

节点状态感知的机会网络分布式协作缓存机制

王汝言^{①②} 杨慧婷^{*①③} 应俊^{①③} 舒娜^{①②}

^①(重庆高校市级光通信与网络重点实验室 重庆 400065)

^②(重庆邮电大学通信与信息工程学院 重庆 400065)

^③(重庆邮电大学光电工程学院 重庆 400065)

摘要: 合理利用节点间的协作关系及相邻节点的缓存资源可有效提高机会网络中节点缓存的利用率与消息的投递率。该文提出一种节点状态感知的分布式协作缓存机制,通过感知消息扩散程度动态估计消息的重要程度,确定消息在缓存操作中的优先级。进而根据节点的活跃度及相对粘度获知给定消息与其目标节点的相遇概率,并针对消息源节点的差异设计分区协作缓存方法,同时在本地图存处于存满状态时动态选取协作节点,实现本地消息到相邻节点的转移,以达到高效利用缓存资源的目的。结果表明,所提出的协作缓存机制能够有效地利用节点有限的缓存资源,并大幅度地改善消息投递率及节点缓存利用率。

关键词: 机会网络; 扩散程度; 相遇概率; 协作节点

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2016)09-2194-08

DOI: 10.11999/JEIT151374

Node Status Aware Distributed Cooperative Caching Mechanism for Opportunistic Networks

WANG Ruyan^{①②} YANG Huiping^{①③} YING Jun^{①③} SHU Na^{①②}

^①(Optical Communication and Network Key Laboratory of Chongqing, Chongqing 400065, China)

^②(College of Telecommunication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

^③(College of Electron Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: By reasonably exploiting the collaborative relationships between nodes and the limited cache resources of neighbor nodes, the cache utilization rate and the message delivery probability of opportunistic networks can be effectively improved. A node status aware distributed cooperative cache management mechanism is proposed in the paper, where the importance degree of a given message can be dynamically estimated by perceiving its spreading degree to determine the caching priority. Furthermore, according to the active degree and the relative viscosity, the encounter probability of a given message to its destination node can be obtained for the design of adaptive cooperative cache management strategies for messages in each cache area respectively. When the cache is fully occupied, the collaboration node within the transmission range is dynamically selected, thus the message transferring to the collaboration node and the efficient utilization of caching resources can be ideally achieved. The results show that the proposed caching mechanism can fully exploit the limited cache resources and greatly improve the message delivery rate and the buffer utilization rate.

Key words: Opportunistic networks; Spreading degree; Encounter probability; Collaboration node

1 引言

随着普适计算技术的发展和移动应用环境的逐

渐成熟,机会网络^[1]得到了广泛的关注。此种网络中的节点分布比较稀疏,节点频繁移动使其连接呈现出显著的间歇性,导致源节点和目标节点之间的端到端路径频繁断裂。为了解决路径断裂所引发的通信中断问题,人们提出了使用“存储-携带-转发”的异步路由方式来传送消息^[2]。然而,由于机会网络中节点移动频繁,使得消息端到端的延时较大,导致消息在网络中滞留较长时间,极大地消耗了中继节点有限的存储空间。与传统的移动自组网相比,机会网络对缓存资源有更高的要求。当网络中有大量消息传输时,由于缓存资源受限,将导致节点的缓存空间被迅速消耗,进而造成网络拥塞。

收稿日期: 2015-12-08; 改回日期: 2016-04-18; 网络出版: 2016-06-16

*通信作者: 杨慧婷 yanghuiping78@163.com

基金项目: 国家自然科学基金(61371097, 61271261), 重庆市自然科学基金重点基金(CSTC2013JJB40001, CSTC2013JJB40006), 重庆市教委项目(KJ1400402), 重庆市青年科技人才培养计划(CSTC2014KJRC-QNRC40001)

Foundation Items: The National Natural Science Foundation of China (61371097, 61271261), Chongqing Natural Science Foundation (CSTC2013JJB40001, CSTC2013JJB40006), Chongqing Education Commission Project (KJ1400402), Youth Talents Training Project of Chongqing Science & Technology Commission (CSTC2014KJRC-QNRC40001)

为解决上述问题,研究人员提出了带有节点协作的缓存管理机制。文献[3]采用面向群组的缓存消息共享机制,各个节点以分布式的方式根据关系紧密程度建立相应的群组,同一群组内的节点能够协作地分享彼此有限的缓存空间,从而为其他节点提供消息缓存服务。然而,文中在设计协作缓存机制时,将“幽灵”消息无差别地广播至通信范围内的所有节点,极大地占用了其他节点有限的缓存空间。文献[4]根据节点转发消息的概率和历史相遇信息来确定网络中心节点,同时将网络中的消息存储至中心节点及其邻近节点,充分利用节点有限的缓存空间,从而实现多个缓存节点之间消息的有效传输。然而,文中为了提高数据的访问率,降低数据的访问时延,将数据都存放于网络的中心地区,极大地增加了网络中心地区节点的缓存负载。与文献[4]类似,文献[5]针对社会性机会网络中的数据访问问题,提出了一种分层协作缓存机制,根据节点的社会特性将缓存空间划分为3个部分,分别存储本地节点、朋友节点及陌生节点产生的消息,并且针对每个部分设计不同的缓存替换策略以提高节点的缓存利用率和消息投递率。然而,当缓存将满时,文中提出将本地访问频率最低的数据进行删除,用数据访问频率最高的数据进行替换,而直接删除访问频率最低的数据会影响其他用户对该数据的访问,增大兴趣用户访问该数据的时延,造成该数据被访问率的进一步降低。现有的研究表明,以人类为载体构成的机会网络呈现出“大世界、小世界”特性^[6],具有相同或相近的社会属性的节点之间,部分活跃节点与其他节点之间连接较为频繁。节点之间社会属性的不同造成其移动模式与活跃程度有明显的差异。因此,在节点之间采用协作的方式进行消息的缓存时,应充分考虑节点之间活跃程度的差异,以提高消息的缓存效率。

为了解决上述问题,在利用消息扩散程度动态估计的基础上,本文提出了一种节点状态感知的分布式协作缓存机制(Node Status aware Distributed Cooperative Caching strategy, NS-DCC)。未对节点的移动模式作同质假设,而是基于社会网络理论,通过感知节点的活跃度及相对粘度,评估节点间的相遇概率,并根据消息的源节点将消息分类缓存,以协作的方式将节点缓存内的消息替换为与目标节点具有较高相遇概率的消息。这样所有的节点都参与消息的缓存过程,而非将所有消息都集中于网络中心地区的节点缓存内,以此提高节点的缓存利用率。为了避免盲目地删除缓存内的消息,所提出的策略采用分布式协作缓存转移方法,通过动态建立

给定节点的协作节点集合,将节点缓存内的消息转移至通信范围内的协作节点,而非上述文献中所提出的无差别地广播或直接删除机制,从而提高缓存资源的利用率及消息的投递率。

2 扩散程度感知的消息重要程度估计方法

根据机会网络的消息转发原理可知,消息的扩散程度与其被成功投递的概率直接相关,扩散程度越高的消息到达目标节点的概率越大,导致继续携带此消息的必要性下降,消息的重要程度也就随之下降。可见,消息的扩散程度直接决定了消息的投递状态。因此,本文根据扩散程度估计给定消息的重要程度,从而避免节点在有限的链路持续时间内转发冗余程度较高的消息,以达到更加合理地利用网络资源的目的。

研究表明,机会网络具有较强的社会属性,具有相似社会属性的节点以自组织的形式聚合为一个社区^[7],因此,机会网络中的节点从逻辑上分布在多个相对独立的社区中。根据节点的社会属性可知,不同社区内节点建立连接的密集程度并不相同,所以在估计消息的重要程度时应当综合考虑该消息在各个社区中的扩散程度。

对于分布式运行的机会网络来说,难以通过洪泛的方式获知消息的全部转发过程,因此,本文通过节点在运动过程中所获知的局部历史状态信息近似地预测消息的扩散程度。节点 i 在本地以三元组 $\mathbf{TM}=\langle id_j, mt_{ij}, dt_{ij} \rangle$ 维护其相遇状态表项信息,其中 id_j 表示相遇节点 j 的标识,由社区号 G_j 与编号 b_j 组成; mt_{ij} 为相遇时间; dt_{ij} 为相遇持续时间。节点 i 与 j 相遇后交互双方节点的表项信息 $\mathbf{MT}_i \Leftrightarrow \mathbf{MT}_j$,进而在较短的收敛时间内可建立相遇状态矩阵 $\mathbf{SM}=(\mathbf{MT}_1 \ \mathbf{MT}_2 \ \cdots \ \mathbf{MT}_N)$ 。定义 G 表示网络中的社区集合, G_τ 表示第 τ 个社区,则节点通过在本地图查相遇状态矩阵 $\mathbf{SM}$,搜索节点本地存储的 $\mathbf{SM}$ 中有相同社区号 G_τ 的相遇表项信息,并除去节点编号重复的相遇表项信息,通过查询经过处理后的相遇表项数量即可获知社区 G_τ 的节点数 n_τ 。

$$n_\tau = \text{Num}\{k, k \in \mathbf{SM} \& G_k = G_\tau\} \quad (1)$$

可见,节点在本地生成消息的存储状态集合为 $\mathbf{BS}=\{\mathbf{PBM}_1 \ \mathbf{BM}_2 \ \cdots \ \mathbf{BM}_m \ \cdots \ \mathbf{BM}_n\}$,同时,当节点相遇后进行消息交互时将更新 $\mathbf{BS}$ 。其中 $\mathbf{BM}_m$ 表示消息 m 存储状态信息的布尔矩阵,定义如式(2)所示。

$$\mathbf{BM}_m = \{\mathbf{Cs}_1 \ \mathbf{Cs}_2 \ \cdots \ \mathbf{Cs}_n\} \quad (2)$$

假设节点在本地为每个社区都创建一个消息存储状态表项 $\mathbf{BM}_m$,相遇节点通过交互表项 $\mathbf{BM}_m$,可估计双方节点所在社区存储该消息的状态。显然,

任意社区中可能包含多个节点, 所以每个社区节点存储消息 m 的状态被记录在其布尔矩阵中, 其存储形式如式(3)所示, 其中 $\mathbf{Cs}_i$ 为第 i 个社区内节点关于消息 m 的存储状态向量, $\text{bl}(G_i^j)$ 为社区 G_i 内的任意节点 j 存储消息 m 的布尔状态, G_i^j 为社区 G_i 内的任意节点 j 。

$$\mathbf{Cs}_i = \{\text{bl}(G_i^1) \text{ bl}(G_i^2) \cdots \text{bl}(G_i^n)\}^T \quad (3)$$

$$\text{bl}(G_i^j) = \begin{cases} 1, & \text{节点 } G_i^j \text{ 存储消息 } m \\ 0, & \text{其他} \end{cases} \quad (4)$$

由于网络中消息由节点携带转发, 则可利用网络中携带过该消息的节点数来衡量给定消息在网络中的扩散程度。令 $m_d(t)$ 表示 t 时刻社区 G_i 中存储消息 m 的节点数, K_i 为社区 G_i 内节点的数量, 由上可知, $m_i(t)$ 可通过查询消息的存储状态集合 $\mathbf{BS}$ 得到, K_i 可通过查询本地相遇状态矩阵 $\mathbf{SM}$ 获取, 那么节点可预测消息 m 在社区 G_i 内的扩散程度 $S_i(t)$, 即

$$S_i(t) = m_i(t) / K_i \quad (5)$$

如前所述, 当消息 m 的扩散程度较高时, 继续在该社区内扩散该消息并不能够提高其投递率, 反而占用了其他消息的缓存资源, 从而严重影响消息的缓存利用率。因此, 定义消息在任意社区的扩散程度为

$$S_i(t) = \begin{cases} m_i(t) / K_i, & S_i(t) \leq \gamma \\ 1, & S_i(t) > \gamma \end{cases} \quad (6)$$

其中 γ 为消息扩散度阈值。由于社区内节点联系较为紧密, 社区间节点联系较为稀疏。因此, 目标节点与目标社区内的节点联系较为紧密, 在相同扩散程度下, 目标社区内的消息扩散程度对消息重要程度的影响要高于其他社区, 从而综合考虑消息在目标社区与其他社区的扩散程度, 可得消息的总体扩散程度 $S(t)$ 如式(7), G_d 为消息目标节点所属社区, $S_d(t)$ 为消息在目标节点 G_d 内的扩散程度。

$$S(t) = \prod_{i \in G - G_d} S_i(t) e^{S_d(t)} \quad (7)$$

最终, 根据消息的扩散程度可估计消息 m 的重要程度 $W(t)$ 。

$$W(t) = 1 - \sigma_t S(t) \quad (8)$$

$\sigma_t = t_0 / t_m$ 定义为消息的时间影响因子。消息的初始生存时间为 t_m , 当前的已存活时间为 t_0 。存活时间越长的消息, 其已成功投递率越高, 重要程度就越低。消息的重要程度决定了消息的缓存替换优先级, 重要程度较大的消息在缓存替换时的优先级越高。

3 相遇概率估计的协作缓存替换方法

机会网络具有间断连接特性, 消息需要存储在节点缓存内以等待节点相遇带来的通信机会, 进而执行转发操作。可见, 消息的投递状态与节点间的相遇概率相关。为了提高消息的投递概率, 需估计节点间的相遇概率, 从而在节点建立连接后选择合适的消息进行交互。

3.1 节点活跃度感知的相遇概率估计

如前所述, 消息的传输状态与消息缓存节点和目标节点间的相遇概率直接相关^[8]。消息携带节点的活跃程度越高, 则该节点与其他节点相遇的机会越多, 使得消息的扩散速度越快, 目标节点与消息携带节点相遇的概率就越大, 消息的成功投递率也就越高。由此易知, 消息成功投递概率与节点的活跃度直接相关, 可通过准确地获知节点的活跃度来估计节点间相遇概率, 从而有效地控制消息的转发过程, 达到提高缓存资源利用率的目的。

定义1 节点活跃度: 单位时间内节点与其他节点进入通信范围, 并建立通信连接的次数占网络中全部连接次数的比例。其反映了该节点相对网络中其他节点的活跃程度。

定义2 节点相对粘度: 任意节点针对某个特定节点的活跃度。其反映了该节点相对网络中某给定节点的活跃程度。

根据“大世界、小世界”特性可知, 节点间的关系存在较大差异, 节点与关系较亲密的节点频繁相遇, 而与其他节点相遇频率则较低。因此, 在估计节点活跃度时, 若单纯地考虑节点的活跃度无法准确地评估其携带消息并投递至目标节点的能力, 还需要考虑当前节点对消息目标节点的相对粘度。此外, 消息携带节点通过与目标节点直接相遇的方式也能够实现消息的投递, 因此在估计节点相遇概率时也需要考虑节点与目标节点的直接相遇概率。

综上, 本文所提出的相遇概率估计方法根据当前网络状态, 预测消息缓存节点的活跃度及相对粘度, 进而结合节点与目标节点的直接相遇概率估计节点相遇概率。如前所述, 单位时间内给定节点与网络中其他节点的相遇次数越多, 则通过给定节点所携带的消息成功投递率越高。因此, 给定节点与网络中其他节点的相遇次数是评估节点活跃度的关键参数。定义 Φ 为网络中的节点集合, 那么网络中节点数量为 $N = \text{Num}\{\Phi\}$ 。将给定节点 i 与网络中其他节点的相遇次数 $n_i(t)$ 定义为该节点的相遇总次数, 那么网络中节点相遇次数为各个节点进入其他节点通信范围的相遇总次数, 定义为 $N(t) = \sum_{i=1}^N n_i(t)$ 。进而可得出节点 i 的活跃度 $L_i(t)$ 。

$$L_i(t) = n_i(t) / N(t) \quad (9)$$

同理, 可获知当前节点相对消息目标节点的活跃程度, 即节点相对粘度。

$$D_{id}(t) = n_{id}(t)/n_d(t) \quad (10)$$

其中 $n_{id}(t) = \text{Num}\{\mathbf{MT}_{id}\}$, $\mathbf{MT}_{id} \subseteq \mathbf{MT}_i$ 为节点 i 与目标节点 d 的相遇次数, $n_d(t) = \text{Num}\{\mathbf{MT}_d\}$ 为目标节点 d 与其他节点的相遇总次数。

研究表明, 节点相遇间隔时间服从指数分布^[9], 则节点 i 在 t 时间内与节点 d 相遇的概率为

$$F_{id}(t) = 1 - e^{-\lambda_{id}t} \quad (11)$$

其中 $\lambda_{id} = 1/\overline{\Delta mt_{id}}$ 为节点 i 与 d 的相遇强度; $\overline{\Delta mt_{id}}$ 为节点 i 与 d 的平均相遇间隔时间, 计算如式(12)所示; mt_{id}^k 为第 k 次相遇的时间, 且 $mt_{id}^0 = 0$ 。

$$\overline{\Delta mt_{id}} = E(\Delta mt_{id}) = \sum_{k=0}^{n_{id}(t)} (mt_{id}^{k+1} - mt_{id}^k) / n_{id}(t) \quad (12)$$

假设消息的剩余生存时间为 $t_r = t_{in} - t_0$, 由式(11)和式(12)可得出任意节点 i 与目标节点在 t_r 内相遇的概率 $F_{id}(t_r)$ 为

$$F_{id}(t_r) = 1 - e^{-\lambda_{id}t_r} = 1 - e^{-n_{id}(t)(t_{in} - t_0) / \sum_{k=0}^{n_{id}(t)} (mt_{id}^{k+1} - mt_{id}^k)} \quad (13)$$

综合考虑节点的活跃度 $L_i(t)$ 及相对粘度 $D_{id}(t)$, 即可得到任意节点 i 的相遇概率 $P_{id}(t)$, 如式(14)所示。节点相遇后将交互自身与消息的目标节点相遇概率, 从而在本地存储与消息目标节点相遇概率较高的消息。

$$P_{id}(t) = L_i(t) D_{id}(t) F_{id}(t_r) \quad (14)$$

3.2 消息源感知的分区协作缓存方法

理论研究和实际测量表明, 机会网络中的节点具有极强的社会属性, 节点不仅需要缓存自身产生的消息, 还需为其他节点提供缓存与转发服务^[10]。然而, 为保证自身产生的消息被顺利投递, 节点通常表现出一定的自私性, 即为自身产生的消息赋予较高的缓存优先级。由于节点的缓存空间有限, 当网络中存在大量消息时, 节点自身产生的消息较容易得到存储、携带及转发, 而为其他节点产生的消息分配较少的缓存空间, 此种方式将造成严重的缓存竞争问题。因此, 在设计缓存管理策略时应考虑消息源的因素, 针对自身产生的消息与其他节点的消息分别进行缓存资源的配置。为不失一般性, 本文依据节点缓存内消息的来源, 将节点缓存划分为本地缓存区 Q_L 和协作缓存区 Q_C , 分别存储节点本地产生的消息和其他节点产生的消息。其结构如图1所示。

为了能够提高消息在存储及转发过程中的鲁棒性, 并有效地降低负载, 针对来自不同分区

提出消息源感知的分区协作缓存替换方法, 对于不同缓存区的消息采取不同的缓存替换方法。该方法的基本过程如下: 初始状态下, 对本地缓存区消息设置较高的缓存替换与转发优先级, 而对协作缓存区消息设置较低的优先级。节点相遇后在进行缓存替换或消息转发时, 首先根据消息的重要程度优先向对方节点复制或转移本地缓存区内的消息, 如图2中节点 i, j 优先复制本地缓存区消息集合 Z_L^i 与 Z_L^j , 此时若有剩余缓存空间, 则继续提取协作缓存区消息 Z_C^i 与 Z_C^j ; 然后, 节点交互双方到达消息目标节点的概率, 分别为 $P_{id}(t)$ 和 $P_{jd}(t)$, 并提取彼此自身有较高相遇概率的消息, 即从 Z_L^i, Z_L^j, Z_C^i 与 Z_C^j 中分别提取出 $Z_i(P_{id}(t) < P_{jd}(t))$ 与 $Z_j(P_{jd}(t) > P_{id}(t))$; 最后, 节点将本地缓存区内的消息复制至对方节点的协作缓存区中, 若对方仍有缓存空间, 则将节点协作缓存区内的消息直接转移至对方协作缓存区。上述的协作缓存替换过程如图2所示。

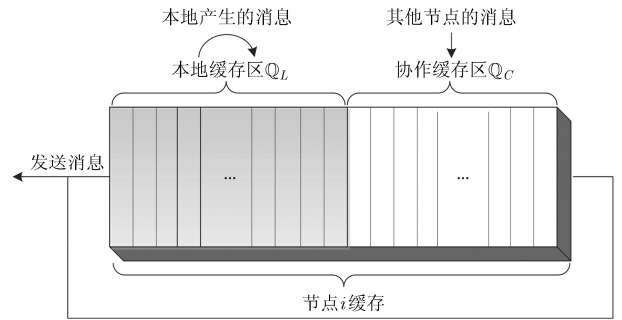


图1 节点缓存分区

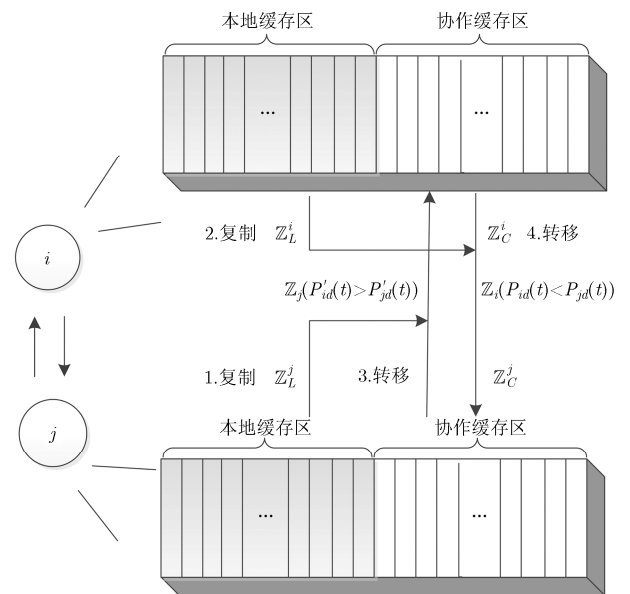


图2 协作缓存替换过程示意图

4 分布式协作缓存转移方法

由于节点的活跃度存在差异,当网络负载较重时,将导致部分活跃节点的缓存出现溢出。传统的缓存管理策略通过直接删除缓存内的消息来释放缓存空间,对消息投递率造成极大的影响。为避免节点直接删除消息造成投递率下降,本文提出协作缓存转移方法。当节点缓存已满时,通过将消息转移至具有剩余缓存空间,且具有较强协作能力的节点,从而以协作的方式缓存消息,以充分利用节点的缓存资源,达到提高消息投递率的目的。

4.1 Top-k 协作节点集选取方法

显然,若协作节点选取不合理会进一步占用其他节点有限的缓存资源,影响其他节点的消息投递过程。因此,协作缓存节点集合的选取是协作缓存转移方法的关键。

如前所述,相遇概率较高的节点再次相遇的机率较大,便于节点之间的缓存转移与回传。由于机会网络中的连接频繁断裂,为了保证消息可被成功转移,还应考虑节点间的相遇持续时间。因此,选取协作节点集时应综合考虑节点间的相遇概率与相遇持续时间。

令 δ_{ij} 表示节点 i 与节点 j 的平均相遇持续时间,计算方法如式(15)所示,并且对其进行归一化处理,如式(16)所示,其中 $n'_i(t)$ 为 t 时刻与节点 i 相遇的不同节点数量。

$$\delta_{ij} = \sum_{k=1}^{n_{ij}(t)} dt_{ji}^k / n_{ij}(t) \quad (15)$$

$$\hat{\delta}_{ij} = \delta_{ij} / \sum_{j \in \mathbf{MT}_i} \frac{n'_i(t)}{n_{ij}(t)} \delta_{ij} \quad (16)$$

定义 $\hat{\sigma}_{ji}^2$ 为归一化相遇持续时间偏移因子,该参数反映了节点 j 与节点 i 相遇的持续时间偏移状态。

$$\hat{\sigma}_{ji}^2 = \sum_{k=1}^{n_{ij}(t)} (dt_{ji}^k - \delta_{ji})^2 / \sum_{j=1}^{n'_i(t)} \sum_{k=1}^{n_{ij}(t)} (dt_{ji}^k - \delta_{ji})^2 \quad (17)$$

因而,可进一步定义节点 j 与节点 i 之间的归一化协作系数 $\hat{C}_{ji}$,其综合考虑了节点间的相遇概率与相遇持续时间状态,反映了节点 j 为节点 i 提供协作缓存服务能力的归一化数值。

$$\hat{C}_{ji} = P_{ij}(t) \hat{\delta}_{ij} \hat{\sigma}_{ji}^2 \quad (18)$$

按照上述方式,节点 i 在每次相遇事件发生后,根据获得的网络状态信息更新各个节点的 $\hat{C}_{ji}$,并选取前 k 个协作系数最大的节点加入其协作缓存节点集 Θ_i 中。若节点缓存已满,可将消息转移至通信范围内的协作节点集 Θ_i 中的节点。

由于同一社区内节点的相遇概率较高,则将消息转移至此类节点可有效提高消息的投递率。若在

节点的通信范围内存在属于消息的目标社区的节点,则也将该节点并入节点 i 的协作缓存节点集 Θ_i 中,因此定义节点 i 的协作缓存节点集 Θ_i 为

$$\Theta_i = \text{Max}_k \{ \hat{C}_{ji}, j \in \mathbf{MT}_i \} \cup G_d \quad (19)$$

当节点缓存已满时,若节点通信范围内存在多个协作缓存节点集 Θ_i 中的节点,节点 i 则按照消息的重要程度将消息广播至协作集 Θ_i 中有剩余缓存空间的节点,并将广播后的消息从本地删除。反之,若节点通信范围内不存在协作节点集 Θ_i 中的节点,节点 i 则按照消息的重要程度优先删除本地缓存区的消息。当接收节点 i 与消息转移节点再次相遇时,若节点 i 此时有剩余缓存空间,则比较双方节点与消息目标节点的相遇概率,若该节点的相遇概率较低,则将消息回传至节点 i 。反之,则不把消息回传给节点 i 。

4.2 NS-DCC 机制

为提高节点的缓存利用率,在利用消息扩散程度动态估计的基础上,本文提出了一种节点状态感知的分布式协作缓存机制(NS-DCC)。当两节点相遇,首先判断相遇节点是否为消息的目标节点,若不是消息的目标节点,则进一步判断相遇节点是否有剩余缓存空间。当相遇节点有剩余缓存空间时,则采用协作缓存替换方法,使得节点在本地存储自身与消息目标节点有较高相遇概率的消息。当节点缓存已满时,则采用协作缓存转移方法,将节点本地消息转移至协作节点,避免消息丢失所造成的消息投递率下降。因此,本文所设计的分布式协作缓存机制的执行流程如下:

步骤 1 若相遇节点为消息的目标节点,直接将消息发送给对方节点;

步骤 2 若相遇节点不是消息的目标节点且都有空余缓存空间,则按照前述的缓存替换方法进行缓存内消息的替换。依据前述估计所得的双方节点与消息目标节点间的相遇概率,并结合消息的重要程度,优先将双方节点本地缓存区内的重要程度较高的消息复制到对方的协作缓存区。若仍有缓存空间,则继续将双方节点协作缓存区内对方有较高相遇概率的消息转移至对方的协作缓存区;

步骤 3 若相遇节点不是消息的目标节点,同时该节点的缓存已满时,则采用所提出的分布式协作缓存转移方法进行节点本地缓存内消息的转移。节点首先动态选取协作节点集,以确定邻居节点中协作节点,进而依据消息的重要程度将较为重要的消息广播至协作节点,并将该消息从节点本地删除;

步骤 4 在分布式协作缓存转移方法中,若在节点的通信范围内未找到合适的协作节点时,将根据

消息的重要程度优先删除节点本地缓存区内重要程度较低的消息；

步骤5 协作缓存消息的回传。节点与存储该节点所转移消息的协作缓存节点相遇后，若本地节点此时有剩余缓存空间，将比较此时双方节点与消息的目标节点的相遇概率。若对方较小时，则将消息重新转移至节点本地，对方节点将该消息从其缓存内删除。反之，则不作任何操作。

5 仿真验证与结果分析

本文使用机会网络仿真环境(Opportunistic Network Environment, ONE)^[11]，分别从3个方面对所提出的协作缓存机制进行性能评估，主要包括消息成功投递率、网络负载率及平均时延。其中，网络负载率定义为消息的冗余投递次数与成功投递消息数之间的比例关系，其反映了所设计的缓存管理机制的运行效率。

$$OR = (N_r - N_s) / N_s \quad (20)$$

其中，OR为网络负载率， N_r 为转发消息的总次数， N_s 为成功投递到目标节点的消息数。消息的冗余投递次数为消息被转发的总次数与成功投递到目标节点的消息数的差值，即消息的无用转发次数。消息成功投递次数为网络中的全部消息成功到达其目标节点的数量。

仿真环境参数设置如表1所示。表中，网络大小表示节点运动范围；行人组中节点移动速度为2~4 km/h，汽车组中节点移动速度为20~35 km/h； γ 为消息扩散度阈值， k 为协作集节点个数默认值。

为了更加全面地验证所提出机制的有效性，本文将与机会网络中传统的缓存管理机制最老删除(Drop-Oldest)、基于消息传播状态的缓存替换机制

(MTSBR)^[12]及带有消息投递率估计的自适应缓存管理策略(ABMDPE)^[13]进行对比。在节点缓存空间不足的情况下，MTSBR机制优先删除副本数目较多的消息，从而有效地抑制网络中该类消息的过分复制转发。若消息的副本数目相等时，则按照消息在网络中的传递速率进行依次删除。ABMDPE则依据所估计消息的投递概率，直接删除已投递概率最高的消息，为已投递概率较低的消息分配缓存资源。

5.1 协作集节点数量对网络性能的影响

该节验证本文提出的NS-DCC中不同协作节点数量设置对网络性能造成的影响，仿真结果如图3~图5所示。

如图3所示，随着协作集中节点数量的增大，消息的成功投递概率呈现出先上升后下降的趋势。这是由于协作节点集的增大，使得节点在缓存将满时有较高的概率将消息转移至有较高协作能力的节点，从而避免了消息被直接删除，提高了消息的投递概率。但是，当协作节点数量过大(超过3个时)，就会导致该消息的副本同时被转移至多个协作节点，从而占用了其他消息的缓存资源，造成消息的整体投递率的下降。因此，消息投递率呈现出先上升后下降的趋势。

图4中，网络负载率随协作节点个数的增大先降低后升高。这是由于当增大协作节点集时，消息被转移至有剩余缓存空间的协作节点的概率增大，消息被删除的概率随之减小。由式(20)可知，消息被成功投递的次数增多，使得网络负载率降低。但是，当协作节点数量超过3个时，节点在进行消息转移时有较高的概率转移至多个协作节点，造成消息的冗余投递次数增多，从而使得网络负载率增大。因此，网络负载率随协作节点数量的增大先上升后下降。

如图5所示，消息的平均时延随协作节点个数的增多呈现先快速下降后逐渐上升的趋势。这是因为协作节点之间进行消息的转移，合理地利用了节点的缓存资源，降低消息被直接删除的概率，使得有更多的节点缓存该消息，则消息被快速投递至目标节点的概率变大。但是当协作节点集过大时，同一消息的副本就会占用多个节点的缓存资源，从而增大了其他消息被删除的概率，因此，造成消息整体投递时延的增大。因此，消息的平均投递时延随协作节点个数先下降后上升。

综上所述，协作节点集的设置所带来的协作开销对网络性能有一定的影响，由上述仿真结果可知，当协作节点数量设置为3时，网络的投递率、负载

表1 参数设置

参数名称	数值
仿真时间(s)	43200
节点运动模型	社区模型
传输范围(m)	10
节点数(个)	126
消息大小(kB)	200~300
γ	0.8
网络大小(m ²)	4500×3400
节点通信方式	Bluetooth
传输速度(kBps)	250
节点组数(个)	行人组(4)，汽车组(2)
消息产生时间间隔(s)	15~115
k	3

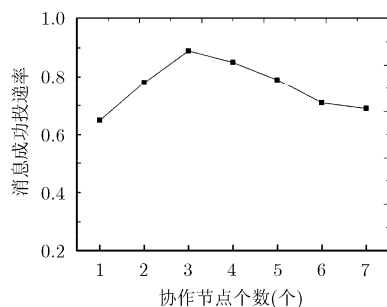


图3 不同协作节点数量下的投递率

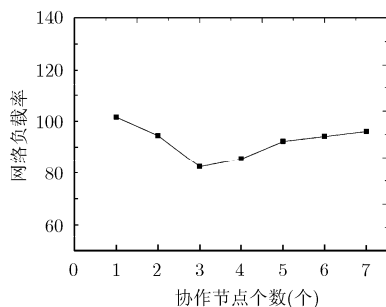


图4 不同协作节点数量下的网络负载率

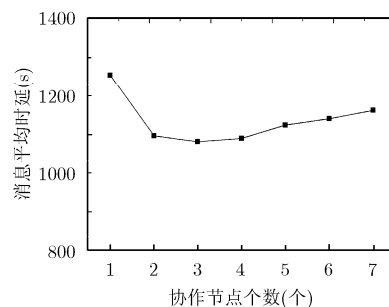


图5 不同协作节点数量下的平均时延

率与消息平均时延可取得较好的性能表现。因此，通过均衡考虑协作开销对网络的投递率、负载率及消息平均时延的影响，本文将协作节点数量设定为3个，以此降低协作缓存中的协作开销。

5.2 缓存空间对网络性能的影响

本小节针对节点缓存空间在4~22 MB的情况对NS-DCC机制与其它3种机制的性能进行比较。

从图6中可知，随着缓存空间的增大，4种机制的投递率都相应增加，这是因为节点的缓存空间越大，消息被删除的可能性就越小，能被成功投递的概率就越大。在缓存空间较小的情况下，NS-DCC机制的性能更突出。同时，在相同情况下，由于NS-DCC机制采用分区协作缓存消息以充分利用自身及其协作节点的缓存空间，因此其投递率较MTSBR机制提高了20.8%，较Drop-Oldest机制提高了24.9%，较ABMDPE提高了13.1%。

从图7中可知，随着缓存空间的增加，负载率越来越小，且NS-DCC的负载率较MTSBR与Drop-Oldest低38.4%以上，较ABMDPE则减少了17%。另外，NS-DCC的负载率变化比较平稳，原因在于该机制采用合理的协作缓存管理策略将已成功投递的消息及时地从缓存中删除，使其受缓存变化的影响较小。

如图8所示，随着缓存空间增大，4种机制的消息平均时延逐渐减少，且NS-DCC机制的平均时延

明显小于另外3种机制。这是因为NS-DCC采用协作的方式将消息存储至其协作节点，使得有较多的空闲缓存资源存储大量消息，从而提高了消息与其目标节点相遇的机会，进而有效降低了消息平均时延。

由上可知，当缓存空间较小时，所提NS-DCC机制较其他3种算法的性能表现更为突出。这是由于NS-DCC对消息的扩散程度进行了有效感知，使得节点可初步估计本地缓存内消息的重要程度。进而，根据所提相遇概率估计方法，估计节点与消息的目标节点的相遇概率。相遇的节点间通过比较双方与消息目标节点的相遇概率，将节点缓存内的消息替换为双方有较高相遇概率的消息。当缓存空间较为有限时，NS-DCC可充分利用节点有限的缓存空间，尽可能地缓存节点有较高投递概率的消息。并且，缓存空间较小易引起缓存溢出，NS-DCC通过将消息转移至协作能力较强的节点，从而避免了消息被直接删除，大大提高了消息被成功投递的概率，增加了成功投递消息的个数，并减少了消息的平均投递时延。因此，当缓存空间较小时，所提NS-DCC机制可取得相对较好的网络性能。

6 结束语

为提高节点的缓存利用率与消息的投递率，本文提出了一种节点状态感知的分布式协作缓存机

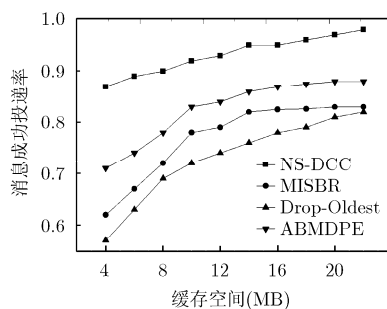


图6 不同缓存空间下的投递率

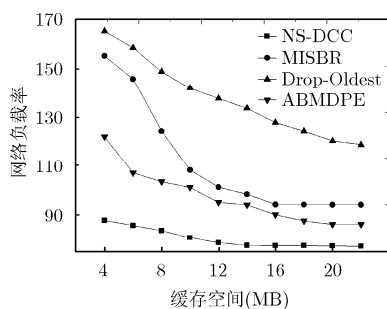


图7 不同缓存空间下的网络负载率

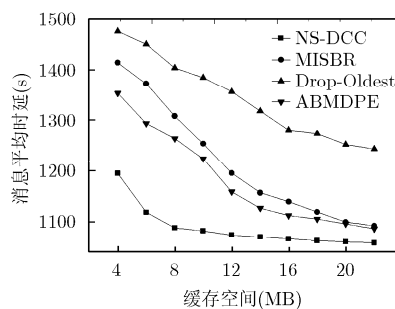


图8 不同缓存空间下的平均时延

制。该机制通过分析消息的扩散程度动态预测消息的重要程度。进而, 通过动态感知节点的缓存状态, 自主决策适用于当前缓存状态的协作缓存机制, 以在本地缓存适当的消息, 或在缓存即将溢出时转移本地消息至合适的协作节点, 避免盲目删除消息。仿真结果表明, 所提出的 NS-DCC 协作缓存机制可有效改善节点的缓存, 并大幅度提升网络性能。

参 考 文 献

- [1] WU Dapeng, ZHANG Hongpei, WANG Honggang, *et al.* Quality-of-protection-driven data forwarding for intermittently connected wireless networks[J]. *IEEE Wireless Communications*, 2015, 22(4): 66–73. doi: 10.1109/MWC.2015.7224729.
- [2] ZHANG Zufan, YI Yinxue, YANG Jing, *et al.* Energy efficiency based on joint data frame fragmentation and cooperative transmission[J]. *China Communications*, 2014, 11(10): 96–105. doi: 10.1109/CC.2014.6969798.
- [3] CHEN Honglong and LOU Wei. GAR: Group aware cooperative routing protocol for resource-constraint opportunistic networks[J]. *Computer Communications*, 2014, 48: 20–29. doi: 10.1016/j.comcom.2014.03.022.
- [4] GAO Wei, CAO Guohong, IYENGAR A, *et al.* Cooperative caching for efficient data access in disruption tolerant networks[J]. *IEEE Transactions on Mobile Computing*, 2014, 13(3): 611–625. doi: 10.1109/TMC.2013.33.
- [5] WANG Yunsheng, WU Jie, and XIAO Mingjun. Hierarchical cooperative caching in mobile opportunistic social networks[C]. Global Telecommunications Conference (GLOBECOM 2014), Austin, USA, 2014: 411–416. doi: 10.1109/GLOCOM.2014.7036843.
- [6] PAN Hui, CROWCROFT J, and YONEKI E. BUBBLE rap: social-based forwarding in delay-tolerant networks[J]. *IEEE Transactions on Mobile Computing*, 2011, 10(11): 1576–1589. doi: 10.1109/TMC.2010.246.
- [7] 吴大鹏, 靳继伟, 吕翊, 等. 节点相似度感知的社会化间断连接无线网络结构检测机制[J]. 电子与信息学报, 2013, 35(1): 141–146. doi: 10.3724/SP.J.1146.2011.01178.
WU Dapeng, JIN Jiwei, LÜ Yi, *et al.* Similarity aware community detecting method for social intermittently connected mobile network[J]. *Journal of Electronics & Information Technology*, 2013, 35(1): 141–146. doi: 10.3724/SP.J.1146.2011.01178.
- [8] 申健, 夏靖波, 付凯, 等. 基于历史相遇概率的容迟容断网络主动拥塞控制算法[J]. 计算机应用, 2014, 34(3): 644–648. doi: 10.11772/j.issn.1001-9081.2014.03.0644.
SHEN Jian, XIA Jingbo, FU Kai, *et al.* Active congestion control strategy based on historical probability in delay tolerant networks[J]. *Journal of Computer Applications*, 2014, 34(3): 644–648. doi: 10.11772/j.issn.1001-9081.2014.03.0644.
- [9] LI Ze and SHEN Haiying. SEDUM: exploiting social networks in utility-based distributed routing for DTNs[J]. *IEEE Transactions on Computers*, 2013, 62(1): 83–97. doi: 10.1109/TC.2011.232.
- [10] LIU Lei, CUI Zehan, XING Mingjie, *et al.* A software memory partition approach for eliminating bank-level interference in multicore systems[C]. Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques, New York, USA, 2012: 367–376. doi: 10.1145/2370816.2370869.
- [11] LILIE L T, BEN OTHMANE L, ANGIN P, *et al.* A simulation study of ad hoc networking of UAVs with opportunistic resource utilization networks[J]. *Journal of Network and Computer Applications*, 2014, 38: 3–15. doi: 10.1016/j.jnca.2013.05.003.
- [12] LIU Yao, WANG Jianxin, ZHANG Shigeng, *et al.* A buffer management scheme based on message transmission status in delay tolerant networks[C]. Global Telecommunications Conference (GLOBECOM 2011), Houston, USA, 2011: 1–5. doi: 10.1109/GLOCOM.2011.6134084.
- [13] 吴大鹏, 张普宁, 王汝言. 带有消息投递概率估计的机会网络自适应缓存管理策略[J]. 电子与信息学报, 2014, 36(2): 390–395. doi: 10.3724/SP.J.1146.2012.01677.
WU Dapeng, ZHANG Puning, and WANG Ruyan. Adaptive buffer management strategy with message delivery probability estimating method in opportunistic networks[J]. *Journal of Electronics & Information Technology*, 2014, 36(2): 390–395. doi: 10.3724/SP.J.1146.2012.01677.

王汝言: 男, 1969年生, 教授, 博士, 研究方向为泛在网络、全光网络理论与技术、多媒体信息处理等。

杨慧婷: 女, 1989年生, 硕士生, 研究方向为机会网络缓存管理及路由技术。

应俊: 女, 1976年生, 副教授, 硕士, 研究方向为电子信息技术。

舒娜: 女, 1980年生, 副教授, 博士, 研究方向为网络性能分析。