

# DTN 中基于节点综合性能的自适应喷射等待路由算法

崔建群<sup>1</sup> 孙佳悦<sup>1</sup> 常亚楠<sup>1</sup> 余东海<sup>1</sup> 邬尧<sup>1</sup> 吴黎兵<sup>2</sup>

<sup>1</sup>(华中师范大学计算机学院 武汉 430079)

<sup>2</sup>(武汉大学国家网络安全学院 武汉 430072)

(jqcui@126.com)

## An Adaptive Spray and Wait Routing Algorithm Based on Comprehensive Performance of Node in DTN

Cui Jianqun<sup>1</sup>, Sun Jiayue<sup>1</sup>, Chang Yanan<sup>1</sup>, Yu Donghai<sup>1</sup>, Wu Yao<sup>1</sup>, and Wu Libing<sup>2</sup>

<sup>1</sup>(School of Computer, Central China Normal University, Wuhan 430079)

<sup>2</sup>(School of Cyber Science and Engineering, Wuhan University, Wuhan 430072)

**Abstract** In delay tolerant network (DTN), due to the frequent changes of network topology, there is no stable link between the end to end, so it is one of the key problems in DTN research to select the appropriate relay node for message forwarding and delivering the message to the destination node in a short time. Aiming at the blindness of relay node selection and the lack of reasonable control over message copies distribution in existing routing algorithms, an adaptive spray and wait routing algorithm based on comprehensive performance of node (CPN-ASW) is proposed: in spray phase, a new metric, called as node similarity, is used to measure the similarity degree of motion trajectory between nodes, and different relay node selection strategies are adopted according to whether the node similarity exceeds the given threshold value, and subsequently the number of message copies are adaptively allocated according to the relative utility value of nodes; in wait phase, an active forwarding strategy based on delivery probability is implemented, if a relay node has a larger delivery probability to the destination node, forwarding the message to this relay node. Simulation results show that compared with the Epidemic, Spray and Wait (SaW), EBR and PBSW, CPN-ASW can effectively control the network overhead while improving the delivery ratio and reducing the average delay.

**Key words** delay tolerant network (DTN); spray and wait routing; node similarity; relative utility value; delivery probability

**摘要** 延迟容忍网络(delay tolerant network, DTN)中,由于网络拓扑频繁变化,端到端之间不存在稳定的链路,如何选择合适的中继节点进行消息转发,使消息在较短时间内交付到目标节点是DTN中研究的关键问题之一.针对现有路由算法中继节点选择的盲目性以及对消息副本的分发缺乏合理控制的问题,提出一种基于节点综合性能的自适应喷射等待路由算法(adaptive spray and wait routing algorithm based on comprehensive performance of node, CPN-ASW):在Spray(喷射)阶段引入节点相似度指标来衡量节点间运动轨迹的相似程度,根据节点相似度是否超过给定阈值采用不同的中继节点选择策略,确定中继节点后,按照节点相对效用值自适应分配消息副本数量;在Wait(等待)阶段实现

收稿日期:2020-11-25;修回日期:2021-06-21

基金项目:国家自然科学基金项目(61672257,61702210,61772377)

This work was supported by the National Natural Science Foundation of China (61672257, 61702210, 61772377).

通信作者:常亚楠(ynchang@mail.ccnu.edu.cn)

主动转发,将消息转发给到目标节点投递预测值更高的中继节点.实验结果表明,与 Epidemic, Spray and Wait (SaW), EBR, PBSW 这 4 种算法相比,CPN-ASW 算法能够有效提高消息投递率,降低网络开销和平均时延.

**关键词** 延迟容忍网络;喷射等待路由;节点相似度;相对效用值;投递预测值

中图法分类号 TP393

延迟容忍网络(delay tolerant network, DTN)是一种在源节点和目标节点之间难以形成稳定的端到端连接链路、利用节点移动带来的相遇机会进行通信的自组织网络<sup>[1]</sup>.正是由于 DTN 的这种通信特点,其采用“存储-携带-转发”的路由机制<sup>[2]</sup>,消息由中间节点携带并随其移动,直至到达目标节点.DTN 适用于大传输时延、间歇性连接、高误码率的极限通信环境,在灾难应急<sup>[3]</sup>、车载网络<sup>[4]</sup>、星际网络<sup>[5]</sup>、手持设备组网<sup>[6]</sup>等领域具有广泛的应用前景,并将成为未来普适计算和泛在网络的重要接入技术.因此 DTN 被认为是实现“无处不在的网络”的一项关键技术,具有重要的研究意义.

在 DTN 路由算法中,SaW(spray and wait)<sup>[7]</sup>算法能够在控制消息副本的前提下实现消息的高效投递.该算法在 Spray(喷射)阶段采用对称分配消息副本数,在 Wait(等待)阶段进行消息的被动传输.文献[7]证明了在独立同分布的节点移动模型中 SaW 算法是最优的,但是在大多数情况下并不存在严格的独立同分布移动模型,因此 SaW 算法仍然有较大的改进空间.

目前对 SaW 算法的改进主要体现在 3 个方面:1)在 Spray 阶段筛选中继节点,根据效用函数选择高效的候选节点作为下一跳,有利于提高消息投递率<sup>[8]</sup>;2)在 Spray 阶段喷射适当比例的副本,根据不同节点的质量和表现,分配不同的副本数量,使副本快速扩散,有效传递<sup>[9-10]</sup>;3)改进 Wait 阶段的传输策略<sup>[11]</sup>.然而,目前大多数 SaW 改进算法只是根据其中的 1 个或 2 个方面进行研究,很少考虑将这 3 个方面综合起来对 SaW 进行改进,性能提高并不明显.

基于以上分析,本文将中继节点选择、自适应分配消息副本数量以及 Wait 阶段的改进综合考虑进来,提出一种基于节点综合性能的自适应喷射等待路由算法(adaptive spray and wait routing algorithm based on comprehensive performance of node, CPN-ASW).主要贡献有 5 个方面:

1) 定义了节点相似度来反映节点运动轨迹的相似程度,利用节点相似度控制消息转发条件,避免

消息在运动轨迹相似的节点间传输,网络开销增大,投递率无明显变化情况的发生.

2) 引入节点活跃度概念来反映节点与其他节点的相遇能力,从而将较多的消息副本分配给活跃度高的节点,使消息能够在短时间内扩散至较多的节点上.另外,采用划分等长的时间窗口来更加准确地估计节点活跃度.

3) 结合节点活跃度和剩余缓存,提出节点相对效用值的概念.根据节点间的相对效用值自适应分配消息副本数,给活跃度高、剩余缓存多的节点分配更多的消息副本,使消息在网络中扩散得更广,进而提高消息到达目标节点的可能性.

4) 在 Spray 阶段提出基于节点相似度和节点相对效用值的喷射策略,使在副本受限的情况下,消息得到有效扩散.

5) 在 Wait 阶段充分利用节点间的相遇机会,提出基于节点投递预测值的主动转发策略,进一步降低传输时延.

## 1 相关工作

目前,在 DTN 的研究中,路由算法主要分为单拷贝路由协议和多拷贝路由协议.典型的单拷贝路由协议有 Direct Delivery<sup>[12]</sup> 和 First Contact<sup>[12]</sup>.这 2 种路由协议网络开销小,但是传输时延很高,且消息成功到达目标节点的概率很低.多拷贝路由协议中最具代表性路由协议有 Epidemic<sup>[13]</sup> 和 Prophet<sup>[14]</sup>.Epidemic 是一种洪泛式路由算法,携带消息的节点向遇到的每一个节点都复制消息,其在网络资源较充足的条件下投递率较高,且传输时延低,但是在真实的网络场景中,网络资源往往受限,其盲目复制会导致网络开销巨大,最终降低网络性能;Prophet 算法利用节点间相遇历史信息和传递性定义一个投递预测值指标来描述节点之间成功传输消息的概率,每当 2 个节点相遇时,选择到达目标节点投递预测值更高的节点作为消息中继节点,但是 Prophet 本质上仍是消息复制路由,相比 Epidemic 网络开销明显

减小,但由于进行中继节点筛选,消息到达目标节点时延较大。

为了将 Epidemic 算法消息投递的高效性与单副本直接传输的简洁性相结合,相关研究者提出了 SaW 路由算法。该路由算法由 Spray 和 Wait 这 2 个阶段组成,消息产生后,通过初始化消息副本数量来显示消息转发次数。SaW 算法采用了 2 种模式,即 Binary 和非 Binary 模式。Binary 模式下,在 Spray 阶段,携带某消息副本数大于 1 的节点向遇到的未缓存该消息的节点转发此消息一半副本,自身保留剩下的消息副本;当节点中某消息副本数为 1 时,节点转入 Wait 阶段,消息以 Direct Delivery 的方式进行传输,即携带消息的节点直至遇到目标节点才将消息转发出去。Binary SaW 可应用于多类场景中,其综合性能要高于 Epidemic 和 Prophet。该算法适用于节点均等分布、运动随机的延迟容忍网络,然而在真实的网络环境中,节点运动不完全随机,且节点将消息传输到目标节点的能力是不同的,因此,SaW 算法在 Spray 阶段的对称分配副本数是不灵活的,并且在 Wait 阶段的被动传输会进一步增加传输时延。针对 SaW 算法的主要弊端,目前国内外研究者已在 SaW 的改进上取得了一些成果。

文献[8]对 Prophet 算法中定义的投递预测值做了进一步改进,节点间的投递预测值不仅考虑相遇次数,还结合了节点间的连接时间。在 Spray 阶段根据消息到目标节点的投递预测值筛选中继节点,Wait 阶段同样执行此操作,直至消息到达目标节点。文献[9]结合节点的投递预测值和节点相似度作为节点效用值,按照此效用值自适应分配副本数。文献[10]提出基于节点历史相遇信息的路由算法 EBR,该算法直接根据节点活跃度自适应地分配消息副本数,将更多的消息副本分配给活跃度更高的节点。文献[15]提出一种基于节点社会性的喷雾等待路由协议 BSW-SN,该算法在 Spray 阶段根据节点的移动模式和节点对消息的转发效能来优化对中继节点的选择,然后按照节点的活跃度自适应分配消息副本数。文献[16]提出一种基于节点质量度的路由算法 QoN-ASW,在 Spray 阶段考虑节点的质量度作为消息副本分配的依据;在 Wait 阶段将消息转发给到目标节点投递预测值更高的中继节点,从而减少消息传输时延。文献[17]提出基于相遇概率的喷射等待路由算法 PBSW,在 Spray 阶段选择与目标节点相遇概率更高的相遇节点作为中继节点,然后根据两者与目标节点的相遇概率分配适当比例

的消息副本。然而,上述文献只是从筛选中继节点、灵活分配消息副本数、改进 Wait 阶段传输策略中的 1 个或 2 个方面进行改进,从而改进策略不够全面。

基于以上分析,本文综合考虑中继节点选择、自适应分配消息副本数量、改进 Wait 阶段传输策略这 3 个方面,提出一种基于节点综合性能的自适应喷射等待路由算法 CPN-ASW。该算法在 Spray 阶段根据节点间的相似程度是否超过给定阈值而采用不同的中继节点选择策略,考虑相遇节点间的活跃度和剩余缓存作为节点的相对效用值,当节点间的相似程度小于给定阈值,直接根据节点的相对效用值自适应地分配消息副本数,否则,选择合适的中继节点后再根据节点相对效用值分配消息副本数;进一步为充分利用节点间的相遇机会,在 Wait 阶段将副本数为 1 的消息转发给到目标节点投递预测值更高的中继节点。

## 2 问题模型

### 2.1 网络模型

假设网络中包含  $n$  个节点,  $G = (V, E)$  为网络拓扑图,  $V = \{v_i \mid 1 \leq i \leq n\}$  表示网络中的节点集合,  $E$  为定义在  $G$  上的边集, 代表节点对之间的连接。每个节点  $v_i$  维护一个历史相遇节点集合  $E_i = \{v_j \mid n_{i,j} \neq 0\}$ , 该集合记录从仿真开始节点  $v_i$  遇到过的每个节点。当节点  $v_i$  和  $v_j$  进入到彼此的通信范围内,  $v_i$  检测  $E_i$  中是否包含节点  $v_j$ , 若节点  $v_j$  不包含在  $E_i$  中, 则节点  $v_i$  将  $v_j$  添加到  $E_i$  中, 并且将  $E_i$  中的  $n_{i,j}$  值加 1; 若节点已包含在  $E_i$  中, 说明两节点曾经相遇过, 此时只将  $E_i$  中的  $n_{i,j}$  加 1。

### 2.2 节点相似度

我们在大多数的 DTN 应用场景中, 具有通信功能的移动设备大多是由人类随身携带的, 人类的移动模式不是完全随机的, 他们会表现出一定的社会特征。例如, 如果 2 个人拥有 1 个或者多个共同的朋友, 那么这 2 个人成为熟人的可能性很高, 这种现象称为“聚类”, 即这 2 个人的运动轨迹相似。如果消息在运动轨迹相似的节点间转发, 那么在增大网络开销的同时, 对提高投递率并没有实质性的帮助。因此, 在本文中提出节点相似度的概念来表示节点间运动轨迹的相似程度, 如果 2 个节点间的相似度高, 则需提高消息转发条件, 进而减少不必要的转发次数。

**定义 1.** 节点相似度。节点  $v_i$  与节点  $v_j$  间的相似度是指节点  $v_i$  和节点  $v_j$  的历史相遇节点集合中

拥有的共同历史相遇节点的个数与这 2 个节点中较大历史相遇节点集合中节点个数的比值,记为  $SD_{(i,j)}$ ,其表示为

$$SD_{(i,j)} = \frac{|E_i \cap E_j|}{\max\{|E_i|, |E_j|\}}, \quad (1)$$

如果  $SD_{(i,j)}$  越大,即节点间的相似度越高,意味着节点  $v_i$  和节点  $v_j$  的运动范围和接触节点越相似.

### 2.3 节点活跃度

在 DTN 中,活跃的节点能够快速传递消息,使消息在较短的时间内扩散至较多的节点上,从而提高投递率,降低传输时延.然而,DTN 中网络拓扑结构频繁变化,节点相遇其他节点的能力在较长的仿真时间内可能波动比较大,因此,本文中通过划分等长的时间窗口来更加准确估计节点活跃度,每当 1 个时间窗口到期,就更新节点活跃度.节点活跃度反映了节点向网络中扩散消息的能力.

假设节点  $v_i$  当前运动时刻位于第  $k+1$  个时间窗口内,在第  $k+1$  个时间窗口内相遇节点的数量是未知的,本文通过历史  $k$  个时间窗口算出的活跃度估计  $v_i$  在当前时间窗口内的活跃度.

**定义 2.** 节点活跃度.节点  $v_i$  的活跃度是指  $v_i$  在当前时间窗口相遇其他节点的能力,记为  $EV_i^{k+1}$ ,其表示为

$$EV_i^{k+1} = \begin{cases} CWC_i^k, k=1, \\ \alpha \times CWC_i^k + (1-\alpha) \times EV_i^{k-1}, k>1, \end{cases} \quad (2)$$

其中,  $EV_i^{k+1}$  就是预估节点  $v_i$  在第  $k+1$  个时间窗口内遇到的不同节点的数量;  $k$  表示节点  $v_i$  经历完整时间窗口的数量;  $CWC_i^k$  表示最近 1 个完整时间窗口内节点  $v_i$  遇到不同节点的数量,每当 1 个时间窗口到期,  $CWC_i^k$  就清 0,开始重新统计新的时间窗口内相遇节点的数量;  $EV_i^{k-1}$  表示从仿真开始至第  $k-1$  个时间窗口到期更新后节点  $v_i$  的活跃度.当  $k=1$  时,即节点  $v_i$  经历过第 1 个完整时间窗口,此时节点  $v_i$  在第  $k+1$  个时间窗口内的活跃度直接表示为在第 1 个时间窗口内所遇到的不同节点的数量;当  $k>1$  时,节点  $v_i$  在第  $k+1$  个时间窗口内的活跃度表示为从仿真开始到第  $k-1$  个时间窗口到期更新的  $EV_i^{k-1}$  和由第  $k$  个窗口内计算得出的  $CWC_i^k$  的加权和,其中  $\alpha$  为活跃度平滑因子.

### 2.4 节点相对效用值

为了使 CPN-ASW 的 Spray 阶段副本分配更具有合理性,本文根据节点间的相对效用值对副本进行分配.节点的活跃度越高,则未来遇到目标节点的可能性越大,则给活跃度高的节点分配更多的消息

副本,这会造成活跃度高的节点易发生缓存溢出的情况,造成消息丢包次数较多,反而没有体现出活跃度高的节点能使消息在网络中快速扩散的优势.为了缓解活跃度高的节点上的拥塞程度,则同时将节点剩余缓存作为消息副本分配的依据,如果节点很活跃,但是剩余缓存已不足,则避免将较多的消息副本分配给此节点,防止节点频繁拥塞导致消息丢包次数明显增加,从而使较多的消息副本不能被有效传递.因此本文综合利用节点的活跃度和剩余缓存,来体现节点相对效用值的高低.节点的活跃度越高,剩余缓存越多,其相对效用值越高,得到的转发该消息的次数越多.

**定义 3.** 节点相对效用值.当节点  $v_i$  和节点  $v_j$  在第  $k+1$  个时间窗口内相遇时,节点  $v_i$  的相对效用值记为  $U_i$ ,其计算为

$$U_i = w \times \frac{EV_i^{k+1}}{EV_i^{k+1} + EV_j^{k+1}} + (1-w) \times \frac{FB_i}{FB_i + FB_j}, \quad (3)$$

其中,  $w \in (0, 1)$  是相对效用值参数,  $EV_i^{k+1}$  和  $FB_i$  与  $EV_j^{k+1}$  和  $FB_j$  分别表示节点  $v_i$  和节点  $v_j$  在第  $k+1$  个时间窗口内相遇时的活跃度和剩余缓存大小.

### 2.5 节点投递预测值

在 DTN 中,节点运动并非完全随机.如果节点  $v_i$  和  $v_j$  在过去的时间内接触比较频繁,那么未来它们的相遇概率也会比较高.Prophet 路由算法正是基于历史预测策略的代表.该算法利用节点间的相遇历史信息和传递性来定义投递预测值(delivery probability),利用投递预测值来描述节点间的相遇概率.当节点  $v_i$  和  $v_j$  相遇时,只有  $v_j$  到消息目标节点的投递预测值更高时,  $v_i$  才将消息传输给  $v_j$ ,通过这种有选择地复制消息副本,有效地降低网络开销.

假如用  $P_{(i,j)}$  来表示节点和节点间的投递预测值,  $P_{(i,j)}$  的计算分为 3 个过程:更新、衰减和传递性.

1) 更新.当节点  $v_i$  和  $v_j$  相遇并建立连接时,更新投递预测值:

$$P_{(i,j)} = P_{(i,j) \text{ old}} + (1 - P_{(i,j) \text{ old}}) \times P_{\text{init}}, \quad (4)$$

其中,  $P_{(i,j)}$  表示节点  $v_i$  和  $v_j$  间的历史投递预测值,  $P_{\text{init}} \in [0, 1]$  是初始化常量.

2) 衰减.当节点  $v_i$  和  $v_j$  在一段时间内没有遇到彼此,对投递预测值进行衰减:

$$P_{(i,j)} = P_{(i,j) \text{ old}} \times y^c, \quad (5)$$

其中,  $y$  表示衰减因子,  $c$  表示节点  $v_i$  和  $v_j$  从上次相遇到目前为止所经历的时间块的个数.

3) 传递性.节点投递预测值也具有传递性,根据

观察,如果节点  $v_i$  经常遇到节点  $v_j$ ,而节点  $v_j$  经常遇到节点  $v_h$ ,那么对目标节点为节点  $v_i$  的消息而言,节点  $v_h$  可能是一个很好的中继节点.这种传递性如何投递预测值  $P_{(i,h)}$ :

$$P_{(i,h)} = P_{(i,h) \text{ old}} + (1 - P_{(i,h) \text{ old}}) \times P_{(i,j)} \times P_{(j,h)} \times \beta. \quad (6)$$

### 3 CPN-ASW 路由算法

基于节点综合性能的自适应喷射等待路由算法 CPN-ASW 分为 Spray 和 Wait 这 2 个阶段.在 Spray 阶段,提出基于节点相似度和节点相对效用值的喷射策略;在 Wait 阶段,提出基于节点投递预测值的主动转发策略.

#### 3.1 Spray 阶段

在真实的 DTN 应用场景中,大量的移动设备通过人来携带,人的行为具有很强的社会性.例如,拥有相同兴趣爱好的人通常会聚集在一起,即这些人的运动轨迹相似.在副本受限的情况下,应提高消息在相似度高的节点间的转发条件,避免网络开销增大,投递率却无明显提高情况的出现.另外,不同种类的节点缓存、移动速度不尽相同,因此承担消息转发任务的能力也有高低.在本文中通过定义节点相对效用值来描述节点转发消息的能力,给相对效用值高的节点分配更多的消息副本,有利于消息得到有效扩散.

基于以上分析,本节提出一种基于节点相似度和节点相对效用值的喷射策略.根据节点相似度是否超过阈值采用不同的中继节点选择策略,确定好中继节点后,按照节点相对效用值分配消息副本数量.具体喷射策略如下.

当节点  $v_i$  和节点  $v_j$  相遇时,若节点  $v_j$  不携带  $v_i$  持有消息  $m_l$ ,节点  $v_i$  向节点  $v_j$  分配消息  $m_l$  副本数量的步骤为:

1) 判断节点  $v_j$  是否是目标节点,若是,则节点  $v_i$  直接将消息的一个副本传递给  $v_j$ ,同时删除该消息;否则,执行步骤 2) 或步骤 3).

2) 若节点间的相似度  $SD_{(i,j)} \geq \tau$ ,说明这 2 个节点相遇节点集合高度重叠,此时为减少消息在运动范围大致重叠且相对效用值相差不大的节点间的转发次数,若满足:

$$U_j > U_i + U_{\text{th}} \times SD_{(i,j)}, \quad (7)$$

则节点  $v_i$  确定节点  $v_j$  为中继节点;否则,  $v_i$  等待与

目标节点或其他节点相遇.其中,  $\tau$  为相似度阈值,  $U_{\text{th}}$  为效用参数.

若转发条件满足式(7),根据节点  $v_i$  和节点  $v_j$  的相对效用值按比例分配消息副本数,设节点  $v_i$  中的消息  $m_l$  副本数为  $L_{i \text{ old}}(m_l)$ ,分配给节点  $v_j$  的消息副本数  $L_{j \text{ new}}(m_l)$  为

$$L_{j \text{ new}}(m_l) = \lfloor U_j \times L_{i \text{ old}}(m_l) \rfloor. \quad (8)$$

节点  $v_i$  自身保留消息  $m_l$  副本数  $L_{i \text{ new}}(m_l)$  为

$$L_{i \text{ new}}(m_l) = L_{i \text{ old}}(m_l) - L_{j \text{ new}}(m_l). \quad (9)$$

3) 若节点间的相似度  $SD_{(i,j)} < \tau$ ,说明这 2 个节点的运动范围相似性较小,此时为了消息的传播范围更广,直接按照节点相对效用值在节点间按比例分配消息副本数,分配给  $v_j$  的消息  $m_l$  副本数  $L_{j \text{ new}}(m_l)$  如式(8)所示,节点  $v_i$  剩余消息  $m_l$  副本数  $L_{i \text{ new}}(m_l)$  如式(9)所示.

当节点  $v_i$  和节点  $v_j$  相遇,  $v_i$  向  $v_j$  分配消息副本过程如算法 1 所示:

#### 算法 1. Spray 阶段算法.

输入: 节点  $v_i$  和节点  $v_j$  相遇;

输出:  $v_i$  即将发送给  $v_j$  的副本数大于 1 的消息集合  $MforwardList$ .

```

① 节点间相互交换变量  $EV_i^{k+1}, FB_i, EV_j^{k+1},$ 
 $FB_j$  ;
② 根据式(1)更新  $SD_{(i,j)}$  ;
③ 根据式(3)计算  $U_i, U_j$  ;
④ FOR  $\forall m_l \in v_i$  and  $m_l \notin v_j$ 
⑤ IF  $L_{i \text{ old}}(m_l) > 1$  THEN
⑥    $v_d = m_l.destination$  ;
⑦   IF  $v_d == v_j$  THEN
⑧     节点  $v_i$  将消息  $m_l$  转发给  $v_j$  ;
⑨     节点  $v_i$  删除消息  $m_l$  ;
⑩   continue;
⑪ ELSE IF  $SD_{(i,j)} \geq \tau$  THEN
⑫   IF  $U_j > U_i + U_{\text{th}} \times SD_{(i,j)}$  THEN
⑬      $L_{j \text{ new}}(m_l) = \lfloor U_j \times L_{i \text{ old}}(m_l) \rfloor$  ;
⑭      $L_{i \text{ new}}(m_l) = L_{i \text{ old}}(m_l) - L_{j \text{ new}}(m_l)$  ;
⑮   END IF
⑯ ELSE
⑰    $L_{j \text{ new}}(m_l) = \lfloor U_j \times L_{i \text{ old}}(m_l) \rfloor$  ;
⑱    $L_{i \text{ new}}(m_l) = L_{i \text{ old}}(m_l) - L_{j \text{ new}}(m_l)$  ;
⑲ END IF
⑳ IF  $L_{j \text{ new}}(m_l) \neq 0$  THEN
㉑    $MforwardList.add(m_l, L_{j \text{ new}}(m_l))$  ;

```

```

②② END IF
②③ IF  $L_{i \text{ new}}(m_l) == 0$  THEN
②④ 节点  $v_i$  删除消息  $m_l$  ;
②⑤ END IF
②⑥ END IF
②⑦ END FOR
②⑧ 将  $M_{forwardList}$  中的消息按剩余生存时
    间从大到小排序;
②⑨ 将  $M_{forwardList}$  中的消息转发给节点  $v_j$ .

```

算法 1 中,行⑦~⑩表示当相遇节点  $v_j$  是消息目标节点时,直接将消息转发给  $v_j$  并删除节点  $v_i$  中此消息;行⑪~⑯表示当节点  $v_i$  和节点  $v_j$  相似度大于阈值时,筛选合适中继节点后再按照节点相对效用值计算分配消息副本数;行⑰~⑲表示  $v_i$  和  $v_j$  的相似度未超过阈值,直接按照节点相对效用值计算在节点间分配消息副本数;行⑳~㉑表示将符合条件的消息添加到  $M_{forwardList}$  中;行㉒~㉓表示按照消息剩余生存时间从大到小的顺序将  $M_{forwardList}$  中消息转发给  $v_j$ .

假设网络中的消息总数为  $M$ ,在算法 1 中,主要时间消耗在行④~㉑,时间复杂度为  $O(M)$ ,因此,该算法的时间复杂度为  $O(M)$ .

### 3.2 Wait 阶段

在传统的 SaW 路由算法中,当节点所携带的某个消息的副本数为 1 时,进入到 Wait 阶段,该阶段存在弊端:节点携带消息直到遇到目标节点才将消息转发出去,没有充分利用节点间的相遇机会.

基于上述分析,本节提出基于节点投递预测值的主动转发策略.在 Wait 阶段利用 Prophet 路由算法中的投递预测值来描述节点间成功传输消息的概率,当节点  $v_i$  中持有相遇节点  $v_j$  中没有的消息时,以相遇节点  $v_j$  的缓存占用比来调节阈值大小,只有节点  $v_j$  到消息目标节点的投递预测值大于节点  $v_i$  到消息目标节点的投递预测值,并且两者到消息目标节点的投递预测值区分度大于给定阈值时,节点  $v_i$  才将消息  $m_l$  转发给相遇节点  $v_j$ .

当节点  $v_i$  中的消息  $m_l$  只剩下一个消息副本时,与未缓存消息  $m_l$  的节点  $v_j$  相遇,若满足:

$$P_{(j,d)} > P_{(i,d)} + P_{th} \times OB_j, \quad (10)$$

则节点  $v_i$  将该消息转发给节点  $v_j$ .其中,  $P_{(i,d)}$  和  $P_{(j,d)}$  分别表示节点  $v_i$  和节点  $v_j$  将消息  $m_l$  成功投递到目标节点  $v_d$  的概率;  $P_{th} \in [0, 1]$  为预设参数;  $OB_j$  为相遇节点  $v_j$  的缓存占用率.在转发过程中通

过利用相遇节点  $v_j$  的缓存占用比来调节节点间的投递预测值区分度,一方面避免消息在投递概率近似的节点间转发,另一方面随着相遇节点  $v_j$  的缓存占用率不断增大,转发条件越来越高,从而降低网络开销,较大程度上避免了拥塞情况的出现.

在 Wait 阶段,节点  $v_i$  和节点  $v_j$  相遇时,节点  $v_i$  执行的消息转发过程如算法 2 所示:

#### 算法 2. Wait 阶段算法.

输入: 节点  $v_i$  和节点  $v_j$  相遇;

输出:  $v_i$  即将发送给  $v_j$  的副本数为 1 的消息集合  $S_{forwardList}$ .

```

① 根据式(4)更新  $P_{(i,j)}$  ;
② 节点间相互交换变量  $OB_i, OB_j$  ;
③ FOR  $\forall m_l \in v_i$  and  $m_l \notin v_j$ 
④ IF  $L_{i \text{ old}}(m_l) == 1$  THEN
⑤  $v_d = m_l.destination$  ;
⑥ IF  $v_d == v_j$  THEN
⑦ 节点  $v_i$  将消息  $m_l$  转发给  $v_j$  ;
⑧ 节点  $v_i$  删除消息  $m_l$  ;
⑨ continue;
⑩ ELSE IF  $P_{(j,d)} > P_{(i,d)} + P_{th} \times OB_j$ 
    THEN
⑪  $S_{forwardList}.add(m_l, 1)$  ;
⑫ 节点  $v_i$  删除消息  $m_l$  ;
⑬ END IF
⑭ END IF
⑮ END FOR
⑯ 将  $S_{forwardList}$  中的消息按剩余生存时间
    从大到小排序;
⑰ 将  $S_{forwardList}$  中的消息转发给节点  $v_j$ .

```

算法 2 中,行⑥~⑨表示当节点  $v_j$  是消息目标节点时,直接将消息转发给  $v_j$  并删除节点  $v_i$  中此消息;行⑩~⑫表示当节点  $v_j$  不是消息目标节点时,只有满足式(10)才将消息转发给  $v_j$ ,同时  $v_i$  删除此消息;行⑯~⑰表示按照消息剩余生存时间从大到小的顺序将  $S_{forwardList}$  中消息转发给  $v_j$ .

## 4 仿真结果与分析

### 4.1 仿真环境配置

实验采用 ONE<sup>[18]</sup> 仿真平台,将 CPN-ASW 与 Epidemic, SaW, EBR, PBSW 在相同实验环境下进行对比.仿真采用 ONE 自带的赫尔辛基市地图作为

移动场景,大小为  $4500\text{ m} \times 3400\text{ m}$ ,节点数量设置为 126 个,共分为 3 类节点:行人、出租车、有轨电车.其中行人组 80 个节点,出租车组 40 个节点,这 2 组按照基于地图路线的最短路径模型移动,有轨电车组共 6 个节点,按照基于地图的固定路线模型移动.具体的仿真参数设置如表 1 所示,其他参数采用 ONE 平台自带的默认设置.对于初始常量  $P_{\text{init}}$ 、传递因子  $\beta$ 、衰减参数  $y$  这 3 个参数设置可参考文献 [14] 中设定的值,取  $P_{\text{init}}=0.75$ ,  $\beta=0.25$ ,  $y=0.98$ .

Table 1 Simulation Parameters

表 1 仿真参数

参数	值
仿真时间/s	43 200
消息生存周期/min	250
节点缓存大小/MB	12
消息产生间隔/s	25~35
消息大小/MB	0.5~1
消息初始副本数	16
传输速率/Kbps	250
传输范围/m	10
$\tau$	0.8
$U_{\text{th}}$	0.1
$P_{\text{th}}$	0.5
$\alpha$	0.8
$w$	0.3

## 4.2 评价指标

### 1) 投递率

投递率=成功投递到目标节点的消息数量/网络中源节点产生的消息数量.

### 2) 平均时延

平均时延=网络中所有被成功投递的消息从源节点产生开始到目标节点完整接收到所经过的平均时间.

### 3) 网络开销

网络开销.冗余的中继转发次数/成功投递到目标节点的消息数量.

### 4) 平均跳数

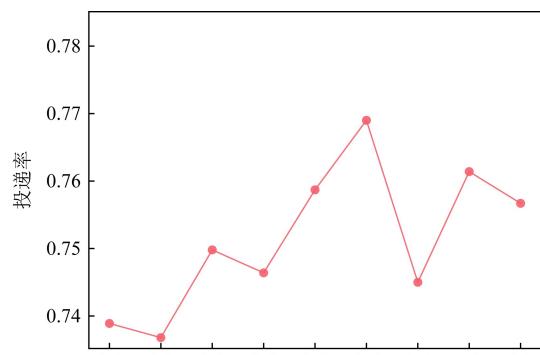
平均跳数.网络中所有被成功投递的消息从源节点产生开始到目标节点完整接收到所经过的平均跳数.

## 4.3 参数设置实验

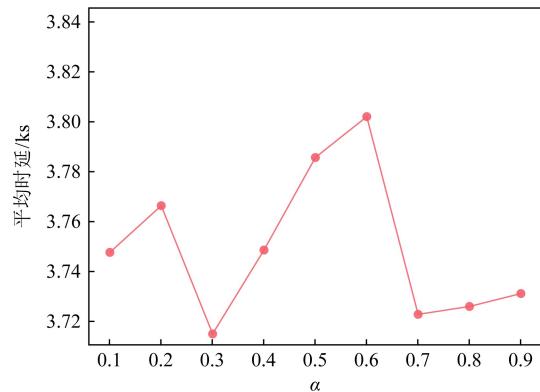
### 4.3.1 活跃度平滑因子 $\alpha$ 设置

$\alpha$  的取值对 CPN-ASW 算法的实验效果影响较

大,在表 1 参数设置条件下更改  $\alpha$  取值,综合考虑投递率和平均时延,  $\alpha=0.8$  时实验效果最佳.如图 1 所示:



(a)  $\alpha$  不同取值下的投递率比较



(b)  $\alpha$  不同取值下的平均时延比较

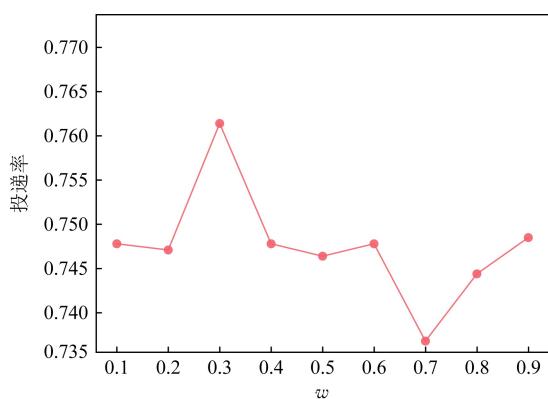
Fig. 1 Performance comparison of CPN-ASW under different  $\alpha$

图 1 CPN-ASW 在  $\alpha$  不同取值下的性能比较

从图 1(a) 中可以观察到,  $\alpha=0.6$  时投递率最高,根据 Markov 预测法<sup>[19]</sup>,任何事件可通过其目前的状况来预测该事件将来各个时刻(或时期)的变动状况.由此可知近期活跃度高的节点在未来一段时间内也将延续较高的活跃性.当  $\alpha$  取值偏重于最近 1 个完整时间窗口活跃度时,更能准确预测节点在未来一段时间的活跃度.但是同时观察图 1(b),  $\alpha=0.6$  时投递率最高,此时平均时延也最高.所以综合考虑投递率和平均时延,  $\alpha=0.8$  时投递率比  $\alpha=0.6$  时要略低,但平均时延有了明显的改善,所以本文中  $\alpha=0.8$  时实验效果最佳.

### 4.3.2 相对效用值参数 $w$ 设置

$w$  的取值直接影响消息副本分配数量的合理性,本文通过在  $w$  不同取值下的投递率变化,得出  $w=0.3$  时投递率最高,如图 2 所示:

Fig. 2 Delivery ratio of CPN-ASW under different  $w$ 图 2 CPN-ASW 在  $w$  不同取值下的消息投递率比较

#### 4.4 算法比较及分析

为了验证 CPN-ASW 算法的高效性,进行了 2 组仿真实验,在不同的消息生存周期(TTL)、不同的节点缓存大小下对这 5 种算法在消息投递率、平均时延、网络开销、平均跳数这 4 个方面进行评估。实验结果表明 CPN-ASW 在提高消息投递率、控制网络开销、降低平均时延方面效果显著。

##### 1) 不同消息生存周期

在 4.1 节的环境配置下,改变消息生存周期,对比各种算法在不同消息生存周期下消息投递率(如图 3 所示)、平均时延(如图 4 所示)、网络开销(如图 5 所示)、平均跳数(如图 6 所示)这 4 个方面的变化情况。

图 3 显示了各种算法在不同消息生存周期下消息投递率的变化情况。其中 CPN-ASW 算法的投递率最高,相比 Epidemic 平均提高了 58.55%,相比 SaW 平均提高了 3.60%,相比 EBR 平均提高了 1.20%,相比 PBSW 平均提高了 11.22%。这主要归

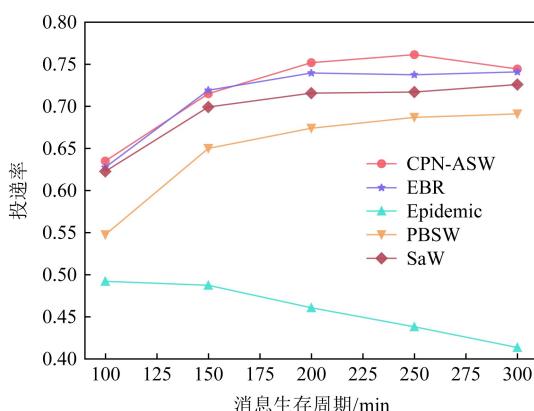


Fig. 3 Delivery ratio under different TTL

图 3 不同消息生存周期下的投递率

结于 2 个原因:1) CPN-ASW 算法在 Spray 阶段根据节点间相似程度是否超过阈值而采用不同的中继节点选择策略,并根据节点间相对效用值实现自适应分配消息副本数,避免消息的盲目转发;2) 在 Wait 阶段实现消息主动转发,选择到目标节点投递预测值更高的中继节点进行转发。同时从实验结果可以看出, Epidemic 算法的投递率远低于其他算法,并且曲线呈现下降趋势,这是因为此算法没有限制消息副本的最大拷贝数,随着消息生存周期的增加,消息在有限缓存中驻留时间更长,盲目洪泛易导致网络中挤压大量消息副本,引起网络拥塞,消息投递率反而下降。

图 4 评估了各个算法平均时延的变化情况。5 种算法的平均时延随着消息生存周期的增加大体上呈现增长态势,因为增加生存周期使原来在短时间内不能到达目标节点的消息能被成功投递。其中 CPN-ASW 的平均时延最小,相比 SaW 下降了 4.44%,相比 EBR 下降 2.08%,相比 PBSW 下降 9.67%。这主要是因为 CPN-ASW 算法在 Wait 阶段实现了消息主动转发,选择更优的中继节点进行传输,加快了消息到达目标节点的速度。Epidemic 算法采用洪泛机制,过多的复制消息容易使网络陷入拥塞,故平均时延相比其他算法来说是最大的。

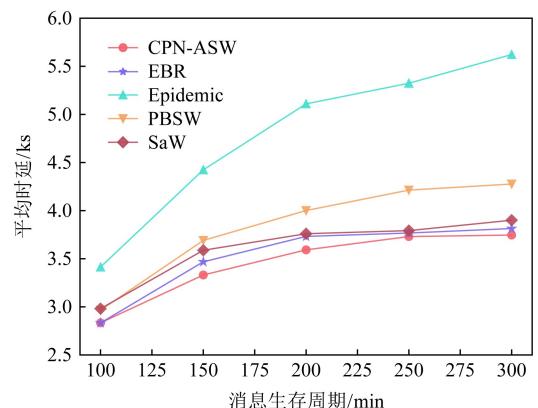


Fig. 4 Average delay under different TTL

图 4 不同消息生存周期下的平均时延

图 5 比较各个路由算法的网络开销。由于 SaW, EBR, PBSW, CPN-ASW 均在一开始限制了消息副本拷贝数,也就是限制了转发次数,故网络开销远低于 Epidemic 算法。CPN-ASW 算法网络开销相对 SaW 平均下降 17.71%,这是因为相对 SaW 算法在 Spray 阶段盲目转发,CPN-ASW 算法在 Spray 阶段避免消息在运动范围相似的节点间转发,在一定程度上减少了不必要的中继转发次数,在 Wait 阶段自适应

控制转发条件,最终网络开销较小.从图 5 中还可看出,PBSW 算法的网络开销最小,其原因主要体现在 2 个方面:1)Spray 阶段只将消息转发给与目的节点相遇概率更高的节点,从而消息转发次数受到限制;2)在 Wait 阶段没有任何优化策略,直到与目的节点相遇时才将消息转发出去,不存在冗余的中继转发次数.

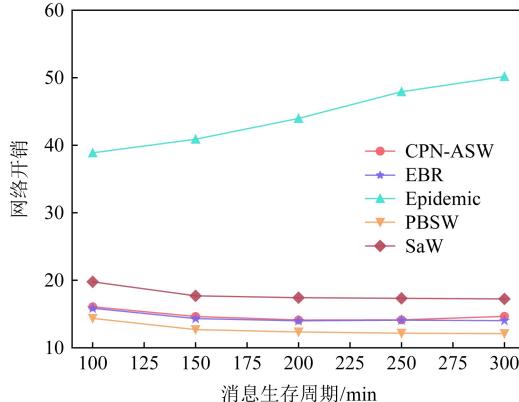


Fig. 5 Overhead ratio under different TTL

图 5 不同消息生存周期下的网络开销

图 6 比较了各个算法的平均跳数.其中 CPN-ASW 相对 SaW 平均跳数下降 4.22%,主要原因在于 CPN-ASW 在 Spray 阶段当节点间运动范围高度相似时选择更优的中继节点分配消息副本数,在 Wait 阶段虽然进行主动转发,但是自适应控制转发条件,最终平均跳数还是明显低于 SaW.其中 PBSW 的平均跳数最低,有 2 个原因:1)Spray 阶段对中继节点的选择更为苛刻,只给到目的节点投递预测值更高的中继节点分配消息副本;2)Wait 阶段被动等待,直到遇到目的节点才进行消息投递.

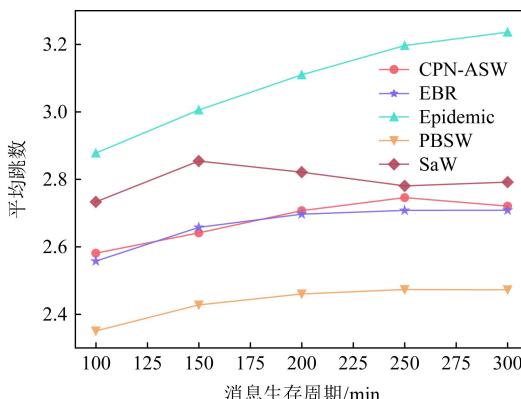


Fig. 6 Average hop under different TTL

图 6 不同消息生存周期下的平均跳数

总体来看,在不同的消息生存周期下,CPN-

ASW 相比其他 4 种算法均表现出较高的投递率和较低的平均时延.并且在网络开销和平均跳数方面较 SaW 也表现出明显的优势.PBSW 算法虽然在网络开销和平均跳数方面表现最优,但是相对于其他 3 种限制消息副本数的算法(SaW, EBR, CPN-ASW)来说,投递率和平均时延方面均表现出明显的劣势.另外也可看出,Epidemic 算法性能最差,说明在资源受限的真实场景中,洪泛式的转发算法并不能取得良好的效果.

## 2) 不同节点缓存大小

在不同节点缓存大小下,比较和分析各个算法在消息投递率(如图 7 所示)、平均时延(如图 8 所示)、网络开销(如图 9 所示)、平均跳数(如图 10 所示)这 4 个方面的变化情况.

图 7 显示了各个算法在不同节点缓存下消息投递率的变化情况.5 种算法的投递率都随着节点缓存的增加而提高,这是因为节点缓存越大,其所能携带的消息数量也越多,由于网络拥塞导致的丢包情况会逐渐缓解,其中 CPN-ASW 算法的投递率最高,相比 Epidemic 平均提高了 67.07%,相比 SaW 平均提高了 5.28%,相比 EBR 平均提高了 2.55%,相比 PBSW 平均提高了 9.12%.

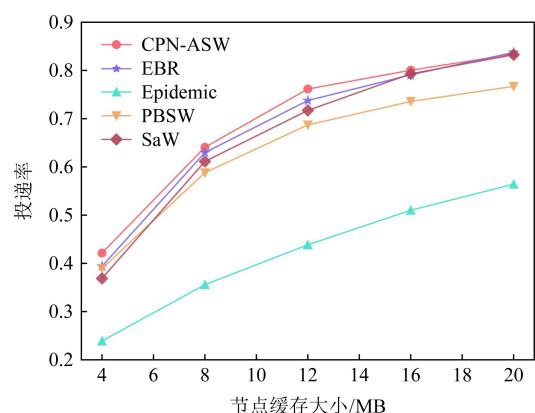


Fig. 7 Delivery ratio under different buffer sizes

图 7 不同缓存大小下的投递率

图 8 比较了各个算法的平均时延.CPN-ASW 的平均时延相比 PBSW 平均下降 10.26%.从图 8 中还可观察到,在节点缓存小于 12 MB 时,CPN-ASW 的平均时延略大于 SaW;当节点缓存超过 12 MB,CPN-ASW 的平均时延开始小于 SaW 的时延.这是由于 CPN-ASW 在 Spray 阶段是按照节点间的相对效用值分配消息副本数,相对效用值综合考虑节点活跃度和节点剩余缓存,当节点缓存本身较小时,节点的缓存利用率一直都很高,相对效用值中活跃度

因素的重要性更为突出,活跃度高的节点易形成拥塞,消息丢包次数变多,需要经过更多的转发次数才能到达目标节点,传输时延较大;随着节点缓存的增加,活跃度高的节点上拥塞情况得到缓解,再加上CPN-ASW在Wait阶段选择更优的中继节点进行转发,最终CPN-ASW在平均时延方面的优越性得到体现.

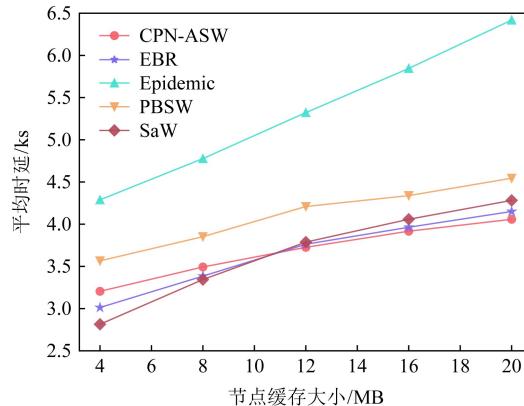


Fig. 8 Average delay under different buffer sizes

图 8 不同缓存大小下的平均时延

图 9 显示的是网络开销随节点缓存的变化情况.当节点缓存逐渐增加,5 种算法的网络开销都逐渐下降,这是因为缓存空间越小,节点就越有可能因为缓存占满采用消息丢弃策略,使消息经过较多的转发次数才能到达目标节点.其中 CPN-ASW 的网络开销较 Epidemic 平均下降了 69.55%,较 SaW 平均下降了 17.99%.此外,通过观察图 9 可知,Epidemic 算法的网络开销随着缓存的增加而急速下降,这表明此算法适用于资源充足的网络,在缓存较充足时能有效降低网络开销,发挥出较好的效果.

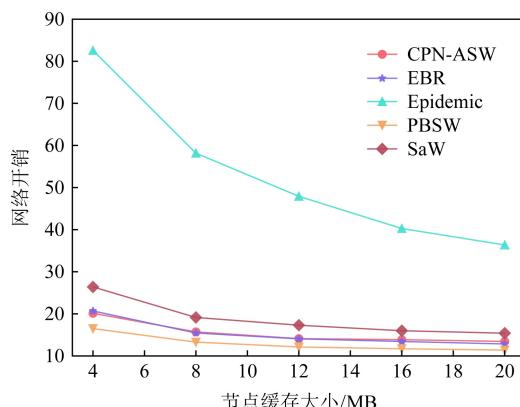


Fig. 9 Overhead ratio under different buffer sizes

图 9 不同缓存大小下的网络开销

图 10 比较各个算法的平均跳数.其中 PBSW 的平均跳数最低,CPN-ASW 平均跳数和 EBR 算法大致相同,CPN-ASW 的平均跳数相对 Epidemic 平均下降了 17.36%,相对 SaW 平均下降了 2.50%.这说明了 CPN-ASW 算法采用的中继节点选择的合理性.

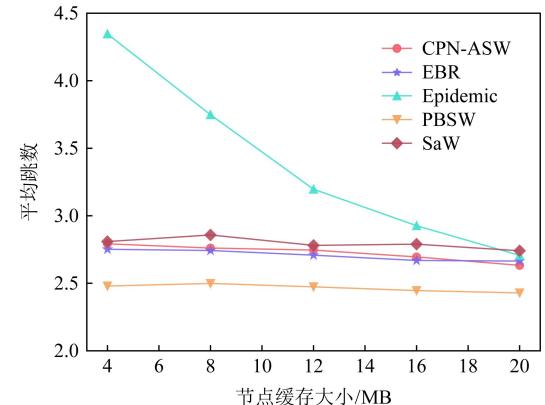


Fig. 10 Average hop under different buffer sizes

图 10 不同缓存大小下的平均跳数

总体来看,在 5 种算法中,在不同节点缓存大小下,CPN-ASW 算法消息投递率最高,当缓存较充足时平均时延最小,且在网络开销和平均跳数方面也表现优越且稳定.

## 5 结束语

本文针对 DTN 长时延、间歇性连接等特点,提出一种基于节点综合性能的喷射等待路由算法 CPN-ASW.该算法在 Spray 阶段根据节点间相似度是否超过给定阈值采用不同中继节点选择策略,并按照节点相对效用值自适应分配消息副本数;在 Wait 阶段将消息转发给到目标节点投递预测值更高的中继节点.由对比算法分析得出,本文提出算法能够有效提高消息投递率,控制网络开销和降低平均时延.

然而,从实验中也可观察到,CPN-ASW 算法相对于传统 SaW 来说,在节点缓存较小时平均时延较高.为进一步降低平均时延,设置一种有效的缓存管理策略是下一步研究工作的重点.

**作者贡献声明:** 崔建群负责提出研究选题、优化论文;孙佳悦负责设计算法、设计实验并开展实验分析;常亚楠负责设计论文框架;余东海、邬尧负责调研整理文献、协助论文实验;吴黎兵负责提出指导意见并修改论文.

## 参 考 文 献

- [1] Xiao Mingjun, Huang Liusheng. Delay-tolerant network routing algorithm [J]. Journal of Computer Research and Development, 2009, 47(7): 1065–1073 (in Chinese)  
(肖明军, 黄刘生. 容迟网络路由算法[J]. 计算机研究与发展, 2009, 47(7): 1065–1073)
- [2] Wang Hezhe, Wang Huiqiang, Zhu Jinmei, et al. OCIGM: An optimized control information generation method for DTN routing [J]. Journal of Beijing University of Posts and Telecommunications, 2017, 40(1): 79–83 (in Chinese)  
(王贺哲, 王慧强, 朱金美, 等. OCIGM: 面向 DTN 路由的优化控制信息生成方法[J]. 北京邮电大学学报, 2017, 40(1): 79–83)
- [3] Trono E M, Fujimoto M, Suwa H, et al. Generating pedestrian maps of disaster areas through ad-hoc deployment of computing resources across a DTN [J]. Computer Communications, 2017, 100(1): 129–142
- [4] Cuka M, Shinko I, Spaho E, et al. A simulation system based on ONE and SUMO simulators: Performance evaluation of different vehicular DTN routing protocols [J]. Journal of High Speed Networks, 2017, 23(1): 59–66
- [5] Tornell S M, Calafate C T, Cano J C, et al. DTN protocols for vehicular networks: An application oriented overview [J]. IEEE Communications Surveys & Tutorials, 2015, 17(2): 868–887
- [6] Xiao Mingjun, Wu Jie, Huang Liusheng. Community-aware opportunistic routing in mobile social networks [J]. IEEE Transactions on Computers, 2014, 63(7): 1682–1695
- [7] Spyropoulos T, Psounis K, Raghavendra C S. Spray and wait: An efficient routing scheme for intermittently connected mobile networks [C] //Proc of ACM SIGCOMM Workshop on Delay-Tolerant Networking. New York: ACM, 2005: 252–259
- [8] Yu Chen, Tu Zhongqiu, Yao Dezhong, et al. Probabilistic routing algorithm based on contact duration and message redundancy in delay tolerant network [J]. International Journal of Communication Systems, 2016, 29(16): 2416–2426
- [9] Li Jianbo, Jiang Shan, Song Youmei, et al. A multi-scheme adaptive routing algorithm based on spray and wait for delay tolerant networks [J]. International Journal on Smart Sensing and Intelligent Systems, 2015, 8(4): 2136–2158
- [10] Nelson S C, Bakht M, Kravets R. Encounter - based routing in DTNs [C] //Proc of IEEE INFOCOM. Piscataway, NJ: IEEE, 2009: 846–854
- [11] Spyropoulos T, Psounis K, Raghavendra C S. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility [C] //Proc of the 5th Annual IEEE Int Conf on Pervasive Computing and Communications Workshops. Piscataway, NJ: IEEE, 2007: 79–85
- [12] Jones E P C, Li L, Schmidtko J K, et al. Practical routing in delay-tolerant networks [J]. IEEE Transactions on Mobile Computing, 2007, 6(8): 943–959
- [13] Vahdat A, Becker D. Epidemic routing for partially connected ad hoc networks [R]. Durham, North Carolina: Duke University, 2000
- [14] Lindgren A, Doria A, Schelen O. Probabilistic routing in intermittently connected networks [J]. ACM SIGMOBILE Mobile Computing and Communications Review, 2003, 7(3): 19–20
- [15] Zhao Yuhong, Yin Zili, Zhang Xiaolin. Spray and wait routing protocol of node sociality in opportunistic network [J]. Computer Simulation, 2018, 35(7): 231–236 (in Chinese)  
(赵宇红, 尹自立, 张晓琳. 基于节点社会性的机会网络喷雾等待路由协议[J]. 计算机仿真, 2018, 35(7): 231–236)
- [16] Cui Jianqun, Cao Shuqin, Chang Yanan, et al. An adaptive spray and wait routing algorithm based on quality of node in delay tolerant network [J]. IEEE Access, 2019, 7: 35274–35286
- [17] Kim E H, Nam J C, Choi J I, et al. Probability-based spray and wait protocol in delay tolerant networks [C] //Proc of Int Conf on Information Networking. Los Alamitos, CA: IEEE Computer Society, 2014: 412–416
- [18] Keränen A, Ott J, Kärkkäinen T. The ONE simulator for DTN protocol evaluation [C/OL] //Proc of the 2nd Int Conf on Simulation Tools and Techniques. New York: ACM, 2009 [2020-06-01]. <https://dl.acm.org/doi/pdf/10.4108/ICST.SIMUTOOLS2009.5674>
- [19] Zhang Yang, Wang Xiaoming, Lin Yaguang, et al. Message forwarding strategy based on Markov decision process in opportunistic networks [J]. Journal of Frontiers of Computer Science & Technology, 2016, 10(1): 82–92 (in Chinese)  
(张杨, 王小明, 林亚光, 等. 基于马尔可夫决策过程的机会网络转发策略[J]. 计算机科学与探索, 2016, 10(1): 82–92)



**Cui Jianqun**, born in 1974. PhD, professor. Member of CCF. Her main research interests include opportunistic networks, the Internet of things, mobile networks, and application layer multicast.

**崔建群**, 1974 年生. 博士, 教授, CCF 会员. 主要研究方向为机会网络、物联网、移动网络、应用层组播.



**Sun Jiayue**, born in 1996. Master. Her main research interests include delay tolerant networks, social networks and wireless communication.

**孙佳悦**, 1996 年生. 硕士. 主要研究方向为延迟容忍网络、社交网络和无线通信.



**Chang Yanan**, born in 1984. PhD, associate professor. Her main research interests include wireless mesh networks, delay tolerant networks, and social networks.  
常亚楠,1984年生.博士,副教授.主要研究方向为无线mesh网络、延迟容忍网络、社交网络.



**Yu Donghai**, born in 1997. Master. His main research interests include delay tolerant networks, social networks and wireless communication.  
余东海,1997年生.硕士.主要研究方向为延迟容忍网络、社交网络和无线通信.



**Wu Libing**, born in 1995. Master. His main research interests include delay tolerant networks, social networks and wireless communication.  
邬尧,1995年生.硕士.主要研究方向为延迟容忍网络、社交网络和无线通信.



**Wu Libing**, born in 1972. PhD, professor. Member of CCF. His main research interests include network communication, grid computing, and the Internet of things.  
吴黎兵,1972年生.博士,教授,CCF会员.主要研究方向为网络通信、网格计算、物联网.

## 2022 年《计算机研究与发展》专题(正刊)征文通知 ——数据安全与智能隐私保护研究

人工智能随着大数据、云计算和物联网技术的蓬勃发展,已广泛渗透到科学研究、数字经济、健康与医疗卫生等各大领域,给人类生产生活带来了极大的便利.机器学习的模型训练和推理预测都依赖于大规模数据集,而这些数据中可能包含着用户的敏感或隐私信息.针对人工智能中日益加剧的数据安全与隐私保护需求,国内外学者获得了一系列重要研究成果.然而,如何在保护用户数据安全的同时,实现人工智能技术的高可用性和高效性仍然是学术界和工业界广泛关注的具有挑战性的研究课题.

为进一步推动我国学者在数据安全与智能隐私保护领域的研究,及时报道我国学者在数据安全与智能隐私保护方面的最新研究成果,《计算机研究与发展》将于 2022 年 10 月出版网络与信息安全专题——数据安全与智能隐私保护研究,主要聚焦数据安全理论基础与应用及智能隐私保护关键支撑技术的研究,欢迎相关领域的专家学者和科研人员踊跃投稿.

### 征文范围(但不限于)

- 密码学与数据安全基础理论
- 机器学习隐私保护技术
- 数据系统安全
- 人工智能系统的攻击、检测与防御技术
- 面向智能隐私保护的密码算法与协议
- 云计算安全
- 分布式系统与边缘计算隐私保护
- 人工智能的应用安全与隐私保护

### 征文要求

1) 论文应属于作者的科研成果,数据真实可靠,具有重要的学术价值或推广应用价值,未在国内外公开发行的刊物或会议上发表过,不存在一稿多投问题.作者在投稿时,需向编辑部提交版权转让协议.

2) 论文一律用 word 格式排版,格式体例参考近期出版的《计算机研究与发展》文章的要求(<http://crad.ict.ac.cn>).

3) 论文请通过期刊网站(<http://crad.ict.ac.cn>)进行投稿,投稿时提供作者的联系方式,并在作者留言中注明“网络与信息安全 2022 专题”(否则按自由来稿处理).

### 重要日期

征文截止日期: 2022 年 6 月 10 日

录用通知日期: 2022 年 7 月 20 日

作者修改稿提交日期: 2022 年 8 月 5 日

出版日期: 2022 年 10 月

### 特邀编委

徐秋亮 教授 山东大学 xql@sdu.edu.cn

张玉清 教授 中国科学院大学、国家计算机网络入侵防范中心 zhangyq@ucas.ac.cn

董晓蕾 教授 华东师范大学 dongxiaolei@sei.ecnu.edu.cn

### 联系方式

编辑部: crad@ict.ac.cn, 010-62620696, 010-62600350

通信地址: 北京 2704 信箱《计算机研究与发展》编辑部

邮 编: 100190