

面向视频应用的 BitTorrent 协议优化

李敏宁^{1,2}, 鱼 滨¹, 杨 力¹

(1. 西安电子科技大学 计算机学院, 陕西 西安 710071;
2. 渭南职业技术学院, 陕西 渭南 714000)

摘 要: BitTorrent 协议是一种基于 P2P (Peer to Peer) 的文件共享协议, 具有“用户越多, 下载速度越快”的特点, 尤其是针对大文件的下载, 更能体现出系统的优越性。视频媒体流是近年来网络用户占有量最多的一类文件, 随着消费者带宽的进一步增加, 越来越多的人希望得到像视频点播这样的多媒体应用。传统的 C/S 架构在当用户数目急剧增加时, 容易产生服务器过载现象, 使用服务器集群或分布式系统成本增大。BitTorrent 能够较好地降低服务器负担, 提高系统的扩展性和健壮性。但是 BitTorrent 针对普通大文件的共享设计, 对文件的片段选择是无序进行的, 不能有效地支持视频大文件的点播和直播。文中针对视频大文件的点播需求, 对 BitTorrent 协议的片段选择算法进行优化, 实现了视频文件的点播和直播。

关键词: BitTorrent 协议; P2P 技术; 片段选择算法; 视频流

中图分类号: TP31

文献标识码: A

文章编号: 1673-629X(2015)11-0115-05

doi: 10.3969/j.issn.1673-629X.2015.11.023

Optimization on BitTorrent Protocol for Video Application

LI Min-ning^{1,2}, YU Bin¹, YANG Li¹

(1. College of Computer, Xidian University, Xi'an 710071, China;
2. Weinan Vocational & Technical College, Weinan 714000, China)

Abstract: BitTorrent protocol is a file sharing protocol based on P2P (Peer to Peer), with "the more users the download faster" feature. It reflects the superiority of the system especially for large files. In recent years, video media streams are a class file that the most amounts of Internet users occupied. With the further increase of consumer bandwidth, more and more people want to get the multimedia applications such as video on demand. The traditional C/S architecture, when a sharp increase is conducted in the number of users, prone to overload the server, using the server cluster or distributed system to cause the cost increases. BitTorrent can better reduce server load and improve the scalability and robustness of the system. But BitTorrent is designed for ordinary share large files, selecting the file fragments are disorder conduction, and cannot effectively support on-demand and live video for large files. In this paper, for video-on-demand needs of large files, fragment selection algorithm is optimized of BitTorrent protocol, realizing on-demand and live of video files.

Key words: BitTorrent protocol; P2P technology; piece selection algorithm; video stream

1 概 述

目前, 宽带音视频业务发展迅速, 应用广泛, 例如远程教学系统、视频点播系统、视频会议系统等等。因此, 大幅度降低此类业务的建设成本是目前需要研究的基本问题之一。

流媒体 (Streaming Media) 是一种音/视频网络传输技术, 是在互联网上实时顺序地传输和播放音/视频等多媒体内容的连续数据流。流媒体技术包括流媒体数据采集、音/视频编码、解码、存储、传输、播放等

领域^[1]。

从技术角度来说, 流媒体传输可分顺序流媒体传输和实时流媒体传输两种, 其中流媒体传输的效率受媒体播放速率、当时的网络状况、缓存大小等因素的影响^[2]。

顺序流媒体传输是指对流媒体格式文件的顺序下载, 在下载文件的同时用户可以观看, 但不支持回放或快进, 通过这些功能的限制来减少网络的抖动。

实时流媒体传输指观看过程中, 用户可以快进或

收稿日期: 2015-03-18

修回日期: 2015-06-19

网络出版时间: 2015-11-04

基金项目: 国家自然科学基金资助项目 (61172147, 61202390)

作者简介: 李敏宁 (1975-), 女, 硕士, 讲师, 研究方向为计算机技术; 鱼 滨, 博士, 教授, 硕导, 研究方向为软件工程。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20151104.0955.104.html>

后退。这一功能应用于网络播放,对技术的要求相当高,如果只是简单在顺序播放的基础上增加这一功能,频繁的拖动将会使服务器不堪重负,势必需要更多的技术途径来解决^[3]。

网络数据流的需求特征可以归结为四个参数:可靠性、延迟、抖动和带宽。这四个参数决定了一个数据流所要求的服务质量^[4]。

由此可以看出,文件共享的应用要求与网络视频服务要求有很大的不同,文件共享需要严格的可靠性,而对网络的延迟、抖动几乎没有要求,对带宽的要求也不是特别高。但网络视频不同,它是一种实时性的服务,可以容忍错误,但对网络的延迟、抖动和带宽非常敏感。

近年来,P2P 的强大技术和应用潜力引起了业界越来越多的关注,合理应用 P2P(Peer to Peer)技术有可能使建设成本下降一个数量级,从而有利于节约教学成本,提高教学效率。BitTorrent 是近些年应用很广的一种 P2P 软件^[5]。

BitTorrent 通过 Web 发布共享文件的 .torrent 信息,Tracker 服务器帮助 Peers 间相互连接,使用(Tit-for-Tat)方式让 Peers 在下载的同时互相传递数据。由于 P2P 的特性,BitTorrent 使用与 C/S 模式完全不同的工作方式,使其在文件传输上不再受服务器带宽的限制,尤其在共享大文件时,它比 C/S 模式的优势更大。

通过分析 BitTorrent(以下简称 BT)协议发现,要保证文件的快速传输,采取了对等发布、流水作业、片段选择算法和阻塞算法等技术^[6]。但这些技术是针对快速传输普通文件设计的,依据它的片断选择算法策略,所下载的文件片断没有顺序,不适合传输多媒体流。通过修改 BT 协议的片段选择算法,改变片断的选择顺序,可以解决这个问题。

2 BT 协议中的片段选择算法分析

在 BT 源代码^[7]中,有一个 PiecePicker.py 文件,文件中定义了一个 PiecePicker 类,它的功能就是完成片断选择算法。在 PiecePicker 类中,最核心的函数是 next()。

它综合片段选择算法的四种策略,来计算出下一个被选择的下载片段。其中,变量 started 和 seedstarted 是实现严格优先级的,它们记录了已经开始下载的片段,而变量 numinterests 是用来实现最少优先的^[8-9]。具体代码如下:

```
#计算下一个被选择的片段。
defnext(self, havefunc, seed=False):
    bests=None
```

```
bestnum=2 * * 30
#根据严格的优先级策略,从已经开始下载的片段中选择。
if seed:
    s=self.seedstarted
else:
    s=self.started
#“严格优先”策略是和“最少优先”策略结合使用的。也就是说,在满足“严格优先”的片断中,再去选择一个满足“最少优先”的片断。“最少优先”还有一个限制,就是如果从来没有收到过 have 消息,也不能被选择。havefunc(i)判断片段是否收到过 have 消息,若否,则不被选择。

for i in s:
    if havefunc(i):
        #“最少的优先级”策略。
        if self.numinterests[i] < bestnum:
            bests=[i]
            bestnum=self.numinterests[i]
        elif self.numinterests[i] == bestnum:
            bests.append(i)
#经过“严格优先”和“最少优先”策略之后,可能返回多个候选片断,随机选择一个并返回。

if bests:
    return choice(bests)
#若是经过以上步骤,仍未选择出一个片断,那就随机选择一个。即“随机的第一个片断”策略。self.config['rarest_first_cutoff']的默认值已经由 3.3 版的 1 改为 4.03 版的 4,“随机的第一个片断”也应该叫做“随机的前 4 个片断”。

if self.numgot<self.config['rarest_first_cutoff']:
    for i in self.scrambled:
        if havefunc(i):
            return i
    return None
#若是拥有的片断已经比较多了,就由最后一段程序来选择片断,还是最少优先的思想。“最后阶段模式”主要针对的是子片断,所以这里没有提及。

for i in xrange(1, min(bestnum, len(self.interests))):
    for j in self.interests[i]:
        if havefunc(j):
            return j
#若仍然选不出来,返回 None。
return None
```

PiecePicker 类在 download.py 中被实例化。首先是初始化,然后把自己已有的片断告诉给 complete(),以后就不要从中选择了。再者,当一个 SingleDownload 对象获取对方片断拥有状况的位图时,交由 got_have()来处理。需要进行片断选择时就调用 next()方法,选出下一个需要下载的片断^[10]。

3 算法的改进

文中把 peer 的上传速度分为 4 类,使最大上传速

度除以每个 peer 的上传速度,大于等于 8 的为第 3 类,大于等于 4 的为第 2 类,大于等于 2 的为第 1 类,最快的为第 0 类。第一步先对已经连接的 peer 测速,改变 next() 函数中的最后一段代码,伪代码如下:

```
for i in xrange(len(self.connections)):
    c=self.connections[i]
    if c.upload. interested and not c.download.is_snubbed():
        preferred.append((c.download.get_rate(), i))
preferred.sort()
pmax=max(preferred)
pclass=[0]*len(preferred)
for i in range(len(preferred)):
    if pmax/preferred[i]>=8:
        pclass[i]=3
    elif pmax/prefreed[i]>=4:
        pclass[i]=2
    elif pmax/prefreed[i]>=2:
        pclass[i]=1
    elif pmax/prefreed[i]>=1:
        pclass[i]=0
```

原程序中有一个列表变量 numinterests 记录了当前每个片断由几个 Peers 拥有,初始值为[0,0,...,0],它与另外两个变量 interests 和 pos_in_interests 合起来共同实现最少优先策略。这样选到的片断的顺序是随机的^[11-12]。

现在增加一个变量 width,它是一个宽度,用来限定目前需要下载的片断的范围,初始值为[0,1,2,3,4,5,6,7],即优先下载 0~7 号片断,每当完成一个就在 complete(self, piece)方法中改变它的值。它和前面几个变量一起,共同决定需要下载的片断。伪代码如下:

```
i=2*3-1
dmax=2*3
m=len(preferred)#可以提供片断的 peer 的个数
d=[2*3]*m
ps=list()
for i in range(m):
    ps.append([])
    while i>=0:
        for j in range(m):
            if d[j]==dmax:
                ps[j].append(window[i])
                i-=1
        d[j]=d[j]-2*pclass[j]
    dmax=max(d)
```

4 新旧算法分析

从源代码的分析可知,BT 的片段选择算法的优先策略主要由两部分组成,即“随机的前 4 个片段”算法和“最少优先”算法。将片段用 N 表示,设总片段数为

m ,则所有片段可以表示为:

$$N = \{N_1, N_2, \cdots, N_m\}$$

那么原算法经过随机选择在 N 个片段中先任意选择出 4 个片段进行下载,假设为 $N_{10}, N_{16}, N_{80}, N_{101}$,第二步才开始最少优先策略。假设第二次选择片段 N_7 ,则此时的下载片段为:

$$N = \{0, 0, 0, 0, 0, 0, N_7, 0, 0, N_{10}, 0, 0, 0, 0, 0, N_{16}, 0, \cdots, 0, N_{80}, 0, \cdots, 0, N_{101}\}$$

其中,0 表示未被加载的片段,即断点。

经过一段时间后,整个视频文件出现大面积被下载,但断点很多。如果播放,就会出现马赛克现象。

可以看出,在整个片段选择过程中,从不考虑片段的顺序,只追求片段选择的效率^[10-11]。显然,这种选择不适用于视频文件的播放。

修改后,系统先对已连接的 Peers 进行测速,选出速度最快的 Peers,将整个视频文件的前 8 个片段分配给这些 Peers,然后再执行最少优先策略,则此时的下载片段变成:

$$N = \{N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8, 0, \cdots, 0\}$$

显然,虽然短时间内视频文件没有下载完成,但前面的片段优先被下载,可以连续播放,保证了视频文件按顺序播放,有效地实现了边看边播的目的。完成了协议优化的目的。

但是,应该认识到,片断选择算法改进之后,虽然更适合于视频应用,降低了视频的延迟时间,但片断的分散性不如以前的系统,这对下载速度也会产生一定的影响。

5 模拟实验及结果分析

在算法基础上,搭建实验平台,对 BT 原协议和新协议编译后置于局域网实验环境中进行验证。对比两种实验效果,收集并分析实验数据,得出结论。

5.1 实验环境

网络中共有 17 台主机,其中,1 台为服务器,1 台做种子节点,其余 15 台为下载节点客户机。在 BT 网络中,由于每个节点加入网络的时间不同,节点所处的带宽和节点自身的配置不同,被共享文件分割成的片段也不相同。假设有 8 个下载节点加入网络,片段分布如下:

$$[x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8] = [N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8]$$

其中, x 代表节点上所拥有的片段数; N 代表种子文件分割成的片段数。

实验网络的拓扑结构如图 1 所示。

硬件环境:网络仿真实验室,服务器: Intel® Celeron® CPU2.66 GHz/2.67 GHz,1 GB 内存;客户机:

Intel® Core™ i5-3470 CPU @ 3.20 GHz 3.20 GHz, 4 GB 内存。

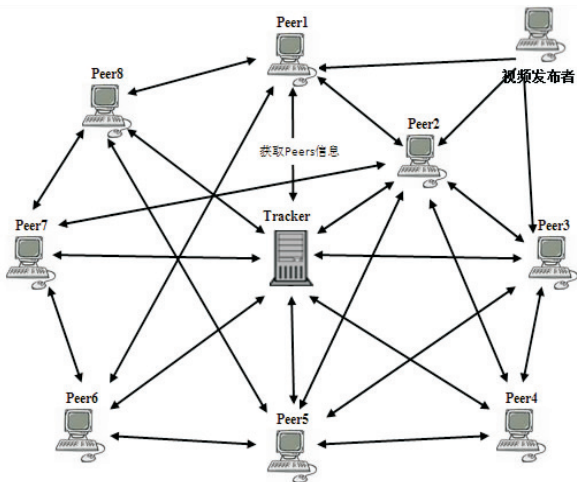


图1 实验网络的拓扑结构

软件环境：

服务器: Windows Server 2008 Enterprise; 客户机: Windows 7 Professional。

在安装“BT 服务器”前, 需要利用以下几种软件配置服务环境:

Python 软件包: Python3.4.3; BitTorrent 服务器程序: MyBT3.0; Web 服务器: 使用 Windows Server 2008 自带的 Web 服务器; 发布制作工具 Bitcomet1.38; 该工具用来制作“.torrent”文件, BT 服务器就是通过此文件来实现资源共享的; BT 下载工具: BitTorrent4.4.0; 视频播放软件: 奇速影音(支持 BT 种子的播放软件)。

服务器 IP 地址: 192.168.44.200; 服务器端口: 6969; 实验用视频文件: 电视剧《三国演义》第27集《三顾茅庐》(154.2 MB), 动画片《汽车总动员》(548.91 MB), 网络教育视频(414.67 MB)等, 如图2所示。通过收集和分析不同大小、不同类型的视频文件在相同的实验环境中的实验数据, 达到实验目的。

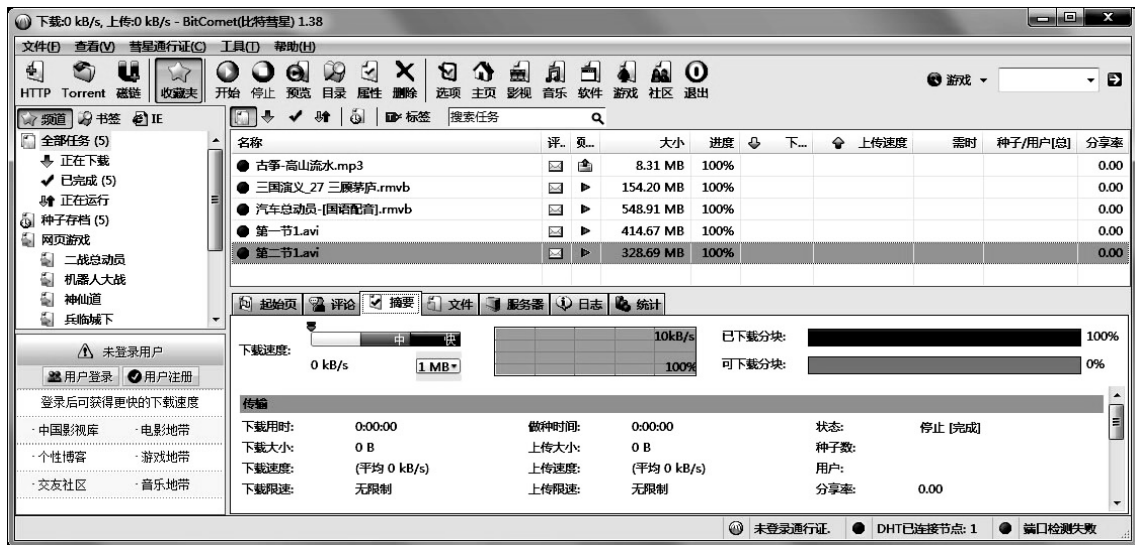


图2 实验用视频文件

实验的主要目的有两个:

- (1) 修改前视频的播放是不是断点播放、马赛克严重, 以验证文中协议改进的可行性;
- (2) 协议修改后视频的播放是不是有序播放, 且播放速度不受影响, 以证明协议优化后优于原协议。

5.2 实验过程

(1) 原协议的实验结果。

第一步, 打开服务器上的 MyBT3.0, 进行必要的设置。

第二步, 启动 Bitcomet, 选择“三国演义_27 三顾茅庐.rmvb”, 生成一个种子文件。

第三步, 登录服务器端口 <http://192.168.44.200:6969/>, 注册用户, 上传种子文件, 等待其他 Peer 下载。

第四步, 启动 BitTorrent 客户端, 通过端口 6969 登

录服务器, 选择刚刚制作的三国演义_27 三顾茅庐.rmvb.torrent, 下载种子, 然后请求下载, 与种子提供者联系, 下载源文件, 然后自己成为新的种子, 等待其他节点下载。

第五步, 开始共享文件下载。

启动局域网内其他两台计算机, 计算机 A 的 IP 为 192.168.44.161, 计算机 B 的 IP 为 192.168.44.162, 均安装 BitTorrent 客户端。先后利用三国演义_27 三顾茅庐.rmvb.torrent 下载电影《三国演义》之《三顾茅庐》。

A 机客户端运行后, 很快与 tracker 服务器通信, 接着种子 Peer 发现了 A 并向其传送数据片断, Peer A 开始接收数据。

第六步, 播放正在下载的电影文件。

在 A 机上打开播放器, 播放正在被 BitTorrent 下载

的三国演义_27 三顾茅庐.rmvb 文件,此时由于数据过少,文件格式不能被迅雷识别。当下载到 34% 的时候,文件可以被播放,播放效果如图 3 所示。



图 3 改进前的播放效果

通过以上实验可以看出,BitTorrent 能够出色地完成大文件的下载任务,调整种子和其他 Peers 的上传速度,能够很快地提高下载速度。但若是用于视频直播,片断选择的随机性导致了播放效果很不理想,一是时间延迟比较长,二是由于电影文件中有空洞而导致马赛克现象。

通过 NS2 模拟外网实验,能得出相似的结果,这里不再赘述。

(2) 改变片段选择算法后的实验结果。

对源代码的片断选择算法做了改动之后,重新编译 BT 协议生成新的 BT 客户端,然后仍然按上面的方案进行实验,结果虽然不是特别理想,但明显比原效果要好一些。同样在 A 机上的播放效果如图 4 所示。



图 4 改进后的播放效果

5.3 实验现象分析

局域网的线路状况比较好,Peers 也比较稳定,所以改进前后的上行和下载速度一致。由于原始的 BitTorrent 协议是针对无序大文件设计的,虽然下载速度快,但是其片断选择算法的片断选择顺序是随机的,导致电影播放中时延长、马赛克现象频发甚至发生停止

播放^[13-14]。而优化后的 BitTorrent 协议着重修改了片断选择算法中的片段选择顺序,将大文件的片段利用变量分组,按照分组的顺序兼顾组内片段的速度,使得视频文件达到边播边下的目的,所以时延缩短,在下载开始后不久就可以播放了。

同时应该认识到片断选择算法在 BT 系统中的重要性,原始 BitTorrent 协议的片断选择算法采用最少优先策略,保证了系统中稀少的片断优先下载,从而使整个系统更健康的运行,不至于某一个片断出现空缺现象,最终导致下载失败。如果整个系统中新的 BT 客户端很多,一定会对系统产生影响。

在 BT 客户端的行为发生转变—从文件下载变为视频播放之后,系统中种子的数量会减少。因为这两种行为之间有很大的不同。文件下载时用户即使在线,下载过程也是放在后台运行,当用户再次关注甚至直接离开的时候,可能已经下载完成并保持在线做种很长时间了,这对整个系统是非常有益的。因为系统中种子越多性能越好,而转变为视频播放后,运行模式变成前台运行。用户观看完一个视频后常常会关闭视频环境,导致退出 BT,种子数随之减少。

6 结束语

通过分析流媒体技术和用户对视频文件下载的特殊需求,对比流媒体服务与 BitTorrent 大文件分发系统的不同,BitTorrent 协议中的片段选择算法是针对普通大文件设计的,下载时为了追求速度,设计为无序下载,而视频文件的直播和点播需要快速有序的下载支持。针对 BitTorrent 的片段选择算法,提出了优化设计方案。一方面,加入测速算法,计算出网络中速度最快的节点;另一方面,对协议进行重新设计,改变原算法的无序断点下载,采用一个新的宽度变量实现有序下载。

搭建实验环境,开发相应的原型系统,对算法和协议的实际效果进行测试和验证。结果表明,新算法能够有效减少播放时的断点现象,更加适用于视频点播和直播。

参考文献:

- [1] 张蓉蓉. P2P 点播流媒体传输机制研究及仿真[D]. 成都:电子科技大学,2011.
- [2] 于樊鹏,沈海利. 基于 P2P 应用组播技术视频系统在 IPv6 下的研究与实现[J]. 计算机科学,2012(S2):19-22.
- [3] 闫 丁. P2P 直播流媒体系统的性能测量与优化设计[D]. 上海:上海师范大学,2011.
- [4] 周琼枝. P2P 网络流媒体性能分析优化研究[D]. 广州:华南理工大学,2013.

部分性能指标上都占有优势,显示出良好的优越性。利用 NCGA 进行函数优化所得到的函数均值,也非常接近函数的最优值,显示出良好的稳定性。

5 结束语

由于紧致遗传算法进化机制的优越性,已经得到越来越多的关注,文中在研究国内外紧致遗传算法的基础上,提出了一种新型的紧致遗传算法,以及该算法的详细步骤。针对紧致遗传算法收敛性及收敛速度等方面研究内容较少,文中提出了一些对其收敛性方面分析的方法。最后利用所提出的新型紧致遗传算法进行了函数的仿真分析,证明了算法的正确性。

紧致遗传算法作为一种新兴进化算法,还属于单变量低阶的概率模型进化算法,因此下一步的工作可考虑与高阶概率模型进化算法进行良好的交互和融合,提高其全局寻优性。

参考文献:

- [1] Harik G R, Lobo F G, Goldberg D E. The compact genetic algorithm[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(4): 287-297.
- [2] Amr B, Ibtehal M, Basma M, et al. Solving protein folding problem using elitism-based compact genetic algorithm[J]. Journal of Computer Science, 2008, 4(7): 525-529.
- [3] Soheil R, Hadi A, Guy V. A simple real-coded compact genetic algorithm and its application to antenna optimization[C]//Proceedings of Asia-Pacific microwave conference. [s. l.]: [s. n.], 2007.
- [4] 刘 振, 胡云安, 彭 军. 协同进化扩展紧致量子进化算法[J]. 控制与决策, 2014, 29(2): 320-326.
- [5] 叶 苗, 程小辉. 改进的紧致遗传算法求解族状旅行商问题[J]. 微电子学与计算机, 2013, 30(8): 7-12.
- [6] 宋 波. 基于紧致遗传算法的 N 次检测测试集压缩[J]. 牡丹江师范学院学报: 自然科学版, 2012, 20(3): 6-7.
- [7] Droste S. A rigorous analysis of the compact genetic algorithm for linear functions[J]. Natural Computing, 2006, 5(3): 257-283.
- [8] 刘 振, 史建国, 高晓光. 紧致遗传算法及其在武器目标分配中的应用[J]. 计算机工程与应用, 2008, 44(30): 229-231.
- [9] 杨有龙, 高晓光. 紧致遗传算法的进化机制分析[J]. 控制理论与应用, 2003, 20(3): 415-418.
- [10] Seok J H, Lee J J. A novel compact genetic algorithm using offspring survival evolutionary strategy[J]. Artificial Life and Robotics, 2009, 14(4): 489-493.
- [11] Ahn C, Ramakrishna R. Elitism-based compact genetic algorithms[J]. IEEE Transactions on Evolutionary Computation, 2003, 7(4): 367-385.
- [12] 张文修, 梁 怡. 遗传算法的数学基础[M]. 西安: 西安交通大学出版社, 2003.
- [13] Rastegar R, Hariri A. A step forward in studying the compact genetic algorithm[J]. Evolutionary Computation, 2006, 14(3): 277-289.
- [14] Liu Z, Hu Y A. Estimation of distribution immune genetic algorithm and its convergence analysis[J]. TELKOMNIKA: Journal of Electrical Engineering, 2013, 11(1): 123-129.
- [15] Chen T S, He J, Sun G, et al. A new approach for analyzing average time complexity of population-based evolutionary algorithms on unimodal problems[J]. IEEE Transactions on Systems, Man, Cybernetics, Part B: Cybernetics, 2009, 39(5): 1092-1106.

(上接第 119 页)

- [5] 郑纬民, 胡进锋, 代亚非, 等. 对等计算研究概论[J]. 中国计算机学会通讯, 2005, 7(2): 38-51.
- [6] 张景伟, 田馨铭. 利用 BT 技术设计视频点播系统[J]. 电脑知识与技术, 2009, 5(12): 3262-3264.
- [7] 王裕邦, 卢显良, 段翰聪, 等. BT 流量控制系统的设计与实现[J]. 计算机应用研究, 2007, 24(9): 214-216.
- [8] 夏辉宇, 孟令奎, 娄书荣. BitTorrent 的影像流式传输模型研究[J]. 测绘学报, 2013, 42(2): 225-232.
- [9] Cohen B. Incentives build robustness in BitTorrent[C]//Proc of workshop on economics of peer-to-peer systems. [s. l.]: [s. n.], 2003: 68-72.
- [10] Wang H, Liu J, Xu K. Measurement and enhancement of BitTorrent-based video file swarming[J]. Peer-to-Peer Networking and Applications, 2010, 3(3): 237-253.
- [11] Qiu D, Srikant R. Modeling and performance analysis of BitTorrent-like peer-to-peer networks[J]. ACM SIGCOMM Computer Communication Review, 2004, 34(4): 367-378.
- [12] Xu D, Hefeeda M, Hambrusch S, et al. On peer-to-peer media streaming[C]//Proceedings of 22nd international conference on distributed computing systems. [s. l.]: IEEE, 2002: 363-371.
- [13] 苗文健, 杨雅辉. 面向视频播放的 BT 协议下载算法的改进[J]. 计算机工程与应用, 2010, 46(18): 68-70.
- [14] 刘宏亮. BitTorrent 核心算法研究与改进[D]. 北京: 北京交通大学, 2008.

面向视频应用的BitTorrent协议优化

作者：[李敏宁](#), [鱼滨](#), [杨力](#), [LI Min-ning](#), [YU Bin](#), [YANG Li](#)

作者单位：[李敏宁, LI Min-ning\(西安电子科技大学 计算机学院, 陕西 西安710071; 渭南职业技术学院, 陕西 渭南714000\)](#), [鱼滨, 杨力, YU Bin, YANG Li\(西安电子科技大学 计算机学院, 陕西 西安, 710071\)](#)

刊名：[计算机技术与发展](#) 

英文刊名：[Computer Technology and Development](#)

年, 卷(期): 2015, 25(11)

引用本文格式: [李敏宁](#). [鱼滨](#). [杨力](#). [LI Min-ning](#). [YU Bin](#). [YANG Li](#) [面向视频应用的BitTorrent协议优化](#)[期刊论文]

-[计算机技术与发展](#) 2015(11)