

基于稀疏 Ad hoc 网络环境的链路预补偿算法研究*

陈春梅, 江虹, 刘雷

(西南科技大学信息工程学院, 四川绵阳 621010)

摘要: 着重研究在通信链路易中断的稀疏 Ad hoc 网络环境中的数据运输问题。提出了链路预补偿算法, 首先给出了关键节点的定义与探测算法, 然后给出了功率补偿算法, 通过增大补偿节点的功率, 增强关键节点周围的连接度, 进而提高数据的接收率和吞吐量; 并且在 NS2 模拟器中, 对该算法搭建仿真平台进行了性能评估。仿真结果表明, 链路预补偿算法能够提高网络的吞吐量和数据接收率。

关键词: 稀疏; Ad hoc 网络; 链路预补偿; 接力路由

中图分类号: TP393 **文献标志码:** A **文章编号:** 1001-3695(2010)03-1095-04

doi:10.3969/j.issn.1001-3695.2010.03.080

Research on links precompensation algorithm in sparse Ad hoc networks

CHEN Chun-mei, JIANG Hong, LIU Lei

(School of Information Engineering, Southwest University of Science & Technology, Mianyang Sichuan 621010, China)

Abstract: The research focuses on communication problem in sparse Ad hoc networks where links easily disrupt. This paper presented links precompensation. Firstly gave the definition of critical nodes and the method to detect them, then proposed power control compensation algorithm, through improving nodes' power to improve the network connectivity and eliminate the impact of the critical node. In the NS2 simulator, designed simulation platform for the scheme. The results show that the links precompensation algorithm can improve network throughput and data delivery ratio.

Key words: sparse; Ad hoc networks; links precompensation; relay routing

稀疏 Ad hoc 网络由于节点密度相对较低, 无线通信范围有限, 节点间易出现间歇性连接, 网络经常出现分割。传统的路由协议在寻找不到源端与目的端的路由时, 数据将被丢弃, 所以不适合稀疏 Ad hoc 网络环境。

节点的移动性、稀疏性、能量限制容易导致网络分割。确保整个网络的连通性, 不仅需要复杂的控制算法, 且很难实现, 在稀疏的环境中几乎是不可能的。因此本文提出了提高网络局部连通性的链路预补偿算法。首先定义了关键节点, 然后设计了关键节点探测算法和预补偿算法, 并给出在 NS2 模拟器中算法的仿真实验结果与分析。

1 关键节点定义

应用于连通、密集型 Ad hoc 网络中的关键节点被定义为由于此节点的失效会导致网络分割成两个或更多部分的节点, 这种关键节点是全局关键节点。而对于稀疏 Ad hoc 网络, 由于网络拓扑可能已经呈现分割状态, 寻找这种全局性的关键节点意义不大。因此本文定义的关键节点是局部关键节点, 这些关键节点的失效将导致其周围某些邻居节点间不能相互通信, 邻居节点可以是一跳邻居也可以是两跳邻居。关键节点的定义如下: 如果节点 a 的邻节点间没有本机的中转就无法互相通信, 则说明节点 a 不能缺少, 此时节点 a 是关键节点; 如果节点 a 的邻节点间可以互相通信, 则说明即使节点 a 出现障碍也不影响其他邻节点间的通信, 此时节点 a 不是关键节点。

如图 1 中, 左边区域与右边区域已经分割, 左边区域节点 A 、 B 、 C 和 D 中的任何一个节点失效都不会影响到其他节点间的通信, 所以左边区域所有的节点都不是关键节点。如果去掉节点 H 及其相关联的通信链路, 其四个邻居节点 J 、 I 、 G 、 F 之间就不能互相通信, 说明节点 H 的影响很大, 地位很“关键”, 此时 H 是关键节点。相反, 如果节点 L 失效, 其邻居节点 I 和 M 仍然能够通过 H 、 J 、 K 互相通信, 所以节点 L 失效与否对其他节点的通信没有影响, 节点 L 不是关键节点。

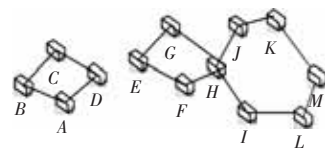


图 1 网络中关键节点示意图

2 关键节点探测算法

依据关键节点的定义对关键节点进行探测。通过在节点与其邻居节点间交互探测消息来计算是否为关键节点, 通过邻居节点间传送同一个数据包来统计邻居节点间的通信链路。节点随机选择其一个邻居节点, 命令此邻居节点向所有其他邻居节点发送一个统计包, 并要求收到此统计包的邻居节点发送应答消息, 依据收到的应答消息个数与发送的统计包的个数 (即邻居节点的个数减 1) 比值 η 来比较邻居节点间的链路连接状况。

收稿日期: 2009-07-09; 修回日期: 2009-08-31 基金项目: 国防基础重点项目资助 (A3120060264)

作者简介: 陈春梅 (1977-), 女, 四川广安人, 讲师, 主要研究方向为计算机通信技术 (infocomputer@163.com); 江虹 (1968-), 男, 重庆人, 副教授, 博士, 主要研究方向为网络通信与系统; 刘雷 (1984-), 男, 硕士研究生, 主要研究方向为计算机网络与通信技术。

网络中每个节点独立发起关键节点探测算法,算法是分布式的。考虑尽量减少探测消息对网络负荷的影响,节点并不是连续发起查找,而是周期性发送探测消息。

关键节点探测算法定义了 search_keynode_pkt1、search_keynode_pkt2 和 search_keynode_pkt3 三个探测消息类型。其中,search_keynode_pkt1 为节点发起探测的命令消息;search_keynode_pkt2 为统计包;search_keynode_pkt3 为应答消息。关键节点查找算法工作原理如图 2 所示。

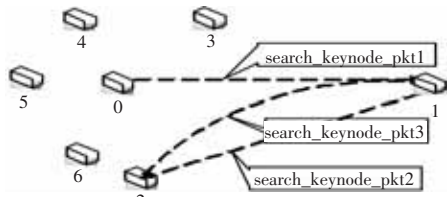


图 2 关键节点查找算法工作原理示意图

以图 2 为例,关键节点探测算法步骤如下:

a) 以 0 节点为例,node0 查找本机的邻居列表,若其邻居节点的个数大于 1,则继续执行,如果没有邻居节点,说明此节点是一个孤立节点,只有一个邻居节点也不符合判断的条件。

b) node0 有六个邻居节点,随机选择其中一个邻居节点 node1,并向其发送 search_keynode_pkt1 探测消息。此消息包括 node0 的邻居节点列表 neighbor_list、邻居节点的个数 nbnum、本机的地址 Ip0addr。

c) node1 接收到 search_keynode_pkt1 消息后,在本机的路由表中删除到 node0 的路由,这样向 node0 的邻居节点发送的数据包就不会通过 node0 转发。

d) 向 node0 的所有邻居节点发送 search_keynode_pkt2 消息,此消息包括 node0 邻居节点的个数 nbnum、node0 的地址 Ip0addr、node1 的地址 Ip1addr。

e) 例如 node2 是 node0 和 node1 的邻居,那么 node2 将会收到 search_keynode_pkt2 的消息,然后 node2 发送给 node0 一个确认消息 search_keynode_pkt3。此消息的目的地址由收到的 search_keynode_pkt2 中拷贝,包括 search_keynode_pkt2 消息中的 node1 地址、node0 的 nbnum。

f) node0 根据接收到的 search_keynode_pkt3 消息的个数与本机的 nbnum 相比较,如果大于 η (η 值可调),则说明是关键节点,即若 node0 失效,则有大于 $(1 - \eta)\%$ 邻节点间不能通信;如果小于 η ,说明 node0 的邻居节点不需要自己的中转,大于 $\eta\%$ 的邻居节点间也能互相发送数据,node0 不是一个关键节点。 η 的值可以根据具体的应用环境和性能指标进行相应调解和设置。

关键节点探测算法的流程图如图 3 所示。

3 补偿节点选择

可以从两个方面对关键节点进行补偿:a) 从节点的移动控制方面,可以让某个节点移动到关键节点的位置,以防关键节点出现故障带来的影响;b) 可以考虑增大其某个邻居节点的发射功率,从而覆盖关键节点的通信区域,这样就解除了关键节点的影响。第一种方法需要设置额外节点,对于节点稀疏的环境不符合实际。所以,本文选择第二种方法,通过增大发射功率覆盖关键节点的通信区域来提高关键节点周围的连通性,消除关键节点的影响,即便关键节点失效,也不会对其他邻

居节点间的通信造成影响。

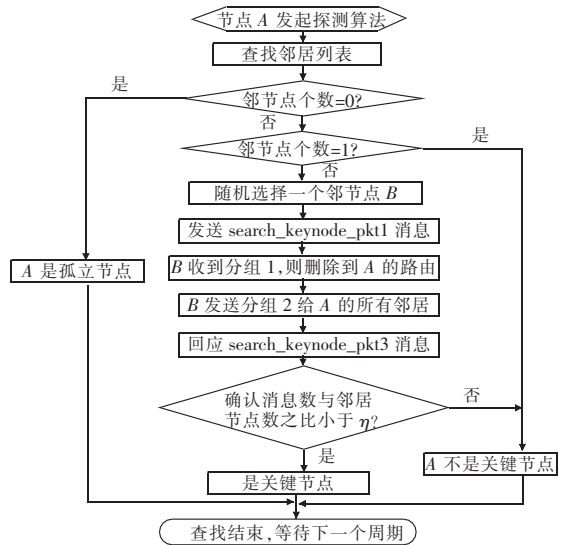


图 3 探测算法流程图

补偿节点的选择要综合考虑补偿节点自身的能力和尽可能降低补偿关键节点对其他节点造成的影响。主要考虑以下几个因素:

a) 增大相同的功率,可以使关键节点周围的连接度增幅最大。补偿节点在关键节点的邻居节点中选取距离关键节点最近的邻居节点,覆盖关键节点的通信区域需要增加的传输功率最小。

b) 作为补偿的节点本身的剩余能量。如果节点本身剩余能量有限,为其他节点补偿则会加速节点能量的消耗,此节点的失效可能会造成新的链路中断。

c) 补偿节点也是关键节点。如果节点本身是关键节点,本身的地位就很关键,需要受到保护,也不易作为补偿节点。

d) 增大功率对其他节点的影响。由前面对功率控制的影响分析可知,增大传输功率可能会对其他节点间的通信造成干扰,降低信道的利用率。但是针对节点稀疏、节点间距离较远的情况,增大传输功率对其他节点间的通信一般不会造成影响。

e) 不同的补偿周期补偿节点的选择。如果一个关键节点其“关键”地位持续几个探测周期,每次选择同一个补偿节点是不合理的,这样会使得补偿节点的能量过快消耗。所以,在本次补偿周期选择的补偿节点,在下一个补偿周期不能再被选,应该选择距离次近的邻居节点。通过这样轮流为关键节点补偿,均衡消耗各节点能量。

综合以上因素,补偿节点选择流程如图 4 所示。

在图 5 中,假设节点 A 是一个关键节点,其无线信号传输半径为 r ,此时节点 B 距离 A 最近,假设同时 B 符合上述所有条件,B 被选为补偿节点。节点 B 增大自己的传输半径为 $d + r$ (d 为补偿节点与关键节点之间的距离),这样,节点 B 就覆盖了节点 A 的通信区域,节点 A 的失效就不会对其他邻居节点的通信造成影响。

4 补偿算法

补偿算法定义三种数据类型:距离请求消息(location request message)、距离应答消息(location reply message)和补偿

请求(compensation request)。前两种消息用于确定补偿节点,第三种消息用于通知补偿节点具体的补偿值。

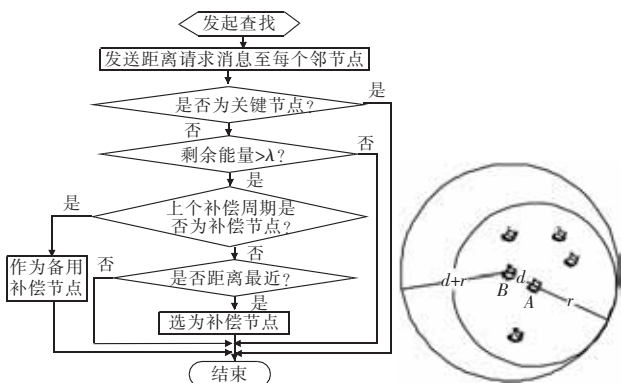


图 4 补偿节点选择流程图 图 5 补偿节点选择示意图

a) 距离计算。利用分组内携带的发射信息和物理层反馈的接收信息,由式(1)计算两节点之间的距离。

$$P_r = P_t [\lambda / (4\pi d)]^\beta G_t G_r \quad (1)$$

其中: P_r 为接收功率; P_t 为发送功率; λ 为载波波长; d 为源节点和目的节点间的距离; G_t 为发射机天线增益; G_r 为接收机天线增益。 β 取决于无线传播模型,对于 TwoRayGround 双径地面反射, $\beta=4$ 。

b) 补偿请求。当节点判断自己是关键节点时,如果其能量即将耗尽,或者出现某种故障即将退出网络,则发起补偿算法,请求其他节点的补偿。

补偿算法的具体步骤如下:

(a) 当节点判断自己是关键节点时,如果需要补偿,首先关键节点发送 location_request message 消息给其所有的邻节点,以寻找距离自己最近的邻居节点。

(b) 邻节点收到此分组后,如果其本身不是关键节点、剩余能量满足条件,并且在上一个周期不是补偿节点,计算与关键节点之间的距离,将此值写入 location_reply 消息,发送至关键节点,如果不满足条件则不回应消息。

(c) 关键节点收到应答消息后,选择距离最近的邻居节点,并发送包含具体功率增加的信息,功率增加后补偿节点的通信半径为两点之间的距离 d 加上关键节点的通信半径 r 。

(d) 补偿节点根据式(1)计算新的发射功率,根据关键节点的状况调整发射功率。

(e) 功率增加后,不能让补偿节点长期保持大功率的发射,节点轮流补偿。

5 仿真结果及分析

仿真实验主要比较扩展补偿算法的 OLSR 协议和传统 OLSR 协议在稀疏 Ad hoc 网络环境中的通信性能,从而确定补偿算法是否能有效改善稀疏 Ad hoc 网络性能。

选择数据接收率、网络吞吐量、能量消耗几个参数作为比较的指标。

节点的移动性会导致网络拓扑不断改变,网络中的关键节点也会随之改变,所以有必要周期性地对关键节点进行查找。查找周期过长,就无法跟踪网络拓扑的改变,关键节点不能及时得到补偿;查找周期过短,则会增大网络的开销。因此合理地选择查找周期是很有必要的。

5.1 查找周期对补偿算法的影响

a) 仿真环境。仿真场景为 $1\ 000\text{ m} \times 1\ 000\text{ m}$ 稀疏节点环境,关键节点探测周期分别为 5 s、10 s、15 s、20 s、25 s;仿真时间为 100 s;节点无线传输半径为 200 m;仿真节点数为 30;节点的运动速度将在 (1,5) m/s 内随机选择;节点停留时间为 1 s;设置 η 为 20%。

b) 数据流设置。采用 CBR(常速数据流)数据流,通信源随机设置 20 个,每个数据源持续发送总时间为 5 s。设定发送速率为每秒两个数据包,数据包长为 1 024 Byte。

选择数据接收率和路由开销作为比较的两个性能指标。路由开销是平均每发送一个应用层数据包需要发送多少路由层的控制分组,即所有发送的路由层数据总和与应用层数据总和之比。

如图 6 所示,随着探测周期由 5 s 增大到 25 s,数据接收率由 55% 下降至 25%,补偿算法的作用很低,路由开销由 30 降低至 15。探测周期为 10 s 时的路由开销比 5 s 时的基本降低一半左右,而数据接收率仅降低了 10% 左右。探测周期在 10~25 s 之间时,路由开销下降不明显,而数据接收率反而下降很快。探测周期的大小对网络性能有很大影响,综合考虑数据接收率和路由开销两个指标,在中低速环境中,探测周期选择在 10 s 左右,既能保证数据的接收率,也不会过重增加网络的负荷。

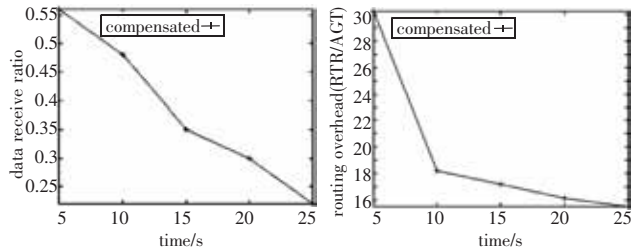


图 6 不同探测周期对数据接收率和网络开销的影响

5.2 不同移动速率下的吞吐量和接收率比较

a) 仿真环境。仿真场景为 $1\ 000\text{ m} \times 1\ 000\text{ m}$ 稀疏节点环境,关键节点探测周期为 10 s;仿真时间为 100 s;节点无线传输半径为 200 m;仿真节点数为 30;节点的运动速度分别将在 (0,1) m/s、(1,5) m/s、(5,10) m/s、(10,15) m/s、(15,20) m/s、(20,25) m/s、(25,30) m/s、(30,35) m/s、(35,40) m/s 中随机选择;节点停留时间为 1 s。

b) 数据流量。为了直接考察路由性能,采用 CBR,通信源设置为 20 个,设定发送速率为每秒两个数据包,数据包长为 1 024 Byte。网络中的数据量最大为 40 Kbps,接近于实际工作中的 Ad hoc 网络的流量负载。

c) 仿真过程。为了对比补偿算法对网络性能的影响,在探测到关键节点后,令关键节点失效一短暂时间,然后恢复正常。每一次运行 NS 模拟器都会导入一个节点运动场景文件和流量生成文件。为了尽可能地实现仿真的可靠性,对于每一组结果数据,都要求采用不同的运动场景仿真 10 次,然后求平均值。总共生成 180 个运动场景文件,需要运行 180 次仿真过程。

d) 结果分析。仿真结果如图 7 所示,扩展了补偿算法的协议在数据接收率和吞吐量上比传统 OLSR 路由协议提高了

一倍左右;随着节点运动速度的增加,补偿算法的优势逐渐减弱,并且,网络在低速的环境中数据接收率也仅能达到 50% 左右。分析原因如下:

(a) 节点的运动速度加快,导致网络的拓扑结构急剧变化。关键节点也在很快地改变,补偿算法跟不上网络结构的变化,可能刚补偿的关键节点很快不再是关键节点,这样补偿算法就不能发挥优势。考虑到补偿算法带来的控制消息的增加,查找周期不能设置得太小。在中低速的环境中,如 1~5 m/s,在每一个查找周期内,节点移动的最大距离仅在 10~50 m,相对于 200 m 的无线传输半径,节点的移动不会使网络的拓扑产生急剧的变化,所以补偿算法在中低速的环境中能很好地改善网络的性能。

(b) 用 setdest 工具生成的是稀疏的随机分布模型,网络拓扑初始化时并不一定是一个全连通的网络,补偿算法只能改善局部的网络性能,对于已经产生分割的网络区域不起作用。所以对于随机移动模型,在稀疏环境中,网络的吞吐量和接收率比较低。

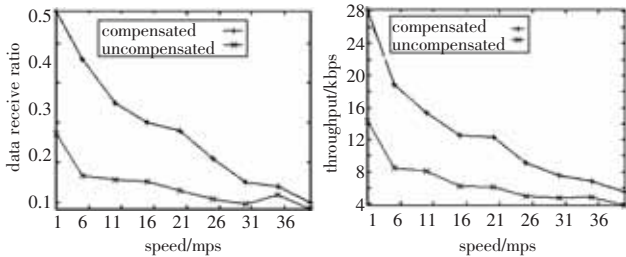


图 7 不同速率下补偿前后吞吐量和数据接收率比较

5.3 补偿前后能耗的比较

本实验模拟一个小型网络,具体分析节点移动对网络拓扑和补偿算法能耗影响。

仿真环境为 500 m × 500 m 仿真区域,5 个移动节点,仿真时间为 50 s,节点的移动速度在 (0,2)m/s 范围内随机选择,查找周期为 10 s,选择在 AODV 路由协议上扩展补偿算法。实验结果如图 8、9 所示。

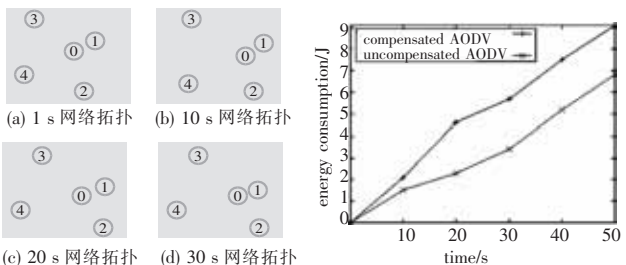


图 8 网络拓扑的变化

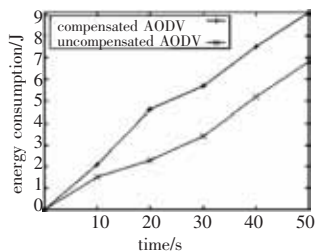


图 9 补偿前后能耗比较

由图 8 (a) 中可以看出,节点 0 为网络中的关键节点, [1~4] 号节点为节点 0 的邻居节点,节点 2 与 3、1 与 2 的通信都需要节点 0 的中转,所以节点 0 的失效会造成整个网络通信中断。由于节点的运动速度较低,网络的拓扑变化不大,30 s 后拓扑基本不变。如果关键节点即将失效则发起补偿请求,节点 1 距离节点 0 最近,为 73 m,同时节点 1 不是关键节点,其剩余能量也满足条件,所以选择节点 1 为补偿节点,增大节点 1 的传输功率对关键节点补偿,并由式(1)计算出节点 1 新的传输功率为 0.978 1 W。10 s 时,节点 0 仍然是关键节点,根据补偿节点选择依据,在这个补偿周期选择次近的节点 4 作为补偿

节点,同时节点 1 的发射功率恢复到原值;20 s 时,节点 1 距离节点 0 已经很近,节点 2 与 3、1 与 2 的通信可以通过节点 1 的中转,节点 0 不再是关键节点,因此 20 s 后不需要再补偿。

由图 9 可以看出,补偿后网络中整体能量消耗略有增加,但是相差不大。影响能耗的主要因素是发送数据分组的多少和传输功率的大小。传输功率增加,相应地增加了节点能量的消耗。但是在关键节点失效后,由于 AODV 路由协议是按需式路由,源节点和目的节点之间没有通信路由时,源节点发送路由查找请求分组、等待应答、超时重传,这些控制分组相应地增加了网络的能耗;补偿算法不是采取连续补偿,也节省了网络的能耗。所以网络的整体能耗相差不大,功率补偿算法并没有使得网络的整体能耗增大很多。

6 结束语

本文给出了关键节点的定义,即关键节点是网络某局部区域中的关键节点,并不是整个网络的关键节点。网络中的关键节点是重点维护的对象,它的失效会造成网络中某些节点通信的中断。提出了链路预补偿算法来预防关键节点失效所造成的影响,并对该算法的仿真结果进行了分析,补偿算法在中低速环境中可以有效地改善稀疏 Ad hoc 网络的性能,提高数据的接收率和吞吐量;通过对模拟环境中节点运动的详细分析,表明功率补偿对网络的能耗影响很小,并分析了其原因。补偿算法简单,容易实现,可以有效提高稀疏 Ad hoc 网络的性能。

参考文献:

- [1] WISITPONGPHAN N, BAI Fan, MUDALIGE P, *et al.* On the routing problem in disconnected vehicular Ad hoc network [C]//Proc of the 26th IEEE International Conference on Computer Communications. 2007:327-339.
- [2] CHAKERES I D, BELDING-ROYER E M. AODV routing protocol implementation design [C]//Proc of the 24th International Conference on Distributed Computing Systems Workshops. 2004:698-703.
- [3] JOHNSON D B, MALTZ D A, HU Y C. Dynamic source routing protocol for mobile Ad hoc network (DSR) [R/OL]. (2003-04-16). <http://www.cs.cmu.edu/~dmaltz/internet-drafts/draft-ietf-manet-dsr-09.txt>.
- [4] AISSANI M, SENOUCI M R, DEMIGNA W, *et al.* Optimizations and performance study of the dynamic source routing protocol [C]//Proc of the 3rd International Conference on Networking and Services. Washington DC: IEEE Computer Society, 2007: 107-116.
- [5] ZHAO W, AMMAR M, ZEGURA E. A message ferrying approach for data delivery in sparse mobile Ad hoc networks [C]//Proc of the 5th ACM International Symposium on Mobile Ad hoc Networking and Computing. New York: ACM Press, 2004: 187-198.
- [6] 王永胜, 吴德伟, 刘勇. 基于 NS2 网络仿真研究 [J]. 计算机仿真, 2004, 21 (11): 257-259.
- [7] 于斌, 孙斌. NS2 与网络模拟 [M]. 北京: 人民邮电出版社, 2006.
- [8] LI Qun, RUS D. Communication in disconnected Ad hoc networks using message relay [J]. *Journal of Parallel and Distributed Computing*, 2003, 63 (1): 75-86.
- [9] FALL K. A delay-tolerant networking architecture for challenged Internets [C]//Proc of Conference on Applications, Technologies, Architectures and Protocols for Computer Communications. 2003: 27-34.
- [10] JAIN S, FALL K, PATRA R. Routing in a delay tolerant networking [C]//Proc of Conference on Applications, Technologies, Architectures and Protocols for Computer Communications. 2004: 145-158.