

文章编号:1007-5321(2012)02-0041-05

基于转发倾向度的固定路径摆渡路由算法

赵广松, 陈 鸣

(解放军理工大学 指挥自动化学院, 南京 210007)

摘要: 为了降低容迟网络(DTN)中摆渡路由算法对摆渡路径的依赖,提出了一种基于转发倾向度的固定路径摆渡路由算法(FTFPF). 区别于当前算法主要关注如何为摆渡节点设计合理的摆渡路径,该算法使得摆渡节点可在较为随意的摆渡路径上运动,而 DTN 依然可取得很好的性能. 根据 DTN 上层应用的不同需求(时延敏感或报文丢失敏感),FTFPF 算法可通过调节算法参数来满足不同的应用需求. 仿真结果表明,FTFPF 算法在报文递交率、平均时延和报文在缓存逗留时间等方面性能均优于基于最优摆渡路径路由算法(OPWP).

关键词: 容迟网络; 接触; 摆渡路由; 递交率; 平均时延

中图分类号: TP393.1

文献标志码: A

Forward Tendency Based Fixed Path Ferry Routing Algorithm

ZHAO Guang-song, CHEN Ming

(Institute of Command Automation, People's Liberation Army University of Science and Technology, Nanjing 210007, China)

Abstract: Forwarding tendency based fixed path ferry routing algorithm (FTFPF) is proposed for delay tolerant network (DTN) without focusing on how to design reasonable ferry routes. This algorithm can maintain DTN good performance even when ferry nodes move on arbitrary ferry routes. According to different performance requirements of DTN applications (delay sensitive or loss sensitive), FTFPF can adjust its parameters to satisfy these different types of requirements. Simulations show that FTFPF outperforms optimized way-points (OPWP) algorithm in many aspects, such as the message deliver ratio, the average delay and the average buffer time of messages etc.

Key words: delay tolerant network; contact; ferry routing; deliver ratio; average delay

DTN^[1]的主要特征是节点间的连接具有间歇性,源节点和目的节点间缺乏完整的端到端连接. DTN 的路由方式不同于传统网络,它采用了存储—搬运—转发的方式进行报文传递. 为了提高节点间的接触机会,文献[2]提出了摆渡路由,即通过在 DTN 中添加摆渡节点,承担报文中继功能. 目前关于摆渡路由的研究很多,但都是关注于摆渡路径的设计. 而本文关注于无法为摆渡节点设计最优路径的场景,设计了 FTFPF 算法. 该路由算法使摆渡节点在非最优的摆渡路径上,依然能较好地为其他节

点提供摆渡服务. FTFPF 算法可通过调节算法参数来满足不同的上层应用需求. 仿真结果表明,FTFPF 算法取得了很好的性能.

1 摆渡节点的益处

1) 增强 DTN 的性能. 通过在 DTN 中放置一些具有摆渡功能的节点,网络的报文递交率大幅度上升,平均时延显著下降.

2) 增强社区间报文的交换. pocket switched network 这样的 DTN 通常具有社区特性. 没有摆渡节点

收稿日期: 2011-05-20

基金项目: 国家重点基础研究发展计划项目(2012CB315806); 江苏省自然科学基金项目(BK2010133); 国家自然科学基金项目(61070173,61103225)

作者简介: 赵广松(1984—),男,博士生, E-mail: guangsongzhao@126.com; 陈 鸣(1956—),男,教授,博士生导师.

在社区间进行报文的中继,网络性能难以令人满意.

3) 基于社会网络分析的 Bubble Rap^[3] 路由算法通常选择高连接度节点作为报文转发的中继节点.这会导致少量高连接度节点的负载加重而崩溃.通过让摆渡节点在网络中移动,将流量转移到摆渡节点上,增强系统的健壮性.

针对摆渡路由的研究主要关注于如何设计合理的摆渡路径.但实际中,很多原因限制了摆渡路径的设计,包括:① 设计最优摆渡路线是 NP-Hard 问题;② 建筑物、湖泊等阻挡的因素,限制了最优摆渡路径的穿行;③ 无权控制摆渡节点的行走路线.本文的关注点是在无法设计最优摆渡路径时提出了 FTFPF 算法.转发倾向度同时考虑了转发节点与摆渡节点的接触概率及资源限制.

2 FTFPF 算法设计

2.1 摆渡节点运动行为的设计

1) 摆渡路线的要求. FTFPF 仅要求摆渡路线尽量穿越网络中节点密集的地理区域.如图 1 所示,要求摆渡路线尽量穿过 3 个节点密集区域.

2) 摆渡节点的速度设置.为了设计的简易性,摆渡节点采用恒定速度运动.

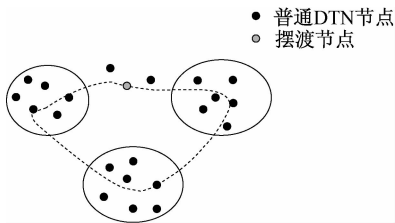


图1 摆渡路线的基本要求

2.2 转发倾向度的定义

FTFPF 算法采用转发倾向度作为中继节点的选择依据.转发倾向度的定义为

$$ap + bB$$

其中, p 为该节点与摆渡节点接触的概率; B 为该节点的剩余缓存; a 和 b 为各自权重.对 B 进行归一化处理,转发倾向度重定义为

$$ap + b \frac{B - B_{\min}}{B_{\max} - B_{\min}} = ap + be$$

其中 $e = B/B_{\max}$ 为缓存空闲率.当 $a \gg b$ 时,节点会倾向性地将报文转交给与摆渡节点接触概率较大的节点,使得跨社区报文可尽快被递交给摆渡节点,因此此时报文平均时延较小.当 $b \gg a$ 时,节点会倾向性地将报文转交给缓存剩余空间大的节

点,缓存溢出较小,路由算法的报文递交率就较大.因此, a 和 b 的取值体现了 DTN 是关注时延大小,还是报文成功递交率,参数的取值可由不同的应用需求决定.

2.3 FTFPF 路由过程

FTFPF 算法需要 2 个输入:① 每个节点的社区类别;② 每个 DTN 节点与摆渡节点的接触概率.在已知 2 个输入的前提下,FTFPF 算法的路由过程如下.

1) 当 2 个普通节点 A 与 B 接触时,它们首先将目的节点为对方的报文递交给对方.

如果 2 个节点属于同一社区,那么 A 与 B 比较各自中心度 (centrality) 测度大小.如果 B 的 centrality 测度大于 A ,那么 A 将缓存中目的节点属于该社区的报文转交给 B ;反之亦然.然后比较转发倾向度,如果 B 的转发倾向度大于 A ,那么 A 将目的节点不属于该社区的报文转交给节点 B ;反之亦然.

如果 2 个节点不属于同一社区,那么 A 将缓存中目的节点与 B 属于相同社区的报文转交给 B ; B 做同样操作.然后 A 与 B 比较转发倾向度的大小,如果 B 的转发倾向度大于 A ,那么 A 将目的节点不属于 A 所属社区的报文转交给 B ;反之亦然.

2) 当普通 DTN 节点 A 与摆渡节点 F 接触时, A 会把缓存中目的节点与 A 不属于相同社区的报文转交给 F ,依靠 F 将这些报文摆渡到相应的社区中.同时, F 会把缓存中目的节点与 A 属于相同社区的报文转交给 A .

2.4 社区标记与接触概率计算

在运行 FTFPF 算法前,需要一个预热期.根据预热期的统计数据,获取 FTFPF 路由算法的输入.

1) 普通节点的社区标记.本文采用分布式 K-CLIQUE 算法来检测社区,它可以使每个 DTN 节点能给自己打上社区标签.

2) 普通节点与摆渡节点的接触概率计算.通过查看预热期的接触次数来计算接触概率值.

3 性能评估与比较

3.1 实验场景的设置

本文采用赫尔辛基科技大学开发的离散事件模拟器 ONE 作为仿真平台.实验场景模拟了 3 个 DTN 节点密集区域,每个节点在某个固定范围的地理区域内随机运动.每个地理区域内部存在几个兴趣点 (POI).FTFPF 算法下的摆渡路线就是把

每个地理区域的中心用直线相连,形成摆渡线路,如图 2 所示.

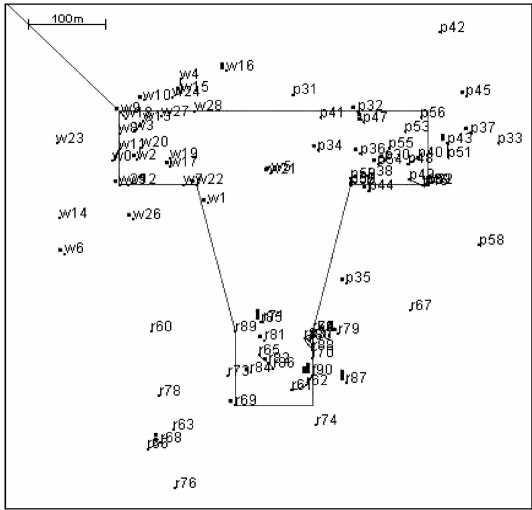


图 2 FTFPF 算法下的摆渡路线

3.2 实验参数设置

1) 普通节点的缓存设置为 5 MB,缓存采用先入先出(FIFO)置换策略,移动速度在 $[0.5, 2]$ m/s 均匀分布,共 90 个普通节点;摆渡节点的速度为 5 m/s,缓存容量设置为 1 GB.

2) 节点无线设备的覆盖范围为 20 m,传输带宽为 250 KB/s;节点访问每个 POI 的概率为 0.1.

3) 平均每隔 20 s,DTN 网络就产生 1 条报文(源和目的地址均匀分布,称这个流量为基准流量);每条报文大小为 100 KB.

4) 整个仿真过程模拟时长为 12 h.

3.3 Bubble Rap 算法、OPWP 算法与 FTFPF 算法性能比较

为了比较公平性,允许 OPWP 算法^[4]中的普通节点也参与报文中继.将 FTFPF 算法、Bubble Rap 算法、OPWP 算法及修改后的 OPWP(OPWPM)算法进行性能比较.其中 OPWP 算法下摆渡节点的运动路线如图 3 所示.

1) 各算法的递交成功率累积分布与时延的变化关系

设置报文的生存期为 12 h,网络流量为基准流量.由图 4 可见,在相同时延限制下,FTFPF 算法可取得最高的递交率,OPWP 算法和 OPWPM 算法次之,Bubble Rap 算法的递交率最低.

2) 各算法下递交率和平均时延随跨社区报文比例变化情况

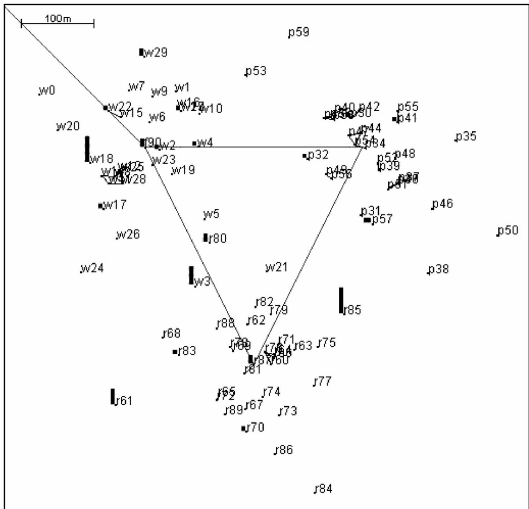


图 3 OPWP 算法下的摆渡路线

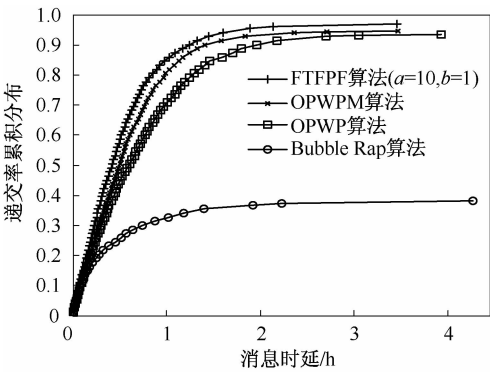


图 4 各路由算法下递交率累积分布与时延的关系

参数设置与 3.2 节 1) 相同.从图 5 可以看出,跨社区报文由 10% 变化到 60% 时,FTFPF 算法、OPWP 算法和 OPWPM 算法的递交率均可保持在 94% 以上.而 Bubble Rap 算法的递交率大幅下降.在 OPWP 算法、OPWPM 算法和 FTFPF 算法中,节点可将报文上载给摆渡节点,减少缓存溢出,降低报文丢包率.

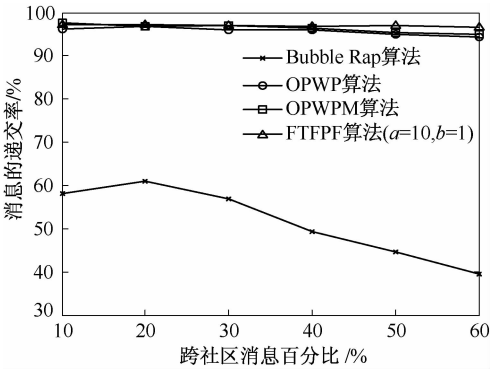


图 5 递交率与跨社区报文比例的关系

3) 各算法下报文在缓存中逗留时间与跨社区报文比例关系

参数设置与 3.2 节 1) 相同. 由图 6 可见, 随着跨社区报文比例的增长, OPWP 算法的缓存逗留时间迅速增大. 而在 FTFPF 算法中, 节点可将报文递交给与摆渡节点接触概率较大的节点, 报文可快速上载给摆渡节点, 降低了报文在缓存中的逗留时间, 平均逗留时间仅为 100 s. 而 Bubble Rap 算法的报文平均缓存逗留时间仅为 200 s 左右, 原因在于该算法缺乏摆渡节点, 造成节点缓存溢出(见图 7), 因此报文的平均缓存逗留时间较短.

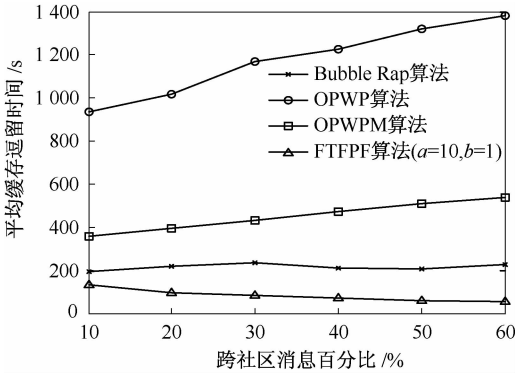


图 6 缓存逗留时间与跨社区报文比例的关系

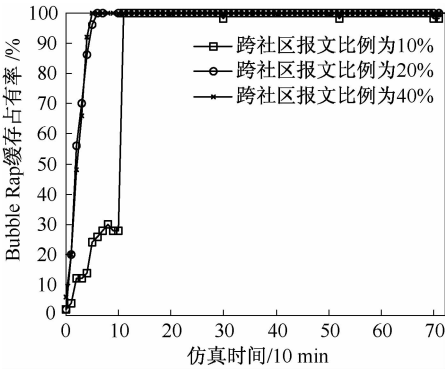


图 7 在 Bubble Rap 算法下节点 20 的缓存占有率变化情况

4) 在 FTFPF 算法下, 与摆渡节点接触概率较大的节点缓存占有率随跨社区报文的比例变化

从各社区中挑选 1 个与摆渡节点接触概率较大的节点(节点 20、56 和 86), 跟踪其缓存占有率随跨社区报文比例变化情况. 由图 8(a) 可见, 当 $a = 1$ 、 $b = 10$ 时, 3 个节点的缓存占有率随跨社区报文增加而缓慢增长. 因为当 $a = 1$ 、 $b = 10$ 时, 缓存空闲率大的节点易被选为跨社区报文的中继节点. 一旦节点缓存占有率变高, 就很难再被选作中继节点, 避免了

节点的缓存占有率进一步升高. 由图 8(b) 可见, 当 $a = 10$ 、 $b = 1$ 时, 节点 20、56 和 86 的缓存占有率随跨社区报文比例增长而迅速增加. 即使节点缓存占有率变得很高, 它们仍被选作报文的中继节点, 因此节点缓存占有率一直较大.

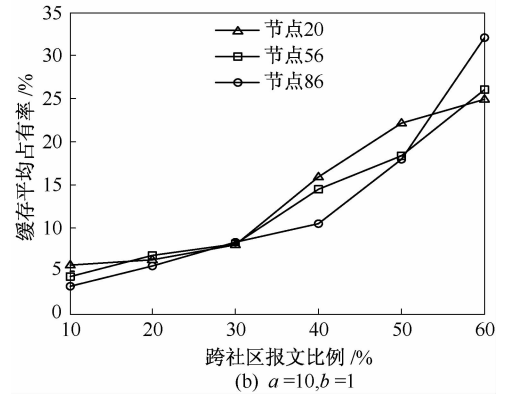
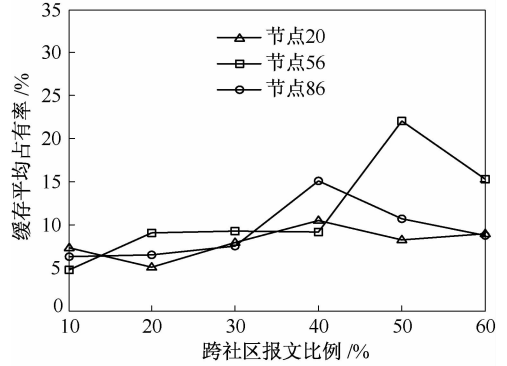


图 8 FTFPF 算法下缓存占有率与跨社区报文的比例变化

如图 9 所示, 当 $a = 1$ 、 $b = 10$ 时, 节点 20 的缓存占有率仅在 $[0, 25\%]$ 区间内变化, 在整个仿真过程中较为平稳. 当 $a = 10$ 、 $b = 1$ 时, 节点 20 的缓存空间占有率波动剧烈.

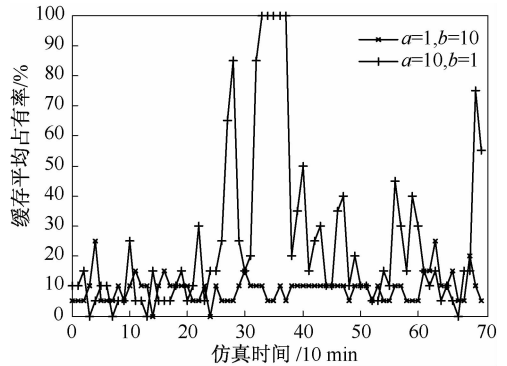


图 9 跨社区报文比例为 60% 时, 节点 20 在不同参数设置下, 缓存平均占有率的变化

5) FTFPF 算法的参数与性能关系

下面考察不同数目摆渡节点条件下, a/b 与平均时延、递交率的关系. 流量为基准流量的 2 倍, 跨社区报文比例为 60%. 由图 10 可见, a/b 与平均时延成反比. 由图 11 可见, 只有 1 个摆渡节点时, a/b 与报文递交率成反比. 摆渡节点数增加到 3 个时, 递

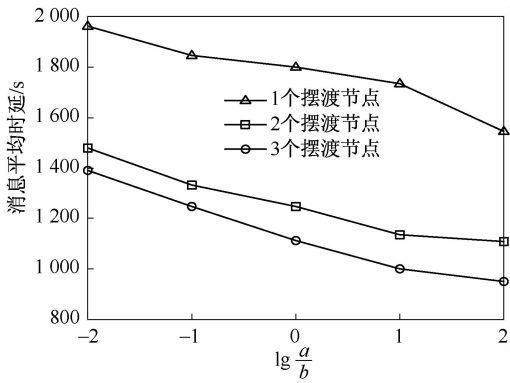


图 10 a/b 与平均时延的关系

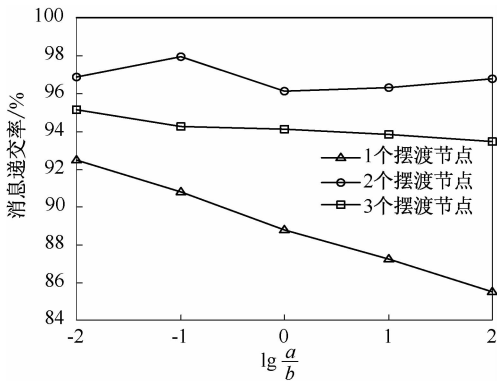


图 11 a/b 与递交率关系

交率不再随着 a/b 增大而下降. 这是由于随着摆渡节点增加, 节点的缓存溢出越来越少. 因此, 当摆渡节点达到一定数量时, 通过增大 a/b 值不仅不影响递交率, 而且可以降低报文的平均时延.

4 结束语

本文设计了基于接触概率的固定路线摆渡路由算法, 使得 DTN 节点可以主动将跨社区报文汇聚给转发倾向度大的节点, 从而改善网络性能.

FTFPF 算法不需要对摆渡路径进行复杂的规划, 也不需要干预普通节点的运动行为, 仿真结果表明, FTFPF 算法能取得比 OPWP 算法更好的递交率. 针对 DTN 上层应用属于时延敏感或丢失敏感的类型, 可通过调节转发倾向度的权重参数, 使 FTFPF 算法适应不同应用的性能需求.

参考文献:

- [1] Kevin F. A delay-tolerant network architecture for challenged Internets[C]//SIGCOMM 2003. Karlsruhe: ACM Press, 2003: 27-34.
- [2] Zhao Weirui, Ainmar M, Zegura E. A message ferrying approach for data delivery in sparse mobile ad hoc networks [C]//MobiHoc 2004. Roppongi: ACM Press, 2004: 187-198.
- [3] Pan Hui, Crowcroft J. Bubble rap: social-based forwarding in delay tolerant networks[C]//MobiHoc 2008. Hong Kong: ACM Press, 2008: 241-250.
- [4] Tariq M, Ammar M, Zegura E. Message ferry route design for sparse ad hoc networks with mobile nodes[C]//MobiHoc 2006. New York: ACM Press, 2006: 37-48.