ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.7, July 2009, pp.1909−1920
doi: 10.3724/SP.J.1001.2009.03284

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

一种对等网中基于相互信任的两层信任模型*

金 瑜[1+], 古志民[1], 顾进广[2], 赵红武[2]

[1](北京理工大学 计算机科学技术学院,北京 100081)

[2](武汉科技大学 计算机科学与技术学院,湖北 武汉 430081)

Two-Level Trust Model Based on Mutual Trust in Peer-to-Peer Networks

JIN Yu[1+], GU Zhi-Min[1], GU Jin-Guang[2], ZHAO Hong-Wu[2]

[1](School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China)

[2](College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430081,China)

+ Corresponding author: E-mail: wustjy@263.net

Jin Y, Gu ZM, Gu JG, Zhao HW. Two-Level trust model based on mutual trust in peer-to-peer networks. *Journal of Software*, 2009,20(7):1909−1920. http://www.jos.org.cn/1000-9825/3284.htm

Abstract: The reputation model is one of the most important methods that can be used to construct trust between peers in peer-to-peer systems. However, almost all reputation models for P2P applications are purely decentralized. They have many defects such as slow convergence speed of trust in node, complicated trust management and overwhelming network cost. So to solve these problems TLT (two-level trust), a two-level trust model, is proposed in this paper. In TLT a series of trust clusters are spontaneously formed that are the minimum unit of trust evaluation. Every trust cluster includes some members and a cluster header. There is a mutual trust relationship between the cluster header and member node. For example, in order to increase inter-cluster service trust the cluster header checks the service performance of members and eliminates malicious members by using the concept of intra-cluster service trust; while member nodes, aiming to heighten the service reputation and receive good quality services, also examine the management capability of the cluster header and isolate the malicious cluster headers by employing the concept of proxy trust. Analyses and simulations show that malicious behaviors can be quickly identified in TLT because of the fast convergence speed of trust value and TLT is scalable because of its simple trust management and small network overhead.

Key words: peer-to-peer network; trust; reputation; two-level; security

摘 要: 在P2P系统中,声誉模型是建立节点间信任关系的重要方法之一,但现有的P2P声誉模型几乎都是纯分散式的,具有信任收敛慢、信任管理复杂和网络开销大等缺点.在TLT(two-level trust)中,节点自发组织为信任簇,信任评价以簇为单位.每个簇由簇首和成员节点组成,簇首和成员节点之间是一种相互信任的关系:簇首为了提高自身的簇间服务信任,利用簇内服务信任观察成员节点的服务性能,过滤恶意的成员节点;成员节点为了提高服务声誉和接受更好的服务,利用代理信任考察簇首的管理能力.分析和仿真结果表明:在TLT中,节点的信任值收敛快,恶意行为

* Supported by the Ministry of Education-Intel Special Research Foundation for Information Technology under Grant No.MOE-INTEL-08-10 (国家教育部-英特尔信息技术专项科研基金)

Received 2007-08-28; Accepted 2008-02-20

能够被快速识别;TLT 可扩展性好,如信任管理简单和网络开销小.

关键词: 对等网络;信任;声誉;两层;安全
中图法分类号: TP393 文献标识码: A

目前,大多数 P2P 系统在设计时都忽略了安全性需求[1],假定系统中所有参加者都按照协议正常工作,但由于 P2P 系统的开放性和匿名性,这个假定是不现实的.恶意用户可以在系统中散布虚假、伪劣,甚至是恶意的内容和服务[2],而不用担心受到惩罚,如 P2P 文件共享系统中的 VBS.Gnutella 蠕虫病毒和木马[3];KaZaA 系统中超过 50%以上的流行歌曲文件受到了污染[4]等.利用基于声誉[5]的信任机制,可以缓解上述问题.

基于声誉的信任系统大致可以分为两类:集中式和纯分散式[6].在集中式系统里,有一个信任管理中心负责收集、计算和发布节点的信任信息.信任管理非常简单,节点的信任值收敛速度快,但集中式系统必然伴随着单点失败和性能瓶颈等问题;在纯分散系统里,不存在集中的信任服务器,节点相互合作来实现信任管理.因此,信任管理要比集中式的系统复杂得多,节点的信任值收敛也慢[7].目前,几乎所有 P2P 应用中的基于声誉的信任模型都是纯分散式的[8−13],最有代表性的是 P2PRep[8].P2PRep 具有简单、安全的优点,并能以很小的开销集成到 Gnutella 0.4 协议中.P2PRep 的缺点也很明显:1) 系统中的声誉投票完全分散在各个节点上,并且声誉投票是现时计算,换言之,获取一个节点的声誉信息要经过漫长的搜索过程,甚至拥有该信息的节点可能在搜索边界以外或者下线了,因此节点的信任值收敛较慢;2) 系统中所有节点的角色相同,没有考虑节点之间的异构性;3) 可扩展性差,信任管理以单个节点为单位,当系统规模扩大时,信任管理会变得非常复杂;共享文档和声誉查询采用的都是广播机制,网络开销非常大.

针对 P2PRep 的缺点,我们结合集中式和纯分散式声誉模型的优点,提出了一种部分分散(具体指两层)的信任模型(two-level trust,简称 TLT).TLT 模型主要针对无结构 P2P 应用.在 TLT 里,一些能力强的节点创造新的信任簇并自荐为簇首(类似于文献[14]中的 Ultrapeer 和文献[15]中的 Supernode),其余节点加入信任簇,成为这些簇的成员.簇首为了自身利益,集中管理成员节点.TLT 中的两层信任主要是指簇内信任和簇间信任,簇内信任是指簇首与成员节点之间的相互信任关系;簇间信任则是簇与簇之间的信任关系.簇内信任是内层信任,而簇间信任是外层信任.簇内信任是为簇间信任服务的,也就是说,管理簇内信任的目的是为了提高簇间信任.

综上所述,我们可以看出:在 TLT 中,簇首具有重要地位,簇首是否可信是非常关键的问题.然而,目前簇首选择主要是依据硬件条件,如 CPU 周期、内存和网络带宽等[14,15],没有考虑簇首的行为属性.这种簇首选择方法是随机的,即成员节点可以加入任何满足这些硬件条件的簇首,也就是说,假定所有簇首都是可信的.因此,随机的簇首选择算法是静态的,不能反映簇首的行为特性,不能处理簇首恶意的情况.因此,随机的簇首选择算法具有很大的安全隐患:恶意节点可以通过提高硬件条件而成为簇首,从而对系统实施更大的破坏[16].基于此,我们在 TLT 中提出了基于信任的簇首选择算法,成员节点依据自身观察,考察簇首行为.本文的主要贡献包括:

1) 提出了一种两层信任模型.该模型具有如下优点:由于以信任簇为单位,信任管理简单;簇首集中管理成员节点的服务信任,信任值收敛快,网络开销小;该模型下簇首相互合作管理所有成员节点的信任,不存在单点失败和性能瓶颈问题.

2) 提出了基于信任的簇首选择算法.成员节点利用代理信任管理簇首,动态选择可信的簇首加入,提高了服务声誉.

3) 提出了相互信任的概念.簇首与成员节点之间是一种相互信任的关系.簇首利用簇内服务信任的概念管理成员节点;成员节点利用代理信任这个概念选择可信簇首.

本文第 1 节描述相关工作.第 2 节详细介绍我们的模型.性能评价在第 3 节.第 4 节进行总结并提出未来研究方向.

1 相关工作

EBay[17]是一种典型的集中式声誉系统.每次交易以后,买卖双方都要向回馈论坛递交 rating,表明对本次交

易的感受.EBay 使用声誉这个概念来预计用户将来的性能以及帮助他们选择交易对象.并且,EBay 使用了一个集中的服务器管理 peers 之间的通信与交互,从根本上排除了纯分散声誉系统的复杂性,但集中式的声誉系统对 P2P 系统来说是不合适的.

P2PRep[8]是纯分散式声誉系统的典型代表,适用于无结构的 Gnutella 系统.在 P2PRep 协议中,peers 通过一个分布式轮询协议来跟踪和共享其他 peers 的声誉信息.P2PRep 非常简单且安全性高,很容易集成到 Gnutella 0.4 文件共享协议.由于 P2PRep 需要 peers 相互合作来管理其他节点的声誉信息,信任管理复杂;并且,节点的声誉投票分散在整个系统中,信任值收敛慢;再者,声誉轮询是采用洪泛机制进行发布的,网络开销较大.文献[9]只是在 P2PRep 的基础上提出了一个资源声誉的概念,其他方面均与 P2PRep 相同.因此也具有与 P2PRep 类似的缺点.

在适用于结构化 P2P 应用的声誉系统里,节点(我们称其为子节点)的共享文档索引和声誉信息都由其他节点(我们称其为母节点)来保存,并且母节点还要代表子节点响应查询.PeerTrust[10]是针对结构化 P2P 应用——P-Grid[18],而提出来的声誉管理系统.PeerTrust 使用了交易量、交易满意量等几个基本指标来计算节点的信任值,简单、有效,但它具有很大的风险性,因为如果恶意节点想要攻击某个节点的声誉,它只需实行 1 次计算就可以找到该节点的母节点.并且,PeerTrust 紧耦合于 P-Grid 体系结构,很难扩展到其他 P2P 系统.文献[11]则是在 PeerTrust 的基础上,提出了一种动态信任模型,能够解决恶意节点摇摆行为、合谋等攻击.

EigenTrust[19]也是适用于结构化 P2P 应用(如 Chord[20])的纯分散式声誉管理系统.在 EigenTrust 里,每个节点具有全局信任值和局部信任值,全局信任值是唯一的,而局部信任值有多个.凡使用过某个节点服务的节点都保存有该节点的局部信任值,而该节点的全局信任值是通过对这些局部信任值的迭代而得到的.EigenTrust 系统的可扩展性差,表现在:当系统规模变大时,全局信任值收敛非常慢;每个节点需要存储所有其他节点的全局信任值,这很难在大规模系统中得以实现.EigenTrust 还需要有预前可信节点存在,这在 P2P 系统中也是不可行的.文献[21]试图提出一种新的全局信任模型,改善 EigenTrust 中的全局信任值的收敛问题,并且还解决了 EigenTrust 中没有考虑的诸如冒名、诋毁和协同欺诈等安全问题.在以上这些结构化声誉系统中,都没有考虑母节点恶意的情况,如母节点可以给出错误的响应和声誉信息等,对子节点进行攻击.

2 TLT

2.1 TLT概述

TLT 包括两种信任关系,即服务信任和代理信任.服务信任又分为簇内服务信任和簇间服务信任;代理信任又分为代理需求信任和代理响应信任.代理信任与簇内服务信任都属于簇内信任,是内层信任,而簇间服务信任是外层信任.服务信任由簇首管理,其中成员节点的簇内服务信任由簇首集中管理,而簇间服务信任则是纯分散管理.由于每个簇作为整体向外簇提供服务,因此成员节点的服务成果直接影响本簇的簇间服务信任.在 TLT 里,簇首是簇的代表,也是该簇的簇间服务信任的拥有者,即簇首是用管理付出的代价来换取簇间服务声誉的.为了得到更高的簇间服务信任值,簇首通过簇内服务信任过滤恶意的成员节点.代理信任则是由成员节点管理.成员节点入簇的目的是享受好服务,提高服务声誉,因此,它们可以使用代理信任来选择可信的簇首.从以上描述可以看出,簇首和成员节点是一种相互信任的关系,可以实现双向选择.

2.2 基于信任的簇首选择

簇首的主要任务是代替成员节点发出查询,从多个具有响应资格的成员节点中选择合适的一个代表本簇去响应查询等.因此,恶意的簇首可能随意丢弃成员节点的服务请求,或者选择恶意节点为该成员节点提供服务;恶意簇首也可能胡乱地代替成员节点响应服务请求,使成员节点被要求提供根本不可能提供的服务,即对成员节点进行 DoS 攻击,损坏成员节点的服务声誉.为了保障自身利益,成员节点必须对簇首建立代理信任关系,选择可信簇首.

2.2.1 代理信任计算

假定一个成员节点(如 N_j)向簇首(如 N_i)发出服务请求,N_i 代替 N_j 将此请求发送到系统中,然后从所有的响应中,选择一个合适的来向 N_j 提供服务.服务完毕后,N_j 根据本次的服务结果,对 N_i 的代理需求信任进行如下更新:

$$PRqT(N_j,N_i)=\frac{M_{Rq}}{N_{Rq}+C_{Rq}} \tag{1}$$

其中,$PRqT(N_j,N_i)$表示 N_j 对 N_i 的代理需求信任;N_{Rq} 是簇首 N_i 代替成员节点 N_j 向系统发送服务请求的总次数;M_{Rq} 则是其中成功请求的总次数(成功请求是指获得了满意服务的请求).C_{Rq} 为一个比较小的正常量(一般取 10 以内的数),用来调整 $PRqT(N_j,N_i)$的值.当 N_{Rq} 较小时,不论 M_{Rq} 是多少,使用 C_{Rq} 能够保证 $PRqT(N_j,N_i)$小于 1;当 N_{Rq} 足够大时,$PRqT(N_j,N_i)$的值就接近于成功需求占总需求的比例.

假定成员节点 N_j 被请求提供一个服务,很显然,是簇首 N_i 代替 N_j 对这个服务请求进行了响应,因为在我们的两层信任模型里,服务请求只能由簇首处理.N_j 依据这个服务请求的具体内容,对 N_i 的代理响应信任进行如下更新:

$$PRsT(N_j,N_i)=\frac{M_{Rs}}{N_{Rs}+C_{Rs}} \tag{2}$$

其中,$PRsT(N_j,N_i)$表示 N_j 对 N_i 的代理响应信任;N_{Rs} 是簇首 N_i 代替成员节点 N_j 响应服务请求的总次数;M_{Rs} 则是诚实响应的总次数.诚实响应是指真实和正确的响应,即本地拥有被请求的资源或服务,真正能够响应该服务请求.C_{Rs} 的取值及用途与 C_{Rq} 相同,这里不再赘述.

成员节点对簇首的代理信任是代理需求和响应信任的综合,即只要它们其中之一发生了变化,则代理信任就要进行如下更新:

$$IPT(N_j,N_i)=\frac{r\times N_{Rq}}{N_{Rs}+N_{Rq}}\times PRqT(N_j,N_i)+\frac{N_{Rs}-(r-1)\times N_{Rq}}{N_{Rs}+N_{Rq}}\times PRsT(N_j,N_i) \tag{3}$$

其中,$IPT(N_j,N_i)$是 N_j 对 N_i 的综合代理信任.如式(3)所示,在计算综合代理信任时,代理需求信任和代理响应信任具有不同的权重.其中,代理需求信任的权重是增加权重因子 r 与代理需求的总次数占总的代理任务的比例的乘积.代理响应信任的权重则为 1.0 与代理需求信任权重之差.顾名思义,使用 r 的目的是增加代理需求信任的权重,即增加代理需求信任的变化对综合代理信任的影响程度.在我们的模型中,$1.0\prec r\prec\left(1+\frac{N_{Rs}}{N_{Rq}}\right)(\text{if } N_{Rq}\neq 0)$,它能够保证成员节点收到满意服务以后,快速地增加对簇首的代理信任,因为对于成员节点来说,享受好服务是最重要的.如果 $IPT(N_j,N_i)$小于一个给定的阈值 T_θ,N_j 则会退出该簇,重新寻找可信簇首加入.

2.2.2 可信簇首请求

每个成员节点有 3 个可信簇首列表:CTCHL(current trusted cluster header list),BTCHL(backuping trusted cluster header list)和 RTCHL(recommended trusted cluster header list).CTCHL 存储当前可信簇首,BTCHL 存储备用的簇首,RTCHL 是用于推荐的可信簇首.RTCHL 是 CTCHL 的子集,该成员节点同时加入了 RTCHL 里的所有簇首,并且这些簇首还可以继续接纳新的成员节点.

当 CTCHL 列表的长度小于给定的阈值 $T_{\min}$ 并且 BTCHL 为空,即成员节点需要加入新簇,但本地可信簇首缓存没有备用簇首时,该节点就要发出可信簇首需求消息 TCHLR(trusted cluster header list request),请求其他节点推荐可信簇首.TCHLR 主要包括 3 个字段:*N*,*TTL* 和 *FN*.*FN* 是指转发消息节点.*N* 是希望接收者推荐可信簇首的个数.一般 *N* 的初值为 CTCHL 列表最大容量的整数倍.只要某个节点给出可信簇首推荐,那么当该节点再次转发 TCHLR 时,*N* 值就会减少刚推荐的可信簇首的个数.*TTL* 字段与 Gnutella[14]查询消息的 *TTL* 字段意义相同,取值也相同,都为 7.在我们的模型里,TCHLR 消息在系统中的存活时间是由 *TTL* 和 *N* 字段联合决定的.即,只要它们其中的任意一个减为 0,则 TCHLR 消息就要被丢弃.因此,我们的模型采用了受限的洪泛机制来分布 TCHLR 消息.

2.2.3 可信簇首推荐

当成员节点(如 N_j)收到 TCHLR 消息时,N_j 首先查看本地的 RTCHL 的长度是否满足请求推荐的可信簇首个数.如果 RTCHL 的长度大于 TCHLR 消息的 N 字段,则表明 N_j 可以独自推荐所需的可信簇首,给出推荐以后,N_j 将 TCHLR 消息的 N 字段减为 0;如果 RTCHL 的长度小于 TCHLR 消息的 N 字段,则表明 N_j 不能独自满足需求,必须联合其他连接来共同完成推荐任务.可信的簇首推荐消息的格式为{RN,$Length$,$index_1$,N_{x1},$IPT(RN,N_{x1})$,…,$index_i$,N_{xi},$IPT(RN,N_{xi})$…},其中,RN 指的是推荐节点,$Length$ 字段取值为 $RTCHL$ 的长度,$index_i$ 表示当前推荐的可信簇首在推荐列表中的序号,N_{xi} 表示推荐的可信簇首,$IPT(RN,N_{xi})$意义见等式(3).为了转发可信簇首列表请求,N_j 对 TCHLR 消息进行复制,假设共复制了 m 份,$TCHLR_i$ 表示第 i 个复制消息,$TCHLR_0$ 表示 N_j 初始接收并处理的 TCHLR 消息.m 可按下面的等式取值:

$$m=\begin{cases}0 & \text{if } |RTCHL|\geq TCHLR_0.N\\ |CTCHL|-1 & \text{else if } TCHLR_0.FN\in CTCHL\\ |CTCHL| & \text{else}\end{cases} \tag{4}$$

其中,所有 TCHLR 复制消息的 N 字段的和等于本节点未能推荐的可信簇首的个数,于是有下式成立:

$$\sum_{i=1}^{m}TCHLR_i.N=\begin{cases}0, & \text{if } |RTCHL|\geq TCHLR_0.N\\ TCHLR_0.N-|RTCHL| & \text{else}\end{cases} \tag{5}$$

最后,如算法 1 所示,N_j 对这些复制消息的 TTL 和 N 字段重新赋值,将剩余的要求分配给这 m 个复制消息,并分别将它们转发给当前的簇首.

算法 1. SettingTCHLRMessages(*TCHLR*,*x*).

输入:长度为 m 的 TCHLR 消息数组;x.

//x 是本节点不能推荐的可信簇首个数

//TCHLR 为消息数组名

输出:无

```
Begin
    If(m≠0) then
        a=x/m
        b=x%m
        For(i=0;i<b;i++)
            TCHLR[i].N=a+1
            TCHLR[i].TTL--
        Endfor
        For(i=b;i<m;i++)
            TCHLR[i].N=a
            TCHLR[i].TTL--
        Endfor
    Endif
End
```

如果一个簇首(如 N_i)接收到 TCHLR 消息,则它对该消息的处理方式与成员节点不同.在我们的模型里,当成员节点选择可信簇首时,赋予来自其他成员节点的推荐要比簇首的自荐具有更重要的地位.因此簇首一般不会发出自荐消息,只是将该需求全部转发给成员节点来处理,除非它没有成员节点.假设 MNL 是簇首的成员节点列表,存储簇首的当前成员节点.$CHNL$ 是簇首的簇首邻居列表,存储当前的邻居簇首.为了转发可信簇首列表请求,N_i 对 TCHLR 消息进行复制,假设共复制了 n 份,$TCHLR_i$ 表示第 i 个复制消息,$TCHLR_0$ 表示 N_i 初始接收并处理的 TCHLR 消息.n 可按下面的等式取值:

$$n=\begin{cases}0 & \text{if } |MNL|==0 \,\&\, \&TCHLR_0.N==1\\ |CHNL|-1 & \text{else if } TCHLR_0.N\succ 1 \,\&\, \&TCHLR_0.FN\in CHNL\\ |CHNL| & \text{else if } TCHLR_0.FN\notin CHNL\\ |MNL|-1 & \text{if } |MNL|\neq 0 \,\&\, \&TCHLR_0.FN\in MNL\\ |MNL| & \text{else if } TCHLR_0.FN\notin MNL\end{cases} \tag{6}$$

然后,N_i 对这些复制消息的 *TTL* 和 *N* 字段重新赋值,*TTL* 均减 1,TCHLR 复制消息的 *N* 字段的和满足如下等式:

$$\sum_{i=1}^{n}TCHLR_i.N=\begin{cases}TCHLR_0.N-1 & \text{if } |MNL|==0\\ TCHLR_0.N & \text{else}\end{cases} \tag{7}$$

同样地,N_j 按照算法 1,将可信簇首需求分配给这 *n* 个复制消息,并将它们转发给簇首邻居或成员节点.

2.3 簇内服务信任管理

2.3.1 簇内服务信任计算

一个交易结束以后,消费节点就会向簇首汇报 rating,表明对本次交易的满意程度.如果服务节点属于外簇,则消费簇首还必须以本簇的名义给服务提供簇汇报一个簇服务 rating.本文假定节点汇报的 ratings 都是真实、有效的,关于虚假回馈的问题,我们在文献[22]里进行了讨论.Rating 的格式大致是{*CN*,*SN*,*R*},*CN* 字段是指消费节点,*SN* 字段是指服务节点,如果该交易发生在同簇成员节点之间,则 *CN* 是真正的服务消费节点,即成员节点;如果交易发生在不同簇的成员节点之间,则 *CN* 是服务消费节点的簇首.*R* 是新的经验值,表示服务消费者对本次服务的满意程度.取值为 1 时表示服务结果是满意的;为–1 则表示不满意.服务提供节点(如 N_j)的簇首(如 N_i)接收到该 rating 后,对 N_j 的簇内服务信任作如下更新:

$$TV_0(N_i,N_j)=\begin{cases}\mu\times TV_0(N_i,N_j)+(1-\mu)\times R & \text{if } CN\in MNL\\ \sigma\times TV_0(N_i,N_j)+(1-\sigma)\times R & \text{else}\end{cases} \tag{8}$$

其中,$\mu=\dfrac{\alpha}{\omega_{intra\text{-}cluster}}$,$\sigma=\dfrac{\alpha}{\omega_{inter\text{-}cluster}}$ $0\le\alpha\le\omega_{intra\text{-}cluster}\prec\omega_{inter\text{-}cluster}\prec 1$ 且 $\omega_{intra\text{-}cluster}+\omega_{inter\text{-}cluster}=1$.$TV_0(N_i,N_j)$表示累积的 N_i 对 N_j 的簇内服务信任值.α 是经验学习因子.$\omega_{intra\text{-}cluster}$ 和 $\omega_{inter\text{-}cluster}$ 为折扣因子,它们能够影响簇内服务信任的变化速度.如前所述,成员节点对外簇的服务质量直接关系到簇首的簇间服务信任,因此为了自身利益,簇首会根据不同的服务对象,对成员节点采取不同的奖惩制度.$\omega_{intra\text{-}cluster}$ 小于 $\omega_{inter\text{-}cluster}$ 就能保证这一点.例如,成员节点为外簇成员提供了服务,若服务质量好,则簇首的簇间服务信任就会提高;若服务质量较差,则簇首的簇间服务信任就会降低.然而,如果成员节点为本簇成员提供了服务,则无论好坏,影响的只是成员对本簇的信心,没有影响簇首的簇间服务信任.因此对外簇提供了一个好服务后,簇内服务信任上升的程度比为本簇服务的要大;同理,为外簇提供了一个坏服务,簇内服务信任下降的程度也比为本簇服务的要大.这一机制能够鼓励成员节点对外簇提供好服务.

2.3.2 入簇管理

当一个新节点(如 N_j)提出入簇申请时,簇首(如 N_i)就会根据等式(9),评价 N_j 的信任状况.我们假定 N_j 在提出入簇申请之前,已经对 N_i 的代理信任进行了评价,即认为它是可信簇首.现在由 N_i 评价 N_j 的服务信任,决定是否批准其入簇申请.因此,这里又体现了成员节点与簇首之间的相互信任关系.

$$T(N_i,N_j)=\lambda\times T_0(N_i,N_j)+(1-\lambda)\times T_1(N_i,N_j) \tag{9}$$

其中,$T(N_i,N_j)$表示 N_i 对 N_j 的综合信任值;$T_0(N_i,N_j)$代表 N_i 对 N_j 的综合簇内服务信任值;$T_1(N_i,N_j)$为 N_i 对 N_j 的综合簇间服务信任值.如果 $T_1(N_i,N_j)\neq 0$ 表示 N_j 曾经当过簇首.λ和$(1-\lambda)$分别为综合簇内服务信任和簇间服务信任的权重.在 TLT 里,$0\le\lambda\le 0.5$ 意味着与簇内服务信任相比,簇间服务信任更能影响该节点的综合信任状况.因此,这个机制能够鼓励条件好的节点成为簇首.综合簇内服务信任可以按照等式(10)来计算:

$$T_0(N_i,N_j)=\beta\times DT_0(N_i,N_j)+(1-\beta)RV_0(N_j) \tag{10}$$

其中,$DT_0(N_i,N_j)$是 N_i 对 N_j 直接的簇内服务信任;$RV_0(N_j)$是 N_j 的簇内服务声誉,即除 N_i 以外的其他簇首对 N_j 直接的簇内服务信任,换言之,即 $RV_0(N_j)$是 N_i 对 N_j 间接的簇内服务信任.β和$(1-\beta)$分别是直接和间接簇内服务信任的权重.并且 $0.5\le\beta\le 1$,表明簇首更看重自己的直接经历.$DT_0(N_i,N_j)$可以按照等式(11)来计算,其中,*M* 是 N_j 加入

N_i 为簇首的簇的总次数;而 $RV_0(N_j)$的计算公式为等式(12),其中,N 是 N_j 加入除 N_i 以外其他簇首的总次数.

$$DT_0(N_i,N_j)=\frac{\sum TV_0(N_i,N_j)}{M} \tag{11}$$

$$RV_0(N_j)=\frac{\sum_x TV_0(N_x,N_j)}{N}(N_x\neq N_i) \tag{12}$$

$T_1(N_i,N_j)$的计算方法见第 2.4.2 节.经过计算,我们得到了 N_i 对 N_j 的综合的信任值——$T(N_i,N_j)$.如果 $T(N_i,N_j)$不小于一个给定的阈值 T'_θ,则 N_i 就会同意 N_j 的入簇申请,与其建立连接,并赋初始簇内服务信任值.TLT 为了欢迎新节点加入系统,将 T'_θ 取值为本簇成员节点的初始簇内服务信任值.

2.3.3 服务请求处理

当接收到一个服务请求时,簇首为了自身利益,会选择最合适的成员节点响应该请求.若请求服务的节点属于其他簇,则本簇成员的服务结果直接关系到簇首的簇间服务信任;若该请求是本簇成员节点发出的,则成员节点服务质量又影响到请求节点对簇首的代理信任.据此,簇首可以依赖簇内服务信任最高的成员节点.然而,过度依赖簇内服务信任最高的成员节点会导致其过载,从而产生不好的结果,即便是其有良好的愿望提供优质服务.因此,当选择成员节点响应服务请求时,簇首必须在簇内服务信任和负载之间加以权衡.

现在我们假定簇首 N_i 接收到一个文件查询请求 Q,请求共享文档 f;T''_θ 是响应查询的簇内服务信任阈值,能够用来抵制恶意成员节点响应查询请求,并且还能平衡成员节点之间的负载.算法 2 给出了 N_i 处理 Q 消息的过程.

算法 2. ProcessingQuery(Q,MNL, T''_θ).

输入:Q,MNL 和 T''_θ ;

输出:响应节点.

```
Begin
    List S
    null → S      //产生空列表
    if (Ni 本地拥有 f && Ni 没有过载) then
        {Ni}∪S → S
    else
        for(∀Nx∈MNL)
            if (Nx 本地拥有 f && TV0(Ni,Nx)≥T''θ ) then
                {Nx}∪S → S
            endif
        endfor
    endif
    if (|S|>0) then
        return ∀Ny∈S
    else
        return null
    endif
End
```

2.4 簇间服务信任管理

簇间服务信任管理也是由簇首来完成的,主要包括收集簇间的服务 ratings,计算簇间服务信任和评价簇间服务信任,为本簇需求服务的成员节点挑选服务提供者.

2.4.1 簇间服务信任计算

所有成员的服务成果都归簇首拥有,所以,簇首积极收集来自其他簇的服务 rating.在 TLT 里,簇服务 rating 仅仅只是本簇为他簇提供服务的临时凭证,在簇间服务信任评价中是不起作用的.所以必须将这些簇服务 rating 换算成簇间服务信任.假定服务消费簇的簇首为 N_i,提供服务簇的簇首为 N_j,则我们有:

$$TV_1(N_i,N_j)=\frac{X(N_i,N_j)}{Y(N_i,N_j)} \tag{13}$$

其中,$TV_1(N_i,N_j)$表示 N_i 对 N_j 的在一段时间内的簇间服务信任值;$Y(N_i,N_j)$为 N_j 簇的成员节点在该阶段内给 N_i 簇内成员节点提供服务的总的次数,即 N_i 以簇首的名义发送给 N_j 的簇服务 rating 的总个数;而 $X(N_i,N_j)$是这些簇服务 ratings 中 R 为 1 的个数,即满意 ratings 的总数.

2.4.2 簇间服务信任评价

当有多个簇响应本簇的查询时,簇首可以将这些响应全部转交给请求服务的成员节点,由该节点选择服务提供者;也可以代替成员节点选择.TLT 里采用的是第 2 种方式,因为簇首的眼界要比成员节点宽得多,能够识别更多的节点,这也为节点加入好簇提供了一种激励机制.在 TLT 里,为成员节点选择好的服务提供者是簇首的义务,也能够提升该簇首的代理信任.簇首选择服务提供者的标准是响应簇的簇间服务信任,因为簇间服务信任高的簇可以说明该簇内可信的成员节点比例较大,或者该簇的簇首对成员节点的管理比较严格.现在我们假定 N_i 是服务请求簇的簇首,N_j 是响应簇的簇首,则 N_i 对 N_j 的综合簇间服务信任可以如下评价:

$$T_1(N_i,N_j)=\delta\times DT_1(N_i,N_j)+(1-\delta)\times RV_1(N_j) \tag{14}$$

其中,$T_1(N_i,N_j)$是 N_i 对 N_j 的综合簇间服务信任;$DT_1(N_i,N_j)$是 N_i 对 N_j 直接的簇间服务信任;$RV_1(N_j)$是 N_i 对 N_j 间接的簇间服务信任.δ和$(1-\delta)$分别是直接和间接簇间服务信任的权重.并且 $0.5\leq\delta\leq1$,表明簇首更看重自己的直接经验.$DT_1(N_i,N_j)$和 $RV_1(N_j)$的计算方式分别与 $DT_0(N_i,N_j)$和 $RV_0(N_j)$相似,这里不再赘述.

3 实验及结果分析

3.1 实验配置

我们在 PeerSim 1.0 平台上,用 Java 语言实现了 TLT 模型.我们依据 Zipf 定律,初始化了节点拥有的文档以及将来要发出的查询.仿真程序以周期(cycle)为单位进行.在 PeerSim 里,周期是指仿真程序运行的一个时间片段.在一个仿真周期内,PeerSim 仿真调度器会执行一系列控制和协议组件.实验参数由表 1 给出.

Table 1 Simulation parameters configuration

表 1 仿真程序参数配置

Configuration parameter	Value	Configuration parameter	Value	Configuration parameter	Value
Size of the system	1 000	$T_{\min}$	1	$\beta,\delta,\omega_{inter\text{-}cluster}$	0.6
Number of the cluster headers	100	T_θ	0.5	r	1.2
$\lvert CTCHL\rvert$	≤3	T'_θ,T''_θ	0.0	$\lambda,\omega_{intra\text{-}cluster}$	0.4
$\lvert MNL\rvert$	≤30	α	0.3	C_{Rq},C_{Rs}	1.0

3.2 评价指标

我们使用了 3 个指标来评价 TLT 的性能:(1) 平均成功下载率,用 R_{succ} 表示;(2) 声誉系统运行过程中产生的消息总数,用 M_{total} 表示;(3) 平均恶意簇首选择率,用 $R_{malicluster}$ 表示.其中 R_{succ} 是声誉系统最常用的评价指标之一,能够反映信任收敛的速度;而声誉系统的网络开销则一般用 M_{total} 来评价;$R_{malicluster}$ 则是为了说明本文的可信簇首选择算法的有效性而定义的一个新的评价指标,它能够评价系统隔离恶意簇首的效率,从而间接地反映声誉系统的效率.

3.3 隔离恶意簇首

在这个实验里,我们考察了可信簇首选择算法的效率.为了比较目的,我们也实现了随机的簇首选择算法.如图 1 所示,随着仿真的运行,我们的可信簇首选择算法非常有效.例如,在前 40 个周期,$R_{malicluster}$ 从 0.35 快速降到 0.005,然后在如下的周期里稳定为 0.005;而在随机簇首选择算法中,$R_{malicluster}$ 保持不变.从中可以看出,TLT 只需花费很短的时间,就能隔离并孤立恶意簇首,因为当成员节点发现簇首不值得信任时,就会退出该簇,并且从 RTCLH 表中删除该簇首,不再推荐它为可信的簇首;而随机的簇首选择算法没有任何机制识别恶意簇首的行

为,当然就更谈不上隔离恶意簇首了.

以上实验说明了 TLT 在隔离恶意簇首方面表现出来的有效性,但是,可信簇首选择算法的最终目的是降低恶意簇首对成员节点造成的伤害,提高成功交易的概率.因此,本实验使用 R_{succ} 这个标准来衡量恶意簇首对系统造成的影响.如图 2 所示,当恶意簇首增加时,R_{succ} 在两种模型里都在持续降低,但我们模型中的 R_{succ} 要比随机选择算法中的大得多.例如,当恶意簇首占总簇首的比例由 0.1 增加到 0.4 时,在随机选择算法中,R_{succ} 从 0.836 降低到 0.572,减少了 45.96%;而在 TLT 中,R_{succ} 从 0.950 降低到 0.661,减少了 43.66%.因此,采用了可信的簇首选择算法以后,R_{succ} 能够提高 9%~12%.

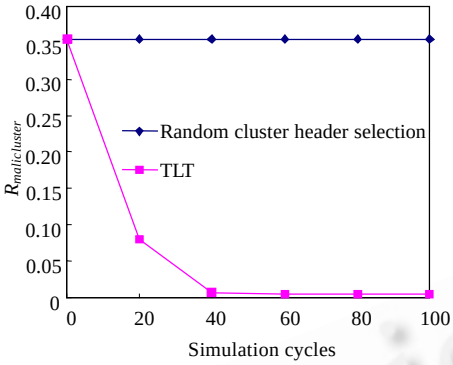

Fig.1 Decrease of $R_{malicluster}$ with simulation cycles

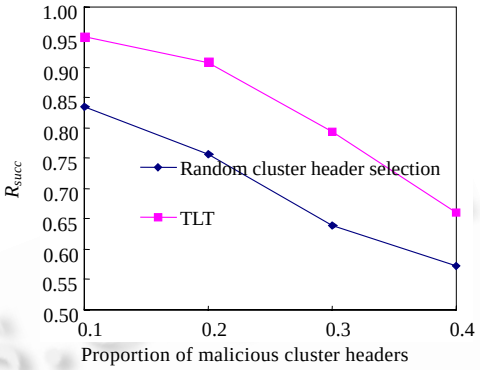

Fig.2 Decrease of R_{succ} as the increase of number of malicious of cluster headers

图 1 $R_{malicluster}$ 随着仿真程序的运行而降低

图 2 R_{succ} 随着恶意簇首的增加而降低

3.4 网络开销

这个实验的目的是考察系统的网络开销.为了叙述方便,我们称采用洪泛机制发布可信簇首需求消息的方法为未受限的洪泛.在这个实验中,我们比较了 TLT、未受限的洪泛和 P2PRep 这 3 个模型所产生的消息量.如图 3 所示,随着仿真的运行,3 个模型中的 M_{total} 都在快速增长,但我们的方法中的消息量最小,P2PRep 次之,未受限的洪泛最大.例如,当仿真程序从第 20 个周期运行到第 100 个周期时,在 TLT 中,M_{total} 从 0.485 增加到 18.54 百万个;在 P2PRep 中,M_{total} 从 1.417 增加到 37.98 百万个;在未受限的洪泛方法中,M_{total} 从 0.891 增加到 72.201 百万个.所以,P2PRep 中的消息量约为 TLT 的 2~3 倍,未受限的洪泛中的 M_{total} 则高达 TLT 的 4 倍.TLT 中的网络开销非常小的原因包括:(1) 成员节点的信任值由簇首集中管理,无须查询;(2) 模型采用受限的洪泛机制在发布可信簇首需求消息时,只要 TCHLR 消息的 *TTL* 和 *N* 字段中任意一个降为 0,这个消息就被丢弃,不再转发.

3.5 攻击模型

在这一节里,我们考察 TLT 对几个常见攻击策略的健壮性和有效性.

攻击模型 A.当恶意节点被选择为服务提供者时,它们依据一定的概率提供恶意服务.在这种攻击模型下,恶意节点的行为具有摇摆性,很难预测.假定 f 是恶意节点提供低质量服务的概率.很明显,f 越大,系统中的平均成功交易的概率就越低.在这个实验中,我们测量了两组数据:当 f 等于 0.5 时的 R_{succ},即恶意节点交替提供高和低质量的服务;f 为 1 时的 R_{succ}, 即恶意节点总是提供恶意服务.如图 4 所示,当恶意节点采取这种攻击模型时,TLT 要比 P2PRep 灵敏得多.当 f 为 1 时,随着仿真程序的运行,R_{succ} 在 P2PRep 中几乎没有变化,而当采用 TLT 模型时,R_{succ} 从 0.757 增加到 0.866.因此与 P2PRep 相比,采取 TLT 模型能够将 R_{succ} 提高 27 个百分点;当 f 为 0.5 时,随着仿真程序的运行,P2PRep 中的 R_{succ} 从 0.979 降低到 0.869,而在 TLT 中,R_{succ} 持续地从 0.826 增加到 0.923.有趣的是:在前 60 个周期,P2PRep 中的 R_{succ} 比 TLT 中的要高,而在余下的周期,出现了相反的情形,原因在于:(1) 与纯分散的 P2PRep 相比,层次的 TLT 中的交易量要大得多,因此在仿真初期,有更多的恶意节点被选择为服务提供者,从而导致较低的 R_{succ};(2) 在 TLT 中,成员节点的服务信任由簇首集中管理,收敛速度快.而 P2PRep 中节

点的信任是纯分散式管理,收敛较慢.因此,随着仿真的运行,簇首快速识别了恶意成员节点,不再选择它们为服务提供者,从而导致了 R_{succ} 的增长.

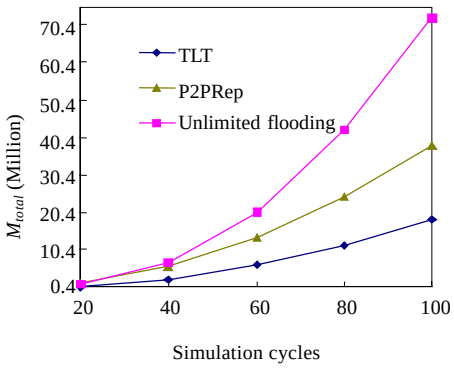

Fig.3 Increase of M_{total} with simulation cycles

图 3 M_{total} 随着仿真程序的运行而增加

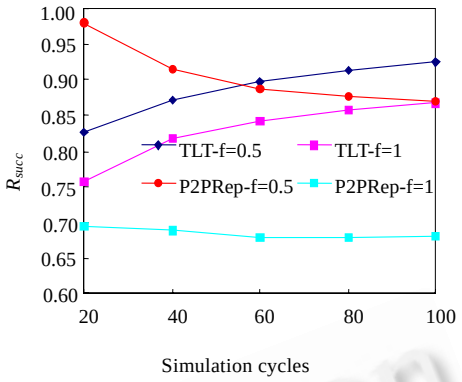

Fig.4 Change of R_{succ} in the attack model A

图 4 R_{succ} 在攻击模型 A 中的变化

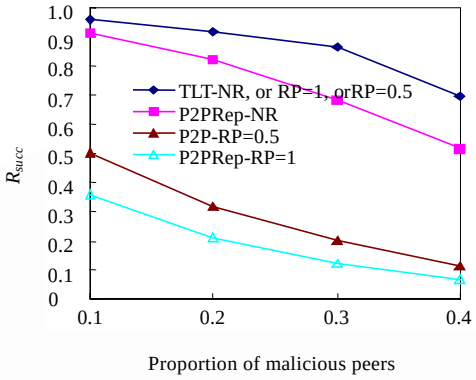

Fig.5 Change of R_{succ} in the attack model B

图 5 R_{succ} 在攻击模型 B 中的变化

攻击模型 B.无论本地是否拥有被查询的文档,恶意节点依据一定的概率响应收到的查询.如果被选择为服务提供者,则提供虚假或恶意文件.恶意节点采用这种攻击模型的目的是为了增加作恶的机会,对系统进行更大程度的破坏.假定 RP 是恶意节点响应查询的概率.显然,RP 越大,R_{succ} 越小.在这个实验里,我们收集了 3 种情况下的 R_{succ}:一种是正常响应(normal response,简称 NR),即恶意节点本地有被查询的文档才响应该查询,只是该文档的质量不好或被污染了;第 2 种是 RP 为 1,即恶意节点对接收到的查询都进行响应;第 3 种是 RP 为 0.5,即恶意节点只对其中的一半进行响应.第 2 种和第 3 种情况都是非正常响应.如图 5 所示,当系统中恶意节点数目增加时,3 种情况下的 R_{succ} 都在下降.与 P2PRep 相比,TLT 中的 R_{succ} 要大得多,并且 R_{succ} 在 3 种情况下是完全相同的.例如,当恶意节点的比例从 0.1 增长到 0.4 时,在 3 种情况下,TLT 中的 R_{succ} 都是从 0.959 下降到 0.693,下降了 27.7%.其原因在于,对查询的响应是由簇首控制的,簇首为了自身利益,总是选择服务性能好的成员节点响应查询,因此,在 TLT 中,恶意节点根本没有机会实施这种攻击模型.然而,在 P2Prep 中,是否响应查询,属于节点的自愿行为,因此这给恶意节点提供了一个实施响应攻击的绝好机会.正如图 5 所示,当恶意节点采取非正常响应时,P2PRep 系统性能急剧恶化.如当 RP 为 0.5 时,R_{succ} 从 0.502 下降到 0.115,减少了 77.1%,而当 RP 为 1 时,R_{succ} 则从 0.360 下降到 0.067,减少了 81.4%.

攻击模型 C.当恶意行为被识别以后,恶意节点退出系统,然后以新的身份再次加入系统,重新寻找机会破坏系统.如图 6 所示,当恶意节点增多时,两种模型中的 R_{succ} 都在下降,然而与 P2PRep 相比,我们模型中的 R_{succ} 明显要高一些.例如,当恶意节点的比例由 0.1 增加为 0.4 时,P2PRep 中的 R_{succ} 从 0.911

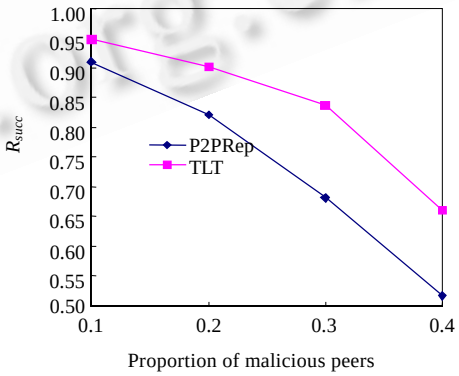

Fig.6 Change of R_{succ} in the attack model C

图 6 R_{succ} 在攻击模型 C 中的变化

下降到 0.516,下降了 43.36%;而在 TLT 中,R_{succ} 从 0.948 571 下降到 0.659 859,减少了 30.44%.因此,采用 TLT 声誉模型,R_{succ} 能够提高 3~14 个百分点.TLT 比 P2PRep 更健壮于这种攻击的原因在于:在 TLT 中,簇首为了自身利益,对新加入节点有一个考察过程,仅赋予其较低的簇内服务信任初值,使得新节点被选择为查询响应者的机率比较小,从而导致被选择为服务提供者的机会要少.因此,在我们的模型中,恶意节点以新身份重新入簇并不能得到更多的实惠;而在 P2PRep 中,响应查询由恶意节点自己决定,因此,恶意节点可以通过多响应查询来提高选择为服务提供者的机会,从而对系统实施更大程度的破坏.

4 结论和未来研究方向

本文提出了一种新的两层信任模型.通过仿真实验和分析说明,我们的模型具有如下优点:(1) 簇首和成员节点之间是相互信任的关系.成员节点可以依据代理信任考察簇首的行为,隔离和孤立恶意簇首,簇首则可以根据簇内服务信任观察成员节点在簇内的服务性能,将恶意成员节点驱逐出簇;(2) 节点的信任值收敛速度快.簇首为了自身利益,集中管理节点的簇内服务信任,能够快速识别恶意成员节点;(3) 声誉管理系统是可扩展的,表现在以下几个方面:簇首相互合作,共同管理成员节点的信任和声誉信息,不存在单点失败和性能瓶颈问题;系统中的最小活动单位是信任簇,而不是单个节点,因而简化了信任管理的复杂性;TLT 不需要查询成员节点的信任信息,并且可信簇首需求也没有采用基本的洪泛,而是采用一种受限的洪泛机制,因此网络开销较小.

TLT 采用了类平均方法来分配可信簇首需求,没有考虑每个连接上的实际推荐能力,因此接下来我们可以考虑采用加权平均方法来获取可信簇首推荐,即本次要求一个连接推荐可信簇首个数与其上次推荐的个数有关.下一步我们还要开发 TLT 模型的原型系统,并将之应用到我们的 P2P Web 缓存系统,在真实的 P2P 环境中评价 TLT 的性能.

References:

[1] Sieka B, Kshemkalyani AD, Singhal M. On the security of polling protocols in peer-to-peer systems. In: Caronni G, ed. Proc. of the IEEE 4th Int'l Conf. on Peer-to-Peer Computing (P2P 2004). Los Alamitos: IEEE Press, 2004. 36−44.

[2] Divac-Krnic L, Ackermann R. Security-Related issues in peer-to-peer networks. In: Steinmetz R, ed. Proc. of the P2P Systems and Applications. Berlin, Heidelberg: Springer-Verlag, 2005. 529−545.

[3] VBS.Gnutella, 2008. http://www.symantec.com/securityresponse/writeup.jsp?docid=2000-1218113-5230-99

[4] Liang J, Kumar R, Xi Y, Ross KW. Pollution in P2P file sharing systems. In: Makki k, ed. Proc. of the IEEE Infocom 2005. Los Alamitos: IEEE Press, 2005. 1174−1185.

[5] Resnick P, Zeckhauser R, Friedman R, Kuwabara K. Reputation systems: Facilitating trust in Internet interactions. Communications of the ACM, 2000,43(12):45−48.

[6] Jsang A, Ismail R, Boyd C. A survey of trust and reputation systems for online service provision. Decision Support Systems, 2005,43(2):618−644.

[7] Curtis N, Safavi-Naini R, Susilo W. X^2Rep: Enhanced trust semantics for the XRep protocol. In: Jakobsson M, ed. Proc. of the 2nd Conf. of Applied Cryptography and Network Security (ACNS 2004). Berlin, Heidelberg: Springer-Verlag, 2004. 205−219.

[8] Cornelli F, Damiani E, di Vimercati SDC. Choosing reputable servents in a P2P network. In: Lassner D, ed. Proc. of the 11th Int'l World Wide Web Conf. (WWW 2002). New York: ACM Press, 2002. 376−386.

[9] Damiani E, di Vimercati SDC, Paraboschi S. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In: Atluri V, ed. Proc. of the 9th ACM Conf. on Computer and Communications Security (CCS 2002). New York: ACM Press, 2002. 207−216.

[10] Xiong L, Liu L. PeerTrust: A trust mechanism for an open peer-to-peer information system. Technical Report, GIT-CC-02-29, Atlanta: Georgia Institute of Technology Press, 2002. 1−26.

[11] Chang JS, Wang HM, Yin G. DyTrust: A time-frame based dynamic trust model for P2P systems. Chinese Journal of Computers, 2006,29(8):1301−1307 (in Chinese with English abstract).

[12] Marti S, Garcia-Molina H. Limited reputation sharing in P2P systems. In: Breese J, ed. Proc. of the 5th ACM Conf. on Electronic Commerce. New York: ACM Press, 2004. 91−101.

[13] Yu B, Singh MP, Sycara K. Developing trust in large-scale peer-to-peer systems. In: Cybenko G, ed. Proc. of the IEEE 1st Symp. on Multi-Agent Security and Survivability. Los Alamitos: IEEE Press, 2004. 1−10.

[14] Lime Wire LLC. Gnutella Protocol Develop. 2008. http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html

[15] KaZaa. 2008. http://www.kazaa.com/us/help/glossary/supernodes.htm

[16] Lo V, Zhou D, Liu Y, GauthierDickey C, Li J. Scalable supernode selection in peer-to-peer overlay networks. In: Diego S, ed. Proc. of the Int'l Workshop on Hot Topics in Peer-to-Peer Systems (Hot-P2P 2005). Los Alamitos: IEEE Press, 2005. 18−25.

[17] EBAY. Ebay feedback forum. 2008. http://pages.ebay.com/services/forum/feedback.html?_trksid=p3907.m36

[18] Aberer K. P-Grid: A self-organizing access structure for P2P information systems. In: Batini C, ed. Proc. of the 9th Int'l Conf. on Cooperative Information Systems (CoopIS 2001). Berlin, Heidelberg: Springer-Verlag, 2001. 179−194.

[19] Kamvar S, Schlosser M, Garcia-Molina H. The EigenTrust algorithm for reputation management in P2P networks. In: Lawrence S, ed. Proc. of the 12th Int'l World Wide Web Conf. (WWW 2003). New York: ACM Press, 2003. 640−651.

[20] Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for Internet applications. ACM SIGCOMM Computer Communication Review, 2001,31(4):149−160.

[21] 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

[22] Jin Y, Gu ZM, Gu JG, Zhao HW. A new reputation-based trust management mechanism against false feedbacks in peer-to-peer systems. In: Benatallah B, ed. Proc. of the 8th Int'l Conf. on Web Information Systems Engineering (WISE 2007). Berlin, Heidelberg: Springer-Verlag, 2007. 62−73.

附中文参考文献:

[11] 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):1301−1307.

[21] 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

金瑜(1973－),女,湖北应城人,博士生,CCF 学生会员,主要研究领域为分布式计算,P2P 信任研究.

顾进广(1974－),男,博士,副教授,CCF 高级会员,主要研究领域为语义网,软件工程.

古志民(1964－),男,博士,教授,博士生导师,主要研究领域为可扩展计算,并行计算.

赵红武(1974－),男,副教授,主要研究领域为分布式数据库.