

容迟网络中基于社会自私性的路由算法

周朝荣^{1,2}, 徐小琼^{1,2}, 马小霞¹, 杨柳¹

(1. 四川师范大学物理与电子工程学院 成都 610101; 2. 无线传感器网络四川省高校重点实验室 成都 610101)

【摘要】容迟网络中, 由于资源受限, 节点设备会随着资源的可用状况表现出一定程度的自私性。此外, 不同的消息对应着不同的业务类型, 为了保证服务质量, 需要在路由算法以及缓存管理中考虑消息的优先级。考虑实际容迟网络中的社会自私性与消息优先级两方面的特点, 在缓存管理机制设计的基础上, 提出了对应的路由算法。基于真实移动轨迹的仿真实验表明, 该算法优于现有的路由算法, 能够在提高消息交付率的同时保持较低的网络开销。

关键词 缓存管理; 容迟网络; 消息优先级; 网络开销; 社会自私性

中图分类号 TP393 **文献标志码** A **doi**:10.3969/j.issn.1001-0548.2016.02.016

Routing Algorithm Based on Social Selfishness in Delay Tolerant Networks

ZHOU Zhao-rong^{1,2}, XU Xiao-qiong^{1,2}, MA Xiao-xia¹, and YANG Liu¹

(1. School of Physics and Electronic Engineering, Sichuan Normal University Chengdu 610101;

2. Key Laboratory of Wireless Sensor Networks, Sichuan Province Higher Education System Chengdu 610101)

Abstract In delay tolerant networks (DTNs), the node equipment due to the limited resources behaves selfishly to some degree with the situations of available resources. Moreover, different messages may correspond to different types of services. In order to guarantee the quality of service (QoS), the message priority should be taken into account in routing algorithms and buffer management. In this paper, a routing algorithm is proposed by considering the two features in actual DTNs, i.e., the social selfishness and message priority. The extensive simulation experiments with real mobility traces show that the proposed algorithm can enhance the message delivery probability, keep the low network overhead, and perform better than existing routing algorithms.

Key words buffer management; delay tolerant network (DTN); message priority; network overhead; social selfishness

近年来, 容迟网络(DTN)受到了研究人员的重点关注, 并应用于深空通信、战场、灾区、偏远山区等极端环境。DTN不具有传统MANET(mobile Ad-hoc network)中网络拓扑始终连通的理想假设, 这就使得传统MANET中的路由协议不再适用, 相应地, 路由算法成为DTN相关研究中的关键。这类路由算法通常基于“存储-携带-转发”的消息传输机制, 节点在接收到目的地为其他节点的消息时, 先将该消息存储在本地缓存中, 并携带该消息移动直至遇到合适的中继节点进行转发^[1]。

目前, 容迟网络中已有的路由算法主要从时间、空间以及正面社会属性等方面出发设计路由机制, 并且通常假设节点愿意帮助其他节点携带或者转发消息, 较少考虑节点设备资源受限的因素。而在实

际情况中, 节点设备的资源尤其是能量通常是受限的, 帮助其他节点转发消息将会导致资源的额外开销。因此, 节点会随着自身资源的占用情况表现出一定的社会自私性, 即拒绝帮助其他节点携带或转发消息。此外, 不同节点产生的不同消息在重要性以及时延限制等方面有着重要的区别。为了保证服务质量, 消息在转发的过程中应该设置不同的优先级。进一步地, 考虑到节点缓存的限制, 节点设备只能携带有限数量的消息, 那么当节点缓存已满时, 新消息到达还需要考虑如何根据消息的优先级分配缓存的问题, 即节点的缓存管理。

1 相关工作

文献[2]提出了基于传染病的路由算法, 其算法

收稿日期: 2015-03-17; 修回日期: 2016-03-02

基金项目: 四川省教育厅重点项目(14ZA0033, 15CZ0004); 无线传感器网络四川省高校重点实验室开放项目(SCWSN201404)

作者简介: 周朝荣(1975-), 男, 博士, 副教授, 主要从事无线网络方面的研究。

机制如下：当某一移动节点接触到其他节点后，无条件地将所携带的消息转交给其他节点。该算法在初始阶段交付率较高，但随着时间的推移，中继节点帮助其他节点携带或者转发消息所导致的额外能量开销会加快这类节点的能量枯竭；此外，缓存空间的迅速占用将导致新消息无法被转发，最终降低交付率。为此，文献[3]提出基于转发概率的路由算法PROPHET，该算法通过比较当前节点与邻居节点到目的节点的预估交付率来确定是否转发消息，这可在一定程度上减少网络中的消息副本数，降低节点设备的能量消耗，但该算法同样假设中继节点无条件地帮助转发，不涉及节点的转发意愿。文献[4]提出了社会自私性敏感的路由算法(social selfishness aware routing, SSAR)，虽在一定程度上考虑节点的自私性，但忽略了节点自私性会随着资源占用情况发生变化这一因素。而在实际情况中，当节点设备难以保证自身生存时是不愿意帮助其他节点携带并转发消息的；但在资源占用较为宽松的情况下又会表现出一定程度的合作倾向，即愿意帮助其他节点携带并转发消息。此外，现有的研究未涉及消息优先级的问题，尤其是针对不同的优先级如何在消息转发以及缓存管理等方面区别对待，以最终保证服务质量、提高节点用户的满意度。

为了避免现有路由算法的不足，考虑消息优先级与缓存管理两方面的特点，并在度中心性这一社会属性的基础上结合社会自私性会随着资源占用情况发生改变的特点，本文提出了消息优先级与社会自私性的路由算法(message priority and socially selfish routing, MP_SSR)并基于真实的节点移动轨迹以及随机的移动场景设计仿真实验。

2 消息优先级与社会敏感性路由算法

本节首先考虑消息的优先级，然后，基于消息优先级设计对应的缓存管理策略，最后，结合度中心性与社会自私性两类社会属性设计路由机制。

2.1 消息优先级

在传统的DTN路由算法中，各个节点在转发消息时，缓存中的消息均按照随机顺序或者先进先出(first in first out, FIFO)的顺序进行转发。但在实际情况中，不同节点的不同消息因内容形式的不同，或者不同的响应要求，往往对应着不同的优先级。为了模拟真实环境，根据消息的特点，可将消息分为不同的类型，如语音(voice)、视频(video)、短信(message)、邮件(email)，其优先级分别定义为 P_1 、 P_2 、

P_3 、 P_4 ，在每条消息产生时加前缀来区别。在转发时，优先级高的消息先转发，优先级低的消息后转发，若两条消息的优先级一致，则按照FIFO方式进行转发，从而使得高优先级的消息极早被转发，确保不同业务类型的服务质量。

2.2 缓存管理

由于移动性的要求，节点设备通常对体积与质量有着特殊的要求，这导致节点的缓存资源受限。为了给待接收的消息预留缓存空间，节点缓存饱和时需要丢弃部分消息。最常见的缓存管理是FIFO策略，其实质是删除缓存中存储时间最长的消息，但未考虑消息自身的特点以及节点对该消息的交付能力。为此，消息丢弃策略需要考虑消息的重要性以及预估的交付概率两方面因素，通过比较新消息与缓存中已有消息的优先级与预估交付概率，来更新节点的缓存。消息是否被缓存，通过比较缓存概率(cache probability, CP)实现，对应的缓存概率CP为：

$$\text{New_CP} = \alpha * \text{New_MsgPt} + (1 - \alpha) * \text{New_PerdFor} \quad (1)$$

$$\text{Old_CP} = \beta * \text{Old_MsgPt} + (1 - \beta) * \text{Old_PerdFor} \quad (2)$$

式中，新消息的优先级为New_MsgPt；缓存中已有消息的优先级为Old_MsgPt；预估新消息以及已有消息通过该节点成功交付目的节点的概率分别为New_PerdFor与Old_PerdFor； α 、 β 分别为消息优先级与预估交付概率所对应的调节参数。

缓存管理策略如下：新消息到达时，若节点缓存已满，比较新消息与缓存区中已有消息的缓存概率CP，若新消息的缓存概率小于缓存区中已有消息的缓存概率，不接收该消息；若新消息的缓存概率大于缓存中某一消息的缓存概率，删除该消息，让出空间来接收新消息。缓存管理策略同时兼顾新消息与缓存中已有消息的优先级以及不同消息的预估交付概率两方面的因素，不仅有助于提高消息的交付率，还可提升节点的满意度。

2.3 社会属性

在DTN中，节点设备的携带者通常为具有社会关系的人或者群体，相应地，节点表现出一定的社会属性。MP_SSR算法综合考虑节点社会属性中的度中心性以及社会自私性，设计基于节点转发意愿的消息转发机制。

2.3.1 度中心性

节点的影响力取决于该节点在网络中的位置，位置越中心的节点其影响力通常越大。社交网络分

析中, 通常采用度中心性(degree centrality, DC)^[5]描述节点的社会影响力。在包含 N 个节点的网络中, 节点最大可能的度值为 $N-1$, 为方便比较可对度中心性进行归一化处理, 相应地, 度为 K_i 的节点的归一化度中心性为:

$$DC_i = \frac{K_i}{N-1} \quad (3)$$

节点的度中心性越大, 所关联的其他节点越多, 该节点就越有可能位居网络的中心, 具有更大的影响力, 将其选作中继节点转发消息有可能达到更高的交付性能。

2.3.2 社会自私性

在真实的社会网络中, 人通常具有一定的自私性。节点设备通常由人携带, 并因自身资源的限制, 往往不情愿为其他节点转发消息, 或者仅仅帮助那些与自己有一定社会关系的节点转发消息^[6]。此外, 即便转发消息, 社会关系的亲疏也会导致不同的表现, 相对于关系比较疏远的, 人们更情愿为亲近者提供更好的服务^[4]。因此, 根据节点的可用剩余能量, 可将网络中的节点分为3大类:

1) 慷慨节点 S_0 : 这类节点在网络中有持续的能量供给以及较大的缓存空间, 在整个网络的运行周期可无条件地帮助其他节点转发消息。

2) 半自私节点 S_1 : 这类节点因为能量的限制, 仅可帮助与自身有亲密关系的节点转发消息, 如同一社团的其他节点。

3) 极度自私节点 S_2 : 这类节点的剩余能量极其有限, 仅可满足自身的生存。因此, 只接受目的节点为自身的消息, 且不为其他节点转发消息。

在消息转发的过程中, 首先确定节点的自私性, 然后根据节点的自私性转发消息。节点自私性的确定方法如下: 初始阶段, 网络中仅仅包含极度自私节点与慷慨节点, 分别占总节点数目的比例为 ξ 与 $1-\xi$; 为了保证网络的连通性, 极度自私节点从度中心性较低的节点中选取, 随着能量的消耗, 节点的自私性随之改变: 当节点能量低于初始能量的 μ 时, 慷慨节点变为半自私节点; 当节点能量低于初始能量的 ν 时, 半自私节点变为极度自私节点, 其中, $\nu < \mu$ 。此外, 节点间的社团关系可以通过社团检测算法确定^[5]。

3 算法设计与评估指标

本节给出MP_SSR算法的流程、对应的伪码以及性能评估指标。

3.1 算法流程以及对应的伪码

本文算法结合度中心性以及社会自私性, 同时考虑社交网络的小世界特性, 即消息只经过较少的中继节点就可到达目的节点^[7]。当消息的转发跳数小于3时, 将消息转交给度中心性较高的节点。算法的流程如下:

- 1) 根据节点的当前能量, 确定其自私性。
- 2) 节点根据消息优先级对缓存中的消息进行排序。
- 3) 节点 N 与 M 接触之后, 若 M 中的消息的转发跳数小于3, 且 N 的度中心性大于 M , 则 M 向 N 转发消息。
- 4) 若 M 中的消息跳数大于或等于3, 则根据自私性进行如下的操作:

① 若 N 为极度自私节点 S_2 , 判断 M 的消息缓存中是否有目的地为 N 的消息; 若有, 则 M 按照消息的优先级向 N 转发这些消息。

② 若 N 为半自私节点 S_1 , 判断 N 与 M 是否归属同一个社团。若是, 比较 N 的预估交付概率是否高于 M 的预估交付概率。若 N 的预估交付概率高于 M , 则 M 按照消息的优先级向 N 转发消息。

③ 若 N 为慷慨节点 S_0 , 比较 N 的预估交付概率是否高于 M 的预估交付概率, 若是, 则 M 按照消息的优先级向 N 传递消息。

类似的操作也对节点 N 进行。整个算法流程对应的伪码如下, 其中, 预估交付概率 $PerdFor$ 的计算参照PROPHET^[3]中的方法。

算法: MP_SSR算法流程, 节点 M 的伪代码。

- 1) 节点根据剩余能量确定自私性;
- 2) 将缓存中的消息按照消息优先级的方式进行排序;
- 3) for 消息 i in 节点 M do
- 4) if N 已经与 M 断开连接 then
- 5) break;
- 6) end if
- 7) if 消息的当前跳数小于3 then
- 8) if $DC_N \geq DC_M$ then
- 9) 向节点 N 转发这条消息;
- 10) end if
- 11) else if 消息的当前跳数大于或等于3 then
- 12) switch (节点 N 的自私性)
- 13) case S_0 : if $(PerdFor)_N \geq (PerdFor)_M$ then
- 14) 向节点 N 转发这条消息; end if
- 15) case S_1 : if M 与 N 属于同一个社团 then
- 16) if $(PerdFor)_N \geq (PerdFor)_M$ then

- 17) 向节点 N 转发这条消息; end if
 18) end if
 19) case S_2 : if N 为消息的目的节点 then
 20) 向节点 N 转发这条消息; end if
 21) else
 22) continue
 23) end if
 24) end for

3.2 性能评估指标

路由算法常用的性能评估指标包括消息交付率与网络开销, 考虑到现有算法较少涉及消息优先级的因素, 定义新的性能评估指标: 加权消息交付率 (weighed_delivery_prob, WDP), 即基于优先级成功转发的加权消息数占网络中总的加权消息数的比例, 此处的权值为消息的优先级, 对应的公式为:

$$WDP = \frac{P_1N_1 + P_2N_2 + P_3N_3 + P_4N_4}{P_1M_1 + P_2M_2 + P_3M_3 + P_4M_4} \quad (4)$$

式中, N_1 、 N_2 、 N_3 、 N_4 分别对应成功交付的语音、视频、短信、邮件消息数目; M_1 、 M_2 、 M_3 、 M_4 分别为语音、视频、短信、邮件的消息总数; 权值简单选取为消息的优先级。加权消息交付率能够更好地反映重要消息的交付性能以及节点的满意度。

4 实验结果分析

ONE(opportunistic network environment simulator)^[8]是基于JAVA的DTN路由算法仿真平台, 具有可配置的图形化界面, 便于仿真结果的分析与处理。为此, 本文基于ONE 1.5.1平台, 使用和配置相应的能量模块, 验证提出的算法的性能。

4.1 仿真场景介绍

考虑到仿真实验的真实性与可用性, 采用3种典型的仿真场景: Pmtrs^[9]真实移动轨迹、Infocom06^[10]真实移动轨迹以及随机场景Random。

Pmtrs记录了米兰大学49个移动设备的真实移动轨迹, 这些设备分发给教员、博士生以及技术人员, 记录持续19天。记录完成后对相关数据进行处理, 同一时刻接触双方均有记录的数据被保留, 同一时刻仅有单方面记录的数据被舍弃。最终, 共有11 895条接触记录被保留。

Infocom06记录了2006年IEEE INFOCOM迈阿密会议78个参会人员携带的蓝牙设备以及20个静止设备之间的接触信息。

3种仿真场景的说明如表1所示。

表1 仿真场景

仿真数据	Pmtrs	Infocom06	Random
设备	PMTRs	iMotes	PHONE
网络类型	-	蓝牙	蓝牙
持续时间/d	19	3	0.5
间隔时间/s	-	120	-
节点数目	49	98	40

4.2 仿真参数设置

为了体现社会自私的影响并保证较高的网络连通性, 节点自私性的初始设置如下: 极度自私节点占总节点的比例 $\xi = 5\%$, 优先从度中心性较低的节点中选取, 其余为慷慨节点。随着能量的消耗, 节点的自私性随之改变: 慷慨节点 S_0 变为半自私节点 S_1 的能量阈值比为 $\mu = 50\%$; 半自私节点 S_1 变为极度自私节点 S_2 的能量比阈值为 $\nu = 20\%$ 。为了表征消息的不同优先级, P_1 、 P_2 、 P_3 、 P_4 分别设置为4、3、2、1, 调节参数设置为 $\alpha = \beta = 0.5$ 。此外, 考虑到各个场景对应着不同的仿真时长, 对于仿真时长较高的场景, 应该允许节点设备缓存更多的消息分组以及消息生存时间更加宽松, 因此, 对应的参数设置如表2所示。

表2 仿真参数设置

仿真参数	Pmtrs	Infocom06	Random
缓存空间/kb	1 000~2 000	200~300	30~70
消息大小/kb	1	1	1
消息TTL/min	500	300	50
初始能量/mAh	3 000~5 000	3 000~5 000	3 000~5 000
节点数	49个移动节点	78移动节点+ 20静止节点	40个移动节点

4.3 仿真结果讨论

基于ONE平台在不同场景、不同仿真时长的前提下比较MP_SSR、PROPHET与SPRAY_WAIT^[11]3种算法的路由性能。

4.3.1 交付率与加权交付率

图1~图6分别给出在Pmtrs、Infocom06以及Random场景下不同算法的消息交付率与加权消息交付率。从图中可以看出, MP_SSR算法的消息交付率与加权消息交付率均优于其他两种算法。这是由于MP_SSR在消息转发过程中对节点缓存中的消息根据优先级进行排序, 并优先转发预估交付概率高的消息; 同时, 合理分配节点的缓存, 避免因缓存不足导致重要消息的丢失, 从而在一定程度上提高了消息交付率以及节点的满意度。

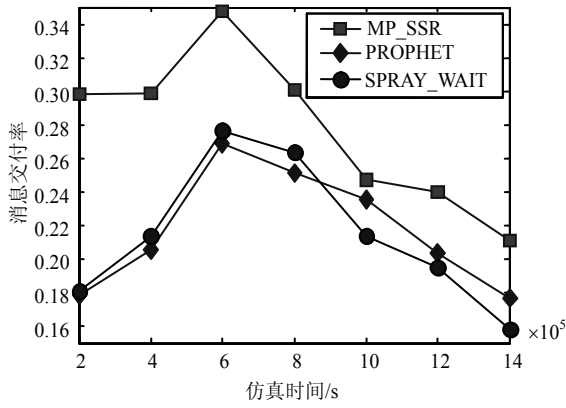


图1 Pmtrs场景下的消息交付率

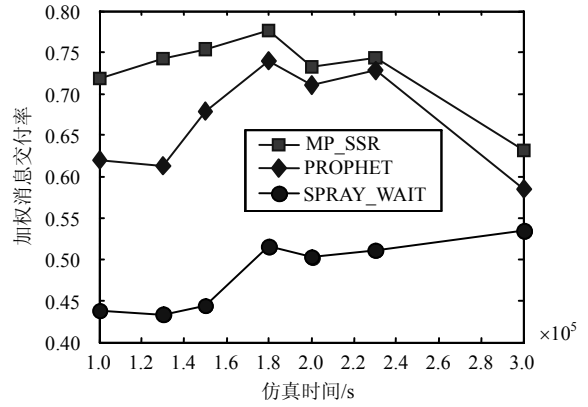


图5 Infocom06场景下的加权消息交付率

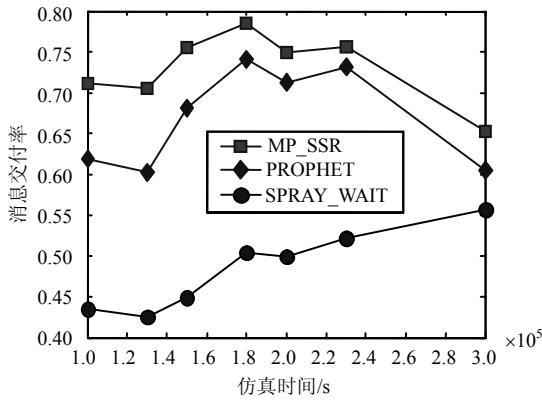


图2 Infocom06场景下的消息交付率

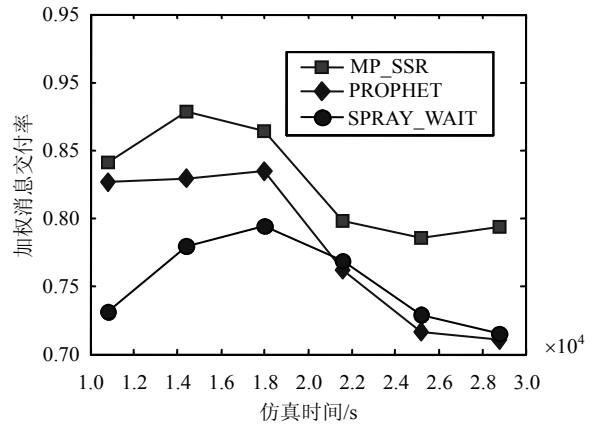


图6 Random场景下的加权消息交付率

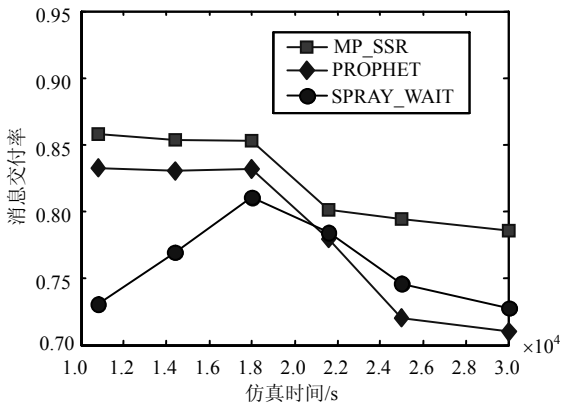


图3 Random场景下的消息交付率

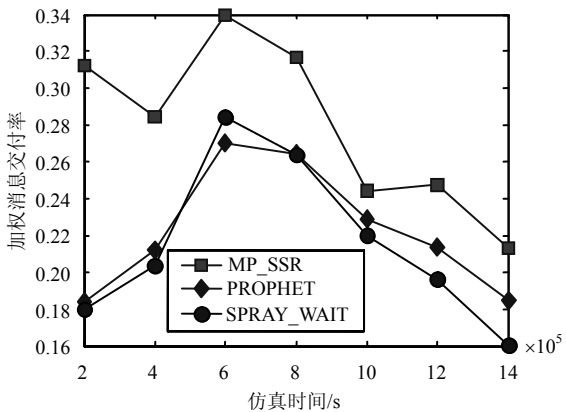


图4 Pmtrs场景下的加权消息交付率

4.3.2 网络开销

图7~图9分别给出在Pmtrs、Infocom06以及Random场景下的网络开销。

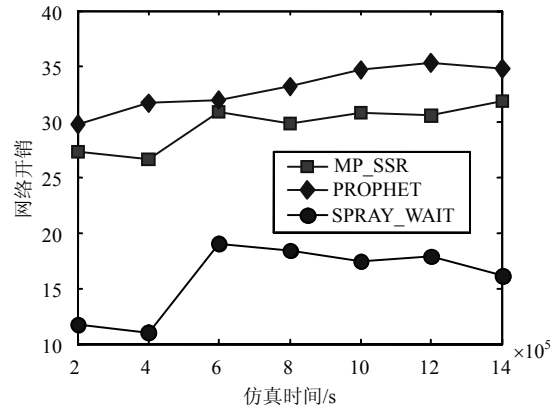


图7 Pmtrs场景下的网络开销

从图中可以看出, MP_SSR算法的网络开销始终处于居中位置, 并与PROPHET的开销在变化趋势上具有相似性。但由于前者考虑了节点的社会影响力与社会自私性, 在转发时根据节点的剩余能量来决定是否转发消息, 并根据消息的转发跳数采取不同的操作, 从而降低了网络的开销。此外,

SPRAY_WAIT是单副本路由算法,虽然网络开销一直保持在最低的水平,但相对于网络开销而言,消息交付率通常是评价算法优劣的更重要指标,因此,综合看来,MP_SSR具有更好的网络性能。

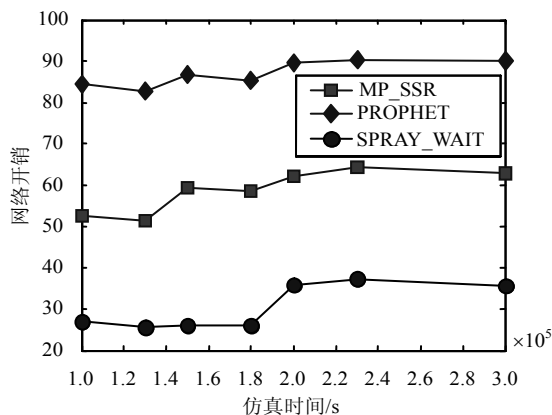


图8 Infocom06场景下的网络开销

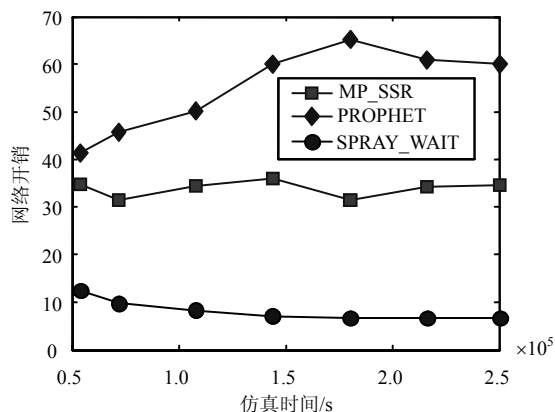


图9 Random场景下的网络开销

5 结束语

本文针对容迟网络中现有路由算法的不足,结合消息的优先级以及社会属性尤其是社会自私性两方面的特点,在合理分配节点资源的前提下,提出了新的路由算法,并分别在真实的移动场景以及随机场景下进行了仿真。仿真结果表明,该算法有着较好的网络性能,能够在提高消息交付率的同时保持较低的网络开销。

参 考 文 献

- [1] CHILIPIREA C, PETRE A, DOBRE C. Energy-aware social-based routing in opportunistic networks[C]// Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA). Barcelona, Spain: IEEE, 2013: 791-796.
- [2] VAHDAT A, BECKER D. Epidemic routing for partially connected ad hoc networks[R]. Durham, North Carolina: Duke University, 2000: 1-16.
- [3] LINDGREN A, DORIA A, SCHELEN O. Probabilistic routing in intermittently connected networks[J]. ACM SIGMOBILE Mobile Computing and Communications Review, 2003, 7(3): 19-20.
- [4] LI Q, GAO W, ZHU S, et al. A routing protocol for socially selfish delay tolerant networks[J]. Ad Hoc Networks, 2012, 10(8): 1619-1632.
- [5] LU Z, WEN Y, CAO G. Community detection in weighted networks: Algorithms and applications[C]//Proceedings of the International Conference on Pervasive Computing and Communications (PerCom). Sydney, Australia: IEEE, 2013: 179-184.
- [6] MIAO J, HASAN O, MOKHTR S, et al. An investigation on the unwillingness of nodes to participate in mobile delay tolerant network routing[J]. International Journal of Information Management, 2013, 33(2): 252-262.
- [7] WEI K, ZENG D, GUO S, et al. Social-aware relay node selection in delay tolerant networks[C]//Proceedings of the 22nd International Conference on Computer Communications and Networks (ICCCN). Nassau, Bahamas: IEEE, 2013: 1-7.
- [8] KERANEN A, OTT J, KARKKAINEN T. The one simulator for DTN protocol evaluation[C]//Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques. Arizona, USA: IEEE, 2009.
- [9] MERONI P, GAITTO S, PAGANI E, et al. Data set unimi/pmtr[DB/OL]. [2014-09-30]. <http://crawdad.cs.dartmouth.edu/unimi/pmtr>, Dec. 2008.
- [10] SCOTT J, GASS R, CROWCROFT J, HUI P, et al. DataSetCambridge/haggle/imote/infocom2006[DB/OL]. (2009-05-29). <http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/infocom2006>.
- [11] SPYROPOULOS T, PSOUNIS K, RAGHAVENDRA C. Spray and wait: an efficient routing scheme for intermittently connected mobile networks[C]//Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking. New York, USA: ACM, 2005: 252-259.

编辑 漆 蓉