 AFCJB 之间选定。它排除了第五个路径 AFCJB 并且选择了路径 AFCHB 作为最后的解。

[0128] 用于负载传播的等开销多路径的选择

[0129] 在许多连网(networking)应用中,经常有利的是使用若干等开销的路径,尤其是如

果这可以以一致的方式来实现。通过使用所述打破平局算法的两种变形,有可能使用节点对之间的两个等开销的路径(在这两个路径存在的情况下)。图 12 示出常见的连网场景,其中边节点 X 和 Y 各自双归宿在核心节点 A、B、C、D 的全网状结构(full mesh)上。为有冗余,每个边节点被连接到两个核心节点,其中节点 X 被连接到核心节点 A 和 B 并且节点 Y 被连接到节点 C 和 D。每个核心节点被连接到所有其它核心节点,例如节点 A 被连接到节点 B、C 和 D。这个拓扑结构的问题是,如果在节点对之间只有一个最短路径被使用,则在正常情况下将浪费大量的访问容量(access capacity)。当在两个节点之间存在多个等开销最短路径时,所述打破平局算法的两种变形可以被用于一致地选择正好两个路径。由所有节点约定的任何协定可以被用于在等开销的路径之间进行选择。一种特别方便的协定是选择具有最小标识符的第一路径和具有最大的标识符的第二路径。在图 12 中,由于核心节点完全网状互连(fully meshed),在边缘节点 X 和 Y 之间存在四个等开销的路径:(X, A, C, Y), (X, A, D, Y), (X, B, C, Y), (X, B, D, Y)。所述打破平局算法的两种变形将选择这两个路径:

[0130] (X, min(A, B), min(C, D), Y) 以及

[0131] (X, max(A, B), max(C, D), Y)

[0132] 由于节点标识符是唯一的, $\min(A, B) \neq \max(A, B)$ 并且 $\min(C, D) \neq \max(C, D)$:这两个路径最大程度相异:它们仅共同具有它们的端点。在图 12 中,两个被选择的路径为路径 (X, A, C, Y) 和路径 (X, B, D, Y)。

[0133] 在上文中所描述的打破平局方法的重要特性之一是对网络的改变对打破平局的结果没有影响,所述对网络的改变不影响路径集合中打破平局需要为其从路径之间选择的那个。这种改变可能包括对不在被选择的路径上的网络的部分的移除,诸如出故障的节点或链路。另一重要特性是,当多个路径等开销的路径被使用时,一个路径中的故障不影响其它路径的稳定性。类似地,链路的增加将仅影响等开销的路径中的一个而不会同时影响两个。这对于网络的稳定性是重要的。

[0134] 图 13A 和 13B 示意了本发明的打破平局方法的其它重要特性:

[0135] ● 在有等开销的路径的情况下单个故障不会促成回路;

[0136] ● 故障不会既使回路闭合也使根节点的附接的点移位;

[0137] ● 故障不会产生更短的路径;

[0138] ● 所述打破平局算法防止等开销的路径的评级改变最短路径。

[0139] 图 13A 和 13B 用具有节点 A、B、C、D 和 R 的简单的网络拓扑结构来示意这些特性。考虑图 13A, R 和节点 A-D 的集合之间的最短路径使用链路 R-A。从节点 A 到达节点 C 有对两个等开销的支路的抉择。使用在上文中所描述的打破平局方法中的一个,支路 A-B-C 被一致地选择而不是支路 A-D-C。类似地,在反方向上,链路 C-B-A 被一致地选择而不是 C-D-A。图 13B 示出当链路 R-A 出故障时在稍后的时间点处的情况。节点 R 现在通过次最佳链路 R-C 连接到节点 A-D 的集合。从节点 C 到达节点 A 有对两个等开销的支路的抉择。同样地,支路 C-B-A 被一致地选择而不是支路 C-D-A。在不使用这种一致的打破平局算法的情况下,回路 A-B-C-D-A 可能在链路 R-A 中的故障之后产生,其中节点 A 和 B 在它们的行为上表现为缓慢且杂乱而节点 C 和 D 表现灵活。这种特性对于为组播转发保证无回路特别有用。

[0140] 本发明不受在本文中所描述的实施例的限制,其可以被修改或者改变而不脱离本发明的范围。

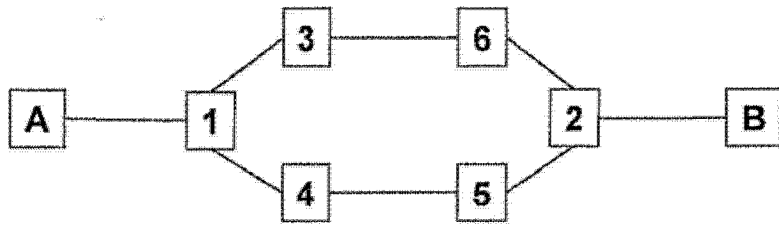


图 1

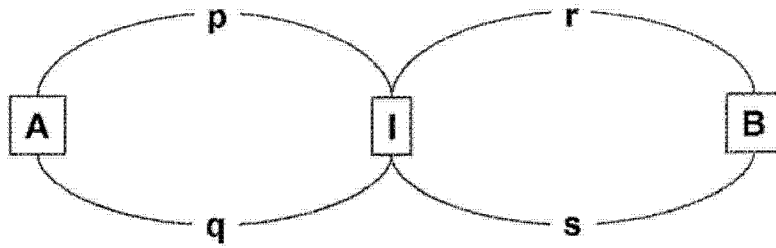


图 4

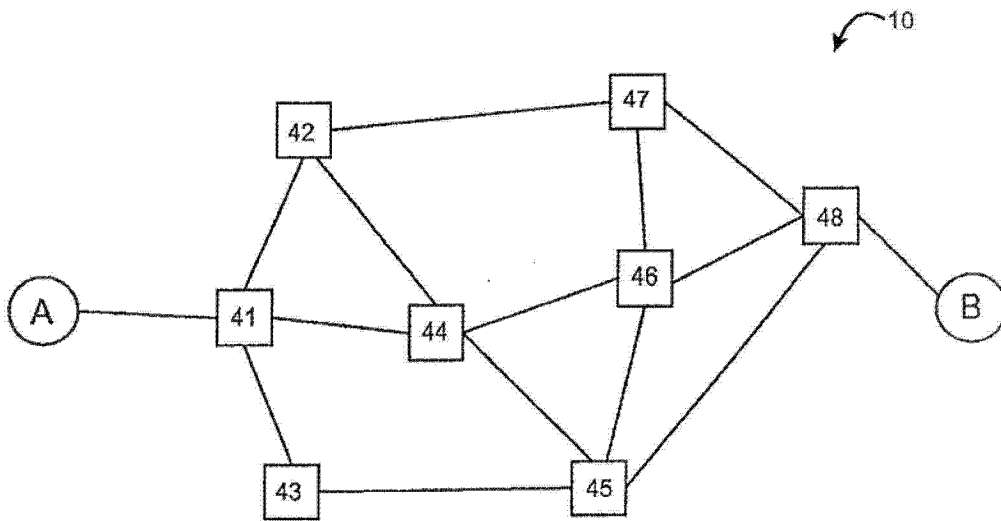


图 2

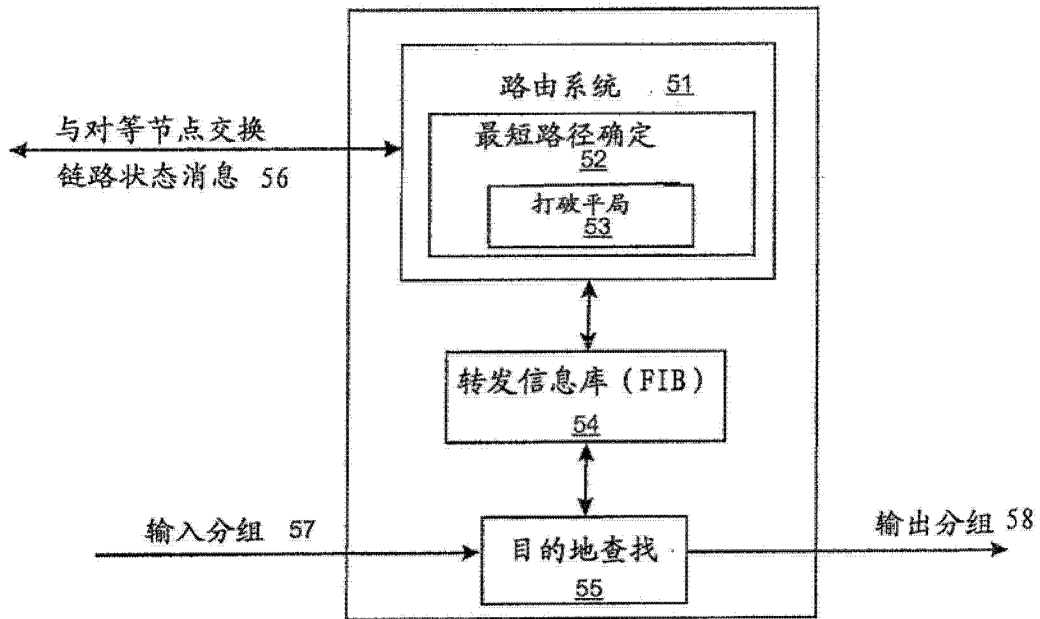


图 3

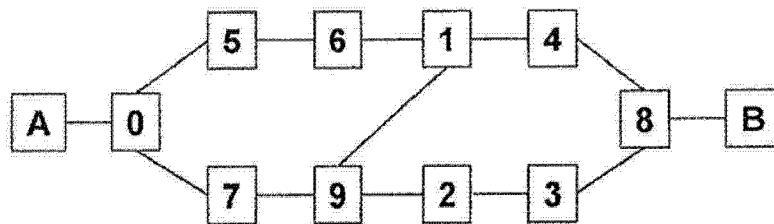


图 5

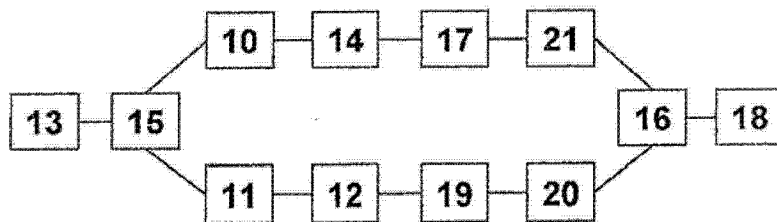


图 7

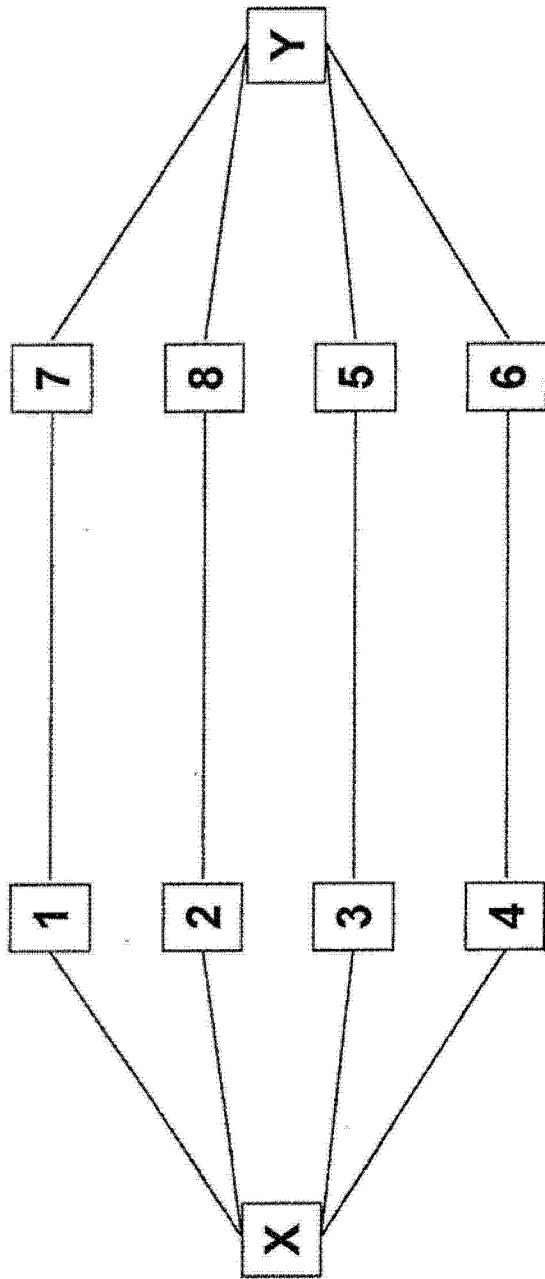


图 6

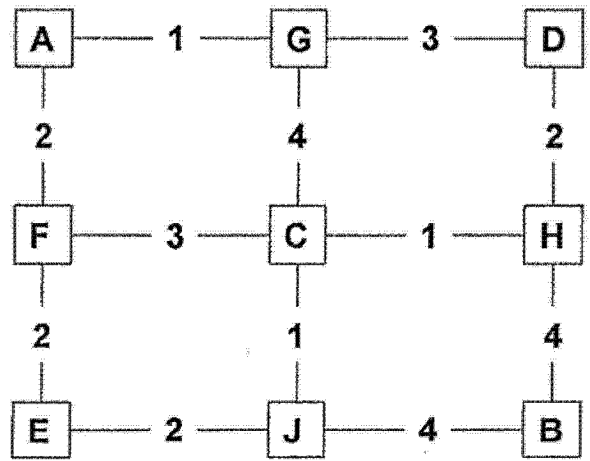


图 8

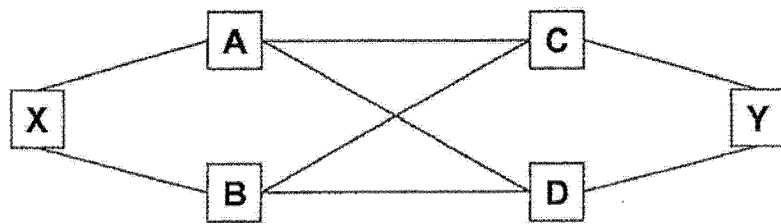


图 12

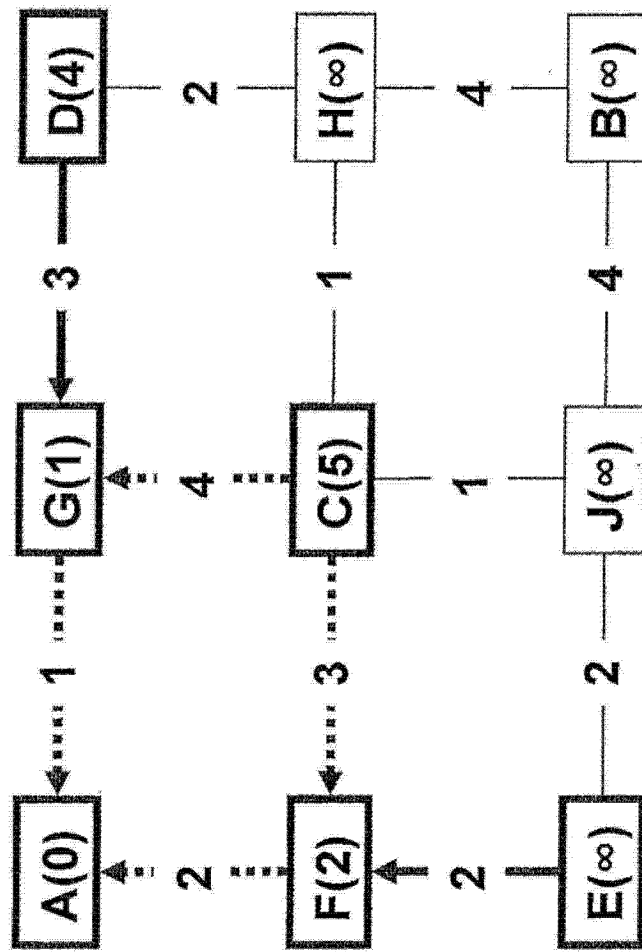


图 9

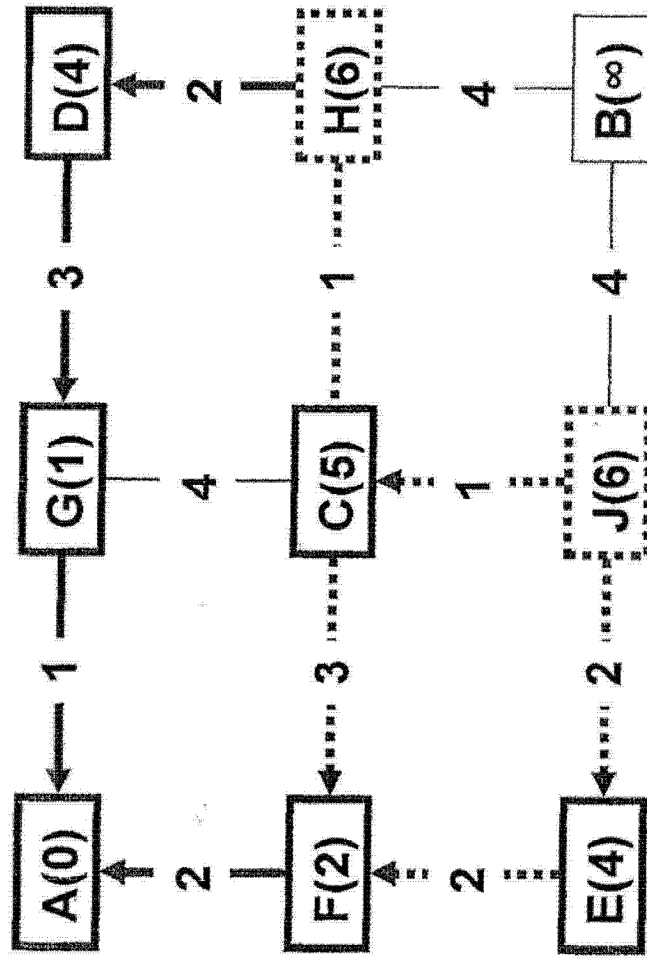


图 10

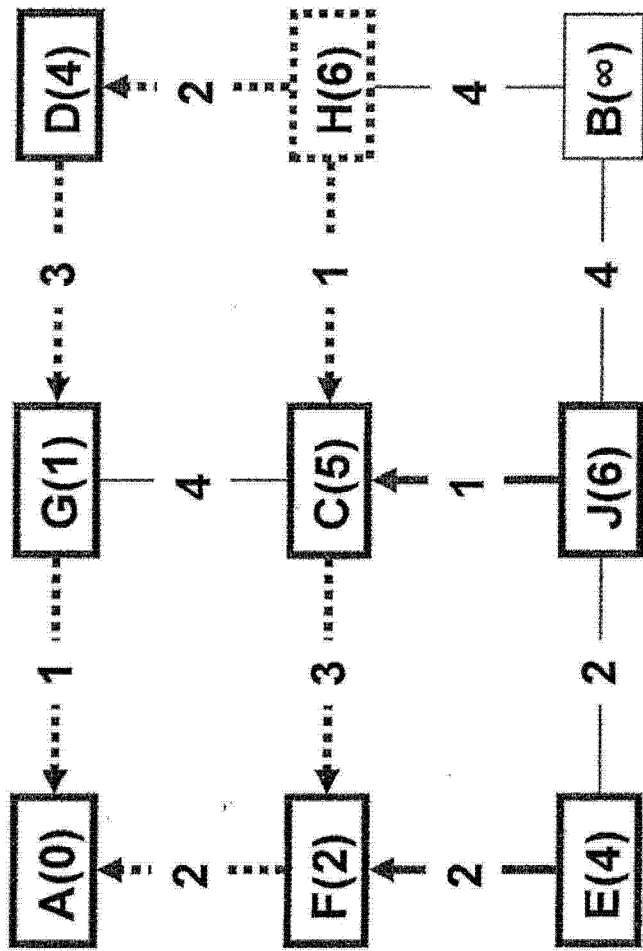


图 11

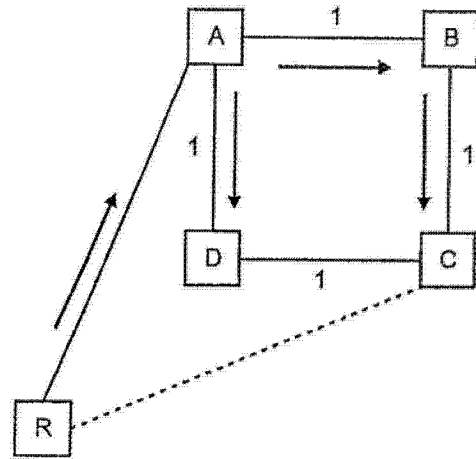


图 13A

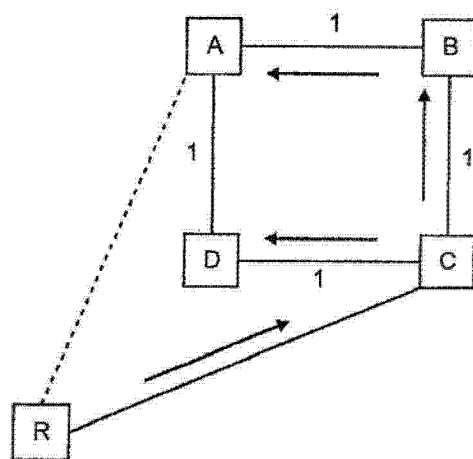


图 13B