

# 一种基于区域划分的DTN路由算法

韩进<sup>1,2</sup> 石进<sup>3</sup> 任勇军<sup>1,2</sup>

(南京信息工程大学江苏省网络监控工程中心 南京 210044)<sup>1</sup>

(南京信息工程大学计算机与软件学院 南京 210044)<sup>2</sup> (南京大学信息管理学院 南京 210046)<sup>3</sup>

**摘要** 在DTN网络组成的无向图中,随机运动节点的运动范围会在一段时间内局限在某个连通度高的区域,因此在该时间段内节点应优先处理目标接收节点与其同区域的消息,而其离开该区域时,应优先交互接收节点在其它区域的消息。由此,提出基于区域划分的DTN路由算法。该算法使用随机实验的方法实现对DTN网络无向图的区域划分,并以节点自身所在区域及消息接收节点所在区域作为消息中转的依据实现DTN消息的路由。实验结果表明,该算法与PRoPHET、Epidemic、SAW相比,能达到较高的消息传输率,并有效降低消息副本的传输与复制。

**关键词** DTN路由,区域划分,区域位置判断

**中图分类号** TP309.2 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.10.024

## DTN Routing Algorithm Based on Region Segmentation

HAN Jin<sup>1,2</sup> SHI Jin<sup>3</sup> REN Yong-jun<sup>1,2</sup>

(Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing 210044, China)<sup>1</sup>

(College of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China)<sup>2</sup>

(School of Information Management, Nanjing University, Nanjing 210046, China)<sup>3</sup>

**Abstract** The randomly moving DTN nodes are often trapped for a period in highly connectivity regions of a undirected graph, which consists of DTN's paths. So in the period, the messages whose receiver is in the same region with its sender in current local region should be exchanged first. Meanwhile, the messages whose receiver is in different regions with its sender nodes in current local region should be exchanged first when the nodes are leaving the message sender's current local region. According to this strategy, a new DTN routing algorithm was presented in this paper. In the algorithm, a undirected graph is segmented into regions by random experiment method, and DTN's messages are routed according to a node's currently local region and the currently local region of receivers of messages. The results of experiments show that compared to PRoPHET, Epidemic, SAW, the algorithm can get relatively high message delivery successful rate and effectively reduce message copies transit times.

**Keywords** DTN routing, Region segmentation, Region location judgment

## 1 引言

容延/容断网络(Delay /Disruption-Tolerant Networks, DTN)是一种由多移动节点组成的自组织网络。DTN与传统的移动自组织网络(Mobile Ad hoc Networks, MANET)不同,MANET中节点位置和节点之间的路径是不会变化的。与MANET相比,DTN具有高度的移动性、低节点分布密度和节点通讯范围短的特点<sup>[1]</sup>,因此DTN的路由算法对于DTN网络效应的发挥有着重要的影响。

根据对先验知识的掌握程度,DTN的路由算法又分成零知识路由、部分知识路由和全知识路由3类<sup>[2]</sup>。使用零知识路由算法的DTN又称为机会网(Opportunistic Network)。对DTN路由算法研究较多的有Direct Delivery, First Con-

tact, Epidemic, PRoPHET<sup>[3]</sup>, Spray and Wait (SAW), Spray and Focus (SAF), MaxProp<sup>[4]</sup>等。其中:Epidemic中节点会在移动过程中将所有消息都发送至相遇节点。这种洪泛方式虽然会一定程度上提高消息传输率,但消息的大量转发会消耗掉有限的系统资源,反而降低了消息传输率。为了限制消息副本复制数量,出现了分阶段的传输算法,如:Spray and Wait, Spray and Focus等。

为了能进一步提高消息的传输率同时降低消息路由带来的消耗,又出现了根据接收概率评估来选择消息接收节点的PRoPHET算法,或依据接收代价评估选择传递消息的Max-Prop算法。它们将概率和代价估算作为消息中转的决策依据,在保证一定的传输率下,有效地减少了消息副本的复制与传输率,但是估算有偏差。由此针对不同的DTN应用场景,

到稿日期:2014-10-31 返修日期:2015-02-09 本文受江苏高校优势学科建设工程资助项目,国家自然科学基金:基于随机化的内核完整性保护技术研究(61100197),2014年校教改提升工程(N1885014041),2012年校教改提升工程项目(N1885012119)资助。

韩进(1974-),男,博士,讲师,主要研究方向为信息安全、Ad hoc网络,E-mail:hjhaohj@126.com;石进(1976-),男,博士,副教授,主要研究方向为信息安全;任勇军(1974-),男,博士,主要研究方向为信息安全。

利用先验知识设计 DTN 路由算法就成为提高 DTN 路由效率的重要途径。

与现有利用先验知识的 DTN 路由算法不同,本文的算法将 DTN 路径组成的无向图根据各部分的连通度划分成不同的区域,使得 DTN 节点在高连通度区域内运动的时间远多于区域间运动的时间,再结合节点自身所在的区域以及消息接收节点所在区域的判断,以此作为 DTN 消息选择路由的决策依据。

与本文类似的研究工作<sup>[5]</sup>虽然也提到关于域的概念,但其所指的域划分是以节点为中心、以预定的节点跳数为域的半径,与本文所述的区域划分概念不同。而文献<sup>[6]</sup>也利用了地理信息来作为路由的决策,主要用于车辆作为节点的 DTN 网络中,通过计算最小传递估算时间来作为路由决策,并混合了单消息拷贝与多消息拷贝的机制,因此与本文的算法本质上还是有区别的。

## 2 基于区域划分的 DTN 路由算法

### 2.1 算法基本思想

大多零知识或部分先验知识 DTN 网络中节点都随机选择目的地,然后沿着预定的路径前进。DTN 网络中路径可形成无向图,如图 1 所示。

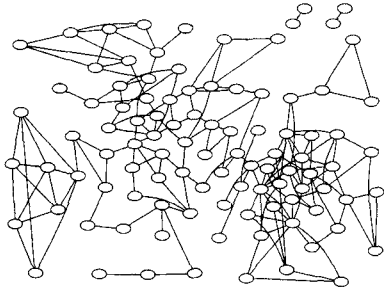


图 1 The ONE 提供的邻接图

从图 1 中所示的无向图可以很清楚地看出,按节点彼此邻接的紧密程度可以将图划分成几个不同的区域,即 D1, D2, D3, 以及除了这 3 个区域之外的第四部分 D4, 如图 2 所示。由图 2 可知,若当前节点在 D3 区域随机运动,那么它只有一条路径可以从 D3 区域脱离,进入到 D4 区域中,相反也是如此。因此按如上的区域划分,节点在某个区域内移动的时间会远远超过它在区域之间的移动时间。

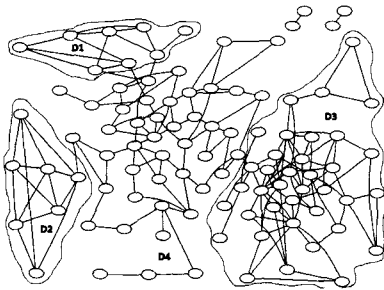


图 2 邻接图的划分

由此本文基于区域划分的 DTN 路由算法思想是:

1. 依据无向图的区域划分,在 DTN 消息路由时可以根据消息接收节点所在的区域,将消息划分成区域内和区域外消息。

2. 当节点在区域内移动时,应与其它同区域的节点多交

换接收同区域的消息,因为该节点会有相对大的概率继续在本区域内移动一段时间。

3. 而节点开始区域间移动时,应尽可能多接收其它节点中转的区域外消息,因为该节点能够再回到原区域的概率较小,因此目标接收节点在原区域的消息再通过它完成传递的可能性较小。

### 2.2 基于区域划分的 DTN 路由算法的实现

#### 2.2.1 无向图的区域划分算法

实现上述路由算法首先需要无向图的区域进行划分。设无向图  $G$  为  $(V, E)$  的二元组,其中  $V$  为  $G$  中顶点的集合,  $E$  为  $G$  中边的集合。那么无向图的区域划分可以由式(1)~式(3)表示:

$$G = \{G_1, G_2, \dots, G_n\} \quad (1)$$

$$V(G) = V(G_1) + V(G_2) + \dots + V(G_n) \quad (2)$$

$$E = \{e | e = (v_1, v_2) \cap v_1 \in V(G_i) \cap v_2 \in V(G_j) \cap i \neq j\} \quad (3)$$

其中,  $G_i$  代表划分的区域,  $V(G_i)$  为  $G_i$  区域中包含的节点数也即该区域的阶。  $E$  为划分后不同区域间连接边的集合。

由本文路由算法思想可知,无向图的区域划分并不意味着式(3)中的  $E$  集合的元素个数  $|E|$  应尽可能少。例如,图  $G$  划分成两个区域  $G_1$  和  $G_2$ , 其中  $|V(G_1)| = 1, |V(G_2)| = n-1$ , 虽然这二者之间只有一条边邻接,达到最小,但如此划分显然不符合本文路由算法的初衷。因此,本文路由算法的无向图划分还应满足条件:

$$\delta_{ij} = ||V(G_i) - V(G_j)||, \max(\delta_{ij}) \leq \Delta \cap \min(|E|)$$

其中,  $\delta_{ij}$  为区域  $i$  与区域  $j$  的节点数量差的绝对值,即满足各区域节点数最大差值小于  $\Delta$  时,实现区域间邻接边的数量最小。本文使用随机实验的方式来予以实现。其算法如下所示:

GraphDivide(VLL, TL, N)

1. VL ← [], node, min ← MAX\_Integer
2. Vertices ← [];
3. for i ← 1 to VLL
4. node travels randomly in Graph;
5. VL[i] ← visited vertex
6. Trace[] ← splits VL into traces  $\cap$  length of traces = TL
7. for i ← 1 to length of Trace[]
8.  $N_i$  ← number of vertices in Trace[i]
9.  $V_i$  ← vertices in Trace[i]
10. e ← number of outer edges of  $V_i$
11. if ( $N_i = N \cap e < \min$ ) min ← e, Vertices ←  $V_i$
12. Return Vertices

该算法首先采用随机漫游的方式获取 VLL 长度的无向图顶点访问序列,再把该序列按预定的长度 TL 切分成多个子序列。遍历这些子序列,若某个子序列中包含的无向图顶点数  $N_i$  等于目标划分区域应包含的顶点数 N,且该子序列中包含的点集  $V_i$  的外邻接边数比当前顶点集合 Vertices 的外邻接边数少,则更新最小外邻接边数以及当前顶点集合。最后返回 Vertices。

该算法与本文路由思想是一致的,无向图中顶点连通度高的区域,在随机漫游过程中,有大概率出现一段长的包含该区域顶点的序列。算法先要确定目标划分区域顶点数 N,以满足本文路由算法的无向图划分算法前部分条件,再选择适当的子序列长度 TL,计算各子序列中包含的顶点数以及子序列顶点集的外邻接边数,求出满足具有最小外邻接边数的

点集并返回。由于本算法是求外邻接边数最少的顶点数为  $N$  的子序列,计算的结果是收敛的,因此在反复迭代计算下,会有较大概率发现最优解。本文的实验结果也证明了上述算法的正确性。

### 2.2.2 节点区域位置的判断算法

本文路由算法需要对消息接收者所在区域进行判断,以便将节点内存储的消息分成区域内消息和区域外消息。本文的节点区域位置判断算法如下:

NodesLocationJudgement(Node<sub>i</sub>)

1.  $NL[j] \leftarrow P_j(G) \cap i \neq j, NLTimer \leftarrow [0]$
2. while node<sub>i</sub> Traveling
3. if(meeting with Node<sub>j</sub>)
4.  $NL[j] \leftarrow P_j(G) = 1, NLTimer[j] \leftarrow 0$
5.  $TmpNL \leftarrow$  get Node<sub>j</sub>'s NL
6.  $TmpNLTimer \leftarrow$  get Node<sub>j</sub>'s NLTimer
7. for  $k \leftarrow 1$  to length of TmpNL
8. if  $(NLTimer[k] > TmpNLTimer[k] \cap k \neq i \cap k \neq j)$
9.  $NL[k] \leftarrow TmpNL[k]$
10.  $NLTimer[k] \leftarrow TmpNLTimer[k]$
11. if(current decay period end)
12. for  $k \leftarrow 1$  to length of NL
13.  $NL[k] = \gamma(NL[k])$
14.  $NLTimer[k] ++$

在上述算法中,DTN网络构建时每个节点内初始化有NL列表与NLTimer列表,NL列表为其它节点所在区域的概率; $P_j(G)$ 即Node<sub>j</sub>在域G的概率,初始时随机赋予节点j的 $P_j(G)$ 值。NLTimer列表是节点j的 $P_j(G)$ 的衰减累计数, $\gamma$ 为衰减函数。

当节点i与节点j相遇时,则置NL列表中 $P_j(G)$ 的值为1,其中G为节点i所在的区域,同时重置节点j的 $P_j(G)$ 的衰减累计数为零。再获取节点j的NL列表与NLTimer列表并遍历,若发现第三方节点k在节点j的衰减累计数小于本节点的衰减累计数,则意味着节点j中关于节点k的位置信息较新,以此重置自己NL列表与NLTimer列表中节点k所在区域概率以及衰减累计数。

若当前衰减周期结束,则对NL列表中所有的节点区域概率进行衰减,并累加NLTimer列表中的衰减累计数。值得注意的是,上述算法中通过衰减累计数来判断节点区域信息更新的程度。该数由节点根据自身的DTN网络时间片来进行累加,因此并不需要使用全局同步时钟。

### 2.2.3 消息路由中转策略算法

GrapeDividedRouter(Node<sub>i</sub>)

1.  $G \leftarrow$  the area Node<sub>i</sub> is traveling
2. for  $i \leftarrow 1$  to Node<sub>i</sub>'s Msgs length
3. if  $(!Msgs[i] \in Node_j \text{ 's } Msgs \cap Msgs[i] \text{ 's hops} < HOP\_Limit)$
4. if  $(Node_i \text{ moveout } G \cap Node_j \text{ moveout } G \cap Msgs[i] \in OUTERMSG)$
5. Send Msgs[i] to Node<sub>j</sub>
6. continue;
7. if  $(Node_i \text{ movein } G \cap Node_j \text{ moveout } G \cap Msgs[i] \in OUTERMSG)$
8. Send Msgs[i] to Node<sub>j</sub>
9. continue;
10. if  $(Node_i \text{ movein } G \cap Node_j \text{ movein } G \cap Msgs[i] \in INNERMSG)$
11. Send Msgs[i] to Node<sub>j</sub>
12. continue;

13. if  $(Node_i \text{ moveout } G \cap Node_j \text{ movein } G \cap Msgs[i] \in INNERMSG)$
14. Send Msgs[i] to Node<sub>j</sub>
15. continue;
16. Send Msgs[i] to Node<sub>j</sub> with probability  $\alpha$

在上述算法中,节点Node<sub>i</sub>与Node<sub>j</sub>相遇后,先遍历Node<sub>j</sub>节点中没有且转发跳数小于预先设定的消息转发跳数的消息,针对该消息有选择性地中转。

选择的依据有:在两节点各自移动的方向,以及消息为区域内或区域外,一共有7种组合,上述算法中行4,7,10,13这4种情况下会将该消息直接发送给Node<sub>j</sub>(如行4情况下,当两节点都从当前区域移出时,只需要交互区域外的消息即可;而行10情况下,本节点移出而对方节点移入,因为将本节点掌握的区域内消息发送给对方)。而其它的4种情况,将按预定的概率 $\alpha$ 发送消息至Node<sub>j</sub>,以弥补在区域位置估算时带来的区域定位偏差,本文设置 $\alpha$ 值为0.3。

## 3 实验

### 3.1 区域划分与节点区域位置判断算法实验

本文的区域划分算法实验如下所述。在实验中采用具有10个顶点的无向图来表示DTN网络中的路径,顶点之间的连接概率为0.8,无向图边长度也即DTN的路径长度服从[20,50]的连续均匀分布,如图3所示,其对应的划分如图4所示。

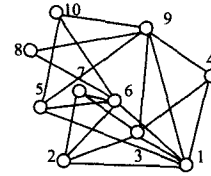


图3 随机生成的DTN无向图

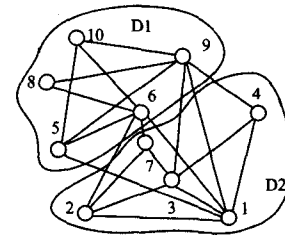


图4 区域划分算法的实验结果

不失一般性,本文将20次随机生成的无向图划分成顶点数各为5的两个区域,设目标划分区域的漫游顶点的序列长度为100000。实验结果表明,本区域划分算法可以发现正确的区域划分,但也发现子序列的长度设置对于实验结果有明显影响,如图5所示。原因在于子序列取得越长,序列中顶点数等于N的概率越小,造成满足算法条件的子序列也就越少,从而增加了划分结果的偶然性。

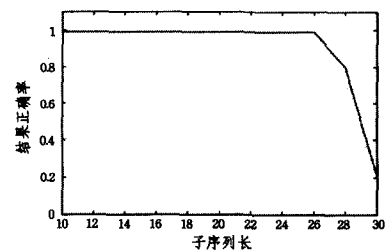


图5 子序列长度对结果的影响

本文的节点区域位置判断算法采用的是半衰期衰减函数： $m = M * (0.9)^{(t/T)}$ ， $t$ 为当前的DTN模拟时间片数， $T$ 为半衰期。在实验中，一共放置了8个随机移动的节点，其移动速度服从 $[1, 3]$ 及 $[3, 6]$ 的连续分布。由图6可知，算法的正确率随着模拟时间片的的增长而增长，这是因为模拟时间长，有利于节点运动中不断地交互节点区域信息，全局节点区域定位信息的精确度会因此上升。当节点步速加快时，区域位置判断算法的准确率会提高，这也是因为交互区域信息的频率相应加快了。

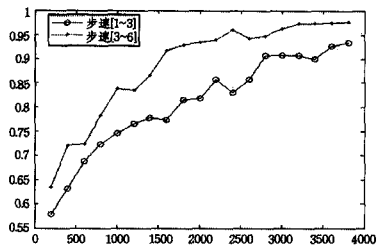


图6 节点区域位置判断算法的实验结果

### 3.2 路由算法对比实验

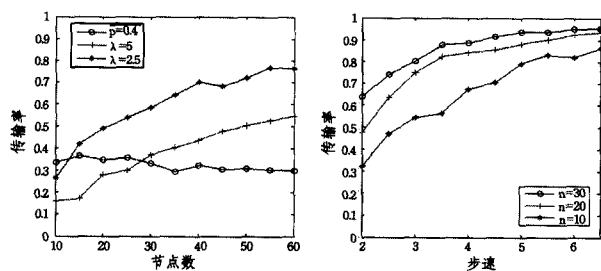
本文的实验以 ONE 模拟器<sup>[7]</sup>为基础进行了二次开发，采用了其中 SAW 及 Epidemic 的 Router 类代码，并对 PRoPHET 算法进行修改，将其节点移动模式改为随机漫游，但仍估算消息传递成功的概率来选择相遇节点中转消息，以便于算法的性能对比。

本文路由算法性能测试的实验结果如图7所示。图7(a)是测试消息数量以及DTN节点数对消息传输率的影响，其中节点步速连续分布范围为 $[1 \sim 3]$ ，节点数为 $[10, 60]$ ，节点内存为128，单个消息大小连续分布范围为 $[2 \sim 12]$ 。

一共测试了两种消息生成模式，一是每个时间片内产生消息数与节点数成比例，图中比例为0.4；另一种每个时间片内产生的消息数为均匀分布，期望分别为2.5和5。从结果来看，第一种模式下传输率基本稳定，并随着节点数增加略有下降，因为内存不变而每个时间片内产生的消息数随着节点数增加而增加，引发内存不足，造成传输率下降。在第二种方式下，随着节点数增加，传输率呈逐步上升的趋势，且时间片内产生消息数量少的传输率高，其原因依然与内存量相关。

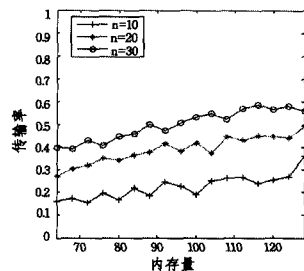
图7(b)所示为步速对传输率的影响。在这次实验中，采用的步速范围为连续均匀分布，期望范围为 $[2 \sim 6.5]$ ，节点内存为128，消息大小范围为 $[1 \sim 3]$ ，消息生成采用第二种模式，每个时间片内产生的消息数期望为2.5，进行3次实验，节点数分别为30, 20, 10。从实验结果来看，步速对于本文算法的传输率影响较大，原因在于一方面节点移动速度加快，造成节点间消息中转频率加快；另一方面即为前述的节点区域位置判断的准确度提升，使得本文算法消息中转策略的有效性也相应提升。

图7(c)所示为测试内存对消息传输率的影响，其中步速为均匀分布，期望为2，消息大小范围为 $[1 \sim 3]$ ，每个时间片内产生的消息数为均匀分布，期望为2.5，进行3次实验，节点数分别为30, 20, 10。从实验结果来看，内存量对于传输率有一定影响，但是影响不是太大。



(a)消息数量对传输率的影响

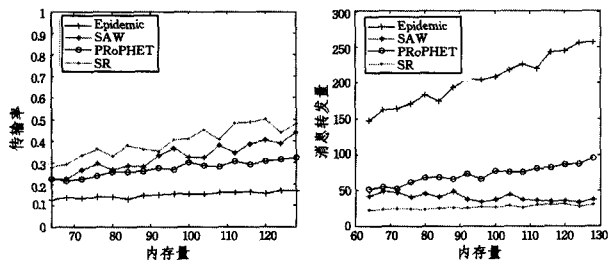
(b)步速对传输率的影响



(c)内存对传输率的影响

图7 基于区域划分的DTN路由算法的性能测试

与 SAW、Epidemic、PRoPHET 算法的对比如图8所示，其中步速为连续均匀分布，期望为2，消息大小范围为 $[1 \sim 3]$ ，采用第二种模式产生消息数，期望为2.5，节点数取20。限于篇幅，本文只给出在不同内存量下各算法的传输率以及消息中转数量的对比。从图8(a)对比可以看出，本文的算法(SR)与其它算法相比传输率是最高的；从总体上来看，本文实验的 SAW、Epidemic、PRoPHET 算法的传输率与文献<sup>[8]</sup>描述的基本一致。



(a)数据传输率对比

(b)转发消息数量对比

图8 与 SAW、Epidemic、PRoPHET 算法的性能对比

图8(b)是各算法在不同的节点内存情况下路由过程中的消息转发的数量。其中 Epidemic 转发的消息量最大，且随着内存增加快速增长。经实验分析得知 Epidemic 算法的感染式消息转发方式会导致节点内存被快速占满，而节点内新的消息会替换掉原消息，因此造成节点内消息的更新速度加快。节点内存增加，存储的消息数量会增长，则相遇时交换的消息量也相应增加，再当节点内存被占满后，泛滥造成消息更新频率提升，从而出现上述现象。图8(a)中 PRoPHET 的传输率几乎没有因为内存增长而得到提升，原因即在于此。

PRoPHET 算法、SAW 和本文算法都通过各自策略抑制了消息副本的复制扩散，所以伴随着内存量的增加，转发的消息数量并没有显著增长。值得注意的是，本文算法是所有对比算法中消息转发数据最少的而且数量相对稳定，原因在于本文算法符合 DTN 无向图的节点运动规律，基于区域的消息中转策略既抑制了消息副本的复制传输，又能有效实现区

(下转第 125 页)

字仍然能实现全分集,即采用 TDMA 传输方案时分集增益为 16。从图 2、图 3 中可看出,TDMA 传输方案的可靠性优于其他两种方案,因为 TDMA 传输方案以传输效率的降低换取了可靠性的提高。这两种方案以分集增益的降低换取了自由度的提高,即以可靠性的降低换取了有效性的提高。从图中还可看出,所提方案的可靠性高于文献[10],因为两种方案的分集增益分别是 8 和 4,从而也验证了理论分析的正确性。

**结束语** 对于每个用户均配置 4 根天线的 X 信道,已有的基于空时编码和干扰对齐的干扰消除方案和所提方案都通过在码字引入零向量、对码字进行预编码以及接收端的线性处理和非线性处理消除了多用户干扰,而所提方案在保持相同自由度的情况下,提高了分集增益。此外,该干扰消除方案并非局限于 R2-STBC,若 X<sub>i</sub> 和 Y<sub>i</sub> 是其他形式的完美空时分组码,也可用类似的方法消除多用户干扰。

### 参 考 文 献

[1] 田心记,倪水平,胡紫巍,等. MIMO-MAC 中改进的空时码传输方案 [J]. 北京邮电大学学报,2013,36(4):95-98  
Tian Xin-ji, Ni Shui-ping, Hu Zi-wei, et al. Improved Space-Time Coded Transmission Scheme for MZMO-MAC [J]. Journal of Beijing University of Posts and Telecommunications, 2013, 36(4):95-98

[2] Gesbert D, Shafi M, Shiu D. From theory to practice: an overview of MIMO space-time coded wireless systems [J]. IEEE Journal on Selected Areas in Communications, 2003, 21(3):281-302

[3] 田心记,李亚. X 信道中新的空时码传输方案 [J]. 北京邮电大学学报,2014,31(7):99-102  
Tian Xin-ji, Li Ya. New Space-Time Coded Transmission

Scheme for X Channel [J]. Journal of Beijing University of Posts and Telecommunications, 2014, 31(7):99-102

[4] Maddah-Ali M, Motahari A, Khandani A. Communication over MIMO X channels: Interference alignment, decomposition, and performance analysis [J]. IEEE Transactions on Information Theory, 2008, 54(8):3457-3470

[5] Shi L, Zhang W, Xia X G. Space-Time Block Code Designs for Two-User MIMO X Channels [J]. IEEE Transactions on Communications, 2013, 61(9):3806-3815

[6] Rao X B, Ruan L Z, Lan K N. CSI feedback reduction for mimo interference alignment [J]. IEEE Transactions on Signal Processing, 2013, 61(18):4428-4437

[7] Xie B, Li Y, Minn H, et al. Adaptive interference alignment with CSI uncertainty [J]. IEEE Transactions on Communications, 2013, 61(2):792-801

[8] Li L B, Jafarkhani H, Jafar S A. When Alamouti codes meet interference alignment: transmission schemes for two-user X channel [C] // IEEE International Symposium on Information Theory (ISIT). 2011:2577-2581

[9] Li F, Jafarkhani H. Space-time processing for X channels using precoders [J]. IEEE Transactions on Signal Processing, 2012, 60(4):1849-1861

[10] Ganesan A, Rajan B S. Interference alignment with diversity for the 2x2 X network with three antennas [C] // IEEE International Symposium on Information Theory. Honolulu, 2014: 1216-1220

[11] Zaki A, Wang C, Rasmussen L K. Combining interference alignment and Alamouti codes for the 3-user mimo interference channel [C] // IEEE Wireless Communications and Networking Conference. Shanghai, China, 2013:3563-3567

(上接第 116 页)

域内与区域间的消息传递。

**结束语** 基于 DTN 网络组成的无向图中,各部分的连通度差异会造成节点在某段时间内局限于某个区域的情况,本文提出一种基于区域划分的 DTN 网络路由算法。首先,本文使用随机实验的方法解决了网络无向图的区域划分问题,再使用估算以及更新定位信息的方式实现全局节点的区域定位判断,以此为依据,实现了消息路由策略。

性能和对比实验表明,本文的消息路由算法能实现较高的消息传输率,同时有效抑制了消息中转的数量。这对于 DTN 网络来说,不仅节约了存储资源,而且减少了消息中转所消耗的电能。

本文的算法属于部分先验知识算法,需要计算 DTN 网络无向图的区域划分,且节点在运动过程中要掌握自身所在的区域,但无需全局同步时钟。与其它先验知识算法相比,本文的先验知识主要是地理信息方面,在大多数应用领域中都可以得到满足,要求较低。后期研究工作可通过节点实时移动位置实现自主区域划分,以进一步减小对先验知识的依赖程度。

### 参 考 文 献

[1] Ali A, Panda M, et al. Improving the transport performance in delay tolerant networks by random linear network coding and

global acknowledgments [J]. Ad hoc Networks, 2013, 11(8): 2567-2587

[2] Su Jin-shu, Hu Qiao-lin, Zhao Bao-kang, et al. Routing Techniques on Delay/Disruption Tolerant Networks [J]. Journal of Software, 2010, 21(1):119-132

[3] Doria L A, et al. Probabilistic routing in intermittently connected networks [J]. ACM SIGMOBILE Mobile Computing and Communications Review, 2003, 7(3):19-20

[4] Burgess J, Gallagher B, et al. MaxProp: Routing for Vehicle Based Disruption Tolerant Networks [C] // INFOCOM. 2006:1-11

[5] Shi K. Semi-Probabilistic Routing in Intermittently Connected Mobile Ad Hoc Networks [J]. J. Inf. Sci. Eng., 2010, 26(5): 1677-1693

[6] Soares V N G J, Rodrigues J J P C, Farahmand F. GeoSpray: A geographic routing protocol for vehicular delay-tolerant networks [J]. Information Fusion, 2014, 15:102-113

[7] Keränen A, Ott J, Kärkkäinen T. The ONE simulator for DTN protocol evaluation [C] // Proceedings of the 2nd International Conference on Simulation Tools and Techniques. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009:55

[8] Sun Jian-zhi, Liu Nai-rui, Zhang Ying-xin, et al. Performance Analysis of Typical Routing Algorithm in Opportunistic Network [J]. Computer Engineering, 2011, 37(16):86-89