

对等网络中基于团体的分布式路由协议中的可靠路由规则^{*}

周世杰 秦志光 刘锦德

(电子科技大学计算机科学与工程学院 成都610054)

摘要 分析了对等计算的基本概念,对当前对等计算中流行的查找和路由协议进行了概括性的分析。提出了一个基于团体的分布式路由协议(CSDR),并讨论了该协议的基本概念。对CSDR中的可靠性路由规则(DRR)进行了详细分析和论述,并给出了具体的协议和算法。

关键词 对等计算, 分布式计算, 路由协议, 团体

The Dependable Routing Rule in the Community Sensitive Distributed Routing Protocol for Peer-to-Peer Network

ZHOU Shi-Jie QIN Zhi-Guang LIU Jin-De

(College of Computer Science and Engineering University of Electronic Science and Technology of China, Chengdu 610054)

Abstract Peer-to-Peer (P2P) employs distributed computing technology to achieve resource sharing directly between peers. P2P is also considered to be an alternative to distributed computing and is similar to Grid computing. The core of P2P is its naming approach and routing algorithm. So this paper will concentrate on analyzing and discussing this special aspect of P2P system. The routing architecture of the past and current P2P system can fall into two classes: un-structured and structured. The current routing protocol used in these popular P2P system is discussed. Then, this paper presents a community sensitive distributed routing protocol (CSDR) for P2P computing. Compared with other P2P routing and naming protocol. CSDR takes into account the application requirement. In CSDR, the P2P network consists of many locally dependent communities. Each peer belongs to at least one community. The peers which fall into more than one community is called multi-role peers (MP). Through these MP peers, the message can be routed among many communities. The overview of CSDR is addressed firstly. Then, its dependable routing rule (DRR) is also designed and analyzed. In DRR, the neighbor routing table (NT) is employed to achieve determination property, which allow request is delivered reliably.

Keywords Peer-to-Peer, Distributed computing, Routing protocol, Community sensitive

1 对等网络

对等计算具有两个特征。其一是信息的交流在两个或者多个具有计算能力的节点(Peer)之间直接进行,不需要集中控制机制来保证网络可靠持久运行。其二是参与对等网络通信的节点是网络边界设备(Edge Device),即这些对等网络中的节点分布在网络中,形成了一个逻辑上完全独立的、全分布式计算环境。上述两个特征意味着对等网络必须依赖现有的网络基础设施来提供服务,即对等网络构建在现有的Internet网络框架基础之上,但提供了与现有Client/server模型不同的网络服务。对等网络通过节点地位的相对等同性,在解决现有网络服务存在单点失效的问题的同时,也面临着与分布式计算相同的难题:如何进行资源的有效存储、发现和访问。与其他分布式计算技术不同,对等计算通过对路由和查找的集成,来考虑资源的管理问题。为了讨论方便,除特殊说明之外,以下的路由协议均指路由与查找协议。

本文在分析现有P2P路由协议的基础之上,提出了一个与问题领域相关的路由查找协议:基于团体的分布式路由协议(Community Sensitive Distributed Routing Protocol for Peer-to-Peer Network,简称CSDR),该协议通过与应用紧密结合,可提高对等网络中路由与查找的效率。文章第2节讨论

和分析现有的P2P路由协议,第3节详细分析和设计基于团体的分布式路由协议,第4节对相关工作进行分析之后,最后对全文进行了总结。

2 对等网络中的分布式路由协议

对等网络的路由与查找协议位于现有路由协议之上(如P2P的命名系统不依赖现有的域名系统)。依据不同的P2P网络结构,现有的P2P路由协议可分为结构化路由协议(Structured Routing Protocols)和非结构化路由协议(un-Structured Routing Protocols)。结构化路由协议与结构化的P2P网络(Structured P2P)相对应。在结构化P2P网络中,节点之间按照一定的方式组织起来,构成一个逻辑上独立的网络。在这类系统中,某些节点可能担当路由消息传递者的角色,其他节点则可依赖这些中间节点进行信息交换。如果中间节点集中部署在一个地方,则形成了集中索引P2P网络(Central Index P2P)。Napster是集中索引的典型系统之一。在该类系统中,一个节点维护网络资源的位置信息,其他节点只能通过该中心节点查找该信息。在获得资源信息之后,节点之间直接进行资源传送,无需通过中间节点。如果集中索引服务由多个分布式的节点来提供,则属于另外一种类型的结构化P2P网络:分布式索引(Distributed Index P2P)。分布式索引

^{*} 四川省科技厅项目(编号:01GG0712);国家863项目(编号:2002AA142040)。周世杰 博士研究生,主要研究方向:P2P,信息安全技术,中间件技术。秦志光 博士,教授,博士生导师,主要研究方向:信息安全技术,ITS技术。刘锦德 教授,博士生导师,主要研究方向为开放分布式系统与中间件技术,移动代理技术。

引 P2P 利用多个节点提供负载均衡和内容分发服务,因而在一定程度上缓解了单个节点所带来的服务瓶颈问题。流行的网络搜索引擎 Google 是这类系统的典型代表。如果所有节点角色平等,则形成了另外一种重要的结构化 P2P 网络:分布式 Hash 路由表(Distributed Hash Table Routing)。以 Chord, Tapestry, Pastry 和 CAN 为代表的该类系统,赋予每一资源一个全局标识符,并采用分布式 Hash 算法,将该资源存储到相应的结点上。当需要查询资源,只需要按照相应的规则,即可访问该资源。

与结构化 P2P 网络相比,非结构化 P2P 中节点和资源之间则没有必然的联系:资源分布存储在整个网络中,任何节点都有可能参与一次查找操作中的消息传递,也有可能应答该查找请求(如果存储有该资源或相关信息)。以 Gnutella 系统为代表的非结构化 P2P 系统一般采用“泛洪”式路由协议来查找资源,即当需要执行查找请求时,节点通过消息广播方式向自己的“邻居”节点发送请求,如果收到请求的节点可满足该请求,则返回相应的应答消息,否则继续向自己的邻居广播该请求。该过程直到请求得到满足或者通过的节点跳数(Hops)达到一定的阈值为止。

结构化 P2P 网络显著优点之一是查询所需要的网络资源(网络带宽)开销小,查询延迟小,其缺点是用于维护节点之间关系的资源开销较大(除集中索引系统)。如在 Chord 等分布式 Hash 路由中,为了保证当节点动态加入/退出系统时路由表的准确性,必须依靠额外的协议来更新路由表,因此尽管 Chord 查找消息仅需要 $\log_2 N$ 的路由开销,但节点加入或退出却需要 $O(\log_2 N)^2$ 的路由开销,因而对动态的网络,用于路由维护的开销较大。此外,这类系统也缺乏按关键字查询的能力。集中式索引系统虽然提供了按关键字查询的功能,但由于存在单点故障和服务瓶颈问题,难以应用到大规模分布式 P2P 应用中。与此相反,非结构化 P2P 网络的优点是简单,无需额外的路由维护协议,其缺点是查询开销大,在大规模网络中的资源浪费难以接受。以 Gnutella 为例,当每个节点可同时向 7 个邻居节点广播请求,请求允许到达的最多跳数为 5,则传递的消息数为 $\sum_{i=1}^5 7^i = 19607$ 。虽然分析结果表明在跳数为 4~5 时查询结果已经比较令人满意,但这样的网络资源开销在现有的网络基础设施条件下仍难以接受。

为了提高查找的效率,某些系统提供了结构化 P2P 和非结构化 P2P 之间的折中方案:混合式 P2P (Hybrid P2P Routing)。在这类系统中,一些节点被赋予更多的责任来担当消息传递者(Relay)的角色。这些“中间人”形成了一个逻辑上的结构化 P2P 路由网络(类似于目前 Internet 中的路由器),可对消息进行路由转发。其他节点之间则通过这个更高层次的逻辑网络进行消息传递。由于在整个网络中随机分布的节点仍然居多,因此从整体来看,混合 P2P 系统也呈现出非结构化的特点。如 JXTA 系统中的 search hub 和 Gnutella I (第二代 Gnutella, 简称 G2) 中的 superpeer 等均充当了中间人的角色,因而这些系统可归入混合 P2P 系统。混合 P2P 通过维护一定的网络结构,使之在保持非结构化的优点的同时,尽量减少查询开销,因此这种折中方案具有一定的优越性。但是,在这样的 P2P 网络中,超级节点显得至关重要,如果这些节点出现通信故障,则将影响整个系统的运行。

3 基本的基于团体的分布式路由协议

3.1 概述

基于团体的分布式路由协议的现实依据在于,用户对资

源的需求“因人而异”。比如一个对数学理论有着浓厚兴趣的人,可能更关注与数学相关的资源;而对喜欢体育的人,可能更加关注新闻信息。此外,即使同一个人也可能希望探索多个领域的资源,但这些资源从本质上看也可以进行归类。因此,按照生物学中“物以类聚,人以群居”的原理,有着共同兴趣爱好的人易建立相互关系。人们这种心理需求的差异,在 P2P 网络中则体现为对资源需求的差异。通过对爆炸式资源的重组,有可能简化人们对信息资源的存储、查找和使用,从而提高资源的利用率。对等网络中的基于团体的分布式路由协议(CSDR)的目标之一既是满足人们对不同资源的需要,并提供简便、可靠、可信的资源存储与访问手段,从而有效改善对资源的利用率。

与其他 P2P 系统不同,CSDR 中的节点在存储资源时,尽可能选择对该资源感兴趣的节点来存放。只有在不能找到这样的节点或需要提高资源的可用性时,才考虑将该资源存储到其他节点上。为了实现这一目的,需要对资源和节点进行“归类”(Grouping)。CSDR 对资源的归类体现在节点的归类中。节点(Peer, p)由两个部分组成:节点团体和节点标识符。节点团体(Peer Community, pc)是节点期望归属的群体标识,描述了用户感兴趣的问题领域。节点标识符(Peer Identifier, pi)则是该节点在整个系统中的唯一标志,用以区别其他节点。根据节点团体,可方便节点找到兴趣相同的节点,从而组成一个逻辑上的兴趣组,有效地进行组间信息交流、资源共享等。而节点标识则是该节点在整个网络中的身份象征,其作用类似于 TCP/IP 网络中的 IP 地址。因此,任意节点可表示为: $p = (pc, pi)$ 。此外,为了讨论方便,我们将节点 p 属于团体 PC_x 表示为 $p \in PC_x$ 或 $p.pc = PC_x$,而节点 p 的节点标识为 pi_x 表示为 $p.pi = pi_x$ 。CSDR 网络中所有节点用 P 表示,所有团体用 PC 表示,所有节点标识用 PI 表示,则 $P = PC \times PI$ 。团体 x 中的所有节点表示用 $x.PI$ 表示。对于资源,我们认为它从本质上没有分类的区别,因此可以用全局唯一的资源标识 RI (Resource Identifier)来表示。

衡量分布式路由算法协议性能的指标主要包括时间复杂度和消息复杂度。时间复杂度指一次完整路由的时间开销。消息复杂度指一次完整路由中,消息传递所经过的通信链路数(Hops)。

由于 P2P 网络将路由与查找操作整合为一体,因此设计路由算法时必须同时考虑对资源的有效访问。根据对资源和节点的定义,我们也可以对 P2P 网络路由协议进行形式化的定义。分布式对等网络路由协议 PR 可以用一个三元组来表示: $PR = \langle RI, P, R \rangle$, RI 和 P 分别是网络中的资源和存储资源的节点, $R = \bigcup_{i=1}^x R_i$ 是一组路由规则(Rules)。因此设计一个 PR , 实际就是确定 RI, P 的取值,以及设计一组 RI 与 P 二者之间的映射规则。

3.2 基本协议分析与设计

节点标识 PI 可用 n 位(n bites)长度的节点标识环空间(Peer Identifier Circle)来表示,可以对节点的 IP 地址进行 Hash 运算生成(比如 SHA-1)。而节点团体 PC 则用具有 m 位(m bites, 一般而言 $m \leq n$)长度的团体标识环(Community Identifier Circle)上的唯一标识来表示,节点在加入 P2P 网络时初始化该信息,以后可按照自己的需要重新修改设置。CSDR 中所有节点组成一个 $n \times m$ 的二维圆环(Tori)。任意节点改变所属团体,在 (PC, PI) 组成的二位坐标系中则表现为节点的“直线运动”;节点在自己节点标识所对应横线上的“自我复制/消失”。“复制”对应节点将自己添加到另外的团体

的操作,“消失”则对应节点退出一个团体的操作。图1是一个3×2的CSDR网络(读者应理解为一个二维圆环)。其中在PC₂这个团体中有三个节点(2,5,6),而节点2则属于两个团体:PC₁和PC₂。每个节点至少加入一个团体后才能进入CSDR网络(本文随后段落将讨论节点如何加入CSDR网络)。由此可见,一个节点可以加入一个或者多个团体,一个团体可以包含一个或者多个节点。

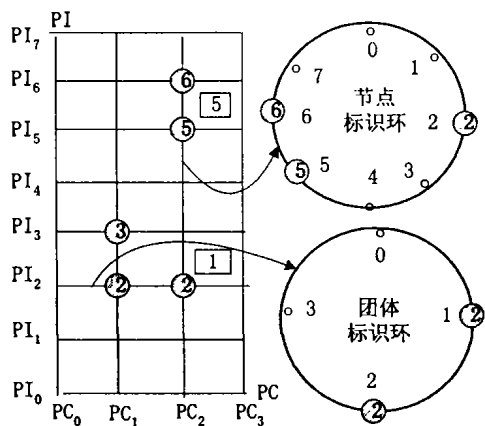


图1 一个团体中的节点及标识

在CSDR中,资源标识RI可视为n位(n bites)长度的资源标识环空间(Resource Identifier Circle),每一资源可用该标识空间上的一个标识RI来唯一表征,RI可以对资源的内容或其中的关键字用与节点标识相同的Hash函数来生成,因此RI实际是资源的索引值。通过对资源标识的分析,使得人们可以从浩瀚的信息海洋中提炼出对自己真正有益的信息来,从而减少查询资源的时间和精力开销。资源分配采用与分布式Hash表(DHT)相似的方法:RI总是分配到指定团体PC_x中节点标识PI_y与RI相等或者接近的节点。如图1中,标识为5的资源存放在PC₂团体的节点PI₅上,而标识为1的资源则存放在节点PI₂上。

3.3 可靠路由规则(DRR)设计

为了传递消息,CSDR网络中的任意节点p需要维护一个路由表,该路由表包括所有的邻居节点的信息。在CSDR网络中,如果同一团体内两个节点之间不存在中间节点,则该节点互为邻居节点。这种只包含邻居节点的路由表我们称之为邻居路由表(Neighbor Table,简称NT表)。图2描述了图1所示CSDR网络的团体路由表的组成。NT表中每个表项包含两部分:团体标识和节点标识列表。团体标识表明节点在哪个团体中,节点标识则指明了具体的邻居节点的信息。在图2中,节点2,5是节点6的邻居节点,节点6,2是节点5的邻居节点,节点5,6是节点2的邻居节点(图2中没有包含后面将要讨论的保留团体和保留邻居节点)。

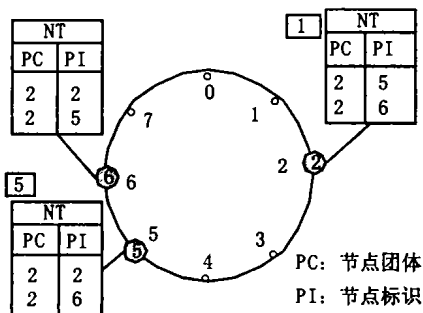


图2 节点的邻居路由表示意图

节点PC_x查询资源RI_y时,需提供期望的该资源所在的团体标识PC_y,节点按照图3所示算法来确定路由。路由过程简单明了:PC_x在自己的NT表中查找与PC_y属于同一个团体的节点s,如果s上存储了RI_y,则s就是目的节点,查找过程结束;如果s上没有RI_y的信息,s重复PC_x的路由过程,直到找到资源RI_y,整个路由过程结束。

```
//The Dependable Routing Algorithm of peer x in
//CSDR.NTx is the neighbor table of peer x
//Input:NTx,PCy,RIy
//Output:Path
DRRoute(NTx,PCy,RIy)
s=SelectPeerFromNT(PCy,NTx)
if s <> NULL then
  Append the s to Path
  if RIy is in s then return Path
  else DRRoute (PCy,NTs)
return Path
//End of Route operation
SelectPeerFromNT(PCy,NTx)
if s in NTx || s.PC=PCy || s.PC()PC0 then return s
if s in NTx || s.PC=PC0 then return s
//End of SelectPeerFromNT
```

图3 CSDR中的可靠路由算法(DRR)

以图4为例(图中为节省篇幅,没有标出保留邻居节点信息),PC=1的节点希望查找属于团体PC_y=2的资源RI_x=6,则通过查询自己的NT表,节点1知道邻居PC=2的节点属于期望的团体,请求被传递到节点2;由于节点2没有该资源的相关信息,因此通过查询自己的NT表,将请求传递到自己的邻居节点5或6(图中表明节点2选择了节点5作为下一路由的节点);同样,节点5将请求传递给节点6,该节点存储了资源RI_x的信息,请求得到满足,路由过程结束,最终得到的路由路径为:1→2→5→6。显然,通过简单路由算法得到的路径是随机路径,因而有可能不是最佳路径,还需对算法进行优化,本文后续章节将讨论该问题。

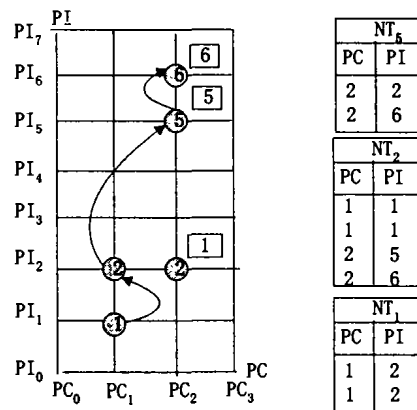


图4 CSDR简单路由协议示意图 (PC_x=1,PC_y=PC₂,RI_y=6)

此外,还存在“孤岛节点”问题:如果一个节点仅属于一个团体,且该团体中不包含属于其他团体的节点,则这样的节点称之为孤岛节点(Island Peer),相应的团体叫做“孤岛团体”(Island Community)。如果CSDR系统中存在孤岛节点,通过其NT表只能查找团体内的资源,无法查询团体外的资源,即无法与其他团体中的节点通信。CSDR可采用“保留团体”(Preserve Community)来解决这个问题:PC₀这个团体是系统保留团体,任何节点均是该团体的一员(讨论孤岛节点和孤岛团体时,不考虑保留团体的存在),保留团体中的邻居节点叫做保留邻居节点,相应的路由表称之为保留邻居路由表(Preserve Neighbor Routing Table,简称PNT表)。除非特别

说明,保留邻居节点和团体内的邻居节点统称为邻居节点。DRR 算法中的邻居路由表也包括保留邻居节点。那些孤岛节点可以通过与保留团体中自己的邻居节点通信,确保 SelectPeerFromNT 操作返回一个确定值,从而保证 DRR 算法的正确性和可靠性。只要查询的资源在 CSDR 网络中,使用简单路由算法可准确定位该资源,因此该算法可称之为“可靠路由规则”(Dependable Route Rule, DRR)。节点只有在不能找到其他的路由路径时,才需要考虑使用保留团体进行消息传递,因为这有可能降低协议的性能。考虑图5中 $PC=0$ 的节点希望查找属于团体 $PC_2=2$ 的资源 $RI_x=5$,通过保留团体进行通信得到的路径如图5所示。

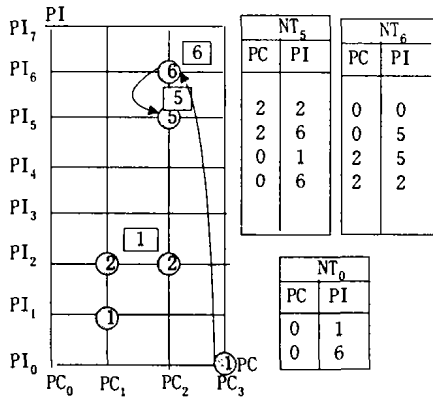


图5 CSDR 通过保留团体进行通信的示意图($PC_x=0$, $PC_y=PC_2$, $RI_x=6$)

4 相关工作

在文[16,27]中,对 P2P 的基本概念进行了详细分析和论述,文[27]对各种 P2P 系统和基于 P2P 的应用进行了分类和总结。文[24,25]则对分布式 Hash 表路由协议(DHT)所面临的各种问题进行了分析,对 DHT 中的安全需求也进行了概述性论述。文[22,26]分析了对等计算中的“局部性”现象,从而启迪我们可以利用 P2P 进行数据存储系统的开发。在安全对等计算方面,文[25,30]从总体方面进行了分析,但没有提出具体实现的方法和技术。文[31,34]对基于 P2P 的信息存储技术进行综述,但没有涉及如何具体实施等问题。国外相关的研究项目包括 Chord^[5,6], PAST/Pastry^[9-11], CAN^[12], Tapestry^[7]等,这些项目和相应的系统均采用类似 DHT 的路由算法,因此与 CSDR 有许多相似的地方。但与之相比,CSDR 更加注重资源的有效访问。

总结 基于团体的分布式路由协议(CSDR)充分结合应用需求,将一个对等网络划分为多个逻辑上独立的团体,提高了资源的查找效率。团体中多角色节点允许请求跨团体传递。此外,节点仅需依据本地的邻居路由表即可进行可靠消息路由。但是,如何创建和维护邻居路由表,还需要进一步研究,这也是本文进一步的研究工作之一。此外,从协议性能角度来看,DRR 规则远未达到最优路由的要求,因此,也需要对协议进行优化,从而减少消息的通信开销,并提高协议的性能。

参考文献

- 1 Napster Home Page. <http://www.napster.com/>
- 2 Gnutella. <http://gnutella.wego.com>
- 3 The Gnutella protocol specification, 2000. <http://dss.clip.com/GnutellaProtocol04.pdf>
- 4 Clark I, Sandberg O, Wiley B, Hong T. Freenet: A distributed anonymous information storage and retrieval system. In: Proc. of the Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, July 2000. 311~320

- 5 Stoica I, Morris R, Karger D, et al. Chord: A scalable peer-to-peer lookup service for internet applications. In: Proc. of ACM SIGCOMM, San Diego, California, Aug. 2001. 149~160
- 6 Dabek F, et al. Building Peer-to-Peer Systems With Chord, a Distributed Lookup Service. In: Proc. of the 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII), Schloss Elmau, Germany, May 2001
- 7 Zhao B, Kubiawicz J, Joseph A. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. [Tech. Rep. UCB/CSD-01-1141]. Computer Science Division, U. C. Berkeley, Apr. 2001
- 8 Plaxton C, Rajaraman R, Richa A. Accessing nearby copies of replicated objects in a distributed environment. In: Proc. of ACM SPAA, June 1997
- 9 Rowstron A, Druschel P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: Proc. of the 18th IFIP/ACM Intl. Conf. on Distributed Systems Platforms (Middleware 2001) Nov. 2001
- 10 Rowstron A, Druschel P. Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility. In: Proc. of the 18th ACM SOSP (Ban., Canada, Oct. 2001), pp. 188~201
- 11 Druschel P, Rowstron A. PAST: A large-scale, persistent peer-to-peer storage utility. In: Proc. of HOTOS Conf., 2001
- 12 Ratnasamy S, Francis P, Handley M, Karp R, Shenker S. A scalable content-addressable network. In: Proc. of ACM SIGCOMM, San Diego, California, Aug. 2001. 161~172
- 13 Dabek F, Kaashoek M F, Karger D, Morris R, Stoica I. Widearea cooperative storage with CFS. In: Proc. of the 18th ACM SOSP Banff, Canada, Oct. 2001. 202~215
- 14 Kubiawicz J, et al. Oceanstore: An architecture for global-scale persistent storage. In: Proc. of ASPLOS. ACM, Nov. 2000
- 15 Rhea S, Wells C, Eaton P, et al. Maintenance-free global storage in OceanStore. Submission to IEEE Internet Computing, 2001
- 16 Yang B, Garcia-Molina H. Comparing Hybrid Peer-to-Peer Systems. [Technical report]. Stanford University, February 2001. Available at: <http://dbpubs.stanford.edu/pub/2000-35>
- 17 Matthew J, Robshaw B. MD2, MD4, MD5, SHA and other hash functions. [Technical Report TR-101]. RSA Laboratories, 1995. version 4. 0
- 18 SHA-1. 1997. American National Standards Institute, American National Standard X9.30.2-1997: Public Key Cryptography for the Financial Services Industry- Part 2: The Secure Hash Algorithm (SHA-1)
- 19 Buyya R. Economic models for Management of the Resources in Peer-to-Peer and Grid Computing. In: Proc. of the Commercial Applications for High-Performance Computing Conf. 2001
- 20 JXTA 2001. The JXTA home page <http://www.jxta.org>
- 21 Leigh P, Benyola P. Future Developments in Peer Networking. Equity Research, White Paper, Raymond James & Associates, INC, 2001
- 22 Milgram S. The Small World Problem. Psychology today. 1, 61
- 23 Singh M. Peering at Peer-to-Peer Computing. IEEE Internet Computing, 2001, 5(1): 4~5
- 24 Sylvia R, Scott S, Ion S. Routing Algorithms for DHTs: Some Open Questions. In: 1st Intl. Workshop on Peer-to-Peer Systems, March 2002
- 25 Emil S, Robert M. Security Considerations for Peer-to-Peer Distributed Hash Tables. In: 1st Intl. Workshop on Peer-to-Peer Systems, March 2002
- 26 Adriana I, Matei R, Foster I. Locating Data in (Small-world?) Peer-to-Peer Scientific Collaborations. In: 1st Intl. Workshop on Peer-to-Peer Systems, March 2002
- 27 Milojicic D S, et al. Peer-to-Peer Computing. HP Laboratories palo alto, HPL-2002-57, March 2002
- 28 Steve W, David M D, Gene K, Yaroslav F. Distributed Search in P2P Networks. IEEE Internet Computing, 2002, 6(1): 68~72
- 29 P2P Working Group home page. <http://www.peer-to-peer-wg.org>
- 30 P2P Working Group White Paper. <http://www.peer-to-peer-wg.org/tech/security/whitepapers/SecurityRequirementsDocumentversion0.9.2.pdf>
- 31 卢锡成, 李东升, 王意洁, 等. 基于对等模式的分布存储技术研究. 见: 第12届全国信息存储技术学术会议(NCIS2002), 上海, 2002
- 32 Zhou Shijie, Qin Zhiguang. P2P-Base Floating Information Storage System. [Technique Report in IBM Technique Center]. UESTC, 2003
- 33 Zhou Shijie, Qin Zhiguang. P2P based distributed key management system. 研究与发展(增刊), 2003
- 34 周世杰, 秦志光, 张峰, 等. 基于 P2P 的信息存储技术. 见: 第12届全国信息存储技术学术会议(NCIS2002), 计算机研究与发展, 2003年增刊, 上海, 2002