

# 基于博弈论的 P2P 激励机制的研究与设计

李磊,董健全

(上海大学 计算机工程与科学学院,上海 200072)

**摘要:**针对 P2P 系统中普遍存在的搭便车和公共悲剧问题,提出了一种基于博弈论的 P2P 服务质量激励机制。在分析节点在网络中的贡献和收益的基础上,通过引入激励值的概念来体现网络节点服务质量的高低,同时给出了有关节点服务质量四个方面的定义。实验表明,该激励机制能够有效地抑制节点的自私行为和提高网络的服务质量,并促使每个节点自发地共享自己的资源,从而构建一个良好的 P2P 环境。

**关键词:**P2P;激励机制;博弈论;激励值;服务质量

**中图分类号:**TP393

**文献标识码:**A

**文章编号:**1673-629X(2009)05-0005-04

## Research and Design of An Incentive Mechanism of P2P Based on Game Theory

LI Lei, DONG Jian-quan

(School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China)

**Abstract:** According to the universal questions like free riding and tragedy of common in the P2P systems, propose incentive mechanism of P2P service quality based on the game theory. After analyzing the contributions and income foundations of nodes in the network, by driving the concept of the incentive value to incarnate the size of the service quality, give definitions about the grade of nodes' service quality from four aspects. The simulation experiments proves that this incentive mechanism can restrain the selfish actions of the nodes and improve the quality of the network's service effectively, also it can urge every node sharing their resources spontaneously, so a good P2P environment will be built.

**Key words:** P2P; incentive mechanism; game theory; value of incentive; service quality

### 0 引言

在 P2P (peer-to-peer) 系统中所有节点都是对等的。这些节点既是客户机同时又是服务器,称之为对等机,这种对等计算模式具有诸如无中心、自组织以及高可扩展性等特征。然而由于 P2P 系统中的节点往往更多地表现出自兴趣和理性,单个节点的目标往往是使自身的网络效用(network utility)<sup>[1]</sup>最大化,这样导致了以下问题:

(1) 搭便车(free-riding)问题,正如文献[2]指出,70%的 Gnutella 用户不共享任何文件,接近 50% 的文件查询命中仅来自 1% 的 Gnutella 用户;

(2) 公共的悲剧<sup>[3]</sup>(tragedy of common)问题,即网络资源作为非排他性占有的公共资源,被大多数 P2P

节点无节制地使用。

如何使节点自发地共享自己的资源不仅是困扰 P2P 系统的问题,同时也严重困扰着大规模计算资源共享系统。PPay<sup>[4]</sup>和 Fileteller<sup>[5]</sup>等系统使用了微支付手段解决激励问题,但是微支付手段原则上要求第 3 方和安全身份认证,这与 P2P 的理念相背。FAR-SITE<sup>[6]</sup>和 Pastiche<sup>[7]</sup>则限制节点的占用存储资源与其贡献的存储资源相当,而 Pastiche 作为一个 P2P 的协作备份系统,节点的贪婪问题仍然没得到有效解决。

讨论了一种基于博弈论的激励机制,确定了节点的贡献值,基于节点的收益函数提出了一个激励值的概念来分析网络中节点服务质量的高低,同时给出了节点服务质量四个方面的定义和计算公式,通过模拟实验,论证了该激励机制能有效地提高网络的服务质量并能使网络中自私节点的数量大大减少。

### 1 节点贡献值的确定

现有激励机制的研究都是基于节点的贡献,文献

收稿日期:2008-09-08

基金项目:上海市重点学科建设项目(J50103)

作者简介:李磊(1983-),男,安徽寿县人,硕士研究生,研究方向为数据库、人工智能、网络;董健全,副教授,研究方向为网络、数据库、人工智能。

[8]提出了一种节点对网络贡献大小的激励机制来增加网络中共享文件的用户的比例,使得网络有更多的可用资源,减少搭便车行为。该文献只给出了节点贡献的表现形式,这里给出节点贡献的定义公式。用  $B_{\max}$  表示节点  $i$  所能提供的最大上传带宽;  $B_0$  表示网络中的节点至少提供上传带宽的大小;  $F_{\text{high}}(j, t)$  表示节点  $i$  共享的第  $j$  个文件在  $t$  时刻的热度值;  $F_{\text{size}}(j)$  表示节点  $i$  共享的第  $j$  个文件的大小;  $n$  是节点  $i$  共享的文件数目;  $t$  表示节点  $i$  加入 P2P 网络后的在线时间; 用  $P_{\text{spend}}(j, t)$  表示节点  $i$  加入网络后在  $t$  时刻的贡献值,其计算公式如式(1)所示。

$$P_{\text{spend}}(i, t) = \frac{B_{\max}}{B_0} \times \int_0^t \sum_{j=1}^n (F_{\text{high}}(j, t) \times F_{\text{size}}(j)) \quad (1)$$

## 2 激励值的引入

文献[9]提出的激励模型中,给出了节点的效用函数,即节点加入系统所需付出的代价和从系统所获得的收益之差,但它并没有体现出节点间的服务质量的差异。这里引入一个激励值的概念,它的大小表示为请求节点提供服务质量的高低,即若某个节点的激励值越高,则当其请求资源时所获得的服务质量越高;若某个节点的激励值越低,则当其请求资源时所获得的服务质量就越低。用  $P_{\text{intensive}}(i, t)$  表示节点  $i$  在  $t$  时刻的激励值,其计算公式如式(2)所示。

$$P_{\text{intensive}}(i, t) = U(i, t) + \theta(t) \quad (2)$$

在式(2)中,  $U(i, t)$  表示为节点  $i$  在  $t$  时刻的效用函数<sup>[9]</sup>,在文献[9]基础上对效用函数中节点加入系统的代价进行了改进,其计算公式如式(3)所示。

$$U(i, t) = -P_{\text{spend}}(i, t) + \frac{P_{\text{spend}}(i, t)^a}{1 + P_{\text{spend}}(i, t)^a} \sum_{j=1, j \neq i}^m W_{ij} \times P_{\text{spend}}(j, t) \quad (3)$$

在式(3)中,  $\frac{P_{\text{spend}}(i, t)^a}{1 + P_{\text{spend}}(i, t)^a}$  为一个差异服务概率函数,表示节点贡献的资源越多,则获得服务的概率就越大;  $W_{ij} (1 \leq i \leq m, 1 \leq j \leq m)$ , 其中  $m$  为节点个数,表示在节点  $j$  所贡献的所有资源中,对节点  $i$  有用的资源所占的比重。

在式(2)中,  $\theta(t)$  为计算节点激励值时所用的衰减函数,引入该函数的目的是初始化刚加入网络的新节点使其有一定的激励值,同时调整那些激励值高的节点使其激励值趋近于一个稳定的阈值,其计算公式如式(4)所示。

$$\theta(t) = -w_i \int_0^t \frac{1}{t+1} \quad (4)$$

在式(4)中,  $w_i$  是节点  $i$  用于衰减的权值。

## 3 节点的服务质量

为了体现节点的服务质量,从节点的下载速度、下载优先级、TTL 跳数以及请求延时等四个方面来考虑。参照节点的激励值,依次给出了四方面的定义。

### 3.1 下载速度的定义

下载速度是指网络中某个请求节点所分配到的对于某个文件的下载带宽。对于请求节点的下载速度的分配,取决于它们自身的激励值占有所有请求节点的激励值的比重,用  $P_{\text{BD}}(i, t)$  表示  $t$  时刻节点  $i$  所获得的下载速度的大小,其计算公式如式(5)所示。

$$P_{\text{BD}}(i, t) = \frac{P_{\text{intensive}}(i, t)}{\sum_{i=1}^m P_{\text{intensive}}(i, t)} \times B_{\max} \quad (5)$$

当  $P_{\text{intensive}}(i, t) = 0$  时,该请求节点所获得的下载速度也为 0,即该节点将无法下载文件。当  $P_{\text{BD}}(i, t) < \xi$  时,则该请求节点处于阻塞状态,进入等待队列。其中,  $\xi$  是一个阈值,表示网络中节点上传带宽的最小单位。

### 3.2 下载优先级的定义

下载优先级是指当某个节点出现热点文件时,对于所有请求该节点的节点请求,将按各个请求节点的优先级顺序,从高到低依次为其提供访问服务。对于请求节点下载优先级的分配,只是针对那些处于阻塞状态的节点,即如果各个请求节点获得的下载速度都大于阈值  $\xi$ ,则它们之间的优先级没有区别。既然是在等待队列中,对各个请求节点进行优先级排序,那么就不能仅仅取决于节点之间的激励值的大小,还要考虑它们的等待时间,用  $P_{\text{PRI}}(i, t)$  表示在  $t$  时刻节点  $i$  的优先度,其计算公式如式(6)所示。

$$P_{\text{PRI}}(i, t) = P_{\text{intensive}}(i, t) \times \frac{P_{\text{wait}}(i, t)}{\sum_{i=1}^m P_{\text{wait}}(i, t)} \quad (6)$$

在式(6)中,用  $P_{\text{wait}}(i, t)$  表示时刻  $t$  节点  $i$  已经阻塞的时间。所谓优先度是衡量节点对下载文件优先级的一个数值。等待节点的优先度越大,则表示该节点的优先级越高;反之,则表示该节点的优先级越低。

### 3.3 TTL 跳数的定义

额外增加 TTL 跳数指适当地增加请求节点在文件搜索过程中的 TTL 跳数,以此来扩大文件搜索的范围,以及文件搜索的命中率。对于请求节点的额外 TTL 跳数的增加,主要取决于它们一定时间内所积累的平均激励值的多少,用  $P_{\text{TTL}}(i, t)$  表示  $t$  时刻节点  $i$  所获得的 TTL 额外增加的跳数值,其计算公式如式(7)所示。

$$P_{TTL}(i, t) = \frac{1}{kP_{intensive0}} \times \sum_{t=1}^k P_{intensive}(i, t) \quad (7)$$

在式(7)中,  $\frac{1}{k} \sum_{t=1}^k P_{intensive}(i, t)$  表示节点  $i$  在  $k$  个时间间隔内的平均激励值;  $P_{intensive0}$  为系统的一个常量, 表示给请求节点增加额外 TTL 跳数所需的最小激励值。当  $\frac{1}{k} \sum_{t=1}^k P_{intensive}(i, t) < P_{intensive0}$  时,  $P_{TTL}$  为 0, 即不增加额外的 TTL 跳数。

### 3.4 请求延时的定义

请求延时是指网络中某个请求节点发出请求到收到请求回复的间隔时间。对于网络中节点的请求延时, 不仅取决于网络的传输带宽, 还取决于下载连接源节点的等待时间, 用  $P_{YS}(i, t)$  表示  $t$  时刻节点  $i$  的请求延时, 其计算公式如式(8)所示。

$$P_{YS}(i, t) = \frac{P_{intensive}(i, t)}{\sum_{i=1}^n P_{intensive}(i, t)} \times P_{wait}(i, t) \times \frac{B_0}{B_{max}} \quad (8)$$

当  $B_0 \geq B_{max}$  时, 节点的请求延时较长。

## 4 实验

### 4.1 系统实现

该机制的主要思想是: 运用博弈论原理及相关模型, 分析节点与节点间的博弈行为, 通过节点的贡献、节点的激励值以及节点的服务质量等方面的定义与计算, 激励 P2P 网络中的各个节点自发地贡献拥有的资源以提高网络的服务质量。其相关算法部分伪代码如下:

```
While(博弈终止条件)//循环一次, 进行一次博弈, 直到博弈终止
```

```
{
    For(int i = 1; i < PeerNum; i++) { //PeerNum 为网络中节点的个数
```

```
        CalculatePeerContribution(i); //计算节点的贡献值
```

```
        IsSelfishNode(i); //判断该节点是否是自私节点
```

```
        CalculatePeerIncome(i); //计算节点的收益值
```

```
        CalculatePeerIncentive(i); //计算节点的激励值
```

```
        TotalContribution = TotalContribution + CalculatePeerContribution[i] //计算网络中所有节点的贡献值
```

```
        TotalIncome = TotalIncome + CalculatePeerIncome[i] //计算网络中所有节点的收益值
```

```
}
```

```
For(int j = 1; j < PeerNum; j++) {
```

```
    CalculatePeerDownloadSpeed(i); //计算节点的下载速度
```

```
    CalculatePeerDownloadSeq(i); //下载优先级
```

```
    CalculatePeerRequestTime(i); //请求延时
```

```
    CalculatePeerTTL(i); //TTL 跳数
```

```
PeerServiceQuantity(i) =
```

```
w1 * CalculatePeerDownloadSpeed(i) + w2 * CalculatePeerDownloadSeq(i) + w3 * CalculatePeerRequestTime(i) + w4 * CalculatePeerTTL(i);
```

```
// 计算节点的服务质量, w1, w2, w3, w4 为下载速度, 下载优先级, 请求延时, TTL 跳数对应的权值
```

```
TotalServiceQuantity = TotalServiceQuantity + PeerServiceQuantity(i); //计算网络服务质量
```

```
}
```

### 4.2 性能分析

为了评价基于博弈论的 P2P 激励机制的性能, 进行了仿真实验<sup>[10]</sup>。假设网络中有 1000 个节点, 节点分成三类: 一类节点称为自私节点 SN(Selfish Node), 其贡献的资源小于从其它节点获得的资源; 另外一类节点称为无私节点 AN(Altruism Node), 其贡献的资源大于从其它节点获得的资源; 第三类为其它节点 ON(Other Node), 该类节点既不贡献资源, 也不从其它节点获得资源。在 P2P 仿真系统中, 初始化 SN 类节点所占比例为 30%, AN 类节点所占比例为 60%, ON 类节点所占比例为 10%。

#### 4.2.1 网络总收益、节点贡献总量和网络总激励值的分析

实验一模拟了在引入该激励机制的 P2P 系统中, 随着博弈次数的增加, 网络总收益、节点贡献总量及网络总激励值的变化情况。图 1 为博弈次数与网络总收益的关系图, 可以看出, 在网络形成的初期, 即节点新加入网络的时候, 网络的总收益并不是很高, 但随着节点与节点之间博弈次数的增加, 整个网络的总收益将大幅上升, 并逐渐趋向一个稳定值, 这个定值就是博弈论中所谓的纳什均衡点。图 2 为博弈次数与节点贡献总量的关系图, 通过节点之间的重复博弈, 网络中的节点的贡献总量不断提高, 这从另一个侧面反映出每个节点对网络的贡献量在不断增大, 最后, 贡献总量趋向于平衡, 表示节点之间的博弈已经达到了纳什均衡。图 3 为博弈次数与网络总激励值的关系图, 可以看出, 虽然各个节点通过相互博弈, 不断使自己的收益增加, 但它们所得到的激励值并没有随之单调递增。整个网络的总激励值, 即所有节点的激励值之和, 始终保持在一种平稳的状态。因为节点的激励值不仅包括它的收益值, 而且包括它的衰减值。随着时间的推移, 衰减函数会定期地减少节点的激励值, 以保证网络总激励值保持平稳。

#### 4.2.2 系统服务质量变化的分析

实验二模拟对比了基于激励机制的 P2P 系统和没有引入激励机制的 P2P 系统随着博弈次数增加, 系

统服务质量的变化情况。从图 4 可以看出,在采用文中激励机制的系统中,网络的服务质量要比未采用本文激励机制的系统要好,这是因为未采用激励机制的 P2P 系统对贡献自己的资源没有积极性,获得的收益就少,并不会获得较大的激励值来提供服务质量。而采用激励机制的 P2P 系统恰好相反,节点贡献的资源多少与获得的收益直接关联,贡献的源越多,获得的收益就越多,进而其激励值就越大,获得的服务质量就越好。

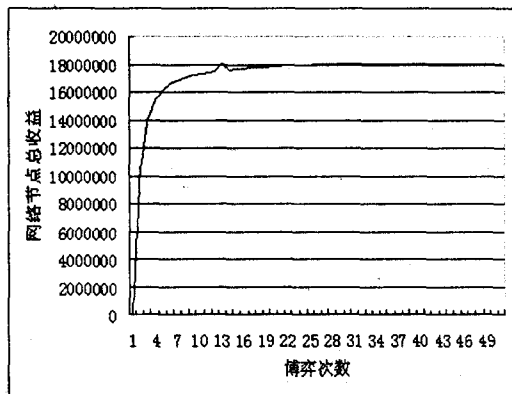


图 1 博弈次数与网络节点总收益关系图

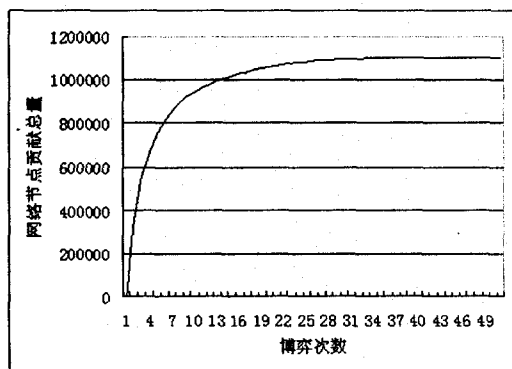


图 2 博弈次数与网络节点贡献总量关系图

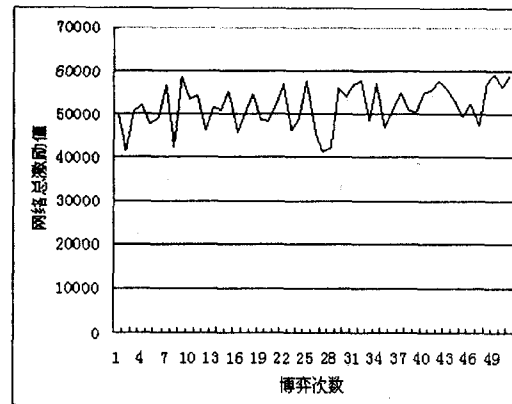


图 3 博弈次数与网络总激励值关系图

#### 4.2.3 本机制对“自私节点”的抑制作用

实验三模拟对比了基于激励机制的 P2P 系统和没有引入激励机制的 P2P 系统随着博弈次数增加,系

统中自私节点的变化情况。从图 5 可以看出,在未采用文中激励机制的系统中,自私节点随着博弈次数的增加上升的较快,而在采用激励机制的系统中则相反,自私节点随着博弈次数增加下降的很快,可见,该激励机制能有效地抑制节点的自私行为。

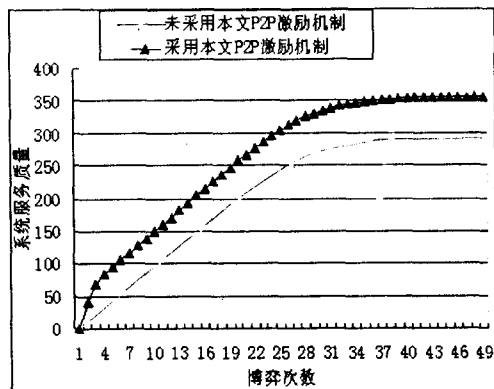


图 4 博弈次数与系统服务总量关系图

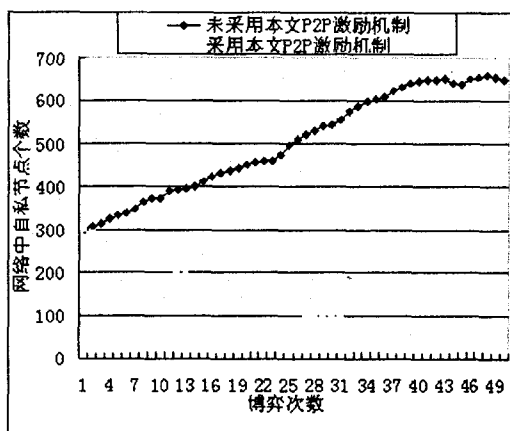


图 5 博弈次数与网络中节点自私个数关系图

## 5 结束语

针对 P2P 系统中普遍存在的 Free Riding 现象和 Tragedy of Common 现象,提出了一种基于博弈论的 P2P 激励机制,该机制定义了节点对网络的贡献,运用经济学中的博弈论思想,分析了节点在网络中可能的行为策略及贡献和获益,提出了一个激励值的概念来表示该节点服务质量的高低,最后通过模拟实验验证了该项研究能有效提升网络中节点的服务质量,抑制节点的自私行为。

#### 参考文献:

- [1] Shneidmain J, Parkes D C. Rationality and self-interest in peer to peer networks[C]//Second international workshop on peer-to-peer systems (IPTPS). [s.l.]: Springer-Verlag Press, 2003: 47-52.

#### 4 注册表监控

如果恶意脚本采用了变种或者加密手段,或者是最新出现的恶意脚本,它们将躲过第一道防线,顺利得到 WSH 的执行<sup>[7]</sup>。而接下来它们绝大多数会去修改注册表,这时注册表监控模块将发挥作用,该模块主要是监视注册表中的某些重要子项是否被改动,若在网页被放行后的一小段时间内(例如 1 秒钟内)被改动,判断为恶意脚本所为,则将注册表恢复为原先状态,并发出警告和关闭网页。该模块做成一个独立的线程,在钩子函数的最后被创建<sup>[8,9]</sup>。主要代码如下:

Public Sub RegMonitor() '注册表监控函数的定义

Dim hKey As Long, Ret As Long, hEvent As Long, lFilter As Long

Dim i As Long, j As Long

Ret = RegOpenKeyEx (HKEY\_LOCAL\_MACHINE, "SOFTWARE", 0&, KEY\_NOTIFY, hKey) '打开 SOFTWARE 项

hEvent = CreateEvent(0, True, False, vbNullString)

lFilter = REG\_NOTIFY\_CHANGE\_NAME Or REG\_NOTIFY\_CHANGE\_ATTRIBUTES Or REG\_NOTIFY\_CHANGE\_LAST\_SET Or REG\_NOTIFY\_CHANGE\_SECURITY

i = RegNotifyChangeKeyValue(hKey, True, lFilter, hEvent, True)

j = WaitForSingleObject(hEvent, 1000) '等待 1 秒钟,即监控 1 秒钟

If j = WAIT\_OBJECT\_0 Then

MsgBox "注册表被改动,监控程序将进行恢复!", vbOKOnly, "警告"

i = RegRestoreKey(hKey, App.Path & "\RegBack.reg",

&0)

End If

RegCloseKey (hKey)

ClosHandle (hEvent)

End Sub

#### 5 结束语

阐述了一种恶意网页防护系统的设计与实现,设计上采用了按时间顺序依次监控浏览器和注册表的较为周密的方案,实现上主要采用了钩子技术。经实践表明,该系统具有较好的实时防护效果。

#### 参考文献:

- [1] 鲍欣龙,罗文坚. 可用于恶意脚本识别的注册表异常行为检测技术[J]. 计算机工程, 2005, 31(8): 137-139.
- [2] 唐晓东,何连跃. 一种恶意代码防护方法及其实现[J]. 计算机工程, 2005, 31(12): 143-145.
- [3] 梁 庚,李 文. 应用系统钩子和内存映像实现一类进程间的通信[J]. 计算机工程与科学, 2005, 27(11): 96-97.
- [4] 申晓龙,许文雨. Windows 钩子技术的研究与应用[J]. 成都信息工程学院, 2005, 20(4): 380-382.
- [5] 余姜德,于志平. Windows 钩子技术在病毒程序中的应用[J]. 现代计算机, 2005(2): 83-86.
- [6] 李永革,张维明. 基于浏览器的安全问题研究[J]. 计算机工程, 2002, 28(1): 55-56.
- [7] Grimes R A. 恶意传播代码: Windows 病毒防护[M]. 北京: 机械工业出版社, 2004.
- [8] Beveridge J, Wiener R. Win32 多线程程序设计[M]. 侯 捷译. 武汉: 华中科技大学出版社, 2002.
- [9] Richter J. Programming Application for Windows[M]. US: Microsoft Press, 1999.
- [2] Adar E, Huberman B. Free riding on Gnutella[R]. [s. l.]: Xerox PARC, 2000.
- [3] Feldman Y, Laizk. Quantifying disincentives in peer-to-peer networks[C]//workshop on economics of peer-to-peer systems. CA: Springer-verlag, 2003: 117-122.
- [4] Yang B, Molina H G. Micropayments for peer-to-peer systems[C]//Proceeding of the 10th ACM Conference on computer and Communications Security. Washington: ACM press, 2003: 300-310.
- [5] Ioannidis J, Ioannidis S, Keromytis A D, et al. Fileteller: paying and getting paid for file storage[C]//In Proceedings of the Sixth Annual Conference on Financial Cryptography. Bermuda: Springer-Verlag, 2002: 282-299.
- [6] Adya A, Bolosky W J, Castro M, et al. Federated, available, and reliable storage for an incompletely trusted environment [C]//In Proceedings of the 5th Symposium on Operating Systems Design and Implementation. Boston: ACM Press, 2002: 12-14.
- [7] Cox L P, Murray C D, Noble B D. Pastiche: making backup cheap and easy[C]//In Proceedings of the 5th Symposium on Operating Systems Design and Implementation. Boston: ACM Press, 2002: 285-298.
- [8] 宿建宗,李秉智. P2P 文件共享框架中激励机制的研究[J]. 重庆邮电学院学报, 2006, 18(1): 123-125.
- [9] 陈志琦,苏德富. 基于博弈论框架的 P2P 激励模型[J]. 计算机工程, 2005, 31(16): 118-121.
- [10] 曾宇光,陈志刚. JXTA 路由改进算法的研究与实现[J]. 微计算机信息, 2007, 33: 223-225.

(上接第 8 页)