



研究与开发

## 基于兴趣社区的高效路由与缓存管理算法

任智, 王坤龙, 李秀峰

(重庆邮电大学移动通信技术重庆市重点实验室, 重庆 400065)

**摘要:** 针对现有 BEEINFO 算法中存在控制消息冗余、未考虑节点多邻居消息转发和对缓存中消息管理不合理的问题, 提出了一种基于兴趣社区的高效路由与缓存管理算法——ERCMAON。该算法通过精简控制消息, 增加对节点多邻居情形的路由设计, 降低了系统开销和消息转发时延; 同时, 通过优化节点缓存管理机制, 降低了有用信息被删除的概率, 从而可以提高消息的投递成功率。通过与现有的 BEEINFO、Epidemic 和 ProPHET 算法进行仿真验证, 结果表明, 与 BEEINFO 算法相比, ERCMAON 投递成功率至少提高 2.0%, 数据投递开销和归一化控制开销分别降低至少 9.7% 和 1.7%, 同时消息传输时延至少降低 2.4%。

**关键词:** 机会网络; 社区; 高效; 缓存管理; 路由算法

**中图分类号:** TP393.4

**文献标识码:** A

**doi:** 10.11959/j.issn.1000-0801.2018160

## An efficient routing and cache management algorithm based on interest-community for opportunity networks

REN Zhi, WANG Kunlong, LI Xiufeng

Chongqing Key Lab of Mobile Communication Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

**Abstract:** Aiming at the problem of control message redundancy existing in BEEINFO algorithm, unconsidered node multi-neighbor message forwarding problem and unreasonable management of message in cache, an efficient routing and cache management algorithm named ERCMAON which based on community of interest was proposed. The message forwarding delay was reduced by streamlining control messages, increasing the route design for multiple-neighbor nodes. At the same time, by optimizing the node cache management mechanism, the probability of deleting useful information was reduced, which could improve the success rate of message delivery. Simulation results show that compared with the BEEINFO algorithm, the delivery success rate of ERCMAON algorithm increases by at least 2.0%, the data delivery overhead and the normalized control overhead reduces by at least 9.7% and 1.7% respectively. At the same time, the message transmission delay reduces at least 2.4%.

**Key words:** opportunity network, community, efficient, cache management, routing algorithm

收稿日期: 2017-11-23; 修回日期: 2018-04-16

基金项目: 国家自然科学基金资助项目 (No.61379159); 长江学者和创新团队发展计划基金资助项目 (No.IRT1299)

**Foundation Items:** The National Natural Science Foundation of China (No.61379159), Program for Changjiang Scholars and Innovative Research Team in University (No.IRT1299)



## 1 引言

机会网络的研究最初来源于早期的延迟容忍网络,在消息转发过程中,源节点和目的节点之间不一定存在完整的路径,但随着时间的推移,节点移动带来的相遇机会可以实现消息的转发,机会网络就是通过节点的相遇机会来实现网络节点之间的通信<sup>[1-2]</sup>。具有相同兴趣的人(如同学或同事)通常会聚在一起讨论并分享他们的兴趣,他们经常联系并组成一个社区。人所携带的智能设备(如手机)作为网络节点,因此也具有社会属性,通常用不同的标签标记以表示其类别。此外,不同兴趣节点会生成大量不同的数据,利用不同的数据类别代表节点的兴趣对节点社区进行划分。考虑节点之间社区关系的机会网络称为基于社区的机会网络,虚拟社区和社会连接是其中最常用的概念。同时,考虑节点的社区因素帮助消息路由和转发也是基于社区的机会网络区别于其他类型网络最重要的特征。

由于社区网络中节点需要经常与其他节点相互进行消息转发,对节点的能量消耗较大;同时,社区网络中为了提高数据的到达率,采用多副本的数据传输形式,导致对节点缓存占用较大。为节约节点缓存与能耗,本文将对社区网络中的冗余开销以及缓存管理进行优化,降低网络的系统能耗与开销,提高社区网络性能。

## 2 相关工作

基于社区的机会网络的路由算法是在机会网络的经典路由算法下发展起来的,主要有以下一些路由算法。

Epidemic<sup>[3]</sup>算法是采用泛洪的方式将消息转发给每个相遇节点,直至消息到达目的节点。该方式使网络中的消息副本数过多,增大了整个网络的开销。ProPHET<sup>[4]</sup>算法通过交换概要向量和预测概率矢量判断对方节点与目的节点的相遇概

率,携带消息的节点将消息转发给与目的节点相遇概率更高的相遇节点,大大地降低了网络中该消息的副本数。

Label<sup>[5]</sup>算法基于节点的兴趣属性将网络中的节点划分到不同社区,给不同兴趣的节点分配不同的label(标签),拥有相同label的节点属于同一社区。Bubble Rap<sup>[6]</sup>算法综合考虑了节点的中心度和社区结构来提高消息的传输效率,每个节点至少有一个label,相同label的节点属于同一个社区。消息总是向中心度高的节点转发,但消息汇聚在中心度高的节点的缓存中,有可能会造成消息的拥塞或者节点因能量消耗过快而死亡。STBID<sup>[7]</sup>算法提出一种节点间社会连接强度的计算机制,不同社会连接强度的节点分配具有不同的用于发送数据的令牌个数。SROR<sup>[8]</sup>算法和TDFSON<sup>[9]</sup>算法提供了节点社会关系的计算方法,用于在数据转发时选择最优的转发节点。PaSS<sup>[10]</sup>算法利用计算节点间的社会相似度以及位置相似度选择合适的下一跳转发节点实现数据传输。EEMF<sup>[11]</sup>算法综合考虑了节点的剩余能量与社会相似性进行数据转发。

BEEINFO 算法<sup>[12]</sup>是一种人工蜂群算法,它的设计来源于 SAN (socially-aware networking) 框架下的基于兴趣的转发机制。BEEINFO 算法利用移动节点的社会属性、移动规律和学习能力来探测动态环境的密度和社会连接度。此外,它通过个体兴趣对社区进行分类,兴趣信息虽然非常小,却足以预测社区密度和社区连接度,同时也能节约缓存和能耗等资源。

消息丢弃算法<sup>[13]</sup>主要包括头部丢弃策略、尾部丢弃策略和随机丢弃策略。即当缓冲区将要溢出时,从缓冲区的头部、尾部随机删除缓冲区的消息。缓存替换算法<sup>[14]</sup>描述了 BEEINFO 在缓存不足情况下的操作:在社区间转发阶段,要发送到源社区外的消息将会被替换。对于这些消息,根据源节点记录的不同目的社区的密度大小进一

步决定替换的顺序：低密度的消息会被优先替换，如果密度相同，则替换较老的消息。在社区内转发阶段，即源节点和目的节点在同一社区，则源节点根据目的节点的社会连接度决定替换顺序，目的社会连接度最低的消息最先被替换；如果社会连接度相同，则替换最老的消息。CABM<sup>[15]</sup>算法与缓存替代算法类似，同样对消息目的节点与本节点的社区紧密程度对缓存中的消息优先级进行排序。

通过对 BEEINFO 算法的深入研究，发现 BEEINFO 算法中存在控制消息冗余、未考虑节点多邻居消息转发问题和对缓存中消息管理不合理的问题。为解决以上问题，提出了一种面向机会网络基于兴趣社区的高效路由与缓存管理算法——ERCMAON (an efficient routing and cache management algorithm based on interest-community for opportunity network)，该算法通过精简以及合并控制消息减少了冗余的控制开销，同时增加对节点多邻居情形的路由设计，降低了两两通信带来的系统开销和消息转发时延；同时，基于消息的转发次数优化节点缓存管理机制，降低了有用信息被删除的概率，从而可以提高消息的投递成功率。并从理论分析和仿真验证了所提算法的性能。

### 3 系统模型与问题描述

#### 3.1 系统模型

**定义 1** (移动模型) 机会网络中的节点具有一定的社会性，其运动并不是完全随机的，经常相遇的节点之间有一个较高的接触概率，在未来相遇的可能性也较大。

**定义 2** (社区密度<sup>[12]</sup>) 表示节点到对应社区熟悉度。假设在时间  $T$  内，节点遇到的社区  $i$  内节点的次数为  $n$ ，则当前时刻  $t$  的社区密度计算式为式 (1)，而式 (2) 是对该社区未来密度的预测。

$$\text{Density}_i(t) = \sum_{t=0}^T n \quad (1)$$

$$\begin{aligned} &\text{Density}_i(t + \Delta t) \\ &= \alpha \times \text{Density}_i(t - \Delta t) + (1 - \alpha) \times \text{Density}_i(t) \end{aligned} \quad (2)$$

其中， $\alpha$  是社区密度的预测因子 (常量)，表明过去密度信息对现在密度信息的影响权重。社区的密度会随着时间衰减，计算式为：

$$\text{Density}_{\text{new}} = \text{Density}_{\text{old}} \times \gamma^k \quad (3)$$

其中， $\gamma$  是衰退指数 (常量)， $k$  是距离上次更新的时间间隔。式 (1) 用于在非同社区节点相遇时，更新到彼此社区的密度信息，式 (3) 用于距离上次更新  $k$  时间后，再次更新到对应社区的密度信息，式 (2) 用于预测未来  $\Delta t$  后到对应社区的密度信息。

**定义 3** (社会连接度<sup>[13]</sup>) 用来表示相同社区 (兴趣) 的节点之间社会关系的强弱。节点的社会连接度信息只能在社区内部发挥作用，社会连接度计算公式如式 (4) 所示，而式 (5) 是对未来社会连接度的预测。

$$\text{SoTie}_{i,j}(t) = (\lambda_{i,j} \times d_{i,j}(t)) / T \quad (4)$$

$$\begin{aligned} &\text{SoTie}_{i,j}(t + \Delta t) \\ &= \beta \times \text{SoTie}_{i,j}(t - \Delta t) + (1 - \beta) \times \text{SoTie}_{i,j}(t) \end{aligned} \quad (5)$$

其中， $\lambda_{i,j}$  是接触频率 (在每个时间窗口  $T$  内，节点  $i$  与  $j$  的相遇次数)， $d_{i,j}$  是接触时间 (在每个时间窗口  $T$  内节点  $i$  与  $j$  的接触总时间)， $\beta$  是预测因子 (常量)。社会连接度也会随着时间衰减，如式 (6) 所示。

$$\text{SoTie}_{\text{new}} = \text{SoTie}_{\text{old}} \times \gamma^k \quad (6)$$

式 (4) 用于在同社区节点相遇时，更新彼此的社会连接度信息，式 (6) 用于距离上次更新  $k$  时间后，再次更新与对应节点的社会连接度信息，式 (5) 用于预测未来  $\Delta t$  后与对应节点的社会连接度信息。

#### 3.2 问题描述

通过研究发现，现有以 BEEINFO 算法为代



表的基于兴趣划分的社区网络存在以下3个问题。

### (1) 控制消息存在冗余

BEEINFO 算法节点相遇后的交互过程如图 1 所示, 两个节点相遇, 相互交换社区信息  $I$  和消息摘要列表  $Lm$  后, 根据相遇节点是否同社区交换社区密度信息  $Density$  和社会连接度信息  $SocTie$ , 再根据对方节点的密度信息或社会连接度信息, 比较与对方节点消息列表中各消息目的地址的社会关系强度, 然后向对方节点提出请求 (REQ), 最后发送对方节点所请求的消息。研究发现该算法存在冗余的控制消息, 当前节点接收到对方节点的密度信息或社会连接度信息, 可以判断出对方将要请求的数据分组, 因此不用等待对方请求, 可以直接发送数据, 同时, 后发送  $I+Lm$  消息的节点可以将  $I+Lm$  消息和社区密度或社会连接度消息合并传输。

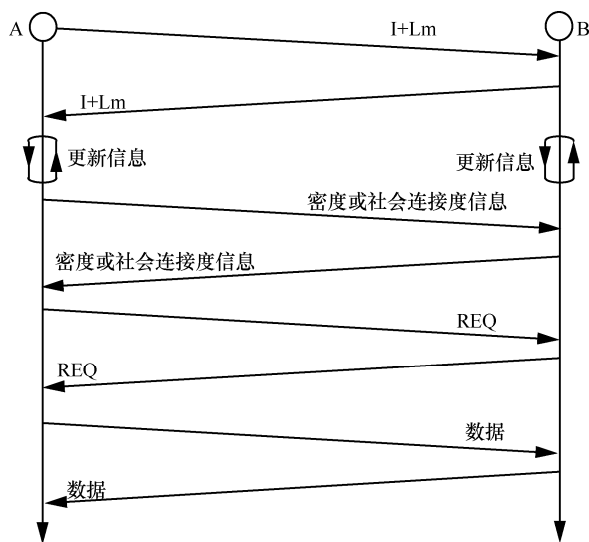


图1 节点相遇后的交互过程

### (2) 节点多邻居问题

BEEINFO 算法中只研究了两节点的相遇情况, 考虑实际情况中, 多个节点同时相遇的情形经常发生, 如图 2 所示。

如果简单地将两节点的消息转发过程应用到多节点相遇的消息转发过程中, 会带来不必要的冗余开销; 当多邻居彼此不在通信范围内, 但

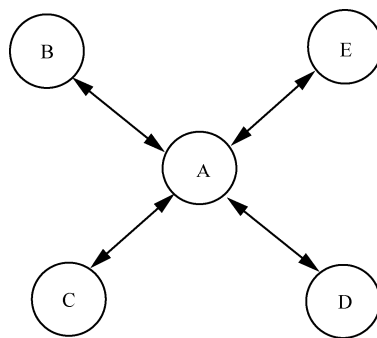


图2 节点多邻居情形

彼此都有消息需要对方转发时, 本节点可以充当中继节点加快消息的转发, 因此多邻居情形有必要考虑。

### (3) 缓存管理

原 BEEINFO 算法为了更高效地传输数据, 只考虑当前节点和目的节点的关系, 把缓存中的消息根据转发的优先级按从高到低的顺序排列, 当缓存不足时根据消息转发优先级相反的顺序删除缓存消息。但是该机制没有考虑网络中消息的副本个数, 只存在唯一副本的消息如果被删除则会影响网络性能, 被转发过的消息原则上其重要程度应当降低。

## 4 ECMAON

针对以上所述的 3 个问题, 本文提出了 ERCMAON, ERCMAON 通过增加节点多邻居的消息转发策略、精简冗余的控制消息和改进节点间的缓存管理机制, 提高消息转发效率。

### 4.1 ECMAON 新机制

#### (1) 精简控制消息

相遇节点在交互  $I+Lm$  列表消息时, 后发送  $I+Lm$  列表消息的节点在发送消息前, 先根据对方节点的社区信息  $I$  更新记录的到对方社区的密度信息 (update density, 相遇节点不同社区) 或本社区的社会连接度信息 (update SocTie, 相遇节点同社区), 计算双方节点与消息列表  $Lm$  中每个消息目的节点的社会连接度 (同社区时) 或到消息目的节点的社区密度 (不同社区时), 若本节点及对方节点都与消息的目的节点位于不同社区, 但

对方节点到目的节点的社区密度高于本节点,或者对方节点与消息的目的节点位于同社区而本节点与目的节点位于不同社区,或者本节点及对方节点都与消息的目的节点位于同社区,但对方节点到目的节点的社会连接度高于本节点,这3种情形均可以表明对方节点与本节点相比,有更高的概率与消息的目的节点相遇,因此将列表中对应的消息直接发送给对方节点,减少对方节点进行信息请求所带来的开销。然后根据对方节点的消息列表  $L_m$ , 从本节点消息列表中去掉对方列表中存在的消息,减少发送  $I+L_m$  列表的长度以降低开销,同时在发送本节点的  $I+L_m$  列表时,当  $I+L_m$  列表与本节点到其他社区的密度信息和到本社区其他节点的社会连接度信息总长度不超过数据分组中数据部分的最大长度时,将更新到其他社区的密度信息和到本社区其他节点的社会连接度信息加入消息后面一起发送,减少单独发送密度信息和社会连接度信息引起的冗余开销,改进后的交互过程如图3所示。

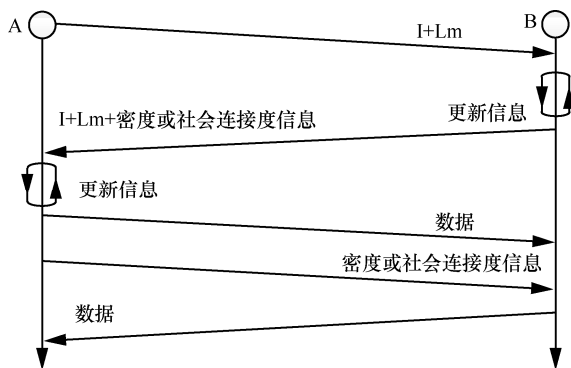


图3 节点相遇后新交互的过程

## (2) 增加多邻居消息转发

针对多节点相遇问题,不失一般性,以3个节点为例,在节点发现阶段,节点A接收到节点B和节点C的hello消息,得知B、C同时在节点A的通信范围内,此时节点A通过广播发送自身的  $I+L_m$  列表消息,节点B、C接收到节点A的  $I+L_m$  列表消息后,更新记录的到对方社区的密度信息或本社区的社会连接度信息,再向A发送本节点

的  $I+L_m$ +密度或社会连接度消息,节点A在收到节点B、C的  $I+L_m$ +密度或社会连接度消息后,计算节点B、C及本节点消息列表  $L_m$  中每个消息目的节点的社会连接度(同社区时)或消息目的节点的社区密度(不同社区时),通过与消息目的节点的社会关系以及社会连接度和社区密度信息判断与目的节点的相遇概率,若节点B、C与目的节点的相遇概率更高,则将列表中对应的消息直接发送给节点B或C。其次,节点A在接收到节点B、C的  $I+L_m$ +密度或社会连接度消息后,提取节点B、C到各个社区的社区密度及社会连接度,在通过广播发送自身密度或社会连接度消息时,密度或社会连接度消息中到每个社区的密度值及社会连接度取3个节点的最大值(若节点C中消息m的目的节点所在社区为5,节点A、B、C所在社区为1、2、3,并且节点A、B、C到社区5的密度分别为5、10、7,则节点A在发送自身密度或社会连接度消息时将到社区5的密度值写成10,此时节点C会将消息m转发给A,然后节点A再将消息m转发给节点B,加速了消息的转发)。节点B、C收到节点A的密度或社会连接度消息后,将需要转发的消息转发给节点A,节点A收到节点B、C转发的消息后,提取出需要转发的消息再转发给对应节点B和C。具体交互流程如图4所示。

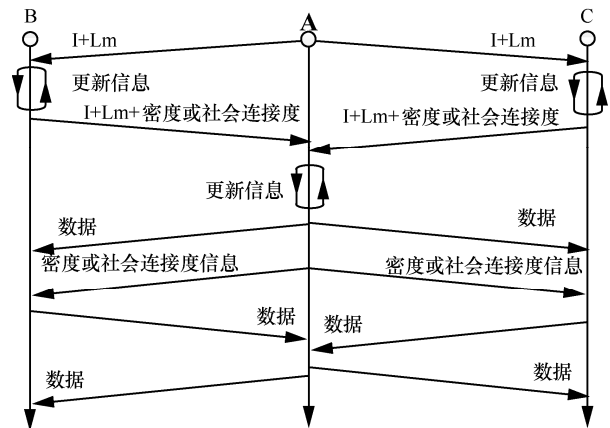


图4 多节点消息交互过程



### (3) 改进缓存管理机制

改进的缓存管理机制优先考虑消息在网络中的唯一性,即节点自身产生并且还未转发过的消息具有最高的优先级。然后考虑节点与消息目的节点的社会关系,社会关系越强其对应的消息的缓存优先级越高,最后考虑当前节点对消息的转发次数,转发次数越少,消息优先级越高,具体的消息转发优先级的顺序如下:

- 由节点自己产生且未转发出去的消息优先级最高;
- 消息的目的节点与当前节点同社区的消息,按当前节点与消息目的节点之间的社会连接度排序,对应社会连接度大的消息优先级高;
- 消息目的节点与当前节点不同社区的消息,按当前节点记录的到目的社区的密度大小排列,对应社区密度大的消息优先级高;
- 当前节点对自身产生的消息以及从其他节点接收到的消息,记录每个消息本节点转发过的次数,转发次数越少,消息优先级越高;
- 节点周期性地根据缓存列表中消息的变化情况调整列表顺序,当缓存不足时根据列表排序进行有序的删除操作。对应的缓存消息优先级示意如图 5 所示。

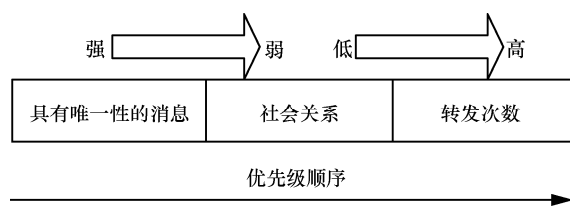


图 5 缓存消息优先级示意

## 4.2 ECMAON 的操作

ERCMAON 的操作过程如下。

**步骤 1** 节点 A 在网络中运动,当遇到其他节点并建立邻居关系后,根据邻居节点的不同个数,执行不同操作,若相遇节点为 1 个,则

向对方节点发送自身兴趣消息和携带消息的摘要信息,转向步骤 2;若相遇节点为多个,则向邻居节点广播自身兴趣消息和携带消息的摘要信息,转向步骤 3。

**步骤 2** 收到对方的兴趣和消息摘要信息后,更新所记录的社区密度信息和社会连接度信息,向对方发送本节点的  $I+Lm$ +密度或社会连接度消息,转向步骤 4。

**步骤 3** 收到广播的兴趣和消息摘要信息后,更新所记录的社区密度信息和社会连接度信息,向节点 A 发送本节点的  $I+Lm$ +密度或社会连接度消息,转向步骤 5。

**步骤 4** 收到对方的  $I+Lm$ +密度或社会连接度消息,计算双方节点与消息列表  $Lm$  中每个消息目的节点的社会连接度(同社区时)或消息目的节点的社区密度(不同社区),通过与消息目的节点的社区关系以及社会连接度和社区密度信息判断与目的节点的相遇概率,若对方节点与目的节点的相遇概率更高,则将列表中对应的消息直接发送给对方节点,然后向对方节点发送本节点的密度或社会连接度消息,转向步骤 6。

**步骤 5** 节点 A 计算邻居节点(假设有两个 B、C)及本节点与自身消息列表  $Lm$  中每个消息目的节点的社会连接度(同社区时)或消息目的节点的社区密度(不同社区),通过与消息目的节点的社区关系以及社会连接度和社区密度信息判断与目的节点的相遇概率,若节点 B、C 与目的节点的相遇概率更高,则将列表中对应的消息直接发送给节点 B 或者 C。其次,节点 A 在接收到节点 B、C 的  $I+Lm$ +密度或社会连接度消息后,提取节点 B、C 到各个社区的社区密度及社会连接度,在通过广播发送自身密度或社会连接度消息时,密度或社会连接度消息中到每个社区的密度值及社会连接度取 3 个节点的最大值,转向步骤 7。

**步骤 6** 收到节点 A 的  $I+Lm$ +密度或社会连

接度消息后, 执行与步骤 4 相同的操作, 转发对应消息。

**步骤 7** 节点 B、C 收到节点 A 的  $I+L_m$  密度或社会连接度消息后, 计算本节点及节点 A 与消息列表  $L_m$  中每个消息目的节点的社会连接度(同社区时)或消息目的节点的社区密度(不同社区), 通过与消息目的节点的社区关系以及社会连接度和社区密度信息判断与目的节点的相遇概率, 若节点 A 与目的节点的相遇概率更高, 则将列表中对应的消息直接发送给节点 S, 转向步骤 8。

**步骤 8** 节点 A 收到节点 B、C 转发的消息后, 提取出需要转发的消息再转发给对应节点 B 和 C。

每次操作前, 判断当前节点的缓存空间是否已满, 如果缓存已满, 则按照 ERCMAON 缓存管理机制处理。

该算法的伪代码描述如下, 其中  $N$  表示邻居节点个数,  $I_i$  表示节点  $i$  的社区属性,  $L_{mi}$  表示节点  $i$  所携带的消息列表,  $D_{i,d}$  和  $S_{i,d}$  表示节点  $i$  与目的节点的社区密度及社会连接度,  $M_{ij}$  表示节点  $i$  的第  $j$  个消息。

```

When node A encounters other nodes
if  $N=1$ 
    then A unicasts( $I_A+L_{m_A}$ ) to the encounter
    the encounter updates  $D_E$  and  $S_E$  after receiving ( $I_A+L_{m_A}$ )
    for( $i=1; i<L_{m_E}; i++$ )
        for( $j=1; j<L_{m_A}; j++$ )
            if( $M_{A,i} = M_{E,j}$ ) deletes it from the  $L_{m_E}$ ;
            then sends ( $I_E+L_{m_E}+D_E/S_E$ ) to A
            A updates  $D_A$  and  $S_A$  after receiving ( $I_E+L_{m_E}+D_E/S_E$ )
        for( $i=1; i<L_{m_A}; i++$ )
            for( $j=1; j<L_{m_E}; j++$ )
                if( $M_{A,i} = M_{E,j}$ ) deletes it from the  $L_{m_A}$ ;
                A calculates  $D_{E,d}$ ,  $S_{E,d}$ ,  $D_{A,d}$ ,  $S_{A,d}$ 
            if( $((I_A \neq ID \ \&\& \ IE = ID) \ || \ (I_A = ID \ \&\& \ IE \neq ID$ 

```

$\&\& \ S_{E,d} > S_{A,d})$

$\ || \ ((I_A \neq ID \ \&\& \ IE \neq ID \ \&\& \ D_{E,d} > D_{A,d}))$

A 把  $M_{A,i}$  发送给相遇者

then A 把( $D_A/S_A$ )发送给相遇者

相遇者运行 9~16 行的相同进程

else if  $N>1$

then A 把( $I_A+L_{m_A}$ )广播给所有相遇者

每个相遇者都运行 4~8 行的相同进程

A 运行 9~16 行的相同进程

A 将  $DA / SA$  重置为自身与其所有相遇者之间的最高值

then 把( $D_A/S_A$ )广播给所有相遇者

每个相遇者运行 9~16 行的相同进程

节点 A 在必要时中继它收到的消息

end if

## 5 ERCMAON 仿真验证及分析

本文利用 ONE 仿真平台对 ERCMAON 的相关性能进行仿真验证, 并与 BEEINFO、Epidemic 和 ProPHET 算法对比相关性能的优劣。

### 5.1 网络场景及参数设置

仿真采用的场景为 ONE 中提供的 default\_scenario, 区域面积为  $4\ 500\text{ m} \times 3\ 400\text{ m}$ , 并将该区域划分为 6 个社区, 社区中包含两类移动节点: 行人和公交车。并设置其对应的移动速度、通信半径和消息传输速率。以节点缓存大小为变量, 研究在节点缓存大小不同时下各算法的性能。仿真中设置的具体参数见表 1。

### 5.2 仿真结果及分析

#### 5.2.1 消息投递成功率

消息投递成功率指源节点的消息成功转发到目的节点的数目占有源节点产生的消息总量的百分比, 计算式为:

$$S = N_t / N_f \quad (7)$$

其中,  $N_t$  表示成功转发的数据分组的数目,  $N_f$  表示所有源节点生成的消息总量。



表 1 仿真参数设置

| 仿真参数                              | 数值                |
|-----------------------------------|-------------------|
| 场景面积                              | 4 500 m×3 400 m   |
| 仿真时间/h                            | 80                |
| 人节点个数                             | 80                |
| 公交车节点个数/个                         | 10                |
| 社区个数/个                            | 6                 |
| 人节点移动速/(m·s <sup>-1</sup> )       | [0.5,1.5]         |
| 公交车节点移动速/(m·s <sup>-1</sup> )     | [1.2,10]          |
| 人节点通信半径/m                         | 10                |
| 公交车节点通信半径/m                       | 1 000             |
| 消息大小/MB                           | [0.5, 1]          |
| 人消息传输速度/(Mbit·s <sup>-1</sup> )   | 2                 |
| 公交车消息传输速度/(Mbit·s <sup>-1</sup> ) | 10                |
| 消息生存时间/h                          | 10,15,20,25,30,35 |
| 消息间隔时间/s                          | [10, 35]          |
| 节点缓存/MB                           | 10                |
| 移动模型                              | 工作日运动模型           |
| $\alpha$                          | 0.3               |
| $\beta$                           | 0.3               |
| $\gamma$                          | 0.98              |

投递成功率对比如图 6 所示, 由图 6 可知, ERCMAON 比 BEEINFO 算法及另外两种算法的投递成功率至少高 2.0%, 其原因是: 节点交互过程中的冗余控制开销减少, 使得消息在相遇节点间的传输加快, 在缓存大小一定时, 消息的传送成功率会因此提高; 节点多邻居情形下, 中间节点可以作为中继节点协助其他不在通信范围内的邻居节点进行消息的转发, 可以提高消息的传送成功率; 优化的缓存管理机制保证网络中唯一副本的消息可以在网络优先转发, 不会因为与目的节点之间的社会连接强度低就被删除, 网络整体的投递成功率会提高; 同时社会关系强度代表着节点间的熟悉度和消息的可达率, 利用社会关系强度进行缓存排序, 有利于提高消息的可达率和整个网络的传递成功率。

### 5.2.2 数据投递开销

数据投递开销指将数据分组成功传输到目的节点所需消耗的通信开销 (为完成数据消息顺利传输到目的节点, 网络中需要传输的总比特数,

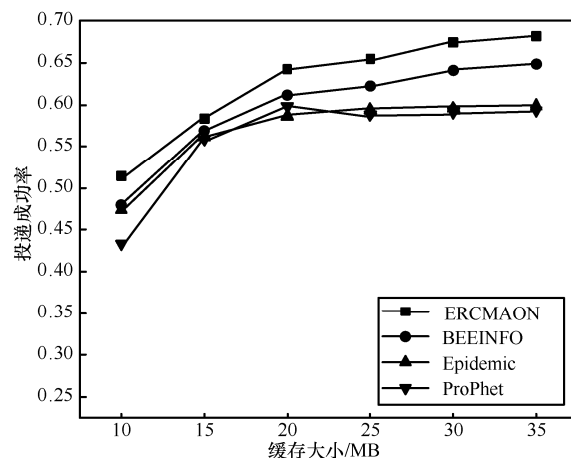


图 6 投递成功率对比

包括数据消息和控制消息 (即在数据转发前的其他进行交互的消息, 包括 hello 消息、社区属性与消息摘要列表消息、社区密度或社会连接度消息、请求消息) 的比特总数。

数据投递开销对比如图 7 所示, 由图 7 可知, ERCMAON 比 BEEINFO 及其他两种算法的数据投递开销至少低 9.7%, 其原因是: ERCMAON 减少了节点交互过程中的冗余控制开销; 在节点多邻居节点情形下, 利用广播发送控制消息减少了分别发送的控制开销; 被转发过的消息因已经转发到了更合适的节点, 所以该消息的投递成功率不会受到很大的影响, 把该类消息的缓存优先级降低之后, 当缓存不足时, 优先删除转发过的消息可以减少不必要的转发开销。

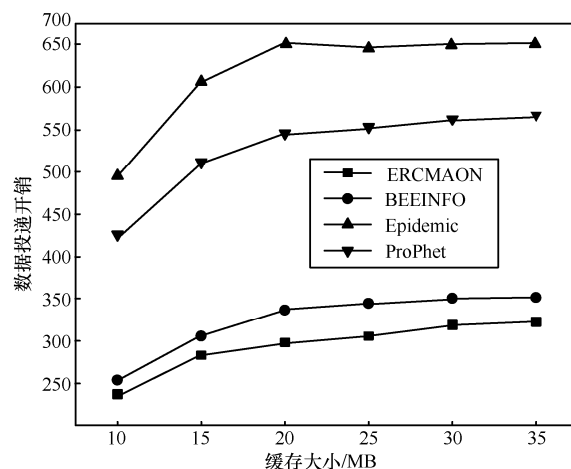


图 7 数据投递开销对比

### 5.2.3 归一化控制开销

归一化控制开销指网络中消息成功送达目的节点过程中, 控制消息的总比特数占整个网络中传输的总比特数(即控制消息与数据消息比特数总和)的比例。控制开销可以衡量所提算法的效率, 可以反映出将数据分组成功传输到目的节点所需消耗的通信开销占比, 其计算式如下:

$$C = N_c / (N_c + N_d) \quad (8)$$

其中,  $N_c$  表示所有节点产生的控制分组开销总量,  $N_d$  表示被目的节点接收到的数据分组开销总量。

归一化控制开销对比如图 8 所示, 由图 8 可知, ERCMAON 比 BEEINFO 至少低 1.7% 的归一化控制开销。其原因是: ERCMAON 算法减少了节点交互过程中的冗余控制开销, 使得消息的总体控制开销降低, 由于在 ERCMAON 中消息的成功率增加, 所以归一化控制开销也得到降低; 在节点多邻居节点情形下, 利用广播发送控制消息减少了分别发送的控制开销, 同时中间节点作为中继节点时可以加快消息的转发速率和投递成功率, 因此对应的归一化控制开销得到降低; 采用 ERCMAON 的节点对缓存中的消息重新排序, 转发过的节点的重要程度降低, 转发过的消息优先被删除, 这样既提高了消息的整体投递率, 又减少了传送到目的社区的消息在目的社区外的传递。

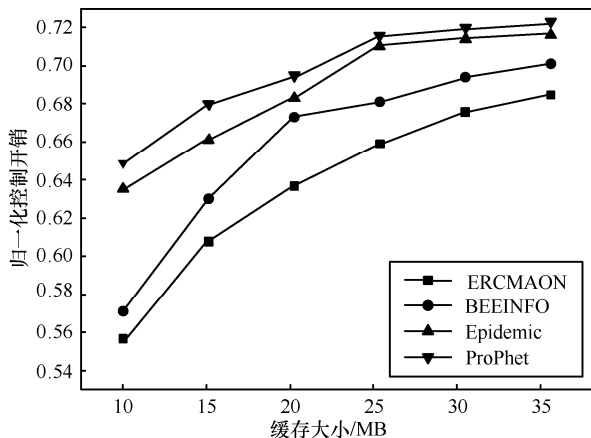


图 8 归一化控制开销对比

### 5.2.4 传输时延

传输时延反映成功传输一个数据分组到目的节点所需要的平均时间, 其计算式如下:

$$\bar{T} = \sum_{i=1} T_i / N \quad (9)$$

其中,  $N$  表示目的节点接收到的数据分组的数目,  $T_i$  表示数据分组  $i$  的时延。

消息传输时延对比如图 9 所示, 由图 9 可以看出, ERCMAON、BEEINFO 算法的传输时延比 Epidemic 和 ProPHET 算法的传输时延高, 其主要原因是 Epidemic 和 ProPHET 算法对消息转发的限制更少, 网络中消息副本更多, 携带消息的节点在更短时间内成功投递消息的可能性更高。ERCMAON 的传输时延比 BEEINFO 算法至少要低 2.4%, 其原因是: ERCMAON 减少了节点交互过程中的冗余控制开销, 消息在相遇节点间传输得更快; 在节点多邻居情形下, 消息经中间节点中继时可以明显降低消息的传输时延。

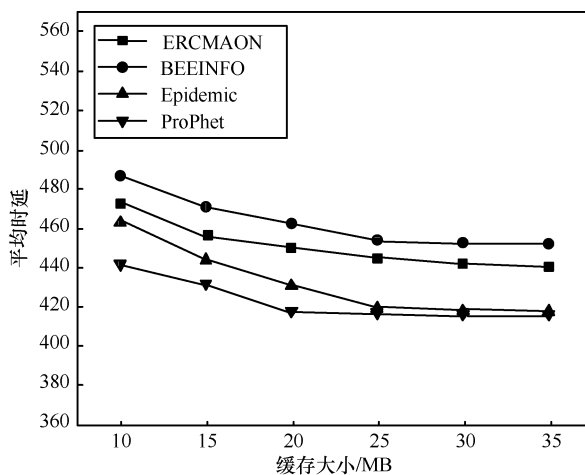


图 9 消息传输时延对比

## 6 结束语

本文提出了一种基于兴趣社区的高效路由与缓存管理路由算法——ERCMAON, 该算法通过合并控制消息、优化缓存管理以及节点多邻居情形的路由设计, 降低了系统开销并增强了消息传输



成功率。但是, 目前的研究是假设节点入网前已分配有唯一且固定的兴趣(社区), 对于社区的划分并未详细探究, 未来将针对节点的社区划分进行研究, 同时设计单一节点同属多个社区情形的路由机制。

## 参考文献:

- [1] STAVROULAKI V, TSAGKARIS K, LOGOTHETIS M, et al. Opportunistic networks[J]. IEEE Vehicular Technology Magazine, 2011, 6(3): 52-59.
- [2] HOSSMANN T, NOMIKOS G, SPYROPOULOS T, et al. Collection and analysis of multi-dimensional network data for opportunistic networking research[J]. Computer Communications, 2012, 35(13): 1613-1625.
- [3] VAHDAT A, BECKER D. Epidemic routing for partially-connected Ad Hoc networks[J]. Master Thesis, 2000.
- [4] LINDGREN A, DORIA A. Probabilistic routing in intermittently connected networks[J]. ACM Sigmobile Mobile Computing & Communications Review, 2004, 7(3): 239-254.
- [5] MEI A, MORABITO G, SANTI P, et al. Social-aware stateless forwarding in pocket switched networks[C]//2011 IEEE INFOCOM, April 10-15, 2011, Shanghai, China. Piscataway: IEEE Press, 2011: 251-255.
- [6] PAN H, CROWCROFT J, EIKO Y. BUBBLE rap: social-based forwarding in delay-tolerant networks[J]. IEEE Transactions on Mobile Computing, 2010, 10(11): 1576-1589.
- [7] WANG Y, WU J. Social-tie-based information dissemination in mobile opportunistic social networks[C]//2013 IEEE 14th International Symposium and Workshops on aWorld of Wireless, Mobile and Multimedia Networks (WoWMoM), June 4-7, 2013, Madrid, Spain. Piscataway: IEEE Press, 2013: 1-6.
- [8] WONG G, JIA X. A novel socially-aware opportunistic routing algorithm in mobile social networks[C]//International Conference on Computing, Networking and Communications, Jan 28-31, 2013, San Diego, USA. Piscataway: IEEE Press, 2013: 514-518.
- [9] BECKER C, SCHLINGA S, FISCHER S. Trustful data forwarding in social opportunistic networks[C]//Ubiquitous Intelligence and Computing, Dec 18-21, 2013, Vietri, Italy. Piscataway: IEEE Press, 2013: 430-437.
- [10] JANG K, LEE J, KIM S K, et al. An adaptive routing algorithm considering position and social similarities in an opportunistic network[J]. Wireless Networks, 2016, 22(5): 1537-1551.
- [11] ZHANG S, WANG X, YAO M, et al. Community-based message transmission with energy efficient in opportunistic networks[C]//International Conference on Web Information Systems Engineering, November 16-18, Zhangjiajie, China. Berlin: Springer Press, 2016: 411-423.
- [12] LI J, LIU L, XIA F. BEEINFO: data forwarding based on interest and swarm intelligence for socially-aware networking[C]//International Conference on Mobile Computing & Networking, September 30-October 4, 2013, Miami, USA. New York: ACM Press, 2013: 175-178.
- [13] ERRAMILI V, CROVELLA M. Forwarding in opportunistic networks with resource constraints[C]//The 3rd ACM Workshop on Challenged Networks, September 15-15, 2008, San Francisco, USA. New York: ACM Press, 2008: 41-48.
- [14] XIA F, LIU L, LI J, et al. BEEINFO: interest-based forwarding using artificial bee colony for socially aware networking[J]. IEEE Transactions on Vehicular Technology, 2015, 64(3): 1188-1200.
- [15] ZHOU J, LIN Y, ZHOU S, et al. Community-based adaptive buffer management strategy in opportunistic network[C]//International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, November 16-18, 2016, Zhangjiajie, China. Berlin: Springer Press, 2016: 16-25.

## [作者简介]



任智(1971-), 男, 博士后, 重庆邮电大学移动通信技术重庆市重点实验室教授、博士生导师, 主要研究方向为宽带自组网与无线通信。



王坤龙(1993-), 男, 重庆邮电大学移动通信技术重庆市重点实验室硕士生, 主要研究方向为物联网理论技术及无线 MAC 协议。



李秀峰(1991-), 男, 重庆邮电大学移动通信技术重庆市重点实验室硕士生, 主要研究方向为机会网络路由算法。