P2P网络中一种基于定向查询的分布式信誉系统

许笑旻,叶 枰,陶 军
(东南大学 计算机网络和信息集成教育部重点实验室,江苏 南京 210096)

摘 要:P2P网络中的安全问题越来越受到关注,广泛采取的策略是建立信誉系统。提出了一种基于定向查询的分布式P2P信誉系统,节点交互历史表记录与本地节点有交互经验的各节点信息,而本地节点的信誉信息分散存储在交互历史表中各节点上的信誉信息表中。信誉查询时查询节点根据被查询节点交互历史表的信息直接定向到提供被查询节点信誉信息的节点,恶意节点表和表尾信息表保证了信誉信息的安全。仿真实验的结果表明该信誉系统可以有效地减少无效交互的次数,遏制恶意节点对信誉信息的破坏。

关键词:P2P安全;分布式信誉系统;定向查询;恶意节点
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2009)08-0235-05

A Distributed Reputation System Based on Directed Request in P2P Networks

XU Xiao-min, YE Ping, TAO Jun
(Ministry of Edu. Key Lab. of Computer Network & Info. Integration, Southeast Univ., Nanjing 210096, China)

Abstract: With more and more important P2P security concerned, the strategy of reputation systems is widely used. In this paper, a directed request based reputation system is proposed. The NID(Node Interaction History) list records the nodes who have interaction experience with the local node, while the reputation of the local node is distributed stored in the reputation lists of the nodes in NID list. In the process of request serching, the request node can directly get the reputation of requested node according to the record in NID list. The reputation security is guaranteed by the malicious nodes list and the table-tail information list. Simulation results show that the reputation system can effectively reduce the invalid interactions and prevent reputation from the damage of malicious nodes.

Key words: P2P security; distributed reputation system; directed request; malicious node

0 引 言

P2P网络又称对等网,网络中所有节点地位平等,同时扮演着服务器和客户端的角色,这是P2P网络与传统的客户/服务器模式最大的区别。随着P2P技术的不断发展,P2P在商业、科研、军事等领域的应用越来越广泛。现有的P2P研究工作不再仅限于各种P2P应用,也包括了随着P2P的广泛应用带来的安全、负载等问题。

P2P网络具有匿名性、动态性、自治性和异构性,即每个节点自愿加入网络,可以随意进出于网络,可以随意选择自己的标识,不同的节点有不同的能力和可靠性。这些特性一方面体现了P2P网络的灵活性,另一方面也带来了安全方面的问题。P2P网络对于各类攻击缺乏抵抗力,许多调查表明在当今最流行的一些P2P网络中恶意节点的攻击行为已经大量存在,最普遍的攻击是虚假文件攻击,恶意节点响应网络中所有的查询,提供虚假的文件给其他节点下载。

为了解决P2P网络中的安全问题,广泛采取的策略是建立信誉系统。信誉系统帮助交互双方在交互前根据对方过去的行为来判断对方的可靠性,从而决定是否进行此次交互。P2P网络中由于恶意节点的存在,节点在收到一个陌生节点的响应后很难决定是否接受对方的服务。信誉系统可以帮助本地节点根据其他节点与对方交互的经验来判断对方的可信程度,从而决定是否接受服务。在文献[1~5]中证实了信誉系统确实可以使得P2P网络更加安全。

1 现有研究基础

现有的信誉系统主要分为集中式和分布式两类。

收稿日期:2008-12-10;**修回日期**:2009-02-26
基金项目:国家自然科学基金重大研究计划项目(90604003);国家自然科学基金项目(60603067)
作者简介:许笑旻(1983-),男,江苏南通人,硕士研究生,研究方向为Overlay应用和P2P安全;陶 军,副教授,研究方向为网络安全。

集中式信誉系统中信誉信息存储在部分超级节点上，系统实现简单，但超级节点的存在破坏了 P2P 原有的节点平等性。所以目前对于 P2P 信誉系统的研究主要集中在分布式信誉系统。

P2Prep[6]，Xrep[7]等信誉系统提出了一种分散存储，广播式查询的机制。节点的信誉信息分散存储在与之交互过的各个节点上，查询时广播查询信息，由与之交互过的节点响应并反馈对应的信誉值。这类信誉系统实现较简单，但缺陷也很明显。广播式查询效率比较低，网络负载比较大。同时由于查询方式的高负载，在实际情况中每条查询报文的传播跳数被限定，因此某个节点的查询范围相对固定，在拓扑图中总是以自己为中心向周围发散的一个区域，这样导致查询到的信誉信息具有局限性。此外广播式查询中得到节点反馈的信誉信息时并不能知道这个信誉信息是否过时，所以每个节点的反馈在最后的信誉值计算中的影响力相同的，无法体现实时性。而且恶意节点可以在未与对方节点交互过的情况下随意进行评价，这为各种恶意行为提供了很大的漏洞。

PRIDE[8]，R－Chain[9]等信誉系统提出了一种本地存储，单点查询的机制。节点的信誉信息存储在本地，查询时只需要在本节点进行查询操作。此类信誉系统查询效率高，但实际环境中提供服务的节点在系统所有节点中占的比例极小，而本地存储的信誉系统又将信誉信息存储和计算的负载加在这少数节点上，带来负载不均衡的问题，这样对于这些提供服务的节点是不公平的，导致愿意提供服务的节点更加减少。同样恶意行为也很难防范。服务节点恶意删除最新的部分信誉信息，导致查询到的是过去的信誉信息，掩盖了服务节点最近可能的恶意行为。由于信誉信息的本地存储，服务节点能够很容易的在本地发现对自己评价较差的节点，从而进行报复。文献[10,11]则考虑将信誉机制引入到访问控制中。

文中结合了上面两类信誉系统的优点，针对它们各自的缺陷，提出一种基于定向查询的信誉系统。

2 基于定向查询的信誉系统

2.1 系统模型概述

本系统中的信息存储方式结合了分散存储和本地存储，分散存储的是信誉信息，分散存储在与本地节点有交互经验并提供对其评价的各个节点上；本地存储的是节点信息，即这些提供评价的节点的相关信息在本地存储。查询某节点信誉信息时，首先从此节点得到提供评价的节点信息，然后直接查询这些节点得到信誉信息，这就是定向查询的概念。

系统模型如图 1 所示。

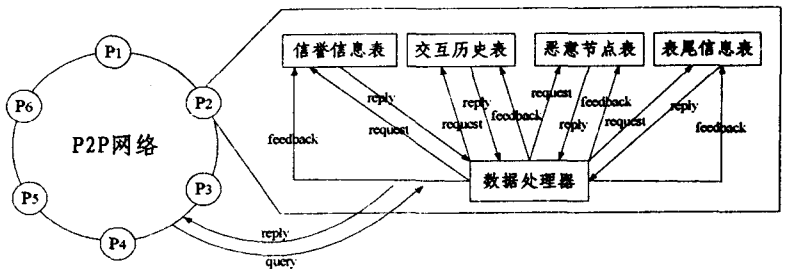

图 1 基于定向查询的信誉系统模型

1)节点标识(NID)。

本信誉系统中的每个节点都拥有一对公私钥来进行签名等安全操作，公私钥对由节点自行生成。系统利用公钥的摘要作为标识，一方面具有可变性，随着公钥的变化而变化；另一方面，变化的代价是相应的信誉值丢失，只能作为新节点加入网络，得到较低的信誉初始值。

2)信誉值模型。

本系统采用简化了的 PeerTrust[8]信誉计算模型：

$$T(w,u) = \alpha * T'(w,u) + \beta * \frac{\sum_{i=1}^{I(w,u)} T(u,i) * P(w,i)}{I(w,u)}$$

，其中 $T(w,u)$ 表示节点 w 对于节点 u 的信誉评价，$T(w,u) \in (0,1)$，$T'(w,u)$ 表示节点 w 对于节点 u 原先的信誉评价，α、β 是权值，$I(w,u)$ 表示节点 w 从节点 u 交互历史表中选择的查询节点数，$P(w,i)$ 表示节点 w 对节点 i 的信任值。如果节点 w 是第一次与节点 u 交互，即节点 w 上原先没有节点 u 的信誉值，这时 $\alpha = 0, \beta = 1$。

3) 信任值模型。

信任值是对于节点提供的信誉信息可信度的评价，目的是防止恶意节点提供不真实的信誉信息。本系统采用较简单的信誉值兼信任值的方法，即 $P(w,i) = T(w,i)$。

4)交互历史表。

交互历史表记录了本地节点作为服务节点时，接受服务并提供评价的用户节点的相关信息，这些信息以链表形式存储。每次交互结束后，服务节点更新本地的交互历史表，将此次交互中的用户节点的相关信息加在链表的尾端。交互历史表中的每个单元包括的内容有：交互标识(TID)，用户节点的 NID，用户节点的签名。TID 用来标记节点提供过的每次服务。用户节点对交互历史表中前一信息单元和本信息单元进行签名，一方面保证了本信息单元无法被撰改，另一方面

保证了服务节点无法在连续的表单元之间插入任何数据。

5)信誉信息表。

信誉信息表是本地节点作为用户节点时,对服务节点评价的集合。每次交互结束后,用户节点对服务节点提供的服务进行评价,更新本地的信誉信息表。表中的每个单元是某个服务节点的信誉信息,包括最新 TID、信誉值和服务节点 NID。最新 TID 是本地节点与此服务节点最近一次交互的 TID,它对于后面讨论的重复查询等问题有帮助。

6)恶意节点表。

恶意节点表中每个单元就是一个恶意节点的信息,包括恶意节点 NID 和时戳。时戳记录节点被加入恶意节点表的时间,用于恶意节点表的定时更新。恶意节点是由一次交互中的不良行为决定的,由于存在着无意的恶意行为,还有部分节点可能改正过去的恶意行为,所以必须对恶意节点表进行定时更新。

7)表尾信息表。

表尾信息表中记录的服务节点都是以本地节点作为其交互历史表表尾的,即服务节点的最新一次服务是提供给本地节点的。表中每个单元包括服务节点 NID 和服务节点私钥加密的 TID。其中加密 TID 作为交互时服务节点提供给用户节点的交互证据。

2.2　系统交互过程

1)用户节点发送交互请求,在回应的服务节点中筛选出不在本地恶意节点表中的节点,发送交互历史表请求。

2)服务节点收到请求,返回本地的交互历史表。交互历史表的规模可能很大,整体传输增加负载,影响效率,本系统中服务节点返回最新的 100 个交互历史表单元。

3)用户节点联系服务节点交互历史表表尾节点,查询其表尾信息表,如此服务节点在表中,证实服务节点提供的用户列表是最新的,继续下一步;否则表明服务节点有欺骗行为,放弃该节点,并将其加入恶意节点表中。

4)用户在服务节点交互历史表中随机选取若干个节点进行信誉值查询(查询前必须确认这些节点不在恶意节点表中,如果在,放弃该节点,重新选取),然后根据信誉值模型计算服务节点的信誉值,本系统中设定阈值 0.5,选择信誉值超过该阈值的服务节点,如有多个,选择信誉值最高的。

交互历史表是以交互次数为单位增长的,所以同一节点可能多次出现在交互历史表的不同单元中,在随机查询过程中可能出现重复查询同一节点的现象。本系统在查询某节点对于服务节点的评价前,比较交互历史表中该节点的 TID 和该节点信誉信息表中此服务节点的最新 TID,如果相同则查询,如果不同,可能发生了多次查询,放弃查询。

5)服务节点将私钥加密的 TID 发送给用户节点,提供了参与本次交互的证据,为用户节点表尾信息表的更新提供信息。用户节点验证收到的加密 TID,如果不是本次交互应有的 TID,将服务节点记录在恶意节点表中,放弃该节点;如果是,继续下一步。

6)服务节点为用户节点提供服务。

7)交互结束后,用户节点评价服务节点,更新自己的信誉信息表。更新完成后,用户节点将服务节点交互历史表更新所需的签名信息发送至服务节点。服务节点更新交互历史表,发送确认信息至用户节点。

8)用户节点收到确认信息后,将服务节点加入到本地表尾信息表中,并将交互证据发送至服务节点交互历史表原表尾节点,对方节点验证后在表尾信息表中删除此服务节点,此次交互过程结束。

如用户节点未收到确认信息,即服务节点否认交互过程,用户节点将服务节点加入本地恶意节点表,并向服务节点交互历史表原表尾节点提供交互证据,对方节点验证后将信誉信息表中该服务节点的信誉值修改为一个较低的值。交互过程结束。

9)如果服务节点在一定时间内未收到用户节点的更新信息,即用户节点拒绝提供评价,将用户节点加入本地恶意节点表,交互过程结束。

一次成功的交互过程如图 2 所示。

2.3　系统解决的问题

1)查询机制的效率问题。本系统采取的定向查询的方法是广播式查询和本地存储单点查询两种机制的结合体,效率较高。

2)信誉信息的安全问题。节点的信誉信息分散存储在与之交互过的用户节点上,无法随意修改。同时签名技术和表尾节点表保证了节点无法在交互历史表中恶意删除或增加任意项。

3)信誉查询的局域性问题。广播式查询的信誉系统中节点查询的范围有限,得到的信誉值具有局域性。本系统中由于每查必中(即交互历史表中的节点必然有服务节点的信誉信息),在同样的消耗下查询到的信息必然比广播式查询多,信誉信息更加完整。

4)信誉系统的负载均衡问题。本系统不让某节点单独承担信誉信息的存储或计算任务,而是采取分散存储和谁想要谁计算的方法,使得整个信誉系统的负载分散在各个节点上。

5)对于攻击的防御。

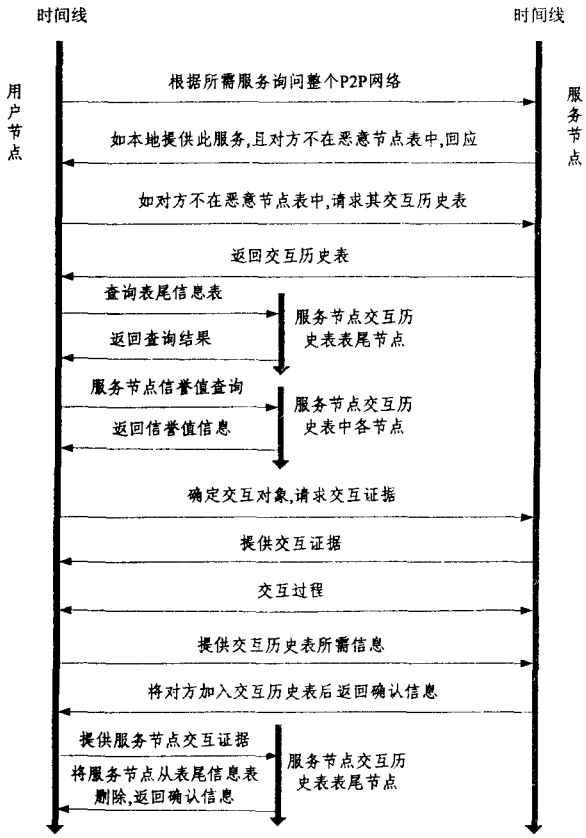

图 2 一次成功的交互过程

否认攻击:利用恶意节点表进行有效的防御。

报复:即服务节点针对给予自己较低评价的节点进行报复,报复方法是给予对方极其差的评价,诋毁对方。本系统中首先规定当本地的信誉信息表被查询时,本地节点必须检查对方节点的 NID 与被查询节点的 NID 是否相同,如果相同查询不被允许。其次,本系统中由于交互历史表的存在,必须交互过才可以提供对对方的评价,不可随意进行报复。

洗刷:即使用新的公私钥和 NID,摆脱原有的较低的信誉值。本系统中采用"冷启动"的方式,即新节点加入时给予较低的初始信誉值,从而保证了洗刷攻击不能得到很好的效果。

勾结:即多个恶意节点形成组织,互相进行好评,以提高信誉。现实环境中,勾结的恶意节点的数量所占节点总数比例极小,而本系统采用随机选取节点进行信誉查询的方法,勾结节点被选中的可能性比较小。

3 仿真评估

本系统的仿真实现选择了 PlanetSim 仿真平台和 Java 语言,保证了仿真代码可以平稳转换为在 Internet 上的实验代码,底层 P2P 路由协议为 Chord[12]。节点数取为 5000 个,假定其中恶意节点在交互中实施恶意行为的概率为 50%,非恶意节点实施恶意行为的概率为 5%。

图 3 的实验中将恶意节点的比例定为 0% ~ 35%,比较了有无本信誉系统这两种情况下,整个网络中的无效交互次数。显然,随着恶意节点数的增加,无信誉系统的网络中无效交互次数也急剧增加,即恶意节点恶意行为成功次数猛增。而有本信誉系统的网络对于恶意行为的防御保持平稳良好的状态。图 4 的实验中比较了两种情况下,随着交互次数的增加,网络中无效交互次数的状态。可以看出,本信誉系统在交互次数大量增加的情况下,依然保持对恶意行为的有效遏制。

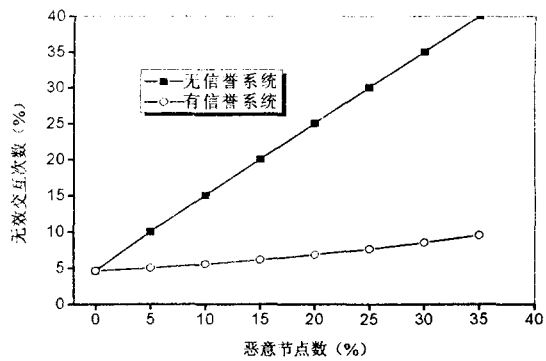

图 3 信誉系统的有无随恶意节点数变化对交互次数的影响

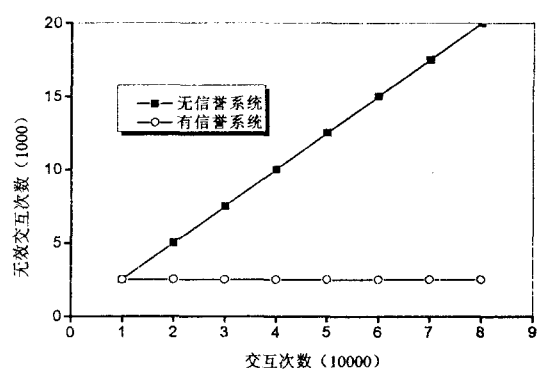

图 4 信誉系统的有无对无效交互的影响

图 5 的实验中比较了本信誉系统与 P2Prep 的负载情况。在交互次数不断增加的情况下,采用广播式查询的 P2Prep 系统中的信誉查询次数也大量增加,负载较大。相比较而言,本系统由于定向查询的优势,信息查询带来的负载有限,适合现实环境的需要。

4 结束语

文中提出了一种基于定向查询的 P2P 信誉系统,即查询节点根据被查询节点交互历史表信息直接定位到目标节点进行信誉信息查询。节点上的恶意节点表给予恶意节点有效的惩罚,表尾信息表防止节点恶意删改交互历史表。系统不依赖于特定 P2P 路由机制,所以适用于大多数 P2P 网络。在仿真实验中,本信誉

系统体现出了对恶意行为的有效遏制。当然,该系统还需要不断补充和完善,如节点离线,交互历史表更新等,这些问题都有待今后进一步研究。

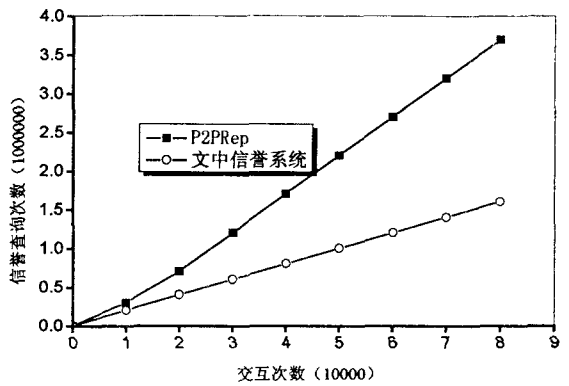

图 5　信誉系统的有无对信誉查询次数的影响

参考文献:

[1] Damiani E, di Vimercati S D C, Paraboschi S, et al. Managing and sharing servents' reputations in p2p systems[J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(4):840-854.

[2] Wang Y. Bayesian network-based trust model in peer-topeer networks[C]//Proceedings of Workshop on Deception, Fraud and Trust in Agent Societies at the Autonomous Agents and Multi Agent Systems 2003 Conference (AAMAS-03). Melbourne, Australia:[s.n.],2003.

[3] 周金洋, 杨寿保, 郭磊涛,等. P2P 网络中一种基于信誉感知的资源发现算法[J]. 小型微型计算机系统,2006,27(10):67-72.

[4] 马新新,耿　技. 对等网络信任和信誉机制研究综述[J]. 计算机应用,2007,27(8):37-42.

[5] 黄全能,宋佳兴, 刘卫东,等. 对等网络信誉机制研究综述[J]. 小型微型计算机系统,2006,27(7):43-50.

[6] Cornelli F, Damiani E, De Capitani S. Choosing reputable servents in a p2p network[C]//In: Proc of the Eleventh International World Wide Web Conference. Honolulu, Hawaii:[s.n.],2002.

[7] Damiani E, De Capitanidi V, Paraboschi D, et al. A reputation-based approach for choosing reliable resources in peer-to-peer networks[C]//In: Proceeding of the 9th ACM conference on Computer and communications security. [s.l.]: ACM Press,2002:207-216.

[8] Dewan P, Dasgupta P. Securiing reputation data in peer-to-peer networks[C]//International Conference on Parallel and Distributed Computing and Systems (PDCS 2004). Cambridge, USA: MIT,2004.

[9] Liu Lintao, Zhang Shu, Ryu Kyung Dong, et al. R-Chain: a self-maintained reputation management system in P2P networks[C]//17th International Conference on Parallel and Disbuted Computing Systems(PDCS-2004). San Francisco, CA, USA:[s.n.],2004.

[10] 张国治,党小超,魏伟一. 基于信任域的 P2P 访问控制模型研究[J]. 计算机技术与发展,2006,16(8):228-230.

[11] 李玲娟,姬同亮,王汝传. 一种基于信任机制的混合式 P2P 模型[J]. 计算机应用, 2006,26(12):22-25.

[12] Stoica I, Morris R, Karger D, et al. Chord: A Scalable Peer-to-peer Lookup Service for Internet Application[C]//In Proceeding of ACMSIGCOMM,2001. San Diego:[s.n.],2001.

(上接第 234 页)

能瓶颈,提供了更好的健壮性和灵活性。但是下列问题需要在以后的工作中加以改进:1)由于移动 Agent 通常都是由管理站产生,这一特点使得本系统还不能完全解决管理站瓶颈的问题;2)域内定位服务器的不确定性使得每个节点都有可能提供定位服务,增加了节点的开销。

参考文献:

[1] 张　鹏. 基于移动代理的网管系统的研究[D]. 西安:西安交通大学,2001.

[2] 曹　阳,陶　舒,尹建华,等. 基于移动 Agent 的分布式网管系统设计与实现[J]. 武汉大学学报:自然科学版,2000,46(3):297-300.

[3] 何　力. 移动代理技术在网络管理中的应用研究和部分实现[D]. 西安:西安电子科技大学,2003.

[4] Greenberg M S, Byington J C, Harper D G. Mobile agents and security[J]. IEEE Communication Magazine, 1998, 36(7):76-85.

[5] Stefano D, Bello L. Naming and locating mobile agents in an Internet environment [C]//Enterprise Distributed Object Computing Conference, EDOC'99. Madrid, Spain:[s.n.] 1999:153-161.

[6] 陈云芳,王汝传,王海艳. 移动代理位置透明性研究[J]. 南京邮电大学学报:自然科学版,2007,27(4):73-79.

[7] 黄烟波,余　鹰. 基于移动代理的分布式网络管理系统的研究[J]. 微计算机信息,2006,22(33):154-156.

[8] Fokus G. Mobile Agent System Interoperability Facilities Specification [S]. OMG TC Document orbos, Geneva, Switzerland:[s.n.],1997.

[9] Wang Ying-Hong, Keh Huan-Chao. A Hierarchical Dynamic Monitoring Mechanism for Mobile Agent Location[C]// Proceedings of the 19th International Conference on Advanced Information Networking and Applications(AINA'05). Tahiti: IEEE Computer Society,2005.