

基于消息重要性的机会网络能量均衡路由算法

陈志刚^{1,2}, 殷滨安^{1,2}, 吴嘉^{1,2}

(1. 中南大学软件学院, 湖南 长沙 410075; 2. “移动医疗”教育部-中国移动联合实验室, 湖南 长沙 410083)

摘 要: 提出了基于消息重要性的能量均衡路由算法——MIEBR(message importance based energy balanced routing algorithm)。MIEBR 对消息重要性进行度量, 并根据消息转发收益确定消息的转发顺序和路由。在节点缓存空间不足时, MIEBR 依据消息缓存价值进行缓存替换。实验结果表明, 在缓存空间和能量受限的机会网络中, MIEBR 均衡节点的能量消耗, 降低了重要消息的传输延时, 在投递成功率和延时等方面优于其他算法。

关键词: 机会网络; 路由算法; 消息重要性; 能量均衡

中图分类号: TP393.4

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2018276

Energy balanced routing algorithm for opportunistic network based on message importance

CHEN Zhigang^{1,2}, YIN Bin'an^{1,2}, WU Jia^{1,2}

1. School of Software, Central South University, Changsha 410075, China

2. "Mobile Health" Ministry of Education-China Mobile Joint Laboratory, Changsha 410083, China

Abstract: A new routing algorithm MIEBR(message importance based energy balanced routing algorithm) was proposed. MIEBR measures the importance of messages. It determines message forwarding order and routing by message forwarding profit. When cache space is insufficient, MIEBR performs cache replacement based on message cache value. Simulation results show that, in cache space and energy constrained networks, MIEBR balances energy consumption of nodes, and reduces the latency of important messages. It outperforms other algorithms in delivery rate and latency.

Key words: opportunistic network, routing algorithm, message importance, energy balance

1 引言

机会网络^[1]具有间歇式连通网络^[2]和延时容忍网络^[3]的特征, 它把节点的移动当作传输的机会, 采用“存储—携带—转发”的模式进行通信。由于机会网络不需要节点间直接存在完整的链路, 近年来, 已经成为无线通信领域内的研究热点^[4-7]。目前已经出现了基于机会网络的具体应用。例如, 在手持设备交换网 PSN(pocket switched network)^[8]中, 智能终端节点通过人们的移动提供局部通信的机

会; MIT 开发的 CarTel^[9]系统中, 车辆可以通过移动与其他车辆获得局部通信的机会, 或与道路旁安装的无线智能传输设备进行通信, 从而将车辆传感器收集的路况和车辆状态等信息传输至 Internet 上的服务器。

机会网络中的节点往往是手持或车载设备^[10-11], 因此, 节点间的数据传输是不稳定的, 节点可能在数据传输完成前离开通信范围, 导致一部分消息未能发送给对方。因此, 在节点间进行数据传输的过程中, 应该充分利用节点相遇的机会, 优先传输重

收稿日期: 2017-12-07; 修回日期: 2018-05-03

基金项目: 国家自然科学基金资助项目 (No.71633006, No.61672540, No.61379057); 中国博士后科学基金资助项目 (No.2017M612586); 中南大学博士后科学基金资助项目 (No.185684)

Foundation Items: The National Natural Science Foundation of China(No.71633006, No.61672540, No.61379057), China Postdoctoral Science Foundation Funded Project(No.2017M612586), The Postdoctoral Science Foundation of Central South University(No.185684)

要性较高的消息。在机会网络的实际应用中,往往有一些消息是比较紧急、比较重要的。比如,在 CarTel 系统中,车辆故障信息要比路况信息更为重要,服务器只有收到车辆故障信息后才能采取紧急救援等措施。在手持设备交换网中,用户根据消息内容的重要程度不同,往往希望重要性较高的消息得到优先传输。

本文将节点能量和缓存空间不足等^[12-13]造成传输中断的因素考虑在内,关注机会网络的实际应用中消息重要程度不同的特点,研究在机会网络单播模式下,即所有消息只有一个目的节点条件下的消息重要性度量方法,提出了基于消息重要性的机会网络能量均衡路由算法 MIEBR(message importance based energy balanced routing algorithm),解决了重要消息得不到优先转发的问题。

2 相关工作

目前的机会网络路由算法分为零信息型和信息辅助型两大类^[14]。零信息型路由算法是指在数据转发过程中不需要额外的信息作为辅助。典型的零信息型路由算法有 Direct Delivery^[15]、Epidemic^[16]和 Spray and Wait^[17]等。

Direct Delivery 的特点是,源节点不通过中间节点转发消息,而是通过节点的移动,直到遇到目的节点时才传输消息。因此,网络中不存在消息副本,网络开销最小,但传输延时最大,很难提高消息投递成功率。Epidemic 的思想是:在节点相遇时,互相发送对方没有的信息。这种方式在不考虑节点缓存空间和能量等因素时,消息投递成功率较高,传输延时较低。但是,在实际应用中,由于网络中存在大量的消息副本,容易造成缓存不足、网络拥塞,导致网络性能急剧下降。Spray and Wait 的思想是:在 Spray 阶段采用喷洒的方式将源节点的信息传输给所有邻居节点。若在 Spray 阶段没有将消息传输到目的节点,则进入 Wait 阶段,携带消息的节点通过移动直到遇到目的节点后将消息转发给目的节点。该方法解决了 Epidemic 消息副本数过多的问题,但由于没有利用节点和网络状态等信息,传输成功率往往不高。

由于零信息型算法没有考虑到实际应用中节点、数据及网络拓扑的特征,使该类算法往往在投递成功率、网络开销、传输延时等 3 个方面达

不到理想平衡。而信息辅助型路由算法则是结合节点的接触历史、剩余能量、移动速度、邻居变化和社会关系等信息提出的算法。典型的信息辅助型算法有 PROPHET^[18]、PROPICMAN^[19]和 Bubble Rap^[20]等。

PROPHET 利用节点间的历史接触时长和接触次数信息来预测节点间的相遇概率,使消息向与目的节点相遇概率高的中间节点传递。当两个节点相遇时,它们的相遇概率增加,否则相遇概率随时间衰退。该算法同时考虑了相遇概率的传递性。相比 Epidemic,该算法大大降低了路由开销,但由于未对消息副本数做出限制,随着消息的传播,网络开销依然较大。该算法未考虑到节点的社会属性,单纯依靠历史相遇概率来选择下一跳节点,存在一定的不合理性。

PROPICMAN 是一个基于节点社会上下文的路由算法。其思想是:节点在转发数据时,将目的节点社会上下文信息发送给两跳以内的邻居节点,邻居节点比较自身和目的节点的社会上下文信息,并将结果返回给该节点,该节点选择两跳范围内社会上下文信息符合程度最高的邻居节点作为未来两跳的路由中转。该算法利用了社会上下文信息,在低路由开销下具有较高的消息投递成功率。但是,该算法未考虑到节点的能量对节点进行消息转发的影响,可能出现因部分节点能量消耗过快而耗尽的现象。

Bubble Rap 算法通过对节点进行聚类,并计算节点的全局和局部中心度。在转发过程中,若消息没有进入目的节点所在的社区,则将消息发送给全局中心度高的节点。若消息进入了目的节点所在的社区,则将消息发送给该社区局部中心度高的节点。该算法太过于依赖中心度高的节点,造成中心度高的节点繁忙,中心度低的节点空闲,从而使网络负载不均衡。

文献[21]提出了 TBR 算法,该算法结合数据分组的 TTL 值、跳数和数据分组大小确定消息的优先级,优先向邻居节点转发优先级高的消息。该算法减小了具有较高优先级消息的传输延时。但是,该算法在确定消息优先级时,未考虑到消息内容的重要程度。

3 模型设计

本文使用到的数学符号说明如表 1 所示。

表 1 符号说明

符号表示	符号意义
w_i	节点第 i 项社会属性值
δ_i	节点第 i 项社会属性重要程度值
θ_m	消息 m 的内容重要程度值
TTL_m	消息 m 的剩余生存时间
h_m	消息 m 的跳数
s_m	消息 m 的大小
∂_m	消息 m 的重要性度量值
ρ_m	消息 m 的缓存价值
E_S	节点转发一个数据分组所消耗的能量
E_R	节点接收一个数据分组所消耗的能量
E_{ACK}	节点返回 ACK 所消耗的能量
B_m	消息 m 的字节数
B_{pkt}	一个数据分组的字节数
$E_{r,a}$	节点 a 转发或接收 m 后的剩余能量
$E_{r(old),a}$	节点 a 转发或接收 m 前的剩余能量
E_{max}	节点的最大能量值
η_a	节点 a 的剩余能量百分比
$G_{m,b}$	消息 m 转发至节点 b 的收益值
M	节点缓存中待转发消息的集合
F	节点缓存中待转发消息的目的地址集合
V	节点的邻居集合
G_m	消息 m 的转发收益值
V^*	使 G_m 达到最大值的邻居节点集合
C	待接收消息节点缓存中的消息集合
A	C 与待接收消息的并集
S_L	节点剩余缓存空间
S_A	A 中所有消息大小的总和
S	节点缓存空间大小
D	从 A 中删除的元素集合
φ	节点死亡收敛速率
σ	节点剩余能量标准差

3.1 节点的社会属性

网络中的每个用户节点在缓存中存储一张自身社会属性信息表，表中包含该节点用户的姓名、住址、专业、工作单位和爱好等信息。本文将这些信息称为社会上下文信息。每条信息包括属性名(attriName)和属性值(value)。为方便节点间进行属性值的比较，同时保护用户的社会上下文隐私，每个属性值映射为长度固定的散列值(hashvalue)，并

且保证网络中的每一个用户节点具有相同的属性名和相同的排序。节点的社会属性信息如表 2 所示。

表 2 社会属性信息

属性名	属性值	散列值
$Attr_i$	w_i	$H(w_i)$

节点的第 i 项社会属性散列值 $H(w_i)$ 是由属性值 w_i 映射得到的长度为 32 bit 的十六进制数，是数据唯一且极其紧凑的数值表示形式。本文采用 MD5 对 w_i 进行映射。

例如，属性名为爱好，属性值为乒乓球，则 $H(\text{乒乓球})$ 的值为 32 bit 的十六进制数且是唯一的 571c4d3c50312404d3c82b8dc02e2805。

定义节点 a 与节点 b 的社会关系强度为

$$R(a,b) = \frac{\sum_{i=1}^l \delta_i H_a(w_i) \wedge H_b(w_i)}{\sum_{i=1}^l \delta_i} \quad (1)$$

其中，用 $\delta_i \in (0,1)$ 来区分不同属性的重要程度， δ_i 越大，表示该项属性越重要。 $H_a(w_i)$ 表示节点 a 第 i 项属性的散列值， l 为节点社会属性数， \wedge 为逻辑与运算。由于每个节点的属性个数和顺序相同，因此只需要 l 次比较就可以得到 $R(a,b)$ 。当两节点所有属性值都相同时，即对于 $i \in [1,l]$ ，有 $H_a(w_i) \wedge H_b(w_i) \equiv 1$ ，则 $R_{max}(a,b) = 1$ 。对于 $i \in [1,l]$ ，当 $H_a(w_i) \wedge H_b(w_i) \equiv 0$ 时，则 $R_{min}(a,b) = 0$ 。所以， $R(a,b)$ 的值域为 $[0,1]$ 。

3.2 消息重要性的度量

消息的重要性由两方面共同决定：1) 消息内容的重要程度，由消息发送方决定；2) 由消息的 TTL 值、跳数和大小决定。由此，定义消息 m 的重要性度量值为

$$\partial_m = \frac{\theta_m}{TTL_m h_m s_m} \quad (2)$$

其中， θ_m 表示消息发送方根据 m 的内容重要程度而设定的值， $\theta_m \in (0,1)$ ， θ_m 越大表示 m 的内容越重要。 TTL_m 为 m 的剩余生存时间。需要注意的是，不同于 TCP/IP 协议中的 TTL ，TCP/IP 协议中的 TTL 表示的是消息可转发的最大跳数，机会网络中的 TTL 表示的是消息的剩余生存时间。 TTL 值越小，表明消息离被删除的时间越近，越需要被优先转发。 h_m 为 m 的跳数， h_m 越小，表明该消息被转发的次数越少，越需要尽快转发。 s_m 为 m 的大小， s_m 越小，

则 m 中每比特的消息重要程度越大。

3.3 消息的缓存价值

机会网络中的节点缓存空间是有限的，当传输过程中接收方缓存不足时，需要删除缓存中的部分消息，从而保证节点具有足够的缓存空间接收数据。从缓存中删除的消息，应该具有最低的缓存价值。

定义消息的缓存价值为

$$\rho_m = TTL_m \theta_m \quad (3)$$

其中， TTL_m 越小，消息 m 被转发过的概率越大。反之， TTL_m 越大，表明消息 m 被转发的概率越小，则消息 m 越需要被优先保留。特别地，当 TTL_m 减为 0，说明该消息已超过其时效期限，缓存价值为 0。 θ_m 越大，即消息 m 的内容越重要，缓存该消息的价值也就越大。

3.4 能量消耗模型

机会网络中的节点往往是移动设备，这些节点的能量是有限的。因此，必须考虑节点能量对消息转发的影响。

一个数据分组在由节点 a 转发至节点 b 的过程中的能量消耗，主要包括节点 a 转发一个数据分组所消耗的能量 E_S ，以及节点 b 接收一个数据分组所消耗的能量 E_R 和返回 ACK 给节点 a 所消耗的能量 E_{ACK} 。由此，建立节点的剩余能量模型。

设消息 m 的字节数为 B_m ，一个数据分组的字节数为 B_{pkt} ，则节点 a 转发 m 后的剩余能量为

$$E_{r,a} = E_{r(oid),a} - \frac{B_m}{B_{pkt}} E_S \quad (4)$$

节点 b 接收 m 后的剩余能量为

$$E_{r,b} = E_{r(oid),b} - \frac{B_m}{B_{pkt}} (E_R + E_{ACK}) \quad (5)$$

其中， $E_{r(oid),a}$ 表示节点 a 转发 m 前的剩余能量， $E_{r(oid),b}$ 表示节点 b 接收 m 前的剩余能量。

定义节点 a 的剩余能量百分比为

$$\eta_a = \frac{E_{r,a}}{E_{max}} \quad (6)$$

其中， E_{max} 为节点的最大能量值。

本文关注的是节点在传输消息过程中的能量消耗，不考虑节点能量衰减等其他因素。

3.5 节点转发消息的收益

节点 a 在转发消息 m 至节点 b 时，若节点 b 是消息 m 的目的节点，则节点 a 转发消息 m 的收益只由消息 m 的重要性度量值 ∂_m 决定。

若节点 b 不是消息 m 的目的节点，则不能只凭消息重要性度量值来确定消息的转发顺序。显然，节点转发重要性较高的消息并成功投递到目的节点，要比重要性较低的消息的成功投递具有更高的收益。但是，若重要性较高的消息未能成功投递到目的节点，则节点的收益为 0。同时，为均衡网络中各节点的剩余能量，延长网络的生命周期，需要将消息转发给剩余能量较多的节点。

由此，定义消息 m 由节点 a 转发至节点 b 的收益为

$$G_{m,b} = \begin{cases} \partial_m [R(b,f) - R(a,f)] \eta_b, & b \neq f \\ \partial_m, & b = f \end{cases} \quad (7)$$

其中，消息 m 的目的节点为 f 。 η_b 表示节点 b 的剩余能量百分比。由基于社会上下文路由算法的有效性可知，与目的节点的社会上下文信息越相似，则节点与目的节点的相遇概率越高，消息投递成功率也就越高。因此，用 $R(b,f) - R(a,f)$ 表示消息投递成功率的增量。

图 1 给出了节点转发消息的实例。节点 a 有两条消息需要转发给节点 b ，节点 b 不是这两条消息的目的节点。设 $\eta_b = 0.8$ ，按照式(7)计算得 $G_{m_1,b} = 0.064$ ， $G_{m_2,b} = 0.192$ 。由此可见，虽然 $\partial_{m_1} > \partial_{m_2}$ ，但因 $G_{m_2,b} > G_{m_1,b}$ ，节点 a 应该优先选择 m_2 进行转发。

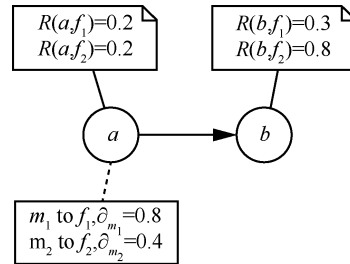


图 1 节点转发消息的实例

4 MIEBR 设计

部分节点能量的过快消耗降低了网络生命周期。在路由选择时，结合消息重要性和节点能量因

素设计路由策略，可均衡节点的能量消耗。结合消息缓存价值设计节点缓存替换策略，可高效利用缓存空间。本章首先设计能量均衡路由策略，然后设计缓存替换策略，最后给出相应的伪代码并分析算法的开销。

4.1 能量均衡路由策略

为了方便描述，记节点缓存中待转发的消息集合为 $M = \{m_1, m_2, \dots, m_u\}$ ，记 M 中目的节点地址集合为 $F = \{f_1, f_2, \dots, f_d\}$ ，记当前时刻下的邻居节点集合为 $V = \{v_1, v_2, \dots, v_k\}$ 。下面描述能量均衡路由策略的具体步骤。

步骤 1 网络初始化。当节点加入网络时，将自己的社会上下文信息和地址进行广播，并请求其他节点的社会上下文信息和地址。根据式(1)计算出节点与其他节点的社会关系强度，并存储在缓存中。

步骤 2 节点 a 将 F 发送至 $\{b | b \in V\}$ 中的每个邻居节点，并与邻居节点相互发送消息汇总矢量 SV(summary vector)。 $\{b | b \in V\}$ 中的每个邻居节点将步骤 1 中计算得到的 $\{R(b, f) | f \in F\}$ ，以及 η_b 发送给节点 a 。

步骤 3 节点 a 确定消息的转发顺序和下一跳节点。对于 $\{m_i | m_i \in M\}$ 中的每一条消息，执行以下步骤。

① 在每个邻居节点的消息汇总矢量 SV 中查找当前消息 m_i ，记没有当前消息 m_i 的邻居节点集为 V_0 ，对于 $\{b | b \in V_0\}$ 中的每个邻居节点，分别根据式(7)计算得到 $G_{m_i, b}$ ，取其中的最大值作为当前消息 m_i 的转发收益值 G_{m_i} ，即

$$G_{m_i} = \max_{b \in V_0} G_{m_i, b} \quad (8)$$

若 $G_{m_i} \leq 0$ ，则不转发该消息。

② 取使 G_{m_i} 达到最大值的邻居节点作为当前消息 m_i 的下一跳节点。因为使 G_{m_i} 达到最大值的邻居节点可能不止一个，因此记为节点集 V^* ，即

$$V^* = \arg \max_{b \in V_0} G_{m_i, b} \quad (9)$$

步骤 4 节点 a 按照消息转发收益值由大到小的顺序依次向步骤 3 中确定的下一跳节点转发消息。若出现消息转发收益值相同的情况，则依据消息在缓存中的起始地址由低位到高位顺序确定消息转发的优先级。若下一跳节点为消息的目的节点，则下一跳节点在收到消息后以广播的方式向网

络中发送接收确认 ACK 分组，网络中各节点在收到接收确认 ACK 分组后在本地缓存中查找该消息，若缓存中存在该消息，则将该消息删除。每条消息的转发过程在消息传输完毕或传输链路中断后结束。每条消息转发过程结束后，消息转发节点和接收节点分别按式(4)和式(5)更新节点的剩余能量信息。

步骤 5 节点 a 维护邻居节点集 V 。节点 a 于每个周期 T 广播 Hello 分组，Hello 分组用来发现新的邻居节点和确认邻居节点是否离开通信范围。邻居节点若收到 Hello 分组，则向节点 a 回复 Hello 的 ACK 分组。若节点 a 发现新的邻居节点，则将其加入 V 中。若节点 a 未收到邻居节点回复的 Hello 的 ACK 分组，则可认为该邻居节点不在通信范围内，将其从 V 中删除。若邻居节点集 V 有变动，则根据式(8)、式(9)更新缓存中消息转发的顺序和消息的下一跳节点。

下面结合具体的例子描述能量均衡路由策略的消息转发过程。图 2 中，节点 a 当前的邻居节点为 b 和 c ，节点 a 缓存中有两条消息 m_1 和 m_2 待转发。对于消息 m_1 ，根据式(7)计算 $G_{m_1, b} = 0.1$ ， $G_{m_1, c} = 0.1$ ，则根据式(8)知 $G_{m_1} = 0.1$ 。对于消息 m_2 ，可得 $G_{m_2, b} = 0.02$ ， $G_{m_2, c} = 0.08$ ，则 $G_{m_2} = 0.08$ 。由 $G_{m_1} > G_{m_2}$ 可知，消息转发顺序为 m_1, m_2 。由式(9)可知，因为 $G_{m_1} = G_{m_1, b} = G_{m_1, c}$ ，则将 m_1 转发至节点 b 和 c ，因为 $G_{m_2} = G_{m_2, c}$ ，则将 m_2 转发至节点 c 。

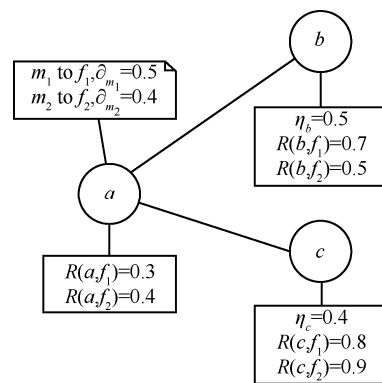


图 2 能量均衡路由策略消息转发过程实例

4.2 缓存替换策略

节点 b 接收消息 m 前，比较 m 的大小 s_m 与节点 b 的剩余缓存空间 S_L 的大小。若 $s_m \leq S_L$ ，则节点 b 接收 m ；若 $s_m > S_L$ ，则节点 b 由于剩余空间不

足不能直接接收 m ，但节点 b 可以将缓存价值较小的消息进行丢弃，满足 $s_m \leq S_L$ 的条件后再接收 m 。由此，建立 0-1 背包问题模型。

不妨设 $C = \{m_1, m_2, \dots, m_{n-1}\}$ 为缓存中的消息，记 m 为 m_n ， $A = C \cup \{m_n\}$ ，记 S_A 为 A 中消息的总大小，节点的缓存空间大小为 S ，则目标是选择哪些消息存入缓存，使消息缓存价值之和达到最大，则有

$$\max \sum_{m \in A} \rho_m p_m \quad (10)$$

$$\text{约束条件为 } \sum_{m \in A} s_m p_m \leq S \quad (11)$$

其中， $p_m \in \{0,1\}$ ，当 m 存入缓存中时， $p_m = 1$ ，否则， $p_m = 0$ 。

0-1 背包问题是一个 NP 难题，可以使用动态规划、记忆化搜索等算法求得该问题的精确解，但考虑到时间和空间复杂度，求精确解的方法是不实用的，本文采用修复法^[22]求该问题的近似解。下面描述缓存替换策略的具体步骤。

- 1) $D = \emptyset$ ， D 表示从 A 中删除的元素集合。
- 2) 查找 A 中 ρ_m / s_m 最小的一条消息，记为 m_{\min} ；若结果为空，则算法结束。
- 3) $A = A \setminus \{m_{\min}\}$ ， $D = D \cup \{m_{\min}\}$ 。
- 4) 若 $S_A > S$ ，则转到步骤 2)；否则，说明缓存装得下 A 中所有的消息，此时查看消息 m 在不在 D 中，若 $m \in D$ ，则算法结束。若 $m \notin D$ ，则执行 $C = C \setminus D$ ， $C = C \cup \{m\}$ ，即删除缓存中属于 D 的消息，然后整理缓存空间，将缓存空间中的消息紧凑排列后，把 m 存入缓存中。

下面结合具体的例子描述缓存替换策略。图 3(a) 描述了当前缓存的状态，缓存中的消息为 m_1 ， m_2 和 m_3 ，此时 m_4 待存入缓存，这 4 条消息按照 ρ_m / s_m 由大到小的排序依次为 m_1, m_2, m_4, m_3 。按照缓存替换策略， $A = \{m_1, m_2, m_3, m_4\} \setminus \{m_4\} = \{m_1, m_2, m_3\}$ ， $D = \{m_3\}$ ，此时并不真正从缓存中删除 m_3 ，如图 3(b) 所示。

此时比较 S_A 和 S 的大小，分两种情况分别讨论。第一种情况为若 $S_A > S$ ，则 $A = \{m_1, m_2, m_4\} \setminus \{m_4\} = \{m_1, m_2\}$ ， $D = \{m_3\} \cup \{m_4\} = \{m_3, m_4\}$ ，此时 $S_A < S$ ，因为 $m_4 \in D$ ，所以算法结束，缓存中不删除任何消息。第二种情况为若 $S_A \leq S$ ，因为 $m_4 \notin D$ ，则删除缓存中的 m_3 ，如图 3(c) 所示，因为剩余缓存空间

不连续，此时缓存还是存不下 m_4 ，必须对缓存消息进行整理，将消息紧凑排列后再存入 m_4 ，如图 3(d) 所示。

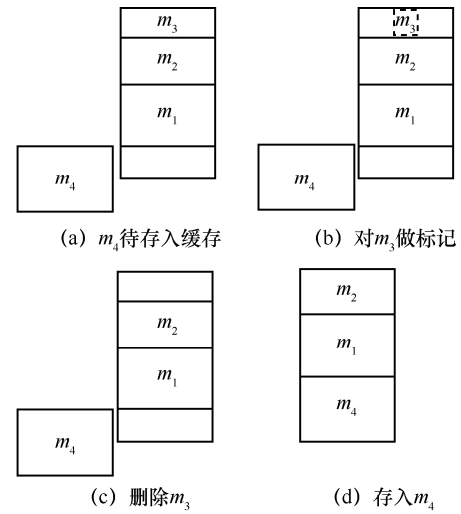


图 3 缓存替换操作实例

4.3 算法伪代码

根据以上的分析，MIEBR 的思想可用如下伪代码描述。

algorithm. MIEBR

MIEBR (M, V, F) {

 SendDest(F, V, SV);

 //向邻居 V 中的所有节点发送 M 的地址集 F 和消息汇总矢量 SV

 Receive $SV, R(b, f)$ and η_b from $V, b \in V$,

$f \in F$;

 //所有邻居节点将 $SV, R(b, f)$ 和 η_b 返回

 SortDecend(M);

 //根据消息转发收益值大小确定消息转发优先级

 for each m in M do { //按优先级转发消息

 Get V^* ; //确定转发消息的邻居节点

 for each b in V^* do {

 call MsgForward (m, b);

 UpdateEnergy();

 //更新消息转发节点和接收节点的剩余能量

 }

 }

 UpdateNeighbor(V, T)

 //每个周期 T ，维护邻居节点集 V

 }

```

MsgForward( $m, b$ ) { //缓存替换策略
     $A = C \cup \{m\}$ 
     $D = \emptyset$ ;
    while ( $S_A > S$ ) { //计算需删除的消息
        find  $m_{\min}$  in  $A$ ;
         $A = A \setminus \{m_{\min}\}$ ;
         $D = D \cup \{m_{\min}\}$ ;
    }
    if  $m \notin D$  {
    if  $D \neq \emptyset$  {
        delete  $D$  from cache and compact cache; }
    send  $m$  to  $b$ ;
    }
}

```

4.4 复杂度分析

MIEBR 虽然提高了节点转发消息的收益, 均衡了节点的能量消耗, 高效利用了节点的缓存空间, 但也带来了额外的开销, 一定程度上降低了消息的剩余生存时间。下面分析 MIEBR 的时间复杂度、空间复杂度和通信复杂度。

1) 时间复杂度

MIEBR 在初始化时, 需计算节点间的社会关系强度, 设网络中节点数为 N , 节点社会属性数为 l , 计算两节点间社会关系强度的时间复杂度为 $O(l)$, 每个节点需要计算与 $N-1$ 个节点的社会关系强度, 则时间复杂度为 $O(Nl)$ 。在计算消息的转发收益时, 需要查询 k 个邻居节点的消息汇总矢量 SV 。因为消息汇总矢量 SV 的第 x 位用 0 或 1 代表该位置对应的消息分组在缓存中不存在或存在, 所以, 查询一次消息汇总矢量 SV 的时间消耗仅为 $O(1)$ 。缓存中共有 u 条消息, 则计算节点缓存中所有消息的转发收益的时间复杂度为 $O(uk)$ 。节点按消息转发收益大小确定消息转发顺序, 排序算法采用归并排序, 时间复杂度为 $O(u \log u)$ 。缓存替换算法中的时间开销主要用于遍历 A , 时间复杂度为 $O(n)$ 。

2) 空间复杂度

节点存储自身的社会上下文信息需要的空间开销为 $O(l)$ 。网络中的每个节点计算与其他节点的社会关系强度值后, 需要存储在缓存中, 空间复杂度为 $O(N)$ 。节点存储 k 个邻居节点发送的消息汇

总矢量 SV , 设一个消息汇总矢量 SV 的长度为 q , 则空间复杂度为 $O(kq)$ 。节点存储消息的下一跳节点地址、消息重要度量值和消息转发收益值, 需要空间为 $O(u)$ 。采用归并排序确定消息的转发顺序, 辅助空间仅为 $O(1)$ 。

3) 通信复杂度

通信复杂度是指 MIEBR 中传输控制消息的开销, 不包括传输数据消息的开销。MIEBR 传输控制消息的开销主要来源于向邻居节点发送的消息目的地址集 F 以及消息汇总矢量 SV , 则通信开销为 $O(d+q)$, 以及邻居节点返回的社会关系强度值和消息汇总矢量 SV , 也为 $O(d+q)$ 。

5 仿真与分析

本文使用 ONE(opportunistic network environment)^[23]对 MIEBR 进行仿真实验, 比较 MIEBR 与典型路由算法 Epidemic、PROPHET 以及 PROPICMAN 的性能。为了更加真实的模拟现实生活中人们的移动规律, 本文选择芬兰首都赫尔辛基地图, 采用 Working Day Movement 模型^[24]进行仿真, 仿真运行 10 次后取平均值作为最终的结果。设置消息产生时的 θ 值在 (0,1) 上服从均匀分布。具体仿真参数设置如表 3 所示。

表 3 仿真参数设置

参数	取值
仿真时间	24 h
仿真区域	4 500 m × 3 400 m
消息大小	50~200 KB
TTL	5 h
消息传输速率	250 kbit/s
消息产生间隔	30 s
最大传输半径	10 m
公共汽车速度	20~50 km/h
行人速度	3~5 km/h
节点初始能量	5 000 eu
E_S	0.3 eu
E_R	0.1 eu
E_{ACK}	0.01 eu

5.1 能量均衡性能

本文用以下指标衡量算法的能量均衡性能。

1) 节点死亡收敛速率。用 t_α 表示在死亡节点数量所占比例为 α 下的网络运行时间。在一定的 α 下,

t_α 越大, 表示节点的平均生存时间越大。定义节点的死亡收敛速率为

$$\varphi = \frac{\beta - \alpha}{t_\beta - t_\alpha} \quad (12)$$

死亡节点比率从 α 变化到 β 时, φ 越大, 表示节点的死亡收敛速率越快, 能量均衡性能越好。

2) 节点剩余能量标准差。节点剩余能量标准差体现了节点之间的剩余能量差异的大小, 节点剩余能量标准差越小, 表示能量均衡性能越好。节点剩余能量标准差为

$$\sigma = \sqrt{\frac{1}{N} \sum_{i \in K} (E_{r,i} - \bar{E}_r)^2} \quad (13)$$

其中, \bar{E}_r 为网络中所有节点剩余能量的均值, K 为网络中所有节点的集合, 节点数为 N 。

设置网络中节点数量为 260 个, 缓存大小为 30 MB, 网络中死亡节点比率从 0.1 上升至 0.5 期间, 每隔 0.1 的比率记录不同算法的网络运行时间。网络运行时间从 1 h 到 7 h, 每隔 1 h 计算不同算法的节点剩余能量标准差。

由图 4 可知, 随着死亡节点比率的增大, 网络运行时间都会有所增加, MIEBR 的网络运行时间最大, 这是因为 MIEBR 控制了消息副本数量, 传输消息次数较少, 同时均衡了节点的能量消耗。MIEBR 的网络运行时间的增加最为缓慢。通过式(12)的计算, 当死亡节点比率从 0.1 变化到 0.5 时, MIEBR、PROPICMAN、PROPHET 以及 Epidemic 的 φ 值依次为 9.3×10^{-5} 、 2.8×10^{-5} 、 3.0×10^{-5} 和 3.6×10^{-5} , 表明 MIEBR 具有最快的节点死亡收敛速率, 能量均衡性能最好。

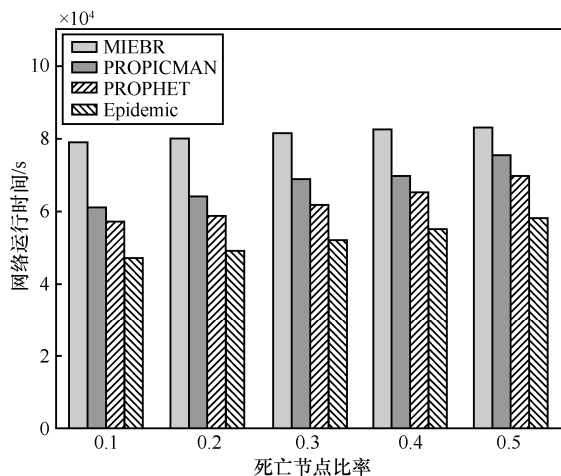


图 4 不同死亡节点比率下的网络运行时间

由图 5 可知, 随着网络运行时间的增加, 不同算法的节点剩余能量标准差都有所增加, MIEBR 的节点剩余能量标准差最小, 表明 MIEBR 的节点剩余能量分布相比之下更加均匀。这是因为 MIEBR 在路由的选择时, 结合了邻居节点的剩余能量比率, 避免了转发能力较强节点能量的过快消耗而导致节点剩余能量差异过大的现象。

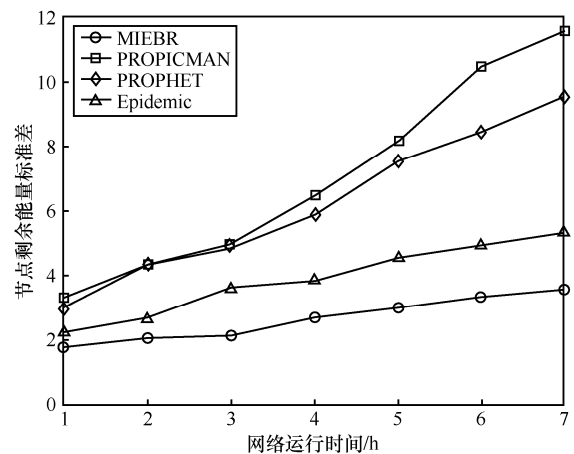


图 5 不同网络运行时间下的节点剩余能量标准差

5.2 节点数量对算法性能的影响

节点数量的增加使节点间相遇次数增多。本节设置节点缓存空间为 30 MB, 随着节点数量的增加, 比较在不同的 θ 区间下 MIEBR 的平均消息延时, 比较 MIEBR、PROPICMAN、PROPHET 和 Epidemic 的消息投递成功率、平均消息延时和网络开销比率。

图 6 显示的是 MIEBR 在不同的节点数量和不同的 θ 区间下的平均消息延时。从图 6 中可知, 不同 θ 区间的平均消息延时都随着节点数量的增加而减小, θ 值越大, 则具有越低的平均消息延时。这是因为 θ 值越大, 则消息重要性度量值就越大, 在消息转发和缓存替换中具有较高的优先级。 $\theta \in (0.75, 1)$ 的平均消息延时分别比 $\theta \in (0.50, 0.75]$ 、 $\theta \in (0.25, 0.50]$ 、 $\theta \in (0, 0.25]$ 平均降低了 4%、9% 和 14%。

图 7 显示的是不同节点数量下的消息投递成功率。从图 7 中可知, MIEBR 有着最高的消息投递成功率, 而 Epidemic 投递成功率最低, 并从节点数量为 200 以后开始下降。这是因为 Epidemic 的消息副本数最多, 导致了缓存空间不足和能量的过快消耗, 当网络中较多的节点死亡后, 投递成功率将大幅下降。而 MIEBR 不但具有节点剩余能量均衡策略, 还

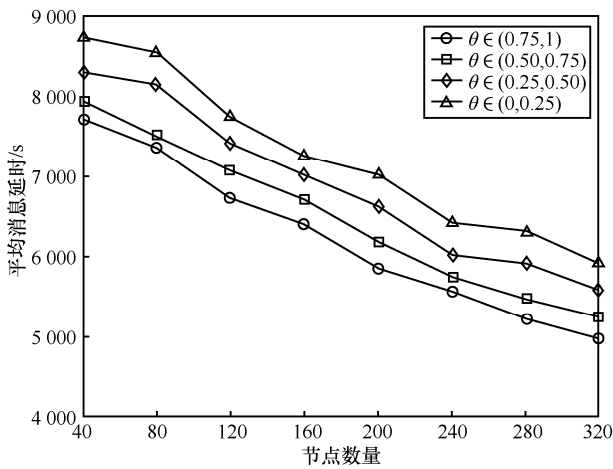


图 6 不同 θ 区间下的平均消息延时

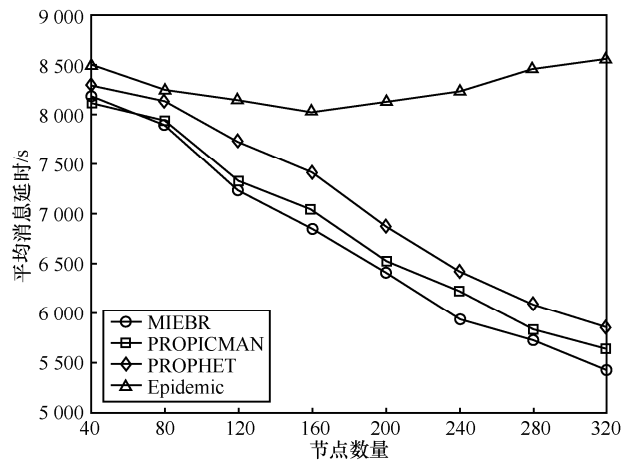


图 8 不同节点数量下的平均消息延时

具有缓存消息替换策略，很好地解决了节点剩余能量不均衡和缓存空间利用不充分的问题。MIEBR 的消息投递成功率分别比 PROPICMAN、PROPHET 和 Epidemic 平均提高了 23%、31%和 52%。

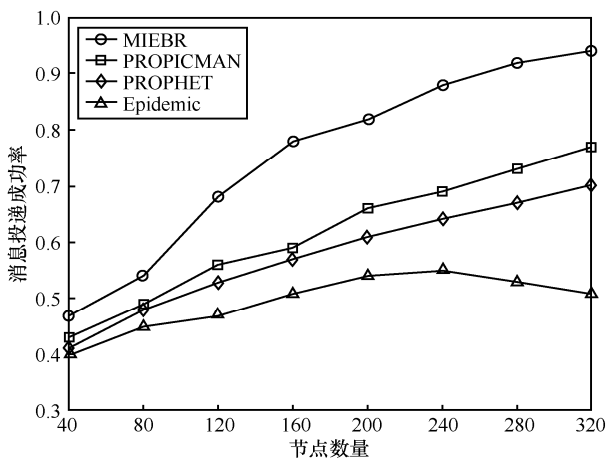


图 7 不同节点数量下的消息投递成功率

图 8 显示的是不同节点数量下的平均消息延时。从图 8 中可知，MIEBR、PROPICMAN 和 PROPHET 的平均消息延时都随着节点数量的增加而快速下降。Epidemic 的平均消息延时最高，并且先下降后增加。这是因为随着节点数量的上升，节点相遇次数的增加导致了传输次数增多，使节点能量消耗过快，节点的过快死亡导致了平均消息延时的上升。同时，Epidemic 大量的消息副本造成了缓存空间不足，影响了消息的接收和转发。MIEBR 的平均消息延时最低，且与 PROPICMAN 较为接近。MIEBR 的平均消息延时分别比 PROPICMAN、PROPHET 和 Epidemic 平均降低了 2%、8%和 19%。

图 9 显示的是不同节点数量下的网络开销比率。从图 9 中可知，网络开销比率都随着节点数量的增加而上升。Epidemic 的网络开销比率最高，而且上升趋势最明显。这是因为 Epidemic 没有控制消息副本的数量。MIEBR 和 PROPICMAN 的网络开销比率相差不大，并且都比 PROPHET 的要小。这是由于这两种算法只选择最佳的邻居节点转发数据，有效控制了网络开销。MIEBR 的网络开销比率分别比 Epidemic 和 PROPHET 平均降低了 70%和 36%。

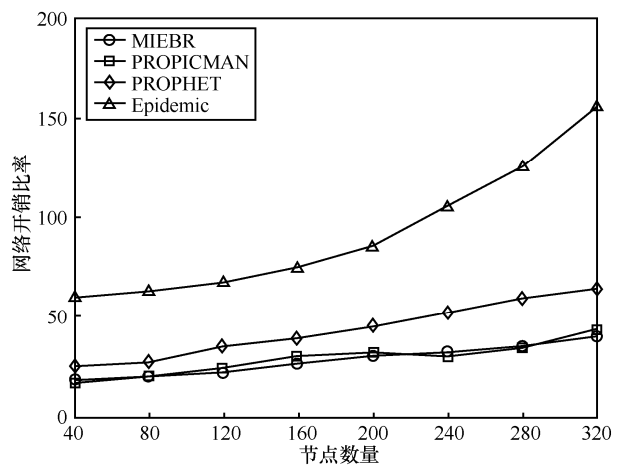


图 9 不同节点数量下的网络开销比率

5.3 缓存空间对算法性能的影响

节点缓存空间的大小是影响算法性能的一个重要的因素。缓存空间越大，则节点可存储转发的消息越多。本节设置节点数量为 280 个，研究缓存空间的变化对算法性能的影响。

图 10 描述了不同缓存空间下的消息投递成功率。从图 10 中可知，4 种算法的消息投递成功率都

随着缓存空间的增加而上升。MIEBR 具有最高的消息投递成功率。

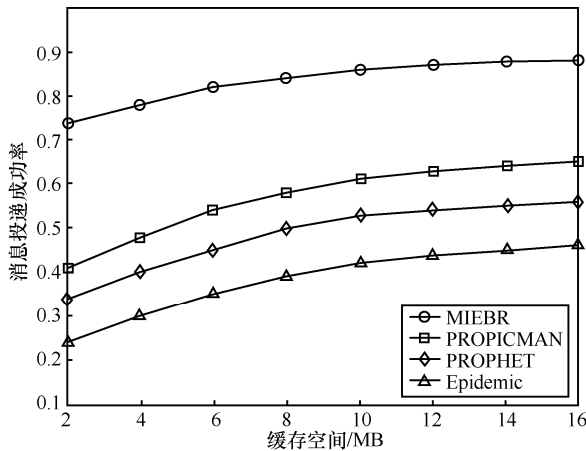


图 10 不同缓存空间下的消息投递成功率

图 11 描述了不同缓存空间下的平均消息延时。从图 11 中可看出，平均消息延时随缓存空间的增加而下降，Epidemic 具有最高的平均消息延时，且下降趋势最为明显。MIEBR 具有最低的平均消息延时，且变化较为平缓。

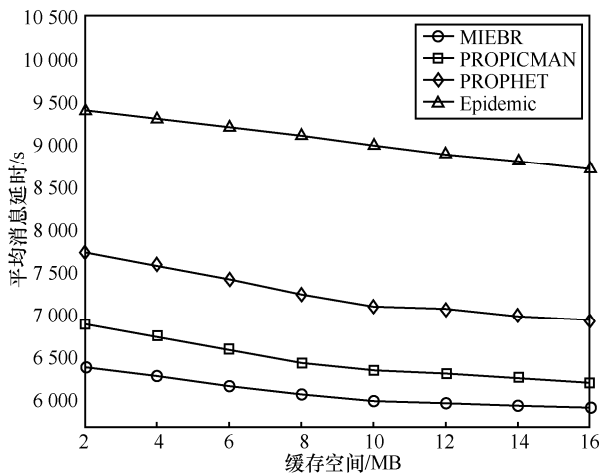


图 11 不同缓存空间下的平均消息延时

图 12 描述了不同缓存空间下的网络开销比率。随着缓存空间的增加，4 种算法的网络开销比率都不断下降。Epidemic 的网络开销比率下降趋势最为明显，MIEBR 和 PROPICMAN 的网络开销比率基本相同且下降趋势较为平缓。

由以上结果可知，节点缓存大小一定程度上影响了算法的性能。在相同的网络环境下，缓存空间越大，则节点可携带的消息越多，当遇到其他节点时，转发的消息量也就越多，因此，消息投递成功

率越高，平均消息延时越低。在数据转发的过程中，若邻居节点缓存不足，PROPICMAN、PROPHET 和 Epidemic 只能等待该邻居节点因缓存中消息的 TTL 值减为 0 被删除或该邻居节点收到目的节点接收确认的 ACK 释放出缓存空间后，继续进行数据传输。因此，未采用缓存替换策略的 PROPHET、PROPICMAN 和 Epidemic 算法性能受缓存空间因素影响较大。而 MIEBR 由于采用了缓存替换策略，使该算法在缓存空间不足时仍具有较好的性能。

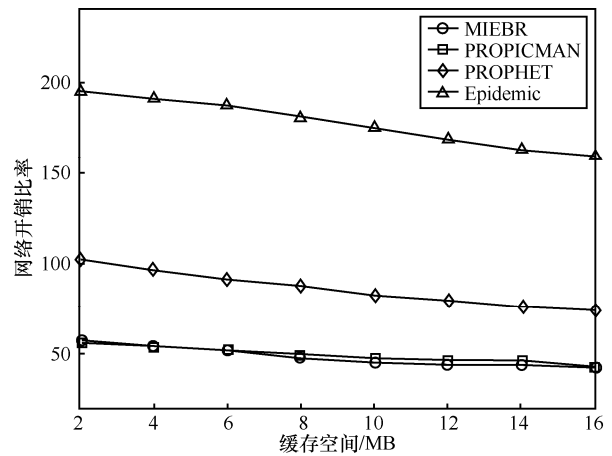


图 12 不同缓存空间下的网络开销比率

6 结束语

本文提出的 MIEBR 通过引入节点用户对消息内容重要程度的设定，提出了消息重要性的度量方法；依据消息的缓存价值，设计了缓存替换策略；根据节点转发消息的收益，设计了能量均衡路由策略。MIEBR 提高了节点转发消息的收益和缓存利用率。实验结果表明，在缓存空间和能量受限的机会网络中，MIEBR 均衡了节点的能量消耗，降低了重要消息的传输延时，提高了消息投递成功率，并具有较低的平均消息延时和网络开销。下一步的工作是设计安全机制和信用机制，保证节点用户社会上下文隐私安全和促进节点用户之间的合作。

参考文献:

- [1] SOELISTIJANTO B, HOWARTH M P. Transfer reliability and congestion control strategies in opportunistic networks: a survey[J]. IEEE Communications Surveys & Tutorials, 2014, 16(1):538-555.
- [2] WU D, ZHANG H, WANG H, et al. Quality-of-protection-driven data forwarding for intermittently connected wireless networks[J]. Wireless

- Communications IEEE, 2015, 22(4):66-73.
- [3] NAVAZ A S S. An efficient intrusion detection scheme for mitigating nodes using data aggregation in delay tolerant network[J]. International Journal of Scientific & Engineering Research, 2015, 6(9):421-428.
- [4] 姚建盛, 马春光, 袁琪. 基于效用的机会网络“物—物交换”激励机制[J]. 通信学报, 2016, 37(09):102-110.
YAO J S, MA C G, YUAN Q. Utility-based barter trade incentive scheme in opportunistic network[J]. Journal on Communications, 2016, 37(09):102-110.
- [5] PAN D R, ZHANG H, CHEN W J, et al. Transmission of multimedia contents in opportunistic networks with social selfish nodes[J]. Multimedia Systems, 2015, 21(3):277-288.
- [6] WU J, CHEN Z G. Reducing energy consumption priority selection of node transmission routing algorithm in opportunistic network[J]. Advances in Information Sciences & Service Sciences, 2014, 6(1):111-119.
- [7] YIN B A, CHEN Z G, WU J. Transmission failure tolerance and node punishment mechanism in opportunistic network based on repeated-game[C]//Communications in Computer and Information Science. 2017, 101-115.
- [8] BARUA R, SHADMAN S, CHAKRABARTY A. PNGP: a social relationship based routing algorithm for pocket switched network[C]//IEEE 19th International Conference on Computer and Information Technology. 2017:25-30.
- [9] HULL B W. Opportunistic sensing and mobile data delivery in the cartel system[M]. Massachusetts Institute of Technology. 2010.
- [10] 王丹, 陈志刚, 吴嘉. 基于机会网络中社交关系的转发机制[J]. 计算机应用研究, 2015, 32(5):1461-1463.
WANG D, CHEN Z G, WU J. Forwarding mechanism in opportunistic networks based on social relations[J]. Application Research of Computers, 2015, 32(5):1461-1463.
- [11] 李豪, 陈志刚, 吴嘉. 基于社区与社会性的机会网络路由算法[J]. 计算机工程, 2015, 41(12):64-68+74.
LI H, CHEN Z G, WU J. Opportunistic network routing algorithm based on community and sociality[J]. Computer Engineering, 2015, 41(12): 64-68, 74.
- [12] CHEN Z G, WU J. Applying a sensor energy supply communication scheme to big data opportunistic networks[J]. KSII Transactions on Internet and Information Systems, 2016, 10(5):2029-2046.
- [13] 叶晖, 陈志刚, 赵明. ON-CRP: 机会网络缓存替换策略研究[J]. 通信学报, 2010, 31(5):98-107.
YE H, CHEN Z G, ZHAO M. ON-CRP: cache replacement policy for opportunistic networks[J]. Journal on Communications, 2010, 31(5):98-107.
- [14] 马华东, 袁培燕, 赵东. 移动机会网络路由问题研究进展[J]. 软件学报, 2015, 26(3):600-616
MA H D, YUAN P Y, ZHAO D. Research progress on routing problem in mobile opportunistic networks[J]. Journal of Software, 2015, 26(3):600-616.
- [15] 孙践知, 刘乃瑞, 张迎新, 等. 机会网络典型路由算法性能分析[J]. 计算机工程, 2011, 37(16):86-89.
SUN J Z, LIU N R, ZHANG Y X, et al. Performance analysis of typical routing algorithm in opportunistic network[J]. Computer Engineering, 2011, 37(16):86-89.
- [16] REN Z, PENG S, CHEN H, et al. Epidemic routing based on adaptive compression of vectors: efficient low-delay routing for opportunistic networks based on adaptive compression of vectors[J]. International Journal of Communication Systems, 2015, 28(3):560-573.
- [17] JAIN S, CHAWLA M, SOARES V N G J, et al. Enhanced fuzzy logic-based spray and wait routing protocol for delay tolerant networks[J]. International Journal of Communication Systems, 2016, 29(12):1820-1843.
- [18] SOK P, KIM K. Distance-based PROPHET routing protocol in disruption tolerant network[C]//International Conference on ICT Convergence. 2013:159-164.
- [19] NGUYEN H A, GIORDANO S, PUIATTI A. Probabilistic routing protocol for intermittently connected mobile ad hoc network (PROPICMAN)[C]//IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks. 2007:1-6.
- [20] HUI P, CROWCROFT J, YONEKI E. Bubble rap: social-based forwarding in delay tolerant networks[J]. IEEE Transactions on Mobile Computing, 2011, 10(11):1576-1589.
- [21] PRODHAN A T, DAS R, KABIR H, et al. TTL based routing in opportunistic networks[J]. Journal of Network & Computer Applications, 2011, 34(5):1660-1670.
- [22] 王熙熙, 贺毅朝. 求解背包问题的演化算法[J]. 软件学报, 2017, 28(1):1-16.
WANG X Z, HE Y C. Evolutionary algorithms for knapsack problems[J]. Journal of Software, 2017, 28(1):1-16.
- [23] 王朕, 王新华, 隋敬麒. 机会网络模拟器 ONE 及其扩展研究[J]. 计算机应用研究, 2012, 29(1):272-277.
WANG Z, WANG X H, SUI J Q. Extending research for ONE simulator of opportunistic network[J]. Application Research of Computers, 2012, 29(1):272-277.
- [24] EKMAN F, KARVO J. Working day movement model[C]// Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing. 2008:33-40.

[作者简介]



陈志刚 (1964-), 男, 湖南益阳人, 博士, 中南大学教授、博士生导师, 主要研究方向为计算机网络及分布式计算。



殷滨安 (1988-), 男, 湖南长沙人, 中南大学硕士生, 主要研究方向为机会网络及网络安全。



吴嘉 (1983-), 男, 贵州贵阳人, 博士, 中南大学讲师, 主要研究方向为机会网络、软件工程及大数据分析。