

## DTN 中基于消息质量度和节点可信度的拥塞控制



崔建群 黄东升 常亚楠 吴淑庆

华中师范大学计算机学院 武汉 430079

(jqcui@126.com)

**摘要** DTN(Delay Tolerant Network)具有间歇性连接、资源有限以及拓扑结构随机动态变化等特点,因此会受到网络资源有限和网络拓扑不确定性的限制,极易产生网络拥塞。针对这一问题,提出了一种基于消息质量度和节点可信度的拥塞控制策略 CCMQ(Congestion Control Based on Message Quality and Node Reliability in DTN)。该策略主要根据消息的质量度划分消息的优先级,在转发消息时,将优先级高的消息优先转发;在选择下一跳节点时,选择节点可信度高的节点进行消息的转发,并充分考虑中继节点自身的属性;在发生拥塞时,消息质量度小的消息被率先丢弃,同时增加了 S-ACK 消息确认删除机制,以释放节点的缓存空间,从而有效缓解节点拥塞。仿真结果表明,相比传统的拥塞控制算法,CCMQ 在消息递交率、网络负载率和平均时延性能方面都有较大的提升。

**关键词** DTN;消息质量度;节点可信度;删除机制;拥塞控制策略

**中图分类号** TP393

## Congestion Control Based on Message Quality and Node Reliability in DTN

CUI Jian-qun, HUANG Dong-sheng, CHANG Ya-nan and WU Shu-qing

School of Computer, Central China Normal University, Wuhan 430079, China

**Abstract** Delay Tolerant Network (DTN) has the characteristics of intermittent connection, limited resources and random dynamic change of topology structure. Therefore, limited network resources and uncertainty of network topology can easily lead to network congestion. In order to solve this problem, this paper puts forward a kind of congestion control strategy (CCMQ) based on message quality degree and node reliability. In this strategy, messages are prioritized according to the quality of the message. When forwarding the message, the message with a higher priority is forwarded firstly. When the next hop node is selected, the node with high reliability is selected for message forwarding, and the attributes of the relay node are fully considered. When congestion occurs, messages with low message quality are discarded firstly, and the S-ACK message confirmation and deletion mechanism are added to release the cache space of the node, so as to effectively alleviate the node congestion. Simulation results show that compared with the traditional congestion control algorithm, CCMQ has better performance in message delivery rate, network load rate and average delay.

**Keywords** DTN, Message quality, Node reliability, Deletion mechanism, Congestion control strategy

## 1 引言

延迟容忍网络(Delay Tolerant Networks, DTNs)泛指部署在极端环境下、由于节点的移动或者节点资源受限等导致节点之间只能间歇性连接、具有高时延的一类网络<sup>[1]</sup>。DTN网络区别于传统 Internet 等网络之处在于,DTN 网络具有间歇性连接、传输时延高、资源受限以及拓扑结构动态变化等特点<sup>[2]</sup>。因此,DTN 网络未来将会被广泛应用于车载网络服务<sup>[3]</sup>、社会学习网络<sup>[4]</sup>以及普适计算<sup>[5]</sup>隐私和安全等方面具体应用中。目前,延迟容忍网络已经成为无线网络领域最为热门的研究领域之一。DTN 路由算法根据是否利用节点的

社会属性以及辅助路由算法决策和设计,将延迟容忍网络路由算法分为社会感知路由算法和非社会感知路由算法两大类。非社会感知路由算法主要有消息渡船算法、复制策略路由算法、转发策略路由算法以及混合策略路由算法等,而社会感知路由算法主要有基于社区的社会感知路由算法和社区独立的社会感知路由算法。对于 DTN 路由算法,大量的研究从不同的角度和出发点对延迟容忍网络做了深入的研究。文献<sup>[6]</sup>提出了一种基于信任机制的 DTN 网络安全路由决策方法,通过记录节点自身携带消息能力的信任度,节点只会选择信任度比自身大的节点进行消息的转发。文献<sup>[7]</sup>针对 DTN 路由提出了一种新颖的奖励协议,该协议通过减少数据

到稿日期:2020-05-03 返修日期:2020-08-02 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金面上项目(61672257);国家自然科学基金青年项目(61702210)

This work was supported by the National Natural Science Foundation of China(61672257) and National Natural Science Foundation Youth Project of China(61702210).

通信作者:常亚楠(yncchang@mail.ccnu.edu.cn)

包的丢失和传输能量的消耗来提高路由的安全性,从而提高整个网络的性能。文献[8]提出了一种基于节点间协作进行消息传输的策略,并研究了以企业最终效用最大化为目标的最优激励政策。文献[9]提出了一种对消息重要程度进行度量的方法,并根据转发收益确定消息的转发顺序和路由。文献[10]对未来深空互联网的网络协议架构进行了深入的分析,并提出了未来深空互联 DTN 中有待进一步研究的几个问题。很少有路由算法针对节点自身的缓存提出优化策略来提高算法的性能<sup>[11]</sup>,因此 DTN 中的拥塞控制算法是目前各种路由协议面临的挑战和需要研究解决的关键问题。

## 2 相关工作

DTN 中的拥塞控制算法的研究已经受到了高度的关注。由于不同的路由算法对应不同的转发策略和路由规则,因此在设计拥塞控制管理机制时也会有所差异。目前,对于 DTN 中的拥塞控制研究主要分为两个方面:一方面是从节点对消息的转发做出限制和选择,从而预防节点发生拥塞;另一方面是从节点对冗余消息的删除做出优化来缓解节点的拥塞。文献[11]提出了一种基于消息权值和转发时延的拥塞控制策略,在拥塞发生时,通过对消息权值和转发时延的计算来确定丢包策略和出包策略,从而提高整个网络的交付率并降低整个网络的传输延迟。文献[12]给出了一种基于归一化混合参数的缓存管理策略(Drop Least Normalized Hybrid Parameter, DLNHP),当节点发生拥塞时,归一化混合参数小的消息将被优先丢弃,以使节点获得足够接收新消息的缓存空间,从而有效地缓解拥塞。文献[13]通过研究现有的缓冲区管理方法,提出了一种基于跳数和 TTL 的缓冲区管理方案(Drop Higher Hop-counts and Low TTL First, DHLTF),该方案针对节点的消息队列采取两种排序方法,优先删除跳数高并且 TTL 低的消息,以保证消息的可靠传输。文献[14]提出了一种自适应的、有效的缓冲区管理方案,根据新到达消息的差异和可用空间来确定需求。为了腾出不可避免的缓冲区,该算法尽量避免冗余消息丢弃,并挑选和丢弃最合适的消息,以最小化开销。这些都是节点对冗余消息的删除机制做出优化,并结合消息自身的属性和本地信息删除最合适的消息。在节点对消息的转发方面,文献[15]根据节点的缓存占用率将节点划分为 3 种状态,消息转发时尽量避免将消息转发给接近拥塞的节点,并且拒绝将消息转发给拥塞的节点。文献[16]提出了一种根据节点在网络中的运动过程动态感知网络中各个节点的拥塞状态,进而分布式地为消息选择最合适的中继节点。

综上所述,对于 DTN 拥塞控制的研究,大多数算法都是利用消息和节点自身的属性来设计转发或删除策略,但是很少考虑将节点对消息的转发以及节点对冗余消息的删除相结合,并通过对节点自身状态的分析来提前避免节点发生拥塞。因此,本文提出了一种基于消息质量度和节点可信度的拥塞控制算法 CCMQ(Congestion Control in DTN Based on Message Quality and Node Reliability),通过在消息转发时对节点进行控制,来优先转发质量度高的消息,提高消息在网络中的递交率,同时结合节点自身的属性合理地选择中继节点,提前避免节点发生拥塞。

传统的缓存管理策略主要有:基于接收时刻的缓存管理策略 DO(Drop-Oldest),丢弃节点内存中存活最久的消息;基于消息生存时间的缓存管理策略 SHLI(Drop-Shortest Life Time First),丢弃剩余生存时间最短的消息;基于消息大小的缓存管理策略 DLA(Drop-Largest First),丢弃节点内存中最大的消息。将 CCMQ 的仿真数据与 DO, SHLI, DLA, DLNHP, DHLTF 这 5 种拥塞控制策略的仿真数据进行分析 and 对比可知,其在消息投递率、传输时延、网络负载等网络性能方面有了较大的提升。

## 3 消息质量度和节点可信度定义

本文提出的基于消息质量度和节点可信度的缓存控制策略是一种混合式拥塞控制策略,其综合了消息在网络中的扩散度和消息的生存时间,定义了消息的质量度,质量度越大,消息在网络中的分布越小,剩余生存时间越长,因此在节点转发消息时,选择质量度大的消息优先转发可以保证网络中消息传递的公平性。另外,在中继节点的选择上,根据节点转发消息的成功率、节点历史相遇信息以及节点的缓存空闲率定义了节点的可信度,可信度越高的节点表示节点在网络中越可靠,节点传递消息的效率就越高。

### 3.1 消息质量度

在多副本洪泛路由算法中,随着消息在网络中的传递,消息在网络中的分布极不均匀,一些消息在网络中扩散得较快,在网络中分布得较广泛,传递到目的节点的概率就较大,但是同时也占用了网络中大量的资源,导致其他扩散慢的消息无法获得公平传递的机会。因此,为了使所有消息都能比较公平地获得网络资源,本文定义了消息质量度来合理地选择消息的转发。

**定义 1(消息的扩散度)** 消息  $m_k$  扩散度表示网络中持有该消息的所有节点数占整个网络中总节点数的比率,反映了消息在网络中的扩散程度。

网络中总的节点数是确定的(如果不确定,可以用一个预估的值代替,对结果没有影响),关键是如何得到网络中持有消息的节点数。本文根据节点的历史相遇信息得出网络中持有消息的节点数,节点中的每个消息都会维护一个该消息经过的节点表  $Q$ ,作用就是记录该消息的历史路径。当两个节点  $N_i, N_j$  相遇并建立连接时,节点  $N_i$  向节点  $N_j$  发送消息  $m_k$ ,如果节点  $N_j$  中不存在消息  $m_k$ ,则节点  $N_i$  会将节点  $N_j$  添加到消息  $m_k$  所维护的节点表  $Q_i$  中,同时节点  $N_j$  会将  $m_k$  所维护的节点表  $Q_j$  中的所有节点添加到节点表  $Q_i$  中。如果两个节点中有相同的消息  $m_k$ ,这时节点  $N_i, N_j$  就会将相同消息  $m_k$  所维护的节点表  $Q_i, Q_j$  进行更新,添加各自没有的节点到各自的节点表中,基本过程如图 1 所示。

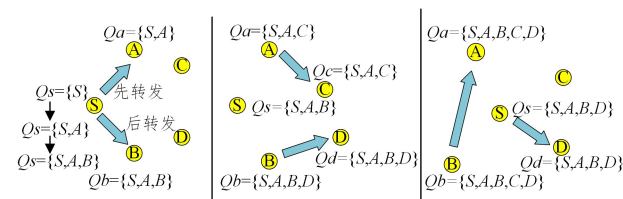


图 1 消息持有的节点表更新过程

Fig. 1 Node table update process

通过上述方式得到的在网络中持有消息的节点数更加接近该消息在网络中实际的节点持有数。消息扩散度的计算式如式(1)所示:

$$D_{m_k} = \frac{N_k}{N} \quad (1)$$

其中,  $D_{m_k}$  为消息  $m_k$  在网络中的扩散度,  $N_k$  为网络中持有消息  $m_k$  的节点数,  $N$  为网络中总的节点数。消息扩散度越大, 该消息在网络中的分布越广, 被转发的次数就越多, 在转发消息时, 为了使扩散度小的消息获得公平的转发机会, 优先转发扩散度小的消息, 提高消息在网络中的投递概率。

**定义 2(消息质量度)** 综合消息的扩散度  $D_{m_k}$  和消息的生存时间, 消息质量度的计算式如(2)所示:

$$Q_{m_k} = \frac{1}{\alpha * D_{m_k} + (1-\alpha) * \left(1 - \frac{T_{dl}}{T}\right)} \quad (2)$$

其中,  $Q_{m_k}$  为消息  $m_k$  的质量度;  $T_{dl}$  为消息  $m_k$  在网络中的剩余生存时间;  $T$  为消息  $m_k$  在网络中总的生存时间;  $S_r = \frac{T_{dl}}{T}$  表示消息的生存率, 反映了消息在网络中存活概率, 生存率高的消息到达目的节点的概率较高;  $\alpha$  为参数, 范围在  $0 \sim 1$  之间,  $\alpha$  的大小会直接影响到消息质量度的准确性。大量仿真实验证明, 当参数  $\alpha = 0.75$  时, 消息质量度  $Q_{m_k}$  对网络性能的提升最大。消息质量度综合考虑了消息在网络中的扩散度以及消息本身的生存时间。在转发消息阶段, 消息质量度大的消息表示该消息扩散度相对较小, 生存率相对较大, 消息的质量高。优先转发质量度大的消息能提高消息到达目的节点的概率。

### 3.2 节点可信度

Epidemic 路由<sup>[17]</sup>是基于洪泛策略的, 对于下一跳节点没有选择, 遇到在自己传播范围内的邻居节点就转发消息, 但是由于 DTN 网络具有资源有限、网络拓扑结构随机动态变化等特点, 盲目地转发消息将会使网络中存在大量的冗余副本, 导致节点能耗增加和缓存溢出, 进而导致网络的资源利用率低和整体运行效率低下。因此, 本文综合考虑了节点已成功转发消息的次数、到当前时刻为止遇到的节点个数和剩余的缓存空间, 基于节点的这 3 种属性给出了一种归一化参数来表示节点可信度。

$$D_a = \omega_1 * \frac{F_{transferred}}{F_{received}} + \omega_2 * \frac{N_{pass}}{N} + \omega_3 * \frac{B_{left}}{B_{all}} \quad (3)$$

其中,  $F_{transferred}$  为节点  $N_i$  到当前时刻为止成功转发的消息数,  $F_{received}$  为节点  $N_i$  到当前时刻为止接收到的消息总数,  $N_{pass}$  表示节点  $N_i$  到当前时刻为止遇到过的节点个数,  $N$  为网络中总的节点数,  $B_{left}$  表示节点  $N_i$  剩余的缓存空间,  $B_{all}$  为节点  $N_i$  初始时的缓存空间,  $C_r = \frac{B_{left}}{B_{all}}$  表示节点  $N_i$  的缓存空闲率。  $\omega_1$ ,  $\omega_2$  和  $\omega_3$  分别表示上述节点 3 种属性对节点可信度  $D_a$  的权重大小,  $\omega_1 + \omega_2 + \omega_3 = 1$ 。  $F_{transferred}$ ,  $N_{pass}$  和  $B_{left}$  越大的节点在网络中对消息的转发率就越高, 节点的活跃性更大, 更有机会遇到目的节点, 节点的性能更好。节点可信度充分考虑了节点的自身属性和历史相遇信息, 在节点选择中继节点进行消息转发时, 可信度越高的节点应该分配更多的资源, 因此将消息转发给可信度更高的节点不但能减轻网络负载, 还能大大提高消息的投递率。

权重的大小决定了节点属性在节点可信度  $D_a$  中所占的比例。为了使节点可信度  $D_a$  能更加准确地反映节点的真实性能, 本文进行了大量的仿真实验, 实验结果表明, 当  $\omega_1 = 0.2$ ,  $\omega_2 = 0.4$  和  $\omega_3 = 0.4$  时, 节点可信度  $D_a$  对网络性能的提升效果最好。

## 4 基于消息质量度和节点可信度的拥塞控制算法 CCMQ

本文提出的基于消息质量度和节点可信度的拥塞控制算法 CCMQ 综合了节点以及节点所携带消息的自身属性, 对中继节点的选择和消息的转发、删除策略进行了改进, CCMQ 的缓存管理策略主要包括中继节点选择策略、消息转发和删除策略以及 S-ACK 消息确认删除机制。

### 4.1 中继节点选择策略

在 DTN 网络中, 节点的性能不同会导致节点传输消息的能力有所差异, 尽可能将消息转发给传输能力强的节点可以提高消息投递率, 同时还考虑了中继节点自身的缓存空间, 降低了网络负载。本文依据节点的可信度和缓存空闲率来选择下一跳节点。

在节点与中继节点建立连接阶段, 节点首先比较中继节点的可信度和自身的可信度, 节点只会将消息转发给比自身可信度高的节点, 同时还考虑了中继节点的自身属性。若中继节点的缓存空闲率大于给定的阈值 *threshold*, 则直接将消息转发给中继节点, 若中继节点的缓存空闲率小于给定的阈值 *threshold*, 则节点将放弃本次转发机会, 避免中继节点发生拥塞。

### 4.2 消息转发和删除策略

在消息转发阶段, 节点基于消息质量度  $Q_{m_k}$  的排队机制将消息根据消息质量度从大到小排队, 让质量度大的消息优先发送, 增加质量度高的消息在网络中的扩散速度, 使网络中扩散率低的节点能够公平地占用网络资源。

在接收消息阶段, 当发送过来的消息较多时, 中继节点仍有可能产生拥塞, 此时就需要采取消息删除策略。首先中继节点会检测自身的剩余缓存空间能否容纳即将到来的消息, 如果可以, 则中继节点会直接将消息存储到自身的缓存中; 如果缓存空间不够, 则中继节点会删除消息队列中消息质量度  $Q_{m_k}$  最小的消息, 直到中继节点的缓存中有足够的空间存储该消息。

### 4.3 S-ACK 消息确认删除机制

当消息经过最后一跳到达目的节点以后, 应立刻停止该消息的副本在网络中的转发, 并且携带该消息的节点应从本地消息缓存列表中删除该消息。及时处理这些消息可以很好地降低网络负载, 提高节点资源的利用率。传统的 ACK 消息确认机制只考虑了消息达到目的节点以后删除网络中该消息的所有副本, 并没有考虑每个节点需要维护的 ACK 表的高效性。本文提出了基于消息生存率的 S-ACK 消息确认删除机制, 在 DTN 网络中为每个节点维护一张 S-ACK 表, 表中存储消息的 id 以及消息生存率。当消息成功到达目的节点时, 将该消息的 id 以及消息生存率存入 S-ACK 表中, 当两个节点相遇时, 将各自 S-ACK 表中消息生存率为 0 的消息删除, 然后交换各自的 S-ACK 表并删除各自节点缓存中存在于

S-ACK表中的消息,这样不仅可以避免大量的冗余消息造成网络拥塞,还可以使每个节点维护的S-ACK表更加简洁高效,从而减轻了S-ACK表在网络中交换和维护所导致的网络负载。

#### 4.4 CCMQ 拥塞控制路由策略

本文提出的CCMQ拥塞控制路由算法是在Epidemic路由算法的基础上,依据定义的消息质量度和节点可信度来选择消息的转发优先级和中继节点。当节点 $N_i$ 携带消息 $m_k$ 遇到节点 $N_j$ 时,两个节点首先交换各自维护的S-ACK表,将各自没有的消息添加进自己的S-ACK表中,根据S-ACK消息确认删除机制,删除S-ACK表中生存率为0的消息和节点 $N_i$ 和 $N_j$ 的消息队列中已经传递成功的消息,接着交换各自节点维护的消息队列 $M_i$ 和 $M_j$ 中每个消息所维护的节点表 $Q_i$ 和 $Q_j$ ,将具有相同消息id的消息所维护的节点表 $Q$ 中没有的节点添加到各自消息的节点表中,然后节点 $N_i$ 判断消息 $m_k$ 是否存在于节点 $N_j$ 的消息队列中。如果节点 $N_j$ 的消息队列 $M_j$ 中不存在消息 $m_k$ ,则根据中继节点选择策略来选择是否转发消息给节点 $N_j$ ,否则不予转发。在转发阶段选择消息转发策略来转发消息,根据消息的质量度优先转发质量度大的消息给中继节点。当节点发生拥塞时,直接采用消息删除策略来缓解节点拥塞。假设节点 $N_i$ 有 $m$ 个邻居节点,消息队列 $M_i$ 有 $n$ 个消息,则该算法的时间复杂度为 $O(m * n)$ 。该算法的具体流程如算法1所示。

#### 算法1 CCMQ 拥塞控制算法

输入:节点 $N_i$ 与 $N_j$ 相遇;Initialize  $Q_i, Q_j$ ; Initialize S-ACK $_i, S$ -ACK $_j$

输出:forwardMessageList

1. exchange S-ACK $_i$  and S-ACK $_j$  for each other
2. exchange  $Q_i$  and  $Q_j$  for each other
3. deleteAkedMessages for  $M_i, M_j$
4. delete Message in S-ACK $_i$  if Message.  $S_r = 0$
5. reverse( $M_i$ . sort(m. getQuality()))
6. FOR  $N_i$  connections with  $N_j$
7. FOR message in  $M_i$
8. IF message not in  $M_j$
9. IF  $N_j$ . getNodePriority() $>$  $N_i$ . getNodePriority()
10. IF  $N_j$ .  $C_r >$  threshold

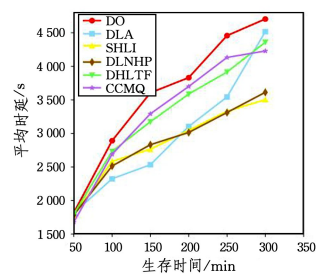
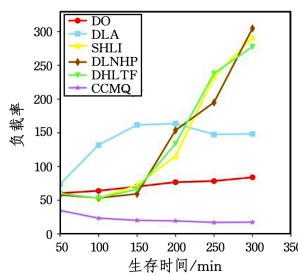
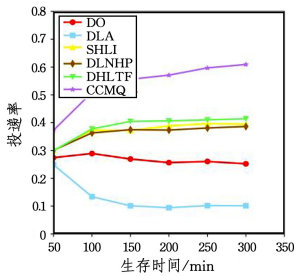


图2 各种算法在不同消息生存时间下的性能

Fig. 2 Performance of different algorithms for different message TTL

图2(a)给出了各种算法在不同的消息生存时间变化下消息投递率的变化情况。其中,CCMQ的投递率最高,DHLTF次之,DLA最低,其他5种算法的投递率都是上升的,而DLA却是下降的,因为DLA将删除缓存中最大的消息,对消息成功转发到目的节点的影响很小,更重要

11. forwardMesaageList. add(Map<message,connection>)
12. END IF
13. END IF
14. END IF
15. END FOR
16. END FOR
17. RETRUN forwardMesaageList

## 5 仿真结果与分析

### 5.1 仿真环境配置

本实验采用ONE<sup>[18]</sup>仿真平台分别在Epidemic路由算法的基础上对DLA,SHLI,DO,DLNHP,DHLTF和CCMQ拥塞控制算法进行仿真,本次仿真采用ONE自带的赫尔辛基市地图作为移动场景,大小为4500m×3400m。具体的仿真参数设置如表1所列,其他参数采用ONE自带的默认设置。

表1 仿真参数

Simulation parameters	Parameter value
simulation time/h	12
Bluetooth bandwidth/kBps	250
Message generation rate/s	25~35
TTL/min	300
Node cache/MB	5
Message size	500 kB~1 MB
Transmission range/m	10
Number of nodes	126
threshold	0.15

### 5.2 仿真结果及分析

为了验证CCMQ算法的高效性,本文进行了3组仿真实验,在不同的消息生存时间、不同的仿真时间以及不同的消息产生速率下,在消息投递率、网络负载率和平均传输时延等网络性能方面对6种算法进行了评估。实验结果表明,CCMQ对网络的性能提升最大。

#### (1) 不同消息生存时间

在前面的环境配置下,改变消息的生存时间,对比各种拥塞控制算法在不同消息生存时间下的消息投递率、网络负载率以及平均时延的变化情况,如图2所示。

的是消息自身的属性决定着消息达到目的节点的概率,随着消息的生存时间变长,消息在缓存中存在的数量也就越多,删除最大的消息可能会删除很多接近目的节点的消息。

图2(b)比较了各种路由算法的网络负载率。CCMQ的

网络负载率最低,并且随着消息生存时间增加,负载率逐渐降低,其他5种算法的负载率都是增加的,这是因为CCMQ在转发消息时将质量度高的消息转发给活跃性大的节点,这样不仅提高了消息到达目的节点的概率,同时减少了消息被转发的次数,而其他5种算法没有对节点和消息做出选择。比较图2(b)中的数据可以得出,相比其他5种算法,CCMQ的网络负载率较低。

图2(c)给出了各种算法的平均时延的变化情况。从图2(c)中可以看出,DO的平均时延最高,CCMQ次之,在消息生存时间达到270 min以后,CCMQ的平均时延比DLA和DHLTF低,这是因为在消息生存时间达到250 min后CCMQ

对于消息的选择效果更好,减少了消息的转发次数,随着消息生存时间越来越长,CCMQ的平均时延趋于稳定。SHLI的平均时延最低,但是当消息的生存时间增长到一定的值时,CCMQ的平均时延将会是最低的。

综上所述,CCMQ在消息投递率和网络负载率方面较其他5种算法有了较大的提升,在平均时延方面,随着消息生存时间变长,CCMQ的平均时延趋于稳定,因此CCMQ在不同的消息生存时间下的网络性能更加优越,更加稳定。

### (2)不同的仿真时间

在不同的仿真时间条件下,比较和分析各种算法在3种路由指标上的变化情况,如图3所示。

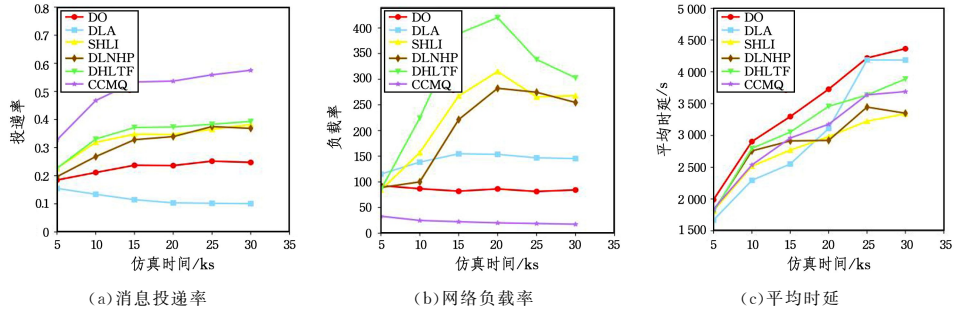


图3 各种算法在不同仿真时间下的性能

Fig. 3 Performance of different algorithms with different simulation times

图3(a)给出了各种算法在不同的仿真时间下消息投递率的变化情况。CCMQ的消息投递率最高,DLA的消息投递率最低,并且DLA的投递率随着仿真时间的增加而降低,其他5种算法都是增加的,这是因为随着仿真时间的增加,DLA删除消息过于盲目,导致消息传递不到目的节点。而CCMQ既选择了消息的转发,又选择了节点的接收,随着仿真时间的增加,传递到目的节点的消息就更多。

图3(b)给出了各种算法的网络负载率。CCMQ的网络负载率最低,其中CCMQ和DO的网络负载率随仿真时间的增加而降低,DHLTF,SHLI,DLNHP,DLA的网络负载率随仿真时间的增加而增加,这是因为当仿真时间变长时,CCMQ和DO的消息投递率提高,并且网络中生存时间很长的消息被删除,大大减少了消息在节点间的转发次数,降低了网络负载。从图3(b)中的数据对比可知,CCMQ对

网络的性能提高较大,效率较高。

图3(c)给出了在不同仿真时间下网络的平均时延的变化。从图3(c)可以看出,随着仿真时间的增加,CCMQ的平均时延逐渐趋于平缓,在仿真时间达到20 ks以后,CCMQ的平均时延比DO和DLA低,与DLNHP也相差不大。

综上所述,导致这些拥塞控制算法存在差异的关键原因是,当仿真时间变长时,网络中的消息数量将变多,消息在内存中存在的的时间和消息的质量度是影响网络性能的主要原因,因此CCMQ和SHLI较好地提升了网络性能。总体来看,CCMQ在消息投递率和网络负载率方面更具优势。

### (3)不同的消息产生速率

消息产生速率表示产生一个新消息所需要的时间,在不同的消息产生速率条件下,对比各种算法在网络性能上的差异,如图4所示。

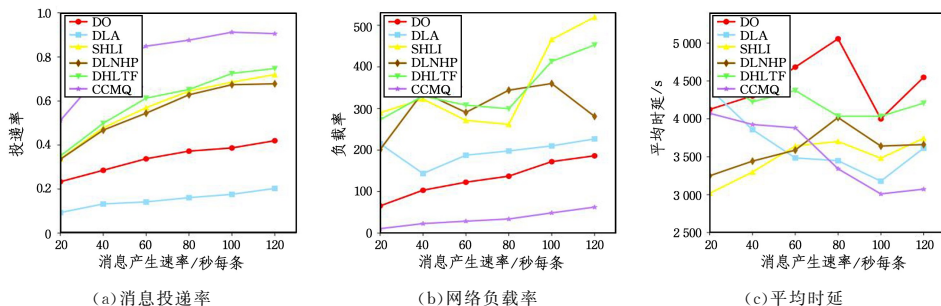


图4 各种算法在不同消息产生速率下的性能

Fig. 4 Performance of different algorithms with different message generation rates

图4(a)给出了各种算法的网络负载率的变化。其中,CCMQ的消息投递率最大,随着消息产生时间间隔逐渐变大,消息产生速率逐渐降低,6种算法的投递率都呈现上升的

趋势,这是因为随着消息产生速率逐渐降低,网络中的消息相较之前大量减少,网络资源利用率变高,消息在网络中被更加快速、准确地转发。但是,对于冗余消息的删除直接影响着消

息能否顺利被转发到目的节点。

图4(b)给出了各种算法的网络负载率随消息产生速率的变化情况。从图4(b)中可以看出,CCMQ的网络负载率最低,SHLI的网络负载率最高,随着消息生存速率逐渐降低,6种算法的网络负载都呈增长的趋势,这是由于当消息产生速率变低时,网络中存在的消息相对较少,资源相对充足,消息在网络中被转发的次数将会变多,成功投递到目的节点的消息变少,因而网络负载逐渐上升。但是,因为CCMQ对节点和消息做出了选择,所以消息的产生速率对CCMQ的影响最小。

图4(c)给出了各种算法在不同的消息产生速率下平均时延的变化情况。随着消息产生速率逐渐降低,DO和SHLI的平均时延呈现上升的趋势,DHLTF,DLNHP,DLA和CCMQ的平均时延呈现下降的趋势,当消息产生速率达到80秒每条以后,CCMQ的平均时延逐渐降为最低,并且一直保持下降的趋势。

综上所述,在不同的消息产生速率条件下,在投递率、负载率以及平均时延方面,CCMQ都是最优的,对网络的性能提升最大。

**结束语** 本文基于Epidemic路由算法提出了一种基于消息质量度和节点可信度的拥塞控制算法CCMQ。该算法根据消息质量度来决定消息转发的优先级,在消息转发时,选择质量度大的消息优先转发,以提高网络中消息共享资源的公平性。在选择中继节点进行转发时,节点只会选择可信度比自身大的节点进行消息的转发,同时还充分利用了节点自身的属性,提高了网络的总体性能,最后通过S-ACK机制优化了网络中的负载,进一步提升了网络的整体性能。仿真结果表明,CCMQ在消息投递率、网络负载率以及平均时延3个方面都优于其他的缓存管理策略,但是CCMQ在估计消息的扩散度方面还不太精确,与实际的数据有一定的差距,经组内研究讨论,接下来的研究将会从消息扩散度的角度出发,进一步考虑更加精确的方法来预估消息在网络中的扩散度。

## 参 考 文 献

- [1] GNZBOORG P, NIEMI V, JÖRGOT T. Fragmentation algorithms for DTN links[J]. *Computer Communications*, 2013, 36(3): 279-290.
- [2] FALL K, FARRELL S. DTN: an architectural retrospective [J]. *IEEE Journal on Selected Areas in Communications*, 2008, 26(5): 828-836.
- [3] FREEMAN L C. Centrality in Social Networks' Conceptual Clarification[J]. *Social Networks*, 1979, 1(3): 215-239.
- [4] BEAUCHAMP M A. An improved index of centrality[J]. *System Research and Behavioral Science*, 1965, 10(2): 161-163.
- [5] CLAUSET A. Finding local community structure in networks [J]. *Physical Review E*, 2005, 72(2): 026132.
- [6] LI F, SI Y L, CHEN Z, et al. Routing decision Method for Opportunistic Network Security based on trust mechanism [J]. *Journal of Software*, 2018(9): 2829-2843.
- [7] GANTAYAT P K, DAS S. Award—Reward Protocol Based on Clustering and Trust Level for Routing in Delay Tolerant Network[J]. *Journal of Applied Security Research*, 2019, 14(3): 270-287.
- [8] WANG R, WU Y H, HUANG H B, et al. Cooperative transmission in delay tolerant network[J]. *Journal of Systems Engineering & Electronics*, 2019, 30(1): 34-40.
- [9] CHEN Z G, YIN B A, WU J. Energy equalization routing algorithm for Opportunistic Networks based on message importance [J]. *Journal of Communications*, 2018, 39(12): 91-101.
- [10] ZHAO K, ZHANG Q. Network protocol architectures for future deep-space internetworking [J]. *ENCE China Information Ences*, 2018, 61(4): 040303.
- [11] TAO Y, GONG Z H. A node weight adaptive DTN congestion control [J]. *Computer Engineering and Science*, 2013, 35(1): 52-56.
- [12] ZHONG C C, LI T, WANG R C. Research on DTN Congestion Control based on enhanced PROPHET Routing [J]. *Journal of Nanjing University of Posts and Telecommunications: Natural Science*, 2016, 36(2): 118-123.
- [13] SOBIN C C. An Efficient Buffer Management Policy for DTN [J]. *Procedia Computer Science*, 2016, 93(7): 309-314.
- [14] MOETESUM M. An adaptive and efficient buffer management scheme for resource constrained delay tolerant networks [J]. *Wireless Networks*, 2016, 22(7): 1-13.
- [15] HUA D, DU X, XU G, et al. A DTN congestion mechanism based on Distributed Storage [C] // *IEEE International Conference on Information Management & Engineering*. IEEE, 2010.
- [16] WU D P, FU X J, ZHANG H P, et al. A delay tolerance network congestion control strategy based on node state awareness [J]. *Acta Electronica Sinica*, 2016, 39(5): 188-194.
- [17] VAHDAT A, BECKER D. Epidemic Routing for Partially Connected Ad Hoc Networks [R]. Technical Report CS2000-06, Duke University, Durham, 2000.
- [18] KERNEN A, OTT J, KRKKINEN T. The ONE simulator for DTN protocol evaluation [C] // *Proceedings of the 2nd International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SimuTools 2009)*. Rome, Italy, 2009.



**CUI Jian-qun**, born in 1974, Ph.D, professor, is a member of China Computer Federation. Her main research interests include opportunity network, Internet of things, mobile network and application layer multicast.



**CHANG Ya-nan**, born in 1984, Ph.D, associate professor, lecturer. Her main research interests include wireless network, social network and Internet of things.