计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2015.20140566
52(12):2707-2724,2015

基于虚拟货币的 DTNs 激励感知低时延路由

蒋庆丰[1,2] 门朝光[1] 李 香[1] 何忠政[1]
[1](哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001)
[2](大庆师范学院计算机科学与信息技术学院 黑龙江大庆 163712)
(qingfeng_jiang@163.com)

A Virtual Currency-Based Incentive-Aware Low Delay Routing for DTNs

Jiang Qingfeng[1,2], Men Chaoguang[1], Li Xiang[1], and He Zhongzheng[1]
[1](*College of Computer Science and Technology, Harbin Engineering University, Harbin* 150001)
[2](*College of Computer Science and Information Technology, Daqing Normal University, Daqing, Heilongjiang* 163712)

Abstract Due to the limited resources such as bandwidth, buffer, energy, and so on, most delay tolerant networks (DTNs) nodes are selfish and do not forward messages for other nodes to save their precious resources, which seriously degrades the routing performance. To stimulate the DTNs selfish nodes to cooperatively forward messages and reduce the message delivery delay, this paper proposes a virtual currency-based incentive-aware low delay routing algorithm, called VCILDR. A delay-based currency payment and allocation strategy is established to encourage selfish nodes to forward messages for other nodes in VCILDR. In this way, the direct beneficial messages are forwarded to the nodes with lower delivery delay and mutually beneficial messages are exchanged at the same time. A bargaining game model of alternating offers is established to determine the exchanged mutually beneficial messages. In addition, a greedy algorithm for solving the model's subgame perfect equilibrium is proposed in this paper. Extensive simulations are carried out on real-world dataset to verify the performance of this incentive-aware low delay routing. The experimental results show that the proposed routing can effectively stimulate DTNs selfish nodes to cooperatively forward messages for others, reduce the message delivery delay and improve the message delivery success ratio at the same time.

Key words delay tolerant networks (DTNs); incentive-aware; selfish; virtual currency; bargaining game

摘 要 由于带宽、缓存、能量等资源有限,延迟容忍网络(delay tolerant networks, DTNs)节点会具有一定的自私性.为节省宝贵的资源,自私节点会拒绝转发其他节点的消息,从而严重影响路由性能.为激励 DTNs 自私节点合作转发,减小消息传递时延,提出一种基于虚拟货币的激励感知低时延路由(virtual currency-based incentive-aware low delay routing, VCILDR).该路由通过建立基于时延的货币支付和分配策略,促使自私节点快速转发其他节点消息,将直接互利消息转发给传递时延小的节点,并交换可交换互利消息.建立轮流出价讨价还价博弈模型,以确定路由中节点的可交换互利消息,并提出一种求解该模型子博弈完美均衡的贪婪算法.在真实数据集上对该激励感知低时延路由的性能进行仿真验证.实验结果表明,该路由能够有效激励 DTNs 自私节点进行合作转发,减小消息传递时延,同时提高消息传递成功率.

收稿日期:2014-06-26;**修回日期**:2015-06-02
基金项目:国家自然科学基金项目(61100004);中央高校基本科研业务费专项资金项目(HEUCF 100607);黑龙江省教育厅科学技术研究项目(12543005)
通信作者:门朝光(menchaoguang@hrbeu.edu.cn)

关键词 延迟容忍网络;激励感知;自私;虚拟货币;讨价还价博弈

中图法分类号 TP393

延迟容忍网络(delay tolerant networks, DTNs)[1]是一种不存在稳定的端到端连接、具有长时延和间歇中断特点的网络,目前已广泛应用于社交网络、车载网络、灾难救援、动物迁徙追踪等领域. 当节点间链路中断时,常规的路由转发方式无法实现消息的传递,需要由中间节点采用"存储-携带-转发"方式临时存储消息,在和目的节点相遇时将消息发送给目的节点.

按照是否对消息进行复制,DTNs 路由协议可以分为 2 类:单副本路由[2-3]和多副本路由[4-6]. 单副本路由中只存在一份消息,节点将消息转发给合适的下一跳节点后立即删除缓存中的消息,下一跳节点按照同样的方式继续寻找下一跳节点,最终将消息发送给目的节点;多副本路由中每个消息同时存在多个副本,在带宽、缓存、能量等资源充足时可以增大消息传递的成功率并减小时延,但相对单副本路由会消耗更多的资源.

上述路由协议假设节点间是非自私、完全合作的,但由于带宽、缓存、能量等资源有限,节点会具有自私性,拒绝转发其他节点的消息,当网络中存在大量自私节点时,上述路由协议的时延等路由性能会严重下降[7-8]. 因此针对 DTNs 节点的自私性,设计一种能够促使节点进行合作转发,减小消息传递时延的激励感知路由具有重要意义.

为促使 DTNs 节点进行合作转发,减小消息的传递时延,本文提出一种基于虚拟货币的激励感知低时延路由(virtual currency-based incentive-aware low delay routing, VCILDR). VCILDR 中节点发送消息时,遇到传递时延小的转发节点时,会将消息发送给此转发节点,以减小消息传递时延,同时为激励转发节点转发消息,需要支付转发节点一定的货币值. 为促使节点快速转发消息,建立了基于时延的货币支付策略和多个转发节点间的货币分配策略,使得成功传递时延越小,转发节点获得货币值越大. VCILDR 中,如果节点将消息转发给传递时延小的对方节点后,其获得的货币值大于转发前的货币值,则直接将此消息转发给对方节点,实现节点间的合作;如果双方节点交换消息后,传递时延变小且获得的货币值大于交换前的货币值,则采取交换方式互相转发消息,实现节点间的合作.

本文主要贡献包括:

1) 提出一种基于时延的货币支付策略和成比例货币分配策略,使得消息源节点根据时延的大小支付相应的货币值并对转发节点进行货币分配.

2) 基于货币支付和分配策略,提出一种激励感知低时延路由 VCILDR. VCILDR 确定路由消息过程包括 4 个步骤:①确定对方消息;②确定自身可发送消息;③确定直接互利消息;④确定可交换互利消息.

3) 建立 2 节点交换可交换互利消息的轮流出价讨价还价博弈模型,并提出一种求解该博弈模型子博弈完美均衡的贪婪算法来确定节点的可交换互利消息.

1 相关工作

针对 DTNs 节点的自私性,如何设计有效的激励感知机制来促使节点进行合作,已经成为该领域的研究热点[9],目前已有一些学者进行了深入的研究. 为评估节点自私行为对路由性能的影响,文献[7]针对 Epidemic[4], Spray and Wait[5]等路由协议在不同节点合作度下的路由性能进行了仿真,结果表明节点合作度低时,路由性能会严重下降,但如果在上述路由转发机制中增加简单的激励合作机制将会明显改善路由性能. 文献[8]提出了一个理论框架来分析节点间不同合作程度对路由性能的影响,其结果表明一个适度的节点合作水平就能极大地提升节点完全非合作时的路由性能.

文献[10]提出的 SSAR 路由将 DTNs 节点自私性分为个体自私性和社会自私性,并且在选择下一跳节点时考虑了节点意愿、资源限制、接触机会等因素,在节点存在社会自私性时取得了较好的路由性能. 文献[11]分析了节点的个体自私性后,提出一种 DTNs 激励感知路由,利用"以牙还牙"(tit-for-tat, TFT)方式,即每个节点为邻居节点转发消息数量与邻居节点为其转发的消息数量相同的方式,来促使自私节点为其他节点转发消息. TFT 策略中节点在每一个间隔会根据确认消息数统计其他节点为自身转发的消息数,然后在下一间隔为其他节点转发相应的消息数. 为解决初始的时候,双方互相转发的消息数为 0,谁都不会为对方转发消息的初始启动问题,TFT 引入慷慨机制,即每个节点在下一

个间隔可以为对方节点多转发一定消息来促使节点合作转发.慷慨机制还可以在一定程度上解决双方节点消息不对称的问题.但 TFT 策略在每一间隔当节点间非对称消息的差额大于一定的消息数时,消息量小的节点不会转发多余的消息,并且在双方可转发消息量波动时,转发消息量不能即时由低的消息量增长到高的消息量,而需要多个间隔的迭代过程,因此影响转发效率.文献[12]针对 DTNs 的自私节点,提出一种基于虚拟货币的安全多层激励路由 SMART.SMART 使用层次货币作为节点转发消息的支付和奖励形式,并且为保证消息安全性,转发节点通过线性对加密技术对所在层进行签名.文献[13]针对 DTNs 的自私节点,设计了 Give2Get 协议.该协议中转发节点需要向源节点提供转发证明,如果将包丢弃不能提供证明,源节点将通过蜂窝网络通知授权中心,授权中心接收到通知后,会将自私节点排出系统.文献[14]针对机会网络中节点自私性问题,结合资源受限的机会网络特性,提出了一种基于买卖模型的节点激励策略,该策略采用货币支付模式,综合考虑节点自身资源、拥有的虚拟货币以及消息属性对消息进行定价,从而激励自私节点合作,同时有效地解决节点盲目合作带来的网络性能退化问题.文献[15]提出 MobiCent 策略来激励 DTNs 自私节点合作转发,该策略结合虚拟货币和加密技术来解决"边插入"和"边隐藏"攻击,并针对目标为最小支付和最小消息传递时延的节点,根据博弈论的算法机制设计理论提供了 2 种不同的支付策略来激励自私节点进行合作,但该激励策略只适用于多副本情况,无法在单副本情况下激励自私节点进行合作.

文献[16]设计了一种自私性 DTNs 中网络服务商和移动节点间可以达到共赢的互惠激励机制,与无激励机制相比,该激励机制能够不断提高网络分发性能.针对 DTNs 的节点自私性,文献[17]提出一种基于交换思想的数据分发机制.该机制中由一些特殊的源节点产生消息,每个节点从源节点下载自己感兴趣的消息为主消息,不感兴趣的消息为附属消息.当 2 个节点相遇,存在互补消息时,即自己的附属消息是对方的主消息,则可以通过交换的方式获得自己的主消息,从而提高消息传递成功率,减小发送时延.为促进自私节点进行合作转发,文献[18]提出 MuRIS 激励策略来促使自私节点进行数据分发,同时 MuRIS 还能够阻止节点的"边插入"攻击.

综上所述,不同于文献[10-14],本文提出的 VCILDR 专注于如何激励自私节点快速转发消息,减小消息传递时延.文献[15]提出的减小消息传递时延的激励感知路由只适合多副本路由,而 VCILDR 是在单副本路由下激励节点快速转发消息.文献[16-18]研究的是一对多的数据分发机制,不同于本文研究的一对一单播路由模式.

2 系统模型

VCILDR 系统模型如图 1 所示,系统包括如下 3 部分:

1) 可信的第三方(trusted third party,TTP)

TTP 作为授权中心负责为每个移动节点颁发公、私密钥证书并且根据票据信息完成 DTNs 节点的货币清算工作.

2) 固定网络

固定网络包括有线的 Internet 和无线访问点 AP 节点,负责连接 TTP 和移动的 DTNs 节点.DTNs 节点可通过 Internet 和 AP 节点由 TTP 处获得公、私密钥.当消息的目的节点和 AP 节点相遇时,将票据信息通过 AP 和 Internet 发送给 TTP,使得 TTP 完成 DTNs 节点货币清算工作.

3) DTNs 节点

图 1 中 $N_1 \sim N_7$ 为 DTNs 节点,代表携带具有蓝牙、WIFI 等高速、短距离无线通信功能设备的行人或车辆.DTNs 节点会向其他远距离的节点发送交通、位置等消息,这些消息的价值会随时间的增长而降低.由于 DTNs 节点的移动和无线通信范围短,远距离节点间链路会经常中断,不存在稳定的端到端连接,所以发送消息时需要由中间节点采用"存

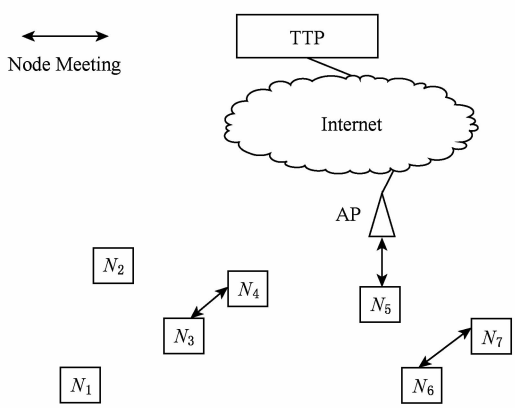


Fig. 1 VCILDR system model.

图 1 VCILDR 系统模型

储-携带-转发"方式进行转发.因此如何设计有效的激励策略来促使 DTNs 节点合作转发消息,减小消息传递时延具有重要意义.

VCILDR 系统模型的工作流程如下:

DTNs 节点携带自己的消息,遇到传递时延比自身小的节点则通过此节点进行转发,并支付转发节点一定的货币值;在缓存有空闲时,则为其他节点转发消息,以赚取一定的货币值.每个节点在为其他节点转发消息时的目标是通过转发不同的消息获得最大的货币值.

源节点 S 发送的消息经过一些转发节点的转发到达目的节点 D 后,目的节点 D 生成并保存如图 2 所示的转发票据,当 D 和 AP 相遇时将转发票据发送给 TTP,由 TTP 完成货币清算工作,将源节点支付的货币按照 3.1 节的货币支付和分配策略分给转发节点.

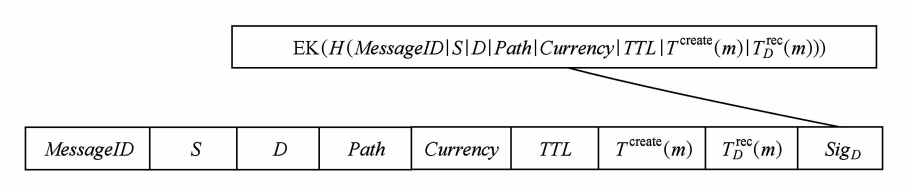


Fig. 2 Forwarding ticket.

图 2 转发票据

转发票据中:

1) $MessageID$ 表示消息标识;

2) S 表示源节点;

3) D 表示目的节点;

4) $Path$ 表示消息转发路径;

5) $Currency$ 表示节点 S 准备为消息支付的最大货币值;

6) TTL 表示消息的生命期;

7) $T^{\text{create}}(m)$ 如表 1 所示,表示消息 m 的创建时刻;

8) $T_D^{\text{rec}}(m)$ 如表 1 所示,表示节点 D 接收到消息 m 的时刻;

9) Sig_D 表示目的节点 D 对消息摘要信息的签名,以保证消息的完整性和机密性.

3 基于虚拟货币的激励感知低时延路由

3.1 货币支付和分配策略

为促使其他节点为源节点快速转发消息,源节点需要根据一定的货币支付策略为所有转发节点支付货币.当消息转发成功后,TTP 需要将源节点支付的货币按照一定的货币分配策略分配给转发节点.本文提出基于时延的货币支付策略和成比例货币分配策略.

1) 基于时延的货币支付策略

为使消息源节点根据时延大小,对消息转发节点进行货币奖励,提出如式(1)所示的基于时延的货币支付策略.

$$R(t)=\begin{cases}Currency(1-t/TTL), \\ \quad 0<t<TTL, \\ 0,\ t\geqslant TTL,\end{cases} \tag{1}$$

其中,$R(t)$ 为源节点为中间转发节点实际支付的货币总值;$Currency$ 为源节点准备为消息支付的最大货币值;TTL 为消息的生命期;t 为将消息传递到目的节点所需时延.

可以看出传递时延越小,转发节点所获得的货币值越大,当传递时延超过消息的生命期 TTL 时,获得的货币值为 0.

2) 成比例货币分配策略

当消息被成功传递到目的节点后,TTP 需为转发路径上的转发节点分配货币.由于 DTNs 中不存在稳定的端到端路径,节点无法预知消息的跳数及转发路径,如果采用平均分配策略,则当一个转发节点将消息转发给下一跳时,会无法确定其将来获得的货币值,转发节点越多,其获得的货币值会越少,因此本文提出如式(2)所示的成比例货币分配策略:

$$R_l=\begin{cases}R(1-r)^{l-1}\times r,\ 1\leqslant l<L, \\ R(1-r)^{l-1},\ l=L,\end{cases} \tag{2}$$

其中,L 为消息的转发节点数;r 为分配比例因子,$0<r<1$;R_l 为第 l 个转发节点获得的货币值;R 为由式(1)计算出的转发节点货币总值,R 和 R_l 满足式(3)的关系:

$$\sum_{l=1}^{L}R_l=R\times\left(r\times\sum_{l=1}^{L-1}(1-r)^{l-1}+(1-r)^{L-1}\right)=R. \tag{3}$$

成比例货币分配策略中每个转发节点获得的货币值与自身所处的转发次序有关,而与转发节点数无关.

3.2 相关定义

本文相关符号及意义如表 1 所示:

Table 1 Notation Description

表 1 符号描述

Notation	Description
$T^{\text{create}}(m)$	The creation time of message m
$T_N^{\text{rec}}(m)$	The time of message m is received by node N
$Time_{N_1-N_2}^{\text{meet}}$	The inter-contact time between node N_1 and node N_2
$Time_{N_1-N_2}^{\text{dur}}$	The duration of contact between node N_1 and node N_2
$Time_N^{\text{delivery}}(m)$	The delivery delay of node N for message m
$R_N^{\text{Before}}(m)$	Currencies obtained by node N before forwarding message m to another node
$R_N^{\text{After}}(m)$	Currencies obtained by node N after forwarding message m to another node
$R_N^{\text{Receive}}(m)$	Currencies obtained by node N when receiving message m
$R_{N1}(m_{N_1}\leftrightarrow m_{N_2})$	The more currencies obtained by node N_1 after exchanging message m_{N_1} for m_{N_2} of node N_2
$H_N(m)$	The forwarding order of node N in the forwarding path of message m

定义 1. 自身可发送消息. 节点 A,B 相遇时,对于节点 A 产生的目的节点为 D 的消息 m_A,如果 B 的传递时延小于 A 的传递时延,即满足式(4),则称 m_A 为节点 A 的自身可发送消息.

$$Time_B^{\text{delivery}}(m_A) < Time_A^{\text{delivery}}(m_A), \tag{4}$$

其中,节点 A,B 的传递时延计算方法如下:

同文献[19-20]一样,假设 DTNs 节点间相遇时间服从参数为 λ_{AD} 的指数分布,λ_{AD} 为节点 A 和 D 的接触率,其值如式(5)所示:

$$\lambda_{AD} = n \Big/ \sum_{l=1}^{n} Time_{A\text{-}D}^{\text{meet}l}, \tag{5}$$

其中,$Time_{A\text{-}D}^{\text{meet1}}, Time_{A\text{-}D}^{\text{meet2}}, \cdots, Time_{A\text{-}D}^{\text{meet}n}$ 为节点 A,D 的 n 次相遇时间样例,$1/\lambda_{AD}$ 代表节点相遇时间的均值. 对于目的节点为 D 的消息 m_A,节点 A,B 的传递时延分别为

$$Time_A^{\text{delivery}}(m_A) = 1/\lambda_{AD} + T_A^{\text{rec}}(m_A) - T^{\text{create}}(m_A),$$
$$Time_B^{\text{delivery}}(m_A) = 1/\lambda_{BD} + T_B^{\text{rec}}(m_A) - T^{\text{create}}(m_A).$$

定义 2. 直接互利消息. 节点 A,B 相遇时,对于节点 A 缓存中目的节点为 D 的非自身消息 m_A,如果将消息 m_A 转发给 B 后,其获得的货币值大于转发前的货币值即满足式(6),则称 m_A 为节点 A 的直接互利消息.

$$R_A^{\text{After}}(m_A) > R_A^{\text{Before}}(m_A). \tag{6}$$

由式(2)可知:

$$R_A^{\text{After}}(m_A) = R(Time_B^{\text{delivery}}(m_A)) \times (1-r)^{H_A(m_A)-1} \times r,$$
$$R_A^{\text{Before}}(m_A) = R(Time_A^{\text{delivery}}(m_A)) \times (1-r)^{H_A(m_A)-1}.$$

定义 3. 可交换互利消息. 节点 A,B 相遇,对于节点 A 中目的节点为 D_A 的非自身消息 m_A 和节点 B 中目的节点为 D_B 的非自身消息 m_B,如果:1)B 传递 m_A 的时延小于 A 的传递时延,A 传递 m_B 的时延小于 B 的传递时延;2)m_A 和 m_B 均不是直接互利消息;3)A,B 交换消息后,双方获得的货币值大于交换前的货币值,即满足式(7),则称消息 m_A 和 m_B 分别为 A,B 的可交换互利消息.

$$\begin{aligned} &Time_B^{\text{delivery}}(m_A) < Time_A^{\text{delivery}}(m_A) \wedge \\ &Time_A^{\text{delivery}}(m_B) < Time_B^{\text{delivery}}(m_B) \wedge \\ &!(R_A^{\text{After}}(m_A) > R_A^{\text{Before}}(m_A)) \wedge \\ &!(R_B^{\text{After}}(m_B) > R_B^{\text{Before}}(m_B)) \wedge \\ &R_A(m_A\leftrightarrow m_B) > 0 \wedge \\ &R_B(m_A\leftrightarrow m_B) > 0, \end{aligned} \tag{7}$$

其中:

$$R_A(m_A\leftrightarrow m_B) = R_A^{\text{Receive}}(m_B) + R_A^{\text{After}}(m_A) - R_A^{\text{Before}}(m_A),$$
$$R_B(m_A\leftrightarrow m_B) = R_B^{\text{Receive}}(m_A) + R_B^{\text{After}}(m_B) - R_B^{\text{Before}}(m_B).$$

由式(2)可知:

$$R_A^{\text{Receive}}(m_B) = R(Time_A^{\text{delivery}}(m_B)) \times (1-r)^{H_A(m_B)-1},$$
$$R_B^{\text{Receive}}(m_A) = R(Time_B^{\text{delivery}}(m_A)) \times (1-r)^{H_B(m_A)-1}.$$

假设节点将消息交换给对方后,对方传递此消息的时延要小于自身,否则 TTP 会通过转发票据的路径信息查找转发节点,如果发现节点将消息转发给传递时延大于自身的节点,则对其进行惩罚.

定义 4. 可交换互利消息对. 可交换互利消息 m_A 和 m_B 的组合称为可交换互利消息对,表示为 $m_A\leftrightarrow m_B$.

可交换互利消息和可交换互利消息对是一对多关系,即一个可交换互利消息可以和其他节点的多个可交换互利消息组成多个可交换互利消息对.

定义 5. 完全交换. 节点 A,B 的所有可交换互利消息对组成的集合称为节点 A,B 的完全交换.

定义 6. 交换方案. 如果节点 A,B 的一个可交换互利消息对集合 $S_{A\leftrightarrow B}$ 为完全交换的子集,且节点 A,B 中都不存在相同的可交换互利消息,则称 $S_{A\leftrightarrow B}$ 为节点 A,B 的一个交换方案.

3.3 激励感知低时延路由

在带宽和缓存有限情况下,节点 A,B 相遇后确定消息发送集合 $Send_A$,$Send_B$,然后按照 $Send_A$,$Send_B$ 中的消息顺序向对方节点发送消息. 确定消息发送集合 $Send_A$,$Send_B$ 的步骤如下:

1) 确定对方消息

如果节点 A 缓存中消息 m_A 的目的节点为 B,则将消息 m_A 加入 $Send_A$. 同理,B 将缓存中目的节点为 A 的消息 m_B 加入 $Send_B$.

2) 确定自身可发送消息

如果消息 m_A 为节点 A 自身可发送消息,且节点 B 同意接收 m_A,则将 m_A 加入消息发送集合 $Send_A$. B 根据获得的货币值大小进行缓存管理,来决定是否同意接收消息 m_A. B 缓存管理算法如算法 1 所示:

算法 1. 缓存管理.

① $SortDescend\ (R(Time_B^{\text{delivery}}(m_B))/Size(m_B))$;
/* 按照消息单位货币值大小进行降序排序 */
② $freeBuffer \leftarrow GetFreeBufferSize()$;
/* 获取空闲缓存大小 */
③ while $freeBuffer < Size(m_A)$
④ $m_{\min} = GetMinUnitRewardMessage()$;
/* 选取单位货币值最小的消息 */
⑤ if $R_B^{\text{Receive}}(m_A)/Size(m_A) \leqslant R(Time_B^{\text{delivery}}(m_{\min}))/Size(m_{\min})$ then
⑥ return false; /* 不同意接收消息 */
⑦ else
⑧ 增加 $m_{\min}$ 到 $deleteSet$ 中; /* 加入待删除消息集合 */
⑨ $freeBuffer \leftarrow freeBuffer + Size(m_{\min})$;
⑩ end if
⑪ end while
⑫ delete $deleteSet$;
⑬ $freeBuffer \leftarrow freeBuffer - Size(m_A)$;
⑭ return true. /* 同意接收消息 */

B 按照非自身消息的单位货币值 $R(Time_B^{\text{delivery}}(m_B))/Size(m_B)$ 大小进行降序排序,如果当前空闲缓存空间小于消息 m_A 的大小,则选取单位货币值最小的消息 $m_{\min}$,如果 $m_{\min}$ 的单位货币值小于接收消息 m_A 后获得的单位货币值 $R_B^{\text{Receive}}(m_A)/Size(m_A)$,则将 $m_{\min}$ 加入到待删除消息集合 $deleteSet$,接着选择下一个单位货币值最小的消息,直到空闲缓存大于 m_A 的大小或遇到单位货币值大于 m_A 的消息. 如果空闲缓存大于 m_A 的大小,则同意接收 m_A,并删除 $deleteSet$ 中的消息,否则拒绝接收 m_A. 当 B 删除消息 $m_{\min}$ 时,需要对 $m_{\min}$ 转发路径中前几跳节点按照式(2)进行货币赔偿. B 确定自身可发送消息过程同 A.

3) 确定直接互利消息

如果消息 m_A 为节点 A 的直接互利消息,且节点 B 同意接收 m_A,则节点 A 将 m_A 加入消息发送集合 $Send_A$. 节点 B 根据自身缓存和获得货币值情况判断是否同意接收 m_A 的过程同步骤 2). 节点 B 确定其直接互利消息的过程同节点 A.

定理 1. 节点一定将直接互利消息发送给传递时延小于自身的节点.

证明. 不失一般性,假设节点 A 将直接互利消息 m_A 发送给节点 B,则:

$$\overset{\text{式(6)}}{\Rightarrow} R_A^{\text{After}}(m_A) > R_A^{\text{Before}}(m_A),$$

$$\overset{\text{式(2)}}{\Rightarrow} R(Time_B^{\text{delivery}}(m_A)) \times r > R(Time_A^{\text{delivery}}(m_A)),$$

$$\overset{\text{式(1)}}{\Rightarrow} (1 - Time_B^{\text{delivery}}(m_A)/TTL) \times r > 1 - Time_A^{\text{delivery}}(m_A)/TTL,$$

$$\overset{0<r<1}{\Rightarrow} Time_B^{\text{delivery}}(m_A) < Time_A^{\text{delivery}}(m_A).$$ 证毕.

4) 确定可交换互利消息

节点 A,B 进行轮流出价讨价还价博弈确定双方可交换互利消息,然后将各自的可交换互利消息分别加入消息发送集合 $Send_A$,$Send_B$. 3.4 节详细阐述节点 A,B 的轮流出价讨价还价博弈过程.

3.4 确定可交换互利消息的轮流出价讨价还价博弈

3.4.1 轮流出价讨价还价博弈模型

假设节点间不能双向同时传输数据,即节点 A 向节点 B 发送完数据后,节点 B 才可以向 A 发送数据,节点间消息交换空间大小为 $ExVol$,则表示节点进行消息交换可多获得的货币值和交换方案 $S_{A\leftrightarrow B}$ 关系的收益函数 $U_N(S_{A\leftrightarrow B})(N=A,B)$ 为

$$\begin{aligned} &U_N(S_{A\leftrightarrow B}) = \sum_{m_A\leftrightarrow m_B \in S_{A\leftrightarrow B}} R_N(m_A \leftrightarrow m_B) \\ &\text{s.t.} \sum_{m_A\leftrightarrow m_B \in S_{A\leftrightarrow B}} Size(m_A + m_B) \leqslant ExVol. \end{aligned} \tag{8}$$

在确定可交换互利消息时,选择不同的消息进行交换,节点获得的货币值将不同. 由于节点自私性,每个节点都会选择能够带来最大收益的交换方案,即目标为 $\max U_N$. Rubinstein[21] 提出的轮流出价讨价还价博弈是通过非合作博弈理论来研究讨价还价问题的,适合于节点的效用最大化思想[22],因此,本文通过建立节点 A,B 交换消息的轮流出价讨

价还价博弈模型来确定双方可交换互利消息.

节点 A,B 确定可交换互利消息的轮流出价讨价还价博弈模型如图 3 所示:假设节点间数据传输速度为 $Speed$,则交换可交换互利消息的时间 $t=ExVol/Speed$. 假设讨价还价次数为 k,节点 A,B 双方采取轮流出价方式进行协商,由 A 先出价,提出一种交换方案 s_A^1,如果节点 B 接受此方案,则达成协议,停止继续讨价,双方获得的收益分别为 $U_A(s_A^1)$ 和 $U_B(s_A^1)$,否则在经过 t/k 时间之后进行第 2 次讨价,由节点 B 先提出交换方案 s_B^2,节点 A 根据其获得的收益决定是否同意接受,依此类推,直到第 k 次讨价,节点 B 出价,提出交换方案 s_B^k,如果节点 A 接受可获得收益 $U_A(s_B^k)$,否则收益为 0. 可以看出,如果每次协商不成功会使交换空间减小 $ExVol/k$,给双方都带来收益损失,直到第 k 次没有协商成功,双方没有交换任何消息,获得的收益均为 0.

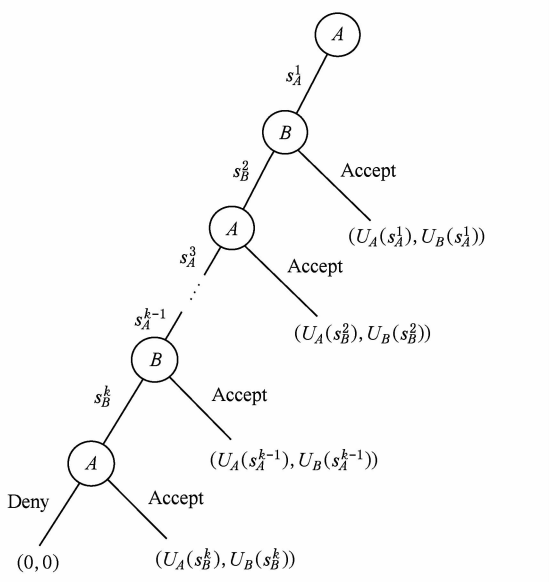


Fig. 3 Bargaining game model of alternating offers.

图 3 轮流出价讨价还价博弈模型

轮流出价讨价还价博弈是一个典型的完全信息动态博弈,节点 A,B 讨价还价博弈模型的完全信息动态博弈扩展式为[23]

$$\Gamma=\{N,H,P,U\},$$

其中:

① 参与者集合 $N=\{A,B\}$;

② 全历史集合:

$$\begin{aligned}H=\{&(s_A^1,\text{Accept}),(s_A^1,s_B^2,\text{Accept}),\cdots,\\&(s_A^1,s_B^2,s_A^3,\cdots,s_A^{k-1},\text{Accept}),\\&(s_A^1,s_B^2,s_A^3,\cdots,s_A^{k-1},s_B^k,\text{Accept}),\\&(s_A^1,s_B^2,s_A^3,\cdots,s_A^{k-1},s_B^k,\text{Deny})\};\end{aligned}$$

③ 参与者函数:

$$\begin{aligned}P(\varnothing)&=A,\\P(s_A^1)&=B,\\P(s_A^1,s_B^2)&=A,\cdots,\\P(s_A^1,s_B^2,s_A^3,\cdots,s_A^{k-1})&=B,\\P(s_A^1,s_B^2,s_A^3,\cdots,s_A^{k-1},s_B^k)&=A;\end{aligned}$$

④ 收益函数:

节点 A,B 第 i 次讨价提出交换方案的收益函数 $U(s_A^i)$,$U(s_B^i)$ 如式(8)所示,其中第 i 次讨价的交换空间为 $ExVol\times(k-i+1)/k$.

3.4.2 轮流出价讨价还价博弈的子博弈完美均衡

定义 7. 子博弈完美均衡[23]. 在完全信息动态博弈 $\Gamma=\{N,H,P,U\}$ 中,如果一个策略组合 s^* 在所有该博弈的子博弈 $\Gamma(h)=\{N_s,H_s,P_s,U_s\}$ 中都是纳什均衡,那么我们就称策略组合 s^* 为子博弈完美均衡,即对于任意参与者 $i\in P_s$ 和任意 s_i,式(9)成立.

$$U_i(O_h(s^*))\geqslant U_i(O_h(s_i,s_{-i}^*)),\tag{9}$$

其中,h 为任意真子历史,$O_h(\cdot)$ 为子博弈 $\Gamma(h)$ 的结果函数.

节点 A,B 间确定最终交换消息的过程即为求解轮流出价讨价还价博弈的子博弈完美均衡过程,子博弈完美均衡结果即为最终确定的交换方案. Rubinstein[21] 假设贴现因子 $\delta(0<\delta<1)$ 固定时,证明了子博弈完美均衡的存在性,其中 δ 表示节点在第 i 次协商中结束讨价还价时获得的收益与在 $i-1$ 次协商中结束讨价还价时获得的收益的比值. 本文节点 A,B 的讨价还价博弈模型中,节点在第 i 次协商中结束讨价还价时获得的收益小于等于在 $i-1$ 次协商中结束讨价还价时获得的收益,但不存在固定的比例关系. 虽然不存在固定的贴现因子,但节点 A,B 的讨价还价博弈模型中,同样存在子博弈完美均衡且在第 1 次讨价还价中即可确定子博弈完美均衡结果.

定理 2. 假设当接受和拒绝其他节点的交换方案获得的收益相同时,节点会选择接受,则节点 A,B 的有限次轮流出价讨价还价博弈必存在子博弈完美均衡且在第 1 次讨价还价中即可确定子博弈完美均衡结果.

证明. 采用逆推法来求解子博弈完美均衡. 逆推法是指从动态博弈的最后一步往回推,以求解动态博弈的均衡结果. 如图 3 所示,在第 k 次讨价时,由 B 提出交换方案,节点 A 确定是否接受,如果不接受则双方不会交换任何消息,双方收益均为 0. 因

为当接受和拒绝其他节点的方案获得的收益相同时,节点会选择接受,所以 A 必然会接受 B 提出的任何满足 $U_A(S_B^k)\geqslant 0$ 的方案 S_B^k. 因此如图 4 中①所示,B 的最优方案为在交换空间小于等于 $ExVol/k$ 情况下选取能够给自身带来最大收益即 $U_B(S_B^{*k})=\max U_B(S_B^k)$,并且使得 $U_B(S_B^{*k})$ 和 $U_A(S_B^{*k})$ 都大于等于 0 的方案 S_B^{*k}. 在第 $k-1$ 次讨价时,由节点 A 提出交换方案 S_A^{k-1},节点 B 确定是否接受,节点 B 只有在其获得的收益大于等于 $U_B(S_B^{*k})$ 时才会接受节点 A 的方案,所以节点 A 的最优方案如图 4 中②所示,为在交换空间小于等于 $2\times ExVol/k$ 情况下选取能够给自身带来最大收益 $U_A(S_A^{*(k-1)})=\max U_A(S_A^{k-1})$,并且使得 $U_A(S_A^{*(k-1)})\geqslant U_A(S_B^{*k})$ 和节点 B 的收益 $U_B(S_A^{*(k-1)})\geqslant U_B(S_B^{*k})$ 的交换 $S_A^{*(k-1)}$. 可以看出,每次讨价节点的最优方案为满足其下一次讨价双方节点收益情况下,最大化自己的收益. 假设第 $i(1\leqslant i\leqslant k)$ 次讨价由节点 A 出价,交换空间为 $ExVol\times(k-i+1)/k$,节点 A 的最优方案为满足式(10)的 S_A^{*i}.

$$\begin{aligned} &U_A(S_A^{*i})=\max U_A(S_A^i),\\ &\text{s.t. } U_A(S_A^{*i})\geqslant U_A(S_B^{*(i+1)})\ \wedge\\ &\quad U_B(S_A^{*i})\geqslant U_B(S_B^{*(i+1)}). \end{aligned}\tag{10}$$

节点 A 为最大化自己的收益而确定最优方案过程是一个 0-1 背包过程,因为第 i 次讨价的交换空间比第 $i+1$ 次讨价的交换空间大 $ExVol/k$,所以必存在满足式(10)的 S_A^{*i}. 依次类推,直到第 1 次讨价,如图 4 中③所示,A 提出方案 S_A^{*1},B 必然接受. 所以节点 A,B 的轮流出价讨价还价博弈的子博弈完美均衡为:节点 A 在第 i(i 为奇数)次讨价选择方案 S_A^{*i},在第 $i+1$ 次讨价接受节点 B 的任何满足 $U_A(S_B^{i+1})\geqslant U_A(S_A^{*(i+2)})$ 的方案 S_B^{i+1},拒绝 $U_A(S_B^{i+1})<U_A(S_A^{*(i+2)})$ 的方案 S_B^{i+1},B 同理. 子博弈完美均衡结果即最终交换方案为第 1 次讨价确定的 S_A^{*1}.

证毕.

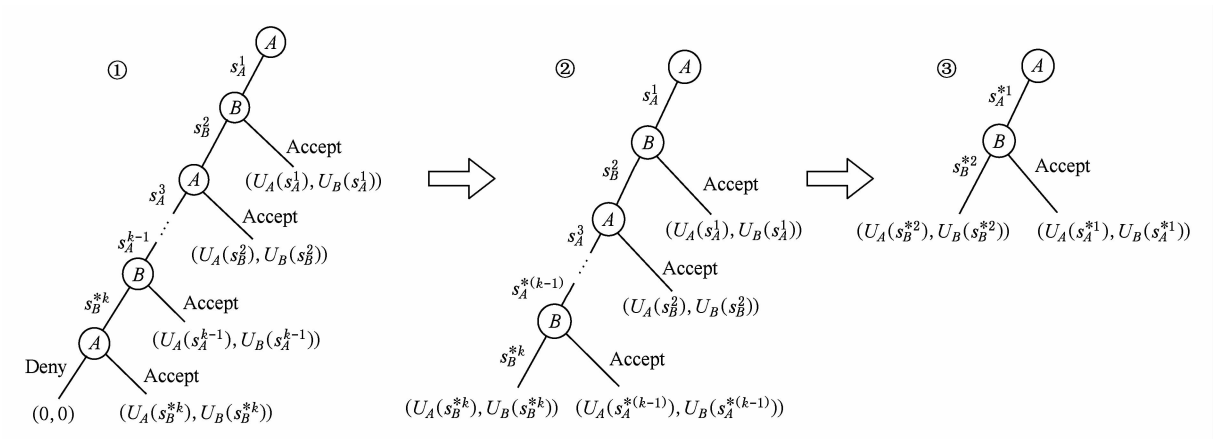


Fig. 4 Solving the subgame perfect equilibrium result.

图 4 求解子博弈完美均衡结果

3.4.3 求解子博弈完美均衡的算法

节点单次讨价决定交换方案的问题是 NP 难解的 0-1 背包问题,因此本文提出算法 2 所示的贪婪算法求解节点 A 的单次讨价交换方案,节点 B 同理.

算法 2. 求解单次讨价交换方案.

① $S_A^{*i}\leftarrow S_B^{*(i+1)}$;

② $sumSize\leftarrow Size(S_B^{*(i+1)})$; /* $sumSize$ 赋值为交换方案 $S_B^{*(i+1)}$ 的所有消息大小之和 */

③ $SortDescend(R_A(m_A\leftrightarrow m_B)/Size(m_A+m_B))$ in $FullExchange$; /* 按照单位收益大小对可交换互利消息对进行降序排序 */

④ for $m_A\leftrightarrow m_B$ in $FullExchange$

⑤ if m_A 和 m_B 不在 S_A^{*i} 中 then

⑥ $sumSize\leftarrow sumSize+Size(m_A+m_B)$;

⑦ if $sumSize>ExVol\times(k-i+1)/k$ then

⑧ $sumSize\leftarrow sumSize-Size(m_A+m_B)$;

⑨ else

⑩ 增加 $m_A\leftrightarrow m_B$ 到 S_A^{*i} 中;

⑪ end if

⑫ end if

⑬ end for

/* 行⑭~㉚替换 S_A^{*i} 中收益小的可交换互利消息对 */

⑭ $SortAscend(R_A(m'_A\leftrightarrow m'_B)/Size(m'_A+m'_B))$ in S_A^{*i};

/* 按照单位收益大小对 S_A^{*i} 中可交换互利消息对进行升序排序 */

⑮ for $m_A \leftrightarrow m_B$ in $FullExchange$

⑯ if m_A 和 m_B 不在 S_A^{*i} 中 then

⑰ for $m'_A \leftrightarrow m'_B$ in S_A^{*i}

⑱ $sumSize \leftarrow sumSize + Size(m_A + m_B) - Size(m'_A + m'_B)$;

⑲ $U_B(S_A^{*i}) \leftarrow U_B(S_A^{*i}) + R_B(m_A \leftrightarrow m_B) - R_B(m'_A \leftrightarrow m'_B)$;

⑳ $U_A(S_A^{*i}) \leftarrow U_A(S_A^{*i}) + R_A(m_A \leftrightarrow m_B) - R_A(m'_A \leftrightarrow m'_B)$;

㉑ if $R_A(m_A \leftrightarrow m_B) > R_A(m'_A \leftrightarrow m'_B) \wedge sumSize < ExVol \times (k-i+1)/k \wedge U_B(S_A^{*i}) \geqslant U_B(S_B^{*(i+1)})$ then

㉒ replace $m'_A \leftrightarrow m'_B$ with $m_A \leftrightarrow m_B$;

㉓ else

㉔ $sumSize \leftarrow sumSize - Size(m_A + m_B) + Size(m'_A + m'_B)$;

㉕ $U_B(S_A^{*i}) \leftarrow U_B(S_A^{*i}) - R_B(m_A \leftrightarrow m_B) + R_B(m'_A \leftrightarrow m'_B)$;

㉖ $U_A(S_A^{*i}) \leftarrow U_A(S_A^{*i}) - R_A(m_A \leftrightarrow m_B) + R_A(m'_A \leftrightarrow m'_B)$;

㉗ end if

㉘ end for

㉙ end if

㉚ end for

算法 2 中,设节点 A 在第 i 次讨价时决定交换方案,首先将 S_A^{*i} 赋值为 $S_B^{*(i+1)}$,然后按照交换消息后的单位收益 $R_A(m_A \leftrightarrow m_B)/Size(m_A + m_B)$ 即可多获得的单位货币值大小对完全交换集合 $FullExchange$ 中的可交换互利消息对进行降序排序;然后从上到下依次选取可交换互利消息对 $m_A \leftrightarrow m_B$,如果当前交换空间未满且不包含可交换互利消息 m_A 和 m_B,则将 $m_A \leftrightarrow m_B$ 加入 S_A^{*i},否则选择下一个可交换互利消息对,直到遍历所有可交换互利消息对,此时 $U_B(S_A^{*i}) \geqslant U_B(S_B^{*(i+1)})$.

接着按照节点 A 单位收益大小对 S_A^{*i} 中可交换互利消息对进行升序排序,按照单位收益从小到大的顺序依次选择可交换互利消息对 $m'_A \leftrightarrow m'_B$;然后从集合 $FullExchange$ 中按照 A 单位收益大小选择收益大的可交换互利消息对 $m_A \leftrightarrow m_B$,如果满足替换条件:

$$R_A(m_A \leftrightarrow m_B) > R_A(m'_A \leftrightarrow m'_B) \wedge sumSize + Size(m_A + m_B) - Size(m'_A + m'_B) < ExVol \times (k-i+1)/k \wedge U_B(S_A^{*i}) \geqslant U_B(S_B^{*(i+1)}),$$

则用收益大的 $m_A \leftrightarrow m_B$ 替换收益小的 $m'_A \leftrightarrow m'_B$. 因为 $U_B(S_A^{*i}) \geqslant U_B(S_B^{*(i+1)})$,所以节点 B 一定会接受此方案.

求子博弈完美均衡结果的算法如算法 3 所示:

算法 3. 求解子博弈完美均衡结果.

① 增加所用的 $m_A \leftrightarrow m_B$ 到 $FullExchange$ 中;

② $first, second \leftarrow Random(A, B)$; /* 随机决定节点 A、节点 B 首次出价次序 */

③ for i from k to 1 /* 假设 k 是奇数 */

④ if i is odd number then

⑤ 用 $first$ 执行算法 2;

⑥ else

⑦ 用 $second$ 执行算法 2;

⑧ end if

⑨ end for

算法 3 中首先将所有可交换互利消息对加入集合 $FullExchange$,然后随机决定节点 A,B 的首次出价次序,最后根据节点出价次序执行算法 2. 算法 3 的时间复杂度如定理 3 所示.

定理 3. 求解子博弈完美均衡算法的时间复杂度为 $O(kn^2)$,其中 k 为讨价还价次数,n 为可交换互利消息对数.

证明. 算法 2 中行①将 $S_B^{*(i+1)}$ 中可交换互利消息对赋值给 S_A^{*i} 的时间复杂度为 $O(n)$,行③对可交换消息交换对进行排序,如采用快速排序则时间复杂度为 $O(n \log n)$,行⑤判断 S_A^{*i} 中是否有包含 m_A 和 m_B 的可交换互利消息对的过程为循环遍历过程,时间复杂度为 $O(n)$,所以行④~⑬的时间复杂度为 $O(n^2)$,行⑭对 S_A^{*i} 中可交换互利消息对采用快速排序进行排序,则时间复杂度为 $O(n \log n)$,行⑯时间复杂度同行⑤为 $O(n)$,所以行⑮~㉚的时间复杂度为 $O(n^2)$,因此算法 2 时间复杂度为 $O(n) + 2O(n \log n) + 2O(n^2) = O(n^2)$,算法 3 的时间复杂度为 $O(kn^2)$. 证毕.

算法中交换空间 $ExVol$ 的计算如式(11)所示:

$$\begin{aligned} ExVol &= \min(ExVolNeed, VolRemain), \\ ExVolNeed &= \min(Count_A, Count_B) \times 2 \times Size_m^{\max}, \\ VolRemain &= VolTotal - Size(Send_A^{PreEx}) - Size(Send_B^{PreEx}), \\ VolTotal &= Time_{A-B}^{dur} \times Speed, \end{aligned} \tag{11}$$

其中,$ExVolNeed$ 为节点 A,B 交换可交换互利消息对所需的最大空间;$VolRemain$ 为节点 A,B 通信时确定可交换互利消息前所剩空间大小;$Count_A$,

$Count_B$ 分别为节点 A,B 的可交换互利消息数；$Size_m^{\max}$ 为节点 A,B 中最大的可交换互利消息的大小；$VolTotal$ 为节点 A,B 通信期间可传输数据的总空间大小；$Send_A^{PreEx}$ 和 $Send_B^{PreEx}$ 分别为节点 A,B 确定可交换互利消息前的发送集合.

由式(11)可知，计算 $ExVol$ 的大小需要计算出节点 A,B 相遇时进行通信的时间 $Time_{A-B}^{dur}$. 如图 5 所示，假设节点 A,B 的移动速度分别为 V_A,V_B，通信半径分别为 r'_A,r'_B，则当 2 节点间距离 $r'_{\min}=\min(r'_A,r'_B)$ 时，可以开始传输消息，在经过时间 $Time_{A-B}^{dur}$ 后 2 节点间的距离大于 $r'_{\min}$，则不能传输消息. 假设节点 A,B 可以传输消息时的坐标 L_A^1,L_B^1 分别为 $(X_{A1},Y_{A1}),(X_{B1},Y_{B1})$，目的点坐标 L_A^3,L_B^3 分别为 $(X_{A3},Y_{A3}),(X_{B3},Y_{B3})$，则计算出节点 A,B 通信结束时的坐标 L_A^2,L_B^2 后，即可计算出 $Time_{A-B}^{dur}$，具体求解过程如下：

$$Distance_A=\sqrt{(X_{A3}-X_{A1})^2+(Y_{A3}-Y_{A1})^2},$$
$$\cos\theta_A=(X_{A3}-X_{A1})/Distance_A,$$
$$\sin\theta_A=(Y_{A3}-Y_{A1})/Distance_A,$$
$$V_{AX}=V_A\times\cos\theta_A,$$
$$V_{AY}=V_A\times\sin\theta_A,$$
$$Distance_B=\sqrt{(X_{B3}-X_{B1})^2+(Y_{B3}-Y_{B1})^2},$$
$$\cos\theta_B=(X_{B3}-X_{B1})/Distance_B,$$
$$\sin\theta_B=(Y_{B3}-Y_{B1})/Distance_B,$$
$$V_{BX}=V_B\times\cos\theta_B,$$
$$V_{BY}=V_B\times\sin\theta_B,$$

其中，θ_A,θ_B 分别为节点 A,B 的移动方向和水平方向夹角；$Distance_A$ 为 (X_{A1},Y_{A1}) 和 (X_{A3},Y_{A3}) 的距离；$Distance_B$ 为 (X_{B1},Y_{B1}) 和 (X_{B3},Y_{B3}) 的距离；V_{AX},V_{AY} 分别为节点 A 的水平速度和垂直速度；V_{BX},V_{BY} 分别为节点 B 的水平速度和垂直速度.

则 L_A^2，L_B^2 的坐标分别为：$(X_{A1}+V_{AX}\times Time_{A-B}^{dur},Y_{A1}+V_{AY}\times Time_{A-B}^{dur})$ 和 $(X_{B1}+V_{BX}\times Time_{A-B}^{dur},Y_{B1}+V_{BY}\times Time_{A-B}^{dur})$. 根据 L_A^2,L_B^2 坐标和 2 节点间的距离 $r'_{\min}$，可通过如下一元二次方程组求解 $Time_{A-B}^{dur}$ 的值.

$$(X_{B1}+V_{BX}\times Time_{A-B}^{dur}-(X_{A1}+V_{AX}\times Time_{A-B}^{dur}))^2+(Y_{B1}+V_{BY}\times Time_{A-B}^{dur}-(Y_{A1}+V_{AY}\times Time_{A-B}^{dur}))^2=r'^2_{\min},$$
$$\Rightarrow A\times(Time_{A-B}^{dur})^2+B\times Time_{A-B}^{dur}+C=0,$$
$$A=(V_{BX}-V_{AX})^2+(V_{BY}-V_{AY})^2,$$
$$B=2\times((X_{A3}-X_{A1})\times(V_{BX}-V_{AX})+(X_{B3}-X_{B1})\times(V_{BY}-V_{AY})),$$
$$C=(X_{A3}-X_{A1})^2+(X_{B3}-X_{B1})^2-r'^2_{\min},$$
$$\Rightarrow Time_{A-B}^{dur}=\frac{-B\pm\sqrt{B^2-4AC}}{2A}.$$

$Time_{A-B}^{dur}$ 有 2 个值，一个为 0，另一个非 0 值即为所求解.

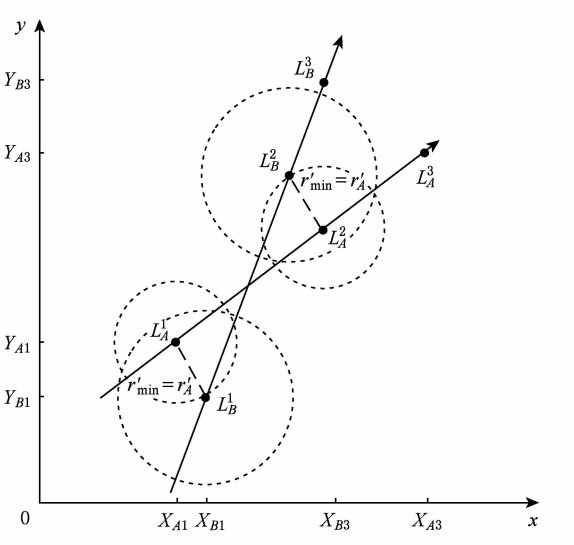


Fig. 5 Communication between nodes.

图 5 节点间通信

4 实验结果和分析

4.1 参数设置

本文采用 ONE[24] 模拟器对路由性能进行验证，ONE 是基于 Java 开发的适用于 DTNs 网络的路由模拟器，具有面向对象、离散事件驱动的特点，但目前还没有提供对物理层和数据链路层的支持. 在真实数据集 Infocom 05[25] 和 Infocom 06[26] 上对路由性能进行仿真验证. 这 2 个数据集通过采集人体携带的移动设备上的通信信息，代表了具有一定社会关系的人类社会网络模型. Infocom 05 数据集收集了 Infocom 05 会议上 41 个设备的信息，时间跨度为 274 883 s；Infocom 06 数据集收集了会议 78 个参会者携带设备的数据信息，时间跨度为 340 744 s. 数据传输速度为 250 KBps，消息大小 500～1 000 KB，消息产生间隔为 5～10 s，目的地址随机产生. 式(1)中每个消息的 $Currency=10$，式(2)中 $r=0.6$. 预热期 30 000 s，为防止尾部效应，最后 30 000 s 仿真结果不统计.

4.2 性能比较

将 VCILDR 和直接发送路由策略 Direct、完全

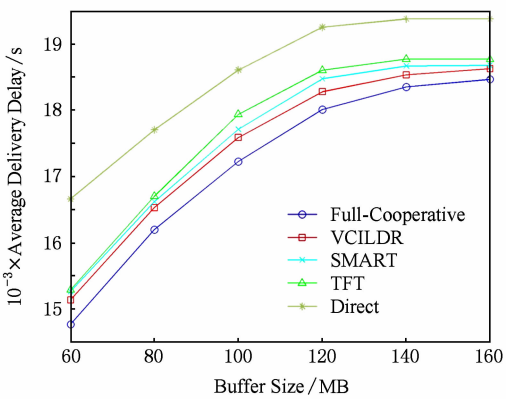
合作转发路由策略 Full-Cooperative 以及文献[11]的 TFT 策略、文献[12]的 SMART 策略进行性能对比. Direct 策略中每个节点只存储自身消息,不为其他节点转发消息,当遇到目的节点时将消息直接发送给目的节点,Full-Cooperative 策略中,每个节点按照消息的传递时延大小来缓存非自身节点消息,转发消息时先转发传递时延小的消息.

使用平均传递时延、传递成功率和负载率 3 个性能评价指标来对路由进行性能评价. 平均传递时延是指所有传递成功消息的传递时延的平均值;传递成功率为传递成功的消息数与发送消息数比例;负载率则为消息转发的总次数与传递成功的消息数的比值,表示成功传递一条消息,平均需要进行多少次转发.

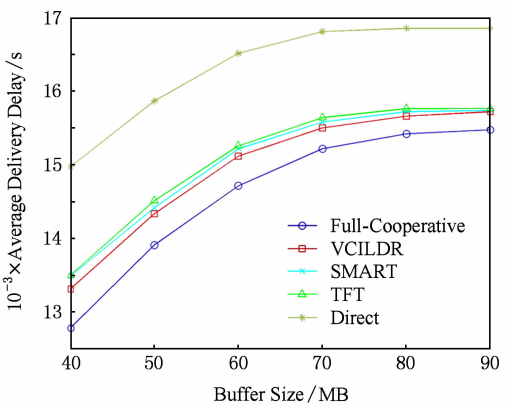
下面通过改变缓存大小、*TTL* 大小、数据传输速度等值来验证路由性能.

1) 缓存对性能的影响

如图 6 所示,可以看出随着缓存的增大,5 种路由策略的平均传递时延都在增大. Direct 策略中,当



(a) Infocom 05



(b) Infocom 06

Fig. 6 Impact of varying buffer sizes on average delivery delay.

图 6 缓存对平均传递时延的影响

缓存增大时节点可以多存储并成功传递部分时延大的消息,所以消息平均传递时延会增大;VCILDR,TFT,SMART,Full-Cooperative 策略中,随着缓存的增大,虽然由于缓存溢出而丢弃的消息会变少,增大消息的传递机会、减少时延. 但缓存的增大会使更多传递时延大的消息在缓存中存留更长时间,并被成功传递,所以平均传递时延会增大. 如果没有足够的缓存,这些时延大的消息会由于不能即时转发而被丢弃. 另外缓存的增大也会使更多的消息在节点数据传输速度有限和接触时间较短时不能被即时转发,增大消息的传递时延,所以平均传递时延随着缓存的增大而增大.

Direct 策略中,由于节点的消息不能够传递给传递时延小的节点,所以平均传递时延最大;由于 TFT 策略中消息非对称时,不能有效即时地转发消息,所以传递时延大于 SMART, VCILDR, Full-Cooperative 策略;由于 SMART 中转发消息时,每个转发节点获得的货币值是相同的,与时延无关,节点不会为减少时延而快速转发消息;而 VCILDR 中传递时延越小,获得的货币值越大,节点为赚取更多的货币值会优先转发、存储传递时延小的消息,从而传递时延小于 SMART 和 TFT 策略;Full-Cooperative 策略由于节点间完全合作转发,所以消息传递时延最小.

缓存对传递成功率的影响如图 7 所示,可以看出,缓存越大,消息传递成功率越大,这是因为缓存越大,节点可以存储更多的消息,减小消息由于缓存溢出而被丢弃的概率,增大消息的传递机会. Direct 策略中,由于每个节点只缓存自身消息,不能即时将消息转发给传递时延更小的其他节点,导致消息在缓存中存留时间过长,由于缓存有限或 *TTL* 过期会被丢弃,所以传递成功率最小;VCILDR 策略中,节点为获取货币会转发自身可发送消息和直接互利消息,同时交换可交换互利消息,所以成功率大于 Direct, TFT, SMART; Full-Cooperative 中由于节点完全合作转发,传递成功率最大.

缓存大小对负载率的影响如图 8 所示,由于 Direct 路由策略中节点只缓存自身消息,不经过其他节点转发,所以负载率始终为 1;其他策略的负载率随着缓存的增大而减小,这是因为随着缓存的增大,虽然成功转发消息数和转发次数都在增大,但成功转发消息数的增大速率大于转发次数的增大速率. Full-Cooperative 策略的负载率最大;由于 VCILDR 中节点间的有效转发,所以负载率大于 TFT 和 SMART 策略,小于 Full-Cooperative 策略.

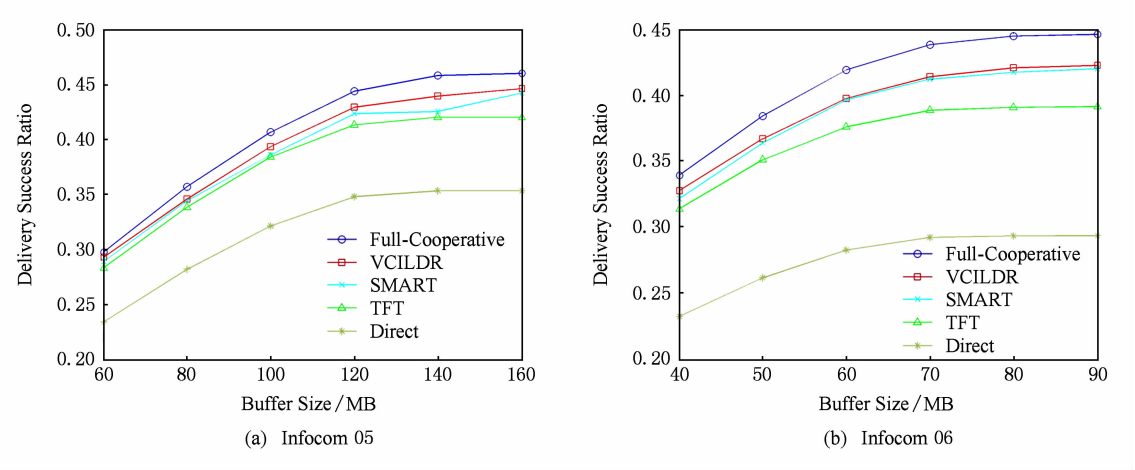


Fig. 7 Impact of varying buffer sizes on delivery success ratio.

图 7 缓存对传递成功率的影响

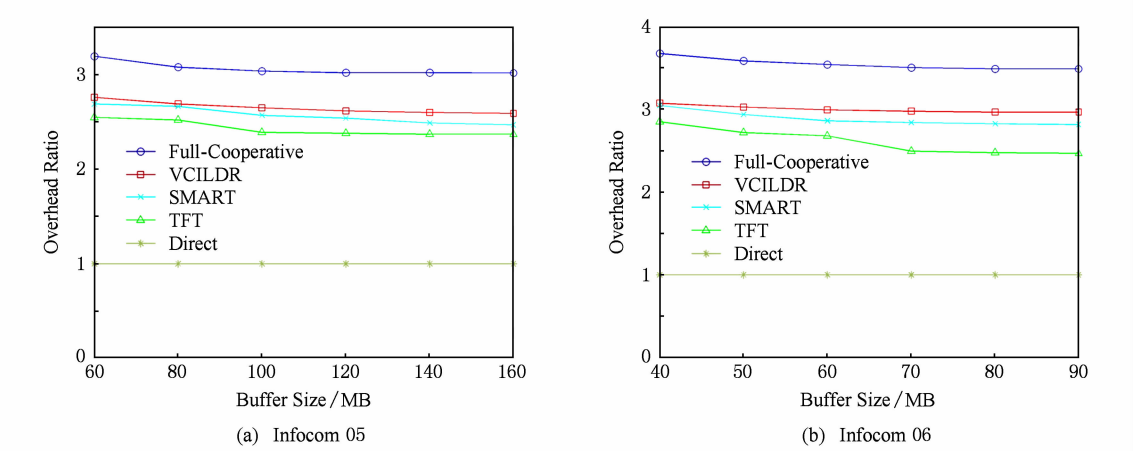


Fig. 8 Impact of varying buffer sizes on overhead ratio.

图 8 缓存对负载率的影响

2) *TTL* 对性能的影响

TTL 对时延的影响如图 9 所示,可以看出 5 种路由策略的消息传递时延都随着 *TTL* 的增大而增大. 因为 *TTL* 增大时,消息存活时间增大,节点可以成功传递更多时延大的消息,所以平均传递时延增大. Direct 策略中,由于每个节点只缓存自身消息,不能即时将自身消息转发给传递时延更小的其他节点,所以导致消息平均传递时延最大;Full-Cooperative

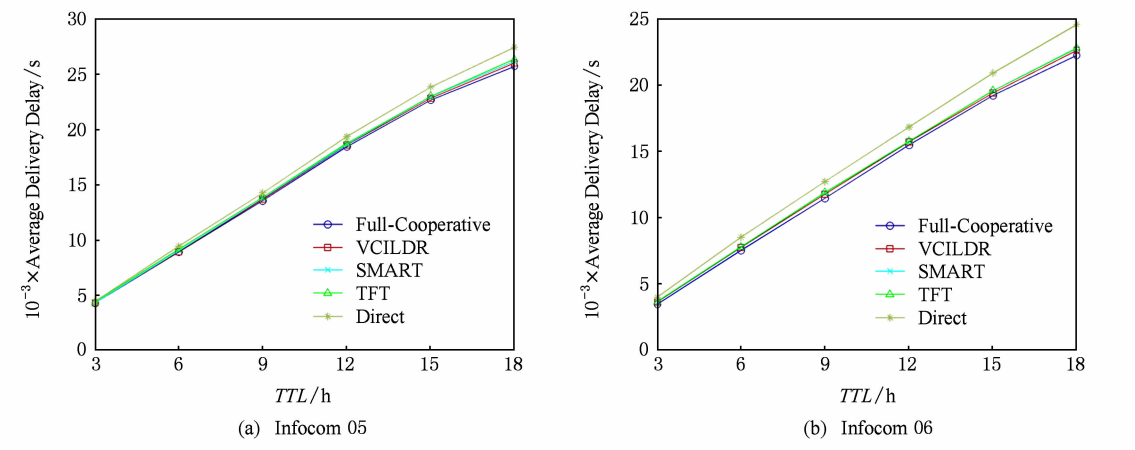


Fig. 9 Impact of varying *TTL* on average delivery delay.

图 9 *TTL* 对平均传递时延的影响

策略中,如果节点自身传递时延小于其他节点则会完全转发其他节点的消息,使消息被快速地传递给时延小的下一节点,所以其传递时延最小;VCILDR 中传递时延越小,获得的货币越多,为获取更多的货币,节点会快速转发消息,所以平均传递时延比 Direct,TFT,SMART 策略小.

TTL 对传递成功率的影响如图 10 所示,可以看出 5 种路由策略的传递成功率都随着 *TTL* 的增大而增大,这是因为 *TTL* 增大,消息可以在网络中增大存活时间,增加传递机会,从而增大传递成功率. Direct 不能即时将自身消息转发给传递时延更小的其他节点,所以传递成功率远低于其他策略;由于 Full-Cooperative 中节点完全转发消息,传递成功率最高;VCILDR 传递成功率高于 TFT 和 SMART.

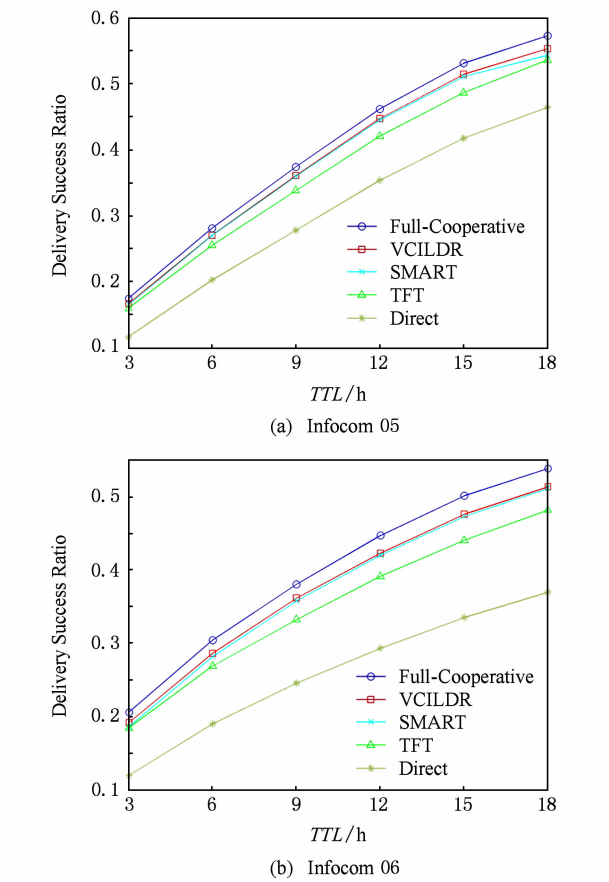


Fig. 10 Impact of varying *TTL* on delivery success ratio.

图 10 *TTL* 对传递成功率的影响

TTL 对负载率的影响如图 11 所示,由于 Direct 策略每个节点只缓存自身消息,不需要经过其他节点转发,所以负载率始终为 1;Full-Cooperative,VCILDR,SMART,TFT 策略的负载率均随着 *TTL* 的增大而增小,这是因为随着 *TTL* 的增大,虽然成功转发消息数和转发次数都在增大,但转发次数的增大速率小于成功转发消息数的增大速率. 由于 Full-Cooperative 策略中,节点完全转发其他节点消息,所以负载率最大;VCILDR 策略中,由于节点有效转发消息,所以负载率略高于 TFT,SMART 策略,但低于 Full-Cooperative 策略.

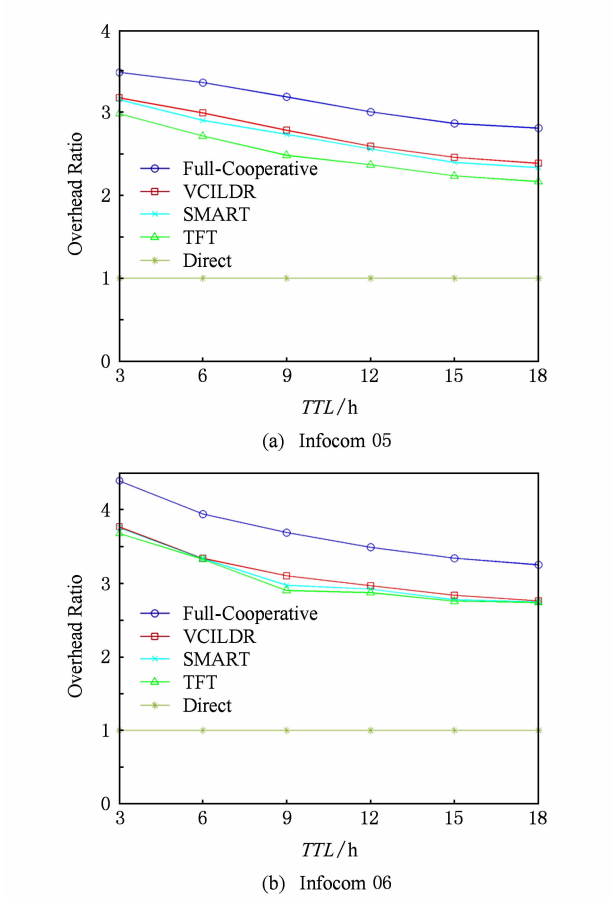


Fig. 11 Impact of varying *TTL* on overhead ratio.

图 11 *TTL* 对负载率的影响

3) 数据传输速度对性能的影响

数据传输速度对消息传递时延的影响如图 12 所示,可以看出随着数据传输速度的增大,消息平均传递时延不断减小. 因为随着数据传输速度的增大,节点相遇时的数据传输机会也增大,消息可以被快速的转发给传递时延小的节点,所以会减小消息成功传递的平均时延,即使消息传递成功率的增大会导致部分时延大的消息被成功传递. Full-Cooperative 策略由于节点间完全合作转发,使消息快速转发给下一跳节点,所以消息传递时延最小;VCILDR 策略的消息传递时延小于 TFT 和 SMART 策略,大于 Full-Cooperative 策略.

数据传输速度对消息传递成功率的影响如图 13 所示,可以看出随着数据传输速度的增大,消息传递成功率不断增大. 因为传输速度越大,节点相遇

时能够转发更多的消息,所以会提高消息传递成功率.Direct 策略由于依靠自身转发消息,传输速度达到一定程度后对其影响不大,其消息传递成功率最小;Full-Cooperative 策略由于节点间完全合作转发,所以消息传递成功率最大;VCILDR 策略的消息传递成功率高于 TFT 和 SMART 策略,低于 Full-Cooperative 策略.

数据传输速度对负载率的影响如图 14 所示,可

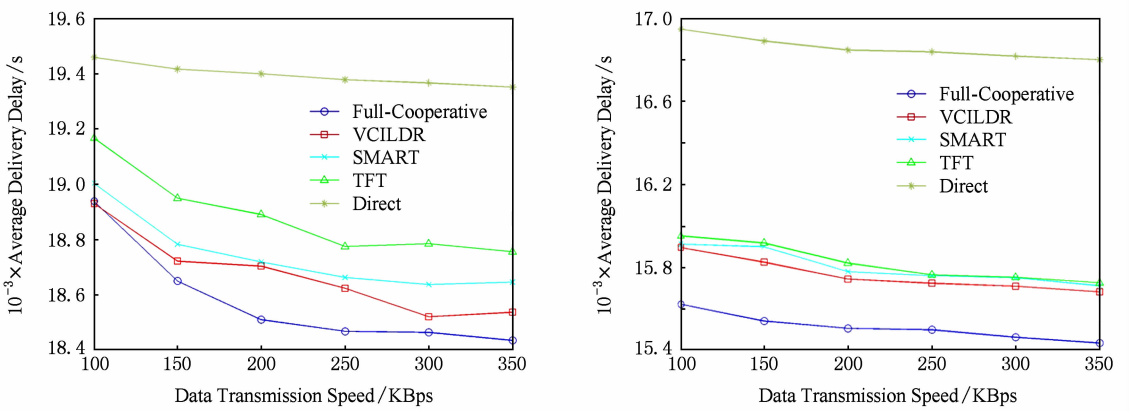


Fig. 12 Impact of varying transmission speed on average delivery delay.

图 12 数据传输速度对平均传递时延的影响

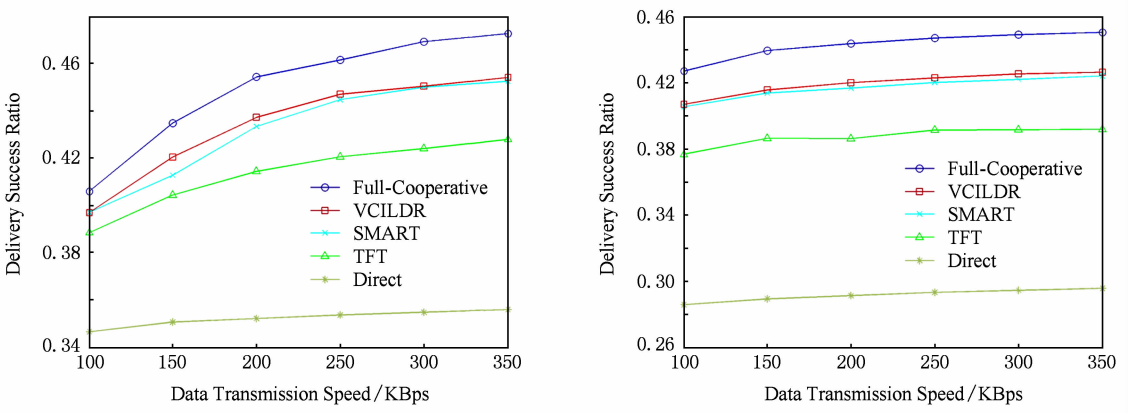


Fig. 13 Impact of varying transmission speed on delivery success ratio.

图 13 数据传输速度对传递成功率的影响

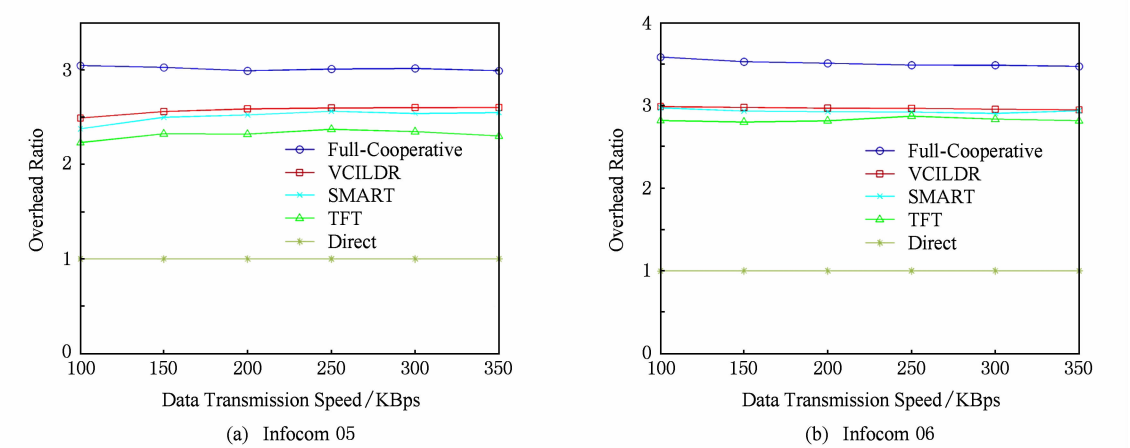


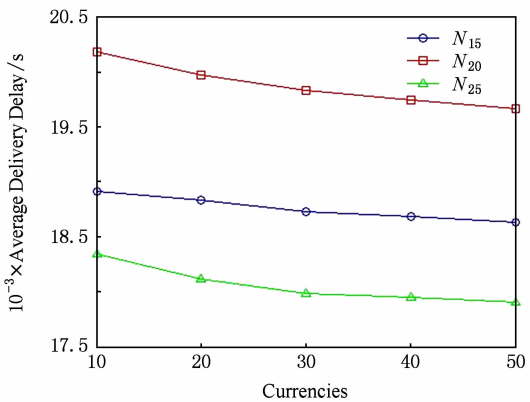
Fig. 14 Impact of varying transmission speed on overhead ratio.

图 14 数据传输速度对负载率的影响

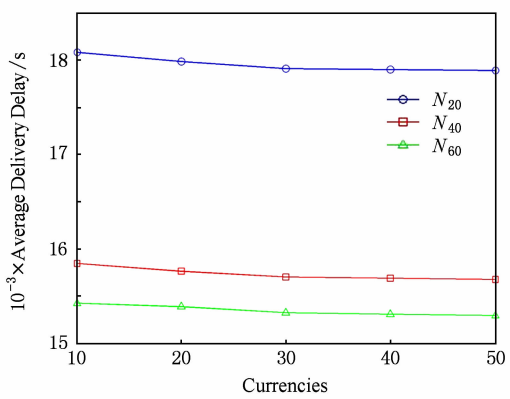
以看出 Direct 策略负载率始终为 1. 对于其他路由策略,一方面数据传输速度增大,会增大消息传输机会从而增大转发次数;另一方面也会增大成功传递消息数,总体上负载率随着数据传输速度的增大而变化不大. Full-Cooperative 策略负载率最大;VCILDR 策略由于有效的消息转发,负载率高于 TFT 和 SMART 策略.

4) 货币值对性能的影响

为验证货币值对性能的影响,在 Infocom 05 和 Infocom 06 数据集上任意选取节点 N_{15},N_{20},N_{25} 和 N_{20},N_{40},N_{60},不断增加他们对每个消息所支付的货币值,而其他节点的货币值保持 10 不变. 货币值对节点传递时延的影响如图 15 所示,可以看出时延随着货币值的增大而减小,这是因为中间节点为赚取更多的货币值,会优先存储、快速转发货币值大的消息,所以货币值越大,传递时延越小.



(a) Infocom 05

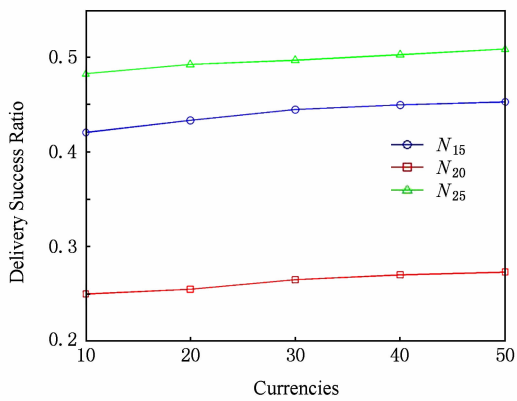


(b) Infocom 06

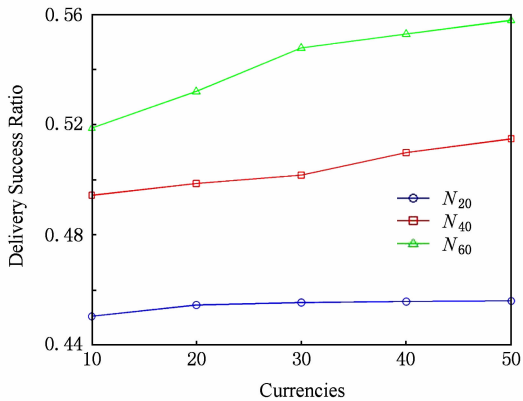
Fig. 15 Impact of varying currencies on average delivery delay.

图 15 货币值对平均传递时延的影响

货币值对传递成功率的影响如图 16 所示,可以看出货币值越大 6 个节点的消息传递成功率越大. 这是因为更大的货币值会激励中间节点快速转发这 6 个节点的消息,在缓存有限时会优先存储货币值大的消息而删除货币值小的消息,所以货币值越大传递成功率越大.



(a) Infocom 05



(b) Infocom 06

Fig. 16 Impact of varying currencies on delivery success ratio.

图 16 货币值对传递成功率的影响

5) 讨价还价次数对节点收益的影响

为验证求解子博弈完美均衡的贪婪算法中讨价还价次数对均衡结果即节点收益值的影响,在 Infocom 05 数据集上任意选取 2 组节点 N_{18},N_{26} 和 N_5,N_{23},在 Infocom 06 数据集上选取 2 组节点 N_{18},N_{48} 和 N_{20},N_{24},分别进行验证. 如图 17 所示. 可以看出当讨价还价次数为 1 时,首先出价节点(Infocom 05 中 N_5,N_{18};Infocom 06 中 N_{18},N_{20})的收益为其收益最大值. 这是因为,由求解子博弈完美均衡的算法可知,先出价节点可以按照自己的收益值大小贪婪选择交换消息,而后出价节点必然接受任何交换方案,否则其收益值为 0. 当讨价还价次数大于 1 时,后出价节点可以在某一阶段按照自己的收益贪婪选择交换消息,由于交换消息时双方收益经常是冲突的,所以导致先出价节点的收益小于讨价还价次数为 1 时的收益. 从图 17 还可以看出,

随着讨价还价次数的增加,双方节点的收益是此消彼长的过程,因为消息交换时是冲突的,在双方博弈过程中,一方节点通过贪婪算法使得自身收益变大时,会使对方节点收益变小.

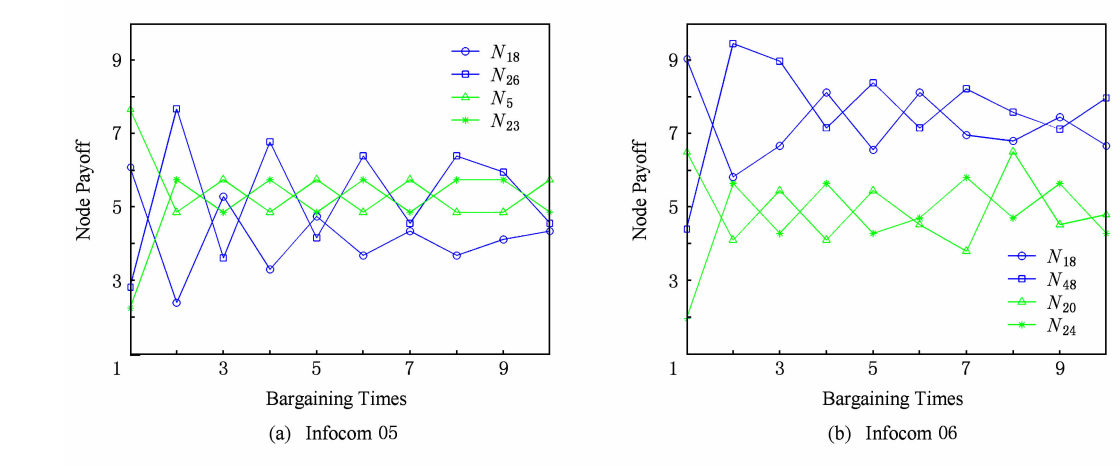


Fig. 17 Impact of varying bargaining times on node payoff.

图 17 讨价还价次数对节点收益的影响

6) 子博弈完美均衡算法对性能的影响

为验证子博弈完美均衡算法对性能的影响,将确定消息发送集合中只包含对方消息、自身可发送消息、直接互利消息的路由定义为 VCILDR1. 将 VCILDR1 和还交换可交换互利消息的 VCILDR 路由进行性能对比. 如图 18～20 所示,可以看出 VCILDR

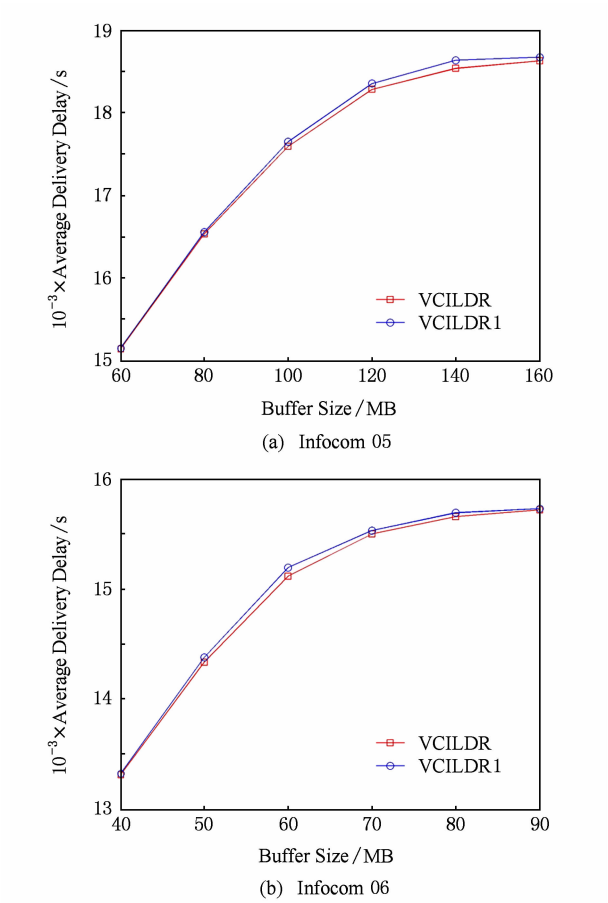


Fig. 18 Impact of subgame perfect equilibrium algorithm on average delivery delay.

图 18 子博弈完美均衡算法对平均传递时延的影响

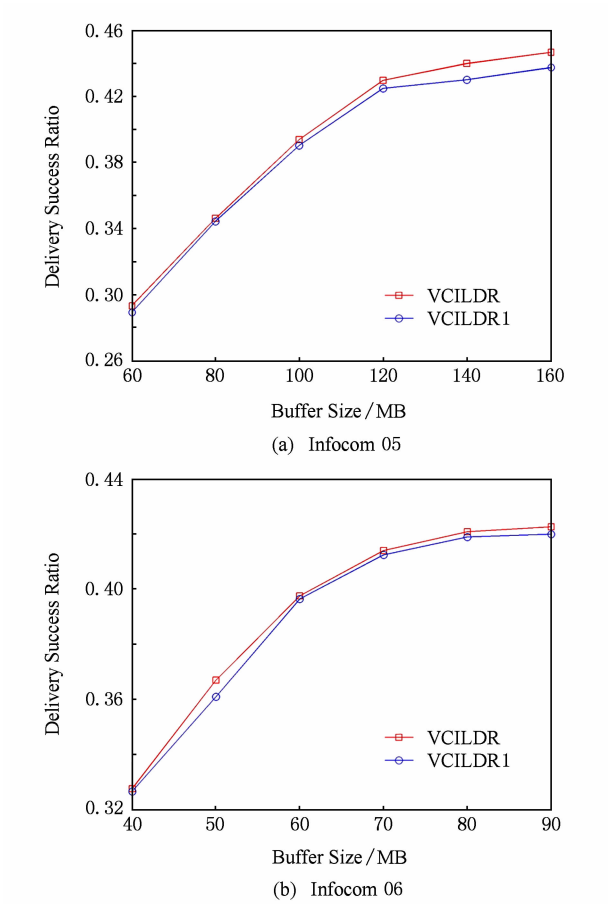
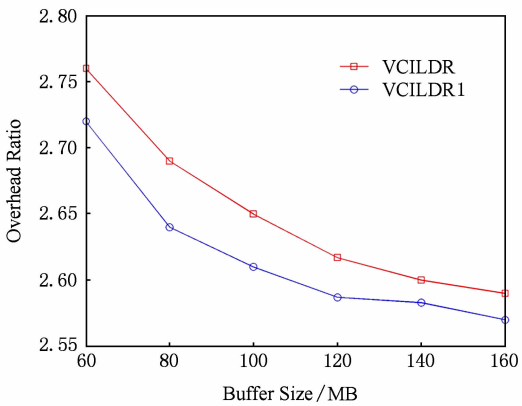


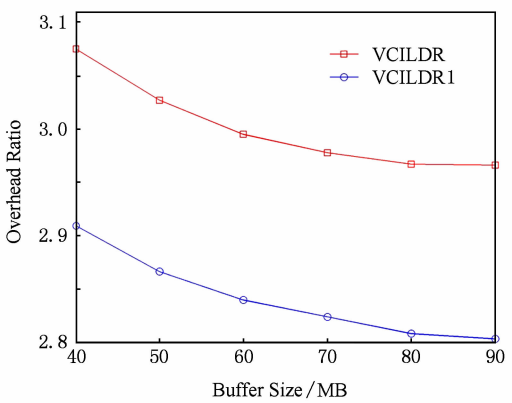
Fig. 19 Impact of subgame perfect equilibrium algorithm on delivery success ratio.

图 19 子博弈完美均衡算法对传递成功率的影响

路由策略能够获得较低的传递时延和较高的传递成功率,负载率略高于 VCILDR1. 由于 VCILDR 路由策略中,相遇节点间通过子博弈完美均衡算法求解子博弈完美均衡确定了双方的可交换互利消息,将消息交换给传递时延小的节点,增加了节点间的合作,所以获得了较低的传递时延和较高的传递成功率. 由于节点间增加了可交换互利消息的交换和转发,所以 VCILDR 路由策略的负载率略高于 VCILDR1.



(a) Infocom 05



(b) Infocom 06

Fig. 20 Impact of subgame perfect equilibrium algorithm on overhead ratio.

图 20 子博弈完美均衡算法对负载率的影响

5 结束语

本文为解决 DTNs 中存在大量自私节点而导致时延等路由性能低的问题,提出一种基于虚拟货币的激励感知低时延路由协议. 通过建立基于时延的货币支付策略和成比例货币分配策略对转发节点进行奖励,使得消息传递时延越小节点获得的货币值越大,促使自私节点为获取货币而进行合作,将直接互利消息转发给传递时延小的节点,同时和其他节点交换可交换互利消息. 为确定双方可交换互利消息,建立节点间交换可交换互利消息的轮流出价讨价还价博弈模型,并提出一种求解该模型子博弈完美均衡的贪婪算法. 实验表明,该激励感知路由能够促使节点进行合作转发,显著减小消息传递时延,同时提高消息传递成功率.

参 考 文 献

[1] Li Xiangqun, Liu Lixiang, Hu Xiaohui, et al. Delay/disruption tolerant network study [J]. Journal of Computer Research and Development, 2009, 46(8): 1270-1277 (in Chinese)
(李向群, 刘立祥, 胡晓惠, 等. 延迟/中断可容忍网络研究进展[J]. 计算机研究与发展, 2009, 46(8): 1270-1277)

[2] Spyropoulos T, Psounis K, Raghavendra C S. Efficient routing in intermittently connected mobile networks: The single-copy case [J]. IEEE/ACM Trans on Network, 2008, 16(1): 63-76

[3] Dalyd E, Haahr M. Social network analysis for information flow in disconnected delay-tolerant MANETs [J]. IEEE Trans on Mobile Computing, 2009, 8(5): 1-16

[4] Vahdat A, Becker D. Epidemic routing for partially connected ad hoc networks, CS-20006 [R]. Durham, North Carolina: Duke University, 2000

[5] Spyropoulos T, Psounis K, Raghavendra C S. Efficient routing in intermittently connected mobile networks: The multiple-copy case [J]. IEEE/ACM Trans on Network, 2008, 16(1): 77-90

[6] Lindgren A, Doria A, Schelen O. Probabilistic routing in intermittently connected networks [J]. SIGMOBILE Mobile Computing Communications Review, 2003, 7(3): 19-20

[7] Panagakis A, Vaios A, Stavrakakis I. On the effects of cooperation in DTNs [C] //Proc of COMSWARE 2007. Piscataway, NJ: IEEE, 2007: 1-6

[8] Karaliopoulos M. Assessing the vulnerability of DTN data relaying schemes to node selfishness [J]. IEEE Communications Letters, 2009, 13(12): 923-925

[9] Wu Yue, Li Jianhua, Lin Chuang. Survey of security and trust in opportunistic networks [J]. Journal of Computer Research and Development, 2013, 50(2): 278-290 (in Chinese)
(吴越, 李建华, 林闯. 机会网络中的安全与信任技术研究进展[J]. 计算机研究与发展, 2013, 50(2): 278-290)

[10] Li Qinghua, Gao Wei, Zhu Sencun, et al. A routing protocol for socially selfish delay tolerant networks [J]. Ad Hoc Networks, 2012, 10(8): 1619-1632

[11] Shevade U, Song Hanhee, Qiu Lili, et al. Incentive-aware routing in DTNs [C] //Proc of the 16th IEEE Int Conf on Network Protocols (ICNP'08). Piscataway, NJ: IEEE, 2008: 238-247

[12] Zhu Haojin, Lin Xiaodong, Lu Rongxing, et al. SMART: A secure multi-layer credit based incentive scheme for delay-tolerant networks [J]. IEEE Trans on Vehicular Technology, 2009, 58(8): 4628-4639

[13] Mei A, Stefa J. Give2Get: Forwarding in social mobile wireless networks of selfish individuals [J]. IEEE Trans on Dependable and Secure Computing, 2012, 9(4): 569-582

[14] Li Yun, Yu Jihong, You Xiaohu. An incentive protocol for opportunistic networks with resources constraint [J]. Chinese Journal of Computers, 2013, 36(5): 947-956 (in Chinese)
(李云, 于季弘, 尤肖虎. 资源受限的机会网络节点激励策略研究[J]. 计算机学报, 2013, 36(5): 947-956)

[15] Chen B B, Chan M C. MobiCent: A credit-based incentive system for disruption tolerant network [C] //Proc of IEEE INFOCOM2010. Piscataway, NJ: IEEE, 2010: 1-9

[16] Zhao Guangsong, Chen Ming. Research of incentive aware data dissemination in selfish opportunistic networks [J]. Journal on Communications, 2013, 34(2): 73-84 (in Chinese)
(赵广松, 陈鸣. 自私性机会网络中激励感知的内容分发的研究[J]. 通信学报, 2013, 34(2): 73-84)

[17] Buttyan L, Dora L, Felegyhazi M, et al. Barter trade improves message delivery in opportunistic networks [J]. Ad Hoc Networks, 2010, 8(1): 1-14

[18] Wang Yan, Chuah M C, Chen Yingying. Incentive based data sharing in delay tolerant mobile networks [J]. IEEE Trans on Wireless Communications, 2014, 13(1): 370-381

[19] Gao Wei, Li Qinghua, Zhao Bo, et al. Multicasting in delay tolerant networks: A social network perspective [C] //Proc of MobiHoc2009. New York: ACM, 2009: 299-308

[20] Krifa A, Barakat C, Spyropoulos T. Optimal buffer management policies for delay tolerant networks [C] //Proc of the 5th Annual IEEE Communications Society Conf on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'08). Piscataway, NJ: IEEE, 2008: 260-268

[21] Rubinstein A. Perfect equilibrium in a bargaining model [J]. Econometrica, 1982, 50(1): 97-109

[22] Li Junlin, Li Tianyou. Bargaining theory and its recent developments [J]. Economic Theory and Business Management, 2005, 25(3): 63-67 (in Chinese)
(李军林, 李天有. 讨价还价理论及其最近的发展[J]. 经济理论与经济管理, 2005, 25(3): 63-67)

[23] Yao Guoqing. Game Theory [M]. Beijing: Higher Education Press, 2007: 107-122 (in Chinese)
(姚国庆. 博弈论[M]. 北京: 高等教育出版社, 2007: 107-122)

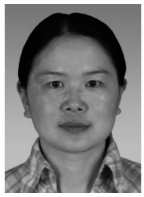
[24] Keranen A, Ott J, Karkkainen T. The ONE simulator for DTN protocol evaluation [C] //Proc of the 2nd Int Conf on Simulation Tools and Techniques. New York: ACM, 2009: 1-10

[25] Scott J, Gass R, Crowcroft J, et al. CRAWDAD Trace infocom05 [EB/OL]. (2006-01-31) [2014-11-22]. http://crawdad.cs.dartmouth.edu/cambridge/haggle

[26] Scott J, Gass R, Crowcroft J, et al. CRAWDAD Trace infocom06 [EB/OL]. (2009-05-29) [2014-11-22]. http://crawdad.cs.dartmouth.edu/cambridge/haggle

Jiang Qingfeng, born in 1983. PhD candidate at Harbin Engineering University. Lecturer at Daqing Normal University. Member of China Computer Federation. His main research interests include delay tolerant networks and social networks(qingfeng_jiang@163.com).

Men Chaoguang, born in 1963. Professor and PhD supervisor at Harbin Engineering University. Senior member of China Computer Federation. His main research interests include mobile computing and trusted computing.



Li Xiang, born in 1975. Associate professor and master supervisor at Harbin Engineering University. Member of China Computer Federation. Her main research interests include distributed computing and mobile computing (leexiang@hrbeu.edu.cn).

He Zhongzheng, born in 1986. PhD candidate at Harbin Engineering University. His main research interests include mobile computing and fault tolerant computing (hezhongzheng@hrbeu.edu.cn).