第43卷 第6A期
2016年6月

计 算 机 科 学
Computer Science

Vol. 43 No. 6A
June 2016

DTN中基于时空和社会性的概率路由算法

贾建鑫 刘广钟 徐 明
(上海海事大学信息工程学院 上海 201306)

摘 要 针对提高传递命中率、减少网络传输延时和网络开销,提出了基于时空和社会性的概率路由算法,称之为GTSP(Geographical area AND Time Combine Social And Probability)。该算法首先根据节点之间经常以大概率相遇的时间段和地理区域确定每个节点以及节点之间在特定时间段和地理区域上的相遇概率较大的几个节点组成的表和共同朋友节点表;然后节点根据GTSP算法进行移动和转发数据包,从而避免了节点在错误的时间段和地理区域内的移动。仿真结果表明,与Prophet路由算法、SprayAndWait路由算法以及SimBet路由算法相比,所提路由算法在平均传输时延、传递命中率、网络开销方面取得了较大的改善。

关键词 延迟容忍网络,GTSP路由算法,时间段,地理区域,社会性,概率
中图法分类号 TP393 **文献标识码** A

Probability Routing Algorithm in DTN Based on Time and Space and Sociality

JIA Jian-xin LIU Guang-zhong XU Ming
(Department of Information Engineering, Shanghai Maritime University, Shanghai 201306, China)

Abstract Aiming to improve the delivery ratio, reduce the delivery latency and reduce network overhead, probability routing algorithm was proposed based on time and space and social, which is called GTSP. Firstly, according to the time span and geographical area that node always has a big probability to encounter each other, the big encounter probability node table of each node and the common shared friends table between nodes in a specific time span and geographical area are ensured. Then the node uses GTSP routing algorithm to motivate and forward data packet, and it avoids the node mobile in erroneous time span and geographical area. Compared with the Prophet routing algorithm, the SprayAndWait routing algorithm and SimBet routing algorithm, the simulation results show that the GTSP has a better performance in delivery latency, delivery ration and network overhead.

Keywords Delay tolerant network, GTSP routing algorithm, Time span, Geographical area, Sociality probability

1 引言

近些年来,无线网络的发展显著地促进了人们对容忍延迟网络(Delay Tolerant Network)的研究[1]。容忍延迟网络是一类新型的网络,它有着一些不同于传统网络的特点。现实中存在许多类型的容忍延迟网络,比如校园区域内、社区中、战场环境中、宇宙空间中、深海环境中等等。

传统的传染路由(Epidemic Routing)算法[2]是在DTN中进行有效路由的最简单的途径,当两个节点相遇时,它们交换自身所有的数据包的信息并且将自身没有的数据包复制到自身的存储器中,这就造成了比较大的网络传输开销。概率路由算法[3]通过节点间过去的相遇记录来预测一个节点将数据包传递到目的节点的交付能力,数据包经常会转发给有更高交付能力的节点。尽管概率路由算法避免了传染路由算法的泛洪式传播方式,但它并没有考虑到时间和地理位置这两个因素,所以在传输延时和交付成功率上还不是最优的。由于人类携带的移动设备通常与特定的社会关系相关,基于社会性的路由算法[4]随即被提出。这种算法将经常联系的节点组成群落,并且选择与目的节点有相似度的节点进行数据包的转发。本质上,社会性路由算法与概率路由算法有一定的相似之处,但是它们都忽略了两个重要的因素:时间和地点。为了克服这些不足之处,本文提出了GTSP路由算法,即每个节点根据节点间的共同朋友表以及每个节点的相遇概率最大的前$m(m\geqslant 1)$个节点表在特定的时间段和地理区域上进行数据包的转发。其中节点间的共同朋友表由两个节点在某个时间段在某个地理区域上相遇概率最大的前m个节点表中相同的节点组成。而节点的概率表由节点在某个时间段在某个地理区域上相遇概率最大的前m个节点组成。GTSP路由算法的设计想法来自于日常生活中社会网络的特点,比如在校园中,我们经常会在某些时间段在某些地理区域中遇见某些同学,这些同学中有我们认识的,也有我们不认识的,原因可能是这些同学的日常作息时间和我们相同,也可能是这些同学的爱好及习惯和我们相同。时间段体现出了作息规律,地理区域体现出了兴趣爱好。根据大概率事件计算出的共同朋

本文受国家自然科学基金项目(61202370),上海市教委科研创新项目(14YZ110),中国博士后科学基金资助项目(2014M561512)资助。
贾建鑫(1988—),男,硕士生,主要研究方向为容忍延迟网络、水下声传感器网络,E-mail:jmakg23@163.com;刘广钟(1962—),男,教授,博士生导师,主要研究方向为分布式数据库、分布式人工智能、计算机网络技术、网格计算、CIMS技术、物流信息化技术等,E-mail:gzhliu@shmtu.edu.cn;徐 明(1977—),男,博士,副教授,主要研究方向为道路网络、智能信息处理、水声传感器网络等,E-mail:mingxu@shmtu.edu.cn。

友节点，它们在数据转发的过程中体现出了很重要的作用。如果能大体确定某些节点出现的时空也就是时间和地点这两个因素，那么在相应的时间段到相应的地理区域上移动便会以很大的概率与某些节点相遇，进而将数据包转发给它们。通过这种方式来进一步减少传输延时并提高交付成功率。

本文第 2 节主要针对相关路由算法进行简单的介绍；第 3 节详细介绍 GTSP 算法的设计过程；第 4 节通过实验仿真模拟对相关的图表进行分析，最后进行总结并对下一步工作进行展望。

2 相关工作

2.1 关于传染路由算法

在传染性路由算法中，比较有代表性的就是传染路由算法(Epidemic Routing，ER)[5]和 Spray And Wait 路由算法[6]。其中 ER 算法的数据传输类似于传染病的感染，每一个节点都需要维护一个摘要向量(Summary Vector)，用来保存将要转发数据包的索引，两个节点相遇以后通过互相交换并比较自身的摘要向量来实现数据包的转发。根据 ER 算法，假设节点 A 的摘要向量为 SV_a，节点 B 的摘要向量为 SV_b。当两个节点相遇时，A 节点首先向 B 节点发送 SV_a，B 节点收到 SV_a 后将其与 SV_b 进行比较，然后生成发往 A 节点的 Request，最后节点 A 将回应信息 Package 返回到 B 节点。

喷射等待路由算法(Spray And Wait)是一种综合了传染路由算法和直接传递路由算法的路由算法。所以，它具有传染路由算法直接传递和快速的特点，是对 ER 算法的一种改进。喷射等待路由算法分为喷射阶段和等待阶段，在喷射阶段，源节点产生将要转发数据的 K 个副本。当源节点遇到其他某个节点时，如果相遇节点没有该数据的副本，那么源节点就将一份副本转发给相遇节点。一直重复进行这个过程，直到源节点将 $K-1$ 个副本都转发给其它相遇节点，自己仅保留一份副本为止。除了源节点外，拥有数据副本的节点为中继节点。在第一阶段采用的是局部传染路由策略或者说受限泛洪策略。下一个阶段为等待阶段，拥有数据副本的节点不再寻找其它中继节点进行转发，而是采用直接传递的路由策略，直到遇到数据副本的目的节点为止，才将数据副本转发给目的节点。

2.2 关于概率路由算法

概率路由算法利用过去的相遇记录来预测未来的交付能力。这些路由算法有 MaxPro、RAPID、PROPHET[7] 等，其中比较有代表性的路由算法就是 PROPHET。在 PROPHET 中，节点的交付能力既要考虑直接相遇概率，又要考虑间接的通过其它节点的传递。概率值的更新基于节点间的相遇和经过的时间单位。假如 A 节点携带着发往目的节点为 D 的数据包。当遇到 B 节点时，根据 PROPHET 算法，当 $P(A,D)>P(B,D)$ 时，A 节点将不会把数据包转发给 B 节点，而是继续携带数据包移动，直到遇到转发概率值更大的节点或者遇到目的节点。当 $P(A,D)<P(B,D)$ 时，A 节点将数据包发送给 B 节点。因为 PROPHET 路由算法的数据包转发策略是根据节点间的转发概率，所以关键在于节点间转发概率的计算。

2.3 关于社会性路由算法

社会性路由算法利用在 DTN 中的社会属性进行数据包的转发。常见的社会性路由算法有 BUBBLE、MOPS、SimBet，这里主要对 SimBet 路由算法[8]进行对比。在 SimBet 路由算法中，每个节点的社会性由节点的介数中心性和相似性组成。SimBet 路由算法的关键是对节点间的相似效用值与介数效用值进行计算。

3 GTSP 路由算法

GTSP 算法的原理：在现实生活中，如果想与某个节点比较快速且准确地以较大可能性相遇，实际上只需要确定两个因素，即时间和空间。本文提出的 GTSP 路由算法既考虑了时间段，又考虑了地理区域。

3.1 建立节点的相遇节点概率表

如图 1 所示，当选取的实验区域在校园范围内(也可以应用在社区范围内或者城市范围内等)时，将整个校园划分为 8 个地理区域，将每个区域进行编号。划分的原则是根据标志性建筑物，比如以图书馆为位置中心的地理区域 1，以健身房为中心的地理区域 2，…，以宿舍为中心的地理区域 8。这种划分思想源于日常生活，在某个时间段，在某个地理区域内总会与一些人经常相遇，这可能是我们之间的作息时间相同，也有可能是我们的兴趣爱好相同。比如，总在图书馆或者图书馆附近遇见某些人，那说明可能我们都比较喜欢学习，更精确一点来说，假如我经常在上午 8 点到 10 点间在图书馆或者图书馆附近的周边区域遇见某些同学，这反映出了我们在上午的作息时间相同，也反映出了我们都比较喜欢读书；再比如说，我经常在体育馆或者附近区域遇见某些同学，那就说明这些同学都比较喜欢运动。在 8 个地理区域的每个地理区域设置一个观测点，warm up 的统计时间为 30 天。将 30 天内的每天的 24 小时按实际需要划分为 $n(n\geqslant 1)$ 个时间段 $T_1,T_2,T_3,\cdots,T_n$，8 个观测点将分别统计在每天的每个时间段在该区域的节点间的相遇次数和节点的相遇标识序列。

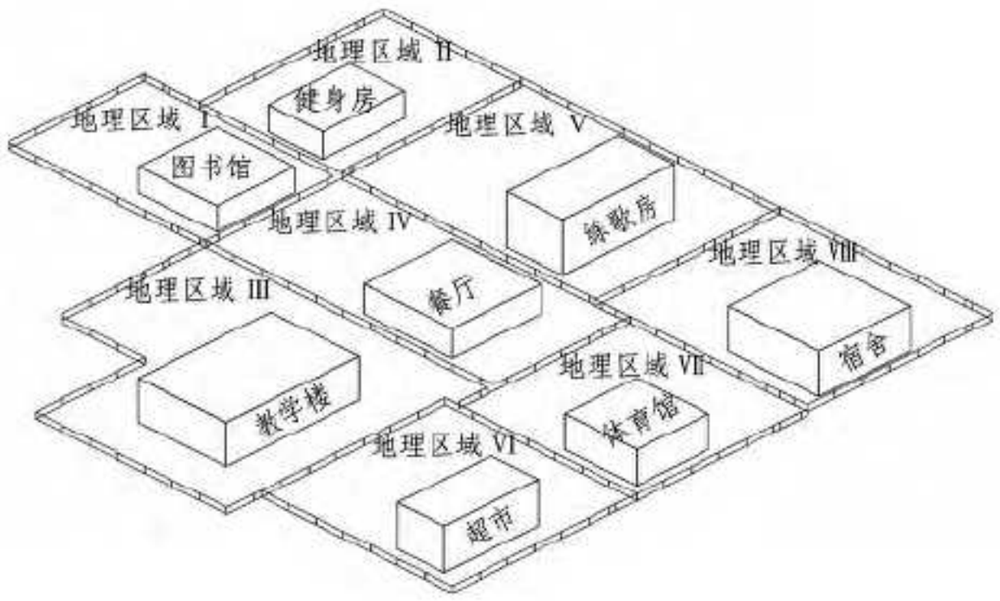

图 1 将校园划分为 8 个地理区域

令 $M(t)$ 表示在时刻 t 节点 a 与某个节点的相遇次数，时间段 T_x 表示时刻 $W(x+1)$ 与时刻 $W(x)$ 之间的差值。如图 2 所示，时刻 $W(1)$ 与时刻 $W(0)$ 之间的差值为时间段 T_1，时刻 $W(n)$ 与时刻 $W(n-1)$ 之间的差值是时间段 T_n。在文中，将每天的 24 小时根据实际需要划分为 n 个时间段，n 的值不能太大也不能太小，一般取合适的值，比如 $n=5$ 或 $n=6(n\geqslant 1,x>0,x$ 与 n 都是整数)。

图 2 时间段划分

相对应地，在时刻 $W(n-1)$ 与节点 a 相遇的某节点的总相遇次数为 $M(n-1)$，在时刻 $W(n)$ 与节点 a 相遇的某节点

的总相遇次数为 $M(n)$。那么在时间段 T_n 内与节点 a 相遇的某节点的总相遇次数为：$count(n)=M(n)-M(n-1)$。所以 a 节点与该节点在时间段 T_n 内以及某地理区域上的相遇概率为：

$$P\{M(n)-M(n-1)=count(n)\}=\frac{[\lambda(W(n)-W(n-1))]^{count(n)}}{count(n)m}e^{-\lambda(W(n)-W(n-1))}$$

假如校园内共有 s 个节点，每个节点有一个唯一的 ID 号。在时间段 $T_x(x>0,x$ 是整数$)$ 内，对于从节点 a 开始的每一个节点都做同样的工作。根据概率公式，计算出时间段 T_x 内某个地理区域上与节点 a 相遇概率最大的前 m 个节点，其中 $P_1>P_2>P_3>\cdots>P_m(m\geqslant 1,m$ 是整数，m 的设置不能太大也不能太小，要适中设计，这里取 $m=5)$，从而得出与节点 a 相遇的节点中相遇概率最大的前 m 个节点的概率表，如表 1 所列。

表 1　节点 a 的相遇概率表

时间段 T_x	对应的相遇节点	对应的相遇概率
a 与 b 相遇	b	$P_1=90\%$
a 与 c 相遇	c	$P_2=85\%$
a 与 d 相遇	d	$P_3=80\%$
a 与 l 相遇	l	$P_4=75\%$
a 与 r 相遇	r	$P_5=70\%$

同样，对于节点 b 来说，在时间段 $T_x(x>0,x$ 是整数$)$，根据概率公式计算出在某个地理区域上与节点 b 相遇概率最大的前 5 个节点，如表 2 所列。

表 2　节点 b 的相遇概率表

时间段 T_x	对应的相遇节点	对应的相遇概率
b 与 d 相遇	d	$P_1=82\%$
b 与 c 相遇	c	$P_2=76\%$
b 与 u 相遇	u	$P_3=70\%$
b 与 r 相遇	r	$P_4=65\%$
b 与 l 相遇	l	$P_5=62\%$

同样，对于剩下的 $s-2$ 个节点，也根据概率公式计算出每一个节点在时间段 T_x 内与它相遇概率最大的前 $m($假设取 $m=5)$ 个节点。每个节点都将它自身的节点相遇概率表放在门卫节点中存储，其中门卫节点是一个地理位置固定的不移动的节点，其它 $s-1$ 个节点都移动到门卫节点的通信区域范围内，得到除自身以外每个节点的相遇概率表的副本。

3.2　建立节点的相遇节点序列以及地理区域标识序列

对于 30 天内每天划分的从 T_1 开始的每个时间段 $T_x(x\leqslant n)$，根据 8 个观测点的统计，得出校园内从节点 a 开始的每个节点在时间段 T_x 内在某个地理区域上与它相遇的节点序列。比如节点 a 在时间段 T_x 内统计出它的相遇节点序列以及相对应的地理区域标识序列分别为：

La(meet)＝b c d e y z d c q l g b c u r…

La(geo)＝III V I VII VI VIII I V VII III II VI V VIII I…

节点 b 在 T_x 内也需要统计它的相遇节点序列以及地理区域标识序列，比如分别为：

Lb(meet)＝f j k p z d v c d l g q r c u…

Lb(geo)＝I II III V VII I III V III III V II II VII VIII …

同理，其余 $s-2$ 个节点也分别在时间段 T_x 内统计出自己的相遇节点序列以及地理区域标识序列。对每一个时间段 T_x，在进行完 30 天的统计后，将每个节点在同一时间段内的相遇节点序列进行合并，比如对节点 a 来说，它将这个月内每天的时间段 T_x 内的相遇节点序列进行合并，使相遇节点序列不断增长，相对应的地理区域标识序列也进行合并。之后，每个节点将自身的相遇节点序列以及地理区域标识序列在门卫节点中存储，其它 $s-1$ 个节点得到除自身外的其余每个节点的节点相遇序列以及地理区域标识序列信息。对每个节点来说，将其存储的 s 个节点的每个节点的相遇节点序列按照字母表顺序进行排序，自然地，地理区域标识序列会与相遇节点序列一一对应。例如：

La(meet)＝b b c c c d d e g l q r u y z…

La(geo)＝III VI V V V I I VII II III VII IVIII VI VIII III…

Lb(meet)＝c c d d f g j k l p q r u v z…

Lb(geo)＝V VII I III I V II III III V II VIII VI III III…

3.3　节点的最长相似序列(Longest similar node sequence)

每一个节点都保存着其余节点在时间段 T_x 内的相遇节点序列以及地理区域标识序列。从节点 a 开始，求它与其它 $s-1$ 个节点的最长相似序列，比如对节点 a 来说，通过算法 1 求出它与节点 b 的最长相似序列。

算法 1

```
Pro LONGESTSIMILARSEQUENCE(Sqa,Sqb)
首先分别对 Sqa 中和 Sqb 中的节点标识按字母序列排序
  lena←length [Sqa]
  lenb←length [Sqb]
  for i←1 to lena
   do ct [i,0] ←0
  end for
  for j←1 to lenb
   do ct [0,j] ←0
  end for
  for i←1 to lena
  do for j←1 to lenb
    do if Sqa(i)=Sqb(j)
       then ct [i,j] ←ct[i−1,j−1]+1
         arrow [i,j] ←"↖"
        else if ct [i−1,j]>=ct [i,j−1]
           then ct [i,j] ←ct[i−1,j]
             arrow [i,j] ←"↑"
           else ct [i,j] ←ct[i,j−1]
             arrow [i,j] ←"←"
        end if
      end if
    end for
  return ct and arrow
  end for
End pro
```

根据算法 1 运行程序得到表 3 所列结果。

表 3　算法 1 运行结果

LSNS(ab)		0	1	2	3	4	…
		Sqb(j)	c	c	d	d	…
0	Sqa(i)	0	0	0	0	0	…
1	b	0	↑0	↑0	↑0	↑0	…
2	b	0	↑0	↑0	↑0	↑0	…
3	c	0	↖1	↖1	←1	←1	…
4	c	0	↖1	↖2	←2	←2	…
5	c	0	↖1	↖2	↑2	↑2	…
6	d	0	↑1	↑2	↖3	↖3	…
…	…	…	…	…	…	…	…

程序的输入参数是节点 a 在时间段 T_x 内的相遇节点序列 Sqa 和节点 b 在时间段 T_x 内的相遇节点序列 Sqb。Lena 是相遇节点序列 Sqa 的长度，Lenb 是相遇节点序列 Sqb 的长度。从节点序列 Sqa 的第一个节点标识开始依次将它与节点序列 Sqb 的每一个节点标识进行比较，如果相等，那么 ct 表元素 ct$[i,j]$的值为其左上方对角线方向的单元格里的值加 1，arrow 表 arrow$[i,j]$的值为“↖”；如果节点标识不相等，同时 ct$[i,j]$上方单元格里的元素值大于等于左方单元格里的元素值，那么取上方单元格里的值，这时 arrow$[i,j]$的值为“↑”，否则取左方单元格里的值，arrow$[i,j]$的值为“←”，最终程序返回的是表 ct 和表 arrow。对于返回的表，从表的最右下角最大的值开始，沿着“↖”一直寻找到表的第一行，arrow 表元素为“↖”所对应的元素就是最长相似节点序列中的元素。

对于节点 a 中存储的所有 s 个节点在时间段 T_x 内的相遇节点序列，从 Sqa 与 Sqb 开始依次运行此算法，Sqa 与 Sqc，Sqa 与 Sqd，…。产生节点 a 与其余 $s-1$ 个节点每一个节点相遇节点序列的最长相似序列(Longest similar node sequence)LSNS。节点 a 需要将它与节点 b 的最长相似节点序列 LSNS(ab)一直到它与第 s 个节点的 LSNS 存储在固定的不移动的门卫节点中。此时门卫节点中共 $s-1$ 个 LSNS。对于节点 b 中来说，它在 warm up 阶段也存储了 s 个节点每一个节点在时间段 T_x 内的相遇节点序列。它需要求出与从节点 c 开始除了与节点 a 以外其余 $s-2$ 个节点的最长相似节点序列，即 LSNS(bc)，LSNS(bd)，…，节点 b 需要将它与节点 c 开始的一直到第 s 个节点的 LSNS 放在门卫节点中。这时门卫节点中共存储 $(s-1)+(s-2)$ 个 LSNS。节点 b 从门卫节点中得到它与 a 节点的 LSNS，因为之前 a 已经将其存储在门卫节点中。依次地，对于节点 c 来说，它需要求出从节点 d 开始的其它 $s-3$ 个节点的每一个 LSNS，节点 c 需要将 $s-3$ 个 LSNS 放在门卫节点中存储。对于节点 c 来说，它与节点 a 的 LSNS 以及与节点 b 的 LSNS 都从门卫节点中得到。依次地，对于第 $s-1$ 个节点来说，它只需要求出与第 s 个节点之间的 LSNS，并将这个 LSNS 存储在门卫节点中。第 $s-1$ 个节点需要从门卫节点得到 $s-2$ 个 LSNS，节点 s 需要从门卫节点得到 $s-1$ 个 LSNS。这样，节点 a 需要运行此程序 $s-1$ 次，节点 b 需要运行 $s-2$ 次，节点 c 需要运行 $s-3$ 次，…，第 $s-1$ 个节点需要运行 1 次。相对于当今时代的移动节点，节点的处理器能很轻松快速地进行处理。而门卫节点共存储 $(s-1)+(s-2)+(s-3)+\cdots+3+2+1=s+(s^2-s)/2$ 个 LSNS。由于门卫节点的地理位置是固定的，并且它的存储器比较充足，对当今硬件的发展来说，存储容量不成问题。

以节点 a 与节点 b 的 LSNS 举例，运行完以上的算法后 LSNS(ab)=c c d d g q r u z…，这个最长相似节点序列需要进一步优化。对于 LSNS 中出现的每一个节点标识，检查它在节点 a 的相遇节点序列中的地理区域标识与它在节点 b 的相遇节点序列的地理区域标识是否相同，如果不同，就将它从 LSNS(ab)中删除，否则保留。体现出的意思是，对于 LSNS(ab)中的第一个节点标识 c 来说，它在 Sqa 中对应的地理区域为 Ⅴ，在 Sqb 中对应的地理区域也为 Ⅴ，这样它就保留在 LSNS(ab)中；对于第二个节点标识 c 来说，它在 Sqa 中的地理区域为 Ⅵ，在 Sqb 中的地理区域为 Ⅶ，所以将它从 LSNS 中删除。因为对于第二个节点标识 c 来说，虽然它在时间段 T_x 内与节点 a 和节点 b 都相遇，但它在地理区域 Ⅵ 与节点 a 相遇，在地理区域 Ⅶ 与节点 b 相遇。这样最终得到优化过的 LSNS(ab)，(Optimal Longest Probability Similar Node Sequence)，简称 OLSNS，相应的优化算法为算法 2。

算法 2

```
Pro OPTIMALLSNS(LSNS(a,b),La(geo),Lb(geo))
    for LSNS 中的每一个节点标识 i
      if (i.La(geo)=i.Lb(geo))
      then i 保留在 LSNS 中
        else i 从 LSNS 中删除
      end if
    end for
    return 优化过的 LSNS(a,b) 以及对应的地理区域标识序列
End pro
```

经过优化的节点 a 与节点 b 之间的最长相似节点序列 OLSNS(a,b)=c d l r u z…，对应的地理位置序列为 Ⅴ Ⅰ Ⅲ Ⅷ Ⅵ Ⅲ…。到这一步，已经可以知道在某个时间段，在某个地理区域上，节点 a 与节点 b 都能在某个地理区域上相遇的节点，这些节点可以作为它们之间传递数据包的公共中介节点。但是以多大的概率与它们相遇呢？比如，对于 OLSNS(a,b)中的第一个节点标识 c 来说，a 与 b 都能在地理区域 Ⅴ 与它相遇，但是如果 a 与 c 以及 b 与 c 的相遇概率非常小，那么这个节点标识 c 以及它的地理区域标识 Ⅴ 并没有什么意义。所以，需要进行下一步的设计。

3.4 最长概率相似节点序列(Longest Probability similar node sequence)以及节点的转发表

之前已经计算出在时间段 T_x 内与节点 a 相遇概率最大的前 m 个节点序列 Max Probability Node Sequence(MPNS)、相应的概率 $P_1(a),P_2(a),\cdots,P_m(a)$，以及每一个概率值对应节点的相对应的地理区域标识。在 warm up 阶段，每个节点已经从门卫节点处得到其余 $s-1$ 个节点的 MPNS。

例如对节点 a 来说 MPNS(a)=b c d l r…，对于节点 b 来说 MPNS(b)=d c u r ……，将 MPNS(a)与 MPNS(b)作为输入参数，根据算法 1 运行程序 LONGESTSIMILARSEQUENCE(MPNS(a)，MPNS(b))，结果返回的是 LPSNS(Longest Probability Similar Node Sequence)最长概率相似节点序列 LPSNS(a,b)=c d l r…，对应的地理区域序列为 Ⅲ Ⅰ Ⅲ Ⅷ…。

然后根据算法 2 运行 OPTIMALLSNS(LPSNS(a,b))进行优化，得到 OLPSNS(Optimal Longest Probability Similar Node Sequence)。即在 OLPSNS(a,b)中的每一个节点标识的地理区域标识要与在 MPNS(a)以及 MPNS(b)中的地理区域标识相一致，优化完的 OLPSNS(a,b)=c d l r…，以及对应的地理位置序列为 Ⅲ Ⅰ Ⅲ Ⅷ…。这个序列代表的是在时间段 T_x 内，在某个地理区域，节点 a 和节点 b 都能以较大概率共同相遇的节点序列，结果会发现这 4 个节点出现在了 OLSNS 中，并且这 4 个节点与节点 a 和节点 b 相遇的概率比较大。例如对于 OLPSNS 中的第一个节点 c，a 节点在时间段 T_x 内在位置 Ⅲ 以 $P_2=85\%$遇见 c，节点 b 以 $P_2=76\%$的概率遇见 c，并且它们都在 $P_1\sim P_m$ 范围内，m 的值一般不会

太大，这里我们取 $m=5$。节点 a 和节点 b 都能以较大的概率在位置 III 与它相遇。由于当节点 a 携带着转发给节点 b 的数据包时，节点 c、节点 d、节点 l、节点 r 是节点 a 和节点 b 在每天的时间段 T_x 内在地理区域 III I III VIII 以较大概率会与其相遇的节点。同理，节点 a 也会依次求出与 c、与 d…与第 s 个节点的 OLPSNS。可以根据 a 的转发表看出 OLPSNS(a,c)=e,f,OLPSNS(a,d)=l,o,i,OLPSNS(a,e)=null,OLPSNS(a,e)=null 是因为节点 e 没有在该时间段内出现。继续依次求出 OLPSNS(a,f),OLPSNS(a,g),…,组成节点 a 的转发表，如表 4 所列。

表 4　节点 a 的转发表

时间段 T_x	最终的节点以及对应的概率值	对应的地理区域
OLPSNS(a,b)	c,d,l,r	III I III VIII
OLPSNS(a,c)	e,f,	IX I
OLPSNS(a,d)	l,o,i,	V II III
OLPSNS(a,e)	null	null
…	…	…

节点 a 将它的转发表放到门卫节点中存储，相应的节点 b 求出自己的转发表，OLPSNS(b,c)，OLPSNS(b,d)，OLPSNS(b,e)…，节点 b 将转发表放到门卫节点中，并从门卫节点中得到自己表中没有的项 OLPSNS(a,b)；同理，其余 $s-2$ 个节点也按此流程进行操作。

到此已经完成了路由设计的思想，如果要与某个节点相遇，只需确定时间和地理区域两个因素。而文中的算法正是通过大概率事件来确定在某个时间段和在某个地理区域与某个节点相遇的可能性，同时确定了时间与空间两个因素。例如，当节点 a 携带着分别发往节点 b，d 的数据包，那么节点 a 只需要查找转发表，找到在时间段 T_x 内与节点 b 和节点 d 的共同朋友节点，以及查找自己的概率表，通过算法 3 进行概率大小比较，然后得出转发节点以及在相应地理区域出现，则节点 a 会以很大的概率碰到转发给 b，d 的共同朋友节点或者直接是碰见 b，d 节点本身，这样极大增加了交付概率，节省了时间，从而避免了节点 a 漫无目的地在任意时刻、任意地理位置随意移动，做无用功，耽误大量时间。再比如节点 a 携带着发给节点 d 的数据包，它或者通过自己的概率表寻找节点，然后将数据包发给它，或者寻找它的转发表中的共同朋友节点，这些共同朋友节点可以作为节点 a 的中介节点继续对数据包进行转发，具体选择方法参见算法 3 的比较。

3.5　GTSP 算法具体设计

通过前面的基础工作以及准备工作，可以得出在节点进行数据包转发时的 GTSP 算法，算法 3 所示是节点 a 将数据包转发给节点 des 时运行的 GTSP 路由算法(取 $m=5$)。

算法 3

```
Pro FORWARD(a 在本时间段 Tx 的概率表,des 离时间段 Tx 最近的时间段 Tk 的概率表)//k>=x
  if(在时间段 Tx 内 OLPSNS(a,des)〈〉null)
    if (a 在时间段 Tx 的概率表内 P1 到 P5 中有对应节点 des 的概率值)
      if(a. P1 对应节点 des)
        then a 在时间段 Tx 到 des 对应的地理区域上将以概率 P1 与节点 des 相遇并将数据包转发给 des
        else (a. Px(x〈〉1 and x>=2 且 x〈〉5)对应节点 des)
        if(a. Px>=a. Pca×des. . Pcdes)
/*Pca 和 Pcdes 分别对应着 OLPSNS(a,des)中的共同朋友节点，这个共同朋友节点是分别在节点 a 和节点 des 的概率表中与节点 a 和节点 des 相遇概率各自最大值的乘积*/
        then a 在时间段 Tx 在概率 Px 所对应节点相应的地理区域上出现，在这节点 a 将以概率 Px 遇见 des 并且将数据包转发给 des
        else 节点 a 会以 Pca×Pcdes 的概率在这个地理区域遇见 des，并将数据包转发给它
        end if
      end if
    else(a 的概率表中 P1 到 P5 都没有对应 des)
    then a 只能通过 OLPSNS(a,des)中的共同朋友节点以 Pca×Pcdes 的相遇概率进行转发
    end if
  else (在时间段 Tx 内 OLPSNS(a,des)=null)
/*说明在时间段 Tx 内，节点 a 与节点 des 没有共同朋友节点*/
    if(a 的概率表中在时间段 Tx 内 P1 到 P5 都没有对应 des)
    then 找到 des 在时间段 Tk 的概率表(k>=x 的第一个值)
    /*时间段 Tk 可能是与 Tx 相等，或者离 Tx 最近的那个时间段，节点 a 在时间段 Tk 移动到 des 概率表 P1 所对应的地理区域，进而节点 des 和 P1 所对应的节点也会以比较大的概率在这个位置与 a 相遇*/
    else(a 的概率表中 P1 到 P5 有对应的 des)
    then 到相应的地理区域会以相应的概率与 des 相遇并将数据包转发给 des
    end if
  end if
End pro
```

GTSP 算法的举例如图 3—图 6 所示。

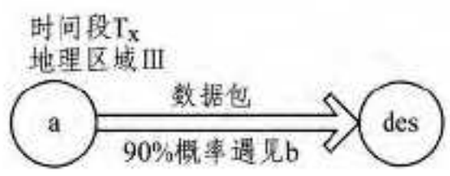

图 3

在图 3 中，在时间段 T_x，当 a 与 des 有共同朋友并且 a 的概率表中有对应节点 des 的概率，当 $a.P_1=90\%$ 对应着节点 des 时，节点 a 会在时间段 T_x 内在地理区域 II 上以 90% 概率遇见 des，并将数据包转发给 des。

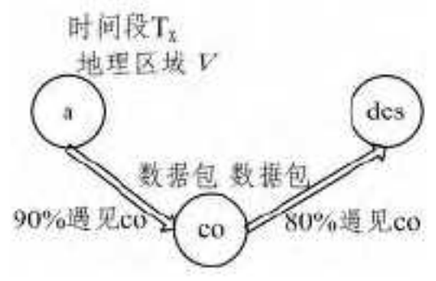

图 4

在图 4 中，在时间段 T_x，a 与 des 有共同朋友并且 a 的概率表中有对应着节点 des 的概率，当 $a.Px(x\langle\rangle1\&\&(x>=2\&\&x<=5))$ 对应着节点 des 时，假设 $a.P3=70\%$ 对应着 des，另一方面 $a.Pca=90\%$，$des.Pcdes=80\%$ 都对应着共同朋友节点 co，因为 $90\%\times80\%=72\%>70\%$，所以在时间段 T_x 到地理区域 V 上移动，选择共同朋友节点 co 进行转发。

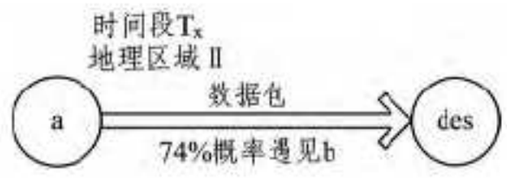

图 5

在图 5 中，在时间段 T_x，a 与 des 在时间段 Tx 没有共同朋友节点，但在 a 的概率表中 P_1 到 P_5 有对应着 des 的概率，假设 P2=74% 对应着节点 des，那么节点 a 在时间段 T_x 在地理区域 II 移动会以 74% 的概率遇见 b，并将数据包转发给 des。

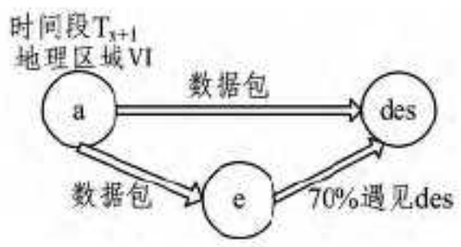

图 6

在图 6 中，在时间段 T_x 内，a 与 des 没有共同朋友节点，并且 a 的概率表没有对应着节点 des 的概率，然后寻找离时间段 T_x 最近的时间段上的 des 的概率表，假如为时间段 T_{x+1}，找到这个表中 P1=70% 所对应的节点，假如为节点 e，那么节点 a 就在时间段 T_{x+1} 到地理区域 VI 上移动，在这个地理区域上移动会以较大的可能性遇到节点 e 或者目的节点 des。

3.6 下一步工作

GTSP 路由算法是针对类似校园的结构，确切的说，应该是类似 local community 的结构，并不针对特别大的区域结构，整个地理范围能太大。实际中可以按照实际需要将其根据标志性建筑物划分为一些地理区域。地理区域的个数不能太多，在总区域面积一定的情况下，地理区域的个数越多，每个区域的面积就越小；如果地理区域太少，每个地理区域的面积过大，统计也会相对不是很精确。同时，本文需要对 3 个变量进行合理的设定：1）时间段的个数；2）地理区域的个数；3）每个节点的与它相遇概率最大的前 m 个节点的个数。同时，算法需要一个优良的处理器和存储器，相信随着科技的发展，当今的处理器的速度和存储器的容量应该能比较轻松地满足需求。下一步将对算法进行优化，以减轻门卫节点的处理器和存储器的负担。

4 GTSP 路由算法仿真分析

本文采用 ONE 仿真平台来对 GTSP 算法进行路由仿真模拟，并且对得出的仿真数据进行比较分析。本文所对比的传统路由算法主要是概率路由算法中的 PROPHET 路由算法、传染路由算法中的 SPRAY AND WAIT 路由算法和社会性路由算法中的 SIMBET 路由算法。通过将 GTSP 路由算法与这 3 种路由算法在数据包传递命中率、平均传输时延和网络开销上进行对比分析，来研究这些路由算法在特定网络环境下的路由性能优劣差异。

目前常用的网络仿真软件有 NS2、OPNET、ONE，其中 ONE(The Opportunistic Network Environment simulator) 是专门用于 DTN 的仿真。ONE 仿真软件是由芬兰的赫尔辛基理工大学网络实验室基于 Java 开发的，主要是基于面向对象的思想，通过实现离散事件驱动，模拟出真实的网络环境。

选取的仿真区域为大学校园，详细的仿真参数如表 5 所列。

表 5 仿真参数

仿真参数	取值
仿真区域	3000m×4000m(校园)
节点个数	600 个
运动模型	Random Waypoint
持续时间	30 天
节点速度	(20,25,30)m/s
节点停滞时间	[0,600](s)
节点的通信半径	30m
节点存储空间	8Gb
数据包产生时间间隔	0.6s
数据包大小	16b

由于 ONE 缺少物理层以及数据链路层，需要的相关信息从仿真软件直接得到，所以仿真结果存在一定的实验误差。仿真通过改变数据包的数量大小，从数据包传递命中率、平均传输延时、网络开销 3 个方面来考察各个算法的性能。在网络中进行通 信的主要目的是正确快速地将数据包传递到目的节点，因此将数据包传递命中率作为网络性能的指标之一，其中数据传递命中率＝传递成功的数据包的数量/源节点产生的数据包的数量。DTN 中数据传输时延和很多因素有关，其中包括节点的移动速度以及网络结构的不断变化。平均传输时延是指所有目的节点接收到数据包的传输时延的平均值。平均传输时延越小，网络的通信效率就越高。网络开销＝网络中已经发出但没有成功传输到目的节点的数据包的数量/已成功传输的数据包数量。对于 DTN 网络来说，网络开销越小，网络的性能越好。

图 7 示出 4 种路由算法的数据包传递命中率。可以看出，随着数据包数量的增加，提出的 GTSP 算法的命中率一直是最高的，因为 GTSP 算法既考虑到了时间段又考虑到了地理区域。

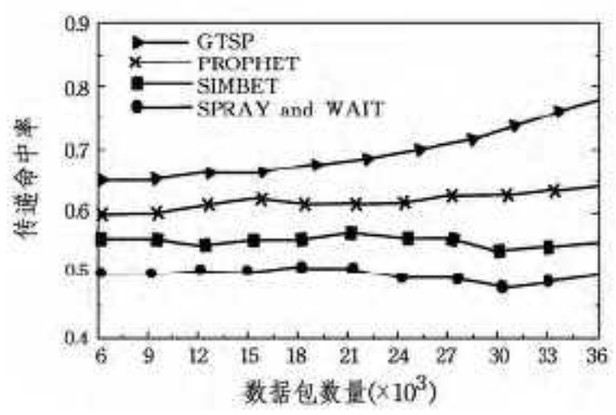

图 7 数据包传递命中率

图 8 示出 4 种路由算法的网络开销对比。由于在开始阶段，GTSP 路由算法会在门卫节点与各个移动节点传递大量的数据信息，因此这一时期的网络开销会比较大，达到一个峰值；但随着算法的继续运行，GTSP 的优势会慢慢体现出来，因为除了初始阶段门卫节点和移动节点需要传递消息外，其它时间移动节点之间基本不需要交换数据信息，所以网络开销会非常低。

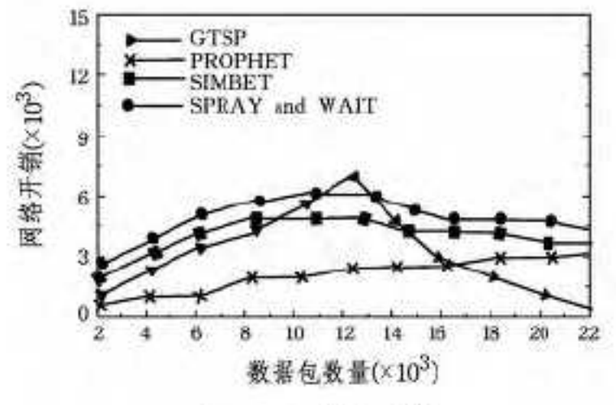

图 8 网络开销

(下转第 309 页)

计算资源动态服务评价模型，并结合用户特征对结果进行了优化。下一步将对本文提出的可用性评估方法和各项指标的确定进行探讨与应用。

参 考 文 献

[1] Varadharajan V, Tupakula U. Security as a Service Model for Cloud Environment[J]. IEEE Transactions on networks and service managment, 2014, 11(1): 60-74

[2] Mustakerov I, Borissova D. A conceptual approach for development of educational Web-based e-testing system[J]. Expert Systems with Applications, 2011, 38(11): 14060-14064

[3] Wu Gang, Wei Yi-min. An Arnoldi—extrapolation algorithm for computing PageRank[J]. Journal of Computational and Applied Mathematics, 2010, 234(11): 3196-3212

[4] Boldi P, Santni M, Vigna S. PageRank as a function of the damping factor[C]// Proceedings of the 14th International Conference on World Wide Web. USA: ACM Press, 2005: 557-566

[5] Klenberg J M. Authoritative sources in a hyper-linked environment[J]. Journal of the ACM, 1999, 46(5): 604-632

[6] Hersovici M, Jacovi M, Maarek Y S. The shark-search algorithm-An application[C]// Proceedings of the 7th International World Wide Web Conference. Brisbane, Australia: ACM Press, 1998: 317-326

[7] Kang Fu-wei, Liu Xiao-dong. A Personalized Ranking Approach via Incorporating Users Click Link Information into PageRank Algorithm[J]. Energy Procedia, 2011, 13: 275-284

[8] Zhai Cheng-xiang. Statistical language models for information retrieval a critical review[J]. Foundations and Trends in Information Retrieval, 2008, 2(3): 137-213

[9] 陈付龙，周雯，王杨，等. 基于 Agent 的云计算资源预选择分层模型[J]. 计算机工程，2009，39(9)：119-122

[10] 王杨，杨娜娜，陈付龙，等. 基于模糊集和 RSS 的云计算资源 Rank 算法[J]. 计算机技术与发展，2013，23(2)：127-130

[11] 余辉，齐丹，李恒达，等. 基于动态基准法的无创血糖检测实验应用研究[J]. 光谱学与光谱分析，2012，32(3)：770-774

[12] 林伯香，孙晶梅，刘起弘，等. 关于动态基准面概念的讨论[J]. 石油物探，2005，44(1)：94-97

(上接第 300 页)

图 9 示出 4 种路由算法的平均传递延时。随着网络中数据包数量的不断增加，GTSP 路由算法体现出了平均延时小的特点。原因是，移动节点对数据包的交付是根据时间和地点来与另一个节点相遇，从而避免了在错误的地理区域和时间段移动，因此节省了时间，降低了平均传递时延。

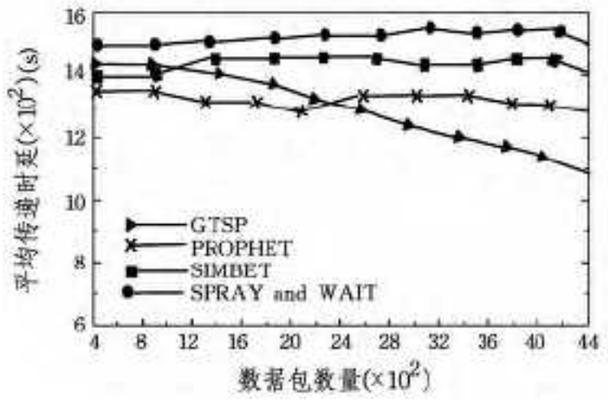

图 9 平均传输时延

结束语 在延迟容忍网络中，传统的路由算法分为传染性路由算法、概率性路由算法以及社会性路由算法，虽然它们各有优势[9]，但它们都没有具体综合考虑时间和空间这两个因素，也就是文中提到的时间段和地理区域两个因素，没有综合考虑时间、地点、社会性和概率性这几方面。本文提出的 GTSP 算法根据节点的概率表和节点间的共同朋友节点在相应的时间段和地理区域上进行数据包转发。在 GTSP 算法中结合了传染路由算法中的洪泛策略、概率路由的特点以及社会性路由的社会性特点。在 GTSP 算法中，门卫节点会利用洪泛策略将存储的相关信息转发给附近的移动节点。在 GTSP 算法中，每个节点会有一个相遇节点概率表，它是根据这些节点相遇概率的大小由大到小排序，每个表项都有对应的相遇节点以及相遇时间段和地理区域，从概率最大的相遇节点开始选出前 m 个节点组成。在 GTSP 算法中，根据社会性的特点，每个节点会建立与其它节点的共同朋友节点，这些共同朋友节点是在某个时间段在某个地理区域进行转发的最佳候选节点。仿真结果表明，GTSP 路由算法与传统的 Spray and Wait 路由算法、Prophet 路由算法以及 SimBet 路由算法相比有更低的平均传输延时和网络开销以及更高的数据包传递命中率。下一步将重点考虑如何进一步减轻门卫节点的存储器和处理器的负担。

参 考 文 献

[1] Jain S, Fall K R, Patra R K. Routing in a delay tolerant network [C]// Proc. SIGCOMM. 2004: 145-158

[2] Vahdat A, Becker D. Epidemic routing for partially-connected ad hoc networks[R]. Duke University, Durham, NC, USA, 2000

[3] Lindgren A, Doria A, Scheln O. Probabilistic routing in intermittently connected networks[J]. Mobile Comput. Commun, Rev, 2003, 7(3): 19-20

[4] Costa P, Mascolo C, Musolesi M, et al. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks [J]. IEEE J. Sel. Areas Commun, 2008, 26(5): 748-760

[5] Jones E, Li L, Schmidtke J, et al. Practical Routing in Delay-Tolerant Networks[J]. IEEE Transactions on Mobile Computing, 2007, 6(8): 943-959

[6] Spyropoulos T, Psounis K, Cauligi S, et al. Efficient routing in intermittently connected mobile networks: themultiple-copy case [J]. IEEE Transactions on Networking, 2008, 16(1): 77-90

[7] Fall K. Delay-tolerant network architecture for challenged internets[C]// Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. ACM, 2003: 27-34

[8] Daly E M, Haahr M. Social network analysis for routing in disconnected delay-tolerant manets[C]// Proceedings of the 8th ACM International Symposium on Mobile Ad hoc Networking and Computing. ACM, 2007: 32-40

[9] 陈元甲. DTN 路由算法的研究与改进[D]. 长沙：中南大学，2010