# P2P网络中避免集散节点形成的控制模型

杨 春[1] 刘玉华[1] 许凯华[2] 陈洪才[1]
(华中师范大学计算机科学系 武汉 430079)[1] (华中师范大学数字空间研究中心 武汉 430079)[2]

**摘 要** P2P网络中集散节点的存在会导致整个系统的抗协同攻击能力大大降低,增加网络的脆弱性。对目前P2P网络中集散节点现象进行了研究,阐述了对集散节点进行层次化处理的控制思想,提出了一种全新的通过控制P2P网络的逻辑拓扑结构来避免集散节点形成的思路,给出了控制模型以及实现控制模型的具体算法,并对算法进行了复杂度分析,最后通过仿真证明了本控制模型能有效控制网络中集散节点的形成,所以本控制模型能提高P2P网络抗协同攻击的能力,增强网络的健壮性,从而达到保障P2P网络可持续健康发展的目的。

**关键词** P2P网络,集散节点,拓扑结构控制,层次化

## Model of Controlling the Hubs in P2P Network

YANG Chun[1] LIU Yu-hua[1] XU Kai-hua[2] CHEN Hong-cai[1]
(Department of Computer Science, Huazhong Normal University, Wuhan 430079, China)[1]
(Research Center of the Digital Space Technology, Huazhong Normal University, Wuhan 430079, China)[2]

**Abstract** This paper researchesd into the hubs in P2P network, and presented a new method to avoid generation of the hubs in the network by controlling the logical topology structure of P2P network. It firstly introduced the controlling ideas about hierarchizing the hubs. Then, it disclosed and interpreted the controlling model, and gave out the concrete method to carry it out. Finally, it validated the controlling model via simulations. The simulation results demonstrate that the work is effective to control the hubs in P2P network. Thus, this model can improve the network competence to defend against coordinated attacks, promote the network robustness, and ensure that the network would develop continually and healthily.

**Keywords** P2P network, Hubs, Topology control, Hierarchical

## 1 前言

P2P即为对等网或点对点技术,是对传统的C/S通信模式的一种变革。在P2P网络中,每个节点的地位都是对等的,每个节点既为其它节点提供服务,又享用其他节点提供的服务。对等点之间通过直接互连共享信息资源、处理器资源、存储资源甚至高速缓存资源而无需依赖集中式服务器[1]。

P2P通信模式为普通用户提供了快捷方便的文件搜索和下载功能,使得P2P的应用迅猛发展,但同时带来了一些弊端。研究发现,P2P网络中存在着一些数量极少但维持的连接数却极高的节点[2-8],这些拥有超大连接数的节点称为集散节点。网络中集散节点的存在会导致整个网络的抗协同攻击能力大大降低,增加网络的脆弱性[3]。如果对此现象不加以控制,集散节点数量会越来越多,以致于这些集散节点发生故障或退出网络,越来越容易导致整个P2P网络服务系统的瘫痪[7]。因此,如何避免集散节点的出现,成为P2P网络可持续健康发展待解决的关键问题之一。本文通过改变P2P网络的逻辑拓扑结构来控制网络中集散节点的形成,以此增强网络的抗脆弱性,保障P2P网络可持续健康发展。

## 2 研究现状

目前,节点的行为特征、网络拓扑结构和搭便车现象等已成为P2P网络研究的新热点。在对Gnutella, Kazaa, BitTorrent等大型P2P系统的用户行为研究后发现,网络中节点的度分布存在类似幂律分布的现象[5]。节点的度是指一个节点拥有的连接数量,节点度服从幂律分布就是说具有某个度$k$的节点数目与这个度$k$之间的关系可以用一个幂函数近似地表示:$P(k)=k^{-\lambda}$,其中,$\lambda$为常数。幂函数曲线是一条下降相对缓慢的曲线(如图1所示),幂律分布意味着网络中存在一些数量少但拥有度数值却极高的集散节点。这些少数的集散节点为广大用户服务,一旦某个集散节点发生故障或退出网络,就会对整个系统造成很大的冲击。并且,集散节点本身负载过重,服务能力有限。可见,这些集散节点分布在网络中必定会降低整个P2P系统的服务性能和质量[4],增加网络的脆弱性,因此P2P网络中存在的集散节点现象有必要引起我们的高度重视。

当前集散节点控制机制主要集中在如何控制单个节点的用户行为。例如,文献[9]利用utility function概念来强制控

到稿日期:2008-04-01 本文受国家自然科学基金资助项目(60673163)资助。

杨 春(1984—),男,硕士研究生,主要研究方向为复杂网络、P2P网络,E-mail:c_yang@yeah.net;刘玉华(1951—),女,博士,教授,主要研究方向为计算机网络、复杂网络、P2P网络、无线网络等。

制单个节点从 P2P 网络中获得信息下载资源的能力。仿真证明，即使是十分简单的 utility function，也能有效控制搭便车行为。文献[10]提出采用博弈论的方法控制节点行为，其基本方法是根据节点对系统贡献的共享程度和信誉级别来决定其从网络系统中获取信息资源的多少和服务质量的高低。文献[11]提出利用市场机制模型来控制虚拟组织中的搭便车行为，该文认为在 P2P 网络等一些分布式应用领域，因为不存在一个有效的中心控制者，所以不太可能采用集中式的控制机制，因此采用一种分布式的协同控制方法是一种必然。采用激励机制[12-14]控制单个节点的行为是目前对 P2P 网络中集散节点控制研究最为广泛的手段，但此类方法可能使系统对部分搭便车者失去吸引力，从而导致用户数量降低。而往往在商业运营中，用户数量是决定整个系统应用成功与否的关键，用户数量降低意味着系统因控制集散节点采取的措施是失败的。因此，这样的控制方案可能会违背使系统更加健壮发展的初衷。

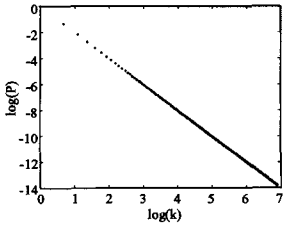



图 1　幂律分布曲线($\lambda=2$)

## 3　集散节点控制模型的建立

除了从节点的用户行为方面控制 P2P 网络中集散节点的形成外，还有一个更值得注意的研究角度，就是通过控制网络的逻辑拓扑结构来解决集散节点形成的现象，并且此方法不会导致前述方法所可能产生的用户数减少的负面效应。在复杂网络研究领域，文献[15]提出过对无尺度网络中的集散节点做层次化处理和分布式处理的方法，从而优化了无尺度网络的拓扑结构，增强了网络的抗脆弱性。本文从控制 P2P 网络的逻辑拓扑结构入手，对集散节点做层次化处理，以控制集散节点，改变 P2P 网络由于存在集散节点带来的脆弱性，有效提高 P2P 网络的抗协同攻击能力与网络的健壮性。

### 3.1　控制思想

当发现网络中某个节点 $V_i$ 即将成为集散节点时，找出节点 $V_i$ 所共享的资源中被请求连接数最多的那部分资源，立即从 $V_i$ 出发，搜索网络中具有这部分资源的其他节点。从这些节点中挑选出两个可再支撑连接数最多的节点作为备用节点，并记录这两个节点的 IP 地址。备用节点与原节点构成二叉树逻辑结构，其中原节点为根节点。然后将新的连接这部分资源的请求转发给备用节点之一，备用节点响应请求并与源请求节点直接建立连接。转发规则为：根据请求数据包的源 IP 地址计算源请求节点离哪个备用节点最近，则转发至此备用节点。如果备用节点又出现集散现象，则对备用节点的共享资源再重复上述方法，最终将形成多层次的二叉树逻辑结构(如图 2 所示)。

经此轮处理后，如果 $V_i$ 的集散现象未得到改观，则对其所共享的其它资源再进行上述处理。重复此过程，直至 $V_i$ 的集散现象消失。最后，整个系统将形成多个二叉树交叉在一起的混合结构(如图 3 所示)。

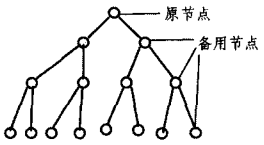



图 2　二叉树逻辑结构

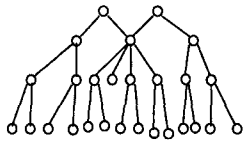


图 3　混合二叉树结构

### 3.2　控制模型

**定义 1**　可再支撑连接数 $L(i)$ 表示除了已拥有的连接外，节点还能再支撑多少个连接。即 $L(i)=R(i)-D(i)$，$R(i)$ 为该节点的阀值，$D(i)$ 为该节点的度数。

**定义 2**　$VQ$ 与 $VL$ 的 IP 地址差值 $|IP(VQ)-P(VL)|$ 就是用 $VQ$ 的 IP 地址按每个点分位减去 $VL$ 的 IP 地址，取正值。例如，假设 $VQ$ 的 IP 地址为 $IP(VQ)=128.11.3.31$，$VL$ 的 IP 地址为 $IP(VL)=222.1.3.2$，$VR$ 的 IP 地址为 $IP(VR)=128.11.3.9$，则 $|IP(VQ)-IP(VL)|=94.10.0.29$，$|IP(VQ)-IP(VR)|=0.0.0.22$。IP 地址差值关系运算则是根据高位优先原则从左至右对每个点分位比较大小，即对 IP 地址差值从第一个点分位开始比较大小，分出大小则结束比较。否则，依次比较大小，直至第 4 个点分位。例如上例中，由于 $|IP(VQ)-IP(VL)|$ 的第一个点分位 94 大于 $|IP(VQ)-IP(VL)|$ 的第一个点分位 0，即可知 $|IP(VQ)-IP(VL)|>|IP(VQ)-IP(VL)|$，而无需再比较剩下的点分位。

具体控制模型如下：

设 P2P 网络中某节点 $V_i$ 的阀值为 $R(i)$，即所能支撑的最大连接数为 $R(i)$。在某时刻 $t$，节点 $V_i$ 接收到新的连接请求 $Q$，此时刻节点 $V_i$ 共享了 $m(m\geqslant 1)$ 个资源 $S$。设 $Q$ 是关于资源 $s$ ($s\in S$)的请求，此时刻 $V_i$ 的度数为 $D(i)$。

若 $D(i)<R(i)-1$，则直接接受请求；

若 $D(i)=R(i)-1$，且 $V_i$ 已确立关于资源 $s$ 的备用节点，则 $V_i$ 将 $Q$ 按后述转发规则 $a$ 转发给备用节点；

若 $D(i)=R(i)-1$，且 $V_i$ 没有确立关于资源 $s$ 的备用节点，则处理过程如下：设此时 $V_i$ 共享的 $m$ 个资源中未确立备用节点的资源有 $m'(m'\leqslant m)$，此 $m'$ 个资源的连接数关系为 $C_{S(1)}\geqslant C_{S(2)}\geqslant\cdots\geqslant C_{S(m')}$，$(C_{S(1)}+C_{S(2)}+\cdots+C_{S(m')}+C_{S(m'+1)}\cdots+C_{S(m)}=D(i)=R(i)-1)$，则节点 $V_i$ 对连接数最多的资源 $S(1)$ 进行网络搜索，得到 $k$ 个节点。

(1)若从此 $k$ 个节点中能选出可再支撑的连接数最多(且不为 0)的两个节点 $VL$，$VR$，则将 $VL$，$VR$ 作为节点 $V_i$ 关于资源 $S(1)$ 的备用节点，并与 $V_i$ 构成二叉树逻辑结构，$V_i$ 为根节点，$VL$，$VR$ 分别为左、右子节点。记 $VL$，$VR$ 的 IP 地址分别为 $IP(VL)$，$IP(VR)$，对于连接请求 $Q$，

a. 若 $s=S(1)$：记发送请求 $Q$ 的源客户机 $VQ$ 的 IP 地址为 $IP(VQ)$，分别计算 $VQ$ 与 $VL$，$VR$ 的 IP 地址差值，即 $|IP(VQ)-IP(VL)|$，$|IP(VQ)-IP(VR)|$，$V_i$ 将 $VQ$ 的请求转发给差值较小的备用节点，$VQ$ 再与此备用节点直接建立连接。例如，设 $|IP(VQ)-IP(VL)|>|IP(VQ)-IP(VR)|$，则 $V_i$ 将 $VQ$ 的请求转发给 $VR$，$VQ$ 再与 $VR$ 直接通信而无需再通过 $V_i$ 转发；

b. 若 $s\neq S(1)$：则 $V_i$ 中止一个最新的关于资源 $S(1)$ 的连接 $L$，$V_i$ 响应请求 $Q$。令 $Q=L$，并将 $Q$ 按转发规则 $a$ 处理。

(2)若只存在一个可再支撑的连接数不为 0 节点，则将之

记为 $VL$。令 $VR$ 为空，$IP(R)=0.0.0.0$，即化为情形(1)。

(3)若不存在可再支撑的连接数不为 0 节点，则节点 $V_i$ 对连接数次多的资源进行网络搜索，再做上述判断处理。如果重复此过程到 $S(m')$ 仍无可再支撑的连接数不为 0 节点，则返回等待重试信息。

如果备用节点 $VL$ 或 $VR$ 的连接数达到 $R(L)-1$ 或 $R(R)-1$，则对该节点进行上述的搜索过程。假设此搜索过程有关资源 $s'$，筛选未确立关于资源 $s'$ 的备用节点的节点作为 $VL$ 或 $VR$ 的备用节点。

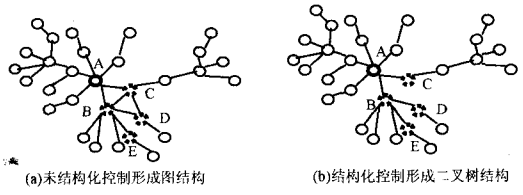
若确立的逻辑二叉树中有节点退出，则用其左子节点 $VL$ 代替该节点。

### 3.3 模型描述

(1)本模型用可再支撑连接数 $L(i)$ 的概念来考虑备用节点的综合性能。$L(i)$ 越大，说明节点的各种性能(计算处理能力、带宽等)越强，为其他节点提供服务的能力就越强。

(2)IP 地址空间管理目标具有聚类特点：地址空间应该尽可能地依据网络基础结构的拓扑，以划分层次的方式进行分配，有利于路由信息的聚类和限制路由表的增大[16]。由该管理目标可知，根据 IP 地址差值的大小关系可以粗略地判断出备用节点 $VL$，$VR$ 中与源请求节点 $VQ$ 物理位置较近者。例如定义 2 中的举例，从 $VQ$ 和 $VR$ 的 IP 地址可以看出二者同属于一个网络域，相比之下 $VL$ 距 $VQ$ 则更远。研究发现，P2P 网络中，如果应用层的重叠网络拓扑结构和物理层实际的网络拓扑结构越接近，则网络不同域之间的流量会越少，而本地网络的高速带宽会得到充分利用[5,6]。因此，本模型根据 IP 地址差值转发连接请求，以尽量使 P2P 网络的逻辑拓扑结构接近于物理拓扑结构。

(3)本控制模型将备用节点与原节点逻辑结构化。如果不加以结构化控制，则备用节点与原节点之间会形成无规则的图结构(如图 4 所示)，而图结构的搜索算法的时间复杂度要高于二叉树结构。另外，P2P 网络是动态变化的，备用节点可能随时退出，而二叉树中节点的更新已有成熟的算法可采用，维护较容易，因此将备用节点与原节点建立二叉树逻辑结构是比较合适的。




(a)未结构化控制形成图结构　　(b)结构化控制形成二叉树结构

图 4　两种结构对比

## 4 控制算法与复杂度分析

### 4.1 控制算法

根据上述控制模型，给出具体算法步骤如下：

STEP0　设接收到请求 $Q$(关于资源 $s$)的节点为 $V_i$，获取 $V_i$ 的阀值 $R(i)$ 和当前的度数 $D(i)$及其共享的资源数 $m$。

STEP1　若 $D(i)<R(i)-1$，则直接接受请求；否则，转 STEP2。

STEP2　若 $V_i$ 已建有关于资源 $s$ 的二叉树，获取其左、右子女 $VL$，$VR$，转 STEP7；否则，转 STEP3。

STEP3　对 $V_i$ 所共享的资源中未确立二叉树的资源($m'$个)的连接数关系进行排序，设 $C_{S(1)}\geqslant C_{S(2)}\geqslant\cdots\geqslant C_{S(m')}$，令 $j=1$，转 STEP4。

STEP4　若 $j\leqslant m'$，转 STEP5；否则，等待重试。

STEP5　对 $S(j)$进行网络搜索，设得到 $k$ 个节点。若能从 $k$ 个节点中选出可再支撑连接数最多(>0)的两个节点，记为 $VL$，$VR$，则转 STEP6；若只存在一个可再支撑的连接数>0 的节点，记为 $VL$，则转 STEP9；否则，转 STEP10。

STEP6　令 $VL$，$VR$ 为 $V_i$ 关于 $S(j)$的左、右子节点，若 $s=S(1)$，转 STEP7；否则，转 STEP8。

STEP7　若 $VR\neq NULL$，$Temp=\min\{|IP(VQ)-IP(VL)|,|IP(VQ)-IP(VR)|\}$；否则，$Temp=VL$。将 $Q$ 转发给 Temp，结束。

STEP8　$V_i$ 中止一个最新的关于资源 $S(j)$的连接 $L$，$V_i$ 接受请求 $Q$。令 $Q=L$，转 STEP7。

STEP9　令 $VR=NULL$，转 STEP7。

STEP10　$j++$，转 STEP4。

### 4.2 算法实现与复杂度分析

设节点所确定的关于某资源 $s$ 的备用节点 $VL$，$VR$ 的记录格式为三元组：$(s,VL,VR)$，每个准集散节点维护一张备用节点表 List_children，此表记录本节点所确立的三元组集合，如表 1 所列。每个备用节点维护一张父节点表 List_parent，记录关于资源 $s$ 的父节点 $V_i$，如表 2 所列。

表 1　List_children

| 资源 | 左子女 | 右子女 |
|---|---|---|
| s | IP(VL) | IP(VR) |

表 2　List_parent

| 资源 | 父节点 |
|---|---|
| s | IP($V_i$) |

由 3.2 节知，当 $D(i)=R(i)-1$ 时，需要搜索表 List_children，查找是否具有资源 $s$ 项。查找成功，则返回左、右子女；若不成功，则返回 false。

当某节点 $V_i$ 新确立关于资源 $s'$的左、右子女即备用节点时，该节点维护的表 List_children($V_i$)需要添加新记录($s'$，$IP(V_iL)$，$IP(V_iR)$)，同时 $V_iL$，$V_iR$ 需向各自维护的表 List_parent($V_iL$)，List_parent($V_iR$)添加记录($s'$，$IP(V_i)$)。

当二叉树中某节点 $V_j$ 退出系统时，对其 List_parent($V_j$)表中每一个资源查询其 List_children($V_j$)表。如查询成功，则将其左子女 $IP(V_jL)$信息发送至该资源对应的父节点，父节点将其 List_children 表中对应该资源项的 $IP(V_j)$信息更新为 $IP(V_jL)$；如查询不成功，则发送 null 信息至该资源对应的父节点，父节点将其 List_children 表中对应该资源项的 $IP(V_j)$信息更新为 null。

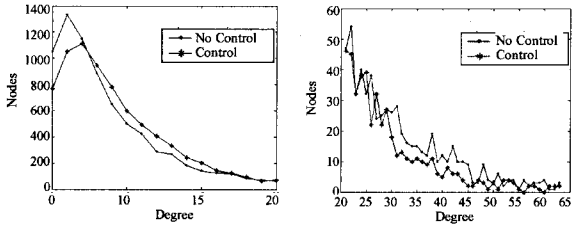
此算法最复杂之处在于第三种情况，即 $D(i)=R(i)-1$ 且 $V_i$ 没有确立关于资源 $s$ 的备用节点。STEP2 需要先搜索 List_children 表，表中至多有 $m$ 项，查找算法最坏情况下需要比较 $m$ 次。其次，在 STEP3 中查找连接数最多的资源，属于内部排序问题，最坏情况下也需要比较 $m$ 次。再次，STEP5 中搜索网络，其时间复杂度 $T$ 取决于现有的 P2P 搜索算法，从 $k$ 个节点中选出可再支撑的连接数最多的两个节点，其比较次数为 $2k-1$。若不存在可再支撑的连接数不为 0 节点，STEP5 过程需重复 $m$ 次，其时间复杂度为 $m*(T+2k-1)$。所以，整个算法总的时间复杂度为 $m+m+m*(T+2k-1)$。由 $m$，$k$ 的意义可知，$m$ 与 $k$ 均为可数的，所以总时间复杂度为 $O(T)$，即取决于 P2P 搜索算法的时间复杂度。

## 5 仿真评估

本文利用 BA 算法[17]产生仿真网络拓扑结构，整个结构包括 8000 个节点。假设有 1000 个文件随机分布在这 8000 个节点上，每个节点共享 100 个互不相同的文件，每个新增节点连接 5 条边。对各个节点的阀值随机取值。但为了能在仿真规模较小的情况下得出结果，假设各个节点的阀值不超过 30。实验采用 Gnutella 仿真网络模型和 Flooding 搜索算法，在采用本文的控制模型和不采用此模型两种情况下，分别模拟了 10000 次网络中资源连接请求。图 5 是两种情况下的节点度分布结果比较图。为了使表达效果更清楚，而将节点的度数划分为了两个区域($D\leqslant 20$ 区间和 $D>20$ 区间)。

图 5(a)显示的是度数低于 20 的区间上两种情况的对比图。在度数小于 7 的区域，利用本控制模型后的节点数量明显低于未采取控制的情况，而在其后的区域，利用本控制模型后的节点数量又略高于未采取控制的情况。这是因为在未采取控制的情况下，网络中资源的连接遵循幂律规则，导致大量的节点仅拥有极少的连接。而在采取控制后，高度数节点的连接转至了低度数的节点，从而均衡了网络中节点的度数。

图 5(b)显示的是度数高于 20 的区间上的对比情况。从图中可以看出，在度数为 20 至 29 的部分，两种情况的结果还没有较大的区别。但在 30 至 45 部分，两种情况的对比已经十分明显。由于本仿真实验设定了各个节点的阀值不超过 30，因此在 30 处形成了明显的分界线。而在高于 45 的部分，由于两种情况下各度数对应的节点数都比较少(低于 5 个节点)，导致两种情况的对比度不明显。但是，采取控制之后最高的度数要低于无控制的情况(采取控制之后网络中的最高度数为 281，而无控制的情况下最高度数为 319)。




(a)度数 $D\leqslant 20$ 区间　　(b)度数 $D>20$ 区间

图 5　节点度分布情况比较

图 5(a)、(b)中呈现上述现象是因为本控制模型在控制集散节点的形成时，将准集散节点的连接请求转发至了度数较低的备用节点，改变了原有的逻辑拓扑结构，打破了幂律规则，从而将网络中节点的度数均衡化。反映在图上的效果即为小度数区域上节点的数量会增多，高度数区域上节点数量会减少，并且网络中最大的度数也会降低。

从本仿真实验可以看出，本模型能够将较高度数节点的连接均衡转移到较低度数的节点上，改变了 P2P 网络的逻辑拓扑结构，从而降低了网络中连接数过高的节点数，同时降低了网络中的最高度数，说明网络中集散节点数量得到了减少，集散现象得到了有效控制。并且，集散节点减少，网络抗协同攻击的能力必定增强，从而提高了网络的健壮性，改变了 P2P 网络由于集散节点的存在带来的脆弱性。因此，本仿真结果说明本控制模型是有效的。

**结束语**　P2P 网络中集散节点的存在会导致整个网络的抗协同攻击能力大大降低，增加网络的脆弱性，影响整个 P2P 系统的服务性能和质量。本文提出了一种全新的通过控制 P2P 网络的逻辑拓扑结构来避免网络中集散节点形成的方法。此方法不会导致激励机制控制方法可能会产生的用户数减少的负面效应，因而具有较高的实际应用价值。本文首先介绍了对集散节点进行层次化处理的控制思想，给出了控制模型及其论证。其次，详述了控制算法的实现方法并分析了此算法的时间复杂度。最后，利用仿真模拟了网络中资源请求的连接过程，对是否利用本控制模型的两种情况进行了对比。实验结果表明，本控制模型能有效控制 P2P 网络中集散节点的形成，从而说明本控制方法能提高整个网络的抗协同攻击能力，增强网络的健壮性，保障 P2P 网络可持续健康发展。

## 参 考 文 献


[1] Chen Shu, Fang Binxing, Zhou Yonglin. The Research and Application of P2P Technology. Computer Engineering and Applications, 2002, 38(13): 20-23

[2] Klemm A, Lindemann C, Vernon M, et al. Characterizing the Query Behavior in Peer-to-Peer File Sharing Systems // Proc. ACM Internet Measurement Conference (IMC). Taormina, Italy, Oct. 2004: 55-67

[3] Wang Wenjie, Chang Hyunseok, Zeitoun A, et al. Characterizing Guarded Hosts in Peer-to-Peer File Sharing Systems // IEEE Global Communications Conference. Globecom, 2004: 1539-1543

[4] Adar E, Huberman B. Free Riding on Gnutella. First Monday, October 2000

[5] Ripeanu M, Foster I, Iamnitchi A. Mapping the Gnutella Network: Properties of Large-scale Peer-to-Peer Systems and Implications for System Design. IEEE Internet Computing, 2002, 6 (1)

[6] Sen S, Wang J. Analyzing peer-to-peer traffic across large networks. IEEE/ACM Trans. on Networking, 2004, 12(2): 219-232

[7] Hughes D, Coulson G, Walkerdine J. Free Riding on Gnutella. Revisited: The Bell Tolls? IEEE Distributed Systems Online, 2005, 6(6)

[8] Krishnan R, Smith M D, Tang Zhulei, et al. The Impact of Free-riding on Peer-to-Peer Networks // Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS). 2004: 199-208

[9] Ramaswamy L, Liu Ling. Free Riding: A New Challenge to Peer-to-Peer File Sharing Systems // Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS). 2003: 220-229

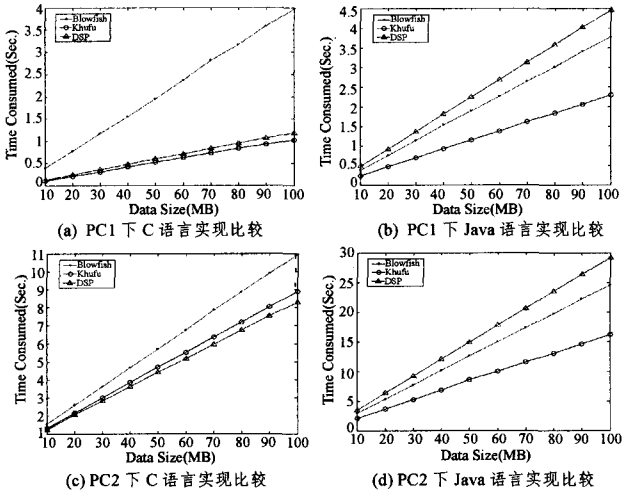
[10] Gupta R, Somani A K. Game Theory As A Tool To Strategize As Well As Predict Nodes' Behavior In Peer-to-Peer Networks // Proceedings of the 11th International Conference on Parallel and Distributed Systems (ICPADS). 2005: 244-249

[11] Xavier P, Cai Wentong, Lee Bu-Sung. Employing economics to achieve fairness in usage policing of cooperatively shared computing resources // Proceedings of 2005 IEEE International Symposium on Cluster Computing and the Grid (CCGrid). 2005: 326-333

[12] Ma R T B, Lee C M, Lui J C S, et al. Incentive P2P Networks: A Protocol to Encourage Information Sharing and Contribution. Performance Evaluation Review, 2003, 31(2): 23-25


(下转第 81 页)

其中C程序在Visual C++ 6.0下编译，Java程序的编译器选择为JavaSoft的JDK1.5。为了提高实验的准确性，我们取10次实验的平均值作为实验结果。密码的操作模式选择为电子密码本(ECB)模式，实验通过对10～100MB随机块数据的加密来计时并计算效率，由于Feistel密码的对称性，DSP，Blowfish和Khufu加/解密速度相等，因此本文只对算法加密的效率进行比较。图2为算法在不同实验环境下的加密效率的比较。



(a) PC1下C语言实现比较　(b) PC1下Java语言实现比较

(c) PC2下C语言实现比较　(d) PC2下Java语言实现比较

图2　加密效率比较

从图2我们可以看出，在不同环境下，Blowfish算法和Khufu算法在两台PC上都有着相当高的运行效率，这是因为Blowfish和Khufu算法都是针对32位处理器设计的，当算法运行在16位处理器或者8位处理器上时，算法加密单位数据所用的时钟周期将显著升高，严重影响算法的工作效率；而且随着处理器性能的降低，Khufu算法相对于DSP的速度优势逐渐降低，在Celeron 400MHz处理器上，DSP的C语言代码加密速度已高于Khufu算法。DSP算法在各种环境下都具有相对较高的运行效率，并且该算法以字节操作为基础，在不同的处理器上具有良好的兼容性。

**结束语**　本文提出了一种新的128位Feistel分组密码算法——DSP。该算法将密钥相关的动态S-盒与密钥相关动态P-盒相结合，隐藏了密码算法的内部结构，从而有效地抵抗差分密码分析和线性密码分析。算法的设计仅使用了异或和查表两种基于字节的基本操作，因此算法具有良好的实现效率和兼容性。在算法的密钥初始化部分，我们采用RC4算法中的状态初始化函数作为置乱算法，实现了快速的密钥扩展，从很大程度上弥补了密钥相关结构算法初始化过程开销巨大、运行缓慢的问题。我们在Pentium处理器上分别用C和Java实现了DSP，实验结果表明，加密解密过程及密钥初始化均十分高效。

## 参考文献


[1] Biham E, Shamir A. Differential cryptanalysis of the Data Encryption Standard [M]. New York: Springer-Verlag, 1993

[2] Matsui M. Linear cryptanalysis method for DES cipher [C]// Advances in Cryptology -EUROCRYPT ' 93 Proceedings. Springer-Verlag, 1994: 286-397

[3] Daemen J, Knudsen L R, Rijmen V. The block cipher Square [C]// Fast software encryption-FSE'97. Haifa, Israel: Springer Verlag, January 1997: 149-165

[4] Advanced Encryption Standard [S]. FIPS-197. National Institute of Standards and Technology, Nov. 2001

[5] Matsui M. New block encryption algorithm MISTY [C] // Fast Software Encryption - 4th International Workshop (FSE'97), LNCS. vol. 1267. Springer-Verlag, 1997: 54-68

[6] Aoki K, Ichikawa T, Kanda M, et al. Camellia: A 128-bit block cipher suitable for multiple platforms-Design and analysis [C]. submitted to NESSIE. Available at: http://www.cryptonessie.org, 2000

[7] Schneier B, Kelsey J, Whiting D, et al. Twofish: A 128-Bit Block Cipher [C]// First Advanced Encryption Standard (AES) Conference. Ventura, California, USA, 1998

[8] Merkle R C. Fast software encryption functions [C]// Proc. CRYPTO'90. LNCS. vol. 537. Springer-Verlag, 1990: 476-501

[9] Schneier B. Description of a new variable - length key, 64 - bit block cipher (Blowfish) [C]// Fast Software Encryption-Porceedings of the Cambridge Security Workshop. Lectures Notes in Computer Science 809. Cambridge, United Kingdom, Springer-Verlag, 1994: 191-204

[10] Biham E, Biryukov A, Shamir A. Miss-in-the-middle attacks on IDEA, Khufu and Khafre [C]// 6th Fast Software Encryption Workshop, LNCS. vol. 1636, Springer-Verlag, 1999: 124-138

[11] Vaudenay S. On the weak keys of Blowfish [C]// Third International Workshop Proceedings. Springer-Verlag, 1996: 27-32

[12] Schneier B. Applied Cryptography [M]. 2 edition. John Wiley & Sons, Inc, Toronto, Canada, 1996


(上接第65页)


[13] Anceaume E, Gradinariu M, Ravoaja A. Incentive for P2P Fair Resource Sharing// Proceedings of the IEEE International Conference on Peer-to-Peer Computing (P2P). 2005: 253-260

[14] Gong Haigang, Liu Ming, Mao Yingchi, et al. Research Advances in Key Technology of P2P-based Media Streaming. Computer Research and Development, 2005, 42(12): 2033-2040

[15] Tao Shaohua, Liu Yuhua, Xu Kaihua, et al. The Strategies against Vulnerability of Hubs in Complex Networks. Computer Engineering and Applications, 2007, 43(2): 151-153

[16] Policies for IPv4 address space management in the Asia Pacific region. http://www.apnic.net/trans/cns/ add-manage-policy.pdf

[17] Albert R, Barabási A L. Emergence of scaling in random networks. Science, 1999, 286: 509-512