



(12) 发明专利

(10) 授权公告号 CN 103200643 B

(45) 授权公告日 2015. 10. 28

(21) 申请号 201310106208. 7

(22) 申请日 2013. 03. 28

(73) 专利权人 西安电子科技大学
地址 710071 陕西省西安市太白南路 2 号

(72) 发明人 王玺钧 盛敏 刘梦霞 张琰
翟道森 李建东

(74) 专利代理机构 陕西电子工业专利中心
61205
代理人 王品华 朱红星

(51) Int. Cl.
H04W 40/10(2009. 01)
H04W 40/04(2009. 01)
H04W 84/18(2009. 01)

(56) 对比文件

CN 102088751 A, 2011. 06. 08,
CN 102946649 A, 2013. 02. 27,
US 20040143842 A1, 2004. 07. 22,
US 20050157661 A1, 2005. 07. 21,

审查员 吴欣

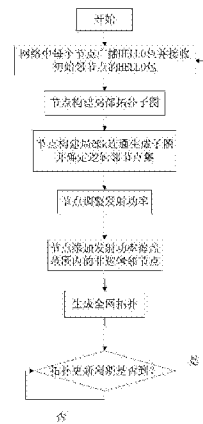
权利要求书2页 说明书7页 附图8页

(54) 发明名称

基于剩余能量感知的分布式容错拓扑控制方法

(57) 摘要

本发明公开了一种基于剩余能量感知的分布式容错拓扑控制方法,主要解决现有技术中不能同时保证节点间剩余能量平衡和网络拓扑容错能力的问题。其实现过程为:网络中的每个节点广播自己的HELLO包并接收初始邻节点的HELLO包,建立局部拓扑子图;基于局部拓扑子图,先根据链路代价权重,构建最短路径树,再按距离权重从小到大遍历所有有向边,构建局部k连通生成子图;根据局部k连通生成子图中的一跳邻节点调整发射功率,并连接此发射功率范围内的非逻辑邻节点;最后由网络中的所有节点以及节点与其逻辑邻节点间的链路构成全网拓扑;上述过程按拓扑更新周期T重复执行。本发明具有延长网络生存期,增强网络容错能力的优点,可用于无线自组织网络。



1. 一种基于剩余能量感知的分布式容错拓扑控制方法,包括如下步骤:

(1) 网络中每个节点 u 发送自己的 HELLO 包,并接收初始邻节点发送的 HELLO 包,该 HELLO 包中包括节点的 ID 序列号、剩余能量信息以及位置信息;

(2) 网络中每个节点 u 构建自己的局部拓扑子图 G_u :

(2a) 网络中的每个节点 u 根据接收到的初始邻节点的 HELLO 包信息,确定自己与初始邻节点的连接关系,以及这些初始邻节点之间的连接关系,建立局部拓扑子图 G_u ;

(2b) 根据局部拓扑子图,每个节点 u 计算局部拓扑子图中任意两个有连接关系的节点 x, y 之间的链路代价权重 $\bar{w}(x, y)$ 及距离权重 $\bar{w}_d(x, y)$ 。

(3) 网络中每个节点 u 构建局部 k 连通生成子图 $S_u = (V(S_u), E(S_u))$:

(3a) 网络中的每个节点 u 将局部 k 连通生成子图 S_u 的节点集合 $V(S_u)$ 初始化成局部拓扑子图中所有节点,将局部 k 连通生成子图 S_u 的边集合 $E(S_u)$ 初始化成空集;

(3b) 基于局部拓扑子图,每个节点 u 根据链路代价权重 $\bar{w}(x, y)$,构建以 u 为根,遍及局部拓扑子图中所有节点的最短路径树 $T_u = (V(T_u), E(T_u))$,其中 $V(T_u) = V(G_u)$ 为局部拓扑子图中所有节点, $E(T_u)$ 为构成最短路径树的所有有向边;

(3c) 把最短路径树 T_u 中的有向边 $E(T_u)$ 全部添加到局部 k 连通生成子图 S_u 中,即 $E(S_u) \leftarrow E(S_u) \cup E(T_u)$, \leftarrow 表示赋值, \cup 表示两个集合的并;

(3d) 对局部拓扑子图中的有向边按距离权重大小进行排序,获得有序的边序列 $E'(G_u)$;

(3e) 遍历 $E'(G_u)$ 中的每条有向边 (x, y) ,若 $(x, y) \notin E(S_u)$,判断节点 x, y 在 S_u 中是否达到 k 连通:若 S_u 中 x, y 没有达到 k 连通,则把 (x, y) 添加到 $E(S_u)$ 中,即 $(x, y) \in E(S_u)$,然后开始遍历下一条边,直至遍历完 $E'(G_u)$ 中的全部有向边;若 S_u 中 x, y 已达到 k 连通,则直接遍历下一条边,直至遍历完 $E'(G_u)$ 中的全部有向边;

(3f) 每个节点 u 将局部 k 连通生成子图 S_u 上的一跳邻节点 v 作为逻辑邻节点,并构成逻辑邻节点集: $LN_u = \{v \in V(S_u) \mid (u, v) \in E(S_u)\}$;

(4) 网络中每个节点 u 确定自己的发射功率,即将发射功率调整为能够覆盖到局部 k 连通生成子图 S_u 中最远的逻辑邻节点所需要的最小功率: $P_{LN_u} = \max\{p_{u,v} \mid v \in LN_u\}$;

(5) 每个节点 u 检查在其发射功率所对应的传输半径范围内是否存在非逻辑邻节点 z :若存在非逻辑邻节点 z ,则把节点 u 到该节点 z 的链路添加到局部 k 连通生成子图 S_u 中;若不存在非逻辑邻节点 z ,则保持原局部 k 连通生成子图 S_u 不变;

(6) 将网络中的所有节点以及每个节点与自己的逻辑邻节点间的链路组合起来,构成最终的全网拓扑,即 $G = (V(G), E(G))$,其中 $V(G)$ 为网络中所有节点, $E(G) = \{(u, v) \mid u \in V(G), v \in LN_u\}$;

(7) 在经过一个拓扑更新周期 T 的时间后,开始重新执行上述步骤 1 ~ 步骤 6。

2. 根据权利要求 1 所述的分布式容错拓扑控制方法,其中步骤 (1) 所述的网络中每个节点 u 发送自己的 HELLO 包,是指网络中的每个节点 u ,以最大发射功率 P_{\max} 向位于距离自己传输半径范围内的所有节点广播一次 HELLO 包。

3. 根据权利要求 1 所述的分布式容错拓扑控制方法,其中步骤 (1) 所述的接收初始邻

节点发送的 HELLO 包,是指网络中的每个节点 u 接收其初始邻节点以最大发射功率 P_{\max} 广播的 HELLO 包。

4. 根据权利要求 1 所述的分布式容错拓扑控制方法,其中步骤 (2a) 所述的确定自己与初始邻节点的连接关系,以及这些初始邻节点之间的连接关系,建立局部拓扑子图 G_u ,按如下步骤进行:

(2a1) 每个节点 u 根据接收到的初始邻节点的 HELLO 包信息,获取并记录发送该 HELLO 包的节点的 ID 序列号、剩余能量信息以及位置信息,根据记录的节点的 ID 序列号确定每个节点 u 与初始邻节点的连接关系,这些初始邻节点组成该节点的初始邻节点集 VN_u ;

(2a2) 每个节点 u 根据自己的位置信息以及初始邻节点的位置信息,计算任意两个初始邻节点 \hat{x}, \hat{y} 间直接传输所需要s的最小发射功率 $P_{\hat{x}, \hat{y}}$:

$$P_{\hat{x}, \hat{y}} = \beta d_{\hat{x}, \hat{y}}^\alpha$$

其中, $\hat{x}, \hat{y} \in VN_u$, β 为接收信噪比门限值,根据接收机的灵敏度和误码率要求确定, α 为路径损耗因子, $d_{\hat{x}, \hat{y}}$ 是初始邻节点 \hat{x} 、 \hat{y} 间的欧式距离,若 $P_{\hat{x}, \hat{y}}$ 小于节点的最大发射功率 P_{\max} ,则确定初始邻节点 \hat{x}, \hat{y} 间存在连接关系,否则,初始邻节点 \hat{x}, \hat{y} 间不存在连接关系;

(2a3) 每个节点 u 根据初始邻节点之间的连接关系,建立局部拓扑子图 $G_u = (V(G_u), E(G_u))$,其中局部拓扑子图 G_u 的节点集合为 $V(G_u) = VN_u \cup \{u\}$, $\{u\}$ 表示节点 u 组成的集合,局部拓扑子图 G_u 的边集合为: $E(G_u) = \{(\hat{x}, \hat{y}) | \hat{x}, \hat{y} \in V(G_u), P_{\hat{x}, \hat{y}} < P_{\max}\}$ 。

5. 根据权利要求 1 所述的分布式容错拓扑控制方法,其中所述步骤 (2b) 中的任意两个有连接关系的节点 (x, y) 间的链路代价权重 $\bar{w}(x, y)$,通过如下公式计算:

$$\bar{w}(x, y) = (P_{x, y} \cdot t)^{x_1} \left(\frac{\varepsilon}{\varepsilon_x'} \right)^{x_2} + (P_y' \cdot t)^{x_1} \left(\frac{\varepsilon}{\varepsilon_y'} \right)^{x_2},$$

其中, $P_{x, y}$ 是最小发射功率, P_y' 为节点 y 在接收信号时消耗的功率, t 为分组传输时间, ε 为所有节点的初始能量值, ε_x' 、 ε_y' 分别为节点 x 、 y 当前的剩余能量值, x_1 和 x_2 为非负的权重因子,且 $x_1 \neq x_2$ 。

6. 根据权利要求 1 所述的分布式容错拓扑控制方法,其中所述步骤 (2b) 中的任意两节点 (x, y) 间的距离权重 $\bar{w}_d(x, y)$,通过如下公式计算: $\bar{w}_d(x, y) = (d_{x, y}, id(x), id(y))$,其中, $d_{x, y}$ 为节点间的欧式距离, $id(x)$ 、 $id(y)$ 分别为节点 x 和节点 y 的 ID 序列号。

7. 根据权利要求 1 所述的分布式容错拓扑控制方法,其中所述步骤 (3a) 中的局部 k 连通采用 k 边连通,即在网络中任意去掉 $k-1$ 条边后,网络仍然是连通的。

8. 根据权利要求 1 所述的分布式容错拓扑控制方法,其中所述步骤 (3a) 中的局部 k 连通采用 k 点连通,即在网络中任意去掉 $k-1$ 个点后,网络仍然是连通的。

9. 根据权利要求 1 所述的分布式容错拓扑控制方法,其中所述步骤 (3b) 中的最短路径树通过使用 Dijkstra 算法或 Bellman-Ford 算法构建。

基于剩余能量感知的分布式容错拓扑控制方法

技术领域

[0001] 本发明属于无线通信领域,特别涉及一种构建网络拓扑结构的方法,可用于 Ad Hoc 网络。

背景技术

[0002] Ad Hoc 网络是一种充满发展潜力的无线网络通信系统,该网络具有的自组织、自配置、自适应以及自愈能力使之能够灵活地用于各种无固定通信基础设施支撑的环境。

[0003] 在影响 Ad Hoc 网络性能的众多因素之中,网络的拓扑结构是不可忽视的一个重要方面,因此如何优化 Ad Hoc 网络的拓扑结构、增强网络拓扑的容错能力并为上层通信协议提供良好的底层拓扑支撑是拓扑控制技术研究的重点。

[0004] 目前,为了提高网络能量使用的有效性,研究者已经提出了许多拓扑控制算法。按照优化目标的不同,这些算法大致可以分为两大类:一类是以最小化网络节点的功耗为优化目标的拓扑控制算法,另一类是以最大化端到端路径上的能量有效性为优化目标的拓扑控制算法。在保证网络结构具有某些基本特性,如网络连通性,邻节点度数有限等的基础上,第一类算法主要是寻找能够使节点功率消耗达到最小的拓扑构建方案,其典型代表有 R. Wattenhofer 等作者在 IEEE INFOCOM 2001 上发表的文章“Distributed topology control for power efficient operation in multihop wireless ad hoc networks”中提到的 CBTC(α) 算法,以及 N. Li 等作者在 IEEE Transactions on Wireless Communications 上发表的文章“Design and Analysis of an MST-Based Topology Control Algorithm”提到的 LMST 算法等;第二类拓扑控制算法的核心思想在于构建端到端最小能量路径的网络拓扑结构,从而为分组路由过程中路径的选择提供良好的底层拓扑支撑,属于这一类的算法主要有 V. Rodoplu 等作者在 IEEE Journal on Selected Areas in Communications 上发表的文章“Minimum energy mobile wireless networks”中提到的 R&M 算法,以及 Y. Shen 等作者在 ACM SIGCOMM Computer Communication Review 上发表的文章“A shortest-path-based topology control algorithm in wireless multihop networks”中提到的 LSP 算法等。总而言之,无论两类算法在能效优化目标和设计思路方面存在何种差异,它们的最终目标都是希望能够尽可能地降低节点的发射功率,从而延长网络的生存期。

[0005] 但实际上,在上述算法中,虽然减小了每个节点的传输功率,但由于没有考虑到各个节点剩余能量的变化和能量损耗的速率,只生成静态网络拓扑,不能平衡节点间的剩余能量,导致不能最大限度地延长网络生存期,即网络开始工作到有一个节点死亡的时间。由于不同节点大多以不同的速率消耗能量,这种能量消耗的不平衡会导致某些节点很快耗尽了自身的能量,导致网络生存期缩短。为了解决这样的问题,有人提出了考虑节点能量的拓扑控制算法,如田野等作者在 IEEE GLOBECOM 上发表的文章“Energy-aware Dynamic Topology Control Algorithm”中提到的 EDTC 算法、X. Chu 等作者在 IEEE INFOCOM 上发表的文章“Cooperative Topology Control with Adaptation for Improved Lifetime

in Wireless Ad Hoc Networks”中提到的 CTCA 算法和 T.Chakraborty 等作者在 IEEE GLOBECOM 上发表的文章“Load and Energy Aware Topology Control in Wireless Ad-hoc Networks”中提到的 LEATC 算法。上述这些算法虽然考虑了节点间剩余能量的平衡,但构建的拓扑十分稀疏,容易造成网络分割。

发明内容

[0006] 本发明的目的在于针对上述现有技术的问题,提出一种基于剩余能量感知的分布式容错拓扑控制方法,以延长网络生存期,增加网络容错能力。

[0007] 为了实现上述目的,本发明网络拓扑控制方法包括如下步骤:

[0008] (1) 网络中每个节点 u 发送自己的 HELLO 包,并接收初始邻节点发送的 HELLO 包,该 HELLO 包中包括节点的 ID 序列号、剩余能量信息以及位置信息;

[0009] (2) 网络中每个节点 u 构建自己的局部拓扑子图 G_u :

[0010] (2a) 网络中的每个节点 u 根据接收到的初始邻节点的 HELLO 包信息,确定自己与初始邻节点的连接关系,以及这些初始邻节点之间的连接关系,建立局部拓扑子图 G_u ;

[0011] (2b) 根据局部拓扑子图,每个节点 u 计算局部拓扑子图中任意两个有连接关系的节点 x, y 之间的链路代价权重 $\bar{w}(x, y)$ 及距离权重 $\bar{w}_d(x, y)$;

[0012] (3) 网络中每个节点 u 构建局部 k 连通生成子图 $S_u = (V(S_u), E(S_u))$;

[0013] (3a) 网络中的每个节点 u 将局部 k 连通生成子图 S_u 的节点集合 $V(S_u)$ 初始化成局部拓扑子图中所有节点,将局部 k 连通生成子图 S_u 的边集合 $E(S_u)$ 初始化成空集;

[0014] (3b) 基于局部拓扑子图,每个节点 u 根据链路代价权重 $\bar{w}(x, y)$, 构建以 u 为根,遍及局部拓扑子图中所有节点的最短路径树 $T_u = (V(T_u), E(T_u))$, 其中 $V(T_u) = V(G_u)$ 为局部拓扑子图中所有节点, $E(T_u)$ 为构成最短路径树的所有有向边;

[0015] (3c) 把最短路径树 T_u 中的所有有向边 $E(T_u)$ 全部添加到局部 k 连通生成子图 S_u 中,即 $E(S_u) = E(S_u) \cup E(T_u)$, \cup 表示两个集合的并;

[0016] (3d) 对局部拓扑子图中的所有有向边按距离权重大小进行排序,获得有序的边序列 $E'(G_u)$;

[0017] (3e) 遍历 $E'(G_u)$ 中的每条有向边 (x, y) , 若 $(x, y) \notin E(S_u)$, 判断节点 x, y 在 S_u 中是否达到 k 连通:若 S_u 中 x, y 没有达到 k 连通,则把 (x, y) 添加到 $E(S_u)$ 中,即 $(x, y) \in E(S_u)$, 然后开始遍历下一条边,直至遍历完 $E'(G_u)$ 中的全部有向边;若 S_u 中 x, y 已达到 k 连通,则直接遍历下一条边,直至遍历完 $E'(G_u)$ 中的全部有向边;

[0018] (3f) 每个节点 u 将局部 k 连通生成子图 S_u 上的一跳邻节点 v 作为逻辑邻节点,并构成逻辑邻节点集: $LN_u = \{v \in V(S_u) \mid (u, v) \in E(S_u)\}$;

[0019] (4) 网络中每个节点 u 确定自己的发射功率,即将发射功率调整为能够覆盖到局部 k 连通生成子图中最远的逻辑邻节点所需要的最小功率: $P_{LN_u} = \max\{p_{u,v} \mid v \in LN_u\}$;

[0020] (5) 每个节点 u 检查在其发射功率所对应的传输半径范围内是否存在非逻辑邻节点 z :若存在非逻辑邻节点 z ,则把节点 u 到该节点 z 的链路添加到局部 k 连通生成子图 S_u 中;若不存在非逻辑邻节点 z ,则保持原局部 k 连通生成子图 S_u 不变;

[0021] (6) 将网络中的所有节点以及每个节点与自己的逻辑邻节点间的链路组合起

来,构成最终的全网拓扑,即 $G = (V(G), E(G))$, 其中 $V(G)$ 为网络中所有节点, $E(G) = \{(u, v) \mid u \in V(G), v \in LN_u\}$;

[0022] (7) 在经过一个拓扑更新周期 T 的时间后,开始重新执行上述步骤 1 ~ 步骤 6。

[0023] 本发明具有如下优点:

[0024] 1) 本发明由于引入了综合反映节点能量消耗速度及剩余能量两方面因素的链路代价函数,并通过周期性地动态调整拓扑结构,使得本发明方法能显著地延长网络的生存时间;

[0025] 2) 本发明由于在保证端到端最小代价路径的基础上实现了 k 连通的网络拓扑构建,增强了网络鲁棒性,提高了整个网络的容错能力。

附图说明

[0026] 图 1 为本发明适用的 Ad Hoc 网络场景示意图;

[0027] 图 2 为 50 节点网络场景时形成的最大功率拓扑;

[0028] 图 3 为本发明的流程图;

[0029] 图 4 为本发明构建局部 k 连通生成子图的子流程图;

[0030] 图 5 为本发明中节点 u 构建局部 k 连通生成子图的示例图;

[0031] 图 6 为本发明中节点 u 调整局部 k 连通生成子图的示例图;

[0032] 图 7 为本发明网络生存期关于节点数的仿真图;

[0033] 图 8 为在相同连通度时本发明与其他拓扑控制算法的仿真对比图;

[0034] 图 9 为在不同连通度时本发明与其他拓扑控制算法的仿真对比图。

具体实施方式

[0035] 下面将结合附图对本发明实施方式做进一步详细描述。

[0036] 参照图 1,本发明使用的 Ad Hoc 网络由 n 个分布在二维平面区域内的同质节点组成。每个节点具有唯一的 ID 序列号和相同的初始能量 ϵ ,并可以通过 GPS 或是其他定位技术来获取它自身的位置信息,除此之外各个节点在物理结构、初始设置、功能特性、参数指标等方面不存在任何差异。在网络中,任意节点间的无线信道为加性高斯白噪声信道。节点通过全向天线与周围节点通信,最大发射功率均为 P_{max} 。任意节点 u 的发射功率 P_u 可以在最小和最大之间连续调节,即 $0 \leq P_u \leq P_{max}$ 。传输半径 r 为对应于节点发射功率的传输距离,任意两个节点之间存在无线链路的充要条件为它们之间的欧式距离小于或等于节点的传输半径 r 。当网络中每个节点均使用最大功率传输时形成的拓扑结构为最大功率拓扑,如图 2 所示,最大功率拓扑表示为: $G_{max} = (V(G_{max}), E(G_{max}))$, 其中 $V(G_{max})$ 为节点集合,表示网络节点, $E(G_{max})$ 为边集合,表示节点间存在的无线链路。

[0037] 参照图 3,本发明的实现步骤如下:

[0038] 步骤 1,网络中每个节点 u 发送自己的 HELLO 包,并接收初始邻节点发送的 HELLO 包。

[0039] 位于节点 u 的传输半径范围内的所有节点,组成节点 u 的初始邻节点集 $VN_u: VN_u = \{v \in V(G_{max}) \mid (u, v) \in E(G_{max})\}$;

[0040] 网络中的每个节点 u 以最大发射功率 P_{max} 向 u 的初始邻节点广播一次 HELLO 包,

HELLO 包中含有节点 u 的 ID 序列号、节点 u 的剩余能量信息以及节点 u 的位置信息；

[0041] 网络中的每个节点 u 接收其初始邻节点以最大发射功率 P_{\max} 广播的 HELLO 包。

[0042] 步骤 2, 每个节点 u 构建自己的局部拓扑子图 G_u 。

[0043] (2a) 网络中的每个节点 u 根据接收到的初始邻节点的 HELLO 包信息, 获取并记录自己所有初始邻节点 v 的 ID 序列号、剩余能量信息以及位置信息, 其中 $v \in VN_u$;

[0044] (2b) 每个节点 u 根据自己的位置信息以及初始邻节点的位置信息, 计算任意两个初始邻节点 \hat{x}, \hat{y} 之间直接传输所需要的最小发射功率 $P_{\hat{x}, \hat{y}}$;

$$[0045] \quad P_{\hat{x}, \hat{y}} = \beta d_{\hat{x}, \hat{y}}^\alpha$$

[0046] 其中, $\hat{x}, \hat{y} \in VN_u$, β 为接收信噪比门限值, 根据接收机的灵敏度和误码率要求确定, 当信号接收信噪比 SNR 大于门限值 β 时该信号可被正确接收, α 为路径损耗因子, $d_{\hat{x}, \hat{y}}$ 是初始邻节点 \hat{x}, \hat{y} 之间的欧式距离;

[0047] (2c) 根据计算的最小发射功率, 判断初始邻节点之间的连接关系, 若 $P_{\hat{x}, \hat{y}}$ 小于节点的最大发射功率 P_{\max} , 则确定初始邻节点 \hat{x}, \hat{y} 之间存在连接关系; 否则, 初始邻节点 \hat{x}, \hat{y} 之间不存在连接关系;

[0048] (2d) 每个节点 u 根据初始邻节点之间的连接关系, 建立局部拓扑子图 $G_u = (V(G_u), E(G_u))$, 其中局部拓扑子图 G_u 的节点集合为 $V(G_u) = VN_u \cup \{u\}$, $\{u\}$ 表示节点 u 组成的集合, \cup 表示两个集合的并, 局部拓扑子图 G_u 的边集合为: $E(G_u) = \{(\hat{x}, \hat{y}) | \hat{x}, \hat{y} \in V(G_u), P_{\hat{x}, \hat{y}} < P_{\max}\}$, 即对于 $V(G_u)$ 中的任意两个初始邻节点 \hat{x}, \hat{y} , 当 $P_{\hat{x}, \hat{y}} < P_{\max}$ 时, 边 $(\hat{x}, \hat{y}) \in E(G_u)$;

[0049] (2e) 根据局部拓扑子图, 每个节点 u 计算任意两个有连接关系的节点 x, y 之间的链路代价权重 $\vec{w}(x, y)$:

$$[0050] \quad \vec{w}(x, y) = (P_{x, y} \cdot t)^{x_1} \left(\frac{\varepsilon}{\varepsilon_x'} \right)^{x_2} + (P_y' \cdot t)^{x_1} \left(\frac{\varepsilon}{\varepsilon_y'} \right)^{x_2}$$

[0051] 其中, $x, y \in VN_u \cup \{u\}$, $P_{x, y}$ 为任意两个有连接关系的节点 x, y 之间直接传输所需要的最小发送功率, P_y' 为节点 y 在接收信号时消耗的功率, t 为分组传输时间, ε 为所有节点的初始能量值, ε_x' 、 ε_y' 分别为任意两个有连接关系的节点 x, y 当前的剩余能量值, x_1 和 x_2 为非负的权重因子, 且 $x_1 \neq x_2$, 由此可以看出, 无线链路的代价不具有对称性, 即 $\vec{w}(x, y) \neq \vec{w}(y, x)$;

[0052] (2f) 根据上述欧式距离和节点 ID 序列号, 节点 u 计算任意两个有连接关系的节点 x, y 之间的距离权重 $\vec{w}_d(x, y)$:

$$[0053] \quad \vec{w}_d(x, y) = (d_{x, y}, id(x), id(y)),$$

[0054] 其中, $x, y \in VN_u \cup \{u\}$, $d_{x, y}$ 是任意两个有连接关系的节点 x, y 之间的欧式距离。

[0055] 步骤 3, 网络中每个节点 u 构建局部 k 连通生成子图 $S_u = (V(S_u), E(S_u))$, 并确定

自己的逻辑邻节点。

[0056] 具体流程如图 4 所示：

[0057] (3a) 网络中的每个节点 u 将局部 k 连通生成子图 S_u 的节点集合 $V(S_u)$ 初始化成局部拓扑子图中所有节点, 将局部 k 连通生成子图 S_u 的边集合 $E(S_u)$ 初始化成空集; 这里的 k 连通包括 k 边连通和 k 点连通, k 边连通是指在网络中任意去掉 $k-1$ 条边后, 网络仍然连通, k 点连通是指在网络中任意去掉 $k-1$ 点边后, 网络仍然连通, 本实例以 k 边连通为例进行说明;

[0058] (3b) 基于局部拓扑子图 G_u , 以链路代价权重为链路权重, 节点 u 通过使用 Dijkstra 算法或 Bellman-Ford 算法, 构建以 u 为根, 遍及 $V(G_u)$ 中所有节点的最短路径树 $T_u = (V(T_u), E(T_u))$, 其中 $V(T_u) = V(G_u)$ 为局部拓扑子图中所有节点, $E(T_u)$ 为构成最短路径树的所有有向边, 从而在局部范围内获得到达局部拓扑子图中任意节点的最短路径;

[0059] (3c) 把最短路径树 T_u 中的有向边全部添加到局部 k 连通生成子图 S_u 中, 即 $E(S_u) \Leftarrow E(S_u) \cup E(T_u)$, \Leftarrow 表示赋值, \cup 表示两个集合的并;

[0060] (3d) 对局部拓扑子图 $E(G_u)$ 中的有向边按距离权重大小进行排序, 获得有序的边序列 $E'(G_u)$;

[0061] (3d1) 比较局部拓扑子图 $E(G_u)$ 中任意两个有向边 (x, y) 和 (x_0, y_0) 的距离权重 $\bar{w}_d(x, y)$ 和 $\bar{w}_d(x_0, y_0)$, 按照距离权重中的每一项依次进行比较, 先比较两个有向边的距离 $d_{x,y}$ 和 d_{x_0,y_0} , 如果距离相等则比较两个有向边头节点的 ID 序列号 $id(x)$ 和 $id(x_0)$, 如果还相等则比较两个有向边尾节点的 ID 序列号 $id(y)$ 和 $id(y_0)$, 比较规则如下:

[0062] $\bar{w}_d(x, y) > \bar{w}_d(x_0, y_0)$

[0063] $\Leftrightarrow d_{x,y} > d_{x_0,y_0}$

[0064]

or $(d_{x,y} = d_{x_0,y_0} \ \&\& \ id(x) > id(x_0))$

[0065]

or $(d_{x,y} = d_{x_0,y_0} \ \&\& \ id(x) = id(x_0) \ \&\& \ id(y) > id(y_0))$

[0066] 其中, \Leftrightarrow 表示等价于, $\&\&$ 表示并且, or 表示或者;

[0067] (3d2) 按照有向边距离权重从小到达的顺序, 对局部拓扑子图中的有向边进行排序, 得到有序的边序列 $E'(G_u)$;

[0068] (3e) 遍历边序列 $E'(G_u)$ 中的每条有向边 (x, y) , 若 $(x, y) \notin E(S_u)$, 判断有向边的头节点 x 和尾节点 y 在 S_u 中是否达到 k 连通: 若 S_u 中有向边的头节点 x 和尾节点 y 没有达到 k 连通, 则把有向边 (x, y) 添加到 $E(S_u)$ 中, 即 $(x, y) \in E(S_u)$, 然后开始遍历下一条边, 直至遍历完 $E'(G_u)$ 中的全部有向边; 若 S_u 中有向边的头节点 x 和尾节点 y 已达到 k 连通, 则直接遍历下一条边, 直至遍历完 $E'(G_u)$ 中的全部有向边;

[0069] (3f) 每个节点 u 将局部 k 连通生成子图 S_u 上的一跳邻节点 v 作为逻辑邻节点, 并构成逻辑邻节点集: $LN_u = \{v \in V(S_u) \mid (u, v) \in E(S_u)\}$, 其中, $V(S_u)$ 为局部 k 连通生成子图 S_u 的节点集合, $E(S_u)$ 为局部 k 连通生成子图 S_u 的边集合;

[0070] 参照图 5, 对于初始邻节点 v_1, v_2, \dots, v_6 , 节点 u 构建局部 k 连通生成子图 S_u , 并确

定 v_2, v_3, v_5, v_6 为其逻辑邻节点, 即 $LN_u = \{v_2, v_3, v_5, v_6\}$ 。

[0071] 步骤 4, 每个节点 u 根据上述确定的逻辑邻节点, 将自己的发射功率调整为能够覆盖到局部 k 连通生成子图 S_u 中最远的逻辑邻节点所需要的最小功率 P_{LN_u} , 即

$$P_{LN_u} = \max\{p_{u,v} \mid v \in LN_u\}。$$

[0072] 步骤 5, 每个节点 u 检查在其传输功率 P_{LN_u} 所对应的传输半径范围内是否存在非逻辑邻节点 z ; 若存在非逻辑邻节点 z , 则把节点 u 到该节点 z 的链路添加到局部 k 连通生成子图 S_u 中, 并把 z 添加到逻辑邻节点集 LN_u 中; 若不存在非逻辑邻节点 z , 则保持原局部 k 连通生成子图 S_u 不变;

[0073] 参照图 6, 示例中的节点 v_4 处于节点 u 的传输范围内, 但又不属于节点 u 的逻辑邻节点, 则将 u 到 v_4 的单向链路添加到 S_u 中, 即 $(u, v_4) \in E(S_u)$, 如图 6 中虚线所示, 并将 v_4 添加到逻辑邻节点集中, 即 $v_4 \in LN_u$ 。

[0074] 步骤 6, 根据上述局部 k 连通生成子图, 网络中的每个节点独立确定与自己的逻辑邻节点的连接关系, 将网络中的所有节点以及每个节点与自己的逻辑邻节点间的链路组合起来, 构成最终的全网拓扑, 即 $G = (V(G), E(G))$, 其中 $V(G) = V(G_{\max})$, $E(G) = \{(u, v) \mid u \in V(G), v \in LN_u\}$ 。

[0075] 步骤 7, 为了能适应网络节点剩余能量的实时动态变化, 在经过一个拓扑更新周期 T 的时间后, 网络中每个节点开始重新执行上述步骤 1 ~ 步骤 6, 以保证周期性地更新链路代价信息, 进而动态调整网络拓扑。

[0076] 本发明的效果可通过仿真进一步说明:

[0077] (1) 仿真条件

[0078] 仿真采用 OPNET 网络仿真软件实现。

[0079] 在仿真场景中, 网络节点随机均匀分布在一个 $1000 \times 1000m^2$ 二维平面区域上。网络中所有节点采用相同的最大发射功率和初始能量, 其中最大发射功率 $P_{\max} = 200mW$, 初始能量 $\varepsilon = 500mJ$ 。网络中每个节点在接收信号时的功率损耗为 $1mW$, 接收信噪比 SNR 门限值 β 取值为 $-76dBm$ 。链路代价权重因子 x_1 取值为 1, x_2 取值为 10, 路径损耗因子 α 取值为 4。

[0080] 网络中每个节点以 2 分组 / 秒的速率向随机选择的节点发送数据分组, 数据分组大小 $L = 1024$ 字节, 数据发送速率 $R = 1Mb/s$, 每传输一个数据分组所需的时间为 $t = L/R$, 发送节点 u 与接收节点 v 消耗的能量分别为 $P_u \cdot t$ 和 $P_v' \cdot t$ 。拓扑更新周期 T 设为 5s。

[0081] 仿真采用理想的 MAC 层, 即不考虑 MAC 层对拓扑控制算法可能造成的影响, 路由协议统一采取能使网络生存期最大化的路由机制 Flow Augmentation 为分组传输建立路由, 并选取与计算链路代价权重公式相同的代价函数作为路由决策标准。仿真中的对比算法均采用 HELLO 机制交互网络信息, 且不考虑 HELLO 分组带来的能量消耗。

[0082] (2) 仿真内容和结果

[0083] 仿真 1, 采用本发明方法在连通度 k 分别为 1、2、3 时对网络生存期进行仿真, 结果如图 7 所示。

[0084] 图 7 表明: 随节点数的增多, 节点平均传输功率降低, 从而使网络生存期延长; 而

在节点数相同的场景下,连通度的增加会造成网络生存期的降低,体现了网络连通度与网络生存期之间的折中。

[0085] 仿真 2,在连通度 $k = 2$ 的条件下用本发明方法与现有的 FLSS 和 LTRT 算法对网络生存期进行仿真对比,结果如图 8 所示。

[0086] 图 8 表明:本发明方法的网络生存期优于其它拓扑控制算法,这证明了本发明方法的有效性。

[0087] 仿真 3,在不同连通度 k 的条件下用本发明方法与现有的 LMST、LSP 和 EDTC 算法对网络生存期进行仿真对比,结果如图 9 所示,其中,本发明方法是在 $k = 2$ 的条件下进行仿真,现有的 LMST、LSP 和 EDTC 算法是在 $k = 1$ 的条件下进行仿真。

[0088] 图 9 表明:本发明方法的连通度和网络生存期都高于现有的 LMST 和 LSP 算法,本发明方法与 EDTC 算法相比,在保持相近的网络生存期的同时提高了网络的连通度。

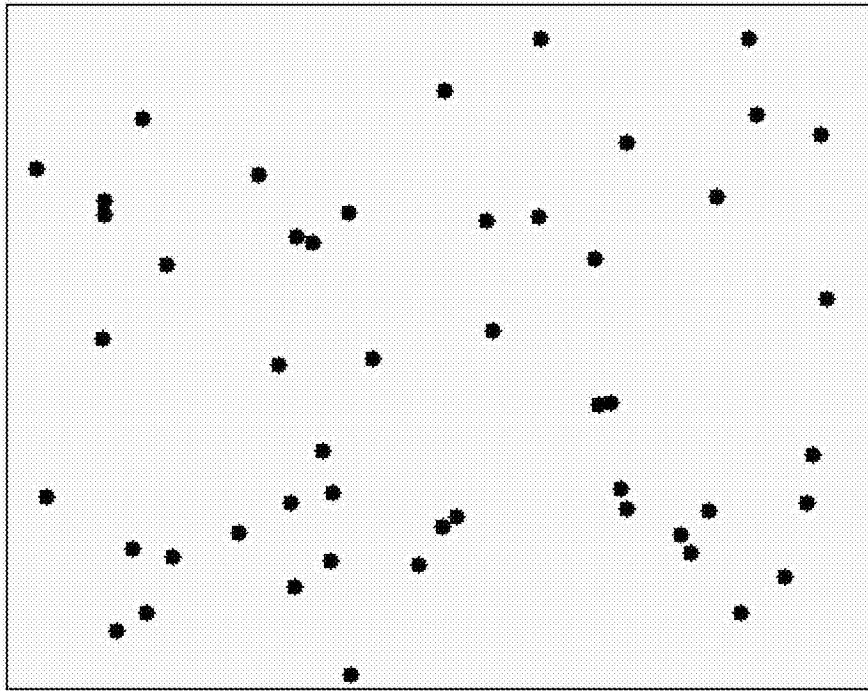


图 1

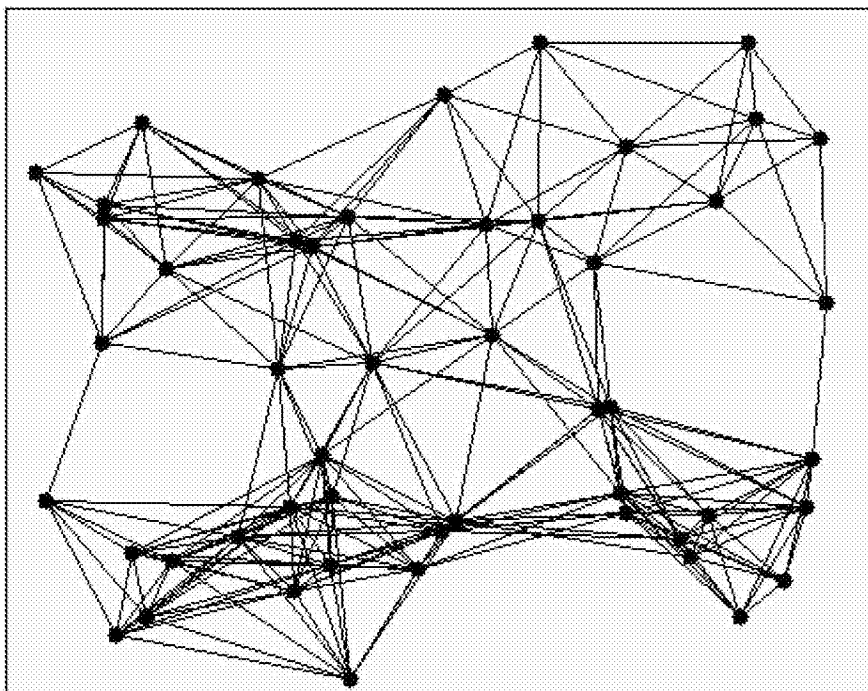


图 2

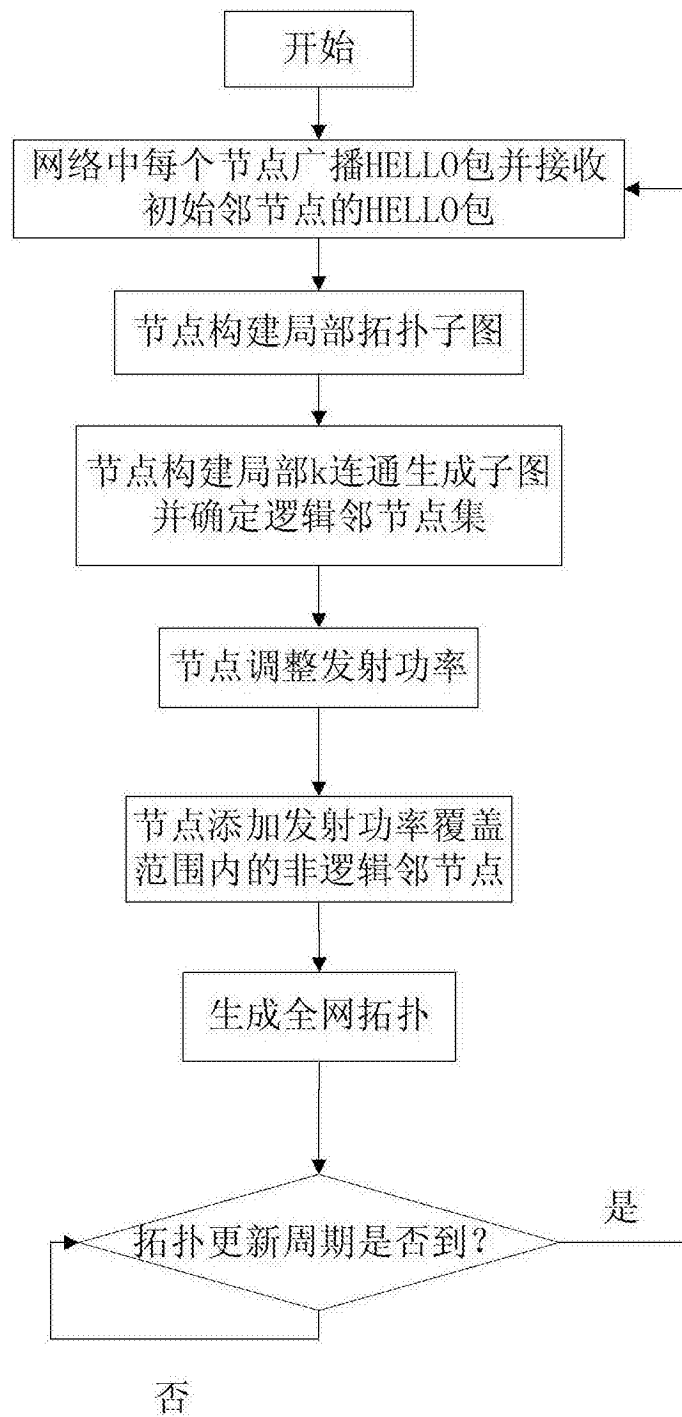


图 3

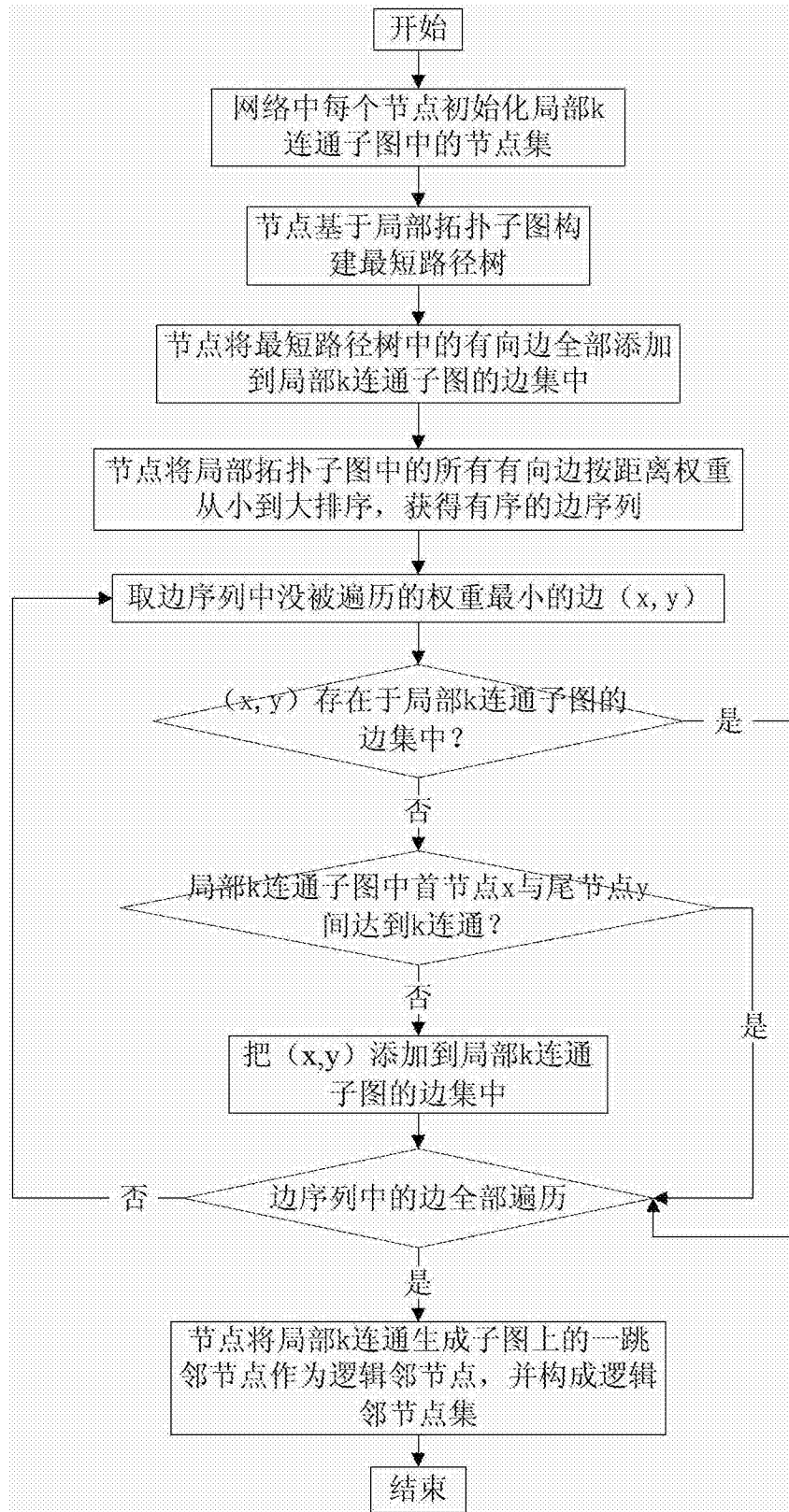


图 4

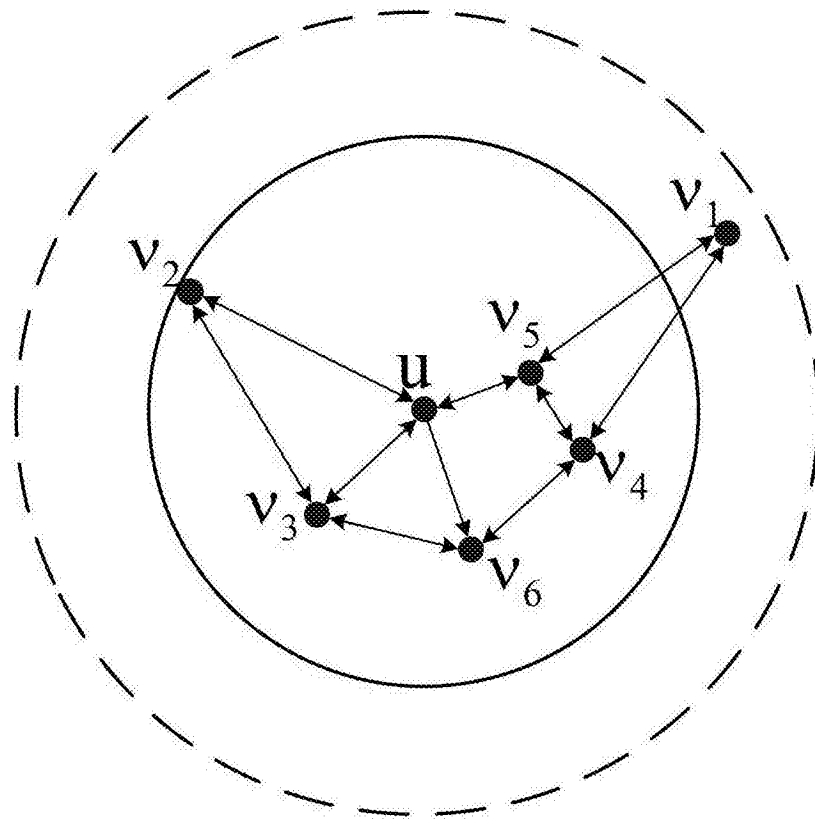


图 5

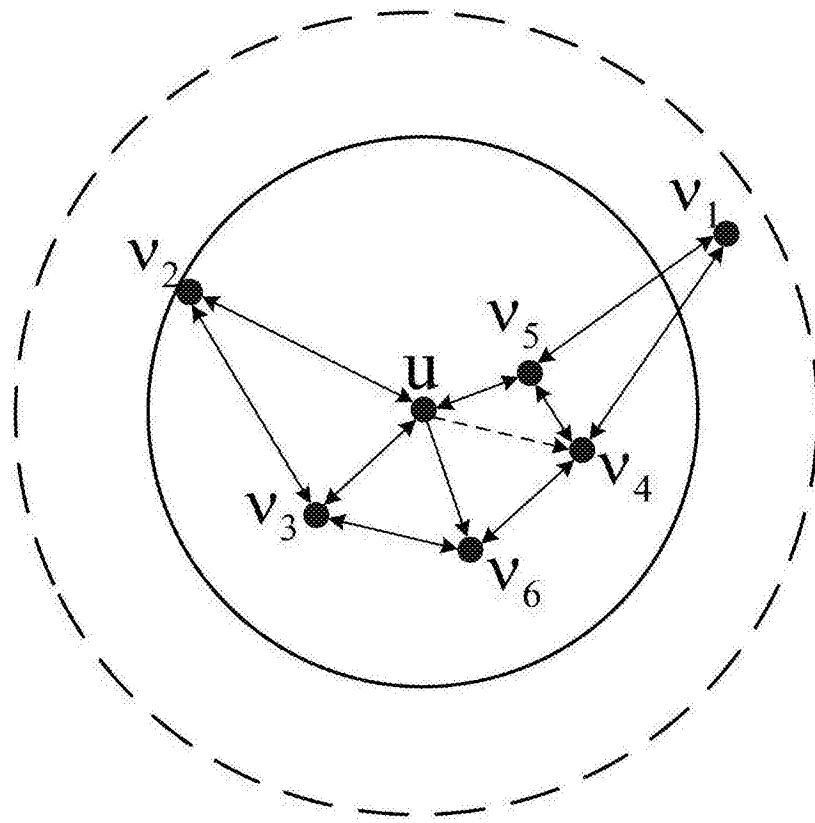


图 6

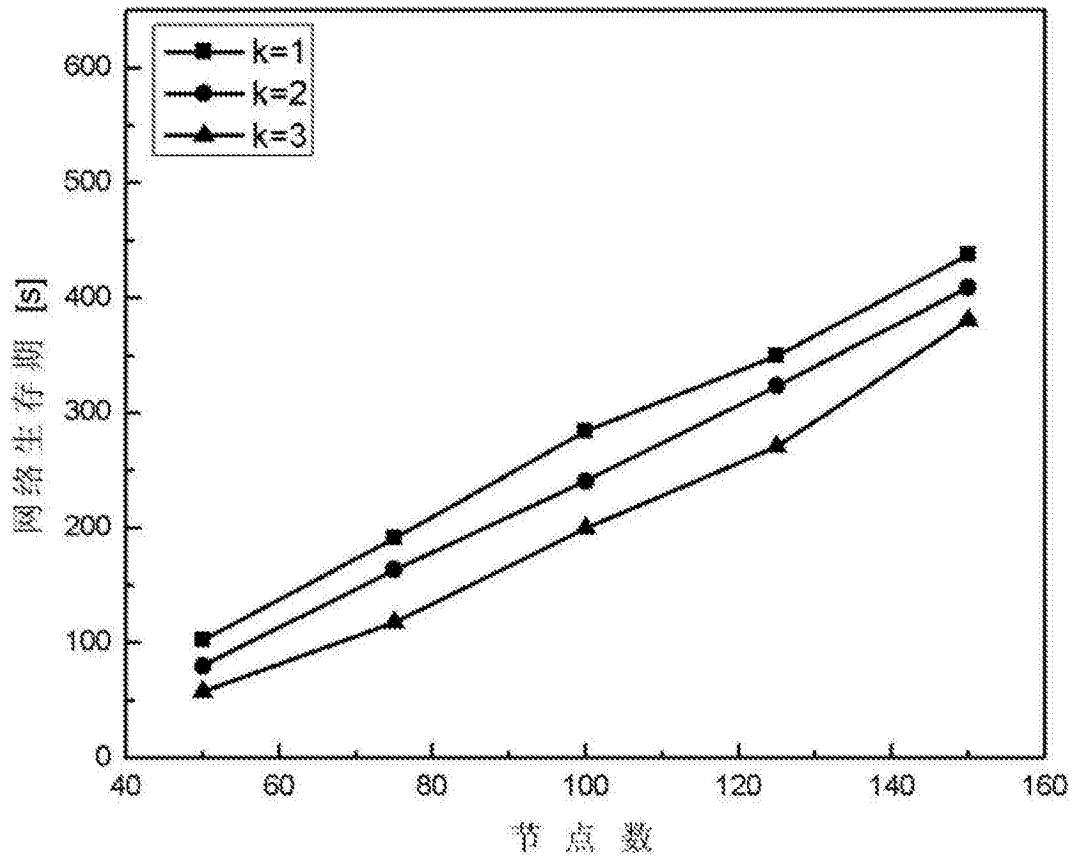


图 7

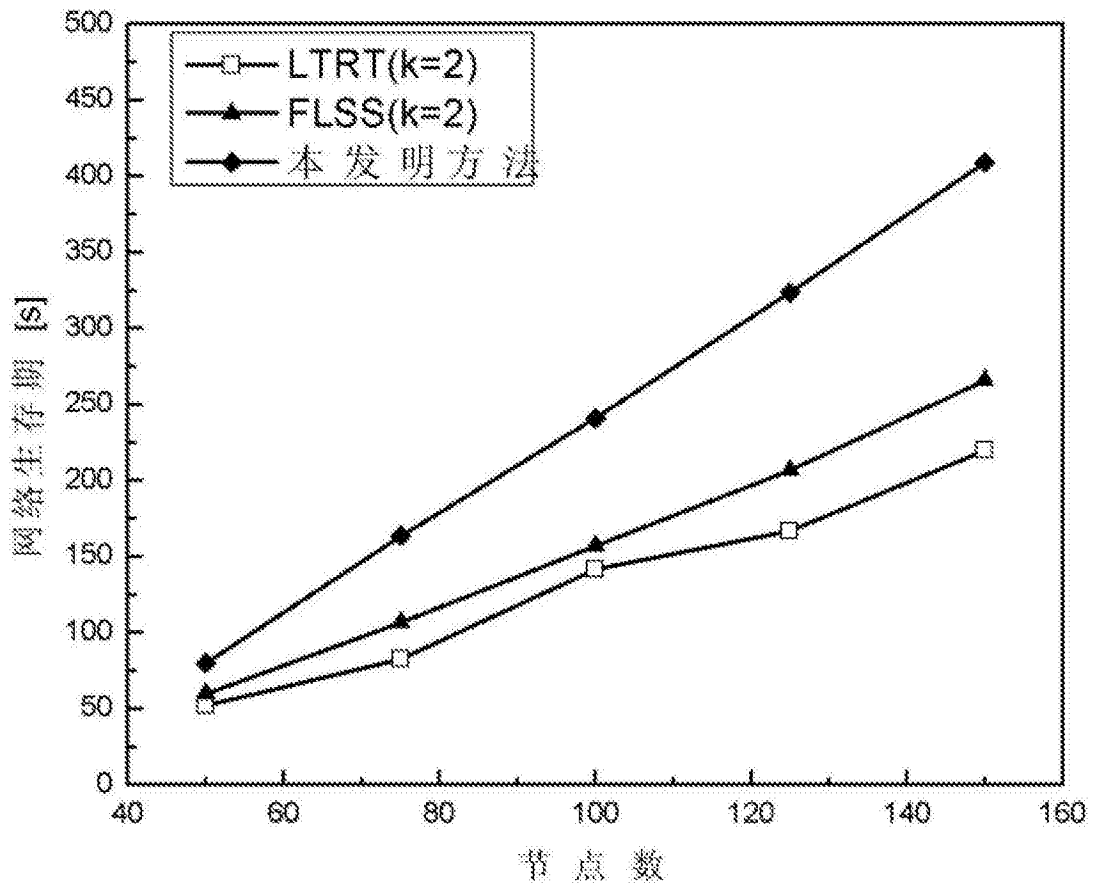


图 8

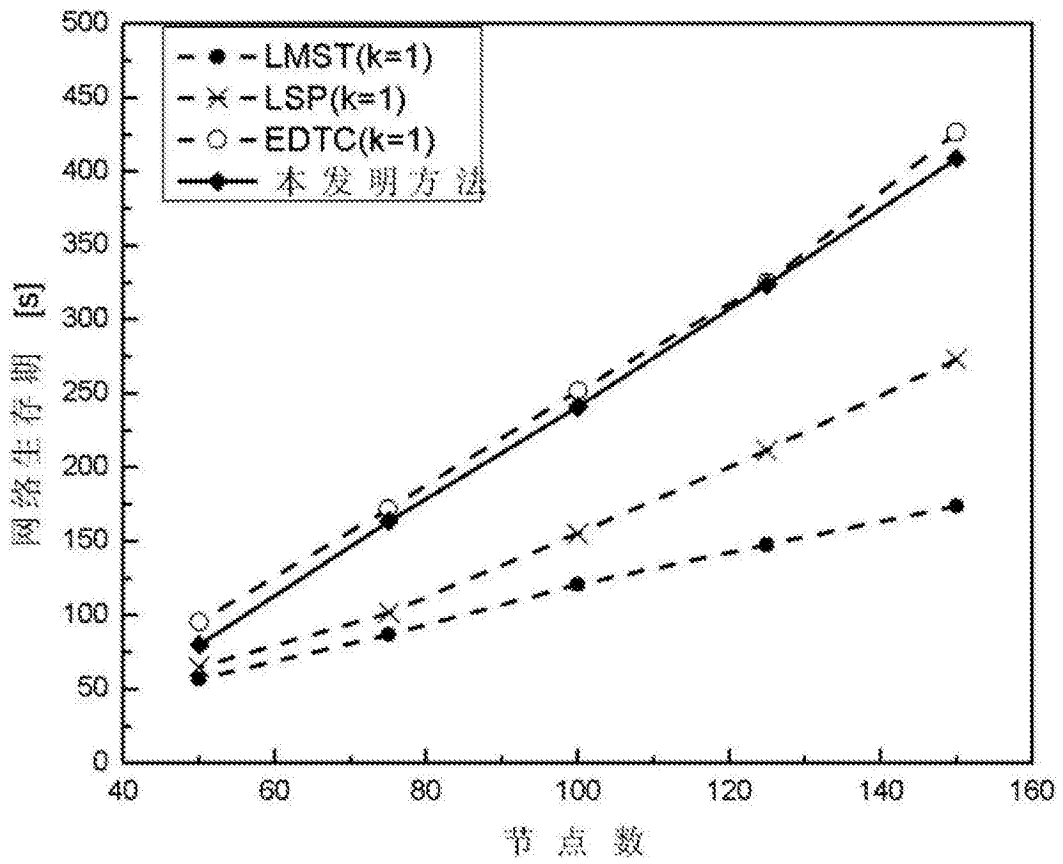


图 9