

基于动态半马尔可夫路径搜索模型的分簇路由方法

王恩¹⁾, 杨永健¹⁾, 李莅²⁾

¹⁾(吉林大学 计算机科学与技术学院, 长春市 中国 130012)

²⁾(吉林大学 软件学院, 长春市 中国 130012)

摘要 在容迟网络环境下, 提出一种基于动态半马尔可夫路径搜索模型的分簇路由方法 CRSMP (clustering routing method based on semi-Markov process and path-finding strategy), 该方法既考虑节点拥有的社会属性所导致的分簇问题, 又考虑到节点间未来一段时间内的最大相遇概率以及对应的相遇时间, 结合分簇结果和相遇情况生成动态路由表, 完成一种单副本的路由方法. 首先依据节点间路径的相似程度进行分簇, 然后运用半马尔可夫模型预测节点间未来某一时刻的相遇概率, 依据源节点和目的节点所在的分簇确定可以应用到路由中的节点集合, 最后根据路径搜索策略找到最优路径, 生成与当前时刻有关的动态路由表. 仿真结果表明 CRSMP 在缓存较小的情况下投递成功率远高于 DirectDeliveryRouter, FirstContactRouter 和 SimBetRouter 三种单副本路由方式以及 Spray and Wait, Epidemic 和 Prophet 三种多副本路由协议. 在 10M 缓存下的 CRSMP 有着与 500M 缓存下的 Epidemic 相近的路由性能. 进一步在真实数据集上进行测试, 测试结果表明 CRSMP 算法依然有着较好的路由性能.

关键词 容迟网络; 半马尔可夫; 分簇; 动态路由表; 路径相似度; 路径搜索
中图法分类号 TP393

A Clustering Routing Method Based on Semi-Markov Process and Path-finding Strategy in DTN

WANG En¹⁾, YANG Yong-Jian¹⁾, LI Li²⁾

¹⁾(Department of Computer Science and Technology, Jilin University, 130012, China)

²⁾(Department of Software Engineering, Jilin University, 130012, China)

Abstract In delay tolerant network, we have proposed a clustering routing method based on semi-Markov process and path-finding strategy (CRSMP), which considers not only the nodes' social properties but also the time and maximum probability of node contacts. This method firstly clusters nodes according to path similarity, then predicts node contact probabilities at some point in the future by applying semi-Markov process and determines a collection of nodes used in the routing on the basis of cluster or clusters containing the source node and the destination node. Lastly, dynamic routing tables related to the current time are obtained by executing path-finding algorithm. Simulation results show that the delivery rate of CRSMP is much higher than DirectDeliveryRouter, FirstContactRouter and SimBetRouter which are also single-copy routing schemes under the condition that the size of caches is small, and the delivery rate of CRSMP is also higher than Spray and Wait,

本课题得到国家自然科学基金(No. 61272412); 吉林省科技发展计划项目-重点项目(No. 20120303)资助. 王恩, 男, 1987年生, 博士研究生, 主要研究领域为容迟网络、社交网络, E-mail: wangen0310@126.com. 杨永健, 男, 1960年生, 博士, 教授, 博士生导师, 主要研究领域为无线通信与网络智能, E-mail: yyj@jlu.edu.cn. 李莅, 男, 1988年生, 硕士研究生, 主要研究领域为容迟网络、无线传感器网络, E-mail: li0755@mails.jlu.edu.cn.

Epidemic and Prophet which are multi-copy routing schemes, and CRSMP with 10M caches is similar to Epidemic with 500M caches in routing performance. Furthermore, experiments with the real data set are done, and CRSMP still keeps better routing performance.

Key words delay tolerant network; semi-Markov process; clustering; dynamic routing table; path similarity; path-finding

0 引言

容迟网络 (Delay-Tolerant Network, DTN)^[1-2] 是一种由于端到端缺乏稳定链路, 导致传输时延较高, 甚至绝大部分时间处于中断状态的一类网络环境, 其早期的应用场景包括: 星际网络^[3-4], 战地网络^[5], 乡村网络^[6], 野生动物监控与追踪网络^[7], PSN (pocket switched network) 网络^[8-9] 等. 2003 年 FALL 在国际会议 SIGCOMM03 上提出这一概念, 之后如何在这种拓扑结构剧烈变化的网络环境下寻求高效的 routing 方法成为该领域研究的热点问题.

近年来, 随着无线通信技术日趋成熟, 通信设备的体积不断缩小, 一些人携带着无线通信设备形成了基于区域的无线网络环境, 在这种网络环境下, 节点的移动方式主要包括区域内的驻留和区域间的行走, 节点所在的区域编号代表了节点所处的状态, 节点进入到下一个区域代表了状态的改变, 节点状态的改变并不是完全随机的, 下一个状态与当前状态存在着某种关系, 而校园网络正是上述网络环境的典型例子. 本文的研究思路来源于以下两个现象:

(1) 大学校园中同学和老师的行为并不是完全随机的, 他们每天在一些区域 (寝室, 食堂等) 会驻留较长时间.

(2) 同学和老师每天有着几乎固定的行动路线, 比如本科生每天大致遵循: 寝室→食堂→教室→食堂→教室→食堂→寝室的运动规律.

出现以上两个现象的主要原因是社会性驱使了节点的运动行为, 致使节点的运动存在周期性, 在这种网络环境下节点间的通信不是单纯地依靠随机相遇来完成, 而是与彼此的社会行为状态产生了密不可分的联系. 校园网络中往往体现出经典的“小世界”现象^[10], 即节点间可以依据其社会属性通过一跳或几跳与其他节点产生联系, 社会属性相同或者相近的节点间会有更多的通信机会, 于是挖掘出网络中有相似运动规律的节点簇必然会对该网络环境下的 routing 问题带来帮助.

传统的 DTN routing 算法按照网络中的消息副本数量分为单拷贝和多拷贝两种类型^[11]. 单拷贝路由方法在网络中只保留特定消息的唯一副本, 这样的路由方式往往由于网络连接中断或者路由选择策略不准确而导致传输成功率不高, 然而单副本路由方式节省网络开销, 能够避免拥塞等网络问题, 尤其是在节点数量庞大的网络环境下多副本的路由协议很难应用, 经典的路由方法包括 DirectDelivery^[12] 和 FirstContactRouter. 多拷贝路由方式又分为基于复制和基于概率两种^[13], 基于复制的路由方法主要是通过增加网络中的副本数量来提高投递成功率, 但是这种路由方法消耗资源较大, 并且极易发生网络拥塞等问题, 经典的路由方法包括 Spray and Wait^[14] 和 Epidemic^[15]. 基于概率的路由方法^[16] 主要是考虑节点间的上下文信息, 依据以往统计的历史信息来预测未来节点的行为方式以及节点间的相遇情况, 进而服务于路由决策. 基于概率的路由方法有 Prophet^[17] 和 MV^[18], Prophet 利用节点间相遇的历史信息和传递性来决定下一跳的选择, 将报文复制给传输可预测性更高的节点. MV 算法则利用节点间的相遇概率来描述报文传输成功的概率, 节点仅向传输概率更高的节点复制报文. 基于概率的路由方法不同于 Epidemic 路由策略, 它并非向所有相遇的节点复制报文, 而是有选择性地复制, 但由于基于概率的路由方法依然通过复制的方式进行报文感染, 因此属于多副本路由策略.

随着容迟网络中路由技术的不断成熟, 研究人员开始关注于节点间的相互关系, 彼此社会属性的相似程度等因素. 文献[19]中基于“小世界”理论, 提出了一种以相似性 (Sim) 与中介中心性 (Bet) 相结合的 SimBet 单副本路由方法, 该方法用节点间的共有邻居数代表彼此的相似性, 将自身对其他节点间连接的帮助程度定义为自己的中介中心性, 既考虑了节点社会属性的影响, 又考虑到以往的连接情况, 路由性能较好. Qinghua Li 等人在文献[20]中提出了一种考虑节点社会自私性的容迟网络路由算法 SSAR, 该方法综合考虑节点间的传输意愿以及相遇可能性, 运用机器学习方法衡量节点的传

输能力, 在节点保持自私性的情况下依然能有很好的路由性能. 文献[21]中, 通过将移动规律相近的节点聚合成最近社交圈, 提出了一种分为簇外喷射、簇间转发和簇内传染三个阶段的社交时延网络路由协议. Tamer Abdelkade 等人在文献[22]中提出了一种 DTN 环境下的基于社交分组的路由算法 SGBR, 该方法通过利用节点所在的社会群体来减少报文的冗余复制, 进而提高路由性能. 在文献[23]中周瑞涛, 曹元大等人通过对网络节点历史运动轨迹点聚类建立其热点活动区域, 把热点区域重叠度较高的节点归为同一社区, 在源节点和目的节点的社区中以洪泛的方式加快消息的扩散和传递速度, 针对不同社区准确地选择中继节点. 文献[24]中讨论了容迟网络的路由算法向社交网络接近的必然趋势, 提出了依据网络拓扑图对路由协议进行分类的想法. 文献[25]中于海征等人利用社会网络中节点间的权值计算方法, 计算出团队间的关系强度矩阵, 消息源节点的团队依据关系强度矩阵选择合适节点作为中继向目的节点传递消息, 考虑到了自私节点对传递的影响, 进而提出了基于社会网络的可靠路由方法. 文献[26]中张振京等人提出了一种新型的混合路由算法 CS-DTN, 基于节点间相遇概率的预测值进行分簇, 簇内依据二分喷射的方式寻找中继节点, 簇间依据节点的运动情况进行预测, 通过更接近目的节点的中继节点路由报文.

综上所述容迟网络中现有的路由方法或者关注依据社会属性的节点分簇问题, 或者关注于节点间相遇可能性问题, 或者关注于节点下一跳的转发选择问题. 本文认为已有的路由方法存在如下缺陷: (1) 已有的分簇方法所依据的分簇条件(节点间相遇次数, 通信时长)变动较大, 导致簇成员变化频繁, 维护分簇的开销较高. (2) 只能预测出未来节点间的相遇可能性, 而不能确定未来某一时刻节点间的相遇概率, 就无法判断节点间最有可能在哪一时刻发生相遇, 导致对未来相遇的估计不够准确. (3) 节点的路由只能决定下一跳的选择, 而不能指导多跳路由.

本文的主要贡献是: (1) 校园网络中人的身份一般不会变化, 人的社会角色决定了其每天运动路径相对固定, 因此提出了基于节点间路径相似度进行分簇的方法, 克服了以往分簇频繁变化, 维护开销大的问题. (2) 借鉴 Yuan^[27]等人描述的节点移动模型, 在其基础上进行改进, 运用半马尔可夫模型预测出节点间未来任意时刻的相遇概率, 进而估计

出节点间最有可能发生相遇的时间点, 这样更有利于报文的路线决策. 例如, 以往的路由策略只能确定两个节点间未来的相遇概率为 0.8, 而这样的相遇可能发生在很久以后, 依据这个概率值进行路由会带来很大的网络时延, 而本文提出的半马尔可夫预测模型很好地避免了这个问题. (3) 克服了一些路由算法只能指导一跳的局限性, 运用搜索策略找到一条到目的节点的投递路径, 生成与时间相关的动态路由表, 尽量多跳地指导报文的投递行为.

实验结果表明本文提出的基于动态半马尔可夫路径搜索模型的 DTN 分簇路由方法 CRSMP (clustering routing method based on semi-Markov process and path-finding strategy) 显著提高了投递成功率, 并且在一定程度上降低了网络平均时延和负载比率.

1 分簇方法

1.1 节点运动轨迹模型

本文在仿真实验中模拟了校园的 10 个区域, 分别为寝室, 食堂, 计算机学院, 数学学院, 自习室 1, 自习室 2, 图书馆, 校外, 体育馆, 银行, 在图 1 中用曲线围成的封闭区域表示, 编号依次为 0-9, 区域内的黑色实心圆圈表示运动的人群, 箭头表示人行走的方向. 本文同时在图 1 的基础上设计了一种尽量模拟校园中学生和老师行为的运动模型, 该运动模型的具体定义如下: 每个节点从图 1 的一个区域内诞生, 诞生时节点就携带了自己的路径信息(区域编号序列), 在诞生区域内随机选择一个目的地, 通过最短路径行走模型移动到该地点, 然后随机选择下一个目的地, 重复 n 次这样的行为, n 的取值取决于区域编号. 当 n 次选择之后, 节点选择向区域编号序列中的下一个区域行走, 到达下一个区域后同样依据该区域的 n 值来重复区域内的行为, 当这个节点走完所有指定路径上的区域时, 回到诞生地循环该运动模型. 在每个区域内对应的 n 值参见实验部分的表 2. 这种运动模型主要仿真校园中学生和老师的两个行为习惯: (1) 校园中学生和老师每天运动的路径都很有规律, 并且没有特殊事件不会更改这种运动行为. (2) 老师和学生到达每个区域后都会停留一段时间, 而往往具有同一种社会属性的人群在相同的区域内停留的时间也相差不大. 因此本文设计的运动模型比较符合真实的校园行为.

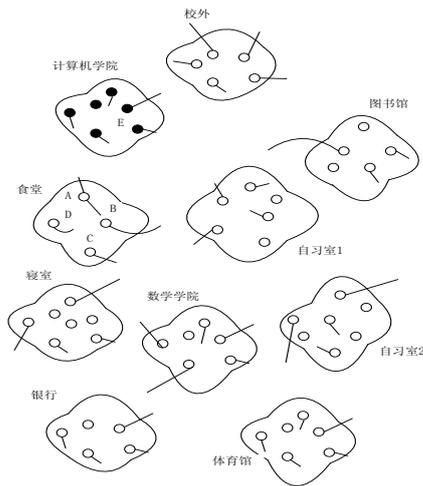


图1 校园网络环境

1.2 节点间路径相似度的计算模型

定义 1. 节点间路径相似度. $G=(V, E)$ 为网络拓扑图, 其中 V 为网络中的节点集合, E 为定义在 G 上的边集. 节点 a 和 b 间的路径相似度定义为边的权值大小, 记为 $W(e_{a,b})$, $W(e_{a,b})$ 的计算见定义 3, 其中节点 $a, b \in V$, $e_{a,b} \in E$ 表示节点 a 和 b 之间的边.

在定义 1 中计算节点间的边权值时需要每个节点知道网络中其他节点的运动轨迹信息, 本文在热启动阶段用如下方法统计轨迹信息:

每个节点本地顺序保存 N 个链表, 其中 N 为网络中的节点总数, 每个链表的长度不固定, 第 i 行链表对应的是 i 号节点的运动轨迹, 节点本地链表的更新包括两个部分:

(1) 当 i 号节点进入地图中设置的一个编号为 n 的可通信区域内, 则将 n 加入到第 i 行链表的末尾.

(2) 当两个节点在通信区域内相遇时, 执行如下更新规则: 依次比较彼此的 N 行链表, 用较长的链表更新较短的链表, 没有表项的链表的长度定义为 0.

通过以上方法, 经过一定量的模拟时间, 整个网络中的每个节点本地保存的 N 个链表能反映网络中所有节点这段时间内的运动轨迹信息, 其中每个节点的第 i 行链表存储第 i 号节点的移动路径.

定义 2. 最长公共子序列. 字符串 A 和 B 的最长公共子序列 L 满足: 对于 A 和 B 的任意公共子序列 K , 都有 $|K| \leq |L|$, 其中字符串 A 和 B 分别为节点 a 和 b 的运动轨迹.

在定义 2 中, 例如 $A="1327468315"$, $B="1275478312"$, 如图 2 所示, 最长公共子序列

$L="1274831"$. 从图 2 中分析得出公共子序列都是严格有序的, 但是最长公共子序列未必唯一, 两个字符串可以有多个长度相同的最长公共子序列. 文中采用 LCS (算法 1) [28] 来计算最长公共子序列的长度, 进而计算路径的相似程度.

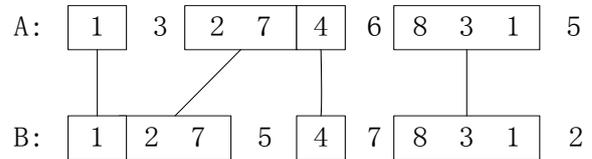


图2 公共子序列图

算法 1. LCS.

```
int LENGH_LCS(char* A, char* B)
{ int C[P,Q];
  FOR(i=P; i>=0; i--)
    FOR(j=Q; j>=0; j--)
      { IF (A[i] == '\0' & B[j] == '\0') C[i,j] = 0;
        ELSE IF (A[i] == B[j]) C[i,j] = C[i+1, j+1]+1;
        ELSE C[i,j] = max(C[i+1, j], C[i, j+1]);
      }
  RETURN C[0,0];
}
```

LCS 算法的时间复杂度为 $O(PQ)$, 其中 P 为字符串 A 的长度, Q 为字符串 B 的长度. 在计算数组 $C[i,j]$ 时, 只用到数组 C 的第 i 行和第 $i+1$ 行. 因此, 只要用两行的数组空间就可以计算出最长公共子序列的长度. 进一步分析可将空间需求减至 $\min(P, Q)$. 该算法采用动态规划的思想, 自底向上递归求解最优值, 即最长公共子序列长度. 用初值为 0 的 $C[P, Q]$ 数组记录整个计算过程中的最优值矩阵. 当 $P=0$ 或 $Q=0$ 时, 空序列是 A 和 B 的最长公共子序列, 故 $C[P, Q]=0$. 经 LCS 算法得到的二维数组中左上角的 $C[0,0]$ 即为最长公共子序列的长度. 例如, $A="1327468315"$, $B="1275478312"$ 时, 经 LCS 计算得到的二维数组如图 3, $C[0,0]=7$, 即 A 和 B 的最长公共子序列的长度为 7.

	1	3	2	7	4	6	8	3	1	5
1	7	6	6	5	4	3	3	2	2	1
2	6	6	6	5	4	3	3	2	1	1
7	5	5	5	5	4	3	3	2	1	1
5	4	4	4	4	4	3	3	2	1	1
4	4	4	4	4	4	3	3	2	1	0
7	4	4	4	4	3	3	3	2	1	0
8	3	3	3	3	3	3	3	2	1	0
3	2	2	2	2	2	2	2	2	1	0
1	2	1	1	1	1	1	1	1	1	0
2	1	1	1	0	0	0	0	0	0	0

图3 LCS 算法的二维数组

定义 3. 字符串相似度. 字符串 A 和 B 的相似度

W 定义为 $W = \frac{2L}{P+Q}$ ，其中 L 为通过 LCS 算法求的最长公共子序列的长度， P 和 Q 分别为 A 和 B 的字符串长度。

1.3 基于节点间路径相似度的分簇模型

定义 4. 集合间的路径相似度. $g_1 \subset V, g_2 \subset V$ 为网络中节点集合 V 的两个子集, $g_1 \cap g_2 = \emptyset$, 集合间的路径相似度 $W(g_1, g_2)$ 定义为

$$W(g_1, g_2) = \frac{2 \sum_{e_{ab} \in E(U)} W(e_{ab})}{n(n-1)}, \text{ 其中 } U = g_1 \cup g_2, |U| = n, E(U) = \{e_{ab} | a \in U, b \in U\} \text{ 表示 } U \text{ 中的所有边集.}$$

定义 5. 网络平均路径相似度.网络平均路径相似度 $W(V)$ 定义为 $W(V) = \frac{2 \sum_{e_{ab} \in E(V)} W(e_{ab})}{n(n-1)}$.

基于节点间路径相似度的分簇模型首先对整个网络的带权拓扑图 (图 4) 依据边的权值 (路径相似度) 进行算法 2 的变换, 生成一棵与拓扑图相对应的树结构 (图 5), 这棵树结构的所有非叶子节点都是集合, 这些集合代表了找到的具有较高路径相似度的节点簇。

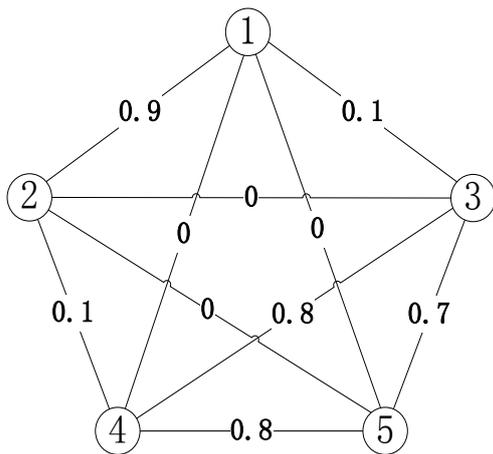


图 4 网络带权拓扑图

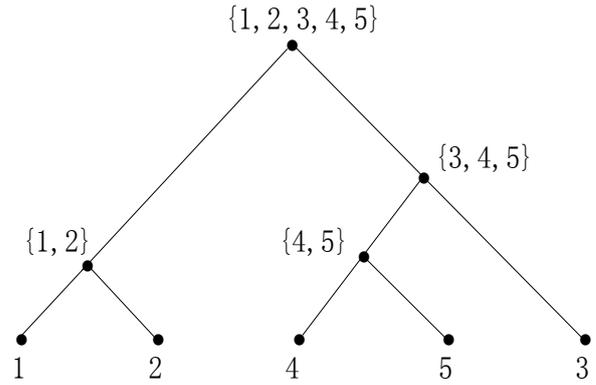


图 5 树结构图

算法 2. 分簇方法^[29].

- ①建立初始的节点分簇 $V_k = \{g_i | i = 1, 2, 3 \dots n\}$, g_i 为单个节点构成的集合分簇, $g_i = \{v_i | v_i \in V_k\}$. 将所有单个节点的集合 g_i 的路径相似度 W_i 设置为整个网络中所有边的均权值 $W(V)$, 遍历所有边集合, 寻找集合间路径相似度最大的 $W(g_m, g_n)$, 从 g_m 和 g_n 随机选择一个成为算法开始的集合 g_k
- ②如果 $|V_k| > 1$ 跳转到③, 否则转到⑩
- ③遍历 V_k , 如果对于 V_k 中任意两个集合间的路径度 $W(g_i, g_j)$ 均小于 $W(V)$ 则跳转到⑨, 否则根据定义 4 寻找使 $W(g_k, g_i)$ 最大的 g_i
- ④如果 $W(g_k, g_i) \geq W(V_k)$ 跳转到⑤, 否则跳转到⑥
- ⑤将 g_i 与 g_k 和并为 g_{ik} , 将 g_{ik} 的路径相似度设为 $W(g_k, g_i)$, 更新分簇 $V_k = V_k \cup g_{ik} \setminus \{g_i, g_k\}$. 跳转到⑦
- ⑥在 $V_k \setminus g_k$ 的分簇中, 挑选路径相似度最大的集合 g_i 成为 g_k 如果 V_k 中路径相似度最大的集合不止一个, 则从这些集合中随机选择一个成为 g_k , 跳到步骤②
- ⑦将通过步骤⑤得到的 g_{ik} 集合放入初始为空的集合 M 中, 集合 M 用来保存每次通过步骤⑤选取得到的分簇
- ⑧遍历 V_k , 选取路径相似度最大的集合 g_i 成为 g_k , 如果 V_k 中路径相似度最大的集合不止一个, 则从这些集合中随机选择一个成为 g_k , 跳到步骤②
- ⑨将剩余 V_k 中的集合选取集合间路径相似度最大的两个集合 g_i 与 g_k 和并为 g_{ik} , 将 g_{ik} 放入第⑦步的 M 中, 更新 $V_k = V_k \cup g_{ik} \setminus \{g_i, g_k\}$, 跳转到②
- ⑩树结构生成完毕, 所有的非叶子节点保存在 M 中, 算法结束

算法 2 的第①步中遍历所有集合, 寻找集合间路径相似度最大的 $W(g_m, g_n)$ 时, 需要计算 C_n^2 对节

点间的路径相似度, 计算的时间复杂度为 $O(n^2)$. 算法最终得到的树形结构有 $(n-1)$ 个非叶子节点, 最外层的循环需要进行 $(n-1)$ 次, 同时, 考虑到合并两个路径相似度最高的节点分簇后, 只需要计算这个新得到的分簇和已经得到的其它分簇之间的路径相似度, 循环体内的计算时间复杂度为 $O(n)$. 因此算法第②到第⑨步的计算时间复杂度为 $O(n^2)$. 综上所述算法 2 的整体时间复杂度为 $O(n^2)$.

通过算法 2 得到的集合 M 中存在一些内部路径相似度不高的分簇, 同时一些分簇之间具有包含关系, 因此需要对得到的集合进行裁剪. 裁剪策略如下: 将 M 中的所有集合的边的平均权值由大到小排序, 删去后一半路径相似度较小的分簇, 遍历 M 中剩余的集合, 如果 M 中的某一个分簇 M_l 被 M 中其他某一分簇所包含, 则将 M_l 从 M 中删除, 如果某些节点不在裁剪后的分簇中, 则自己独立成簇. 通过上述方法得到了内部路径相似度较高的节点分簇, 具体的裁剪算法见算法 3.

算法 3. 集合 M 的裁剪算法.

```

1. MSORT( $M[g_0, g_n]$ ) //使用快速排序算法按路径相似度从大到小排序
2. DELETE( $M[g_{n/2}, g_n]$ ) //删除后一半相似度较小的分簇
3. FOR( $i=0; i++; i < n$ )
4. FOR( $j=i+1; j++; j < n$ )
5.  $R=INCLUSION(g_i, g_j)$  //判断两个集合的包含关系
6. IF( $R=true$ ) THEN
7. IF ( $LENGTH(g_i) <= LENGTH(g_j)$ )
8.   DELETE( $g_i$ )
9. ELSE
10.  DELETE( $g_j$ )
11. END IF
12. END IF
13. END FOR
14. END FOR

```

$INCLUSION(g_i, g_j)$

```

1. { SORT( $g_i$ ) //对  $g_i$  中的元素进行分类
2. SORT( $g_j$ )
3.  $g = LENGTH\_LCS(g_i, g_j)$  //求最长公共子序列
4. IF ( $LENGTH(g) == LENGTH(g_i) \parallel LENGTH(g) == LENGTH(g_j)$ ) THEN
5.   RETURN true
6. ELSE
7.   RETURN false
8. END IF
9. }

```

1.4 分簇信息的维护过程

本文的实验分为两个阶段: 热启动阶段和路由阶段. 在热启动阶段每个节点本地保存 N 个链表, 依据 1.2 节中的统计方法记录网络中所有节点的移动路径信息, 并依据节点间路径相似度的计算模型得到每对节点间的路径相似度, 运用分簇方法对网络中所有节点进行分簇. 在路由阶段应用热启动阶段的分簇结果执行路由方法. 由于容迟网络中节点的运动复杂多变, 即使是校园网络中, 由于节点身份

发生变化, 很可能导致其行为习惯的变化, 因此不能只进行一次分簇, 但毕竟校园网络中节点身份的变化并不频繁, 或者说在很长的时间段内节点的身份不会发生变化, 因此路由阶段可以依据具体的实验场景设置很长的周期, 这也是本文提出的分簇模型维护开销小的原因. 这里特殊强调虽然每个节点都需要对网络中所有节点的运动轨迹进行路径相似度计算并执行分簇方法, 但是由于固定长度的热启动时间决定了每个节点统计得到的其他节点的轨迹信息不一定完全相同, 因此它们做的并不是完全重复的计算, 又由于 DTN 中特殊的网络环境导致集中式的计算方式很难应用, 因此每个节点根据本地存储的节点运动轨迹信息独立运行分簇算法是合理的.

本文针对实验的两个阶段设计如下: 每个热启动阶段的时间长度相同, 每个路由阶段的时间长度也相同, 每个路由阶段末尾的一段时间同时也是下一个热启动阶段 (图 6). 热启动阶段和路由阶段的具体设置见 5.1 节.

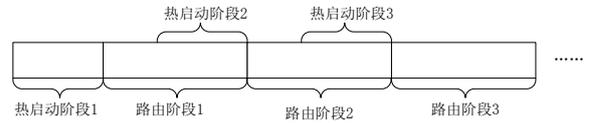


图 6 两个阶段示意图

2 半马尔可夫预测模型

节点 K 的状态集合记为 $L^K = \{1, 2, 3, \dots, l\}$, 用以表示节点 K 所处地点的标号, l 表示总共的地点数, 节点 K 的第 n 个状态 (路径中经过的第 n 个区域的标号) 记为 L_n^K , 节点 K 进入第 n 个状态的时刻 (进入第 n 个区域的时刻) 记为 T_n^K . 对于节点 K 形成了一个时间离散状态离散的马尔可夫链 L_n^K , 即节点 K 的第 $n+1$ 个状态 L_{n+1}^K 只与第 n 个状态 L_n^K 有关, 而与之前的状态无关. 同时对于节点 K 形成了时间连续状态离散的半马尔可夫过程模型 (L_n^K, T_n^K) , 该模型描述了随着时间变化节点 K 的状态转移过程. 节点 K 在第 n 个状态的驻留时间记为 W_n^K , W_n^K 可由 T_{n+1}^K 和 T_n^K 表示, 如 (1) 式.

$$W_n^K = T_{n+1}^K - T_n^K \quad (1)$$

半马尔可夫模型 (L_n^K, T_n^K) 的核心表达式 Z 如 (2) 式. $Z(i, j, t, k)$ 表示节点 K 在 t 时间内由状态 i

转移到状态 j 的概率, 从 (2) 式中可以看出, L_{n+1}^K 的状态取决于当前状态 L_n^K , 与之前的 L_{n-1}^K 等状态无关.

$$\begin{aligned} Z(i, j, t, k) &= P(L_{n+1}^K = j, W_n^K \leq t | L_0^K \dots L_n^K, T_0^K \dots T_n^K) \\ &= P(L_{n+1}^K = j, W_n^K \leq t | L_n^K = i) \end{aligned} \quad (2)$$

根据假设, K 的路径序列满足时间离散状态离散的马尔可夫链 L_n^K , $P(i, j, k)$ 表示节点 K 从状态 i 转移到状态 j 的概率, 统计 L_n^K 的历史序列, 依据马尔可夫链性质, 得到 (3) 式, 其中 num_{ij} 表示 L_n^K 序列中由 i 直接转移到 j 的次数, num_i 表示 L_n^K 序列中 i 出现的次数. (例如, L_n^K 的序列为 2,1,2,3,5, 则 $P(2,1, k) = 1/2$)

$$\begin{aligned} P(i, j, k) &= P(L_{n+1}^K = j | L_n^K = i) \\ &= num_{ij} / num_i \end{aligned} \quad (3)$$

节点 K 由状态 i 直接转移到状态 j 所用的时间小于等于 t 的概率记为 $G(i, j, t, k)$, 表达式如 (4). (例如从 i 直接转移到 j 的统计时间分别有 2,3,3,4,5, 则 $G(i, j, 3, k) = 3/5$)

$$\begin{aligned} G(i, j, t, k) &= P(W_n^K \leq t | L_n^K = i, L_{n+1}^K = j) \\ &= \sum_{u=1}^t P(W_n^K = u | L_n^K = i, L_{n+1}^K = j) \end{aligned} \quad (4)$$

节点 K 的当前状态为 i , t 时间内转移到了其他状态的概率记为 $G(i, t, k)$, 表达式如 (5) 式.

$$\begin{aligned} G(i, t, k) &= P(W_n^K \leq t | L_n^K = i) \\ &= \sum_{j=1, j \neq i}^l G(i, j, t, k) \end{aligned} \quad (5)$$

则依据 (2) (3) (4) 式得出 $Z(i, j, t, k)$ 的具体计算公式, 如 (6) 式.

$$\begin{aligned} Z(i, j, t, k) &= P(L_{n+1}^K = j, W_n^K \leq t | L_0^K \dots L_n^K, T_0^K \dots T_n^K) \\ &= P(W_n^K \leq t | L_n^K = i, L_{n+1}^K = j) \cdot P(L_{n+1}^K = j | L_n^K = i) \\ &= G(i, j, t, k) \cdot P(i, j, k) \end{aligned} \quad (6)$$

半马尔可夫模型 (L_n^K, T_n^K) 的另外一个核心表达式 Q 如 (7) 式. $Q(i, j, t, k)$ 表示节点 K 当前时刻处于 i 状态, t 时间后处于 j 状态的概率, $Q(i, j, t, k)$ 不同于 (2) 式定义的 $Z(i, j, t, k)$, $Z(i, j, t, k)$ 表示节点 K 当前状态为 i , 下一个状态为 j , 并且这个转移的时间不大于 t 的概率, 而 $Q(i, j, t, k)$ 中 i 到 j 不一定是连续的状态转移, 也许在 t 时间内节点 K 经历过其他状态, 只要得到 t 时间后节点 K 处于 j 状态

的概率即可. 特殊强调: $Q(i, j, 0, k) = 0 (j \neq i)$, $Q(i, i, 0, k) = 1$.

$$Q(i, j, t, k) = \begin{cases} \sum_{r=1}^t \sum_{u=1}^t (Z(i, r, u, k) - Z(i, r, u-1, k)) \cdot Q(r, j, t-u, k) & (j \neq i) \\ 1 - G(i, t, k) + \sum_{r=1}^t \sum_{u=1}^t (Z(i, r, u, k) - Z(i, r, u-1, k)) \cdot Q(r, i, t-u, k) & (j = i) \end{cases} \quad (7)$$

对 (7) 式做如下解释: 当 (7) 式中 $j \neq i, r = j, u = t$ 时公式变为 $(Z(i, j, t, k) - Z(i, j, t-1, k)) \cdot Q(j, j, 0, k)$, 由于 $Q(i, i, 0, k) = 1$, 故公式转化为 $(Z(i, j, t, k) - Z(i, j, t-1, k))$ 表示 $t-1$ 秒之前没有发生状态转移, 在最后非常极限的时间 t 到达 j 状态. 另外一种可能: 当 $j \neq i, r \neq j, u \neq t$ 时, 公式 (7) 表示节点 K 在 t 秒内到达过其他状态, 但是最后在 t 秒后到达了 j 状态, 因此当 $j \neq i$ 时 (7) 式能够正确的表示节点 K 当前状态为 i , t 时间后状态为 j 的概率. 同理当 $j = i$ 时也分为两种情况, 第一种情况节点 K 在 t 时间内没有离开 i 状态, 第二种情况 t 时间内节点 K 到达过其他状态, 但是在 t 时间后回到了 i 状态. 第一种情况对应于 $Q(i, i, t, k) = 1 - G(i, t, k)$, 第二种情况对应于 $Q(i, i, t, k) = \sum_{r=1}^t \sum_{u=1}^t (Z(i, r, u, k) - Z(i, r, u-1, k)) \cdot Q(r, i, t-u, k)$, 因此在 $j = i$ 时 (7) 式也能正确表示该概率.

(7) 式运用了动态规划的思想, 其中 $Z(i, j, t, k)$ 可以通过 (6) 式得到, $G(i, t, k)$ 可以通过 (5) 式得到, 初始值 $Q(i, j, 0, k) = 0 (j \neq i)$, $Q(i, i, 0, k) = 1$. 通过动态规划方法逐层求解, 即可算出 $Q(i, j, t, k)$. 本文提出的半马尔可夫预测模型在以下几个方面优于 [26] 中提出的预测模型. 首先 [26] 中设计了一种节点的行为方式: 以 $(1-p)$ 的概率停留在当前区域内, 而以 $p/|L|-1$ 的概率前往其他 $(|L|-1)$ 个区域中的任意一个, 但是在预测模型中却运用基于历史路径信息的马尔可夫模型求解前往其他区域的概率值 p_{ij} , 预测模型并不适用于设计的仿真环境. 而本文节点的运动轨迹采用统计的真实信息, 提出的半马尔可夫预测模型较为适用. 其次 [26] 中 δ_{ij} 表示节点由 i 状态直接变为 j 状态的概率值, 这样的设计不符合真实运动行为, 故本文提出的预测模型在 (7) 式中分为 $i = j$ 和 $i \neq j$ 两种情况讨论, 使预测模型更加准确. 最后 [26] 中节点 t 时间内没有走出当前区域的概率值是通过 t 时间内到达其他区域的概率值来间接计算的, 而本文通过热启动阶段统计停留时间的分布直接计算该概率值, 减小了计算复杂度.

3 相遇概率求解模型

通过半马尔可夫模型可以求出任意节点 K 所对应的 $Q(i, j, t, k)$, 已知节点 a 在 t_a 时刻处于状态 l_a , 节点 b 在 t_b 时刻处于状态 l_b , $t_{\max} = \max\{t_a, t_b\}$, 则 a 和 b 在 t_{\max} 之后的时刻 $t(t > t_{\max})$, 相遇于地点 i 的概率记为 $C(a, b, t, i)$, 如 (8) 式.

$$C(a, b, t, i) = Q(l_a, i, t - t_a, a) \cdot Q(l_b, i, t - t_b, b) \quad (8)$$

根据 (8) 式, a 和 b 在 $t(t > t_{\max})$ 时刻的相遇概率 (任意地点) $C(a, b, t)$ 如 (9) 式.

$$C(a, b, t) = \sum_{i=1}^l C(a, b, t, i) \quad (9)$$

定义 6. 半马尔可夫预测时间窗口. 半马尔可夫预测时间窗口 Q_i 定义为节点 K 的公式 $Q(i, j, t, k)$ 中 t 所能取得的最大值, 为了计算方便, 网络中任意节点的 Q_i 均相同, Q_i 代表了半马尔可夫模型所能预测的最大时间跨度.

定义 7. 相遇概率预测时间窗口. 相遇概率预测时间窗口 C_i 定义为节点 a 和 b 的公式 $C(a, b, t, i)$ 中 t 所能取得的最大值, 为了方便计算, 网络中任意节点对之间的 C_i 均相同, C_i 代表了相遇概率求解模型的最大时间跨度.

每个节点 K 本地保存一个三元组 $\langle k, t_k, l_k \rangle$, k 代表节点编号, t_k 代表进入一个状态的时间, l_k 表示对应的区域 (状态) 编号. 每当 K 进入一个新的区域的时候, 就用最新的三元组替换掉之前的统计三元组, 每当两个节点相遇的时候, 交换彼此没有的三元组, 这样一段时间以后网络中所有节点拥有整个网络的三元组信息.

假设当前时间为 t_n , t_a 和 t_b 均小于等于 t_n , 当要预测 t 时刻 a 和 b 的相遇概率时, 首先考察 $t - t_n$ 和 C_i 的大小关系, 当 $t - t_n \leq C_i$ 时按照 (9) 式正常计算, 否则超出预测能力, 无法计算. 当执行 (9) 式时, 考察 $t - t_a$, $t - t_b$ 与 Q_i 的大小关系, 当 $t - t_a$, $t - t_b$ 均小于等于 Q_i , 则正常计算, 当 $t - t_a$, $t - t_b$ 均大于 Q_i 时 (实验场景中 Q_i 相对于 C_i 比较大, 这种情况几乎不能发生), 将节点 a 和 b 的相遇概率暂时设置为 0. 当 $t - t_a$, $t - t_b$ 只有一个大于 Q_i 时, 不失一般性的假设 $t - t_a > Q_i$, 这时易得 $t_a < t_b$, 需要预测在 t_b 时刻 a 最有可能处于的状态 $l_{\max a}$, 然后用 $\langle a, t_b, l_{\max a} \rangle$ 替换掉节点 a 本地保存的三元组, 继续进行预测. $l_{\max a}$ 的计算方法如下: $l_{\max a} = \arg \max_j Q(l_a, j, t_b - t_a, a)$. 综上所述相遇概率求解模型的流程如图 7.

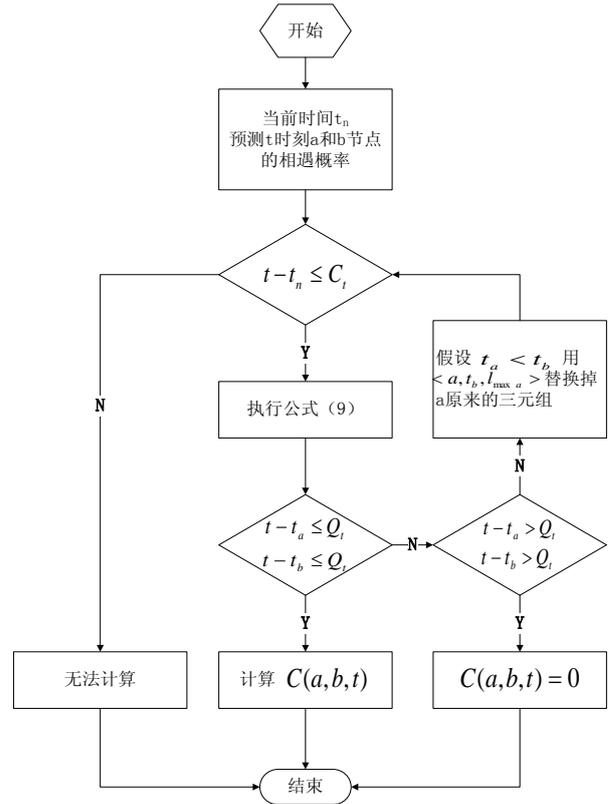


图 7 相遇概率求解模型的流程图

定义 8. a 和 b 在 C_i 时间内的相遇概率 $C_{\max}(a, b, C_i)$. 该概率值定义为在当前时间以后的 C_i 时间内最大的相遇概率, 如 (10) 式, 最大相遇概率所对应的相遇时刻 t_{\max}^{ab} 如 (11) 式.

$$C_{\max}(a, b, C_i) = \max_t C(a, b, t) (t_n \leq t \leq t_n + C_i) \quad (10)$$

$$t_{\max}^{ab} = \arg \max_t C(a, b, t) (t_n \leq t \leq t_n + C_i) \quad (11)$$

根据 (1) - (11) 式, 本文能够计算出网络中任意两个节点间在未来一段时间内的最大相遇概率, 以及对应的相遇时刻, 不同于以往路由方法对于节点间相遇概率的估计, 基于动态半马尔可夫路径搜索模型的 DTN 分簇路由方法不仅能够预测未来相遇的可能性, 同时能够预测出在某一时间两个节点相遇的可能性大小, 这样必然会对接下来的路由方法带来巨大帮助.

4 动态路由表

4.1 路由节点集合的确定

在热启动阶段结束之后, 每个节点可以统计到网络中所有节点的移动轨迹信息, 依据分簇算法得到所有节点的分簇结果, 当源节点 a 要向目的节点

d 路由报文的时候, a 首先判断 a 和 d 是否在一个分簇内, 如果在一个分簇内, 则将该分簇内的所有节点定义为路由该报文的节点集合. 如果不在一个分簇内, 则将 d 所在分簇的所有节点和节点 a 定义为路由该报文的节点集合, 在该集合中运用路径搜索算法生成动态路由表.

4.2 路径搜索策略

定义 9. t 时刻的路由图 $O = (V_t, E_t, t)$. 其中 V_t 为 4.1 节中所确定的参与路由的节点集合, E_t 为定义在 O 上的边集合, $a, b \in V_t$, $e_{a,b} \in E_t$, $e_{a,b}$ 上需要记录 a 和 b 在 C_t 时间内最大的相遇概率 $C_{\max}(a, b, C_t)$ 和最大相遇概率所对应的时刻 t_{\max}^{ab} , 如图 8.

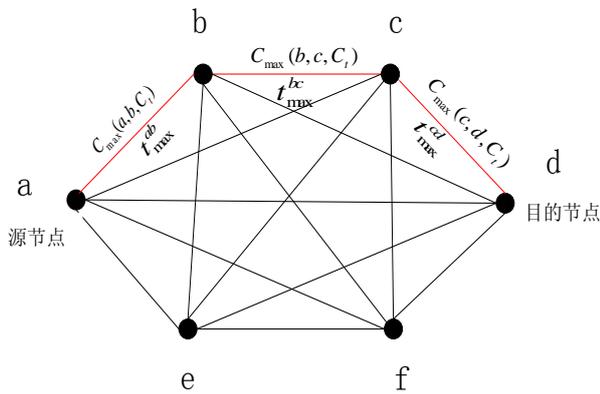


图 8 路径搜索图

定义 10. 源到目的节点的一条可走通路径. 路径 $a_0 a_1 a_2 \cdots a_n$ ($n \geq 1$) 为一条可走通路径, 当且仅当 $t_{\max}^{a_{n+1}a_n} > t_{\max}^{a_n a_{n-1}}$ ($n \geq 1$).

定义 11. 路径投递概率 C_{path} . 对于路径 $a_0 a_1 a_2 \cdots a_n$ ($n \geq 1$), $C_{path} = \prod_{i=0}^{n-1} C_{\max}(a_i, a_{i+1}, C_t)$.

定义 12. 源到目的节点的最优路径. 定义为源到目的节点的可走通路径中 C_{path} 最大的路径.

算法 4 为最优路径搜索算法, 其中 π 表示节点的后继, H 表示节点集合. C 表示节点间最大相遇概率, 1800s 是实验中选取的预测窗口大小.

算法 4. 路径搜索算法.

$FIND_PATH(a, b, t)$

```
1 best ← ∅
2 FIND_NEXT(a, b, t)
3 RETURN best
```

$PATH_PROB(\pi)$

```
1 P_path ← 1
2 FOR each (a, p) in π
3   P_path ← P_path * P
4 RETURN P_path
```

$FIND_NEXT(a, b, t)$

```
1 p ← max{ C_ab(k) | k in [t, t + 1800] }
2 π(a) ← (b, p)
3 IF PATH_PROB(π) > PATH_PROB(best)
4   best ← π
5 FOR each c ∈ H - {a, b}
6   (p, k) ← max{ (C_ac(k), k) | k in [t, t + 1800] }
7   π(a) ← (c, p)
8 IF PATH_PROB(π) > PATH_PROB(best)
9   FIND_NEXT(c, b, t)
10 π(a) ← NIL
```

4.3 动态路由表的维护

节点路由表的初始化: 节点根据实验热启动阶段得到的信息初始化路由表, 依据热启动阶段得到的任意两个节点在预测时间内的最大相遇概率和最大相遇概率对应的时刻, 生成路由图, 依据最优路径搜索算法得到当前时刻该节点到达目的节点的最优路径上的下一跳节点的编号, 生成形如<下一跳节点, 目的节点>的二元组, 进而生成与时间相关的动态路由表.

节点路由表的更新: 新报文的产生驱使动态路由表的更新. 每当有新到报文时, 根据报文的节点计算最优路径上的下一跳节点, 重新生成与当前时刻有关的二元组, 更新路由表中原有的选路方法, 完成动态路由表的更新工作.

综上所述, 基于动态半马尔可夫路径搜索模型的 DTN 分簇路由方法 CRSMP 首先根据节点间路径的相似程度完成节点分簇, 然后依据半马尔可夫预测模型预测出节点当前时刻在 i 并且 $t(t \leq Q_t)$ 时间之后到达 j 的概率值, 进一步计算出任意两个节点间在 C_t 时间内的最大相遇概率以及最大相遇概率值所对应的相遇时刻, 根据源节点和目的节点的分簇关系确定用来路由报文的节点集合, 依据最优路径搜索算法完成动态路由表的建立和更新, 按照动态路由表信息路由报文. 实验结果表明该路由方法明显提高了投递成功率, 减小了网络的平均时延, 并且在一定程度上减小了负载比率.

5 性能评估

5.1 仿真环境设置

为了测试 CRSMP 的路由性能, 本文采用机会网络模拟器 ONE 仿真该实验, 在 ONE 自带的赫尔辛基市地图上进行修改, 设计了一个尽量真实的校园环境作为实验场景 (图 9), 在图 9 (4500m×3400m) 的仿真环境中设计了 10 个区域, 节点间的通信只能发生在区域内, 并且进入同一区域内的节点间均可以通信, 节点间能够通信的充分必要条件是这两个节点同时处于同一个通信区域内, 每个区域是一个半径为 200m 的圆, 用于模拟如表 1 中的校园网络环境. 节点采用 1.1 节中描述的运动轨迹模型, 每个区域内对应的 n 值见表 2.

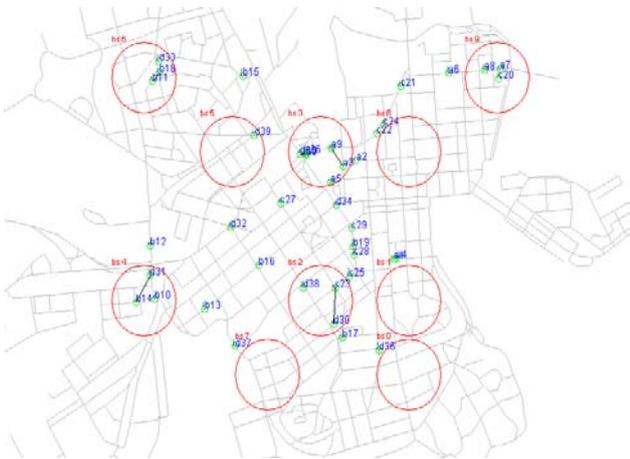


图 9 实验场景图

表 1 区域编号说明

区域编号	校园场景
0	自习室2
1	计算机学院
2	数学学院
3	食堂
4	体育馆
5	图书馆
6	校外
7	银行
8	自习室1
9	寝室

表 2 区域 n 值说明

区域	0	1	2	3	4	5	6	7	8	9
n	4	8	8	2	4	4	3	2	4	8

本文采用调查问卷的方式, 对计算机学院的老师 (20 人), 计算机学院的研究生 (20 人), 数学学院的老师 (20 人), 数学学院的研究生 (20 人) 共 80 人统计了他们工作日内在设定的这 10 个区域中的运动轨迹, 并将这个轨迹信息赋值给节点的区域编号序列, 节点遵循调查问卷统计的轨迹进行运动, 由于调研能力有限, 本文只对学校 80 个人的运动行为作了统计, 但由于调查问卷采取随机发放方式, 因此每一个分组内的 20 个人能够代表典型的校园环境该类身份人员的一般运动轨迹, 各类人员的运动轨迹差异也较大, 适合进行基于路径相似度的分簇算法. 当进行实验时分别从 4 组人群中各抽取 6, 8, 10, 12, 14 个人, 组成节点总数为 24, 32, 40, 48, 56 的实验环境. 为了防止时间粒度过细而导致实验计算量过大, 也为了避免时间粒度过粗致使错过了一些节点间的信息统计, 导致算法精度降低, 本文采用 60s 为一个时间单元. 实验部分依据统计得到的节点运动周期将热启动阶段设置为 9000s, 每个节点 K 在该阶段统计其他节点的移动路径信息, 运用分簇方法得到所有节点的分簇结果, 同时统计 $P(i, j, k)$, $G(i, j, t, k)$ 和 $G(i, t, k)$. 热启动阶段结束之后, 进入路由阶段, 应用 CRSMP 路由方法仿真 18000s, 在路由阶段的最后 9000 秒进入第二个热启动阶段. 由于本文实验条件有限, 只仿真了一个热启动阶段和一个路由阶段, 但已经能够说明 CRSMP 路由算法的性能. 实验参数的详细配置如表 3.

表 3 参数说明

参数	值
仿真时间	18000s
仿真区域大小	4500m×3400m
节点数量	24, 32, 40, 48, 56
节点移动速度	1-2m/s
传输速度	250Kbps
缓存大小	5M, 10M, 15M, 20M, 25M
消息产生时间间隔	[5, 15][15, 25][25, 35][35, 45][45, 55]
TTL	300
报文大小	0.5-1 M
时间单元	60s

半马尔可夫预测时间窗口 Q_i	3600s
相遇概率预测时间窗口 C_i	1800s

本文从以下三方面评估 CRSMP 协议性能：

1. 投递成功率 = 成功投递到目的节点的报文数量 / 网络中产生的报文总数
2. 平均时延 = 消息到达目的节点的平均时间
3. 负载比率 (开销比) = (利用连接成功传递包的次数 - 传递到目的节点的包的个数) / 传递到目的节点的包的个数

5.2 仿真结果分析

为了全面验证 CRSMP 的路由性能，本文针对投递成功率，平均时延和负载比率三个标准来评估 CRSMP 路由方法。前两个标准比较容易理解，负载比率又叫开销比，能反映连接的使用效率，负载比率高说明更多的连接用来传递的数据包都没能成功传递到目的节点，连接的使用效用较低，负载比率低反而说明连接的有效使用率高。

本文考虑到以下几个因素最有可能影响路由算法的性能：节点总数，报文产生速率以及节点缓存大小。由于 CRSMP 为单副本路由方法，因此考虑与另外三种单副本路由协议 (DirectDelivery, FirstContact, SimBet) 以及三种多副本路由方法 (Spray and Wait, Epidemic, Prophet) 进行比较说明 CRSMP 的路由性能。在仿真中依据 Spray and Wait 路由算法初始副本数的不同又分为 4, 6, 8 三种情况。考虑相对比的路由方法中 DirectDelivery, FirstContact, Epidemic, Spray and Wait 和 Prophet 这 5 种大家都比较熟悉，因此着重对 SimBet 路由方法进行介绍。

SimBet 是一种考虑节点社会属性的单副本路由方法，将节点间的共同邻居数定义为相似性 (Sim 效用)，将自身对其他节点间连接的帮助程度定义为中介中心性 (Bet 效用)，依据这两个效用值计算出节点对于 SimBet 路由方法的效用值 (SimBetUtil)，依据 SimBetUtil 的大小进行报文的路由决策。例如节点 a 有消息要转发给节点 d ，当 a 与 b 相遇时，通过 (12) (13) 和 (14) 计算 a 的 SimBet 效用值，节点 b 同理。

$$SimUtil_a(d) = \frac{Sim_a(d)}{Sim_a(d) + Sim_b(d)} \quad (12)$$

$$BetUtil_a = \frac{Bet_a}{Bet_a + Bet_b} \quad (13)$$

$$SimBetUtil_a(d) = \alpha SimUtil_a(d) + \beta BetUtil_a \quad (14)$$

公式 (14) 中 α 和 β 为两个可调参数，满足 $\alpha + \beta = 1$ ，为了确定本文实验场景中的 α 和 β 的取值，首先在默认配置下 (节点总数 40，报文产生速率为每 25-35 产生一个，缓存大小 10M)，测试了不同 α 下 SimBet 的投递成功率 (图 10)。

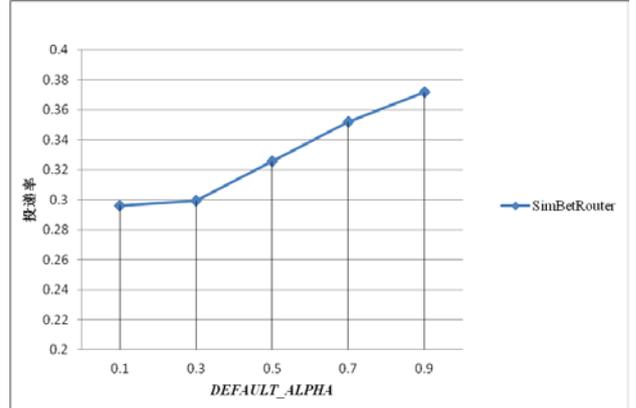


图 10 α 参数设置测试图

从图 10 中可以看出随着 α 的增加，投递成功率也在增加，分析其原因，主要是小缓存下 (10M) SimBet 路由方法依据中介中心性进行路由有可能导致一些节点 (Bet 效用值较高) 迅速发生缓存溢出，导致投递率的下降，因此 β 取值越小，投递效果越好。在接下来的实验中将 α 值设置为 0.9， β 值设置为 0.1。

为了确定半马尔可夫预测时间窗口 Q_i 对 CRSMP 的影响，在默认配置下，将相遇概率预测时间窗口 C_i 设置为 1800 秒，分别设置 Q_i 为 60, 70, 80, 90 个时间单元，测试了路由算法的投递成功率。从图 11 中数据可以看出不同的 Q_i 对路由性能影响不大，但当 Q_i 取值小于 60 个时间单元的时候程序抛出数组越界的异常，说明统计 $G(i, j, t, k)$ 时一些转移时间超出了预测时间窗口，这样会产生很大误差，因此在实验环境中设置 Q_i 为 60 个时间单元，即 3600 秒。

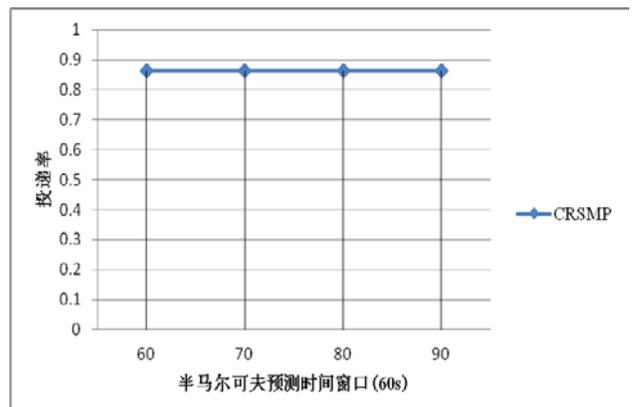


图 11 半马尔可夫预测窗口测试图

为了确定相遇概率预测时间窗口 C_i 对 CRSMP 的影响, 其他设置不变, 将 Q_i 设置为 60 个时间单元, 对不同 C_i 下 CRSMP 的投递成功率进行了测试, 从图 12 中数据可以看出, C_i 取值 1800s 时投递成功率最高, 因此仿真环境将 C_i 设置为 1800 秒.

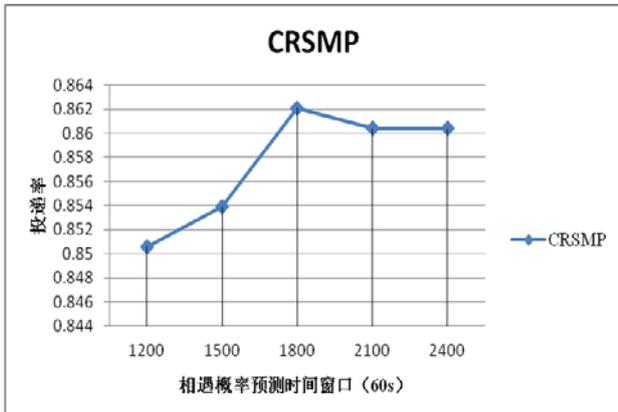


图 12 相遇概率预测窗口测试图

在仿真环境中确定了 SimBet 路由算法的参数取值以及 CRSMP 的 Q_i 和 C_i 取值后, 首先测试不同的节点总数对路由算法的影响, 报文产生速率设置为每 25-35 秒产生一个, 节点缓存大小设置为 10M, 总节点数依次为 24,32,40,48,56, 得到投递成功率 (图 13), 平均时延 (图 14) 和负载比率 (图 15) .

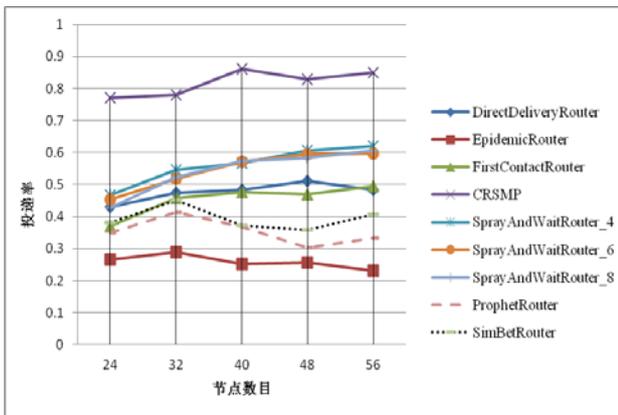


图 13 不同节点数目下的投递成功率

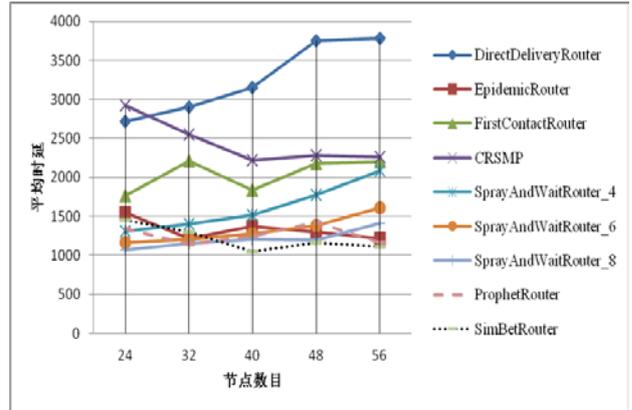


图 14 不同节点数目下的平均时延

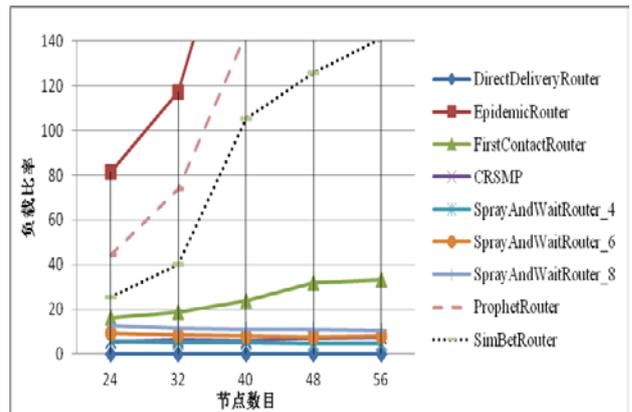


图 15 不同节点数目下的负载比率

从图 13 中数据可以看出 CRSMP 在相同的实验条件下有着很高的投递成功率, 平均在 0.82 左右, 远高于相同条件下的其他路由方法. 由于 SimBet 路由方法一直将报文转发给相似性 (Sim 效用) 和中介中心性 (Bet 效用) 较高的节点, 导致大量报文都流向了连接度高的节点, 致使节点负载不均衡. 据文献^[30]统计, SimBet 路由方法中 10% 的节点承担了 54% 的报文转发和 85% 的报文递交任务, 而这 10% 节点在 10M 的缓存下不足以承担如此庞大的报文数量, 进而丢弃大量报文, 导致投递成功率较低. Prophet 的投递成功率低于 SimBet 路由方法, 说明在节点具有社会属性的网络环境下, 单纯依靠相遇概率来指导报文转发的路由方法并不适用. 在 10M 缓存下 Epidemic 的投递成功率很低, 主要是因为网络中发生了大量拥塞现象, 导致报文无法成功投递, 丢包率很高.

在图 14 中 CRSMP 并没有取得很低的平均时延, 经分析主要是因为 ONE 中平均时延定义为成功投递的报文所经历的平均时间, 而除了 CRSMP 以外的路由方法中长延时的报文在缓存为 10M 的情况下很难投递成功, 因此平均时延相对较低, 在

下面的实验方案中通过增加缓存大小发现，其他路由方法的平均时延急剧上升，而 CRSMP 由于是单副本路由协议，对缓存的变化不敏感，平均时延比较稳定。SimBet 路由方法的平均时延低于其他路由方法，经分析 SimBet 始终选择 *SimBetUtil* 更高的节点作为报文的下一跳中继，而 *SimBetUtil* 高说明该节点与目的节点的相似性和自身的中介中心性较高，这两种性质可以加快报文递交给目的节点的速度，因此 SimBet 路由方法的平均时延最低。

在图 15 中 CRSMP 有着与 4 副本的 Spray and Wait 路由方法差不多的负载比率，明显低于其他几种路由方法。SimBet 路由方法的负载比率仅仅低于 Epidemic 和 Prophet，而高于其他路由方法，这主要是因为 SimBet 路由方法下报文往往需要经过多跳中继节点才能到达连接度较大的节点，然后很有可能在连接度较大的节点上被丢弃（连接度较大的节点上缓存发生溢出），而这些丢弃的报文只增加了转发数据包的次数，并不增加成功投递到目的节点的报文数，因而 SimBet 路由方法的负载比率较高。DirectDeliveryRouter 的负载比率一直为 0，是因为其路由方式决定了其利用连接成功传递包的次数等于传递到目的节点的包的个数。

完成不同的节点总数下路由算法的测试后，本文仿真了不同报文产生速率下路由算法的性能。同样将节点总数设置为 40，节点缓存大小设置为 10M，在不同的报文生成速率下，得到路由方法的投递成功率（图 16），平均时延（图 17），负载比率（图 18）。

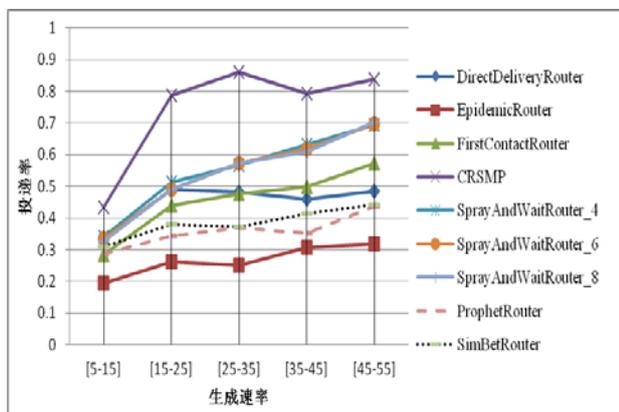


图 16 不同报文生成速率下的投递成功率

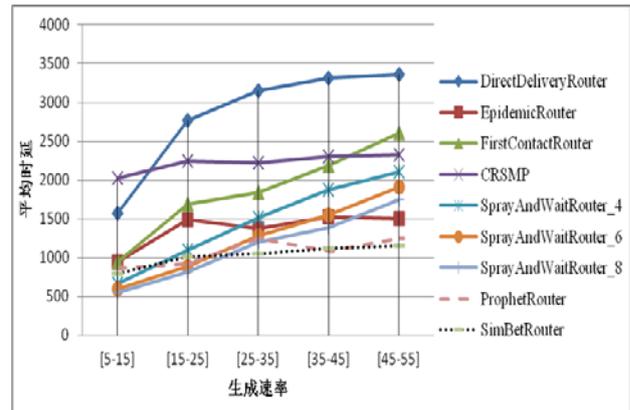


图 17 不同报文生成速率下的平均时延

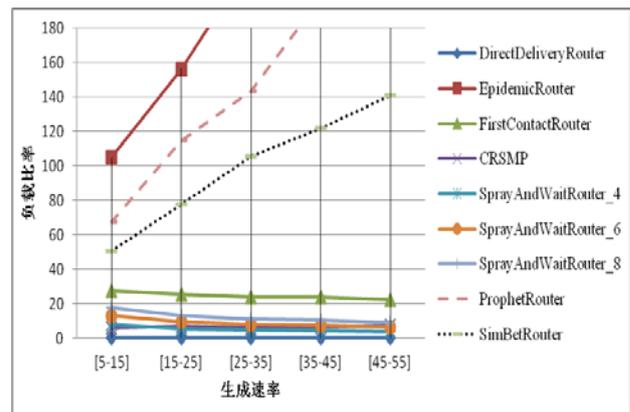


图 18 不同报文生成速率下的负载比率

从图 16 中数据分析得出，不同报文产生速率下，CRSMP 保持较高的投递成功率，当报文产生速度很快时，CRSMP 投递概率呈现稍许下降趋势，因为过多的报文也导致了一部分拥堵的发生，影响了 CRSMP 的投递成功率，但总体来说远高于其他路由方法。图 17 中数据显示 CRSMP 的平均时延很稳定，没有大幅度波动，并且随着缓存的增加，平均时延少于另外两种单副本路由方法（DirectDelivery, FirstContact）。图 18 中数据与图 15 中数据非常相似，在此不再赘述。

接下来仿真了不同缓存大小对路由性能的影响，将节点总数设置为 40，报文的产生速率为每 25-35 秒产生一个，在不同缓存大小下得到不同路由方法的投递成功率（图 19），平均时延（图 20），负载比率（图 21）。

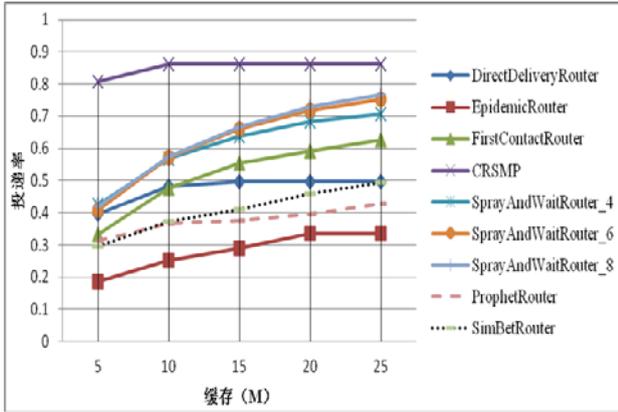


图 19 不同缓存下的投递成功率

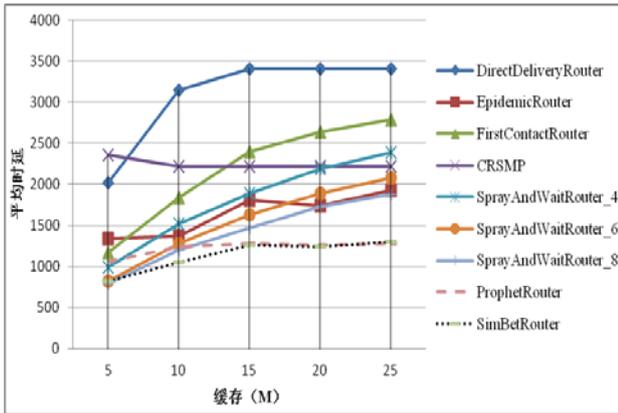


图 20 不同缓存下的平均时延

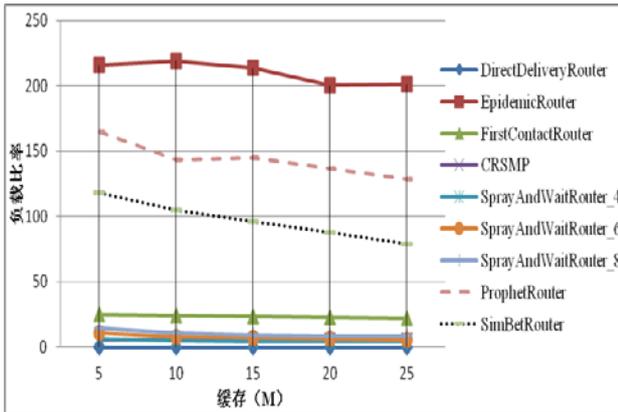


图 21 不同缓存下的负载比率

从图 19 中数据可以看出，CRSMP 对缓存要求非常低，在 5M 的节点缓存下就已经到达 0.8 左右的投递概率，从某种意义上讲该路由方法节省了大量网络资源，这主要是由于其单副本特性所致。SimBet 路由方法随着缓存增加投递率明显提升，验证了本文之前的论述：小缓存下连接度较大的节点负载过高，缓存发生溢出，导致投递成功率下降。在图 20 数据中发现当缓存增大时 CRSMP 渐渐表现出了平均时延方面的优势，之前的分析得到验证。从图 21 中数据可以看出 Epidemic 的负载比率一直

是最高的，而 CRSMP 的负载比率近似于 4 副本的 Spray and Wait 路由协议，性能良好。SimBet 路由方法的负载比率随着缓存增大呈现下降趋势，进一步说明缓存的增加减少了报文的丢弃，增加了成功投递到目的节点的报文数，使负载比率降低。

为了进一步验证 CRSMP 平均时延方面的优势，本文设计了实验来统计大缓存下不同路由方法的 3 个评估标准：投递成功率（图 22），平均时延（图 23），负载比率（图 24）。

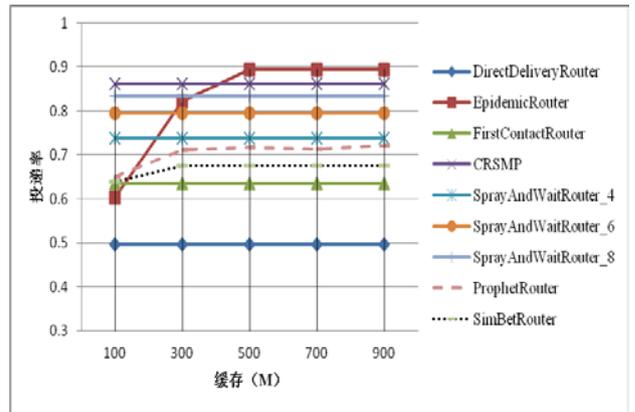


图 22 较大缓存下的投递成功率

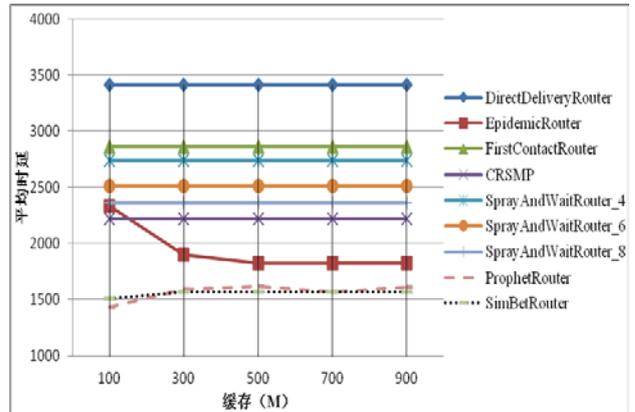


图 23 较大缓存下的平均时延

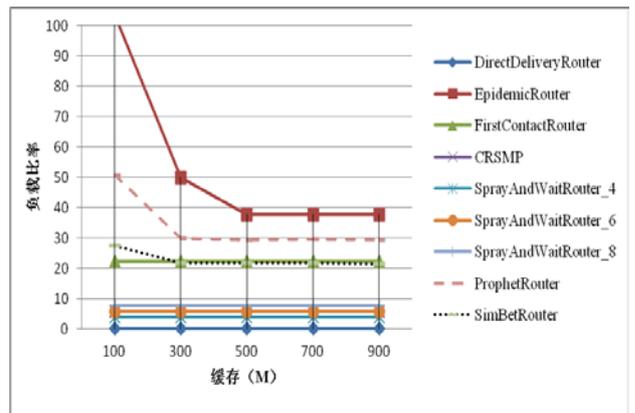


图 24 较大缓存下的负载比率

从图 22 的数据看出当节点缓存近似无限大时, Epdemic 只能达到 0.9 的投递成功率, 而 CRSMP 在 5M 的缓存下就已经达到 0.8 以上的投递成功率, 进一步说明了 CRSMP 的高投递性.图 23 验证了本文的分析, 当缓存增大到一定程度, CRSMP 在平均时延表现出了良好的性能.图 24 显示 CRSMP 的负载比率在大缓存下依然较低, 当缓存增大到一定程度, SimBet 路由方法的负载比率与 FirstContactRouter 相近, 仅仅高于 CRSMP 和其他限制报文转发次数的路由协议.

5.3 真实数据集上的性能评估

为了进一步验证 CRSMP 的路由性能, 本文在真实的数据集上对 CRSMP 在内的 9 种路由方法进行测试, 分别测试了不同缓存大小和不同报文生成速率下这 9 种路由方法的投递成功率, 平均时延和负载比率, 真实数据集上的测试结果表明 CRSMP 与其他路由方法相比依然有着较高的投递成功率, 较小的平均时延和负载比率.

本文在该实验部分采用 NCSU^[31]这一真实数据集, 该数据集统计了著名的美国北卡罗来纳州立大学(North Carolina State University)的在校学生日常行为的 GPS 信息, 该数据集共包含 35 个轨迹记录, 这些记录均以天为单位, 这些记录的行为主体是在学校中随机选取的学生, 因此这些数据比较真实, 具有说服力, 也符合本文的实验场景.

在 ONE 模拟其中导入了由 GPS 信息处理得到平面坐标轨迹, 在模拟器中同时仿真了这 35 个节点的运动行为, 依据这 35 个节点在仿真时间内的运动范围确定了 $15000m \times 10000m$ 的地图大小, 依据这些节点移动的聚集程度和热点区域, 在地图中定义了 9 个通信区域 (图 25), 这些通信区域均为半径 300m 的圆, 节点间能够通信的条件是两个节点同时处于同一通信区域内, 默认节点缓存大小为 30M, 报文生成速率为每 40-45 秒产生一个报文, 其他参数配置与 5.1 节的表 3 相同.

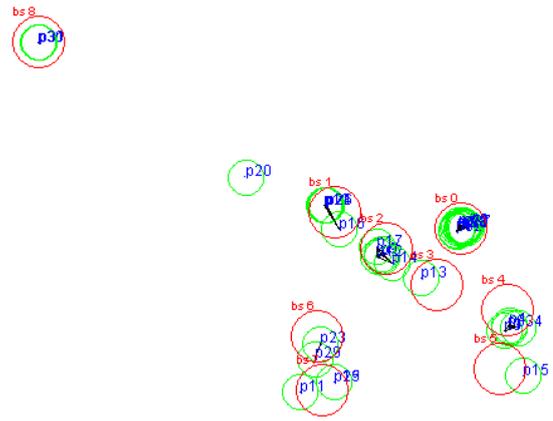


图 25 NCSU 数据集的通信区域设置

首先固定报文的生成速率为每 40-45 秒产生一个报文, 测试不同的缓存大小对 CRSMP 等 9 种路由方法性能的影响.将缓存大小分别设置为 20, 22,24,26,28,30M, 测得 9 种路由方法的投递成功率 (图 26), 平均时延 (图 27) 和负载比率 (图 28) .

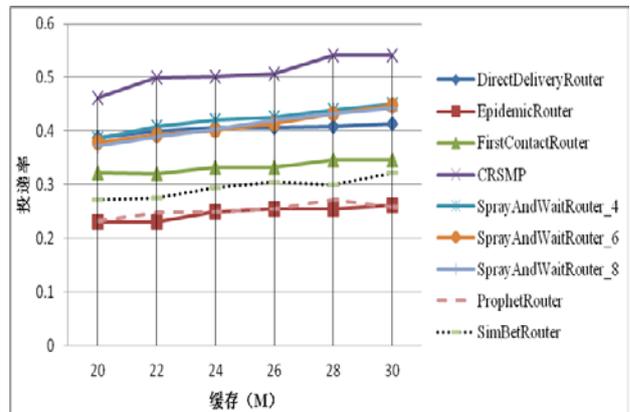


图 26 不同缓存下的投递成功率

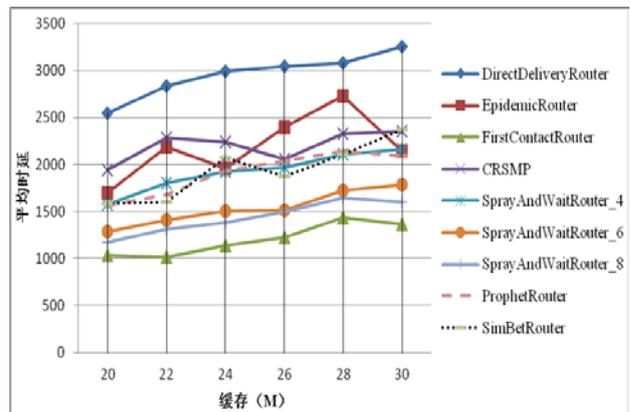


图 27 不同缓存下的平均时延

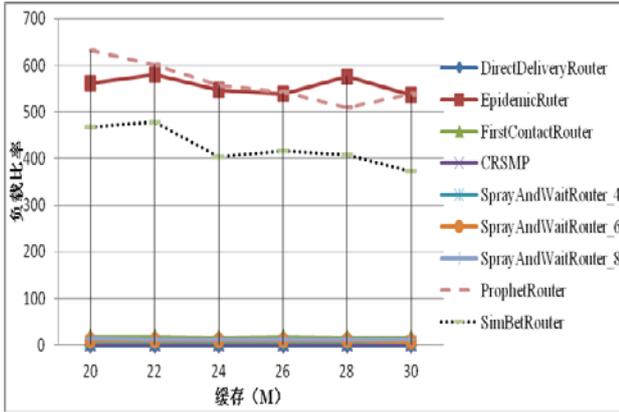


图 28 不同缓存下的负载比率

从图 26 中的数据可见不同的缓存大小下，CRSMP 路由方法与其他 8 种路由方法相比依然有着最好的投递成功率，并且随着缓存大小的增加投递成功率逐渐上升。从图 27 中数据可见 CRSMP 在平均时延方面并没有表现出最好的性能，在之前的仿真实验中也出现了同样的情况，在 5.2 节中已经解释了原因，在此不再赘述。图 28 中的数据 displays CRSMP 的负载比率依旧较低。

接下来固定节点缓存大小为 30M，测试不同的报文生成速率对 CRSMP 等 9 种路由方法性能的影响。将报文生成速率分别设置为 [30-35],[35-40],[40-45],[45-50],[50-55]，测得 9 种路由方法的投递成功率（图 29），平均时延（图 30）和负载比率（图 31）。

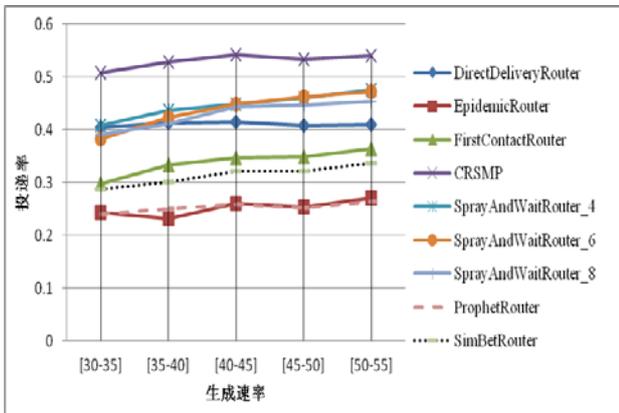


图 29 不同报文生成速率下的投递成功率

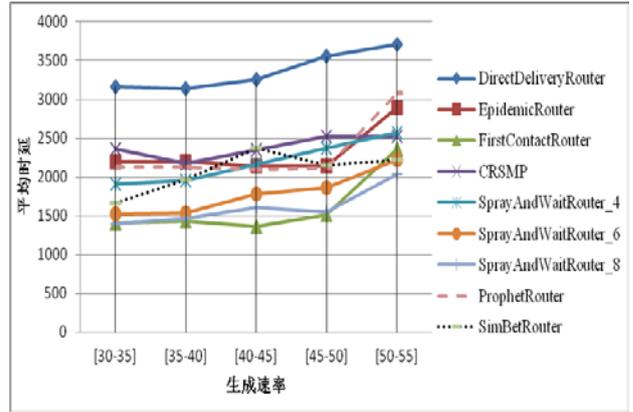


图 30 不同报文生成速率下的平均时延

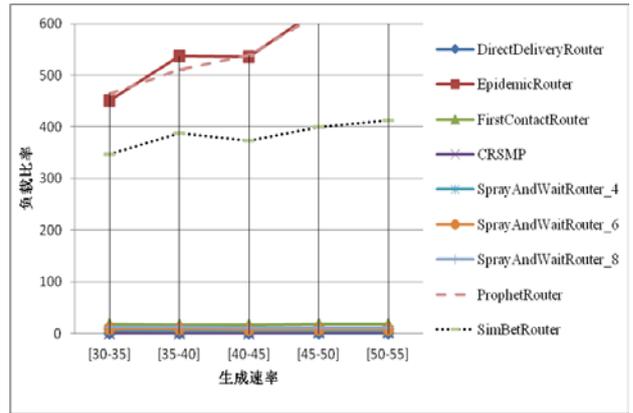


图 31 不同报文生成速率下的负载比率

图 29 中数据可见不同报文生成速率下，CRSMP 的投递成功率高于其他路由方法，与 8 副本的 Spray and Wait 相比依然高出 10 个百分点左右。图 30 与图 31 的数据与 5.2 节仿真实验中的数据相近，在此同样不再赘述。

综上所述，基于动态半马尔可夫路径搜索模型的 DTN 分簇路由方法 CRSMP 与其他路由方法相比，提高了投递成功率，减小了网络平均时延，并且在一定程度上减少了网络负载。

6 总结与展望

本文在机会网络模拟器 ONE 上仿真了近似真实的校园网络环境，设计了一套模拟校园中学生和老师行为的运动模型，提出了基于动态半马尔可夫路径搜索模型的 DTN 分簇路由方法 CRSMP。首先依据仿真环境中节点间的路径相似度对其进行分簇，然后运用半马尔可夫预测模型得到节点间未来某一段时间内的最大相遇概率，以及该概率所对应的相遇时刻。在路由过程中依据源节点和目的节点所在的分簇信息确定参与路由的节点集合，运用最优

路径搜索策略生成与时间有关的动态路由表, 依据路由表信息进行路由. 实验结果表明, 该路由方法与 DirectDeliveryRouter, FirstContactRouter, Epidemic, SimBetRouter, Prophet 以及 Spray and Wait (初始副本分别为 4,6,8) 路由协议相比, 提高了投递成功率, 减小了平均时延, 并且在一定程度上也减小了负载比率. 为了进一步验证 CRSMP 的路由性能, 本文在真实的数据集 NCSU 上对 CRSMP 在内的 9 种路由方法进行测试, 实验结果同样表明 CRSMP 路由方法有着最好的投递成功率和较少的平均时延以及负载比率.

参 考 文 献

- [1] Burleigh S, Hooke A, Torgerson L. Delay-tolerant networking: an approach to interplanetary internet. *Communications Magazine, IEEE*, 2003, 41(6): 128-136
- [2] Fall K. A delay-tolerant network architecture for challenged internets//*Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications(SIGCOMM'03)*. Karlsruhe, Germany, 2003: 27-34
- [3] Akyildiz I F, Akan Ö B, Chen C. InterPlanetary internet: state-of-the-art and research challenges. *Computer Networks*, 2003, 43(2): 75-112
- [4] Long Zhang, Xianwei Zhou, Jiwen Guo. Noncooperative dynamic routing with bandwidth constraint in intermittently connected deep space information networks under scheduled contacts. *Wireless Personal Communications*, 2013, 68(4): 1255-1285
- [5] Krishnan R, Basu P, Mikkelsen J M. The spindle disruption-tolerant networking system// *Proceedings of the Military Communications Conference(MILCOM'07)*. Orlando, USA, 2007: 1-7
- [6] Pentland A, Fletcher R, Hasson A. Daknet: rethinking connectivity in developing nations. *Computer*, 2004, 37(1): 78-83
- [7] Juang P, Oki H, Wang Y. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet// *Proceedings of the ACM Sigplan Notices (SIGPLAN'02)*. Berlin, Germany, 2002, 37(10): 96-107
- [8] Hui P, Chaintreau A, Scott J. Pocket switched networks and human mobility in conference environments//*Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (SIGCOMM'05)*. Philadelphia, USA, 2005: 244-251
- [9] Su J, Chin A, Popivanova A. User mobility for opportunistic ad-hoc networking//*Proceedings of the Mobile Computing Systems and Applications, 2004(WMCSA'04)*. Lake District National Park, UK, 2004: 41-50
- [10] Hsu W, Helmy A. On nodal encounter patterns in wireless LAN traces// *Proceedings of the Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on(WiOpt'06)*. Boston, USA, 2006: 1-10
- [11] Peng Min, Hong Pei-Lin, Xue Kai-Ping. Delivery probability prediction based efficient routing in dtn. *Chinese Journal of Computers*, 2011, 34(001): 174-181(in Chinese)
(彭敏, 洪佩琳, 薛开平. 基于投递概率预测的 DTN 高效路由. *计算机学报*, 2011, 34(001): 174-181)
- [12] Grossglauser M, Tse D N C. Mobility increases the capacity of ad hoc wireless networks. *Networking, IEEE/ACM Transactions On*, 2002, 10(4): 477-486
- [13] Xu Jia, Wang Ru-Chuan, Sun Li-Juan. Research on routing techniques on delay tolerant networks. *Journal of Computer Research and Development*, 2011,48(Suppl.): 271-275(in Chinese)
(徐佳, 王汝传, 孙力娟. 容迟网络路由技术研究. *计算机研究与发展*, 2011,48(Suppl.): 271-275)
- [14] Spyropoulos T, Psounis K, Raghavendra C S. Spray and wait: an efficient routing scheme for intermittently connected mobile networks//*Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (SIGCOMM'05)*. Philadelphia, USA, 2005: 252-259
- [15] Vahdat A, Becker D. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, 2000
- [16] Zhang Long, Zhou Xian-Wei, Wang Jian-Ping. Routing protocols for delay and disruption tolerant networks. *Journal of Software*, 2010, 21(10): 2554-2572(in Chinese)
(张龙, 周贤伟, 王建萍等. 容迟与容断网络中的路由协议. *软件学报*, 2010, 21(10): 2554-2572)
- [17] Lindgren A, Doria A, Schelén O. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 2003, 7(3): 19-20
- [18] Burns B, Brock O, Levine B N. MV routing and capacity building in disruption tolerant networks// *Proceedings of the INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE (INFOCOM'05)*. Miami, USA, 2005: 398-408.
- [19] Daly E, Haahr M. Social network analysis for routing in disconnected delay-tolerant MANETs//*Proceedings of the 8th ACM international Symposium on Mobile Ad Hoc Networking and Computing(MobiHoc'07)*. Montreal, Canada, 2007: 32-40
- [20] Li Q, Gao W, Zhu S. A routing protocol for socially selfish delay

- tolerant networks. *Ad Hoc Networks*, 2012, 10(8): 1619-1632
- [21] Li Zhi, Li Qian-Mu, Zhang Hong. Closely social circuit based routing in social delay tolerant networks. *Journal of Computer Research and Development*, 2012, 49(6): 1185-1195(in Chinese)
(李陟, 李千目, 张宏. 基于最近社交圈的社交时延容忍网络路由策略. *计算机研究与发展*, 2012, 49(6): 1185-1195)
- [22] Abdelkader T, Naik K, Nayak A. Sgbr: A routing protocol for delay tolerant networks using social grouping. *IEEE Transactions on Parallel and Distributed Systems*, 2012, PP(99): 1-1.
- [23] Zhou Rui-Tao, Cao Yuan-Da, Hu Jing-Jing. Community based routing in delay and tolerance networks. *Transaction of Beijing Institute of Technology*, 2012, 32(009): 966-970(in Chinese)
(周瑞涛, 曹元大, 胡晶晶. 基于社区的容迟网络路由方法. *北京理工大学学报*, 2012, 32(009): 966-970)
- [24] Schurgot M R, Comaniciu C, Jaffres-Runser K. Beyond traditional DTN routing: social networks for opportunistic communication. *Communications Magazine, IEEE*, 2012, 50(7): 155-162
- [25] Yu Hai-Zheng, Ma Jian-Feng, Bian Hong. Social network-based trustworthy routing in delay tolerant networks. *Journal on Communication*, 2010, 31(12): 21-26(in Chinese)
(于海征, 马建峰, 边红. 容迟网络中基于社会网络的可靠路由. *通信学报*, 2010, 31(12): 21-26)
- [26] Zhang Zhen-Jing, Jin Zhi-Gang, Shu Yan-Tai. Efficient routing in social DTN based on nodes' movement prediction. *Chinese Journal of Computers*, 2013,36(3):627-635 (in Chinese)
(张振京, 金志刚, 舒炎泰. 基于节点运动预测的社会性 DTN 高效路. *计算机学报*, 2013,36(3): 627-635)
- [27] Yuan Q, Cardei I, Wu J. An efficient prediction-based routing in disruption-tolerant networks. *Parallel and Distributed Systems, IEEE Transactions on*, 2012, 23(1): 19-31
- [28] Yu Hai-Ying. The comparison of the LCS algorithm with the GST algorithm in Strings similarity metrics. *Electronic Sci &Tech*, 2011, 24(3): 101-103(in Chinese)
(于海英. 字符串相似度度量中 LCS 和 GST 算法比较. *电子科技*, 2011, 24(3): 101-103)
- [29] Wang En, Yang Yong-Jian, Zhao Wei-Dan, Liu Lin-Lu. Packet-based routing algorithm in DTN based on the intimacy between nodes. *Journal on Communications*, 2014, (12): 70-77(in Chinese)
(王恩, 杨永健, 赵卫丹, 刘林璐. 容迟网络中基于节点间亲密度的分组路由方法. *通信学报*, 2014, (12): 70-77)
- [30] Zhao Guang-Song, Chen Ming. Congestion control mechanism based on accepting threshold in delay tolerant networks. *Journal of Software*, 2013, 24(1): 153-163(in Chinese)
(赵广松, 陈鸣. 基于接收阈值的容延网络拥塞控制机制. *软件学报*, 2013, 24(1): 153-163)
- [31] Lee K, Hong S, Kim S J. Slaw: A new mobility model for human walk//*Proceedings of the INFOCOM 2009(INFOCOM'09)*. Rio de Janeiro, Brazil, 2009:855-863.



WANG En, born in 1987, Ph.D. candidate. His current research interests include delay tolerant network and social network.

YANG Yong-Jian, born in 1960, professor and Ph.D. supervisor. His current research interests include wireless communication and network intelligence.

LI Li, born in 1988, master candidate. His current research interest include delay tolerant network and wireless sensor network.

Background

Delay tolerant networks (DTNs) are such challenged networks in which end-to-end transmission latency may be arbitrarily long due to occasionally-connected links. Examples of such networks are those operating in mobile or extreme

scenarios such as interplanetary networks, military networks, rural area networks, wildlife tracking sensor networks, pocket switched networks (PSNs), etc. Some of the existing routing protocols in DTNs focus on clustering nodes based on social

attributes, some focus on contact possibilities between nodes, others focus on choosing a proper next hop. However, there are three defects in the majority of the existing routing protocols on the whole. (1) Conditions such as the number of contacts and the length of communication time which clustering bases on change a lot, resulting in frequent changes of cluster members and high maintenance overhead. (2) Most of the existing prediction-based routing protocols focus on the prediction of whether two nodes would have a contact without considering when the contact happens, which leads to contact prediction inaccurate. (3) Routing protocols can only determine the next hop of the current node, but can't guide multi-hop routing. The main contributions of this article are as follows: (1) Considering the identity of people on campus doesn't change and the node's social role determines its daily route that is relatively fixed, a clustering method is proposed according to path similarity, which can reduce cluster changes and maintenance overhead. (2) We have improved a node mobility model and applied semi-Markov process to predict contact probabilities between nodes and their time which is beneficial to choosing a better next hop. For instance, a former routing protocol has predicted successfully that there are two nodes with a high contact probability (maybe 0.8 or higher). However, this contact may

occur after quite a long time since then, which causes a large amount of delivery latency. This problem can be avoided by applying semi-Markov process in this paper. (3) A delivery path to destination node is found and time-related dynamic tables are generated by executing path-finding strategy so that we can overcome the limitations that only the next hop can be determined in most routing protocols and guide as many hops as possible. In this paper, we propose a clustering method based on semi-Markov process and path-finding strategy CRSMP, which firstly clusters nodes according to path similarity, then predicts the maximum contact probabilities within a certain period of time in the future and the corresponding time based on semi-Markov process and determines a collection of routing nodes on the basis of cluster or clusters containing the source node and the destination node, and lastly obtains dynamic routing tables by executing path-finding strategy and forwards packets according to the routing tables.

This research is partly supported by the National Nature Science Foundation of China under Grant Nos. 61272412 and Jilin Province Science and Technology Development Program under Grant Nos. 20120303.