ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, Vol.20, No.6, June 2009, pp.1602−1616
doi: 10.3724/SP.J.1001.2009.03408

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

# 一种基于行程编码的P2P网络动态信任模型*

方 群[1,4+], 吉 逸[2], 吴国新[1,3], 赵生慧[1,3], 吴 鹏[1,3]

[1](东南大学 计算机科学与工程学院,江苏 南京 210096)

[2](东南大学 软件学院,江苏 南京 210096)

[3](计算机网络和信息集成教育部重点实验室(东南大学),江苏 南京 210096)

[4](安徽师范大学 计算机科学系,安徽 芜湖 241000)

## Run Length Coding-Based Dynamic Trust Model for P2P Network

FANG Qun[1,4+], JI Yi[2], WU Guo-Xin[1,3], ZHAO Sheng-Hui[1,3], WU Peng[1,3]

[1](School of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

[2](College of Software Engineering, Southeast University, Nanjing 210096, China)

[3](Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing 210096, China)

[4](Department of Computer Science, Anhui Normal University, Wuhu 241000, China)

+ Corresponding author: E-mail: fq0520@gmail.com

**Fang Q, Ji Y, Wu GX, Zhao SH, Wu P. Run length coding-based dynamic trust model for p2p network. *Journal of Software*, 2009,20(6):1602−1616.** http://www.jos.org.cn/1000-9825/3408.htm

**Abstract**: In this paper, RunTrust, a trust model based on Run-length coding algorithm, performs trust evaluation by compressing records of peers' behaviors which contain more information including time dimension, so it can exhibit good performance in trust evaluation and detecting malicious especially oscillatory behaviors. Moreover, the capability of filtering false feedbacks is also improved. The simulating results indicated that RunTrust has significantly raised the efficiency of trust management system only at a very low cost. The work on RunTrust has paved for the research on trust data compacting.

**Key words**: peer-to-peer network; dynamic trust model; run-length coding; malicious behavior; income

**摘 要**: 基于数据压缩领域中的行程编码理论提出一种 RunTrust 动态信任模型,以系统收益衡量节点合作成果,以经过压缩的节点合作记录作为信任评估依据,既增加了评估依赖的信息量,也保留了时间维度,提高了信任度评估的准确性和动态恶意行为的判别能力;借助基于时间的反馈聚合算法,通过特殊的反馈过滤策略和动态参数调整,能够增强针对恶意反馈的过滤能力.仿真实验结果表明,RunTrust 以牺牲少量处理能力换取系统性能的显著提升.RunTrust 模型的提出为信任数据压缩研究奠定了基础.

**关键词**: P2P 网络;动态信任模型;行程编码;恶意行为;收益

**中图法分类号**: TP393 **文献标识码**: A

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z422 (国家高技术研究发展计划(863))

Received 2007-07-18; Revised 2008-02-19; Accepted 2008-07-02

P2P(peer to peer,对等网)中信任度(trust)是节点合作的基础,评估主体(trustor)依据评估客体(trustee)节点的历史行为,对合作结果进行预期,按照一定策略选取合作对象.可见,信任度评估的准确性是影响系统性能的重要因素.因信任评估是参数估计问题,故评估对象行为历史样本越完整,对其信任的估计就越准确.另一方面,信任度也应该反映出节点行为的动态变化规律,特别是恶意节点有可能针对信任机制采取更加隐蔽的、策略性的攻击行为,且往往相互勾结实施对系统的侵害.在信任评估时,节点除利用自身经验外,将更大程度地依赖于友邻节点提供的推荐信息.传统信任模型多采取平均值算法,受询节点仅提供单一信任值而隐藏节点行为的细节,且由于各种模型评估能力的差异,造成信任评估在精度和动态性方面不够理想,对抗动态恶意行为的能力较弱.

为了建立 P2P 网络的动态信任管理,满足信任度评估在精度和动态适应性两方面的需求,本文提出了一种动态信任模型——RunTrust,它采取行程编码数据压缩方法对节点行为历史记录进行压缩形成行为链,在降低数据量的同时保留了用户行为的时间维度,利用直接信任度和间接信任度的二元组综合描述节点的全局信任度,采取自适应反馈过滤算法对节点反馈进行筛选以提高准确度.另外,还给出信任信息的分布式存储策略、反馈传递及过滤算法以及反馈聚合算法.通过与典型动态信任模型的仿真对比分析,表明 RunTrust 比其他相关模型具有更加准确的行为预测能力、较好的动态适应能力和自我调控能力,能够有效地抑制恶意节点的策略性摇摆行为和勾结行为对系统的攻击,实施代价可接受.

本文第 1 节简要回顾研究者在动态信任模型方面的相关工作,重点分析提高信任模型的评估精度和动态适应能力所采取的方法.第 2 节介绍基于行程的动态信任管理模型 RunTrust,包括周期、链、收益、损失、直接信任度、间接信任度以及全局信任度的概念及计算公式.第 3 节主要介绍基于行为链的反馈过滤和聚合算法.第 4 节讨论 RunTrust 模型针对几种典型的恶意行为采取的检测和抑制策略.第 5 节主要介绍 RunTrust 的分布式实现方案,包括其依赖的基础网络拓扑、信任数据的存储和传递等.第 6 节通过仿真分析揭示 RunTrust 与其他典型信任模型在主要性能指标上的差异.第 7 节总结 RunTrust 模型的优点并指出今后的工作方向.

## 1 相关工作

1994 年,Marsh[1]率先将社会学范畴的信任概念引入计算机网络,信任研究内容主要包括信任模型、信任管理和信任决策,而又以信任模型为基础.经过 10 多年的研究,目前信任管理已经广泛应用于包括电子商务在内的多种领域.P2P网络是近年来出现的一种新型网络类型.2001 年,Aberer等人[2]率先研究了P2P环境下的信任管理问题,并探讨了P2P信任管理的可能性,把信任模型的构造理论分为社会学理论、概率理论和博弈理论 3 类[3],现有研究基本按照这 3 个方向开展.

2001 年以来是P2P信任模型研究的活跃期,国外典型的P2P网络分布式信任模型主要有Aberer等人的基于抱怨(complaint)的信任模型[2]、Damiani等人的基于资源信誉的XRep模型[4]、WANG等人的基于Bayesian的信任模型[5]、Standford的基于特征向量的EigenTrust模型[6]、Khambatti等人的基于角色的信任模型[7]等.此类模型基本上都是静态的模型,对节点行为的动态变化不敏感,也没有考虑复杂的策略性动态恶意行为.国内也有许多大学致力于信任模型的研究,包括清华大学、北京大学、国防科学技术大学、复旦大学、南京大学、武汉大学、华中科技大学等,研究范围涵盖包括信任模型在内的多种信任管理内容.

传统信任模型[1−7]中,信任评估多采取平均值算法,往往忽视时间因素的重要作用.Elofson[8]首先提出动态信任(dynamics of trust)的概念,认为信任对于某个实体来说不是固定的,而是依赖于对评估客体的观察结果,随着观察结果的变化而提高或降低,确定了观察对信任评估的决定性作用,同时,承认时间在信任进化过程中的重要作用.Jonker等人[9]采用形式化方法描述信任的动态性,定义了信任进化函数(trust evolution function)和信任更新函数(trust update function)及其主要性质,并建议所有动态信任模型都应当满足两种函数的约束条件,因而文献[9]已成为指导动态信任模型设计的重要参考.Li等人[10]的工作是对近期动态信任的比较性总结,认为动态性是信任关系量化与预测的最大挑战,分析了动态信任关系的相关概念、主要问题和研究方法,选取新的较为典型的动态信任模型及其使用的数学理论方法进行评述.

EigenTrust模型[6]基于时齐Markov理论和传递信任的概念,通过节点间推荐度的迭代计算目标节点的全局

信任度(即Markov过程转移矩阵对应于特征根 1 的左特征向量).在预先存在可信节点组的假设下,它能够抵御多种威胁,但此假设不具有普遍性;该模型的实现比较复杂,且因节点间密切合作和时间同步要求较高而难以实现.另外,它对节点的服务信任度和推荐信任度不加区别,因而抵抗恶意的能力有限.Dou等人[11]对EigenTrust加以改进,提出对恶意反馈及协同作弊行为的抵制方法,针对 4 类恶意行为分别提出防御策略,抵抗攻击的能力比EigenTrust有显著提高.以上两种模型均采用平均值方法,没有考虑时间维度,因而它们抵制策略性恶意行为的能力较弱.

Xiong等人提出的PeerTrust[12]是较为典型的动态信任模型,它利用自适应宽度的滑动窗口机制动态调整节点行为的监控时间段,能够有效预测节点信任度在近期内的变化趋势,从而提高了抑制策略性恶意节点的能力.但它仍存在难于防止反复建立信任然后实施攻击的摇摆行为和较为复杂的合伙欺骗行为问题,同时,由于滑动窗口宽度较小,在利用PSM算法进行反馈信任度评估时,因节点交集较小而影响评估的准确性.

Duma等人[13]提出了动态信任的测度(metrics)模型,基于行为评价时基序列定义了短期信任、长期信任、惩罚向量等指标,能够全面反映节点在短期和长期内的行为规律,对于策略性的恶意行为有较佳的检测能力,也满足文献[9]中关于信任函数的要求.但是文献[13]只提出了一个理论性的信任评价尺度框架,而没有讨论具体实现的情形,且采用静态的推荐信任度与实际不符.另外,它提出的信任评估算法是基于单个评价的,因而计算频繁,开销较大,且抵御复杂恶意行为的能力有限,仍存在较大的改进余地.Chang等人[14]进一步提出一种基于时间帧的动态信任模型——DyTrust,引入近期信任、长期信任、累积滥用信任和反馈可信度 4 个参数来计算节点信任度,通过反馈控制机制动态调节评估参数,提高了信任模型的动态适应能力.但是,DyTrust的时间帧概念只是对节点历史行为记录的按次划分,并未真正引入时间因素来判别恶意行为(如部分恶意节点密集地增加交互次数以提高信任度),且在单个时间帧内采取平均值法,因而削弱了对抗策略动态恶意的能力.

本文与以上工作最本质的区别在于,RunTrust 采用行程编码算法对节点行为历史进行压缩的同时,仍保留行为发生的时间因素,在综合考察合作上下文的基础上,引入周期收益、积累速率、绝对收益和相对收益等多项指标来辅助节点信任评估,进一步提高信任评估的准确性,增强抵抗恶意行为的能力.另外,本文还给出信任管理与 P2P 网络的接口框架以及基于 Chord 的实现方案,故整个信任模型较为完整、可行.

## 2 基于行程的动态信任模型——RunTrust

P2P 网络鼓励节点之间的合作,合作结果对系统性能有正面或负面的影响,目前,普遍通过调节参与者的信任指标来显示系统对合作结果的取向性,激励更加有效的正面合作.文献[9]中把节点间合作经验表示为一个时间序列:++++–+ –+ – – – –…,其中,+和–分别表示正面和负面经验.其作者认为,+和–出现的时间顺序是影响信任评估结果的关键因素,决定了信任演化的路线.本文认为,节点的行为序列完全显示了该节点的优劣性质,而这种性质不但取决于事件发生的顺序,也取决于发生的间隔时间,因而只要保留节点完整的行为历史(即包括事件和发生时刻)即可以判断其优劣.但在实际分布式的P2P网络中,处理和存储效率是需要重点考虑的问题,因而需要采取更加灵活、高效的方法.

### 2.1 行程编码

从信息论的角度来看,信任评估是对节点行为历史的浓缩,但往往会损失其中的重要特征,造成评估的不精确.行程编码是数据压缩领域中的一种重要的无损压缩方法,其基本思想是将待压缩的数据序列化,然后从序列的开始进行扫描,对于同种数据符号统计数目,在扫描过程中只记录符号和数目及出现的先后次序,扫描结束时压缩即完成.该方法特别适用于有大量连续相同符号的场合,如黑白图片的压缩,能够达到较好的压缩效果.在P2P 网络中,节点的行为历史也具有类似特点,因而可以采取行程编码的方法对其进行压缩,在减少数据量的同时保持其基本信息,提高处理的效率.

图 1(a)显示的是合作发生的时间及结果,其中,向上和向下的箭头分别表示正面和负面评价.很明显,在$[t_1,t_2]$内有连续 6 次的正面评价,而从$t_2$开始有连续 2 次负面评价.利用行程编码的思想,按照评价类型(正面或负面)将图 1(a)中的时间轴划分为若干段,每段内的评价累加至首发时刻即得到图 1(b).比较图 1(b)和图 1(a)可知,

采用行程编码压缩算法后,数据量显著减少.

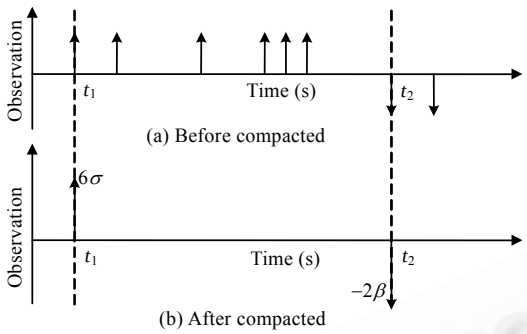



Fig.1 Comparison of before and after run-length compacted behavior history

图 1 采取行程压缩前后的行为历史对比

### 2.2 基本概念

虽然对行为的评价往往受到上下文(context)的影响,不同评价主体对同一行为的评价存在差异,但为了便于讨论问题,我们把不同上下文对评价结果的影响包涵在收益(或损失)的概念中.

**定义 1(收益与损失)**. 假定节点间单次合作结果的正面收益(income,简称收益)与负面收益(loss,简称损失)分别为$\alpha,-\beta(\alpha,\beta>0)$,称$\alpha,\beta$分别为收益率和损失率.多次合作的结果是单次合作结果的代数和.

**定义 2(周期)**. 一个行为周期(period)是由若干个连续的正面行为和若干连续的负面行为构成的时间段,用$P$表示.在一个周期内,节点的行为性质仅发生一次逆变.经过压缩的周期是一个四元组$(V^+,t^+,V^-,t^-)$,其中,$V^+$表示收益,$V^-$表示损失,$t^+,t^-$分别表示正面和负面评价的首发时刻.

**定义 3(链)**. 由多个连续但时间上互不重叠的周期组成的行为过程称为行为链(chain),用$C$表示.若行为链$C$由$n$个周期组成,则有$C=P_1P_2\ldots P_{n-1}P_n$,其中,$P_i(i=1,2,\ldots,n)$表示周期.行为逆变(正→负或负→正)次数称为该行为链$C$的摇摆频率,用$f_C$表示.摇摆频率与周期数一般有$f_C=2n-1$.

有时行为链只包括半个周期,例如,只有正半周期$(V^+,t^+)$或者负半周期$(V^-,t^-)$,则其频率$f_C=0$,表示在观察时间段内节点行为没有发生逆变.

### 2.3 基本评估指标

**定义 4(周期收益)**. 在节点 $j$ 的周期 $P$ 内,若用 $N_j^+,N_j^-$ 分别表示合作结果为正面和负面的次数,则定义节点 $j$ 的周期收益(period income)为

$$v_j(P)=V_j^+ +V_j^- =\alpha N_j^+ -\beta N_j^- \tag{1}$$

EigenTrust[6]模型中取$\alpha=\beta=1$,把收益率与损失率等同对待.但根据人类社会中信任建立的规律,对恶意行为应当采取惩罚措施,有$\alpha<\beta$,即鼓励正面行为,惩罚负面行为.$\beta/\alpha$越大,说明对负面行为的惩罚越严厉.

**定义 5(积累速率)**. 在节点 $j$ 的周期 $P$ 内,若用 $V_j^+,V_j^-,t$ 分别表示正、负面的收益及周期 $P$ 的时长,则定义节点 $j$ 的积累速率(aggregate speed)为

$$s_j(P)=(|V_j^+|+|V_j^-|)/t \tag{2}$$

**定义 6(绝对收益)**. 若 $j$ 的行为链 $C$ 有 $n(\geq 1)$个周期,则 $C$ 的绝对收益(absolute income)为

$$V_j(C)=\frac{1}{f_C+1}\sum_{k=1}^{n}\rho^{n-k}v_j(P_k),\ (\rho<1) \tag{3}$$

其中,$f_C$是行为链$C$的摇摆频率,正常数$\rho(\leq 1)$称为折扣因子(discount factor),用来表示时间流逝对信息价值的影响.综合收益与节点的近期行为密切相关,越早的合作结果对系统的收益贡献率越少,在满意度评估中的重要性也越低.反过来则可以认为绝对收益值对节点行为的近期变化较为敏感.若令$\rho=1,f_C=0$,则公式(3)将褪化为平均

值模型.可见,平均值算法是公式(3)的一种特殊情形.为简化计算,对于两个时间相继且无重叠的行为链$C_1$和$C_2$,可以利用迭代公式(4)计算整个行为链的绝对收益值:

$$V_j(C_1C_2)=\frac{(f_{C_1}+1)\rho^{n_2}V_j(C_1)+(f_{C_2}+1)V_j(C_2)}{f_{C_1}+f_{C_2}+1} \tag{4}$$

其中, $f_{C_1}, n_2, f_{C_2}$ 分别为行为链$C_1$的频率和$C_2$的周期数、频率.

**定义 7(相对收益)**. 若 $j$ 的行为链 $C$ 有 $n(\geq 1)$个周期,则 $C$ 的相对收益(relative income)为

$$V_j^*(C)=\begin{cases}V_j(C)\Big/\sum_{k=1}^{n}V_k^+, & V_j(C)>0 \text{ and } \sum_{k=1}^{n}V_k^+>0\\ 0, & \text{others}\end{cases} \tag{5}$$

其中,$P_k(k=1,2,\ldots,n)$是组成$C$的所有周期.由公式(1)可知, $v_j(P_k)\leq V_k^+$ ,且由$\rho(\leq 1)$和$f_C\geq 0$ 可得 $V_j^*(C)\in[0,1]$.

由公式(5)可知,若节点$j$是诚实节点,其$n=0, f_C=0$ 且在整个行为链中不存在负面收益(即损失),故 $V_j^*(C)=1$;若节点$j$是静态恶意节点,由于其$v_j(P)<0$,故 $V_j^*(C)=0$.对于那些策略性恶意节点,由于摇摆行为的存在,在单个周期内有 $v_j(P_k)<V_k^+$ 且$f_C>0$,故有 $0<V_j^*(C)<1$,且摇摆频率越大,相对收益率越低.

同样地,也可以利用公式(6)简化相对收益的合并计算:

$$V_j^*(C_1C_2)=\frac{(f_{C_1}+1)\rho^{n_2}V_j(C_1)+(f_{C_2}+1)V_j(C_2)}{(f_{C_1}+f_{C_2}+1)\left(\sum_{k=1}^{n_1}V_{1k}^+ +\sum_{l=1}^{n_2}V_{2l}^+\right)} \tag{6}$$

### 2.4 全局信任度评估

**定义 8(直接信任度)**. 若在指定时间段内节点$i$观察得到节点$j$的行为链为$C_{i,j}$,则定义节点$i$对节点$j$的直接信任度$DT_{i,j}$为$C_{i,j}$的相对收益,即

$$DT_{i,j}=V_j^*(C_{i,j}) \tag{7}$$

**定义 9(间接信任度)**. 若在指定时间段内由参与合作的所有节点(不包含评估主体节点)观察得到节点$j$的行为链为$C_j$,则定义节点$j$的间接信任度$IT_j$为$C_j$的相对收益,即

$$IT_j=V_j^*(C_j) \tag{8}$$

**定义 10(全局信任度)**. 信任评估主体通过自身经验得到直接信任度$DT_{i,j}$,通过聚合反馈信息得到间接信任度$IT_j$,将二者结合得到全局信任度:

$$T_j=\max\{\mu DT_{i,j}+(1-\mu)IT_j,\sigma\} \tag{9}$$

公式(9)中的$\mu$代表评估主体的自信指数,当$\mu=1$ 时,该节点仅相信自身的合作经验,因而是武断的;而当$\mu=0$时,它仅相信其他节点的推荐信息,因而是寡断的.$\sigma(>0)$是一个较小的正整数,作为新加入节点在没有任何历史记录时的初始全局信任度,这一措施是为了保证新节点能够有机会参与合作积累信任.

## 3 反馈过滤及聚合

### 3.1 反馈过滤

传统信任模型多以单一的信任度值作为反馈,且对所有反馈不加区分,而仅利用反馈信任度作为权重调整反馈信息在信任评估中的影响程度,影响信任评估精度.PeerTrust,DyTrust 等模型中节点频繁地计算公共节点的评价相似度来获取对方反馈信任度值,由于计算公式复杂,因而会增加系统开销,降低响应速度.RunTrust 以节点行为链作为反馈形式,无形中增加了恶意节点伪造反馈信息的难度.从行为链本身提取特征参数来判别其真伪,降低了通信开销.由公式(9)可知,节点若想提高信任水平必须朝两个方向努力:大量产生正面结果,同时避免摇摆行为;反之,则导致信任度急剧降低.正常情况下,两个节点间的交易不会过于频繁(否则有勾结嫌疑),因而由公式(3)计算出的绝对收益不会太大,而由公式(2)计算的积累速率也不可能过高.另外,由公式(5)可知,

$0 \le V_j^*(C_j) \le 1$,而取值 0 和 1 可能是异常反馈.因此,设计反馈过滤算法见算法 1.

**算法 1**. 反馈过滤算法.

***feedbackFilter*(*feedback*){**

//输入:反馈行为链集合***feedback***=$\{C_{k,j}|k$与$j$有合作$\}$;

//输出:过滤后的反馈信息.

(a) 利用公式(2)、公式(3)、公式(5)分别计算$C_{k,j} \in$***feedback***的$\overline{s_j}(C_{k,j}), V_j(C_{k,j}), V_j^*(C_{k,j})$;

(b) 将 ***feedback*** 按照$V_j^*(C_{k,j}), V_j(C_{k,j}), \overline{s_j}(C_{k,j})$为第 1~第 3 关键字降序排列;

(c) 从 ***feedback*** 删除平均积累速率$\overline{s_j}(C_{k,j}) > \upsilon$的链;

(d) 从 ***feedback*** 删除前$\gamma/2$部分和后$\gamma/2$部分的链;

(e) 返回 ***feedback***.

}

**定理 1**. 若令$n=|\{C_{k,j}|k$与$j$有合作$\}|$,链的平均周期数为$m$,则算法 1 的时间复杂度为$O(n(n+m))$.

证明:算法 1 的总时间复杂度是步骤(a)~步骤(e)各步骤时间复杂度之和:

$$O(nm)+O(n^2)+O(n)+O(n)=O(n(n+m)).$$ □

算法 1 第(c)步中的$\upsilon$是积累速率的阈值,用于限制节点快速积累信任后启动的恶意行为,也用于过滤诋毁反馈,保证所有反馈的节点行为保持平稳地变化.第(d)步采取"去掉最高分最低分"的简单策略,引入一个过滤阈值$\gamma(0<\gamma<1)$,用于估计所有节点中恶意节点所占比例 $c(0<c<1)$.算法 1 对恶意反馈的有效性将在第 4.2 节中讨论.

### 3.2 反馈聚合

节点 $i$ 获得 $j$ 的所有行为链,经过反馈过滤算法将可疑的反馈信息剔除,再将剩余的行为链进行聚合,聚合结果形成针对节点 $j$ 的全局评价信息.节点行为周期聚合是行为链聚合的原子操作,它将两个来源不同但关于相同目标的节点行为周期重叠形成一个行为链,图 2 和图 3 显示了周期与半周期的聚合结果.

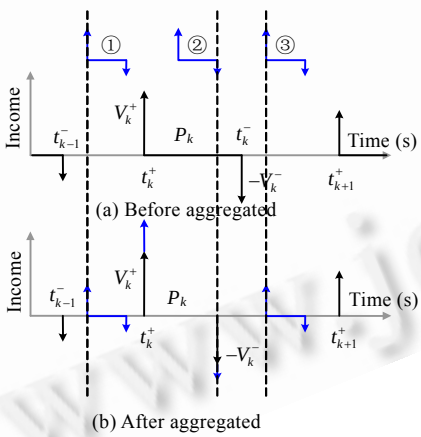



Fig.2 Aggregation of period

图 2 周期的聚合

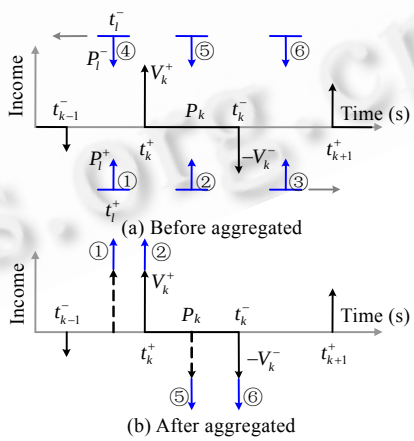



Fig.3 Aggregation of half period

图 3 半周期的聚合

图 2(a)显示的是待聚合周期在时间上不交错的情况,此时聚合方法比较简单,图 2(b)中显示情形①~情形③周期聚合后的结果.图 3(a)中是比较复杂的情况,此时各周期在时间上有重叠,无法进行简单的聚合操作,需要将一个周期拆分为两个半周期分别进行聚合操作,此时有 6 种情况,其中,情形①~情形③显示聚合正半周期,情形④~情形⑥显示聚合负半周期.情形③和情形④属于特殊情况,分别交由后续和前趋周期完成聚合.图 3(b)显示了其他 4 种情形的聚合结果.

定义周期聚合函数:

$$AggreP(P_k,P_l)=\begin{cases}P_lP_k, & t_{k-1}^-<t_l^+<t_l^-<t_k^+\\ (V_k^++V_l^+,t_k^+,V_k^-+V_l^-,t_l^-), & t_k^+<t_l^+<t_l^-<t_k^-\\ P_kP_l, & t_k^-<t_l^+<t_l^-<t_{k+1}^+\\ null, & \text{else}\end{cases} \tag{10}$$

其中, $P_k=(V_k^+,t_k^+,V_k^-,t_k^-)$, $P_l=(V_l^+,t_l^+,V_l^-,t_l^-)$ 表示两个待聚合的周期,前 3 个结果分别对应图 2 中的情形①~情形③这 3 种情形,第 4 种情况表示两个周期在时间上有交错,需要采取分部聚合方法.

定义正周期聚合函数:

$$AggreP^+(P_k,P_l^+)=\begin{cases}(V_k^++V_l^+,t_l^+,V_k^-,t_k^-), & t_l^+<t_k^+\\ (V_k^++V_l^+,t_k^+,V_k^-,t_k^-), & t_k^+\le t_l^+<t_k^-\\ null, & t_l^+\ge t_k^-\end{cases} \tag{11}$$

其中, $P_k=(V_k^+,t_k^+,V_k^-,t_k^-)$, $P_l^+=(V_l^+,t_l^+)$.前两种情况将正周期有效聚合,第 3 种情况返回 *null* 表示两者未能聚合, $P_l^+$ 应当与下一个周期聚合,或者作为独立的正半周期放在行为链的末尾.

定义负周期聚合函数:

$$AggreP^-(P_k,P_l^-)=\begin{cases}null, & t_l^-<t_k^+\\ (V_k^+,t_k^+,V_k^-+V_l^-,t_l^-), & t_k^+\le t_l^-<t_k^-\\ (V_k^+,t_k^+,V_k^-+V_l^-,t_k^-), & t_l^-\ge t_k^-\end{cases} \tag{12}$$

其中, $P_k=(V_k^+,t_k^+,V_k^-,t_k^-)$, $P_l^-=(V_l^-,t_l^-)$.第 1 种情况表示 $P_l^-$ 应当与前一个周期聚合,或者作为一个独立的负半周期放在行为链的开始;后两种情况可以有效聚合.

在多个行为链进行聚合时,先选取起始时间最靠前的一个链作为母链,将其他链拆分为周期使用周期聚合函数聚合在母链上.如果整个周期无法聚合,则再将周期拆分为正负半周期分别使用对应的聚合函数.

**定理 2**. 若有$k$个链,每条链平均有$m$个周期,则$k$个链聚合的时间复杂度为$O(k^2m^2)$.

对于大多数正常节点来说,$m$ 较小且接近常数,因而聚合操作的时间开销取决于行为链数目 $k$.

## 4 针对恶意行为的策略

根据恶意节点间的关系可以将恶意行为分为独立和勾结两种,独立恶意节点单独实施恶意攻击,而勾结是指多个恶意节点共同发动恶意攻击.按照恶意节点实施意图可以将恶意行为分为直接和间接恶意行为:直接恶意行为是指恶意节点在与其他节点交互时产生直接的恶意结果而达到恶意企图,如拒绝合作、提供伪造服务或恶意攻击等;间接恶意行为是指恶意节点通过反馈虚假的推荐而实施恶意企图,例如夸大恶意团伙中其他节点的信誉度或诋毁诚实节点的信誉度.无论是直接或间接恶意行为都将对系统造成损害,不过后者更具有隐蔽性,也较难被发现.按照实施的时间规律还可以将恶意行为分为静态和动态恶意行为:静态恶意行为是指节点自始至终实施恶意攻击;而动态恶意行为则是指恶意节点具有一定的摇摆性,在时间上则反映出诚实行为和恶意行为交替出现.在这两种恶意行为中,前者易发现而后者较为隐蔽,因而难以发现.另外还有两类恶意行为即 White Washing(漂白)攻击和 Sybil(女巫)攻击,其共同点是利用新身份加入系统,重新积累信任而再度伺机发动攻击;其差别在于,前者仅有单一身份而后者拥有多个不同身份,且能够操纵多个身份同时发起攻击,因而后者更难以防御.本文主要讨论 RunTrust 针对独立动态直接恶意行为、动态间接恶意行为和 Sybil 攻击的对策,对其他恶意行为也具有一定的防御能力.

### 4.1 独立动态直接恶意行为

摇摆行为的特征是其时间规律性,即正向行为和负向行为交替出现.在基于平均值的传统信任模型中,恶意节点能够在保持一定信任水平的前提下不断发起恶意攻击.直观来看,摇摆频率越高,节点的恶意程度也就越高.惩罚措施是迅速降低其信任度,以尽量减少其参与合作的机率.

RunTrust 模型与传统模型的本质区别在于对节点行为进行压缩的同时保留关键时间因素,即只关注摇摆行为发生的时刻(即节点行为性质发生逆转的时间)而忽略后续同类行为发生的时间,因而可以准确获知考察时间内指定节点摇摆行为发生的频率.在计算直接和间接信任度的公式(7)、公式(8)中都引入行为链的摇摆频率$f_C$的函数作为惩罚因子,其值越大,信任度值就越低.如果用$(f_C+1)^{-2}$或$2^{-(f_C+1)}$来替代公式中的$(f_C+1)^{-1}$因子,可以收到更好的惩罚效果.

在公式(2)中,设正整数$m$,$n$($m<n$)分别表示两个周期,当行为链长度从$m$增长至$n$时,对于相同的周期收益即$s_{i,j}(P_m)=s_{i,j}(P_n)$,在行为链收益中的贡献却有$s_{i,j}(P_m)/(2m+1)>s_{i,j}(P_n)/(2n+1)$.可见,随着摇摆频率的增加,恶意节点要想积累同等信任值的难度也相应地增加,受到更为严厉的惩罚,迫使恶意节点的攻击间隔时间越来越长.

在计算信任度的公式(7)~公式(9)中还引入折扣因子$\rho(\le 1)$,对于诚实节点,由于它没有摇摆行为,因而折扣因子不影响其信任度的评估.而对于恶意节点,引入折扣因子使其前期积累的信任逐渐被遗忘,在新的信任计算中的影响减弱,可以降低其通过累积信任而实施恶意攻击的可能性.

### 4.2 动态间接恶意行为

传统信任模型中通常使用反馈信任度来消除恶意反馈的干扰,采取比较公共交易节点的全局信任度来评价对方的反馈信任度.而以反馈信任度为权值聚合反馈信息,需要大量的系统开销,同时也无法抵抗摇摆的反馈攻击.因而,(I) 夸大恶意行为可以通过反馈一个正半周期(但积累速率较高)达到提升评估对象信任度的目的;诋毁行为则可以:(II) 通过反馈一个负半周期;或者(III) 用一个摇摆频率较高的行为链来降低评估对象的信任度.

正常情况下有$0\le V_j^*(C_j)\le 1$,而第(III)种恶意行为的$V_j^*$介于0和1之间且趋近于0,对于正常节点(包括恶意节点本身)来说可能性极小,可以判定其来源必定是恶意的.恶意行为的另外两种手法是:(I) 可使$V_j^*=1$;(II) 可使$V_j^*=0$.我们使用算法 1 对节点接收到的反馈信息进行过滤,它具有普适效应,能够对抗典型的夸大、诋毁和勾结等多种恶意反馈行为.影响过滤精度的参数有很多,但过滤阈值$\gamma$最为重要,它是对网络中恶意比例 $c$ 的估计值,需要根据网络当前情况进行动态的调整.

初始时,可以从邻居节点中获取各自的$\gamma$并加以算术平均得$\overline{\gamma}$,以$\min\{\overline{\gamma},\varepsilon\}$作为$\gamma$的初始值,其中,$\varepsilon(0<\varepsilon<1)$是一个较低的值.

但是,按照此种信息过滤策略可能导致合作的成功或者失败,因而产生收益$\alpha$或损失$\beta$.产生收益是因为对节点信任度准确评估的结果,说明参数$\gamma\ge c$;相反,则可能是因为$\gamma<c$ 造成的,因而需要修正$\gamma$值以适应真实情况的变化.$\gamma$的修正公式如下:

$$\gamma'=\min\left\{\max\left\{\gamma\left(1-\frac{V}{\alpha N}\right),\varepsilon\right\},1\right\} \tag{13}$$

其中,$\gamma'$表示修正后取值,$\gamma$代表原有的估计值,$V=\sum_k V_k(C_k)$表示节点在近期内的收益,$N$ 表示近期内的交易次数,$\alpha$仍为收益率.修正$\gamma$的时间间隔可选取近期收益为参照,即如果近期收益低于某阈值则进行修正.

由公式(13)可知,如果当近期收益 $V>0$ 时,修正后的$\gamma'<\gamma$,则说明原有对恶意节点比例估计较高,对反馈信息的过滤条件偏严,虽然可以保持反馈信息的正确性,但也排除了一些正确的反馈信息,产生不公平性;相反,如果当近期收益 $V<0$ 时,修正后的$\gamma'>\gamma$,则说明原有过滤条件偏宽,把恶意反馈作为真实信息参与聚合造成评估不准确.初始时假设$\gamma<c$,每次合作时由于聚合了过多的错误反馈而造成损失$\beta$,且由于$\beta>\alpha$,则调整后的$\gamma'=(1+\beta/\alpha)\gamma>2\gamma$,即修正值以指数速率增长,能够较快地接近实际值.采用上述优化策略经过多次调整,必要时也可以计算邻居节点提供的$\gamma$的平均值,实现对 $c$ 的无偏估计,从而达到较为令人满意的过滤效果.

### 4.3 Sybil攻击

信任系统可以限制恶意节点为所欲为,因而部分恶意节点为了继续实施恶意行为采取了更换身份的方法,利用新的身份实施攻击,同时达到逃避惩罚的双重目的.Sybil攻击中,恶意节点操控多个身份对网络同时发动攻击,在注册多个新身份后,恶意节点希望尽快地积累足够的信任然后实施攻击,但节点匿名性使得P2P网络难以

确认其真实身份,故防止此类恶意行为较为困难.SybilGuard[15]是近期较为成功的一种Sybil攻击防御系统,它基于社会网络在节点间建立信任关系边,利用随机游走方法将节点有效分割形成诚实节点与Sybil恶意节点两个互不相交的节点割集,进而采取措施限制两个割集节点间的联系,从而降低了Sybil节点实施恶意的可能性.

本文从控制Sybil恶意节点的新身份入手,采取两项措施防御Sybil攻击:其一是设置交易次数阈值$\theta$(如10),以限制新加入节点在初期的交易量.即必须超过这一交易量后新节点才能进入正常的信任评估,而在此之前只能拥有初始信任$\sigma$;第2项措施是设置积累速率阈值$\upsilon$(如0.1/s),在评价节点信任度时,同时计算它在行为链中各周期内积累速率(公式(2))的平均值,如果超过$\upsilon$则可判定该节点是恶意节点且处于快速积累期.通过以上两项措施,既可以延长 Sybil 恶意节点新身份的信任积累周期,也可以准确检测此类恶意节点,通过取消其参与合作的资格,能够有效防止 Sybil 等类似攻击.

## 5 动态信任管理系统的分布式实现

### 5.1 信任信息的存储和获取

信任管理系统是 P2P 网络中重要的安全基础设施,理论上认为,它应当与 P2P 拓扑结构无关.图 4 显示 RunTrust 信任管理框架以及信任信息分布式存储.

图4(a)中,RunTrust信任管理借助于P2P层提供的节点定位与数据存储功能提供数据给评估模块,评估结果提交给决策模块,为用户提供决策支持.但在实际系统中,其信任评估时间和安全可靠性部分依赖于底层 P2P 网络的性能.本文采用 Chord 结构化 P2P 网络作为 RunTrust 的底层平台,它在 P2P 领域极具代表性,且实现简单,能够达到较为满意的响应时间.另外,基于 DHT(distributed hash table)为节点 $j$ 指派一个管理节点 $DHT(j)$,存储 $j$ 与其他节点合作的行为链,能够保证可靠性,图 4(b)显示了这种存储方案.

图 4(b)左边显示节点间的 3 种操作,即评价信息维护、合作请求响应以及反馈信息获取.

(1) 评价信息维护.节点$k$与评价目标节点$j$合作后将评价结果发送给$j$的管理节点$DHT_j$,管理节点保存该评价.

(2) 合作请求响应.节点 $j$ 发出合作请求,$j$ 按照一定策略予以响应.

(3) 反馈信息获取.$i$ 在与 $j$ 合作前需要对 $j$ 的信任度进行评估,向其管理节点发出询问,管理节点将指定时间段内的所有相关行为链反馈给 $i$.

图 4(b)右边显示的是管理节点保存的关于节点 $j$ 的反馈信息.左边索引表中的每个项目指向一个行为链,表示其他节点与 $j$ 合作后对 $j$ 的行为评价.每条行为链由首部和周期链表两部分组成,首部包括节点标识、周期数、频率等基本信息;周期链表包含一个或多个周期,周期按时间逆序排列,最近的周期放在最前面便于添加操作.管理节点发送时则按时间顺序排列相关周期,便于聚合操作.

管理节点接收到节点 $k$ 对 $j$ 的评价报告后按照算法 2 添加至对应的行为链.

**算法 2**. 行为链维护算法.

***manChain*(*k*,*j*,*v*,*t*)** {

//输入:源 $k$ 和目标 $j$、评价 $v$ 及发生时刻 $t$;

//输出:更新后的行为链.

(a) 查找行为链$C_{k,j}$.

(b) 如果返回结果为空,则:

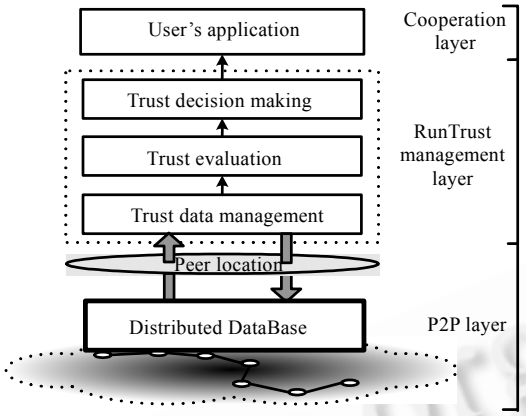
(i) 创建新行为链$C_{k,j}$;创建新周期;将周期加入行为链$C_{k,j}$;在索引表中插入$C_{k,j}$.

(ii) 否则,将评价加入$C_{k,j}$末尾(若由负-正跳变,则产生一个新周期).

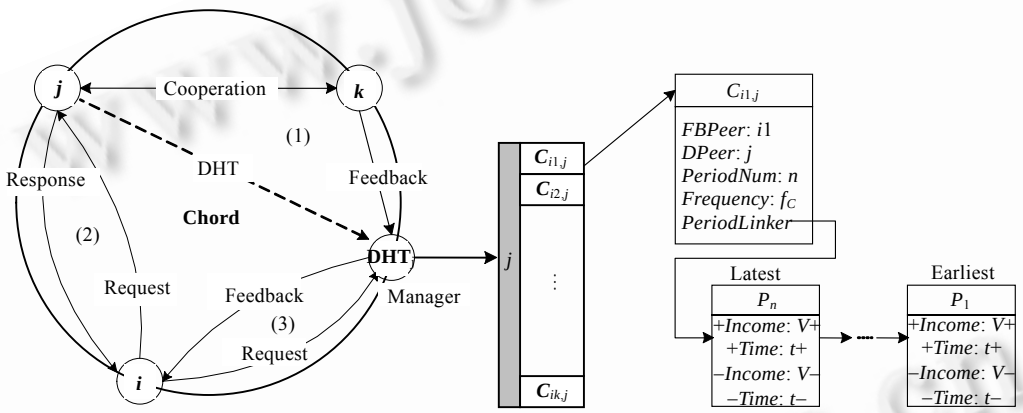
}

**定理 3**. 算法 2 的时间复杂度为 $O(1)$.

为了节省存储空间及考虑时效性,可以对超过定长(如 8 个周期)的行为链进行截断或压缩处理.




(a) Layered RunTrust management framework
(a) 分层的 RunTrust 管理框架




(b) Chord-Based trust storage
(b) 基于 Chord 的信任存储

Fig.4 Framework and distributed trust storage of RunTrust
图 4 RunTrust 框架和分布式信任存储

### 5.2 全局信任度评估算法

全局信任度评估依赖于评估主体 $i$ 自身的直接信任度评估和来自其他参与合作节点的间接信任度评估,节点全局信任度评估算法见算法 3.

**算法 3**. 全局信任度评估算法.

***GlobalTrust*(*j*)** {

//输入:目标节点 $j$;

//输出:节点$i$对$j$的全局信任度$\boldsymbol{T}_j$.

(a) 利用本地存储的行为链$\boldsymbol{C}_{i,j}$计算直接信任度$\boldsymbol{DT}_{i,j}$(公式(7)).

(b) 向管理节点 *DHT*($j$)发出请求,请求反馈近期(自上次请求时间开始)内 $j$ 的行为链.

(c) 等待;确认收到足够数量的行为链后,按照算法 1 对获得的所有行为链进行过滤.

(d) 将通过过滤的行为链进行聚合,生成行为链$\boldsymbol{C}_j$.

(e) 按照公式(8)和公式(9)分别计算间接信任度$\boldsymbol{IT}_j$和全局信任度$\boldsymbol{T}_j$;

(f) 返回$\boldsymbol{T}_j$;

}

**定理 4**. 若响应反馈请求的节点集合为$R$,且$k=|R|$,行为链平均有$m$周期,则算法 3 的时间复杂度为$O(k^2m^2)$.

证明:由定理 1~定理 3 可知,算法 3 总的时间复杂度为步骤(a)~步骤(f)各步骤时间的复杂度之和.

$$O(m)+O(1)+O(k(k+m))+O(k^2m^2)+O(1)+O(1)=O(k^2m^2).$$ □

**推论 1**. 若评估对象集合为$D$,且$n=|D|$,则RunTrust的信任评估处理时间复杂度为$O(k^2m^2n)$.

当所有节点的全局信任度计算完毕后,评估主体可以采取不同的信任决策策略选择合作伙伴:一种策略按全局信任度递降排序选取其中最高者;另一种策略以全局信任度为概率进行随机选择.两者在性能上有一定的差别,前者合作成功率较高但影响公平性,且易产生过载;后者可以达到较好的负载均衡和公平性,但合作成功率受影响.在合作完毕后,RunTrust 根据合作结果的收益来调整评估机制的各项参数,以提高自适应能力.

## 6 仿真与分析

仿真系统主要考察RunTrust动态信任管理模型在抵御多种节点动态行为的适应能力、反馈过滤能力以及在信任信息压缩存储方面的性能.我们选择了几种典型的信任评估模型作为参照,包括传统的平均值模型以及EigenTrust[6],PeerTrust[12],DyTrust[14]等,与本文的RunTrust模型进行比较,评价它们在对抗恶意行为方面的性能差异.

仿真系统基于P2P文件共享应用环境,其中,提供者(provider)节点贡献自身存储的文件资源供其他节点下载,下载完成后,消费者(consumer)节点就资源质量对提供者节点的服务能力进行信任评价.仿真中假设节点合作是一个参数为 10s的Possion过程,即平均每隔 10s有一个合作发生,且节点间相互独立.另外,假设在所有$N$个节点中恶意节点的比例为$r_m$,根据第 4 节对节点恶意行为的分类,在仿真中我们着重模拟了独立直接动态恶意行为、独立间接动态恶意行为以及勾结间接动态恶意行为,它们在所有恶意节点中分别占有 50%,30%和 20%.在计算节点信任度公式中涉及的收益率、损失率和折扣因子等参数见表 1.

仿真程序基于 Eclipse Java 实现,运行环境为 PIV 3.0GHz,512MB.每项仿真均执行 5 次并取其平均值.

**Table 1** Configuration of parameters in our simulation system

**表 1** 仿真系统参数设置

| Description | Symbol | Default |
|---|---|---|
| # of peers | $N$ | 1 000 |
| % of feedback | $M$ | 20% |
| % of malicious peers | $r_m$ | 50% |
| # of collusion groups | $n_g$ | 20 |
| Rate of income | $\alpha$ | 0.03 |
| Rate of loss | $\beta$ | 0.07 |
| Discount factor | $\rho$ | 0.80 |
| Filter threshold | $\gamma$ | 0.10 |

### 6.1 信任信息的存储性能比较

在此项仿真中,主要展示信任数据存储与传输性能在采用 RunTrust 模型后获得的改善.所有模型中交易记录大致分为两部分,即直接交易记录和间接交易记录.通常,前者数据量较小且存储在本地,因而存储和传输开销都较低,我们不考虑这方面的开销;间接交易记录是除评估主体外其他节点获得的经验,且一般存储在管理节点中,因而这部分开销是主要方面.EigenTrust 模型是一种平均值模型,其中仅存储对节点的最新评价,损失了时间因素.在动态 PeerTrust 模型中,采取自适应滑动窗口的方式存储节点间最近的 100 次交易记录,而文献[13]中也存储大量的节点评估记录,虽然能够保证模型的动态性,但存储维护和传输开销也不容忽视.在 RunTrust 模型中,占多数的静态节点的交易记录仅需要一个周期的数据,其存储开销大约 2 倍于单次交易记录(由定义 2),为一常量;对于恶意节点特别是摇摆节点来说,则需要存储多个周期的数据,存储开销略微增加.图 5 中比较了 RunTrust 与 PeerTrust 模型的信任存储开销.

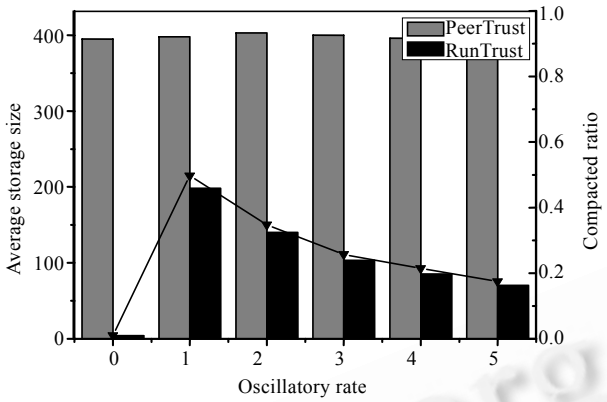



Fig.5 Comparison of trust data storage

图 5 信任数据存储性能比较

我们仿真了 1000s 内、交易时间间隔为 10s 的平均信任数据存储开销情况,其中仅仿真诚实节点、静态恶意节点和独立动态直接恶意节点.另外,在传输中只考虑主要信任数据,协议数据等其他额外数据不考虑.从图 5 可以看出,RunTrust 模型的信任数据存储空间明显低于 PeerTrust,即对信任数据具有良好的压缩效果.并且,不同类型的节点压缩比例也不同,对于静态节点(诚实和纯恶意节点,横坐标为 0)压缩率最高,对于动态恶意节点压缩算法也非常有效,随着节点摇摆比率的升高(横坐标大于 0),压缩率显著下降.另外,因管理节点仅将保存的压缩数据发送给请求者,故也降低了传输开销.

### 6.2 对摇摆行为的敏感程度比较

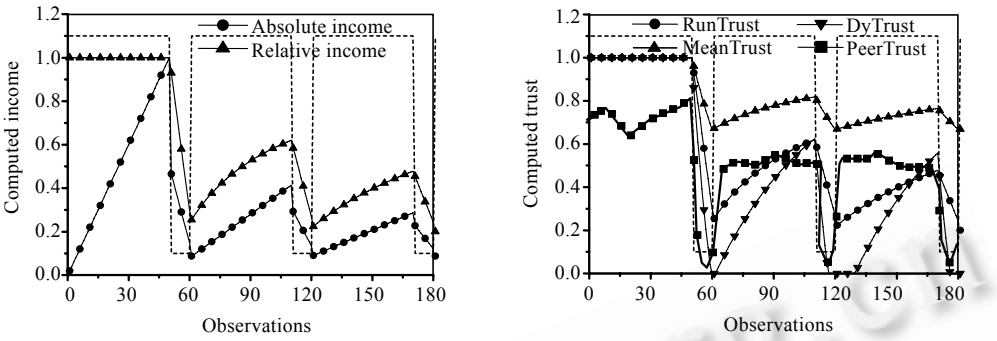
摇摆行为是恶意节点最常表现的行为模式,纵观整个行为记录,恶意节点的行为显现出明显的周期性,即恶意节点先通过良好表现获得正向评价并积累一定的信任度后,再在短期内连续实施恶意攻击.当其信任度下降到一定水平时,它又重新开始新的摇摆周期.我们仿真了正负行为比例为 50:10 的摇摆行为节点 1 000 次观察结果,图 6 显示了 RunTrust 模型中信任值随节点行为周期性变化的情况,对比了 RunTrust 与 PeerTrust,DyTrust 以及平均值法评估摇摆行为节点信任度的能力.

图 6(a)显示了 RunTrust 的信任曲线,反映节点的绝对收益和相对收益(即信任度)的变化情况,其中虚线表示节点行为(设$\alpha$=0.02,$\beta$=0.07,$\rho$=0.8).新加入节点的绝对收益从一个较低的初值开始增加,当其交易次数超过阈值(=10)后,即可获得较高的信任度,而当实施恶意行为时信任度将快速下降.由于有积累速率的限制,因而新节点的信任积累过程相当漫长.相比之下,恶意节点更愿意使用原来的身份,减缓了恶意节点利用新身份攻击的频率.

图 6(b)中仿真了动态直接摇摆行为,它以 50:10 的比例周期性地发起摇摆攻击,节点信任度随攻击行为的发生迅速下降.从图 6 中可以看出,RunTrust 模型与其他模型的信任曲线变化趋势基本相同,对于诚实节点的信任评估结果与其他模型一致.同时,RunTrust 能够准确反映节点的摇摆特征,在相同周期内,RunTrust 模型中信任曲线上升沿的斜率(即信任积累速率)明显要比其他模型平缓,这意味着节点的信任积累更加缓慢;而在信任曲线的下降沿,其斜率则大于其他模型,这说明 RunTrust 对此类恶意节点具有更为严厉的惩罚效果.随着摇摆周期的不断增多,恶意节点要想达到原有的信任水平将更加困难.

### 6.3 参数$\alpha$,$\beta$,$\rho$对信任评估的影响

我们仿真了一个正、负行为比例为 50:10 的摇摆行为节点,在观察 1 000 次结果中其绝对收益和相对收益值的变化.图 7 显示的是不同的收益率、损失率及折扣因子对信任评估的影响.图 7(a)中,我们设置折扣因子$\rho$=0.8,分别比较了两组不同的收益率和损失率情况下的信任曲线;在图 7(b)中设置收益率$\alpha$=0.02,损失率$\beta$=0.07,分别比较了 3 种折扣因子下的信任曲线.

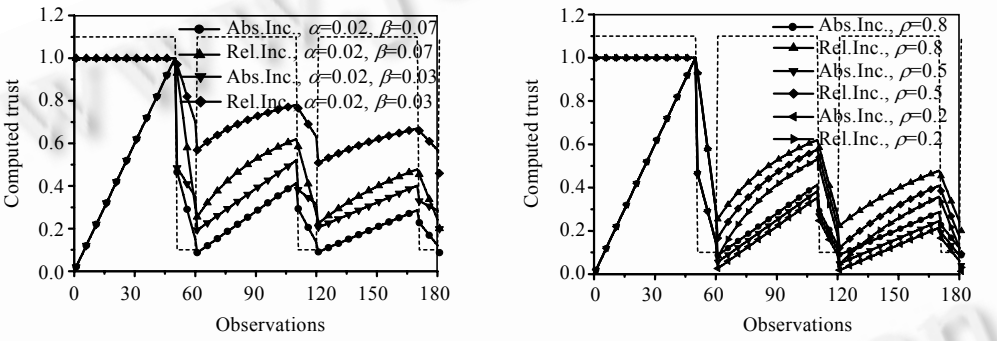



(a) Trust curve of RunTrust
(a) RunTrust 信任曲线

(b) Comparison of many trust models
(b) 多种信任模型比较

Fig.6 Trust evaluation of RunTrust

图 6 RunTrust 的信任度评估




(a) Income rate $\alpha$ and loss rate $\beta$
(a) 收益率$\alpha$ 与损失率$\beta$

(b) Discount factor $\rho$
(b) 折扣因子 $\rho$

Fig.7 Impact of parameters on trust evaluation

图 7 参数对信任评估的影响

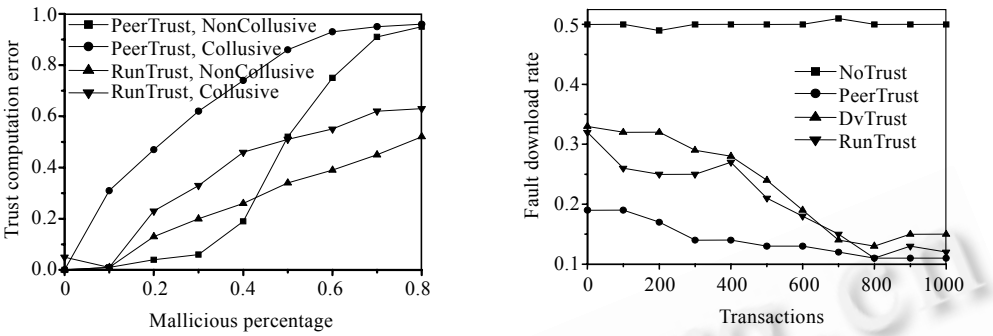
从图 7(a)可以看出,$\alpha/\beta$代表了对负面评估的敏感程度,$\alpha/\beta$越低,说明越敏感.$\alpha/\beta$的比例对绝对收益和相对收益的影响也比较显著,在相同的行为链中,它们随$\alpha/\beta$的升高而增加,随$\alpha/\beta$的下降而降低.图 7(b)显示的是绝对收益和相对收益随折扣因子的增加而提高.但当$\rho$较大时,前期的信任积累对后续周期的信任计算影响较大,体现了折扣因子$\rho$的记忆特性.

### 6.4 恶意反馈过滤能力

我们仿真了恶意反馈中的几种情形,包括静态恶意反馈(夸大和诋毁)、摇摆式恶意反馈及勾结恶意反馈等,实验中设定 3 种恶意节点的比例为 2:1:1,通过执行算法 1 给出的反馈过滤算法,并动态调节反馈过滤阈值$\gamma$,在运行 1000s 内,观察动态调控机制对节点信任度计算误差和错误下载率的影响,其中,信任计算误差判别阈值为±0.5%,若超过,则认为计算错误.

在本实验中,我们分别仿真了节点间独立和勾结两种情形.如图 8(a)所示,信任计算错误率随恶意节点比例增加而升高,并且勾结比无勾结时信任计算误差更严重,说明勾结行为更具隐蔽性且较难发现.与 PeerTrust 相比,RunTrust 更能抑制节点的动态恶意行为,特别是摇摆行为,故后者信任计算误差比前者要低.在图 8(b)中,恶意结点的比例为 20%,比较了 3 种信任模型在无效下载率方面的差异.初始时由于历史记录较少,因而对节点信任度判断失误较多,造成无效下载率较高;但随着经验的增加,并通过反馈过滤阈值的动态调整,错误下载率逐渐

下降.分析图 8 可见,RunTrust 的反馈过滤性能与其他模型较为接近,证明其具有普适性.




(a) Trust computation error (b) Fault download rate

(a) 信任计算误差 (b) 错误下载率

Fig.8 Feedback filter against malicious feedback

图 8 对恶意反馈的过滤能力

## 7 结论和未来工作

本文提出了一种基于行程编码数据压缩理论的动态信任管理模型,采用节点行为记录作为其信任评估的主要依据,并引入时间参数.同时,利用行程编码方法对其进行压缩,不但增加了信任评估依赖的信息容量,而且降低了需要处理的数据量,因而能够得到比现有信任模型更为精确的信任评估结果,对典型的恶意行为也有较强的辨别能力,对恶意的反馈信息有较准确的过滤作用.更为突出的是,RunTrust 通过部分参数的调整,可以转换为其他典型的信任模型,如 EigenTrust 和 PeerTrust,因而具有较强的模拟能力.借助于本模型,可以对其他典型的信任模型进行性能研究和评估.通过实验仿真发现,该模型的信任评估精确度和动态适应性达到或超过了目前典型的动态信任模型.但是,信任数据压缩存储技术的研究仍处于探索阶段,行程编码的描述能力也比较有限,本文仅是在此方向上的初步尝试.未来应当研究复杂结构信任数据的新型压缩编码方式,还需要考虑节点间的时间同步问题,研究更为有效的反馈过滤算法,以提高信任模型精确评估的能力,更好地为 P2P 网络应用提供可信支持.

**References**:

[1] Marsh SP. Formalising trust as a computational concept [Ph.D. Thesis]. Scotland: University of Stirling, 1994.

[2] Aberer K, Despotovic Z. Managing trust in a peer-to-peer information system. In: Proc. of the 10th Int'l Conf. on Information and Knowledge Management (CIKM 2001). New York: ACM Press, 2001. 310−317. http://www.comp.nus.edu.sg/~ooibc/courses/cs6203/managingtrust.pdf

[3] Despotovic Z, Aberer K. Possibilities for managing trust in P2P networks. EPFL IC/2004/84. Lausanne: Swiss Federal Institute of Technology (EPFL), 2004. http://infoscience.epfl.ch/record/52675/files/IC_TECH_REPORT_200484.pdf

[4] Ernesto D, Sabrina DC, Stefano P, Pierangela S, Fabio V. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In: Proc. of the 9th ACM Conf. on Computer and Communications Security. Washington: ACM Portal Press, 2002. 207−216. http://www.comp.nus.edu.sg/~ooibc/courses/cs6203/reputation-based-approach.pdf

[5] Wang Y, Vassileva J. Trust and reputation model in peer-to-peer networks. In: Proc. of the 3rd IEEE Int'l Conf. on Peer-to-Peer Computing. Linköping: IEEE Computer Society, 2003. 150−158. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.62.3463& rep=rep1&type=pdf

[6] Sepandar DK, Mario TS, Hector GM. The EigenTrust algorithm for reputation management in P2P networks. In: Proc. of the 12th Int'l Conf. on World Wide Web. Budapest: ACM Press, 2003. 640−651. http://www-users.itlabs.umn.edu/classes/Fall-2007/csci8980-os/papers/eigentrust-www03.pdf

[7] Khambatti M, Dasgupta P, Ryu KD. A role-based trust model for peer-to-peer communities and dynamic coalitions. In: Proc. of the 2nd IEEE Int'l Information Assurance Workshop. Charlotte: IEEE Computer Society, 2004. 141−154. http://www.khambatti.com/mujtaba/ArticlesAndPapers/khambattim_communities.pdf

[8] Elofson G. Developing trust with intelligent agents: An exploratory study. In: Proc. of the 1st Int'l Workshop on Trust. 1998. 125−139.

[9] Jonker C M, Treur J. Formal analysis of models for the dynamics of trust based on experiences. In: Proc. of the 9th European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW'99). 1999. 221−231.

[10] Li XY, Gui XL. Research on dynamic trust model for large scale distributed environment. Journal of Software, 2007,18(6): 1510−1521 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/1510.htm

[11] Dou W, Wang HM, Jia Y, Zou P. A recommendation-based peer-to-peer trust model. Journal of Software, 2004,15(4):571−583 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15/571.htm

[12] Xiong L, Liu L. PeerTrust: Supporting reputation-based trust in peer-to-peer communities. IEEE Trans. on Data and Knowledge Engineering, Special Issue on Peer-to-Peer Based Data Management, 2004,16(7):843−857.

[13] Duma C, Shahmehri N, Caronni G. Dynamic trust metrics for peer-to-peer systems. In: Proc. of the 2nd Int'l Workshop on P2P Data Management, Security and Trust (PDMST 2005). 2005. 776−781. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1508367

[14] Chang JS, Wang HM, Yin G. DyTrust: A time-frame based dynamic trust model for P2P systems. Chinese Journal of Computers, 2006,29(8):1301−1306 (in Chinese with English abstract).

[15] Yu HF, Kaminsky M, Gibbons PB, Flaxman A. SybilGuard: Defending against sybil attacks via social networks. In: Proc. of the ACM SIGCOMM 2006. http://www.comp.nus.edu.sg/~yuhf/sybilguard-sigcomm06.pdf

**附中文参考文献**:

[10] 李小勇,桂小林.大规模分布式环境下动态信任模型研究.软件学报,2007,18(6):1510−1521. http://www.jos.org.cn/1000-9825/18/1510.htm

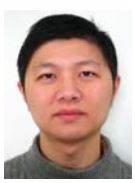
[11] 窦文,王怀民,贾焰,邹鹏.构造基于推荐的 Peer-to-Peer 环境下的 Trust 模型.软件学报,2004,15(4):571−583. http://www.jos.org.cn/1000-9825/15/571.htm

[14] 常俊胜,王怀民,尹刚.DyTrust:一种 P2P 系统中基于时间帧的动态信任模型.计算机学报,2006,29(8):1301−1306.

**方群**(1972－),男,安徽寿县人,博士生,副教授,主要研究领域为分布式信任与安全,可信网络.

**赵生慧**(1970－),女,博士生,副教授,CCF 学生会员,主要研究领域为分布式计算与 Web 服务.

**吉逸**(1957－),女,教授,主要研究领域为计算机网络及应用.



**吴鹏**(1979－),男,博士生,主要研究领域为 P2P 网络,信任管理.

**吴国新**(1956－),男,教授,博士生导师,主要研究领域为分布式计算,网络安全,信息化关键支撑技术,网络性能评价.