

文章编号: 2095-4980(2021)03-0523-05

基于消息传输率的时延容忍网络路由

辛 晨^a, 冯 涛^{*a}, 田志民^b, 崔炳德^a

(河北水利电力学院 a.计算机科学与信息工程学院; b.教务处, 河北 沧州 061001)

摘 要: 针对时延容忍网络(DTNs), 提出基于消息传输率的路由(DTRR)。将每条消息划分若干消息块。当源节点遇见合适的转发节点就传输消息块, 否则就将消息块存于缓存区中。同时, DTRR 路由结合相遇频率、相遇时长和缓存区的可用空间, 推导链路的消息传输率。再依据消息传输率, 选择下一跳转发节点。仿真结果表明, 提出的 DTRR 路由在消息传输率方面的性能优于同类的等待路由(SWR)。

关键词: 时延容忍网络; 路由; 消息块; 相遇时长; 消息传输率

中图分类号: TN914

文献标志码: A

doi: 10.11805/TKYDA2019382

Data transfer rate-based routing for delay-tolerant networks

XIN Chen^a, FENG Tao^{*a}, TIAN Zhimin^b, CUI Bingde^a

(a.School of Computer Science and Information Engineering,

b.Dean's Office, Hebei University of Water Resources and Electric Engineering, Cangzhou Hebei 061001, China)

Abstract: Data Transfer Rate-based Routing(DTRR) is proposed for Delay-Tolerant Networks(DTNs). Each message is divided into several message blocks. When the source node meets the appropriate forwarding node, it transmits the message block, otherwise it stores the message block in the cache. At the same time, the DTRR combines the contact frequency, the duration of the contact and the available space of the cache area to deduce the message transmission rate of the link. The next hop forwarding node is selected according to the message transmission rate. Simulation results show that the proposed DTRR has better performance in message transmission rate than similar Spray-and-Wait Routing(SWR).

Keywords: delay-tolerant networks; routing; message blocks; contact duration; data transfer rate

作为自组织网络的特例, 时延容忍网络(DTNs)具有间歇性连通、稀疏式网络连接特点^[1]。为了应对这些特点, 常采用存储-转发策略传输数据。当未遇见合适的转发节点时, 源节点就将数据存储于缓存区, 等待与转发节点相遇。因此, 如何选择转发节点是实施 DTNs 路由的关键^[2]。此外, 随着移动终端设备容量的增加, DTNs 网络内要传输的数据量越来越大。源节点很难只通过一次与转发节点的相遇^[3], 就把数据转发出去。而多数的传统 DTNs 路由并没有考虑此问题。将这种需要传输大量数据的路由称为数据密集路由。现存的 DTNs 路由主要存在以下需要解决的问题。

首先, 现存的 DTNs 路由通常假定: 每次相遇的时间够长, 足以完成一个节点向另一个节点转发消息。换言之, 一对节点间的路由性能只受它们的相遇频率影响。只是简单地将链路权重量化为相遇频率的函数^[4]。然而, 这与数据密集路由并不相符。在数据密集路由中, 由于节点传输消息尺寸较大, 只依一次的相遇, 很难将数据传输至转发节点。因此, 在计算链路权重时不能只考虑相遇频率。实质上, 链路连通时长对链路质量有直接影响。其次, 多数 DTNs 路由并没有考虑到缓存区大小问题。多数 DTNs 路由只是假定每个节点有足够的缓存区空间。但是, 可用的缓存区空间是一个重要的参数。原因在于: 一条消息可能太大, 中间节点无法将其存于自己的缓存区内。

收稿日期: 2019-10-10; 修回日期: 2020-01-20

基金项目: 河北省教育厅青年基金资助项目(QN2017347); 河北省属高校基本科研业务费青年科研创新资助项目(SYKY2012)

作者简介: 辛 晨(1985-), 男, 在读硕士研究生, 讲师, 主要研究方向为图像处理、通信技术。email:fengtao_168@126.com

*通信作者: 冯 涛 email:2319343381@qq.com

考虑到现在 DTNs 路由的不足, 提出基于数据传输率的路由(DTRR)。DTRR 路由结合相遇频率、相遇时长及可用的缓存区空间 3 项信息, 估计路径的传输率。依据传输率择优选择转发节点, 再由转发节点传输数据。同时, 对缓存区空间进行管理。

1 网络模型

令 $\lambda_{i,j}$ 表示节点 v_i 与 v_j 的相遇频率。而 $1/\lambda_{i,j}$ 表示它们相遇的时间间隔。在时刻 t , 节点 v_i 相遇 v_j 的概率服从指数分布, 即 $\lambda_{i,j}e^{-\lambda_{i,j}t}$ 。因此, 在时间 T 内, 节点 v_i 相遇节点 v_j 的概率可表述为:

$$\int_0^T \lambda_{i,j} e^{-\lambda_{i,j}t} dt = 1 - e^{-\lambda_{i,j}T} \tag{1}$$

式中时间 T 的单位为 s。

每个节点都有一个缓存区, 用于临时存储消息块。网络内所有节点的缓存区大小相同。令 B_i 表示节点 v_i 的缓存区尺寸, 单位消息条数。而 B_{max} 表示最大的缓存区尺寸。此外, 本文引用表 1 所示的标识符。

考虑如图 1 所示的网络模型, 节点 v_i 需在时间 T 将消息 $M = \{m_1, m_2, \dots, m_\ell\}$ 传输至节点 v_j , 其中 ℓ 表示将消息 M 划分的消息块数^[5]。在每次相遇时, 节点 v_i 平均向节点 v_j 传输 c 个消息块。据此, 节点 v_i 需向 v_j 传输 ℓ/c 次。此外, 每条消息最多可存在 L 个复本。

令 $p_{i,j}(T, \ell)$ 表示节点 v_i 能在时间 T 向 v_j 传输消息 M 的概率, 其定义如式(2)所示:

$$p_{i,j}(T, \ell) = 1 - \sum_{k=0}^{\ell/c-1} \frac{(\lambda_{i,j}T)^k e^{-\lambda_{i,j}T}}{k!} \tag{2}$$

式中 k 表示传输的次数。

2 DTRR 路由

DTRR 路由依据以下思想进行转发数据包: 缓存区首部中的第一个消息块的传递速度应高于或者等于该消息同一个复本的最后一个消息块的传递速度。

注意了, 最后一个消息块并不意味着是缓存区的尾端。令 $m_i^{(j)}$ 表示消息 M 的第 j 个复本中的第 i 个消息块, 其中 j 表示复本数的索引号, 且 $1 \leq j \leq L$ 。而 i 表示消息块的索引号, 且 $1 \leq i \leq \ell$ 。假定是按消息块和复本号的升序方式, 将消息块压入缓存区(Buffer)。缓存时, 采用先进先出的原则。

如果消息块 $m_{100}^{(1)}$ 位于缓存区首部, 该消息的同一个复本的最后一个消息块可表述为 $m_\ell^{(1)}$, 而不是 $m_\ell^{(L)}$ 。而一个中间的转发节点不可能拥有消息所有复本。例如, 假定一个转发节点有 120 个消息块, 其消息块号在 $m_{100}^{(1)}$ 至 $m_{179}^{(1)}$ 之间和在 $m_{80}^{(2)}$ 至 $m_{119}^{(2)}$ 。若 $m_{100}^{(1)}$ 位于缓存区的首部, 则 $m_{100}^{(1)}$ 和 $m_{179}^{(1)}$ 分别表示第一个消息块和最后一个消息块。

2.1 缓存区的管理

由于交互的数据量大, 缓存区的管理对数据的传输有重要的影响^[6]。因此, 需有效地管理缓存区。DTRR 路由通过管理发送节点的缓存区, 提高最后一个消息块的传输率。

一个发送节点(源节点)的缓存区如图 2 所示。 $m_i^{(j)}$ 表示消息的第 j 个复本中的第 i 个消息块。假定在时刻 t_1 和 t_2 , 源节点分别向中间节点传输 $m_1^{(1)}$ 和 $m_\ell^{(1)}$ 消息块。源节点需要传输较大尺寸数据, 则满足 $t_2 \gg t_1$ 的几率很大。换言之, 消

表 1 标识符
Table1 Notations

symbol	definition
v_i	Node v_i
$N(i)$	The open neighbor set of v_i
$\lambda_{i,j}$	The contact frequency between v_i and v_j
$\delta_{i,j}$	The contact duration between v_i and v_j
M	A message $M = \{m_1, m_2, \dots, m_\ell\}$ with ℓ blocks
T	The message deadline
L	The number of copies of a message
B_i	The available buffer of v_i for a message, $B_i \in [1, B_{max}]$
η	The number of hops between two nodes
$p_{i,j}(\cdot)$	The weight of the link between v_i and v_j
$H_{i,j}(\cdot)$	The path weight between v_i and v_j

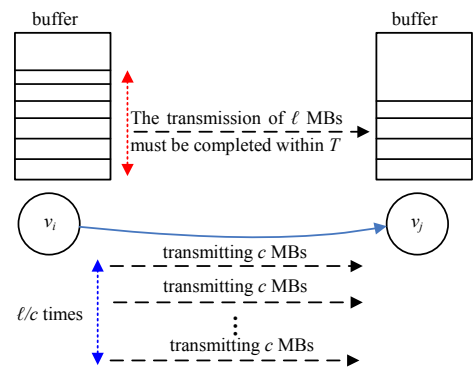


Fig.1 Network model
图 1 网络模型

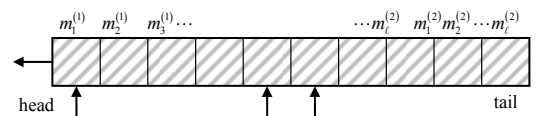


Fig.2 State of the source buffer with L=2
图 2 源节点缓存区的状态(L=2)

息块 $m_1^{(1)}$ 很可能在 t_3 时刻传输至目的节点 v_d 。而 t_3 时刻正是源节点传输第 2 复本的第 1 个消息块之时。

为充分利用消息块的重要性，第 j 个复本 ($j > 2$) 的消息块应以消息块序列号的降序重新排序，如图 3 所示。

2.2 路径指标

本小节推导在时间 T 内将最后一个消息块传输至目的节点的概率^[7]。令 v_s 为源节点或者是拥有消息块的中间节点。假定节点 v_s 在时刻 t 与中间节点 v_j 相遇。由于一个单位时间能传输一个消息块，则离截止时间 T 前，还剩余时间 $T' = T - t - 1$ 。因此，通过从节点 v_j 至目的节点 v_d 的 η 跳路径，在时间 $T' = T - t - 1$ 内，传输节点 v_s 缓存区中第一个消息块的概率可表示为 $\hat{H}_{j,d}^1(T')$ ：

$$\hat{H}_{j,d}^1(T') = \int_0^{T'} \sum_{k=1}^{\eta} A_k^{(\eta)} \lambda_k e^{-\lambda_k(T')} dt = \sum_{k=1}^{\eta} A_k^{(\eta)} (1 - e^{-\lambda_k(T'-t-1)}) \quad (3)$$

式中： λ_k 表示在此路径中第 k 跳的相遇频率； $A_k^{(\eta)}$ 为超指数分布的系数，其定义如式(4)所示：

$$A_k^{(\eta)} = \prod_{j=1, j \neq k}^{\eta} \frac{\lambda_j}{\lambda_j - \lambda_k} \quad (4)$$

接下来，推导在时限 T 内传输最后一个消息块的概率。假定节点 v_j 是 v_s 一个邻居节点。令 l' 表示节点 v_s 缓存区内剩余的消息块数。将 l' 个消息块从一个节点传输至另一个节点的过程服从泊松分布。因此，在剩余时间 T' 内将最后一个数据包传输至目的节点 v_d 的概率 $\hat{H}_{j,d}^\ell(T', l')$ ：

$$\hat{H}_{j,d}^\ell(T', l') = \int_{t=0}^{T'} p_{s,j}(T, l') \times \sum_{k=1}^{\eta} A_k^{(\eta)} (1 - e^{-\lambda_k(T-t)}) dt \quad (5)$$

2.3 数据传输过程

图 4 为传输消息块流程。令 v_i 是源节点或者具有消息块的中间节点，其需要向目的节点 v_d 传输消息^[8-9]，并且需在时限 T 内完成。假定节点 v_i 在时刻 t 与节点 v_j 相遇。

若节点 v_j 就是目的节点 v_d ，节点 v_i 就尽可能地将缓存区的消息块传输至节点 v_j ，直到链路断裂，或者已将最后一个消息块传输完毕。该过程如下所示：

```

if  $v_j = v_d$  then
 $m_x \leftarrow poll(v_i, buff)$ , where  $1 \leq x \leq \ell$  holds
 $v_i$  sends  $m_x$  to  $v_d$ 
until the link is disconnected or  $m_x$  is the last block
if  $v_d$  collects  $m_x$  for all  $1 \leq x \leq \ell$  then
returns SUCCESS
    
```

若节点 v_j 不是目的节点 v_d ，只是一个中间节点，节点 v_i 就计算自己将缓存区首部内消息块，传输至目的节点 v_d 的概率^[10]。即依据式(3)计算传输概率 $\hat{H}_{j,d}^1(T')$ ，再依据式(5)计算传输最后一个消息块的概率 $\hat{H}_{j,d}^\ell(T', l')$ 。

如果 $\hat{H}_{j,d}^1(T')$ 大于 $\hat{H}_{j,d}^\ell(T', l')$ ，则节点 v_i 就直接向节点 v_j 传输消息块；否则，就禁止传输消息。

3 性能分析

3.1 仿真环境

为了更好地分析 DTRR 路由性能，引用 CRAWDDAD 库内的轨迹数据^[11]。这些轨迹数据记录了节点相遇时长、相遇频率参数。

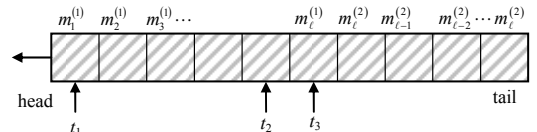


Fig.3 State of the source buffer with the manipulation (L=2)
图 3 源节点缓存区被管理后的状态(L=2)

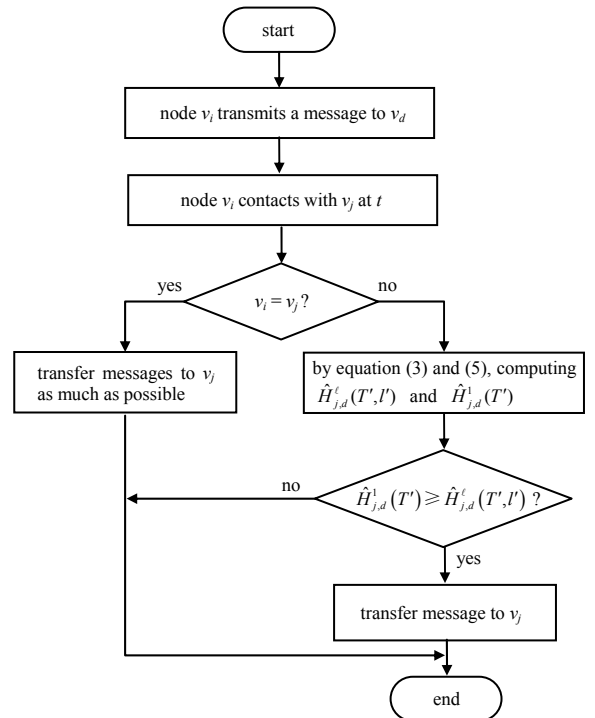


Fig.4 Process of transferring a message block
图 4 传输消息块的流程

仿真参数如下：41 个节点；节点间的接触频率和接触时间由轨迹数据给出；消息有效期在 1 000 至 30 000 单位时间；允许的消息复本数为 1 至 5 个；消息尺寸为 500 至 5 000 块；缓存区空间为 100 MB~3 000 MB；仿真时间为 100 s。选择等待路由(Spray-and-Wait Routing, SWR)^[12]和相遇时长感知路由(Contact Duration Aware Routing, CDAR)^[13]作为参照，分析 DTRR 路由的消息传递率和消息块传递比。

消息传递率表示目的节点接收的消息条数与所发送的消息条数之比，其反映了传输消息的成功率。消息传递率越高，路由性能越优。而消息块传输比是指目的节点在截止时限内所接收的消息块数与源节点传输的总的消息块数之比。由于一条消息有多个复本，并不需要接收消息的所有消息块，因此，消息块传输比越高，传输消息的开销越大。

接下来，分析消息尺寸 ℓ 、消息的复本数 L 、缓存区尺寸 B_{max} 对消息传递率和消息块传输比的影响。

3.2 消息尺寸 ℓ 的变化对性能的影响

本小节分析消息尺寸 ℓ 对消息传递率和消息块传输比的影响，其中消息尺寸 ℓ 从 500 至 5 000 变化。

图 5 显示了 ℓ 对消息传递率的影响。从图可知， ℓ 的增加，降低了消息传递率。这主要是因为： ℓ 越大，每条消息所划分的消息块数越多，这就加大网络负担，最终降低了消息传递率。相比于 SWR 和 CDAR 路由，提出的 DTRR 路由的消息传递率存在优势。即使 $\ell=5\ 000$ 时，DTRR 路由的传递率仍保持 90% 以上。

图 6 显示了消息尺寸 ℓ 对 DTRR 路由的消息块传输比的影响。从图可知，三个路由协议的消息块传输比随 ℓ 的增加呈线性增长关系。而 SWR 的消息块传输比最高，CDAR 路由的最低。这表明，SWR 路由并没有控制开销，只是简单地转发消息块，并没有控制转发消息块的数量。

3.3 消息的复本数 L 的变化对性能的影响

本小节分析复本数 L 对消息传递率和消息块传输比的影响，其中复本数 L 从 1 至 5 变化，如图 7、8 所示。

从图 7 可知，复本数 L 的增加对 CDAR 和 DTRR 路由的消息传递率的影响并不大，几乎没有影响。尤其是当复本数 L 大于 2 后，几乎保持稳定。这也说明，只增长消息的复本数，并不能够提升消息传递率。与图 5 类似，相比于 CDAR 和 SWR 路由，DTRR 路由提高了消息传递率。

图 8 显示了消息的复本数 L 对消息块传输比的影响。从图可知，CDAR 和 DTRR 路由的消息块传输比也不随复本数 L 的增加而变化，当复本数 L 大于 2 后，基本保持稳定不变。而 SWR 路由的消息传输比随复本数 L 的增加而上升。原因在于：在 SWR 路由中，只有遇见了目的节点，才停止转发消息块。

3.4 缓存区尺寸 B_{max} 变化对性能的影响

本小节分析 B_{max} 对消息传递率和消息块传输比的影响，其中 B_{max} 从 100 至 3 000 变化。

从图 9 可知， B_{max} 的增加使 CDAR 和 DTRR 路由的消息传递率缓慢增长，并逐步收敛。而 SWR 路由的消息传递率随 B_{max} 的增加而下降。这主要是因为：SWR 路由每次当缓存区存满才开始传输消息块。图 10 显示了 B_{max} 对消息块传输比的影响。从图 10 可知， B_{max} 的增加，使 CDAR、DTRR 和 SWR 路由的消息块呈增长趋势。而 CDAR 路由的消息块传输比最低，DTRR 的消息块传输比较高，当 B_{max} 增至 3 000 后，它的消息块传输比高于 SWR。

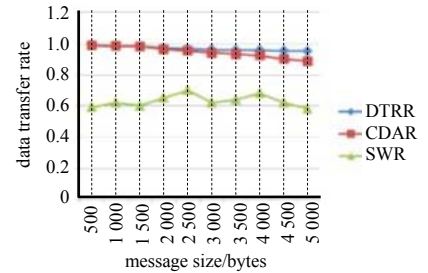


Fig.5 Data transfer rate vs. message size ℓ
图 5 消息传递率随消息尺寸 ℓ 变化

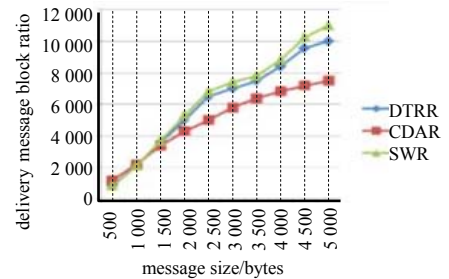


Fig.6 Delivery message block ratio vs. message size ℓ
图 6 消息块传输比随消息尺寸 ℓ 变化

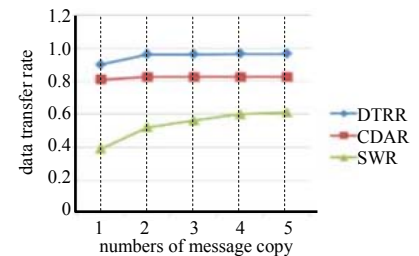


Fig.7 Effect of numbers of message copy L on data transfer rate
图 7 消息的复本数 L 对消息传递率的影响

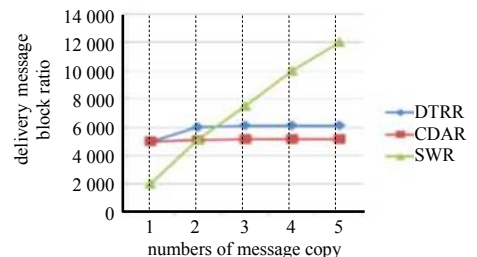


Fig.8 Effect of numbers of message copy L on delivery message block ratio
图 8 消息的复本数 L 对消息块传输比的影响

综上所述, 相比于 CADR 和 SWR 路由, 提出的 DTRR 路由的消息传递率得到提升, 但是其消息块传输比并不是最小。这说明, DTRR 路由是以增加一定的路由开销为代价, 换取消息传递率。换言之, 仍存在控制 DTRR 路由开销的空间。

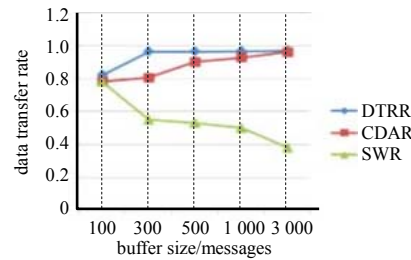


Fig.9 Data transfer rate vs. B_{max}

图 9 消息传递率随 B_{max} 变化

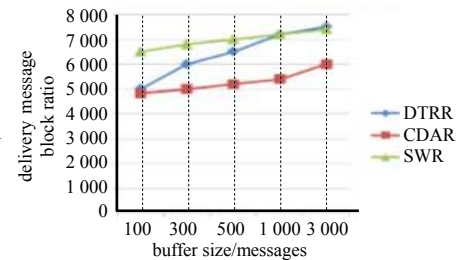


Fig.10 Delivery message block ratio vs. B_{max}

图 10 消息块传输比随 B_{max} 变化

4 结论

针对数据密集型路由, 提出基于消息传输率的时延容忍网络路由协议。DTRR 路由将消息划分为若干消息块, 且每条消息最多可形成多个复本。依据相遇频率、时长以及缓存区的可用空间计算路径传输消息的概率, 再依据概率选择下一跳转发节点。仿真结果表明, DTRR 路由有效地提高了消息传递率, 并控制了开销。

参考文献:

- [1] LE T, KALANTARIAN H, GERLA M. A novel social contact graph-based routing strategy for workload and throughput fairness in delay tolerant networks[J]. *Wireless Communications and Mobile Computing*, 2016, 16(11):1352-1362.
- [2] 尚亚丽. WSNs 中基于能效感知的任播路由[J]. *太赫兹科学与电子信息学报*, 2019, 17(6):1012-1016. (SHANG Yali. An energy-efficient anycast routing in wireless sensor networks[J]. *Journal of Terahertz Science and Electronic Information Technology*, 2019, 17(6):1012-1016.)
- [3] 郭稳涛, 李兵, 何怡刚. 基于社会活性和副本限制的 DTN 路由算法[J]. *电子测量与仪器学报*, 2017, 31(7):1047-1052. (GUO Wentao, LI Bing, HE Yigang. Social activity and copy-limited based DTN routing algorithm[J]. *Journal of Electronic Measurement and Instrumentation*, 2017, 31(7):1047-1052.)
- [4] 覃秋玉, 刘丹丹, 张健. 延迟容忍网络中自适应拥塞控制机制研究[J]. *计算机工程与应用*, 2018, 54(11):109-115. (QIN Qiuyu, LIU Dandan, ZHANG Jian. Adaptive congestion control mechanism for delay tolerant networks[J]. *Computer Engineering and Applications*, 2018, 54(11):109-115.)
- [5] 郑莹, 裴芳, 董龙明. 基于节点综合效能的 DTN 路由算法[J]. *火力与指挥控制*, 2017, 42(2):119-124. (ZHENG Ying, PEI Fang, DONG Longming. Synthetical effectiveness based DTN routing algorithm[J]. *Fire Control & Command Control*, 2017, 42(2):119-124.)
- [6] 薄珏, 乔林, 王丹妮, 等. 基于转发概率的时延容忍网络路由[J]. *中国电子科学研究院学报*, 2019, 14(8):781-785, 797. (BO Jue, QIAO Lin, WANG Danni, et al. Forwarding-probability based-routing in delay tolerant networks[J]. *Journal of China Academy of Electronics and Information Technology*, 2019, 14(8):781-785, 797.)
- [7] 和何, 李琳琳, 路云飞. 战场环境下基于节点移动性的 DTN 路由算法[J]. *火力与指挥控制*, 2018, 43(12):38-41. (HE He, LI Linlin, LU Yunfei. DTN routing algorithm based mobility of nodes in battle environment[J]. *Fire Control & Command Control*, 2018, 43(12):38-41.)
- [8] 李建铎, 刘建明, 李龙, 等. 基于节点关键度的 DTN 路由算法[J]. *计算机仿真*, 2018, 35(4):126-130. (LI Jianduo, LIU Jianming, LI Long, et al. DTN routing algorithm based on key nodes[J]. *Computer Simulation*, 2018, 35(4):126-130.)
- [9] 陆芳, 李建波, 宋有美, 等. 基于局部位置信息和消息投递度的受控传染路由算法[J]. *小型微型计算机系统*, 2018, 39(5):56-61. (LU Fang, LI Jianbo, SONG Youmei, et al. Location position and message delivery ratio based controlled epidemic routing for DTNs[J]. *Journal of Chinese Computer Systems*, 2018, 39(5):56-61.)
- [10] 黄宏程, 熊忠阳, 胡敏, 等. 节点移动状态感知的社会化延迟容忍网络路由策略[J]. *计算机应用研究*, 2017, 34(6):1825-1829. (HUANG Hongcheng, XIONG Zhongyang, HU Min, et al. Routing strategy based on change perception of node mobility characteristic in DTN[J]. *Application Research of Computers*, 2017, 34(6):1825-1829.)
- [11] SCOTT J, GASS R, CROWCROFT J. CRAWDAD dataset Cambridge/haggle(v.2009-05-29)[DB/OL]. CRAWDAD, 2009. [2020-01-20]. <http://crawdad.org/cambridge/haggle/20090529>.
- [12] WANG G, HUI Z, DONG P, et al. A routing scheme based on spray-and-wait and connecting-capability of node[C]// *International Conference on Progress in Informatics & Computing*. Shanghai, China: IEEE, 2017:34-43.
- [13] ZHAO J, ZHUO X, LI Q. Contact duration aware data replication in DTNs with licensed and unlicensed spectrum[J]. *IEEE Transactions on Mobile Computing*, 2016, 15(4):803-816.