

机会网络中基于节点相遇间隔的缓存管理策略

张 峰

(山西大学计算机科学与技术学院 太原 030006)

摘 要 采用存储-携带-转发方式的机会网络不考虑消息在传递过程中经历节点与目的节点的相遇概率,对消息传播状态的估计结果存在较大误差。文中提出一种基于节点相遇间隔的缓存管理策略,利用节点相遇间隔服从指数分布的特性,结合消息在网络中的副本数量,估计出消息的平均投递概率。在节点缓存溢出时,根据估算的消息投递概率对缓存中的消息进行排序,优先删除投递概率低的消息,达到缓存管理的目的。仿真结果表明,使用该方法对节点缓存进行管理,在提高网络中消息投递成功率的同时,有效降低了消息的平均投递延迟和网络负荷。

关键词 机会网络,缓存管理,节点相遇间隔,消息平均投递概率

中图分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.05.008

Node Encounter Interval Based Buffer Management Strategy in Opportunistic Networks

ZHANG Feng

(School of Computer & Information Technology, Shanxi University, Taiyuan 030006, China)

Abstract Opportunistic network which employs store-carry-and-forward pattern doesn't consider encounter possibility of destination nodes and passing nodes during message transmission, leading to a large deviation for the estimation on message transmission status. This paper proposed an encounter interval based buffer management strategy. It facilitates the characteristic of exponential distribution for the encounter intervals between nodes, and the number of message copies is also considered. So the average delivery probability is estimated. All messages stored in the buffer are sorted according to the average delivery probability when the buffer is overflow, and the messages with low delivery probability will be dropped first for buffer management intent. Simulation results show that the proposed strategy achieves better performance compared to existing schemes in respect of the delivery ratio, average latency and overhead ratio.

Keywords Opportunistic networks, Buffer management, Encounter interval, Average delivery probability

1 引言

机会网络是一种不需要提前进行网络部署,也不需要利用运营商提供的基础网络设施就可以完成通信功能的移动自组织网络^[1]。与传统网络相比,在机会网络中进行消息的路由转发时不需要存在源节点到目的节点的完整传输路径,而是利用网络中节点在移动过程中带来的相遇机会,在相遇节点之间进行消息的逐跳转发,最终将消息投递到目的节点。在一些特殊场景(如突发事件^[2]、灾难救助、野生动物追踪、智能交通等)下,机会网络能够利用其固有的特性发挥重要的作用。

机会网络中的节点一般由人携带的便携式无线通信设备构成,如智能手机、平板电脑、车载通信设备等,这些设备的典型特点是体积小、便于携带、通过电池供电、通信带宽和存储空间有限。上述特性以及网络特有的消息转发方式决定了机会网络中消息传输的延时比较大,可靠性比较低,误码率比较高。在这种情况下,主要通过两种方式来提高网络中消息的投递成功率^[3]:一种是路由策略,另一种是缓存管理策略。路

由策略主要解决消息在网络中转发的时机以及转发过程中中间节点的选择问题;而缓存管理策略主要解决在节点缓存空间有限的情况下,如何在缓存空间溢出时对缓存中的消息进行管理的问题。两者互为补充和促进,最终达到提高网络中消息投递成功率的目的。

机会网络中存储-携带-转发的路由方式决定了消息在没有到达目的节点之前,需要长时间缓存在中间节点的本地缓存空间中。长时间的消息缓存以及多副本的路由策略导致节点的通信带宽和存储负载率都非常高。高效的缓存管理策略可以在提高消息投递率的同时,尽可能地降低消息投递延迟。因为消息投递延迟越大,在网络中可能存在的消息副本数和携带消息副本的节点数就越多,需要消耗更多的网络资源,包括节点的缓存资源和消息转发过程中的带宽资源。为了减少网络资源的消耗,尽可能地降低网络负载,需要将消息尽快地投递到目的节点,即降低消息的投递延迟。

进行缓存消息调度和丢弃时主要考虑消息的剩余 TTL (time-to-live)、消息经历节点的数量、消息的复制次数、消息的大小、消息投递代价和距离目的节点的远近等因素^[3]。

收稿日期:2018-03-24 返修日期:2018-07-22

张 峰(1978—),男,博士,讲师,主要研究方向为无线自组织网络、机会网络,E-mail:zhangfeng@sxu.edu.cn(通信作者)。

Dropleast-recently-received (DLR)和 drop oldest (DOA)策略分别根据节点收到消息的时间和消息产生时间对消息进行排序^[4],当节点缓存拥塞时,依次将排序以后的消息丢弃,以释放节点的缓存空间,直到节点上有足够的缓存空间来接收新的消息。基于复制率的缓存管理策略(drop copy ratio,DCR)是根据消息副本的数量和该消息的剩余生存时间来计算消息的复制率,并按照消息的复制率对消息进行排序,在缓存溢出时从队列中依次删除复制率最高的消息,从而达到缓存管理的目的^[5]。在对历史信息进行收集的基础上,安莹等^[6]提出了一种基于节点介数的拥塞检测机制,通过对历史信息的收集和统计来均衡节点之间的开销,避免了个别节点资源消耗过度的问题。文献^[7]在研究自私性机会网络中的节点激励策略时也对网络中节点的缓存进行了管理,文献^[8]中给出了一种结合固定中继节点重要性和消息相关性的缓存管理策略。然而,上述文献提到的缓存管理策略都是基于节点自身存储的历史信息来进行的,没有考虑消息在传递过程中经历节点与目的节点的相遇概率,对消息在网络中的传播状态估计存在一定的偏差。

针对上述缓存管理策略存在的不足,本文通过记录各节点之间相遇的间隔,并估计消息在传递过程中所经历的各个中间节点在消息剩余 TTL 时间内遇到目的节点的可能性,估算出消息在网络中的投递成功概率以及消息在每个节点上的平均投递概率。当节点缓存溢出时,优先丢弃平均投递概率低的消息,以缓存其他更可能投递到目的节点的消息,从而提高节点的缓存利用率,提升网络的整体性能。

2 基于相遇间隔的缓存管理策略

2.1 节点相遇间隔

节点相遇时刻是指节点在网络中移动时,两个刚刚进入彼此无线模块通信范围内、互相可以通过其无线通信模块建立无线通信连接的时刻,即这两个移动节点之间的距离小于其无线通信模块通信半径的时刻。当这两个节点之间的距离大于其无线通信模块的通信半径时,该无线连接就会断开,之后才能继续相遇。我们定义机会网络中节点相遇间隔为相同两个节点之间连续两次相遇时刻之间的差值。

节点之间的相遇间隔与节点的移动方式有关,即与已有文献中所提到的节点移动模型相关。文献^[9]中的研究表明,在 RWP(random way point)移动模型下,节点之间的相遇间隔是一个服从参数为 λ 的指数分布的随机变量 T 。

指数分布的概率密度函数如式(1)所示:

$$f(t) = \begin{cases} \lambda e^{-\lambda t}, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (1)$$

其中, $\lambda = \frac{1}{E(T)}$, $E(T)$ 为随机变量 T 的数学期望。

2.2 消息平均投递概率

在机会网络中,节点之间的消息投递是通过节点在移动过程中的相遇来完成的。而网络中每个消息都是有时效性的,当节点每次产生消息时,通常都会赋予每个消息一个时效值,也就是消息的生存时间 TTL 值。网络中的消息只有在其生存时间内投递到目的节点才有意义,这就要求缓存有消息

的节点在消息的生存时间内和该消息的目的节点相遇,否则消息就会由于时效性的缘故被节点丢弃。如果消息 M 只缓存存在其产生节点(源节点) s 上,不被复制到其他相遇的邻居节点上,则消息 M 在 RWP 移动模型下投递成功的概率等同于节点 s 与该消息目的节点在消息 M 有效生存时间 t 内相遇的概率。因此,消息 M 的投递成功率可以表示为式(2)所示的 $P_s(t)$:

$$P_s(t) = 1 - e^{-\lambda t} \quad (2)$$

在 Epidemic 路由策略下,节点会不断地将其缓存中的消息复制到每个不携带相同消息的相遇节点上,如果其中一个消息 M 在其剩余生存时间 t 内被复制到网络中 i 个不同的相遇节点上,其中 $i < N-1$, N 表示网络中节点的总数, $N-1$ 表示该消息被复制到网络中除该消息的目的节点以外的其他所有节点上。在 RWP 移动模型下,由于网络中每个节点和该消息目的节点的相遇间隔都服从指数分布,其相遇间隔的数学期望分别表示为 $E_1, E_2, \dots, E_i$,并且每个节点在网络中的移动是彼此独立的,只需要其中某个节点在消息 M 的剩余生存时间内和目的节点相遇,就能完成该消息的投递。因此可以将消息 M 的总投递成功率表示为:

$$\begin{aligned} P(t) &= 1 - P\{T > t\} \\ &= 1 - P\{T_1 > t, T_2 > t \dots T_i > t\} \\ &= 1 - P\{T_1 > t\} P\{T_2 > t\} \dots P\{T_i > t\} \\ &= 1 - [1 - P_1(t)] [1 - P_2(t)] \dots [1 - P_i(t)] \\ &= 1 - e^{-\left(\frac{1}{E_1} + \frac{1}{E_2} + \dots + \frac{1}{E_i}\right)t} \end{aligned} \quad (3)$$

由于消息 M 的投递成功率 $P(t)$ 与所有缓存该消息的节点都有关系,因此缓存该消息的每个节点能够投递成功的概率 D_m 可以表示为:

$$D_m = \frac{P(t)}{i} \quad (4)$$

其中, i 表示消息被复制到中间节点的个数, D_m 也称为消息的平均投递概率。

节点上计算出的消息平均投递概率 D_m 越小,说明节点对该消息投递成功的贡献越低,节点是否缓存该消息对整个网络的消息投递成功率影响不大。当节点缓存溢出时,可以优先丢弃 D_m 值较小的消息,以释放缓存空间来存储其他消息,从而提高网络的整体消息投递性能。

2.3 消息副本的统计以及删除

节点在产生每个消息时,将该消息的属性(包括消息 ID、消息产生的源节点 ID、消息目的节点 ID、消息初始 TTL 值等)信息作为消息的一部分,和消息一起在网络中的节点之间进行传递^[10]。为了精确统计每个消息在网络中不同节点之间的复制次数,以及消息每次被复制到的中间节点和该消息目的节点之间的相遇间隔,将每个消息成功复制到中间节点之后,在其尾部添加该中间节点的信息,用来记录该消息在传输过程中的路径信息。该中间节点的信息包括中间节点的 ID、中间节点和消息目的节点的相遇间隔以及相遇间隔的更新时间,根据这些信息就可以利用式(3)计算出每个消息的投递成功率以及该消息的平均投递概率。相比于消息本身所占用的空间大小,消息尾部添加的中间节点信息所占用的空间可以忽略不计,对整个消息的传输没有影响。另外,为了及时

删除网络中无用的消息副本,减小对其他消息投递的影响,针对每个成功投递到目的节点的消息,在其目的节点上产生该消息的一个 ACK 信息在网络中传输。每当两个节点建立完无线连接之后,它们首先将其携带的 ACK 信息传递给对方,并将收到的 ACK 信息和本地保存的 ACK 信息进行合并,同时删除自身缓存中对应 ACK 编号的消息,以此来提高节点的缓存利用率。

2.4 缓存管理策略的算法流程

假设网络中任意两个节点 A 和 B 在移动过程中相遇,其处理和交互消息的流程如下:

1) 记录本次相遇的时刻和这两个节点上次相遇时刻的时间差,并将该差值作为本次的相遇间隔,然后结合前面的统计信息重新计算这两个节点相遇间隔的数学期望。

2) 每个节点都将其保存的 ACK 信息发送给对方,同时在本地删除自身缓存中对应 ACK 编号的消息。

3) 将节点 A 上的每个消息与节点 B 上的消息进行比较,对于两个节点上都存储的消息 m ,互相交互并更新其传递路径字段,使得两个节点对相同消息的传递路径字段保持一致。

4) 在每个节点上计算本地存储的每个消息的平均投递概率,并根据平均投递概率的大小对缓存中的消息进行排序。

5) 节点 A 依次将排序后的消息发往节点 B ,节点 B 只接收其缓存中没有的消息。

6) 如果节点 B 剩余的存储空间能够容纳节点 A 即将发来的消息,那么节点 B 接收该消息并将其存储在节点缓存中。

7) 如果节点 B 剩余的存储空间不足以容纳节点 A 即将发来的新消息,此时需要将新消息的平均投递概率和节点 B 排序后队列尾部消息的平均投递概率进行比较。

8) 节点 B 依次删除其队列尾部的消息,直到节点 B 的剩余存储空间足以接收新消息或者队列尾部消息的平均投递概率大于新消息的平均投递概率。

9) 对于节点 A 成功发送到节点 B 上的消息,两个节点都需要重新修正该消息尾部的中间节点信息:在中间节点信息中加入节点 B 、节点 B 和该消息目的节点相遇间隔的期望,以及该中间节点信息的更新时间。

10) 节点 A 按上述步骤对本地存储的所有消息依次进行处理,直到节点 A 和节点 B 之间的无线连接被中断或节点 A 上存储的消息传递完毕。

对于算法的复杂度,因为要计算网络中消息在传递过程中所经历中间节点和消息目的节点之间的相遇间隔,而消息传递过程中的中间节点不可提前预知,每个节点都需要同时存储本节点到网络中其他节点的相遇间隔以及网络中任意两个节点之间的相遇间隔,所以算法的存储复杂度为 $O(n^2)$,其中 n 表示网络中节点的数量。计算每个消息平均投递概率的时间复杂度为 $O(m)$,其中 m 表示该消息在传递过程中经历中间节点的数量,其最大可能取值为 $n-1$,表示该消息被复制到了除消息目的节点以外的所有其他节点上。

3 性能评估

本节采用 ONE(opportunistic network environment)^[11] 仿真工具来对本文提出的基于节点相遇间隔的缓存管理策略

(EIBM) 的有效性进行验证,并将其与 DLR, DOA, DCR 在消息投递成功率、网络负载率以及消息平均投递延迟方面的性能进行比较。仿真采用的路由机制为 Epidemic 路由,因为在这种路由方式下,消息的复制率和网络的负载都是最大的,更容易造成节点缓存溢出,因此用它来验证缓存管理策略的效果更加明显。

3.1 仿真环境

在进行仿真实验时, ONE 仿真环境中用到的关键仿真参数的设置如表 1 所列。

表 1 实验参数的设置

Table 1 Setting of parameters

参数	值
节点移动速度/(m/s)	5~10
节点缓存大小/MB	10
节点移动模型	Random Waypoint
节点通信范围/m	100
节点等待时间/s	0~120
无线传输速率/(kB/s)	250
消息大小/kB	500~1024
消息产生间隔/s	15~25
消息生存时间 TTL/s	3600
仿真时间/s	20000
仿真区域/m ²	4500×3400

在每次仿真实验之前,先运行 10000s 的时间来对节点之间的相遇间隔进行学习。相同实验环境下,分别统计 4 种策略在不同节点数量和不同缓存空间两种场景下的性能指标,并进行分析与比较。

1) 消息投递成功率:在仿真结束后,所有被投递到目的节点的消息个数与仿真期间所有节点产生消息总数的比值;同一消息被多次复制到不同中间节点时,在该指标统计中将其看作是一个消息。

2) 网络负载率:网络中所有节点上消息传输的总次数与成功投递到消息目的节点的消息数量的比值。

3) 平均投递延迟:所有消息从源节点产生到被成功投递到目的节点之间的时间差值的平均值。

3.2 不同节点数量下的性能评估

在本节的仿真实验里,每个节点的缓存空间都设置为 10 MB,其他仿真参数如表 1 所列。从图 1 中可以看到,随着网络中节点数量的增加,DOA,DCR,EIBM 策略下的消息投递成功率曲线都呈现上升的趋势,而 DLR 策略下的投递成功率没有明显变化。这是因为网络中节点数量增加以后,在节点移动过程中能够遇到其他节点的机会也随之增加,消息有更多的机会被复制到其他的中间节点上。随着存有消息副本的中间节点的增多,这些中间节点在网络中移动时,遇到消息目的节点的机会也会变大,因此消息投递成功率都呈现上升趋势。通过图 1 可以看出,本文提出的 EIBM 策略在相同实验环境下比其他 3 种缓存管理策略在消息投递率上分别提升了 10%,20% 以及 35% 以上。这是因为 EIBM 利用了消息在传递过程中经历中间节点的历史信息,考虑消息在剩余 TTL 时间内到达目的节点的可能性,并在节点缓存溢出时选择平均投递概率小的消息进行置换,从而提升了网络中消息投递的成功率。

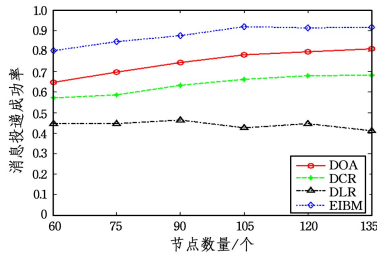


图 1 不同节点数目下的消息投递成功率

Fig. 1 Message delivery ratio with different node numbers

通过图 2 可以看出:随着节点数量的增加,不同缓存策略下的网络负载率都在增加。其中,DCR 和 DOA 策略的网络负载率基本一致;DLR 的网络负载率是最高的;而 EIBM 策略的网络负载率在所比较的 4 种策略中是最小的,在相同情况下要比 DCR 和 DOA 低 10% 左右。这主要得益于 EIBM 策略通过 ACK 机制对冗余消息副本进行了删除。网络负载率越低,对网络资源的消耗就越少,尤其是在网络中节点能量资源和缓存资源有限的情况下,降低网络负载率可以延长节点的使用时间,改善消息投递性能。

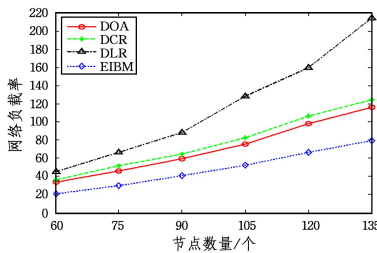


图 2 不同节点数目下的网络负载率

Fig. 2 Overhead ratio with different node numbers

通过图 3 可以看出:随着网络中节点数量的增加,EIBM, DOA, DLR 的消息平均投递延迟呈现下降趋势,但 DCR 的消息投递延迟呈现上升趋势,并且 DCR 策略的消息投递延迟是这 4 种策略中最大的。本文提出的 EIBM 策略结合了投递成功消息的 ACK 机制,释放了更多的缓存资源来投递其他消息,因此能够很好地降低网络中消息的投递延迟。

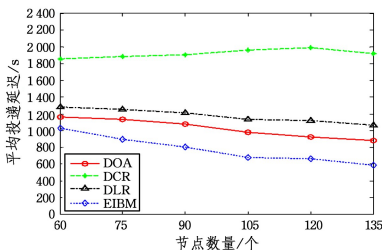


图 3 不同节点数目下的平均投递延迟

Fig. 3 Average delivery latency with different node numbers

3.3 不同缓存空间下的性能评估

本节在不同缓存空间下对 4 种缓存管理策略的消息投递延迟、网络负载率和消息投递成功率进行仿真,并比较其性能。本节的仿真中统一将网络中的节点个数设为 120。

通过图 4 可以看出:在网络中节点缓存空间增加的情况下,4 种缓存管理策略下的消息投递成功率有所改善;但在改

善幅度方面,EIBM 要明显好于其他 3 种缓存管理策略。

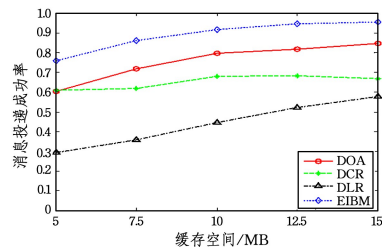


图 4 不同缓存空间下的消息投递成功率

Fig. 4 Delivery ratio with different buffer sizes

通过图 5 可以看出:在网络负载率方面,随着节点缓存空间的增加,EIBM 虽然有轻微的增加趋势,但是总体水平还是低于其他 3 种缓存管理策略。在节点缓存空间为 5 MB 时,EIBM 的网络负载率指标相对于其他 3 种缓存策略取得了最好的效果。这是因为缓存空间较小时,节点更容易拥塞,选择丢弃合适的消息可以有效地降低网络的负载率。

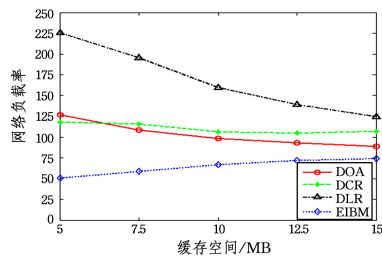


图 5 不同缓存空间下的网络负载率

Fig. 5 Overhead ratio with different buffer sizes

通过图 6 中可以看出:随着节点缓存空间的增加,DOA 和 DLR 策略的消息平均投递延迟有增加的趋势,EIBM 和 DCR 策略的消息平均投递延迟呈现下降趋势;同时,相比于 DCR,EIBM 的投递延迟能够降低 70% 以上。这说明增加节点的缓存空间对 EIBM 策略在降低消息平均投递延迟方面的效果比较明显。

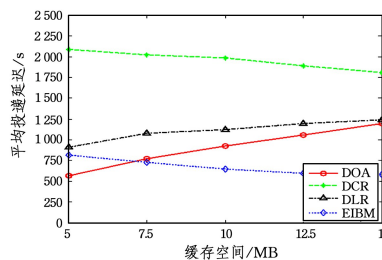


图 6 不同缓存空间下的平均投递延迟

Fig. 6 Average delivery latency with different buffer sizes

从上述实验可以看出,本文提出的 EIBM 缓存管理策略综合考虑了节点之间的相遇间隔、消息的剩余 TTL 值以及携带消息的每个节点在消息剩余 TTL 时间内和消息目的节点之间的相遇概率,并根据消息副本在网络中的分布情况,来计算每个消息的平均投递概率。在节点缓存溢出时,选择投递概率低的消息进行丢弃,进一步提高了节点缓存资源的利用率。DOA 和 DLR 策略没有考虑消息的剩余 TTL 时间;而 DCR 虽然考虑了消息副本数与 TTL 的比值,但是没有考虑携带消息的中间节点和消息目的节点之间的相遇概率;EIBM

综合考虑了上述策略的优点。因此,相比于其他已有的3种策略,所提出的EIBM策略能够提高网络中消息的投递成功率,同时降低网络的负载率和消息的平均投递延迟。

结束语 为了合理利用机会网络中节点上有限的缓存资源,提高网络中消息的投递性能,本文提出了一种基于节点相遇间隔的缓存管理策略。利用RWP移动模型中节点相遇间隔服从指数分布的特性,通过在消息尾部增加消息在网络传输过程中所经历中间节点和目的节点之间相遇间隔的期望值,估算消息在网络中的平均投递概率。在节点缓存溢出时,根据节点上计算的每个消息的平均投递率来对缓存中的消息进行调度,优先丢弃对消息投递成功率影响较小的消息,来达到提高网络中消息投递成功率的目的。通过实验仿真可以看出,与已有的DOA,DCR,DLR相比,所提出的EIBM策略合理使用了资源受限机会网络中有限的节点缓存资源,从而能够降低网络负载率和消息投递延迟,并且提高消息的投递成功率。

参考文献

- [1] CAO Y, SUN Z. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges[J]. IEEE Communications Surveys & Tutorials, 2013, 15(2): 654-677.
- [2] RA M R, LIU B, LA PORTA T F, et al. Medusa: A programming framework for crowd-sensing applications[C]// Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services. ACM, 2012: 337-350.
- [3] JAIN S, CHAWLA M. Survey of buffer management policies for delay tolerant networks[J]. The Journal of Engineering, 2014 (3): 117-123.
- [4] DAVIS J A, FAGG A H, LEVINE B N. Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks[C]// Proceedings of the 5th International Symposium on Wearable Computers. IEEE, 2001: 141-148.
- [5] LIU Q L, PAN Y J, LI Y, et al. Congestion Control Strategy Based on Copy Rate in DTN[J]. Journal of Beijing University of Posts and Telecommunications, 2010, 33(4): 88-92. (in Chinese)
- [6] AN Y, HUANG J W, LUO X, et al. A Node Betweenness Based Congestion Aware Routing Algorithm for DTNs[J]. Journal of Chinese Computer Systems, 2014, 35(9): 2062-2067. (in Chinese)
- [7] LI X L, XUAN M Y. Research on Incentive Strategy of Nodes in Selfish Opportunistic Network[J]. Computer Science, 2017, 44(4): 213-217, 222. (in Chinese)
- [8] MA X B, LI A L, ZHANG X J, et al. Buffer Management Based on Stationary Relay Nodes and Message Correlation in Sparse Opportunistic Networks[J]. Computer Science, 2016, 43(S2): 296-300. (in Chinese)
- [9] LU Y, WANG W, CHEN L, et al. Opportunistic forwarding in energy harvesting mobile delay tolerant networks[C]// IEEE International Conference on Communications (ICC). IEEE, 2014: 526-531.
- [10] ZHANG F, WANG X M, ZHANG L C, et al. A Cellular-learning-automata-based Congestion Control Strategy in Opportunistic Networks[J]. Journal of Sichuan University (Engineering Science Edition), 2016, 48(1): 158-165. (in Chinese)
- [11] 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). 2009: 55.