软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(6):1516−1533 [doi: 10.13328/j.cnki.jos.004625]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种抵抗搭便车行为的概率式连接交换 unchoking 策略*

李治军, 姜守旭, 李晓义

(哈尔滨工业大学 计算机科学与技术学院,黑龙江 哈尔滨 150001)

通讯作者: 李治军, E-mail: lizhijun_os@hit.edu.cn

摘 要: BitTorrent 文件共享系统中的搭便车(free-riding)节点会使系统性能大幅下降,目前,BitTorrent 主要采用 choking 策略来抑制搭便车行为,但与 choking 合作而存在的随机选择节点的 unchoking 策略仍然给搭便车行为提供了机会.提出了一种基于概率连接交换(probabilistic link exchange,简称 PLX)的 unchoking 策略,在实现 unchoking 功能的同时,有效地抑制了搭便车行为.由于搭便车节点不提供上传,所以 choking 以后没有指向搭便车节点的连接,此时, PLX 的连接交换机制就能抑制搭便车节点进入文件共享系统.另外,通过对连接交换概率的数学控制,PLX 可以区分节点对共享系统的贡献,并根据贡献大小调整其在共享网络中的位置,进一步保证了公平性.最后,对 PLX 的影响进行了深入的理论分析和模拟实验验证,结果表明:PLX unchoking 策略较现有的抵抗搭便车的方法更简单、直接,在效果上有明显提升.

关键词: BitTorrent;搭便车;unchoking 策略;连接交换;概率算法

中图法分类号: TP393

中文引用格式: 李治军,姜守旭,李晓义.一种抵抗搭便车行为的概率式连接交换 unchoking 策略.软件学报,2015,26(6):1516−1533. http://www.jos.org.cn/1000-9825/4625.htm

英文引用格式: Li ZJ, Jiang SX, Li XY. Unchoking scheme based on probabilistic link exchange to resist the free-riders. Ruan Jian Xue Bao/Journal of Software, 2015,26(6):1516−1533 (in Chinese). http://www.jos.org.cn/1000-9825/4625.htm

Unchoking Scheme Based on Probabilistic Link Exchange to Resist Free-Riding

LI Zhi-Jun, JIANG Shou-Xu, LI Xiao-Yi

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

Abstract: Free-riding destroy the foundation of BitTorrent file sharing, and result in bad system performance. The choking scheme adopted in BitTorrent nowadays can suppress the free-riding, however the coexisting unchoking scheme in which the random peers are chosen lend opportunities to free-riders. An unchoking scheme based on probabilistic link exchange, or PLX for short, is provided in this paper. The new scheme can suppress the free-riding effectively while guaranteeing the unchoking function. Free-riders can't enter into the system because they will not be unchoked by PLX after the links to free-riders are choked as PLX works based on link exchanges. Furthermore, by virtue of the mathematical designs for the probability of the link exchange, PLX can distinguish the contribution of peers, adjust their location in network according to contribution, and improve the fairness of the system. The in-depth theoretical analyses and experimental evaluations show that comparing with other methods for fighting against free-riding attacks, the PLX unchoking scheme is simple, direct and effective..

Key words: BitTorrent; free-riders; unchoking scheme; link exchange; probatilistic algorithm

P2P 文件共享系统 BitTorrent 采用的文件传输过程为:

(1) 文件被分成若干块(chunk);

(2) 下载节点(leechers)首先从 seed 节点上下载到部分文件块;

* 基金项目: 国家自然科学基金(61370214)
收稿时间: 2011-02-16; 定稿时间: 2014-01-24

(3) 下载节点从别的下载节点那里寻找所需的块下载,同时给别的下载节点上传其所需的块.

此时,下载同一个文件的下载节点就形成按 peer-to-peer(简称为 P2P)方式互相传输数据(文件块交换)的集合,形成一个 file swarming 系统.系统中每个节点在下载的同时也能给其他节点提供上传,系统的服务能力会随着下载节点的增加而增长,不存在传统 ftp 中服务器过载的现象,特别适合于大规模文件共享[1].但对于完全自治的 P2P 环境而言,节点可以不遵守上述文件块交换协议,可以选择只下载而不上传的搭便车行为(free riding).当系统中存在大量的搭便车节点(free riders)时,BitTorrent 节点间的传输就无法形成.

因此,BitTorrent 文件传输协议中需采用激励机制来抑制搭便车行为,其激励机制由 tit-for-tat 和 optimistic unchoking 两个上传节点选择策略组成.tit-for-tat 是激励机制中的 choking 策略:节点选择那些给该节点提供最大下载带宽节点进行上传,这就使提供上传成为得到下载的前提条件,由于搭便车者没有贡献上传带宽,所以搭便车节点会被 choking(阻塞).但如果只运行 tit-for-tat 策略,节点间的上传关系很快就会固定:带宽接近的一对节点互相上传对方没有的数据,当某一方没有对方需要的文件块时,上传关系停止.为解决这一问题,需要有 unchoking 策略来配合,BitTorrent 采用 optimistic unchoking 完成这一功能.节点在执行 optimistic unchoking 时,会随机选取一个节点提供上传.

但正是 optimistic unchoking 策略,使 BitTorrent 非常容易受到搭便车攻击.如,Sirivianos 等人在文献[2]中给出了一种搭便车攻击:构建的搭便车节点从 tracker 那里获得比普通节点更大量的下载节点(leechers)信息,然后,该搭便车节点与这些下载节点都建立连接(所以,该文将其称为 large view exploit).这样,与该搭便车节点连接的下载节点 unchoking 到该节点的概率会变大,搭便车节点就可以不用上传而获得下载.文献[2]得到的实际测量结果是:当一个搭便车节点和一个诚实节点下载同样的文件时,15 次实验中有 12 次是搭便车节点有更小的文件下载饿时间.实际上,如果设搭便车节点 f 保持的连接个数$|l(f)|=n$,诚实节点 h 保持的连接个数$|l(h)|=m$,那么对于现在的 optimistic unchoking 而言,每个 f 的邻居节点 unchoking 到 f 的概率为 $1/m$,如果 BitTorrent 同构,则节点 f 收到总下载带宽的均值为 $d(f)=(n/m)\times(u/5)$(其中,u 是上传带宽),对于节点 h 而言,$d(h)\leqslant(4u/5)+(m/m)\times(u/5)=u$,因此有:

$$d(f)/d(h)\geqslant((nu)/(5m))/u=n/(5m) \tag{1}$$

当 $n\geqslant 5m$ 时,搭便车节点较诚实节点将获得更多的下载服务.另外,即使对于异构系统(设节点分为两类,其上传带宽分别为 u_h 和 u_l),此时 $d(f)=(n/m)\times(E[u]/5)$,$d(h)\leqslant(4u_h/5)+(m/m)\times(u_h/5)=u_h$,于是有:

$$d(f)/d(h)\geqslant((nE[u])/(5m))/u_h=n/(5m)\times(E[u]/u_h) \tag{2}$$

当 $n\geqslant O(5m)$时,搭便车节点较诚实节点仍将获得更多的下载服务,故 large view exploit 的攻击复杂度很低.

除了简单 large view exploit 以外,对 BitTorrent 还可构造更精巧的搭便车攻击,如 BitThief[3]和 BitTyrant[4].如:BitTyrant 不仅控制(shaping)其活动连接的数量,还会给这些连接适当的分配一些上传带宽,且 BitTyrant 会给出夸大的文件块拥有信息(over-reporting)来增大其他节点 unchoking 到 BitTyrant 的概率;另外,文献[5]还给出了利用 Sybil 攻击[6]来增大 view exploit 的方法,等等.这些机制会使 BitTorrent 中的搭便车攻击更加容易、有效.目前,抑制搭便车攻击的研究结果根据其采用的手段主要分为基于支付(payment)[7,8]、基于名誉(reputation)[9]、基于文件块交换(barter)[10,11]和基于拓扑调整(topology)这 4 类[12−14].

在基于支付(payment)的系统中,虚拟现金(virtual money)会控制服务的提供和消费:一个节点在给系统提供服务(如上传文件)后会赚取一定数量的虚拟现金,当该节点需要消费其他节点提供的服务时(如从别的节点那里下载文件)需要支付(这就是 payment 的核心)一定数量的虚拟现金.由于不向系统提供服务的搭便车者不会获得虚拟现金,也就不可能消费其他节点的服务,从而抑制了搭便车攻击.但该方法存在明显的缺陷:虚拟现金和真实的货币具有类似的特点,即,需要一个权威机构产生,否则很容易造成共谋欺骗.因此,Dandelion[7]支付系统需要一个中央服务器来产生和维护虚拟现金.显然,这样的系统不适合象 P2P 这样的大规模分布式系统.Tan 等人给出的方法也存在同样的问题[8].

名誉(reputation)方法处理搭便车攻击的思想实际上和支付类似,都是根据对节点的评价(estimation)来决定是否给予其服务,只是支付是节点自身持有这一评价,而名誉是一种被其他节点持有的评价.在处理服务申请的

竞争时会优先给予名誉高的节点,所以当一个搭便车节点向 BitTorrent 中的一个节点发出文件块下载请求时,会因为名誉最低而不能得到文件块下载服务,从而抑制了搭便车攻击.如 Kang 等人在文献[9]中给出的 credit 就实现了这样的效果.与支付系统一样,名誉值的管理也很容易受到攻击:虽然搭便车节点不能象支付系统一样自身提高名誉,但是多个搭便车节点完全可以通过共谋来互相提高名誉.

支付和名誉都是依靠数值来确定节点对系统的贡献,然后根据这个数值区分诚实节点和搭便车节点.前面的分析表明:这个数值对搭便车攻击并不具有明显的优势,而又会引出新的安全问题,因此在区分诚实节点和搭便车节点时不应该引入数值,而依靠实物交换(即,服务交换或文件块交换),这就是 barter 系统的核心思想.如,文献[10]提出的也是一种被称为 Quota-Encryption 的文件块交换协议,文献[11]中提出的带宽按比例分配也是一种文件块交换(带宽乘以时间就是传输数据的大小)模型.较 tit-for-tat 的带宽交换而言,更为公平的文件块交换必然可以抑制搭便车攻击,但这些交换协议仍然需要 unchoking.文献[10,11]的 unchoking 策略仍然是随机 optimistic unchoking,此时,搭便车攻击仍然可以利用 large view exploit 进行.虽然 Quota-Encryption[10]通过加密使得通过该漏洞获得数据不可用,可以适当抑制搭便车行为,但会引入大量额外负载.

因此,需要一个新的 unchoking 策略从根本上弥补 optimistic unchoking 的漏洞.本文提出的连接交换(link exchange,简称 LX)正是一个满足上述条件的 unchoking 策略.LX 的基本思想是:

(1) 设节点 p 上传连接指向的节点集合为 $N^+(p)$,节点 p 定期地 choking 集合 $N^+(p)$,当然,choking 算法不是本文关注的,所以可以直接使用 tit-for-tat;

(2) 系统中任意两个节点 p 和 q,随机选取 $c_1 \in N^+(p), c_2 \in N^+(q)$,$p$ 将 c_2 加入 $N^+(p)$中,q 将 c_1 加入 $N^+(q)$中.

首先,这个随机的上传连接增加过程可以完成 unchoking 功能;同时,由于交换的是节点 choking 以后的上传连接,搭便车节点在上一轮 choking 竞争中失败而不出现在任何节点的 $N^+(\cdot)$中,所以,LX 的 unchoking 策略仍然没有给搭便车节点机会.显然,本文给出的搭便车处理方法属于拓扑调整(topology)这一类[12−14],但本文的工作与这些研究存在很大区别:

- 文献[12,13]是通过建立和调整一个复杂的覆盖网络来实现激励的,激励效果建立在拓扑结构基础上,理论意义远大于实际意义,FOX[13]在文件下载之前需要建立一棵树状覆盖网拓扑;文献[13]要在树状多播结构上进行定期的拓扑调整,使搭便车节点不可能总是处于多播树中只消费而不贡献的位置上;而文献[14]使用了强化学习手段对节点的行为进行学习,从而提高系统的鲁棒性.对于大规模 P2P 系统而言,庞大而复杂的拓扑调整需要花费大量的代价;
- 与文献[12−14]相比,本文给出的 LX 拓扑调整方法简单、灵活,专门适用于对搭便车行为的抑制中,具有更大的适用价值.

另外,也存在一些与本文相关、但角度不同的研究结果,如文献[15,16]等.文献[15]也是针对 BitTorrent 搭便车攻击的研究,但该文主要针对 seed 进行研究,提出的 seed 带宽分配策略从 seed 角度抑制搭便车攻击,其研究结果和本文的角度不同;而文献[16]主要研究如果控制 BitTorrent 中 choking 和 unchoking 之间配额(即,两类分配策略分别控制的节点上传带宽额度)来提高系统性能.而本文集中研究 unchoking 对系统的影响,本文的主要贡献为:

(1) 本文提出了基于连接交换的 BitTorrent unchoking 策略,该方法简单灵活、可用性好,可直接适用于采用各类采用不同 choking 策略(如带宽 TFT、按比例分配、文件块 TFT 等)的 BitTorrent 中,易于推广;

(2) 本文在连接交换的基础上提出了一种能精细控制节点连接的概率式连接交换 PLX,PLX 不仅能处理搭便车攻击,而且对系统中的各类结构都具有更好的激励效果;

(3) 本文用数学方法对 BitTorrent 中的 choking 策略和本文提出的 unchoking 策略共同作用下的系统进行了深入的理论分析,分析其对系统公平性和文件下载时间的影响,为后续研究奠定了理论基础;另外,本文也是专门针对 BitTorrent unchoking 中首个较为深入的研究结果.

本文第 1 节给出 BitTorrent 中节点上传连接的交换 LX,从公平性和共享效率两方面分析 LX.第 2 节提出筛式网络和概率式连接交换 PLX.第 3 节给出 BitTorrent unchoking 策略 PLXU,并详细分析 PLXU 及其重要参数

对文件共享系统的影响.第 4 节对 PLXU 和现有的 unchoking 策略进行实验对比.第 5 节是本文的结论.

1 BitTorrent 中节点上传连接的交换

1.1 连接交换(link exchange)

定义 1. P2P 系统中两个节点 p,q 上的一个连接交换 $\boldsymbol{LX}(p,q)$定义为一个操作:

$$N^+(p)=N^+(p)\cup\{c_1\in N^+(q)\}\wedge N^+(q)=N^+(q)\cup\{c_2\in N^+(p)\} \tag{3}$$

其中,$N^+(p)$是节点 p 上传连接指向的节点集合.

图 1 给出了一个 $\boldsymbol{LX}(p,q)$操作的实例,其中的实线是执行 LX 之前的拓扑,虚线是执行 LX 后新增的连接.

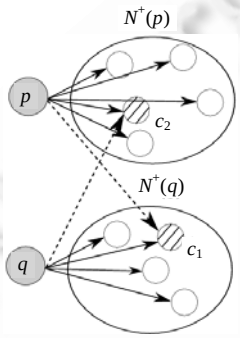

Fig.1 A $\boldsymbol{LX}(p,q)$ instance

图 1 一个 $\boldsymbol{LX}(p,q)$实例

1.2 LX unchoking的安全性分析

对于任意一个搭便车节点 f,f 获得的下载取决于其他节点给 f 的上传,即,那些满足 $f\in N^+(\cdot)$的节点集合,本文将该集合记为 $N^-(f)$.显然,系统的搭便车分析等同于分析搭便车节点 f 的 $N^+(f)$和 $N^-(f)$,系统的公平性分析等同于分析所有节点的 $N^+(\cdot)$和 $N^-(\cdot)$的平均表现.而在 choking 和 unchoking 的作用下,$N^+(\cdot)$和 $N^-(\cdot)$会发生动态变化,此时就变成为两个随时间变化的集合序列,因此,本文最终分析的是系统平稳后诚实节点和不诚实节点的 $\tilde{N}^+(\cdot)$和 $\tilde{N}^-(\cdot)$(其中, $\tilde{N}(\cdot)$是 $N(\cdot)$平稳后的结果).

文献[12]根据文献[19]定义的效用函数将文件共享系统的不诚实节点分为 4 类:

(1) 搭便车节点(free riders,简称 FR)不贡献任何上传带宽,只寻求免费数据的下载;

(2) 一般自私节点(ordinary strategic peers,简称 OSP)通过对带宽分配的控制(但提供正确数据)来最大化其下载速率;

(3) 侵略性自私节点(aggressive strategic peers,简称 ASP)通过给其他节点提供对带宽分配的控制且提供虚假数据来最大化其下载速率;

(4) 保守性自私节点(conservative strategic peers,简称 CSP)是那些最小化欺骗风险的节点,这些节点只有收到下载后才提供上传.

分析 LX 的安全性,将集中分析 LX 对 FR,OSP,ASP,CSP 这 4 类不诚实节点(本文将 OSP,ASP,CSP 统称为 SP)的影响.

定理 1. 在 LX unchoking 下,对于任意 FR f 有:

$$\tilde{N}^-(f)=\varnothing \tag{4}$$

证明:设 t 时刻 $N^-(f)$不为空(可记为 $N_t^-(f)\neq\varnothing$),则必有 $c\in N_t^-(f)$.如果节点 c 是执行 unchoking 后出现在 $N_t^-(f)$中的,则根据图 1,必存在节点 c'满足 $c'\in N_{t-1}^-(f)$;而如果节点 c 是执行 choking 后出现在 $N_t^-(f)$中的,由于 choking 要根据获得的下载来决定是否上传,所以如果 $c\notin N_{t-1}^-(f)$,必有节点 f 在 t–1 时刻给节点 c 提供了上传,

这与 f 是 FR 的假设矛盾.综合两种情况就有 $N_{t-1}^-(f)\neq\varnothing$.同理可推出 $N_{t-2}^-(f)\neq\varnothing,...,N_0^-(f)\neq\varnothing$.由于初始加入的节点没有连接,即 $N_0^-(f)=N_0^+(f)=\varnothing$,得出矛盾.从而对任意 t,满足 $N_t^-(f)=\varnothing$,即有 $\breve{N}^-(f)=\varnothing$. □

定理 1 表明,LX unchoking 机制对 FR 类型节点是完全抑制的.为对该机制抵抗 OSP,ASP,CSP 类型节点的效果进行定量分析,首先要定义一个能评价非诚实节点获利程度及诚实节点受害程度的定量指标:

定义 2. 系统对节点 p 的公平度 $F(p)$定义为节点 p 获得的下载带宽和提供的上传带宽之比:

$$F(p)=\left(\sum_{x\in\breve{N}^-(p)}BW(x)/|\breve{N}^+(x)|\right)/BW(p) \tag{5}$$

其中,$BW(x)$表示节点 x 提供的上传带宽.由于节点 x 同时给 $N^+(x)$个节点提供上传,所以节点 p 收到的是节点 x 上传带宽的$1/|\breve{N}^+(x)|$.

由于 LX 是一种 unchoking 策略,需要与 choking 策略一起工作,因此,也需要将 LX unchoking 和 choking 放在一起来分析.许多研究结果揭示了 BitTorrent 中的 tit-for-tat choking 策略对网络连接的影响:Fan 等人在文献[16]中用 Nash 平衡证明,在相似带宽节点之间建立上传、下载连接是 Nash 平衡点;而 Legout 等人在文献[17]中用详细的实验结果表明,在 BitTorrent 的节点选择策略下,相似带宽节点会出现连接的聚类;Altman 等人得到的研究结果也类似[18].两个相同的结论表明:BitTorrent 中的 tit-for-tat choking 策略会造成相互建立上传连接的节点具有相似的带宽,即:

$$BW(x)\approx BW(y)|_{\forall y\in N^+(x)}\approx BW(z)|_{\forall z\in N^-(x)} \tag{6}$$

定理 2. 如果要求 LX 必须满足条件节点 p 和 q 之间存在路径时才能执行连接交换(本文将这种带路径限制条件的 LX 称为 ***pcLX***,即,path constrained link exchange 的简写,如图 2 所示).在 pcLX unchoking 下,对于一个只包含诚实节点的系统,当该系统达到文献[20]所描述的平衡点时,对系统中的任意节点 p 都有:

$$F(p)=1 \tag{7}$$

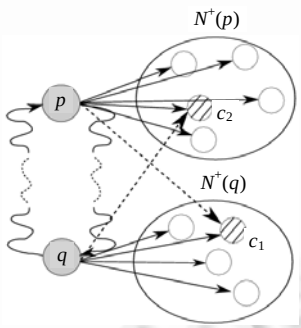

Fig.2 A ***pcLX***(p,q) instance

图 2 一个 ***pcLX***(p,q)实例

证明:$\forall u\in N^+(p)$,系统达到文献[20]所描述的平衡点时,如果节点 u 是被 tit-for-tat 策略选中并保留下来的,则 $BW(u)=BW(p)$(为分析方便,将公式(3)中的≈用=代替);如果节点 u 是被 pcLX unchoking 策略选择的,则由图 1 有 $BW(u)=BW(q)$,其中,$q\in N^-(u)$.因为存在一条从节点 p 到 q 的路径,所以显然有 $BW(p)=...=BW(q)=BW(u)$(这条路径上的连接都是被 choking 选择的).因此,$(\forall u\in N^+(p))BW(u)=BW(p)$蕴含$(\forall d\in N^-(p))BW(d)=BW(p)$,带入公式(3)有:$F(p)=|\breve{N}^-(p)|/|\breve{N}^+(p)|$(此处假定所有节点的$|\breve{N}^+(\cdot)|$相同,该假设符合实际的 P2P 文件共享系统,如 BitTorrent 就有该值都等于 5).

对于节点 p,$N^-(p)$由两部分组成:

(1) 根据 tit-for-tat 交换获得的,是 $N^+(p)$的一个子集(该子集的大小为$|N^+(p)|-1$);

(2) 被其他节点 pcLX unchoking 得到的,节点 p 以外的 $N-1$ 个节点都可能(是一个随机事件)成为这样的节点.

对于任意 p 以外的节点 x,x pcLX unchoking 到节点 p 的事件对应的示性随机变量及其概率分布为

$$U_{xp}=\begin{cases}1, & \Pr(x.pcLX(c),c\in N^-(p))\times\Pr(c.LX(p))\\ 0, & \text{other}\end{cases},$$

其中,$x.pcLX(c)$表示事件 x 选中节点 c 执行 $pcLX$(对应于图 2:$x=p$,$c=q$).如果图 2 中的节点 q 是节点 p 根据随机游走获得的,则通常都有 $\Pr(x.pcLX(c))=1/(N-1)$(关于这一部分,将在第 1.3 节进行详细分析).显然:

$$\Pr(x.pcLX(c),c\in N^-(p))=\sum_{c\in N^-(p)}\Pr(x.pcLX(c))=|N^-(p)|/(N-1).$$

上式中的 $c.LX(p)$表示事件 pcLX 选中的节点 c 选择节点 p 完成最终的连接交换,本文的策略是从 $N^+(c)$中等概率选取一个节点执行连接交换,所以 $\Pr(c.LX(p))=1/m$(为描述方便,令所有节点的$|N^+(\cdot)|$相同且等于 m).所以,

$$E[|N^-(p)|]=|N^+(p)|-1+\sum_x E[U_{xp}]=|N^+(p)|-1+|N^-(p)|/m.$$

两边同时求数学期望,由于 $E[E[|N^-(p)|]]=E[|N^-(p)|]$,所以 $E[|N^-(p)|]=m-1+E[|N^-(p)|]/m$.

求解等式即有 $E[|N^-(p)|]=m=|N^+(p)|$,从而公式(4)成立. □

定理 1 表明:搭便车节点不可能进入一个下载系统(一个通过相互交换文件块来下载同一文件的节点集合),因为提供上传是不被 choking 而进入这一系统的必要条件.定理 2 表明:在这样一个下载系统中,由 tit-for-tat choking 机制和 pcLX unchoking 机制组成的激励策略保证了诚实节点之间的公平性.但有意设计的非诚实节点(SP)可以进入该下载系统,一个显然的结论是:任意 SP 节点 s 为了进入上传带宽等于 B 的节点组成的下载子系统(一个下载系统根据节点提供的上传带宽被 choking 机制分为若干子系统[18]),必须至少给一个节点提供 B/m 上传带宽,而这个上传带宽会 tit-for-tat 回一个 B/m 下载带宽,也即 choking 机制对节点 s 的,所以节点 s 使其获得下载带宽多于上传带宽的唯一途径是 s 不执行 unchoking 操作,而等待其他节点 unchoking 到 s.据此可以得出结论:

定理 3. 在 pcLX unchoking 下,当该系统达到文献[18]所描述的平衡点时,对系统中的任意 SP 节点 s 都有:

$$F(s)\leqslant 1+1/m \tag{8}$$

证明:当节点 s 通过向某个节点 q 提供 b_1 上传带宽而进入到某下载子系统时,此时存在一条从 q 到 s 的连接且 $BW(q)=mb_1$.在分析存在多少个节点能 unchoking 到节点 s 时,类似地定义示性随机变量 U_{xqs},且:

$$\Pr(U_{xqs}=1)=\Pr(x.pcLX(q))\times\Pr(q.LX(s))=1/(N-1)\times 1/m.$$

因此,节点 s 每提供 b_1 上传带宽而能获得 pcLX unchoking 下载带宽的期望值为

$$\sum_x E[U_{xqs}]\times b_1=b_1/m.$$

当节点 s 向其他任意节点提供 b_1 上传带宽后,choking 会使 s 收到 b_1 下载带宽,unchoking 会使 s 收到 b_1/m 下载带宽,节点 s 收到的下载带宽与提供的上传带宽之比为$(1+m)/m$.显然,节点 s 无论怎样提供上传带宽,这个比例都保持不变.因此,无论 SP 节点采用何种带宽分配策略以及是否提供虚假数据,都有公式(5)成立. □

定理 3 给出了 pcLX unchoking 导致任意恶意节点获得非诚实收益的上界为其贡献的 $1/m$:

(1) 显然,随着 m 的增大,恶意节点获得非诚实收益的上界将变小.当 $m\to\infty$时,$F(s)\to 1$.在理论上给出了保证系统中所有节点公平的激励方法(尽管在实际应用维护很大的 m 值较为困难);

(2) 在实际的 BitTorrent 中,choking 和 unchoking 的执行周期是不相等的(BitTorrent 中,choking 的周期是 10s;unchoking 的周期是 30s.令 unchoking 的周期除以 choking 的周期的比值为β),公式(5)给出的结果是在二者周期相同(即$\beta=1$)时的结果.显然,公式(5)中等号右边的真正结果是 $1+1/\beta m$.因此,适当增大β值可以进一步减小恶意节点获得的非诚实收益;

(3) 在公式(5)的证明中,要求 $\Pr(x.\boldsymbol{pcLX}(q))$对于任意的 q 是等概率的,否则,节点 s 会选择 $\Pr(x.\boldsymbol{pcLX}(q))$值高的节点 q 提供上传.此时,$F(s)$可以达到非常大.这就要求在 ***pcLX***(如图 2 所示)时,p 选 q 时需保证 q 是等概率选取的.

1.3 LX unchoking对文件共享效率的影响

Unchoking 机制对 P2P file swarming 系统存在重要的基本用途是:找到更合适的进行文件块相互共享的伙

伴.连接交换(LX)是本文提出的一种 unchoking 机制,因此,LX 在促使 P2P 文件共享系统公平的同时也应该能够找到更合适的文件块交互伙伴.定理 1~定理 3 表明,LX 能够有效抵抗搭便车攻击、促使系统公平、抑制各种恶意节点(确切地说,是 pcLX 完成了上述效果,而 pcLX 是 LX 的一种实现,因此,本文后面内容中的 LX 即指 pcLX).但 LX 能在多大程度上完成其找到更合适文件块共享伙伴这一基本用途,此处需进行详细分析.在分析之前,首先需要明确:对于节点 p,哪些节点是 p 的合适共享伙伴 $FSB(p)$,其中,FSB 是 Fitting Sharing Buddies 的简称.显然,$FSB(p)$是那些具有 p 感兴趣的文件块且其上传带宽等于 $BW(p)$的节点,这是因为,

(1) 对于 $FSB(p)$中任意节点 q,有 $BW(p)\leqslant BW(q)$,否则,节点 p 的 choking 机制会阻断向 q 的上传连接,q 不能成为 p 的共享伙伴.而由于 P2P 系统的对称性,又要求 $BW(q)\leqslant BW(p)$;

(2) 如果 q 没有 p 感兴趣的文件块,就不会建立从 p 到 q 的上传连接,choking 机制使得从 q 到 p 的上传连接也不会建立,q 不能成为 p 的共享伙伴,而此时的对称性也成立.因此有:

$$FSB(p)=\{q|BW(q)=BW(p)\wedge DC(q)\not\subset DC(p)\wedge DC(p)\not\subset DC(q)\} \tag{9}$$

其中,$DC(p)$表示节点 p 下载带的文件块集合.

定理 4. 对于任意节点 p,LX unchoking 机制找到的节点 q 能成为 p 的合适共享伙伴的概率为

$$\Pr(q\in FSB(p))=(1-(l/|N^*(p)|)^l)^2 \tag{10}$$

其中,$N^*(p)$是节点 p 所在的连通分量,l 是节点下载获得的文件块数量.

证明:节点 q 成为 p 的共享伙伴需要满足公式(6)给出的 3 个条件,显然,这 3 个条件是独立的,因此,

$$\Pr(q\in FSB(p))=\Pr(BW(q)=BW(p))\times\Pr(DC(q)\not\subset DC(p))\times\Pr(DC(p)\not\subset DC(q)).$$

LX 根据上传连接找到节点 q 的工作机制,使得 $\Pr(BW(q)=BW(p))=1$.对于概率值 $\Pr(DC(q)\not\subset DC(p))$,由于节点 q 是从 p 的上传邻居(即 $N^+(p)$)或 p 的上传邻居的上传邻居等(用 $N^*(p)$表示该集合)中均匀地随机选取的,所以 $\Pr(DC(q)\not\subset DC(p))$就是从集合 $N^*(p)$中均匀选取、选取到的节点拥有 $DC(p)$以外的文件块的概率.

由于 BitTorrent 采用 rareset first 的文件块选择机制,该机制使每个文件块在相邻节点间的分布数量较为均匀,因此本文假设每个文件块在 $N^*(p)$(由于 unchoking 的周期长于 choking,所以执行 unchoking 时,choking 机制已经使得 $N^*(p)$就是节点 p 所在的连通分量)集合上的分布数量均匀,且分布位置也均匀,即,任意文件块 c_i 都均匀地分布在 $N^*(p)$中的 l 个节点上.对节点 q 中的每个文件块 c,c 也在节点 p 中的概率为 $1/|N^*(p)|$,所以 $DC(q)$中的所有文件块都在节点 p 中(即 $DC(q)\subseteq DC(p)$)的概率为$(1/|N^*(p)|)^{|DC(q)|}$.在 BitTorrent 中,通常每个节点从 seed 那里获得很少的文件块后就开始进行 file swarming,所以可以假定 $N^*(p)$中的每个节点都向 swarming 系统引入 1 个不同的文件块.再根据上面给出的分布均匀假设,有$|DC(p)|\times|N^*(p)|=l\times|N^*(p)|\Rightarrow DC(p)=l$.因此,

$$\Pr(q\in FSB(p))=(1-(l/|N^*(p)|)^l)^2.$$ □

对于 BitTorrent 现有的 optimistic unchoking 机制,对应公式(10)的概率值为$\rho(1-(1/N)^l)^2$,其中,N 是系统中下载同一个文件的节点个数,ρ是这些节点中和 p 共享带宽一样的节点所占的比例.再令$|N^*(p)|=\rho' N$ 表示 LX 机制形成的分量在 N 所占的比率,此时可以定义公式(11)所示的共享增效函数:

$$\Delta(N,l,\rho,\rho')=(1-(l/|\rho' N|)^l)^2-\rho(1-(1/N)^l)^2 \tag{11}$$

来表示 LX unchoking 机制较现有 optimistic unchoking 机制对 BitTorrent 文件下载效率的影响.

图 3 给出了若干参数设置下$\Delta(N,l,\rho,\rho')$的变化曲线,从中可以看出:当 $l<\rho' N=|N^*(p)|$时(图 3 中跃变处的ρ'就是满足条件 $l=\rho' N$ 的ρ'),$\Delta(N,l,\rho,\rho')$非常迅速地增大.这表明:只要 p 所在连通分量提供的文件块数量多于每个节点已经下载的文件块数量时,LX unchoking 下的文件下载效率明显优于 optimistic unchoking 机制.而上述条件实际上是 P2P 文件交换系统工作的必要条件.

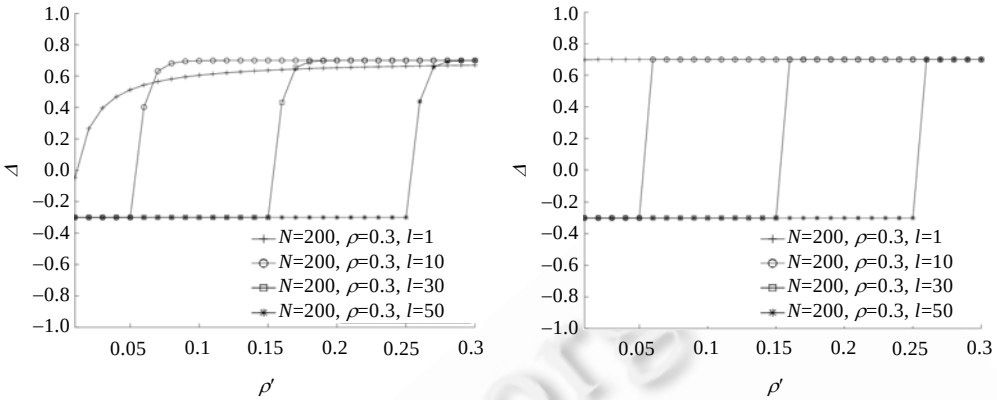

Fig.3 BitTorrent file download efficiency (LX unchoking vs. optimistic unchoking)

图 3 BitTorrent 文件下载效率(LX unchoking 较 optimistic unchoking)

2 筛式网络和概率式连接交换

2.1 $N^*(p)$的扩大

为使新文件块进入文件交换系统(即,满足系统工作的必要条件),要解决的基本问题是如何扩大 $N^*(p)$.

显然,LX unchoking 机制无法扩大集合 $N^*(p)$.因此,目前扩大该集合的唯一方法是一个满足 $BW(q)=BW(p)$ 的节点 q 找到集合 $N^*(p)$中的某个节点 u,并先给该节点上传文件块,从而 choking 机制会建立从 u 到 q 的连接;接下来,$N^*(p)$中的其他节点会 LX unchoking 到节点 q,会建立 q 和 $N^*(p)$中其他节点的连接,q 节点加入集合 $N^*(p)$.

显然,对于目前的由 unchoking 和 choking 组成 BitTorrent 节点选择策略,上述动作只有在节点 q 初次进入系统才会发生,且需要 q 随机找到的节点属于 $N^*(p)$.因此,扩大 $N^*(p)$的方法可以有多种:

(1) 如果新加入的节点对 q 系统没有认识,q 只能随机寻找一个节点执行加入动作.所以此时 $N^*(p)$扩大的概率为ρ',如果单纯依靠节点进入系统来增大 $N^*(p)$的话,ρ'就会很小,ρ'很难扩大.所以此时可以通过修改 tracker 上的信息来增加节点 q 对系统的认识,如在 tracker 上除保存节点标识以外还保存有节点带宽等属性信息,这些信息是节点 q 要进入系统时告诉 tracker 的,tracker 根据 q 的属性选取一些合适的节点(带宽等于 $BW(q)$的节点)ID 返回给 q,然后,q 通过给这些节点提供上传而进入文件交换系统.如果新加入的节点 q 告诉 tracker 的带宽信息是虚假的,则只能给节点 q 带来坏处.

- ✓ 如果 q 提供的带宽信息高于 $BW(q)$,则 tracker 返回的节点集合会很快将 q choking 出系统,q 无法获得数据;
- ✓ 而如果 q 提供的带宽信息低于 $BW(q)$,则 tracker 返回的节点集合都从 q 那里获得比提供给 q 更大的下载带宽,对 q 显然不公平.

从而保证了 tracker 上存储的节点带宽信息是正确的.该方法非常简单、高效,特别适合于工程实现.但该方法需要修改现有 BitTorrent 协议中的 tracker 部分,而且如果节点带宽发生动态变化时,该方法不再适用.因此,该方法的灵活性不好;

(2) 因此,提出一种不修改 tracker 的、不用记录带宽信息的、灵活的 $N^*(p)$扩大方法将是对 LX unchoking 技术的完善,这就是筛式网络和概率式连接交换的基本目的.采用筛式网络(见图 4)以后,网络会像筛子一样将处于错误位置的节点筛向其应该所在的位置.而通过在上面给出的 pcLX 机制上引入精细的概率控制,可以控制筛选的速度,使 $N^*(p)$扩大,使文件交换系统工作的必要条件得以满足,从而达到抑制攻击和提高文件共享效率的双重目的.

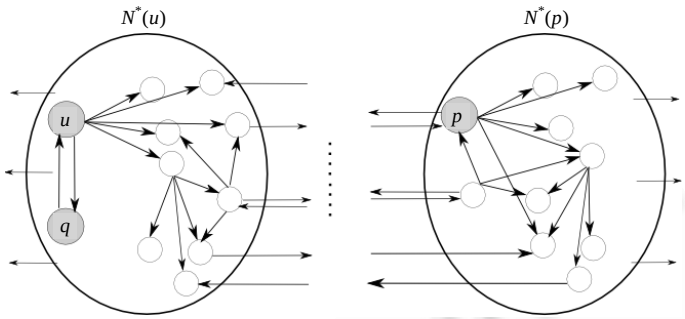

Fig.4 Sieve network

图 4 筛式网络

2.2 筛式网络

当一个新节点 q(本文假设这个节点已经从 seed 那里获得了部分文件块)通过从 tracker 那里获得一些节点信息来加入文件交换系统中时,在 LX uncoking 机制下,节点 q 必须首先给这些节点提供上传带宽,从而接下来的 choking 机制会建立 q 和某节点 u 的连接,再后来的 LX unchoking 会将 q 连入 u 所在的分量(即 $N^*(u)$)中.显然,q 和 u 满足 $BW(u)\leqslant BW(q)$.这是因为如果 $BW(u)>BW(q)$,u 上执行的 choking 机制会阻断和 q 的连接.而出现 $BW(u)<BW(q)$的原因是,节点 q 没有发现带宽高于 $BW(u)$的节点.因此,筛式网络的目的就是将 q 从 $N^*(u)$逐渐筛向 $N^*(p)$(其中,$BW(q)=BW(p)$).

筛式网络是由节点间的上传连接形成的节点关系图,如图 4 所示.在筛式网络中,$N^*(u)$的含义有所变化,它不再是前文所指的 u 上传连接关系的闭包,而是包括 u 在内的某些节点形成的一个簇(较其他节点具有明显紧密的连接).当网络中没有错位节点时(图 4 中的 q 就是一个错位节点),集合 $N^*(u)$就是 u 的上传节点闭包中和 u 上传带宽相等的节点子集.筛式网络中的连接由两部分组成:$N^*(u)$内部(簇内)的连接(简称为内部连接)和各 $N^*(\cdot)$之间(簇间)的连接(简称为外部连接).其中的内部连接就是前文所述的执行 pcLX 时用到的连接,显然,这个通过这些连接 $N^*(u)$中的节点 q 无法发现 $N^*(u)$中的节点,从而 q 无法进入网络中的正确位置.因此,使筛式网络得以工作的关键是那些外部连接,q 正是通过这些连接发现较当前位置更合适的网络位置.

2.3 概率式连接交换(probabilistic link exchange)

定义 3. 对于 P2P 系统中的两个节点 p,q 上的一个概率连接交换 $\boldsymbol{PLX}(p,q,\zeta)$定义为

$$\boldsymbol{PLX}(p,q,\zeta)=\begin{cases}\boldsymbol{pcLX}(p,q), & 1-\zeta\\ \boldsymbol{pcBLX}(p,q), & \zeta\end{cases} \tag{12}$$

其中,**pcBLX** 和 **pcLX** 类似,只是在选择公式(1)的 c_1,c_2 时从 $N^-(p),N^-(q)$中选取;ζ是一个系统可调整的参数,用来表示执行相应类型的 **LX** 的概率.

对于筛式网络(如图 4 所示)中相邻的两个簇 $N^*(u)$和 $N^*(v)$,并且设 $BW(u)<BW(v)$.经过一轮 choking 以后,由于 $BW(u)<BW(v)$且 $N^*(v)$内部存在大量上传连接,所以 $N^*(u)$到 $N^*(v)$的连接不会被阻断,而从 $N^*(v)$到 $N^*(u)$的连接都会被阻断.再执行一轮 choking 以后,$N^*(u)$到 $N^*(v)$的连接也都会被阻断,这是因为 $N^*(u)$中的节点已经不能从 $N^*(v)$那里获得下载了,tit-for-tat 阻断了向 $N^*(v)$的上传.

从这个过程可以看出,choking 会阻断所有筛式网络的外部连接.因此,这些外部连接只能由 unchoking 实现,这正是 PLX 要完成的工作.另一方面,如果外部连接数量太少,会使节点在筛式网络中移动较慢,会导致 l 的增长速度大于 $N^*(p)$的增长速度,影响下载效率;而如果数量太大,导致 LX unchoking 到内部节点的概率降低,降低系统公平性.而 PLX 中的概率就用来实现对外部连接数量控制(在后面的章节,将详细分析这个概率ζ对外部连接数量的影响,进而分析其对系统性能的影响).

3 PLXU:基于概率式连接交换的 BitTorrent unchoking 协议

3.1 PLXU协议

在上述分析基础上,本文提出了基于概率式连接交换的 BitTorrent unchoking 协议,简称 PLXU 协议,该协议的具体细节见算法 1.

算法 1. PLXU unchoking 协议.

1: 令 p_i 为系统中某节点,$N^+(p_i)$($RN^-(p_i)$)为 p_i 的上传连接(最近下载连接)邻居集合,常数$\zeta\in[0,1]$;
2: **for each** $p_i\in P$;
3: **if** p_i 进入执行 unchoking 的周期,且没有收到 $M.TTL$=0,$M.type$=LXQ($BLXQ$)的 M;
4: p_i 产生一个[0,1]的随机数 r;
5: **if** $r\leqslant\zeta$ **then**
6: p_i 从集合$\{p|p\in RN^-(p_i)\wedge BW(p_i)-T\leqslant BW(p)\leqslant BW(p_i)\}$中等概率选取一个节点 p';
7: p_i 产生数据包 M=p_i||TTL||$BLXQ$||p',p_i 随机选取 $p_j\in N^+(p_i)$,将 M 发送给 p_j 并记录路径;
/*数据包格式为:源/目标||TTL||类型||数据内容(s/d||TTL||$type$||$content$)*/
8: **if** $r>\zeta$ **then**
9: **if** $(\exists p\in N^+(p_i))BW(p)>BW(p_i)$ **then** $p'=p$;
10: **else** p_i 从集合 $N^+(p_i)$中等概率选取一个节点作为 p';
11: p_i 产生数据包 M=p_i||TTL||LXQ||p',p_i 随机选取 $p_j\in N^+(p_i)$,将 M 发送给 p_j 并记录路径;
12: **if** p_i 收到数据包 M 且 $M.type$=LXQ($BLXQ$),$M.TTL$>0 **then**
13: p_i 修改数据包 $M.TTL$−1,随机选取 $p_j\in N^+(p_i)$,将 M 发送给 p_j 并记录路径;
14: **if** p_i 收到数据包 M 且 $M.type$=LXQ,$M.TTL$=0 **then**
15: p_i 取出 p_j=$M.s/d$,p_u=$M.content$,unchoking p_u;
16: **if** $(\exists p\in N^+(p_i))BW(p)>BW(p_i)$ **then** $p'=p$;
17: **else** p_i 从集合 $N^+(p_i)$中等概率选取一个节点作为 p';
18: p_i 产生数据包 M=p_j||−||LXR||p',沿着反向路径发送 M;
19: **if** p_i 收到数据包 M 且 $M.type$=$BLXQ$,$M.TTL$=0 **then**
20: p_i 取出 p_j=$M.s/d$,p_u=$M.content$;
21: **if** p_u 在 $N^+(p_i)$中持续停留不大于 2 个周期 **then** unchoking p_u;
22: p_i 从集合$\{p|p\in RN^-(p_i)\wedge BW(p)\geqslant BW(p_i)-T\}$中等概率选取一个节点 p';
23: p_i 产生数据包 M=p_j||−||LXR||p',沿着反向路径发送 M;
24: **if** p_i 收到数据包 M 且 $M.type$=LXR **then**
25: p_i 取出 p_u=$M.content$;
26: **if** $BW(p_u)\geqslant BW(p_i)$或 p_u 在 $N^+(p_i)$中持续停留不大于 2 个周期 **then** unchoking p_u;

(1) 语句 3 中也可以不检查是否收到满足条件的数据包 M,此时并不会导致错误,只会导致更多的节点和其他节点执行多次 LX.而由于只 unchoking 一个节点,所以只有最后一次 LX 才是有效的(当然,也可以设计为第一次 LX 有效,且更容易实现).而加上检查语句以后,可以减少 LXQ(BLXQ)类型的数据包通信数量,降低通信代价.

(2) 语句 7、语句 11 等用来将 LXQ(BLXQ)数据包随机游走到一个执行 LX 操作的目标节点处(即,完成 pcLX),采用随机游走的目的是增加找到节点的随机性,避免攻击(见定理 3 中的情形(3)).LXQ(BLXQ)数据包中的 TTL 域用来控制随机游走的距离,显然,距离越长随机性越好,但造成的通信负载也要大一些.但由于不引起广播,所以一般的 TTL 值不会造成大量负载.另一方面,分布式的随机游走单个用户很难控制,所以即使较小的 TTL 也不会造成定理 3 中的情形(3)的攻击,因此,本文将 TTL 设为 3~6 之间的一个随机数.

(3) 语句 7、语句 11 等中的数据包内容 p'是已经选择好的交换节点,算法 1 采用的是先选择交换内容然后

再寻找和谁交换.这样的处理有两个优点:既可以减小 LXQ(BLXQ)数据包的大小,又需要泄露的邻居节点的带宽信息,因为带宽信息需要作为 LXQ(BLXQ)数据包的内容一起发送,否则,在目标节点处就无法确定该选择哪个节点执行 LX.显然,让除邻居节点以外的节点(邻居节点通过数下载速度可算得带宽)知道自身的带宽信息是不安全的.

(4) 语句 6、语句 9 等中用 $BW(p)$和 $BW(p_i)$之间的大小关系来作为某些断言(如在图 4 中 $N^*(p)$和 $N^*(p_i)$不同且在左边)的条件,当假定一个 $N^*(p)$中各节点的带宽完全相同时,这样的条件是正确的;但对于实际系统,带宽完全相同是不可能的,因此可在语句 6、语句 9 中的比较关系上加上一个阈值,如 $N^*(p)<N^*(p_i)-T$,而这个阈值 T 可通过网络测量结果来确定.

(5) 语句 21 和语句 26 用来限制一个低带宽节点在一个高带宽节点上传连接中持续停留的时间,实际上就是限制图 4 中某个特定的向左外部连接的持续时间.控制该时间的目的是防止一个低带宽节点在 LX unchoking 时向高带宽节点推荐的节点是自身(应该是从其邻居节点的闭包中随机选取一个节点),否则,这些自私节点(使自己总能与高带宽节点建立连接)会成为其他节点向右(假定右边是更高带宽的节点集合)移动的屏障,使图 4 中的节点 q 无法移动到合适的位置上.另一方面,当持续停留时间受到限制时,低带宽节点开始愿意将位于同一 $N^*(\cdot)$的其他节点推荐给高节点,因为被推荐的节点将来也会推荐该节点,使该节点有可能在将来与高节点建立连接.而如果不做推荐的话,向左的连接会减少甚至断开,低带宽节点再也不可能和高带宽节点建立文件交换关系.在节点是理性的这一假设下,语句 21 和语句 26 很好地提供了低带宽节点按协议工作的激励机制.

(6) 在建立向左的外部连接时,PLXU 采用的是 $RN^-(p_i)$而不是前文所述的 $N^-(p_i)$(见语句 6 和语句 22). $RN^-(p_i)$表示最近(recent)给节点 p_i 提供过上传带宽的节点集合,引入 $RN^-(p_i)$的目的是保证向左外部连接的数量.如果在语句 6 和语句 22 中采用 $N^-(p_i)$,由于当前时刻没有带宽低于 p_i 的节点向 p_i 提供上传(choking 机制的直接效果就是断开这样的连接),所以发起 BLXQ 的节点就无法建立向左的外部连接.显然,此处讨论的 $N^-(p_i)$是指从 p_i 那里获得下载的节点集和 $N^*(\cdot)$(和 $N^*(p_i)$左侧相邻)相交的那一部分子集,因此,算法 1 中给出的 $N^-(p_i)$和 $RN^-(p_i)$都指这一子集.在算法实现时,可用操作将 unchoking 时的 $N^-(p_i)-N^+(p_i)$加入到 $RN^-(p_i)$中来修改 $RN^-(p_i)$近似得到该子集.当引入 LX unchoking 机制时,结果(5)的分析表明:低带宽节点愿意将其高带宽邻居推荐给其他节点(语句 9 和语句 16),这就使每个 p_i 的 $RN^-(p_i)$很快扩大.另外,显然有每个执行 BLXU 的节点 u 找到一个低带宽节点的概率大于 0(随着各 $RN^-(p_i)$的扩大,这个概率也会变大),使 u 出现在某个低带宽节点 v 的 $N^+(v)$中,就使 v 出现在 $RN^-(u)$中,语句 9 和语句 16 使 $RN^-(u)$很快扩大,增加了 $N^*(v)$各节点被 BLXU 选中的机会.如果按照大小来控制 $RN^-(u)$(而不是按照时间),即:当$|RN^-(u)|$大于某个阈值时才用新加入的节点替换最早加入的节点,否则直接加入新节点的话,上述分析表明,$N^*(u)$中所有节点的$|RN^-(\cdot)|$会很快达到这个阈值并且停在这个阈值上(因为 $RN^-(u)$的修改条件表明在$|RN^-(u)|$小于阈值时只增不减,而又有节点 u BLXQ 到一个低带宽节点的概率大于 0 的事实,使 $RN^-(u)$一定增加).当所有节点的$|RN^-(u)|$都大于等于 1 时,每个发起 BLXU 的节点都能成功建立一个向左的外部连接,这样就能保证 $N^*(u)$集合向左的外部连接数量为$\zeta\times|N^*(u)|$.

3.2 PLXU协议的安全性分析

由于 PLX 没有改变建立连接是 unchoking 的前提这一基本条件,所以定理 1 仍成立:在 PLXU 下,搭便车节点不能从系统里获得任何下载带宽,所以 PLXU 协议可以完全抵抗搭便车攻击.因此,PLXU 的安全性分析主要集中在诚实节点 p 的公平度 $F(p)$和采取一定策略的恶意节点(即 SP)s 获得的公平度 $F(s)$分析上.在 PLXU 下,高带宽节点会给低带宽节点提供上传带宽(如图 4 所示),所以对于诚实节点和策略式恶意节点而言,其公平性较定理 2 和定理 3 给出的结果要差一些.

由于 PLXU 中有相应的机制来保证文件共享网络成为严格筛式网络,所以在此处的理论分析结果是在完全严格的筛式网络上进行的(采用这样的假设是为了分析的方便).对于筛式网络中处于正确位置的诚实节点 p,由算法 1 的分析结果(6)可知:$N^*(p)$给其左相邻 $N^*(p_L)$提供的上传连接数量为$\zeta\times|N^*(p)|$.而 $N^*(p_L)$提供给 $N^*(p)$的上传连接数量,可分析得出其下界为$\zeta\times|N^*(p)|$,上界为$(1+1/2)\times\zeta\times|N^*(p)|$.在这个上下界中,$\zeta\times|N^*(p)|$是 $N^*(p_L)$ choking 给 $N^*(p)$的连接数,而 $1/2\times\zeta\times|N^*(p)|$是 $N^*(p_L)$中的节点按算法 1 中的语句 9、语句 16unchoking 到 $N^*(p)$

的连接数.由语句 9、语句 16 可知:只有当 p 在当前节点 p_L 的 $N^+(\cdot)$中,另一个低带宽节点才能通过 PLXU 来 unchoking 到 p.而由 choking 原理:p_L 给 p 上传是因为上个 choking 周期 p 给 p_L 上传(unchoking),而 choking 又使当前周期 p 不再给 p_L 提供上传(除非 p 又 unchoking 到 p_L),所以下个周期中 p_L 不再给 p 提供上传.由此可以看出,$N^*(p)$和 $N^*(p_L)$之间的传输关系为:每次执行 unchoking 时,p 向 p_L 提供ζ个(此处分析的是期望值)上传连接,p_L 会 choking 回ζ个上传连接.而在 p_L 给 p 上传的期间(一个 choking 周期),如果和 p_L 执行 PLXU 的其他低带宽节点 p'_L 刚好进入 unchoking 周期,则会建立ζ个由 p'_L 指向 p 的上传连接.由于 BitTorrent 中的 unchoking 周期是 choking 周期的 3 倍,所以发生该事件的概率为 1/3.因此,PLXU 会再引入指向 p 的$\zeta/3$ 个连接.同样的道理, p'_L 会在引入其他低带宽节点指向 p 的$\zeta/3^2$ 个连接,...,共引入$(1/3+1/3^2+...)\times\zeta=\zeta/2$ 个指向 p 的上传连接,这显然是一个上界(假定执行 unchoking 的节点总不重复).对 $p\in N^*(p)$累加该界就能得出 $N^*(p_L)$给 $N^*(p)$提供的上传连接数量的上界.

定理 5. 对于由诚实节点组成的、执行 PLXU 协议的系统中的任意节点 p:

$$F(p)\geqslant 1-(1+3c_3/2-c_1-c_2c_3)\zeta/m \tag{13}$$

其中,$c_1=BW(p_L)/BW(p)$,$c_2=BW(p_R)/BW(p)$,$c_3=|N^*(p_R)|/|N^*(p)|$.

当 $c_3=1$,$c_2=1/c_1=c$ 时:

$$F(p)\geqslant 1-(5/2-c-1/c)\zeta/m \tag{14}$$

证明:计算 $F(p)$需要求出 $N^*(p)$贡献的总带宽和收到的总带宽,在求 $F(p)$的下界时,考虑的是 $N^*(p)$贡献最多而收到最少的情况.根据上面的分析结果,此时 $N^*(p)$从 $N^*(p_L)$,$N^*(p_R)$,$N^*(p)$那里获得 3 部分带宽分别为:

$$\zeta|N^*(p)|BW(p_L)/m,\zeta|N^*(p_R)|BW(p_R)/m,(m|N^*(p)|-\zeta|N^*(p)|-3\zeta|N^*(p_R)|/2)BW(p)/m.$$

因此:

$$F(p)\geqslant(((\zeta|N^*(p)|BW(p_L)/m+\zeta|N^*(p_R)|BW(p_R)/m+(m|N^*(p)|-\zeta|N^*(p)|-3\zeta|N^*(p_R)|/2)BW(p)/m))/|N^*(p)|)/BW(p).$$

化简后即有公式(13). □

定理 6. 在执行 PLXU 协议的系统中,任意 SP 恶意节点 s 满足:

$$F(s)\leqslant 1+c\zeta'/(2m)-\zeta'^2/(2m^2) \tag{15}$$

其中,$c=1+BW(s_R)/BW(s)$,$\zeta'=1-\zeta$.

证明:对于 SP 节点 s 而言,s 可以通过采取刻意设计的节点选择策略和带宽选择策略来使其获得收益的最大化.定理 3 给出了一个 SP 节点如何分配带宽在 pcLX 中获得最大收益:向一个带宽为 b 的节点聚类提供带宽 b/m,从而该节点聚类建立一个 choking 连接,然后等待其他节点 unchoking 到自身而自身不执行 unchoking.因此,在筛式网络(筛式网络和定理 3 的聚类网络的唯一区别是在筛式网络中不同聚类之间存在连接,即,存在外部连接)中,s 需用同样的方法进入一个聚类,然后开始等待被其他节点 unchoking 到.在严格筛式网络假设下,unchoking 到 s 的节点可能是同一聚类中的节点,也可能是右相邻聚类中的节点.由定理 3 证明过程可知:在 PLXU 下,同类节点 unchoking 到 s 的概率为$(1-\zeta)/m$(PLXU 以 $1-\zeta$的概率执行定理 3 的 pcLX,而以ζ的概率执行 pcBLX).

现在需求节点 s 被右相邻聚类 unchoking 到的次数随机变量的期望值.被右相邻聚类 unchoking 到首先需要 s 出现在右相邻聚类中某节点的 $RN^-(\cdot)$中,由于 s 和 $N^*(s)$中的某个节点 s'存在 choking 连接,因此,s'会给 s 推荐一个右相邻聚类中节点 s_R,当 s 向 s_R 提供上传带宽以后,s 进入 $RN^-(s_R)$队列的第 1 个位置.节点 s 不可能一直停留在 $RN^-(s_R)$中,因为即使节点 s 不向其他同类节点推荐 s_R(见语句 9 和语句 16),s_R 也能依靠 BLX 找到其他低带宽节点(见语句 21 和语句 26).而当 $RN^-(s_R)$中有 s 以外的节点时(可以假定该节点诚实),由算法 1 分析结果(5)可知,该节点会很快推荐其他同类节点给 s_R 上传并加入到 $RN^-(s_R)$中.因此,s 停留在队列 $RN^-(s_R)$中第 i 个位置的最长时间(实际上是最多停留的 unchoking 周期数)是伯努利实验 s_R 靠 BLXU 找到其他低带宽节点上的几何分布,s 在 $RN^-(s'_R)$ 中位置 i 最多停留的 unchoking 周期数的期望值为 $1/\zeta$.所以,s 最多停留在 $RN^-(s_R)$中 unchoking 周期数是 $|RN^-(s'_R)|/\zeta$.而停留在 $RN^-(s_R)$中的 s 在一个 unchoking 周期被 $N^*(s_R)$中节点 PLXU 到的次数的期望值为 $\zeta/|RN^-(s'_R)|$,所以 s 在 $RN^-(s_R)$中的停留期间被 $N^*(s_R)$中的节点 PLXU 到的次数的期望值为 1.这表示平均

情况下,s 可以用自身的 1 个上传带宽 $BW(s)$换来 1 个上传带宽 $BW(s_R)$(这是在 s 最长停留的前提下的结论,所以实际运行时 s 所得收益要小于该值,可从模拟实验结果看出).而 1 个 $BW(s)$换 1 个 $BW(s_R)$的收益要求其他同类节点在执行 unchoking 时给 s 推荐的一个高带宽节点,这样,s 才能出现在某个高带宽节点的 $RN^-(\cdot)$中,其概率至多为$(1-\zeta)/m$(需要执行 pcLX,且 LX 后选取的节点是 s,当然还要求发起 pcLX 的节点在 $N^+(\cdot)$中有一个高带宽节点).因此对于 s,1 个 $BW(s)$可以从同簇中换回 $1+(1-\zeta)/m$ 个 $BW(s)$,以$(1-\zeta)/m$ 概率再用 1 个 $BW(s)$从右相邻簇中换回 1 个 $BW(s_R)$(文中的下标 $L(R)$分别表示在筛式网络中左(右)相邻).

所以有:

$$F(s)\leqslant(1-(1-\zeta)/m)(1+(1-\zeta)/m)+(1-\zeta)/m\times((1+(1-\zeta)/m)BW(s)+BW(s_R))/(2BW(s)).$$

化简后即得公式(15). □

对于定理 5 和定理 6 给出的结果,可以看出 PLXU 协议对系统安全性的影响.

(1) 公式(10)和公式(12)表明:在 PLXU 下,诚实节点和恶意节点的贡献与获得之比接近于 1,和 1 相差 $O(1/m)$.当 m 达到无穷大时,系统可以达到理论上的完全公平.而且和针对定理 3 的分析一样,公式(10)和公式(12)都是 choking,unchoking 周期相同时的结果,当二者周期不同时,两个式子和 1 相差的是 $O(1/(m\beta))$.适当增大β值,可以进一步增加系统的公平性;

(2) 对于公式(11),当 $c>2$ 时(即,相邻带宽之间超过 2 倍时),除带宽最高的节点集合以外,诚实节点在 PLXU 中获得的收益大于其贡献;而且随着ζ越大,收益贡献比越大.这是由于 $BW(p_R)$较 $BW(p)$较大时,$N^*(p_R)$给 $N^*(p)$注入的带宽足够填补从 $N^*(p)$中流失的带宽,而且还有多余.从系统总体来看,是带宽最高的那些节点贡献其部分带宽给了系统中其他节点.当带宽最高的节点愿意做这样的贡献时(如这些节点是用来发布该文件的服务器群),此时增大ζ,将使系统中的所有用户满意.而当 $c\leqslant 2$ 时,减小ζ无疑会增大系统的公平性,充分体现了ζ的调节作用;

(3) 对于公式(12)的右边,对ζ求偏导,发现该偏导总大于 0.即,该上界是ζ的一个增函数,也即是ζ一个减函数,得出一个有趣的结论:ζ越大,PLXU 对 SP 节点的抑制效果越好.这是由于 SP 节点需要靠 LXU(而不是 BLXU)获得收益(SP 节点靠 LXU 获得同类节点的 unchoking,SP 节点靠 LXU 将其加入高带宽节点的 $RN^-(\cdot)$中),因此从抑制恶意节点的角度讲,ζ应该大一些,再次体现ζ的调节作用;

(4) 定理 6 表明:PLXU 对控制带宽分配的 SP 节点具有较好的抑制作用,但如果多个 SP 节点共谋或一个 SP 节点假扮多个节点进入系统[6]时,这些节点可以互相推荐来欺骗 PLXU,增加进入高带宽节点 $RN^-(\cdot)$中的机会,此时,算法 1 中的语句 21 和语句 26 不再起作用.显然,BitTorrent 中的 optimistic unchoking 通过随机来抑制此类攻击,虽然 PLXU 也用随机游走引入随机性,但只对诚实节点起作用,SP 节点可以完全不发送 BLXQ 数据包,而直接相互推荐,当然可以引入更为复杂的协议来强制随机游走的执行,但实际上,可在语句 21 和语句 26 中更为精细地记录 $N^+(\cdot)$信息(同样可引入 $RN^+(\cdot)$,由于是针对共谋攻击的版本,所以没有写到算法 1 中),并根据 $RN^+(\cdot)$修改 $RN^-(\cdot)$.如:从 $RN^-(\cdot)$中删除那些在 $RN^+(\cdot)$中停留时间超过一定阈值的节点来抑制 SP 节点在 $RN^-(\cdot)$中的停留时间,增加随机性,同样可获得良好效果(见模拟实验结果).

3.3 PLXU协议的性能分析

根据公式(8)的分析结果,PLXU 协议的性能分析等同于分析一个错误位置节点(由于本文只考虑右移,所以错位节点指其带宽高于当前位置处的聚类带宽,如图 4 中的 q)在筛式网络中移动到右相邻聚类需要花费的平均时间.

定理 7. PLXU 协议下,错位节点 q 向右移动到 $N^*(q_R)$的平均时间为

$$\max\{1,c/\zeta\}\text{个 unchoking 周期} \tag{16}$$

其中,$c=c_4c_5$,$c_4=|N^*(q)|/|N^*(q_R)|$,c_5 是一个在[2/3,1]之间的常数.

证明:显然,节点 q 在原聚类中(向右移动之前)停留的 unchoking 周期个数是伯努利实验“节点 q 给 $N^*(q_R)$中的某个节点提供上传连接”上的几何分布,因为节点 q 给 $N^*(q_R)$中的某个节点提供上传以后,q 就会被该节点

的 choking 机制保留下来,而后续的 LXU 会让 q 找到 $N^*(q_R)$中的其他节点,同时,q 处执行的 choking 会因上传带宽小而断开和原聚类节点的连接,所以节点 q 会很快从原聚类向右移动到下一个聚类.因此,q 在原聚类中停留的 unchoking 周期个数的期望值就是伯努利实验成功概率的倒数.由于 q 移动前所在聚类给其右相邻 $N^*(q_R)$提供的上传连接数量在$\zeta|N^*(q_R)|$和 $3\zeta|N^*(q_R)|/2$ 之间(见定理 5),所以节点 q 出现在这些上传连接(是连接的尾部)中的概率就在 $\min\{\zeta|N^*(q_R)|/|N^*(q)|,1\}$和 $\min\{3\zeta|N^*(q_R)|/(2|N^*(q)|),1\}$之间. □

4 模拟实验

4.1 模拟实验环境

本文采用 NetLogo 模拟工具对 PLXU 进行实验验证.NetLogo 是一个基于多 Agent 的 AI 群落模拟工具,某些 P2P 研究的模拟实验就在 NetLogo 上完成[19].这说明 NetLogo 可以用来模拟 P2P 系统,而且 NetLogo 只需扩展 Agent 对象即可模拟 peer 的各种行为,实现容易;Agent 对象间的通信轻便,特别合适于模拟大规模系统在应用层上的性能表现(当然,NetLogo 并不适合模拟传输层以下的性能表现,但 unchoking 机制对 BitTorrent 文件交换性能的影响是一种典型的应用层表现).用 NetLogo 中的一个 Agent 来模拟 BitTorrent 中的一个 leecher,并给每个进入系统的节点初始化分配(随机分配)若干文件块.在模拟系统的运行时,将时间离散的分割为多个模拟周期,在每个模拟周期中完成:

(1) 每个节点执行一遍 choking 算法,一个模拟周期就是一个 choking 周期.由于本文研究的是 unchoking 机制,所以此处直接使用 BitTorrent 的 tit-for-tat 算法来完成 choking;

(2) 对于系统中的每个节点,每间隔一定数量的模拟周期后执行 unchoking 算法(即,开始 unchoking 周期.本文和 BitTorrent 一样,将 unchoking 周期设为 choking 周期的倍数),模拟实验将比较 BitTorrent 原有的 optimistic unchoking 和本文的 PLXU 算法;

(3) 维护相关数据结构、执行数据包转发等.

本文将从公平性和下载效率两个方面对比 unchoking 机制.主要的性能指标是:

(1) 节点 x 的公平性指标 $F(x)$,定义为公式(3);

(2) 节点 x 的文件下载时间 $T(x)$,为开始下载到下载完成的时间.另外,模拟实验中用到的符号见表 1.

Table 1 Symbles and corresponding meanings used in simulations

表 1 模拟实验中用到的符号及参数

符号	描述	缺省值
N	系统中的节点个数	1 000
m	节点的上传连接个数,即$\|N^+(\cdot)\|$	5
$\|RN^-(\cdot)\|$	节点维护的集合 $RN^-(\cdot)$的大小	30
β	unchoking 周期和 choking 周期之比	3
c	相邻带宽之间的带宽之比	2
TTL	LXQ(BLXQ)数据包随机游走的深度	3
ζ	控制节点在筛式网络中的移动速度	0.2
α	每个 leecher 进入系统时初始化的文件块数量比例	0.02
T	平均文件下载时间	–
$\sigma(F)$	系统公平度,$\sigma(F)=\left[\sum_x (F(x)-1)^2/N\right]^{1/2}$	–

4.2 诚实环境下的性能对比分析

图 5~图 7 对比了诚实环境下 PLXU 和 optimistic unchoking(图中简称为 OU)对公平性和文件下载时间的影响.图 5 对比了α=0.02 时,执行 OU 策略的 BitTorrent 系统和执行 PLXU 策略的 BitTorrent 系统,各节点处的下载时间分布和节点公平度分布.图 6 给出的是α=0.10 时的两种分布,从图中结果可以看出:诚实环境下,PLXU 和 OU 对系统的影响差别不大,只是 PLXU 的公平性就 OU 要好一些(是公式(11)的结果);而在下载时间,两类策略基本没有差别.

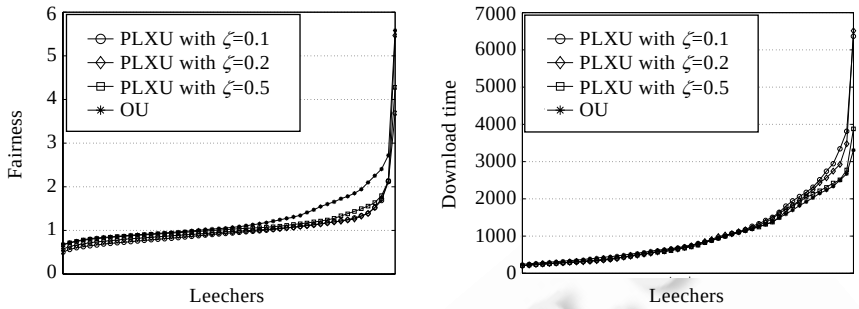

Fig.5 Distribution of nodal fairness and download time under honest environments (α=0.02)

图 5 诚实环境下的节点公平度分布和节点下载时间分布(α=0.02)

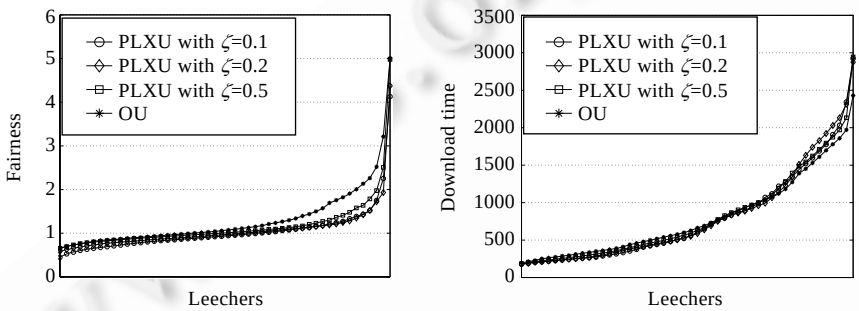

Fig.6 Distribution of nodal fairness and download time under honest environments (α=0.10)

图 6 诚实环境下的节点公平度分布和节点下载时间分布(α=0.10)

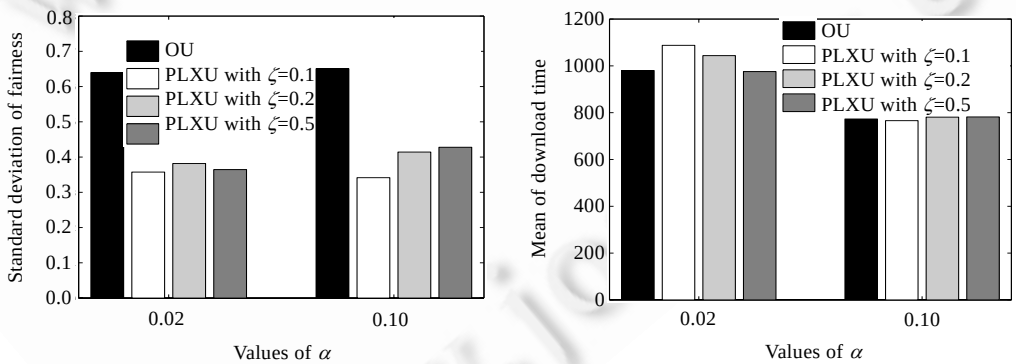

Fig.7 Influence of OU and PLXU on the performance of BitTorrent under honest environments

图 7 策略 OU 和策略 PLXU 对诚实环境下 BitTorrent 系统性能的影响

4.3 搭便车攻击下的性能对比

图 8 和图 9 对比了搭便车攻击下(系统中有 20%的搭便车节点)分别采用 OU 和 PLXU 时,BitTorrent 的节点行为和整体性能.图 8(a)的结果表明:几乎所有的搭便车节点在 OU 下不提供上传却可以获得较大的下载带宽;而在 PLXU 策略下,搭便车节点获得的下载带宽几乎为 0,和理论结果吻合.但在前文的理论分析中,PLXU 下的搭便车节点因不能进入系统而一定获得 0 下载带宽,而图 8、图 9 的结果是获得小带宽,这是由于本实验对要进入系统的节点设置一些由其他节点指向该节点的初始化上传连接,因此搭便车节点也会获得一些节点提供的初始化上传带宽,但只能维持到 choking 之前.这个结果同时表明:即使搭便车节点通过其他手段得到一些上传

连接,也会很快被 PLXU 阻止.

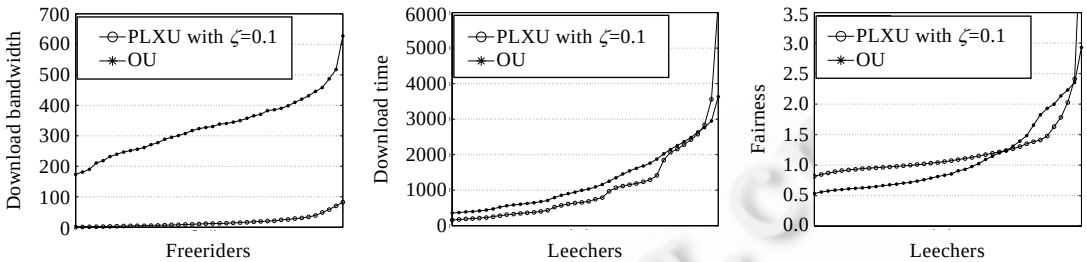

Fig.8 Distribution of nodes' behaviors with free riders

图 8 搭便车攻击下的节点行为分布

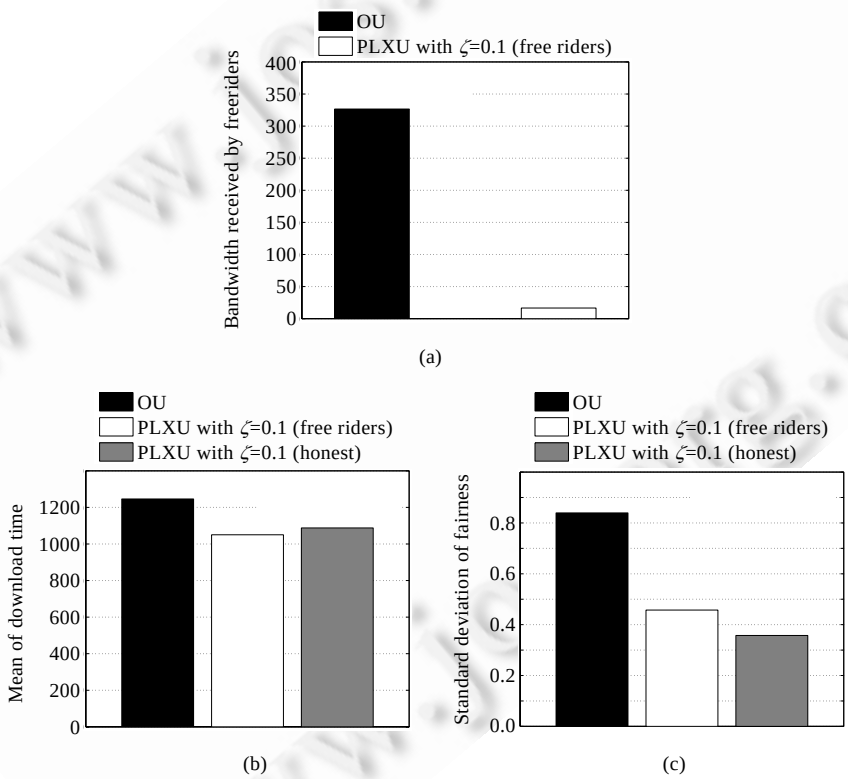

Fig.9 Influence of OU and PLXU on the performance of BitTorrent under free-rider attacks

图 9 搭便车攻击下策略 OU 和策略 PLXU 对 BitTorrent 性能的影响

另外,由于没有搭便车节点对系统上传带宽的消耗,诚实节点的平均下载时间势必整体降低,公平度也变好(如图 8(b)、图 8(c)和图 9(b)、图 9(c)所示,其中的 free riders 表示系统存在搭便车攻击,honest 表示诚实环境);从图中结果也可以看出,搭便车攻击对 PLXU 机会没有任何影响(free riders 设置下和 honest 设置下得到的结果接近),这和前面的理论分析结果一致.

4.4 恶意策略攻击下的性能对比分析

图 10 对比了在恶意策略攻击下(即,系统中存在 SP 节点,且本实验设定系统中有 20%的 SP 节点),分别采用策略 OU 和 PLXU 时 BitTorrent 系统的节点行为和整体性能.图 10(a)对比了两类 unchoking 策略对各 SP 节点及诚实节点的下载时间(公平性)的影响,而图 10(b)给出了系统性能的统计结果.结果表明,PLXU 较 OU 而言具有更好地限制恶意攻击的能力:虽然某些攻击策略对 OU 显得非常有效,但不论 SP 节点选择什么样的攻击策略,PLXU 策略对 SP 节点能获得的利益总能好的抑制,对诚实节点的影响总能限制在一个较小的范围内,这和理论分析结果(公式(11)、公式(12))一致.

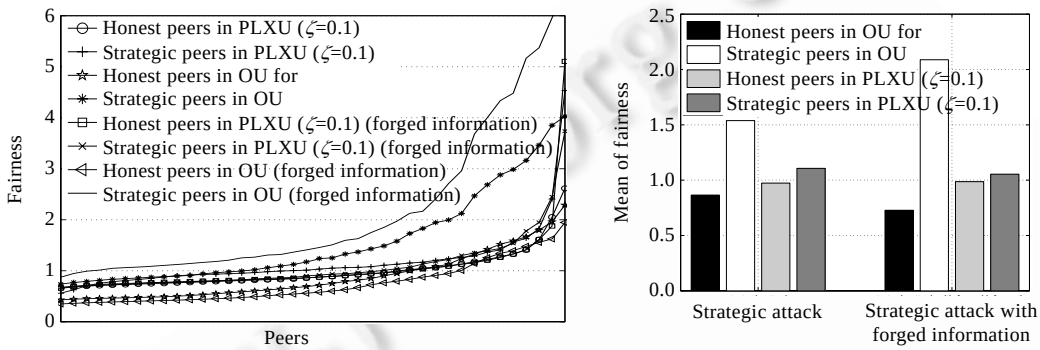

Fig.10 Influence of OU and PLXU on the performance and nodes' behaviors under strategic attacks

图 10 策略 OU 和策略 PLXU 对恶意策略攻击下的各类节点行为表现的影响及性能对比

5 结束语

由于 BitTorrent 系统中的节点行为完全自主,导致出现只下载而不提供上传的搭便车攻击行为,严重降低系统的服务能力.目前采用的方法是 choking+unchoking 机制,其中的 choking 机制用来保证节点提供上传和获得下载间的相等关系(tit-for-tat).choking 可以抑制上述攻击,但 choking 只能阻塞的本质会使节点和系统断开,所以需要在 choking 的同时 unchoking 一些节点.目前,BitTorrent 采用的是随机选择一些节点 unchoking 的策略,但这样的策略会遭受诸如 large view exploit 的攻击,攻击者在不上传的同时却很容易获得下载带宽,甚至比诚实节点获得的下载更多.本文认为,造成 BitTorrent 搭便车攻击的根本原因是目前的 unchoking 策略选择节点不合适.因此,本文提出一种基于连接交换(link exchange,简称 LX)的 unchoking 节点选择策略.该选择方法可从根本上抑制搭便车节点获得下载,因为搭便车节点就不会进入系统.另外,为提高 LX unchoking 策略发现有效下载节点的概率,本文在此基础上提供了一种可调节的概率式连接交换 unchoking 策略,并给出了详细 unchoking 协议实现来 PLXU.

References:

[1] Yang XY, Veciana GD. Service capacity of peer to peer networks. In: Neglia G, ed. Proc. of the IEEE Int'l Conf. on Computer Communications. New York: IEEE Communications Society Press, 2004. 2242−2252. [doi: 10.1109/INFCOM.2004.1354647]

[2] Sirivianos M, Park JH, Chen R, Yang XW. Free-Riding in BitTorrent networks with the large view exploit. In: Douceur JR, ed. Proc. of the 6th Int'l Workshop on Peer-to-Peer Systems. New York: Microsoft Press, 2007. 19−25.

[3] Locher T, Moor P, Schmid S, Wattenhofer R. Free-Riding in BitTorrent is cheap. In: Kohler E, ed. Proc. of the 5th Workshop on Hot Topics in Networks (HotNets). New York: ACM Press, 2006. 85−90.

[4] Piatek M, Isdal T, Anderson T, Krishnamurthy A, Venkataramani A. Do incentives build robustness in BitTorrent? In: Balakrishnan H, ed. Proc. of the 4th USENIX Symp. on Networked Systems Desgin and Implementation. California: USENIX Association, 2007. 1−14.

[5] Shneidman J, Parkes D, Massoulie L. Faithfulness in Internet algorithms. In: Ammar M, ed. Proc. of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems, New York: ACM Press, 2004. 220−227. [doi: 10.1145/1016527.1016537]

[6] Douceur JR. The sybil attack. In: Kaashoek F, ed. Proc. of the 1st Int'l Workshop on Peer-to-Peer Systems. Berlin: Springer-Verlag, 2002. 251−260. [doi: 10.1007/3-540-45748-8_24]

[7] Sirivianos M, Park JH, Yang XW, Jarecki S. Dandelion: Cooperative content distribution with robust incentives. In: Chase J, ed. Proc. of the 2007 USENIX Annual Technical Conf. California: USENIX Association, 2007. 157−170.

[8] Tan G, Jarvis SA. A payment-based incentive and service differentiation scheme for peer-to-peer streaming broadcast. IEEE Trans. on Parallel and Distributed System, 2008,19(7):940−953. [doi: 10.1109/TPDS.2007.70778]

[9] Kang X, Wu YD. A game-theoretic approach for cooperation stimulation in peer-to-peer streaming networks. In: Mattheisen C, ed. Proc. of the IEEE Int'l Conf. on Communications (ICC). New York: IEEE Communications Society Press, 2013. 2283−2287. [doi: 10.1109/ICC.2013.6654869]

[10] Wang J, Shen RM, Ullrich C, Luo H, Niu CY. Resisting free-riding behavior in BitTorrent. Future Generation Computer Systems, 2010,26(8):1285−1299. [doi: 10.1016/j.future.2009.05.014]

[11] Levin D, LaCurts K, Spring N, Bhattacharjee B. BitTorrent is an auction: Analyzing and improving BitTorrent's incentives. In: Seshan S, ed. Proc. of the Special Interest Group on Data Communication. New York: ACM Press, 2008. 243−254. [doi: 10.1145/1402958.1402987]

[12] Levin D, Sherwood R, Bhattacharjee B. Fair file swarming with FOX. In: Druschel P, ed. Proc. of the 5th Int'l Workshop on Peer-to-Peer Systems. New York: Microsoft Press, 2006. 91−96.

[13] Ngan TWJ, Wallach DS, Druschel P. Incentives-Compatible peer-to-peer multicast. In: Jackson M, ed. Proc. of the 2nd Workshop on the Economics of Peer-to-Peer Systems. Harvard University Press, 2004. 49−55.

[14] Izhak-Ratzin R, Park H, van der Schaar M. Online learning in BitTorrent systems. IEEE Trans. on Parallel and Distributed Systems, 2012,23(12):2280−2288. [doi: 10.1109/TPDS.2012.90]

[15] Li ML, Yu JD, Wu J. Free-Riding on BitTorrent-like peer-to-peer file sharing systems: Modeling analysis and improvement. IEEE Trans. on Parallel and Distributed Systems, 2008,19(7):954−966. [doi: 10.1109/TPDS.2007.70783]

[16] Fan B, Lui JCS, Chiu DM. The design trade-offs of BitTorrent-like file sharing protocols. IEEE/ACM Trans. on Networking, 2009, 17(2):365−376. [doi: 10.1109/TNET.2008.2002553]

[17] Legout A, Liogkas N, Kohler E, Zhang LX. Clustering and sharing incentives in BitTorrent systems. ACM SIGMETRICS Performance Evaluation Review, 2007,35(1):301−312. [doi: 10.1145/1269899.1254919]

[18] Altman E, Nain P, Shwartz A, Xu YD. Predicting the impact of measures against P2P networks: Transient behavior and phase transition. IEEE/ACM Trans. on Networking, 2013,21(3):935−949. [doi: 10.1109/TNET.2012.2217505]

[19] Liang ZQ, Shi WS. Analysis of ratings on trust inference in open environments. Performance Evaluation, 2008,65:99−128. [doi: 10.1016/j.peva.2007.04.001]

李治军(1977－),男,内蒙古伊盟人,博士,副教授,CCF 高级会员,主要研究领域为传感器网络,移动感知,普适计算.

姜守旭(1968－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为普适计算,数据库.

李晓义(1981－),男,硕士,主要研究领域为对等网络,网络激励.