ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.4, April 2008, pp.946−955
DOI: 10.3724/SP.J.1001.2008.00946

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种直接评价节点诚信度的分布式信任机制*

彭冬生+, 林 闯, 刘卫东

(清华大学 计算机科学与技术系,北京 100084)

A Distributed Trust Mechanism Directly Evaluating Reputation of Nodes

PENG Dong-Sheng+, LIN Chuang, LIU Wei-Dong

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

+ Corresponding author: Phn: +86-10-51537923, E-mail: pds04@mails.tsinghua.edu.cn, http://www.tsinghua.edu.cn

Peng DS, Lin C, Liu WD. A distributed trust mechanism directly evaluating reputation of nodes. *Journal of Software*, 2008,19(4):946−955. http://www.jos.org.cn/1000-9825/19/946.htm

Abstract: Reputation-Based trust mechanism can efficiently solve the problems of virus flooding and malicious behaviors in P2P network. Most of the existing trust mechanisms use a single reputation value to depict the node's reliability, which cannot prevent malicious nodes from concealing dishonest selling behaviors with honest buying behaviors, and cannot separate the new node from the malicious one. This paper provides a new distributed trust mechanism, which iteratively calculates for each node a global seller reputation value and a global buyer reputation value based on transaction history, and whether a node is trustable or not can be identified from them. Comparison experiments and performance analysis between the mechanism and EigenTrust show that the mechanism can reduce the global reputation value of malicious node rapidly, restrain collusion attack, and decrease probability malicious transactions.

Key words: P2P network; distributed trust mechanism; reputation; DHT (distributed hash table); collusion attack

摘 要: 基于信誉的信任机制能够有效解决 P2P 网络中病毒泛滥和欺诈行为等问题.现有信任机制大多采用单个信誉值描述节点的诚信度,不能防止恶意节点用诚信买行为掩盖恶意卖行为;而且从信誉值上无法区分初始节点和恶意节点.提出一种新的分布式信任机制,基于交易历史,通过迭代求解,为每个节点计算全局买信誉值和卖信誉值,根据信誉值便能判断节点的善恶.仿真实验对比和性能分析表明,与 EigenTrust 算法相比,该算法能够迅速降低恶意节点的全局信誉值,抑制合谋攻击,降低恶意交易概率.

关键词: P2P 网络;分布式信任机制;信誉;分布式哈希表;合谋攻击

中图法分类号: TP393 **文献标识码**: A

P2P技术因其自组织、开放性和匿名等特点成为新的网络应用热点.但是,随着P2P技术的广泛应用,原本不被重视的安全性问题逐渐成为阻碍其发展的主要因素.例如:在P2P文件共享应用中,共享带病毒文件容易造成病毒泛滥;在SETI@HOME项目中,用户通过破解客户端软件快速返回貌似合理的结果使自己看起来做了更多工作,从而骗取更高的贡献排名[1];音乐影像公司为了保护版权共享大量伪造文件,使用户难以找到真实文

* Supported by the National Natural Science Foundation of China under Grant No.90412012 (国家自然科学基金)
Received 2006-07-01; Accepted 2006-12-27

件[2].因此,仅靠节点自律远远不够,必须提供有效的机制消除这些负面影响,提高P2P应用的可用性.

一种解决办法是引入基于信誉值的信任机制,它根据每个节点的网络行为动态计算其局部或全局信誉值,通过信誉值的高低判断节点的诚信度.因此,信任机制能够为节点选择交易对象提供参考依据,并激励节点诚信交易.集中式信任机制因为需要中心服务器而不适用于 P2P 应用;而分布式信任机制设计要解决如下两个问题:1) 如何衡量和计算信誉值;2) 如何存取和管理信誉值.

在多数现有信任机制中,信誉值只能用于比较节点间的相对诚信度,无法直接从信誉值上判断节点是否可信.事实上,很多应用需要直接从信誉值上判断节点的善恶,例如,在电子商务应用中,用户往往会选择信誉值高于一定阈值的节点作为交易对象.另外,用户在网络中的交易行为分为买方和卖方,因此,仅用一个信誉值来衡量节点无法区分同一节点不同角色时的诚信度,因为恶意节点可以用诚信买行为来掩盖恶意卖行为,或者反之.

本文通过对现有信任机制的分析,提出一种新的分布式信任机制,根据节点间的交易历史,动态地为每个节点分别计算出其买全局信誉值和卖全局信誉值,体现节点在买和卖角色时的诚信度,用户根据节点的信誉值便能判断节点的善恶,从而减少恶意节点被选为交易对象的概率.

本文第 1 节对目前信任机制的研究现状进行综述.第 2 节详细描述本文机制.第 3 节给出实验验证和结果.第 4 节进行性能分析.第 5 节给出结论.

1 信任机制的相关研究

基于信誉值的信任机制主要分为集中式和分布式信任机制两类.

集中式信任机制最常见的是电子交易网站(如 eBay,Alibaba)采用的信任机制.交易双方对每次交易给出评价,1,0,–1 分别代表好评、中评和差评,用户信息和交易记录由网站集中管理和统计,供用户参考.这类信任机制简单而有效,前提是用户信任网站.

PageRank算法[3]是最典型的集中式信任机制.该算法用来为每个网页计算出一个等级评估值.它根据网页间的链接关系形成一个关联矩阵,等效于Markov链中的状态转移矩阵,该等效Markov链的稳态概率就是所有网页的等级评估值的分布.

基于信誉的分布式信任机制的主要思想是,网络中每个节点根据本地信息(交易历史或者是信任关系),计算出它对其他节点的本地信誉值c_{ij},得到初始信任矩阵C,然后通过一定的算法扩大节点的信任范围,或者形成节点的全局信誉值.此类研究有文献[4–8].其中:

1) 在eTrust信任机制[7]中,本地信誉值c_{ij}根据交易记录采用模糊逻辑推理的方法产生,全局信誉值则是通过加权平均的方式生成:$T_i=\dfrac{\sum_j w_j\times c_{ji}}{\sum_j w_j}$,加权因子$w_j$综合若干因素采用模糊逻辑推理产生.

2) EigenTrust信任机制[5]的原理和PageRank算法完全一致.在EigenTrust中,本地信誉值的计算为$c_{ij}=\dfrac{\mathrm{Max}(s(i,j)-uns(i,j),0)}{\sum_j \mathrm{Max}(s(i,j)-uns(i,j),0)}$,其中,$s(i,j)$为节点$i$和$j$之间诚信交易的次数,$uns(i,j)$为节点$i$和$j$之间不诚信交易的次数.全局信誉值通过关系式:$t_i^{(k+1)}=(1-\alpha)\left(\sum_j t_j^k\times c_{ji}\right)+\alpha\times p_i$迭代求解.其中,$p_i$为节点$i$的信任基数,$\alpha\in[0,1]$为节点对假定信任基数的信任程度.

EigenTrust 机制的缺点是全局信誉值只是相对值,无法直接从全局信誉值上判断节点是否可信.

信任传递不适用于不信任的情况,如 a 不信任 b,b 不信任 c,a 和 c 的信任关系并不确定.文献[6]提出了一种同时评估节点信任度和不信任度的机制.通过使用两个矩阵 T 和 D 分别描述信任关系和不信任关系.实验表明,采用 One-Step Distrust 计算模型,也就是在信任传递链中,只在最后一次传递中加入对不信任评估(T-D)时,效果最理想.

文献[9]提出了一种机制,用户在与未知节点交易之前,通过主动探测机制了解未知节点的诚信度,是分布式

信任机制的一种新的研究思路.

2 直接评价节点诚信度的分布式信任机制

分布式信任机制由本地信誉评估和全局信誉计算与存取管理两部分组成,相互关系如图 1 所示.节点收集其他节点的信息并进行信誉评估;系统综合所有信誉评估,计算各节点的全局信誉值并进行存取管理以保证其安全和有效性;用户结合本地信誉评估和全局信誉值,选择交易对象实施交易.

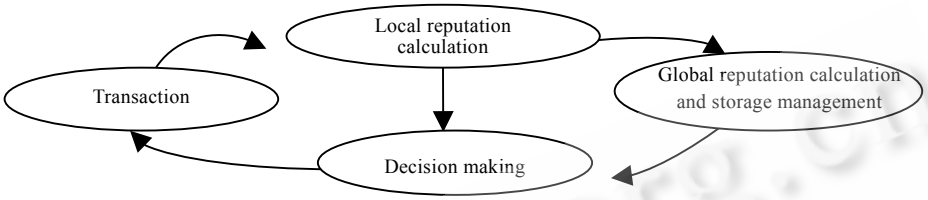

Fig.1 Components of the distributed trust mechanism

图 1 分布式信任机制组成

本地信誉评估和全局信誉计算与存取管理是分布式信任机制的核心,它们为交易决策提供依据,并根据交易结果动态计算节点的本地和全局信誉值.

以下分别从信誉定义、本地信誉评估计算模型、全局信誉值计算模型、全局信誉值分布式计算和存储等方面对本文机制进行详细描述.

2.1 定 义

信誉用于评价网络中节点的诚信度.信誉的数值表示有多种方法,文献[6]算法用 1 和–1 表示信任和不信任,并采用两个矩阵分别评价节点的可信和不可信;EigenTrust[5]算法中的信誉值无法判断节点的善恶,只能比较节点间的相对诚信度.本文采用一种直接评价节点诚信度的信誉定义.为便于描述,给出如下定义:

定义 1. 交易(transaction).

分布式网络中两个节点之间发生一次交互行为称为一次交易,如文件共享网络中的一次下载、电子商务中的一次买卖等.交易发起方为买方,另一方为卖方.

定义 2. 节点的本地信誉评估值(b_{ij},s_{ij}).

节点根据交易历史评价与之发生过交易的节点的诚信度.b_{ij}表示节点i作为买方对卖方节点j的信誉评估,s_{ij}表示节点i作为卖方对买方节点j的信誉评估.b_{ij}和s_{ij}的取值区间为[0,1],初始值为 0.5;高于 0.5 时,认为节点是诚信的,越接近 1,诚信度越高;低于 0.5 时,认为节点是恶意的,越接近 0,恶意度越高.

定义 3. 节点的全局信誉值(Tb_i,Ts_i).

用于从全网角度来评价节点在买行为和卖行为时的诚信度,称为节点的买全局信誉值和卖全局信誉值.Tb_i,Ts_i的取值和意义与b_{ij}和s_{ij}相同.

恶意节点的信誉值低于初始信誉值 0.5 可能引起 white washing 攻击:恶意节点通过更换 ID 以新节点身份加入网络,从而隐瞒其历史.防止这种攻击的方法是加强新用户认证管理,这里不做详细讨论.

2.2 本地信誉评估值(b_{ij},s_{ij})计算模型

每个节点基于交易历史或者社会关系形成对其他节点的信誉评估.由于社会网络中的关系很难量化,本文基于交易历史进行诚信度评估.一种简单而有效的量化方式是:节点每次交易后给出满意和不满意评价,本地信誉评估值用与该节点交易记录中满意的次数所占比例来衡量.因此,b_{ij}和s_{ij}的计算模型为

$$b_{ij}=\frac{N_{bh}(i,j)}{N_{bh}(i,j)+N_{bm}(i,j)\times N_{pnsh}},$$
$$s_{ij}=\frac{N_{sh}(i,j)}{N_{sh}(i,j)+N_{sm}(i,j)\times N_{pnsh}} \quad (1)$$

其中,$N_{bh}(i,j)$,$N_{bm}(i,j)$分别表示节点i作为买方节点与j发生诚信和恶意交易的次数;

$N_{sh}(i,j)$,$N_{sm}(i,j)$分别表示节点i作为卖方节点和j发生诚信和恶意交易的次数;

N_{pnsh}表示对恶意发生恶意交易时的惩罚系数.

可以看出,式(1)有以下性质:

1) 直接评价节点的诚信度,诚信交易越多,信誉值越接近 1;恶意交易越多,信誉值越靠近 0;

2) 通过引入惩罚系数N_{pnsh},使信誉值的下降速率比上升速率快,体现出对恶意交易的惩罚.

2.3 全局信誉值(Tb_i,Ts_i)计算模型

全局信誉值是交易网络对于每个节点的综合评估.文献[4]通过若干次信任传递计算扩展本地信任评估,仍是一种局部信誉评估;EigenTrust[5]算法计算的信誉值是相对值.本文提出的本地信誉评估是一种直接评估机制,直观的全局信誉值计算方法是取所有对该节点的信誉评估值的平均值,但需考虑下列因素:

1) 节点i的买(卖)全局信誉值Tb_i(Ts_i),应该由与其有过交易的卖(买)方节点共同评估;

2) 区别对待不同诚信度节点评估意见,高信任度节点的评估意见比低信任度节点更重要;

3) 节点之间交易次数越多,它们之间的评估意见更可信;

4) 全局信誉值是一个逐渐积累的过程,持续诚信交易的积累才能获得高信誉值.

因此,节点的全局信誉值Tb_i和Ts_i应该是所有与之有过交易节点的全局信誉值、交易次数以及信誉评估值的函数.本文采用加权平均的方法来表述这种关系:

$$Tb_i=\frac{\sum_{j\in Vs_i}Ts_j^{N_w}\times\left(1-e^{-\frac{N_s(j,i)}{5}}\right)\times s_{ji}}{\sum_{j\in Vs_i}Ts_j^{N_w}\times\left(1-e^{-\frac{N_s(j,i)}{5}}\right)},$$
$$Ts_i=\frac{\sum_{j\in Vb_i}Tb_j^{N_w}\times\left(1-e^{-\frac{N_b(j,i)}{5}}\right)\times b_{ji}}{\sum_{j\in Vb_i}Tb_j^{N_w}\times\left(1-e^{-\frac{N_b(j,i)}{5}}\right)} \tag{2}$$

其中,$N_b(j,i)=N_{bh}(j,i)+N_{bm}(j,i)$,$N_s(j,i)=N_{sh}(j,i)+N_{sm}(j,i)$;

N_w表示节点信誉值在计算中的加权系数;

Vb_i表示与节点i有过交易并且从节点i"购买"的节点的集合;

Vs_i表示与节点i有过交易并且节点i向其"购买"的节点的集合.

这种加权平均方法的优点是:

① 通过平均能够综合体现所有节点的观点,并保持全局信誉值的语义不变;

② 在两个加权函数中,一个是交易节点全局信誉值的N_w次方,体现高信誉值节点对评估的重要性;另一个是 1−exp(−$N(j,i)$/5),它随$N(j,i)$负指数增长的特点很适合用于强调节点交易次数的重要性.

但是,式(2)不能满足上述因素 4)的要求,因为一个初始节点只需进行一次诚信交易,其信誉值就上升到 1,不能体现公平性.针对这种情况,对上述模型做如下改进:

设$||V||$表示集合V中的节点个数,$N_{\min\_s}$,$N_{\min_b}$分别表示参与评价一个节点卖全局信誉值和买全局信誉值的最少节点数.构建一种虚拟节点,这种节点v的买信誉值Tb_v=0.5,卖信誉值Ts_v=0.5,对于其他节点的信誉评估值,b_{vi}=0.5,s_{vi}=0.5,$N_b(v,i)$=+∞,$Ns(v,i)$=+∞.式(2)中,

1) 如果$||Vb_i||<N_{\min_s}$,增加($N_{\min\_s}-||Vb_i||$)个虚拟节点补足,使集合Vb_i中的节点数不低于$N_{\min_s}$个;

2) 如果$||Vs_i||<N_{\min_b}$,增加($N_{\min\_b}-||Vs_i||$)个虚拟节点补足,使集合Vs_i中的节点数不低于$N_{\min_b}$个.

通过规定最少参与评估节点数$N_{\min}$,弱化少数节点对某个节点评估的决定作用,能够有效防止集体作弊行为.$N_{\min}$的取值视具体应用而定,实验中$N_{\min_s}$取网络总节点数的 10%,$N_{\min_b}$取网络中卖方节点数的 10%.

2.4 全局信誉值的简化计算模型

式(2)是非线性方程组,其求解一般通过迭代法进行.这种求解方式需多次迭代,因而增加了通信和计算开销,而且很难证明此非线性方程组迭代求解的收敛性.我们采用一种简化算法来逼近这个方程组的解.具体为

$$Tb_i(k+1)=\frac{\sum_{j\in Vsi}(Ts_j(k))^{Nw}\times\left(1-e^{-\frac{Ns(j,i)}{5}}\right)\times s_{ji}}{\sum_{j\in Vsi}(Ts_j(k))^{Nw}\times\left(1-e^{-\frac{Ns(j,i)}{5}}\right)},$$

$$Ts_i(k+1)=\frac{\sum_{j\in Vbi}(Tb_j(k))^{Nw}\times\left(1-e^{-\frac{Nb(j,i)}{5}}\right)\times b_{ji}}{\sum_{j\in Vbi}(Tb_j(k))^{Nw}\times\left(1-e^{-\frac{Nb(j,i)}{5}}\right)} \quad (3)$$

1) 全局信誉值的计算在等间隔的时间片起始处进行;

2) 第 K+1 轮买全局信誉值(卖全局信誉值)的计算以第 K 轮卖全局信誉值(买全局信誉值)为基础;

3) 节点的初始买、卖全局信誉值:$Tb_i(0)=0.5$,$Ts_i(0)=0.5$.

与 EigenTrust 算法每轮求解都要进行多次迭代相比,本文算法每轮只需进行一次通信和计算,因此可以缩小计算间隔时间,以便及时反映节点本地信誉评估值的动态变化.

2.5 节点信誉值的分布式计算和管理

全局信誉值的分布式计算和管理需要考虑安全性和鲁棒性两个因素,防止节点破坏或窜改信誉值,并保证在动态的网络环境下不影响信誉值的计算和访问.采用 DHT(distributed hash table)结构化网络能够有效解决安全性和鲁棒性问题.DHT 网络在全局范围内分配和管理数据,能够防止在节点离开或失效时丢失数据.其存储内容和存储位置的确定关系、高效的查找和存取功能、较少的计算和通信开销以及匿名性等特点,使得它适用于作为全局信誉值计算和存储平台.本文和 EigenTrust 算法都采用这种机制.

2.5.1 网络构建和计算管理节点选择

采用DHT机制(如Chord[10])构建一个结构化网络,并假定认证机制能够保证每个节点i在网络中有全局唯一的名称N_i.对N_i进行hash运算生成$H(N_i)$,取值范围和Chord网络中节点ID范围一致.通过Chord算法找到ID等于$H(N_i)$的节点j,节点j即被选为节点i的全局信誉值计算和管理节点C_i.若考虑网络动态性和单结点易被攻破等问题,可采用不同的hash函数得到多个计算管理节点,全局信誉值按照多数原则决定.

2.5.2 数据验证

正常情况下,假如买方节点i和卖方节点j进行一次交易,如果是诚信交易,双方都满意,$N_{bh}(i,j)$和$N_{sh}(j,i)$都加1;如果是恶意交易,双方都不会增加对方的信誉,评价结果是$N_{bm}(i,j)$和$N_{sm}(j,i)$都加 1.因此,应该有等式成立:

$$N_{bh}(i,j)=N_{sh}(j,i),N_{bm}(i,j)=N_{sm}(j,i),N_{sh}(i,j)=N_{bh}(j,i),N_{sm}(i,j)=N_{bm}(j,i),$$

从而有$s_{ij}=b_{ji}$,$b_{ij}=s_{ji}$.然而,恶意节点可能会通过故意降低对其他节点的信誉评估来诋毁其他节点.节点i在发生诚信交易后仍将$N_{bm}(i,j)$加 1,试图使节点j的Ts_j降低.为了保证公平性,可以利用分布式机制的优势,节点j的信誉值计算节点C_j收集Tb_i,Ts_i,$N_b(i,j)$,$N_s(i,j)$等信息,判断上述等式是否成立,否则采用如下策略修正:

1) 当$b_{ij}<s_{ji}$时,C_j改正$s_{ji}=b_{ij}$;当$b_{ij}>s_{ji}$时,C_j不做处理;

2) 当$N_b(i,j)>N_s(j,i)$时,若$s_{ji}<0.5$,C_j改正$N_s(j,i)=N_b(i,j)$;

3) 当$N_b(i,j)<N_s(j,i)$时,若$s_{ji}>0.5$,C_j改正$N_s(j,i)=\max(N_b(i,j),1)$.

当节点 j 作为买方节点时,按同样方式处理.

2.5.3 信誉值计算

节点i的信誉值计算节点C_i进行的第k轮操作包括:

1) 接收节点i发送过来的信息包括:b_{ij},$N_b(i,j)$,s_{ij},$N_s(i,j)$;

2) 将节点i的相关信息$Tb_i(k-1),b_{ij},N_b(i,j),Ts_i(k-1),s_{ij},N_s(i,j)$发送到节点$j$的计算节点$C_j$;

3) 接收其他计算节点发送过来的$Tb_j(k-1),b_{ji},N_b(j,i),Ts_j(k-1),s_{ji},N_s(j,i)$;

4) 依照第 2.5.2 节验证数据一致性并作相应的处理;

5) 按式(3)重新计算买全局信誉值$Tb_i(k)$和卖全局信誉值$Ts_i(k)$.

3 仿真与结果

3.1 实验描述

3.1.1 网络拓扑

为了验证算法的有效性,本文采用网络拓扑生成软件BRITE[11]构建模拟网络进行仿真实验.模拟网络有500 个节点,节点连接呈Power-Law分布.假设 20%的节点同时具有买卖功能,它们在网络拓扑中的位置随机选择,其他 400 个节点仅具有买功能.

3.1.2 交易模型

买方节点交易前先根据拓扑找到 3 跳之内所有卖方节点,为了模拟交易内容和节点离线等随机性,随机选择其中最多3个作为满足要求的卖方节点.决策模型为:如果买方对卖方的本地信誉评估值低于0.5,则排除此节点;剩下的节点按卖全局信誉值排序,最高的节点选为交易对象,若没有满足要求的节点,交易终止.

实验假设平均每节点发起 100 次交易请求,全网总共 50 000 次.节点交易次数呈 Power-Law 分布.每 800 次交易计算一次全局信誉值.往往网络生成初期节点都诚信交易,因此从 10 000 次交易后开始加入恶意交易.

3.1.3 恶意交易模型

对恶意交易的仿真,本文只假设卖方节点实施恶意交易,如 P2P 文件共享应用中共享垃圾或病毒文件、电子商务中的商业欺诈等.因此,仅在卖方节点交易时加入恶意交易进行仿真.算法的有效性主要采用恶意节点成功实施恶意交易的次数占总交易数的百分比来评估.

当一个恶意节点被选择作为卖方时,决定恶意是否交易有两种方式:1) 概率选择,以概率 m 随机决定是否实施恶意交易;2) 聪明选择,对买全局信誉值最高的比例 c 的节点进行诚信交易,否则实施恶意交易.

恶意节点可能合谋来获取高信誉值,合谋节点相互给对方最高的信誉评估,给其他节点最低的信誉评估.因此我们根据这两种情况建立 4 种恶意交易模型:

	Random selection (RS)	Clever selection (CS)
Independent attack	IR (independent attack with RS) model	IC (independent attack with CS) model
Joint attack (collusion)	JR (joint attack with RS) model	JC (joint attack with CS) model

3.1.4 算法比较

为了比较算法效率,本文编程实现了EigenTrust算法.出于公平,我们对EigenTrust算法进行了改进:将$s_{ij}=sat(i,j)-unsat(i,j)$改为$s_{ij}=sat(i,j)-N_{push}\times unsat(i,j)$,并分别计算节点的买和卖全局信誉值.EigenTrust算法每2 000 次交易计算 1 次全局信誉值.实验中,$N_{push}=9$,$N_{\min\_s}=50$,$N_{\min\_b}=10$.

3.2 实验结果

3.2.1 全局信誉值对比

图 2 中分别选取两个具有代表性的卖方节点将其卖全局信誉值的变化过程描绘出来.一个是诚信卖方节点,另一个是 IR 模型,概率 m=1 的恶意节点.可以看出,本文算法中,恶意节点的卖全局信誉值降低至 0.3,低于初始信誉值,用户很容易区分恶意节点和未交易节点;EigenTrust 算法结果中,尽管恶意节点的信誉值低于诚信节点的信誉值,但根据信誉值无法区分恶意节点和新节点或交易少的诚信节点.

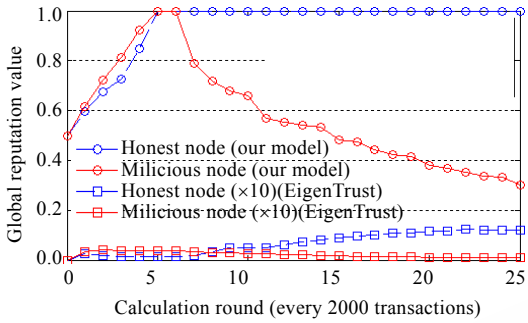

Fig.2 Comparison of global reputation values

图 2 全局信誉值对比

3.2.2 恶意交易模型仿真结果

以下是本文算法在N_w=0,1,2 这 3 种情况下与EigenTrust算法的对比实验结果.每个实验分别选取两个比例值,在恶意节点比例从 10%~50%的 5 种情况下进行对比.所有数据都是多次实验数据的平均值.

(1) IR 模型

在 IR 模型实验中,我们分别选择概率 m=0.5 和 m=1.0 两种情况进行对比,即恶意节点每次以 1/2 的概率实施恶意交易和恶意节点每次都实施恶意交易两种情况.

实验结果表明:1) 两种算法都能较好地抑制恶意交易发生率,即使在有一半卖方节点都是恶意节点时,仍能将恶意交易发生比例控制在 20%以内;2) 恶意交易发生的比例都随着恶意卖方节点的增加而呈线性增加;3) 本文算法效果相对较好,在 50%卖方节点恶意时相比EigenTrust算法有近 5%的降幅;4) 本文算法随着N_w的增加效果也越好,但是效果不明显.如图 3 所示.

(2) IC 模型

在 IC 模型实验里,我们分别选择 c=0.5 和 c=0.1 两种情况进行对比,即恶意节点仅在买方节点属于信誉值最高的 50%或 10%的节点之内时进行诚信交易,而对剩下 50%或 90%的买方节点实施恶意交易.从图 4 可以得出 IC 模型和 IR 模型相同的结论.

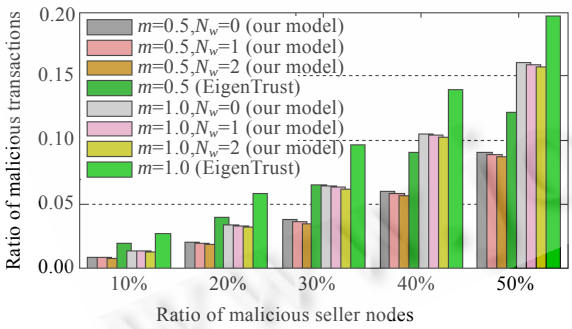

Fig.3 Simulation result of IR model

图 3 IR 模型仿真结果

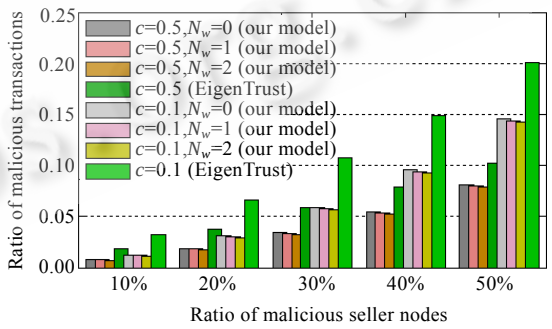

Fig.4 Simulation result of IC model

图 4 IC 模型仿真结果

(3) JR 模型

在JR模型中,由图 5 可以发现:1) EigenTrust算法中恶意交易比例与恶意节点数量呈线性关系,本文算法当N_w=0 时,恶意交易比例与恶意节点比例呈线性关系,而当N_w=1 和 2 时,则呈指数上升趋势;2) m=1.0 相比m=0.5 上升速率更快,即当恶意节点实施恶意交易概率越大时,随着恶意节点的增多,成功实施恶意交易比例的增加速率更快; 3) EigenTrust算法在 50%的恶意卖方节点比例时仍能限制恶意交易比例在 30%以下,而本文算法中恶意交易比例随着N_w和恶意节点比例的增加而增大.

(4) JC 模型

从图 6 中可以看出,JC 模型的仿真结果与 JR 模型基本一致,只是相对而言,EigenTrust 算法对 JC 模型比 JR 模型性能略低些.本模型中,c=0.1 是指恶意节点对信誉值排名后 90%的节点实施恶意交易,而 c=0.5 是指对信誉值排名后 50%的节点实施恶意交易.

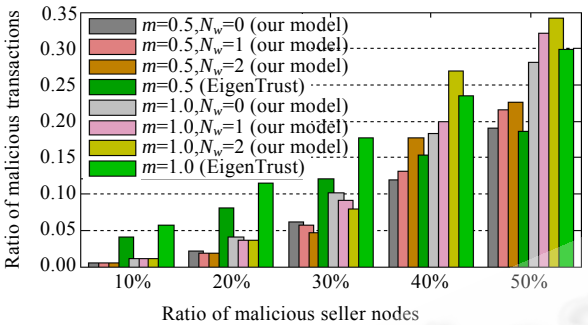

Fig.5 Simulation result of JR model

图 5 JR 模型仿真结果

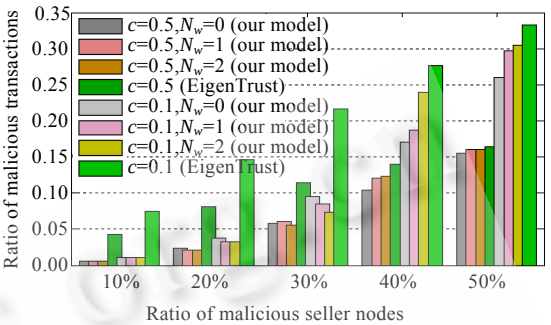

Fig.6 Simulation result of JC model

图 6 JC 模型仿真结果

4 对比分析

4.1 性能对比

从实验结果可以看出,本文算法多数情况下都不同程度地优于 EigenTrust 算法,只是在恶意节点比例高于 40%且恶意节点合谋时,效果不及 EigenTrust 算法.

用户基于全局信誉值选择信誉好的节点作为交易对象.因此,节点误选恶意节点作为交易对象的概率(P_e)可以作为评价信任机制的重要指标.在EigenTrust算法中,节点初始信誉值为 0,诚信节点和恶意节点的信誉值分布区间都是[0,1].在本文机制中,节点初始信誉值为 0.5,信任机制正常情况下使诚信节点信誉值分布在[0.5,1],而恶意节点信誉值是[0,1],因此能够有效降低P_e,附录也给出了相关证明.用户也可以选择不与信誉值低于一定阈值的节点交易,使P_e进一步降低,因此,多数情况下本文机制更优.

在 JR 模型且恶意卖方节点比例高于 40%时,体现出信誉基数的控制作用,此时,EigenTrust 算法效果更好.但在实际网络应用中,恶意节点比例高于 40%且合谋的可能性很小.我们针对合谋模型的结果作定性分析:

(1) 本文算法

网络形成初期,各节点诚信交易,b_{ji}=1,因此,节点的卖信誉值 $Ts_i=\dfrac{\sum(w_j T_{b_j}^{N_w}\times b_{ji})}{\sum w_j T_{b_j}^{N_w}}=1$;随着恶意节点形成合谋,诚信节点和恶意节点具有不同的卖信誉值.用Ths_i和Tms_i表示诚信节点和恶意节点的卖全局信誉值;S_h和S_m表示诚信卖方节点集合和恶意卖方节点集合,则有

$$Ths_i=\frac{\sum_{j\notin Sm}(w_j T_{b_j}^{N_w}\times b_{ji})}{\sum_{j\notin Sm} w_j T_{b_j}^{N_w}+\sum_{j\in Sm} T_{b_j}^{N_w}},\quad Tms_i=\frac{\sum_{j\notin Sh,\,j\notin Sm}(w_j T_{b_j}^{N_w}\times b_{ji})+\sum_{j\in Sm} T_{b_j}^{N_w}}{\sum_{j\notin Sh,\,j\notin Sm} w_j T_{b_j}^{N_w}+\sum_{j\in Sh} T_{b_j}^{N_w}+\sum_{j\in Sm} T_{b_j}^{N_w}}.$$

① 当恶意卖方节点比例较低时,Tms_i相比Ths_i优势不明显.恶意交易时,遭遇恶意交易的买方节点降低b_{ji}值,而其信誉值Tb_j因恶意节点比例低不会明显下降,导致Tms_i下降,这种情况下,恶意交易发生越多,Tms_i下降得越快,结果使恶意节点的卖信誉值低于诚信节点的卖信誉值;

② 当恶意卖方节点比例较高(40%)时,Tms_i比Ths_i优势明显,恶意交易时,买方节点降低b_{ji}值,但其信誉值Tb_j因恶意节点比例高而下降明显,导致诚信节点卖信誉值Ths_i下降,又使Tms_i值升高,而且N_w越大,此种状况越明显.如此循环,导致恶意节点获得比诚信节点更高的信誉值,恶意交易发生越多,这种情况恶化得越快;

③ 当N_w=0 时,信誉值是参与交易的节点本地评估值的平均,因此,一般情况下,性能比N_w=1 和N_w=2 时要差;

在合谋恶意卖方节点比例较高时,N_w=1 和N_w=2 时诚信节点信誉值遭受压制,此时,简单平均方式却能消除该缺点.因此,在恶意节点比例高(>40%)时,N_w=0 时的算法性能比N_w=1 和N_w=2 时要好,甚至优于EigenTrust算法.

(2) EigenTrust 算法

相比之下,EigenTrust 算法在高于 40%的恶意节点比例时仍有较好的效果,这是因为算法的一个改进:假定网络中存在若干固定的诚信节点,它们的全局信誉值始终高于设定的最低值.如果没有这个改进,恶意节点合谋时,由于恶意节点信任互指,EigenTrust 的信任矩阵等价于 Markov 过程的状态转移矩阵,此时,Markov 过程并不是各态历经的,最终全局信誉值都收敛到恶意节点上.通过改进,能够保证诚信节点在这种条件下仍能获得一定的信誉值,使算法在恶意节点合谋时仍然保证一定的可用性,因此,EigenTrust 算法在恶意节点比例很高时仍然有效,很大程度上得益于这些假定的诚信节点的存在.但这种思路也还存在不足之处,即实际网络环境中如何确定这些所谓的诚信节点,因为这些节点必须保证始终如一的诚信.

4.2 算法复杂性对比

分布式算法的另一个评价指标是算法实现的复杂性.

EigenTrust 算法中全局信誉值计算是一个迭代过程,结束条件是所有节点最后两轮迭代计算的信誉值误差低于误差阈值.误差阈值的选取与信誉值的精度要求与节点数量相关.迭代次数多就意味着计算和通信开销大.由文献[5]的图 1 可以看出,EigenTrust 算法每次计算迭代 5 次后已经收敛;而本文的仿真中也发现,EigenTrust 算法的平均收敛时间约为 3 次.而且迭代求解的另一个缺点是迭代计算时有同步要求,否则不容易收敛.

本文算法中,因为不需要迭代,每轮计算仅需 1 次,因此在相同计算和通信开销的情况下,本文算法的计算时间间隔可以缩短为 1/5~1/3,从而能够更快地体现实际信誉值的变化.而且因为不需要迭代,本文算法没有同步要求,因此,算法实现复杂性降低.

5 结 论

本文提出一种分布式信任机制,基于历史交易,并采用分布式迭代方式为每个节点计算买全局信誉值和卖全局信誉值,便于用户判断交易对象的善恶,从而降低恶意交易概率.最后给出了与 EigenTrust 算法的对比实验和性能分析.

References:

[1] Kahney L. Cheaters bow to peer pressure. 2001. http://www.wired.com/news/technology/0,1282,41838,00.html

[2] Liang J, Kumar R, Xi Y, Ross KW. Pollution in file sharing systems. In: Makki K, Knightly E, eds. Proc. of the IEEE Infocom 2005, INFOCOM. New York: IEEE Press, 2005. 1174−1185.

[3] Page L, Brin S, Motwani R, Winograd T. The PageRank citation ranking: Bringing order to the Web. Technical Report, SIDL-WP-1999-0120, Stanford University, 1999.

[4] Lian Q, Peng Y, Yang M, Zhang Z, Dai YF, Li XM. Robust incentives via multi-level tit-for-tat. In: Proc. of the 5th Int'l Workshop on Peer-to-Peer Systems (IPTPS). Santa Barbara, 2006. http://iptps06.cs.ucsb.edu/

[5] Kamvar SD, Schlosser MT, Garcia-Molina H. The EigenTrust algorithm for reputation management in P2P networks. In: Bakonyi P, Hencsey G, eds. Proc. of the 12th Int'l Conf. on World Wide Web. Budapest: ACM Press, 2003. 640−651.

[6] Guha R, Kumar R, Raghavan. P, Tomkins A. Propagation of trust and distrust. In: Feldman SI, Uretsky M, Najork M, Wills CE, eds. Proc. of the 13th Int'l Conf. on World Wide Web. New York: ACM Press, 2004. 403−412.

[7] Song S, Hwang K, Zhou R, Kwok YK. Trusted P2P transactions with fuzzy reputation aggregation. IEEE Internet Computing, 2005, 9(6):24−34.

[8] Aberer K, Despotovic Z. Managing trust in a peer-to-peer information system. In: Paques H, Liu L, Grossman D, eds. Proc. of the 10th Int'l Conf. on Information and Knowledge Management (CIKM 2001). Atlanta: ACM Press, 2001. 310−317.

[9] Swamynathan G, Zhao BY, Almeroth KC. Exploring the feasibility of proactive reputations. In: Proc. of the 5th Int'l Workshop on Peer-to-Peer Systems (IPTPS). Santa Barbara, 2006. http://iptps06.cs.ucsb.edu/

[10] Stoica I, Morris R, Liben-Nowell D, Karger DR, Kaashoek MF, Dabek F, Balakrishnan H. Chord: A scalable peer-to-peer lookup protocol for Internet applications. IEEE/ACM Trans. on Networking, 2003,11(1):17−32.

[11] http://www.cs.bu.edu/brite/

附录

定理. 节点在与未知节点发生交易时,本文算法选择恶意节点作为交易对象的概率低于 EigenTrust 算法.

证明:根据本文实验假设,交易发起节点最多找到 3 个满足要求的节点,通过比较全局信誉值选择最好的节点作为交易对象.这里,我们只证明有 3 种满足要求的节点的情况,其他情况可类似证明.

假设:恶意节点比例为m;T_h和T_m分别表示诚信节点的信誉值和恶意节点的信誉值;恶意节点信誉值的概率密度函数为$f_1(x)$,$x\in[0,1]$;EigenTrust算法中,诚信节点的概率密度函数为$f_2(x)$,$x\in[0,1]$;本文算法中,诚信节点概率密度函数为$f_3(x)$,$x\in[0.5,1]$;各分布函数记为$f_N(x)$;同时,假设两种算法中诚信节点的分布比例相同,因此有$f_3(x)=2f_2(2(x-0.5))$,$x\in[0.5,1]$,下面式子中下标O和E分别代表本文算法和EigenTrust算法,则节点误选恶意节点作为交易对象的概率为

$$Pe=P[N_m=3]+P[N_m=2]\times P[T_h<\max(T_{m1},T_{m2})]+P[N_m=1]\times P[T_m>\max(T_{h1},T_{h2})],$$

$$\begin{cases} P[N_m=3]=C(M,3)/C(N,3)\approx m^3, & N>>1 \\ P[N_m=2]=C(M,2)\times C(N-M,1)/C(N,3)\approx 3m^2(1-m), & N>>1 \\ P[N_m=1]=C(M,1)\times C(N-M,2)/C(N,3)\approx 3m(1-m)^2, & N>>1 \end{cases},$$

$$\begin{cases} P_E[T_h<\max(T_{m1},T_{m2})]=1-\int_0^1\int_0^x\int_0^x f_1(x_1)f_1(x_2)f_2(x)\mathrm{d}x_1\mathrm{d}x_2\mathrm{d}x=1-\int_0^1 F_1^2(x)f_2(x)\mathrm{d}x \\ P_O[T_h<\max(T_{m1},T_{m2})]=1-\int_{0.5}^1\int_0^x\int_0^x f_1(x_1)f_1(x_2)f_3(x)\mathrm{d}x_1\mathrm{d}x_2\mathrm{d}x=1-\int_0^1 F_1^2\left(\frac{1}{2}x+0.5\right)f_2(x)\mathrm{d}x \end{cases}.$$

因为概率分布函数是单调递增函数,而在 $x\in[0,1]$时,有 $\frac{1}{2}x+0.5\geq x$,因此,

$$\int_0^1 F_1^2(x)f_2(x)\mathrm{d}x<\int_0^1 F_1^2\left(\frac{1}{2}x+0.5\right)f_2(x)\mathrm{d}x.$$

比较可得:$P_O[T_h<\max(T_{m1},T_{m2})]<P_E[T_h<\max(T_{m1},T_{m2})]$.

$$P_E[T_m>\max(T_{h1},T_{h2})]=\int_0^1\int_0^x\int_0^x f_2(x_1)f_2(x_2)f_1(x)\mathrm{d}x_1\mathrm{d}x_2\mathrm{d}x=\int_0^1 F_2^2(x)f_1(x)\mathrm{d}x,$$

$$P_O[T_m>\max(T_{h1},T_{h2})]=\int_{0.5}^1\int_{0.5}^x\int_{0.5}^x f_3(x_1)f_3(x_2)f_1(x)\mathrm{d}x_1\mathrm{d}x_2\mathrm{d}x=\int_{0.5}^1 F_2^2(2x-1)f_1(x)\mathrm{d}x.$$

当 $x\in[0.5,1]$时,有 $x\geq 2x-1$,因此,

$$\int_{0.5}^1 F_2^2(2x-1)f_1(x)\mathrm{d}x<\int_{0.5}^1 F_2^2(x)f_1(x)\mathrm{d}x<\int_0^1 F_2^2(x)f_1(x)\mathrm{d}x.$$

比较可得:$P_O[T_h>\max(T_{h1},T_{h2})]<P_E[T_m>\max(T_{h1},T_{h2})]$.

综合比较,可以得出结论:$Pe_E>Pe_O$,即本文算法选择恶意节点为交易对象的概率比EigenTrust算法要小.

彭冬生(1975－),男,江西吉安人,博士生,主要研究领域为信任机制,网络安全.

刘卫东(1968－),男,博士,副教授,CCF 高级会员,主要研究领域为网络计算,网络服务质量.

林闯(1948－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机系统与网络性能评价,随机 Petri 网,逻辑推理模型.