高技术通讯 2012 年 第22卷 第12期:1250 ~ 1256

doi:10.3772/j.issn.1002-0470.2012.12.006

应对 P2P 直播瞬间拥塞的用户访问控制方法①

吴海博②*** 蒋 海* 孙 毅③* 李 军* 李忠诚*

(*中国科学院计算技术研究所 北京 100190)

(**中国科学院研究生院 北京 100049)

摘 要 针对 P2P 直播存在瞬间拥塞,导致用户启动延迟增大、系统服务拒绝率升高,从而降低用户体验的问题,提出了一种基于能力感知的用户访问控制算法。该算法可通过合理控制用户节点的加入速率,避免过多用户同时竞争带宽资源;优先允许高带宽用户节点接入,提高系统的服务能力;兼顾低带宽用户的等待时间,防止其因过长等待而离开。对该算法进行的数学建模分析、模型分析和相关实验表明,该算法能使 P2P 直播系统有效应对瞬间拥塞问题,改善用户服务质量。与两种控制方案的性能对比显示,该算法的平均延迟可分别降低 17% 和 29%,拒绝率可分别减少 60% 和 75%。

关键词 P2P 直播,瞬间拥塞,能力感知,用户访问控制,启动延迟

0 引 言

目前流媒体业务已成为大众网络视频消费的主要实现方式。P2P 直播由于其可扩展性好,部署成本低廉等优势,目前已成为流媒体业务的重要实现方式之一,得到了广泛应用和研究[1,2]。然而相关研究[3,4]表明 P2P 直播存在瞬间拥塞(flash crowd)问题,致使 P2P 直播应用面临挑战。用户在观看热门节目时,大量用户蜂拥而至,竞争有限的带宽资源,致使系统不能及时为大量用户提供服务,用户启动延迟增长,用户体验不佳,某些用户会多次重试甚至最终离开。针对这种情况,本文提出了一种基于能力感知的用户访问控制算法,该方法可有效应对 P2P 直播的瞬间拥塞问题。

1 相关研究

目前关于 P2P 直播中的瞬间拥塞问题的研究主要可分为两大类:实际系统测量和模型分析。文献[4]针对 Coolstreaming 系统进行了实际测量,着重研究瞬间拥塞对系统的影响以及期间的用户行为特征。Hei 等人针对 PPLive 系统进行了实际测量,发现 PPLive 直播春晚时存在瞬间拥塞问题,并对用户数量和用户行为进行了测量与跟踪[5]。文献[6]通过对 Akamai 公司提供的 P2P 直播业务进行测量,验证了大规模 P2P 直播业务的可行性。同时发现在大规模直播系统中瞬间拥塞问题普遍存在,用户具有很高的到达速率和加入失败率。Wu 等人基于 UUSee 系统的历史数据研究了 P2P 直播的拓扑特征,发现系统在发生瞬间拥塞时,用户的邻居数目会有所增加[7],用户间的带宽特性与常规情况相比,并没有显著区别[8]。综上所述,文献[4-8]主要是针对实际系统的测量研究,旨在揭示瞬间拥塞过程中的用户行为和系统特征。在数学建模方面,文献[9]提出了一种描述系统规模随时间变化的数学模型,同时给出了在一定时间约束情况下系统规模的上限值,并揭示了影响系统可扩展性的一些关键因素。文献[10]进一步研究了用户加入和离开模式对系统可扩展性以及用户启动延迟的影响。文献[11]同样提出了一种刻画系统规模随时间变化的数学模型,借助模型揭示影响系统规模的相关因素,并同时给出用户启动延迟的分布。文献[12]提出一种流模型以刻画 P2P 直播系统在发生瞬间拥

① 国家自然科学基金(61003266),973 计划(2012CB315802)和内蒙古自治区自然科学基金(2011MS0902)资助项目。

② 男,1981 年生,硕士;研究方向:P2P 网络,流媒体;E-mail: wuhaibo@ict.ac.cn

③ 通讯作者,E-mail: sunyi@ict.ac.cn

(收稿日期:2011-10-13)

塞时的系统特性。

在其他 P2P 应用方面,Veciana 等人对 BitTorrent 系统的服务能力进行了建模,针对瞬间拥塞的场景,提出一种描述系统服务能力增长情况的模型,并通过 BitTorrent 系统实际数据验证了模型的正确性[13]。关于 Web 服务器的瞬间拥塞,Rubenstein 等人针对一种分布式随机 P2P 协议的可扩展性提出理论分析模型,并实验验证了协议的有效性[14]。

上述研究分为系统测量和数学建模两个方面,主要关注对问题的理解和刻画,但缺少具体有效的解决方法。本文针对瞬间拥塞提出的能力感知的用户访问控制算法,可控制用户的访问速率,优先选择服务能力强的节点加入,同时兼顾低带宽用户的等待时间,防止其等待时间过长而离开。本文同时提出一种数学分析模型,对算法进行性能评价。相关模型分析和仿真实验结果表明,该算法能有效应对瞬间拥塞问题,降低用户启动延迟和系统的服务拒绝率,改善用户的服务质量。

2 基于能力感知的用户访问控制算法

本文提出的基于能力感知的用户访问控制算法的主要思想包括三个方面:(1)将用户访问速率控制在一个合理的水平,以避免用户过分竞争有限的带宽资源而阻碍系统规模的增长;(2)优先允许高带宽用户接入,使其尽快获得资源并转变为服务节点,从而加快系统服务能力的增长;(3)限制低带宽用户的等待时间,当低带宽用户的等待时间接近最大等待时间时及时允许其接入,从而避免低带宽用户超时离开。

2.1 系统架构

本文给出的与用户访问控制相关的系统架构如图 1 所示。访问控制算法运行在索引服务器上。索引服务器通过控制索引信息的返回来控制用户接入,包括控制用户的接入速率以及用户的接入顺序。新到达的用户向索引服务器请求索引信息并同时上报自身相关信息,包括上传带宽(能力),从而使索引服务器实现能力感知。当一个新用户被允许接入时,索引服务器向其返回 peer 列表,此后该用户根据返回的索引信息与相应用户建立连接,并与之相互上传和下载数据。

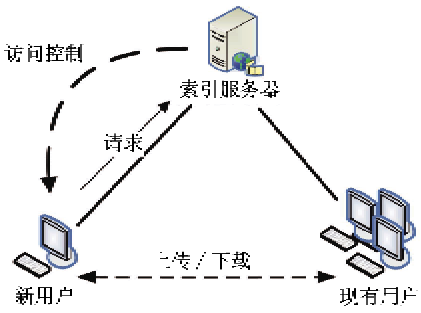

图 1 系统架构图

2.2 用户访问速率的设计

用户访问控制速率是用户访问控制算法的核心,因此我们在讨论具体算法之前首先介绍访问控制速率的设计。如果用户访问速率过高,则无法很好地避免用户对有限带宽资源的竞争,相反,如果用户接入速率过低,又会导致用户等待时间增长。受文献[11]中"系统剩余服务能力"概念的启发,我们根据系统剩余服务能力设计用户访问控制速率。首先提出一种针对带宽相同场景的访问控制速率的计算方法。带宽相同场景下,所有用户带宽均相同,且新用户的到达速率始终高于受控的用户访问速率。此后,我们将带宽相同场景的计算方法扩展到带宽不同的场景。

我们的速率控制方法采用固定的控制间隔,该时间间隔称为时间片。时间片的时长应是从用户开始寻找上游节点到下载到足够数据以满足缓冲区视频播放需求的时间,其典型值是 10 ~ 20s[4]。我们根据当前时间片内系统的剩余服务能力来计算下一时间片内允许接入的用户数目,即用户访问速率,从而给出计算每一时间间隔内用户访问速率的公式。

首先考虑带宽相同场景的情况。表 1 给出了相关符号定义。$h = (u - R)/R$ 表示相对平均用户剩余带宽。e 表示上传带宽利用率,其数值取决于 P2P 直播系统的具体数据分发策略。$ACR(t)$ 表示在第 t 个时间片内允许接入的用户数量。由于 R 代表视频资源码率,且用户观看直播视频,因此R也代表

表 1 符号定义

符号	定义
M	初始系统规模,即系统中全部用户的数目
u(kpbs)	用户的上传带宽(用户带宽均相等的情况)
R(kpbs)	视频码率
h	用户的平均相对剩余能力(为$(u-R)/R$)
e	用户上传带宽利用率
$ACR(t)$	第 t 个时间片内授权访问的用户数目
g	$ACR(1)$ 为第一个时间片内授权访问的用户数目
$\overline{u}(t)$(kpbs)	第 t 个时间片内所有用户的平均上传带宽

用户的平均下载速率,从而 $(u-R)/R$ 表示为用户平均相对剩余上传带宽。根据系统剩余服务能力,即系统剩余上传带宽的数量,我们可得到第一个时间片内允许接入的用户数量

$$ACR(1) = M \times [(u-R)/R] \times e \quad (1)$$

同理,由于 $(M+ACR(1)) \times [(u-R)/R] \times e$ 表示第二个时间片开始时,系统可支持的用户数量,若令 $ACR(1) = g$,则可得到第二个时间片内允许接入的用户数量

$$\begin{aligned} ACR(2) &= (M+ACR(1)) \times [(u-R)/R] \times e \\ &= (1+h \times e) \times g \end{aligned} \quad (2)$$

用类似方法,我们可以得到第三个时间片内允许接入的用户数量

$$\begin{aligned} ACR(3) &= (M+ACR(1)+ACR(2)) \\ &\quad \times [(u-R)/R] \times e \\ &= (1+h \times e)^2 \times g \end{aligned} \quad (3)$$

依此类推,最后我们得到带宽相同的情况下用户访问控制速率的表达式

$$ACR(t) = (1+h \times e)^{t-1} \times g \quad (4)$$

我们可以看出带宽相同场景的表达式,反映在初始规模一定的情况下,系统服务能力的最大增长速度,因此也是用户访问控制速率的上限值。

下面我们考虑带宽存在差异场景下用户访问控制速率的设计,并考虑一些实际因素。现实中,一方面用户通过多种方式接入 Internet(以太网、非对称数字用户线路(ADSL)等),因此用户带宽实际上存在差异,并非全部相等。另一方面,用户到达速率也会对用户访问控制速率产生影响。例如,有时用户到达速率可能会低于访问控制速率,因此可能无法达到式(4)中的上限值,式(4)此时将不再适用。

考虑到实际用户带宽差异和用户到达速率对访问控制的影响,对等式(4)进行扩展,我们得到多种带宽场景下用户访问控制速率的表达式

$$ACR(t) = S(t-1) \times [(\bar{u}(t-1)-R)/R] \times e \quad (5)$$

$\bar{u}(t-1)$ 表示第 $t-1$ 个时间片内系统中所有用户的平均上传带宽,此值可通过索引服务器搜集并统计。因此 $(\bar{u}(t-1)-R)/R$ 代表第 $t-1$ 个时间片内用户的平均相对剩余带宽。

2.3 用户访问控制算法

当检测到产生瞬间拥塞时,新用户的到达速率超过一定的经验阈值,索引服务器将执行用户访问控制算法。当用户到达速率降低或系统已具有足够能力以应对瞬间拥塞时,算法即可终止。如表 2 所示,算法由 3 个过程组成。过程 1 负责维护能力队列(capacity-ordered queue),过程 2 负责维护对应各个时间片的超时队列(timeout queue)。过程 3 是算法的主体,且具体实现基于以上两种队列。

表 2 用户访问控制算法

过程 1:
/* 索引服务器维护能力队列 */
(1) while (存在瞬间拥塞)
(2) 如果收到一个新用户的加入请求
(3) 则将用户放入能力队列,并按能力和时间排序

过程 2:
/* 索引服务器创建超时队列 */
(1) while (存在瞬间拥塞)
(2) 如果第 i 个时间片来临
(3) 创建当前时间片对应的超时队列 TQ_{i+Tp}

过程 3:
(1) 每个时间片内循环执行以下各步:
(2) 按照等式(5),计算当前时间片内允许接入的用户数目;
(3) 如果当前时间片的超时队列为空,即没有用户即将等待超时,则直接从能力队列头部取出 $ACR(t)$ 个用户,向这些用户返回 peer 列表并允许其加入系统;
(4) 否则,如果超时队列非空,且超时队列长度大于等于 $ACR(t)$,则从当前时刻对应的超时队列头部取出 $ACR(t)$ 个用户,向这些用户返回 peer 列表并允许其加入系统,同时从能力队列中删除相应用户信息;
(5) 否则,如果超时队列长度小于 $ACR(t)$,则首先取出超时队列内所有用户,向其返回索引信息并允许其接入系统,并从能力队列中删除相应用户信息;此后再按照剩余用户数目,从能力队列头部取出相应数目的用户,向这些用户返回 peer 列表并允许其接入,同时从多个超时队列中删除相应用户信息。

在介绍具体算法之前,我们首先介绍一下两种队列,即能力队列和超时队列,索引服务器在接收用户索引请求的同时维护两种队列。如图 2(a)所示,能力队列维护着当前所有尚未加入系统的新用户,用户首先按照能力由高到低排序,即高带宽用户更靠近队首,低带宽用户更靠近队尾。对于带宽相同的用户再按时间到达先后进行排序,即到达时间越早的用户越靠近队首,到达时间越晚的用户越靠近队尾。超时队列用于跟踪用户的等待时间,并防止低带宽用户等待时间过长而离开。每个时间片对应一个超时队列,每个超时队列记录着在某时间片内到达但至今尚未进入系统的用户,并记录剩余用户数目随时间的变化情况。如图 2(b)所示,假设用户最长等待时间为 T_p,根据每个时间片内(假设 T_i 时

刻)到达系统的用户,随即可以建立对应于 T_{i+Tp} 时刻的超时队列。如上所述,超时队列是多个队列,每个时间片对应一个。同样队列内用户也按照能力高低进行排列。在时间逐渐逼近 T_{i+Tp} 的过程中,随着用户的不断加入,该队列也会逐渐减少,如果时间到达 T_{i+Tp} 该队列仍不为空,则应优先允许其内用户接入系统,以避免用户等待超时。

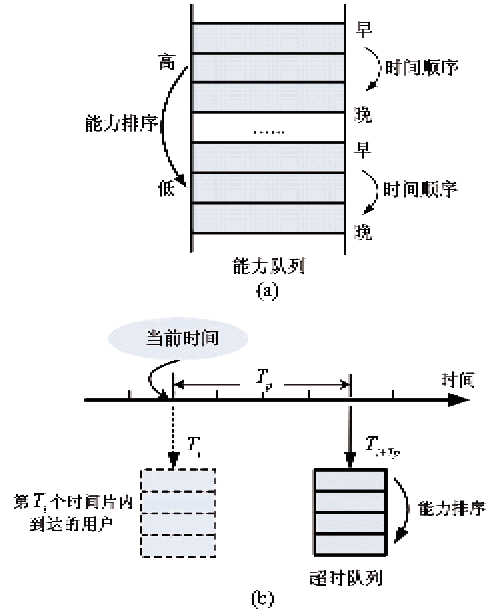

图 2　能力队列、超时队列

我们在此对本文算法——基于能力感知的用户访问控制算法进行介绍,重点阐述作为算法主体的过程 3。在过程 3 中,我们首先按照式(5)计算每一个时间片应加入的用户数目。基于此用户数,算法进而从能力队列或超时队列中取相应数目的用户并允许其加入,并向其返回索引信息(步骤(2))。在一般情况下,算法会从能力队列的首部依次取出用户,并允许其加入(步骤(3))。然而如果当前时间片对应的超时队列不为空,即存在某些用户即将等待超时,此时算法优先允许超时队列中的用户加入(从首部依次取出),然后再考虑从能力队列中选取剩余数目的用户加入系统(步骤(4)、(5))。通过这样的设计,算法既限制了用户的加入速率,又使高带宽用户优先加入,同时兼顾低带宽用户的等待时间。

3　分析模型

为了分析和评价上述算法,我们需要一种能描述瞬间拥塞状况下系统规模随时间变化的数学模型。文献[10]中已经给出一种分析模型,然而该模型无法描述用户访问控制,并假定所有用户具有相同的上传带宽,所以该模型不能用以描述用户访问速率以及用户带宽差异对系统规模增长的影响,因此我们需要设计新的分析模型。

我们将介绍一般性的分析模型,用于刻画系统规模随时间变化的情况,同时考虑用户访问速率控制和用户带宽差异。符号表示如表 3 所示。$S(t)$ 表示系统在第 t 个时间片内的系统规模,并假设码率 $R = xr$,这里表示以 Coolstreaming 为代表的 P2P 流媒体系统中的子流数目[2],x 表示子流的数目,r 表示每个子流的位速率。我们用 u 表示平均用户上传带宽。同时我们用 $v(t)$ 表示第 t 个时间片内用户的访问速率;假设按能力高低将用户分为 w 级,并用 $v_i(j)$ 表示第 i 类用户在第 j 个时间片时刻的访问控制速率。对于用户带宽存在差异的场景,系统规模随时间变化的情况可以表示如下:

表 3　模型相关参数

符号	定义
M	初始时系统规模,即系统中全部用户的数目
$S(t)$	到第 t 个时间片为止的系统规模(用户数目)
R (kbps)	视频码率 ($= xr$)
x	子流的数目
r (kbps)	每个子流的位速率
u (kbps)	用户平均上传带宽
u_i (kbps)	第 i 级用户的上传带宽
$v(t)$	t 时刻授权访问上传带宽 u 的用户数目
$v_i(t)$	t 时刻授权访问上传带宽为 u_i 的用户数目
$\bar{h}$	用户的相对平均剩余能力 ($= (u - R)/r$)
h_i	用户 i 的相对平均剩余能力 ($= (u_i - R)/r$)
w	所有用户上传带宽种类的数目
k	用户的邻居数目 ($\geqslant x$)

$$E[S(t)] \approx S(t-1) + \Big(\sum_{i=1}^{w}\sum_{j=1}^{t} v_i(j) + M - S(t-1)\Big) \times \sum_{i=x}^{k} C_k^i p(t,k,\bar{h})^i (1 - p(t,k,\bar{h}))^{k-i} \tag{6}$$

其中 $p(t,k,\bar{h}) \approx \bar{h} \times \alpha(t)/k$ 同样代表一个新用户从现有用户中获得一个单位上传带宽资源的概率。$\bar{h}$ 表示用户的相对平均剩余能力,w 表示所有用户上传带宽的类别数,k 是用户邻居的数目,并且 $\alpha(t)$ 的表达式如下:

$$\alpha(t) = S(t-1)/\Big(\sum_{i=1}^{w}\sum_{j=1}^{t} v_i(j) + M - S(t-1)\Big) \tag{7}$$

在此有必要说一下访问控制速率、算法以及模型三者间的关系。本文中的访问控制速率是由系统服务能力的增长情况决定，由式(5)计算可得每个时间片内允许进入的用户数目。基于这个数值，算法进一步决定每个时间片内允许哪些用户进入系统，以使用户总数等于该数值。最后，基于当前已经进入系统的用户，分析模型可以计算出每个时间片内，其中有哪些用户能够获得足够资源并变成服务节点，从而刻画系统规模随时间的变化情况。

4 算法性能评价

本节将通过模型分析和相关实验对算法进行验证和评价，4.1 节至 4.3 节将首先通过一系列实验验证算法设计思想的正确性。我们将分别讨论控制用户访问速率的作用、能力感知的作用以及将两者结合的有效性。4.4 节给出用户访问控制算法的性能评价，并验证算法的有效性。

4.1 用户访问控制的作用

我们将控制用户访问速率的方法与不控制用户访问速率的方法进行比较。对于无用户访问速率控制方法，为简便起见，我们假定所有用户同时到达，从而模拟实际用户的到达情况。对于控制用户访问速率的方法，用户被允许以恒定速率($v=3000$ 个/时间片)接入。两种方案中用户上传带宽均假定相同($u=60$kbps)。借助文献[10]推论 2 中的模型，我们对两种方法进行比较。如图 3 所示，实验结果表明，控制用户访问速率的方法要优于对用户访问速率不加控制的方法，即控制用户访问速率有助于系统应对瞬间拥塞。

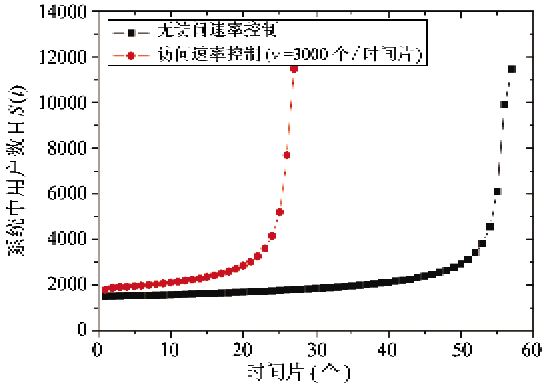

图 3 有访问速率控制与无访问速率控制对比

4.2 能力感知的作用

我们对能力感知方法与无能力感知方法进行比较，以说明能力感知的作用。不失一般性，我们仅考虑存在两种用户上传带宽的情况(多级带宽的情况可类比)。对于有能力感知的方法，允许数目为 $N/2$，带宽为 u_1 的用户先进入系统，数目为 $N/2$，带宽为 u_2 的用户后进入系统。我们采用式(6)对两种方法进行比较，并假定 $u_1=800$kbps，$u_2=400$kbps，$h=5$，$h_1=25/3$，$h_2=5/3$。如图 4 所示，实验结果表明，能力感知的方法相比于无能力感知的方法，系统规模会更快增长。高带宽用户的提前加入可以提高系统服务能力的增长速率，从而使系统能更好地应对瞬间拥塞。

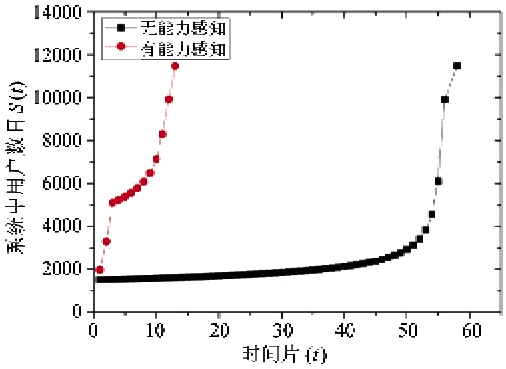

图 4 能力感知与无能力感知方案对比

4.3 能力感知和访问控制相结合

本节中我们将讨论能力感知与用户访问控制相结合方法的性能，并将其与单独采用能力感知以及单独采用访问控制的方法进行比较。我们使用式(6)中的模型，相关参数设置如下：对于控制用户访问速率的方法：$u=600$kbps，$h=5$，$v=2000$ 个/时间片。对于能力感知的方法：$u_1=800$kbps，$u_2=400$kbps，$h_1=25/3$，$h_2=5/3$。对于结合方法：$u_1=800$kbps，$u_2=400$kbps，$h_1=25/3$，$h_2=5/3$，$v=2000$ 个/时间片。

在图5中可以看到能力感知和访问控制相结合

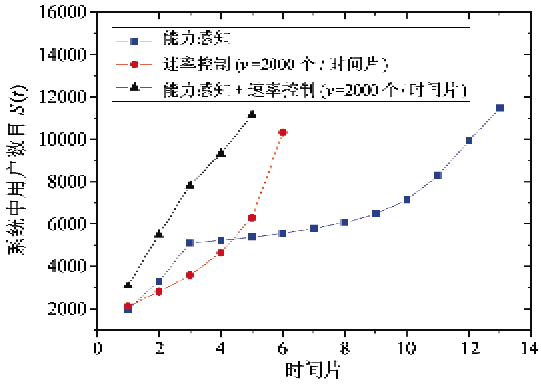

图 5 结合方案与单独能力感知、单独速率控制对比

的方法最快,在 5 个时间片内系统便可为所有到达用户提供服务。其次是控制用户访问速率的方法,能在 6 个时间片内接纳所有用户。能力感知的方法最慢,大约需要 13 个时间片。通过本节的实验,我们从而验证了结合方法的可行性和有效性。

4.4 算法性能评价

我们同时对算法进行性能评价。为方便起见,我们用方案 1 表示我们的算法,包含能力感知,速率控制,并控制低带宽用户的等待时间。方案 2 考虑变速的速率控制以及控制用户的等待时间,方案 3 采用能力感知的方法,并使用户按恒定速率加入,同时控制用户的等待时间。我们将从三个方面对三种方法进行比较:系统规模的增长速率,平均用户启动延迟、系统的服务拒绝率。

实验的相关参数设置如下:$M = 500$,$N = 60000$(全部新到达用户数目),$R = 300$kbps,*timeout*(最大等待时间)=3 时间片,$e = 90\%$,$k = 20$,$x = 5$。

方案 1:高带宽和低带宽用户各以每个时间片 3000 个的速度到达,用户访问速率借助式(5)来计算,且假定 $u_1 = 800$kbps,$u_2 = 400$kbps,用户最长等待时间为 3 个时间片。

方案 2:用户按每个时间片 6000 的速度到达,并同样采用式(5)来计算用户访问速率,并假设所有用户带宽均相同($u = 600$kbps),且用户最长等待时间为 3 个时间片。

方案 3:高带宽和低带宽用户各以每个时间片 3000 个的速度到达,用户访问速率为恒定值($v(t) = 4000$ 个/时间片),$u_1 = 800$kbps,$u_2 = 400$kbps,并且用户最长等待时间为 3 个时间片。

(1)系统规模增长速率

如图6所示,方案1在三种方案中系统规模的增长速度最快,系统能在 16 个时间片内达到大约 6000 个用户的系统规模。其次是方案 2,在 22 个时间片内可达到 50000 个用户的规模。方案 3 最慢,需要 31 个时间片才可达到大约 46000 个用户的规模。结果表明,方案 1 优于其它方法,这是因为方案 1 考虑适当限制用户的访问速率,并优先让带宽高的用户加入,从而加快了系统规模的增长速度。仅仅考虑一两种因素均达不到结合方法的性能。此外,通过比较方案 2 和方案 3,我们可以看出访问控制的改善作用要大于能力感知的改善作用。

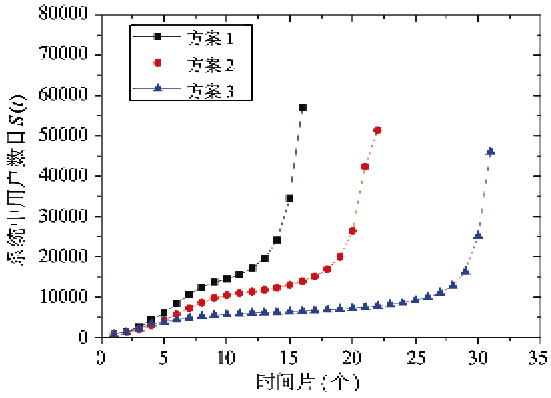

图 6　三种方案的系统规模增长速率对比

(2)平均启动延迟

如图 7 所示,关于平均启动延迟,方案 1 也同样优于其他两种方法。在方案 1 中的平均启动延迟大约为 2 个时间片,而方案 2 中大约为 2.7 个时间片,而方案 3 中的平均启动延迟则接近 3 个时间片。原因是方案 1 优先让带宽高的用户接入,并适当限制用户的访问速率,同时兼顾了低带宽用户的等待时间,从而使平均启动延迟大大缩短。

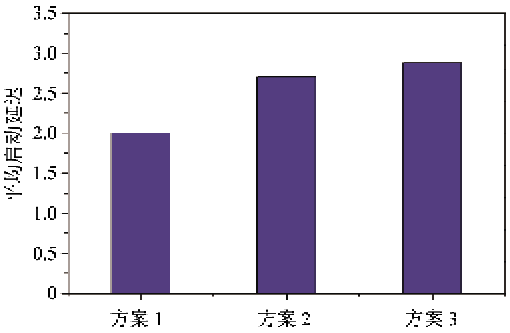

图 7　三种方案的平均启动延迟对比

(3)拒绝率

我们针对拒绝率对三种方案进行比较,结果如图 8 所示。方案 1 拒绝率仅有 5%,而方案 2 和方案 3 的拒绝率则高达 15% 和 25%。很明显,方案 1 中较低的拒绝率源自于较高的系统规模增长速率以及较小的启动延迟。系统能使用户更快加入,因而用

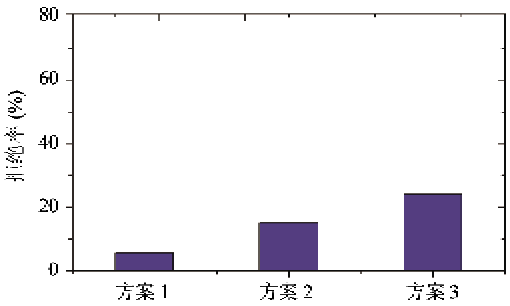

图 8　三种方案的拒绝率比较

户等待时间较短,所以用户离开率减小,即服务拒绝率较低。

5 结 论

针对 P2P 直播瞬间拥塞问题,本文提出一种基于能力感知的用户访问控制算法。该算法能够合理限制用户访问速率,允许高带宽用户优先接入,并且兼顾低带宽用户的等待时间。本文同时提出一种基于概率论的分析模型,并借助模型对算法进行性能分析。模型分析和仿真实验表明,该算法能有效加快系统规模的增长,降低用户启动延迟,并能改善系统的服务拒绝率。

参考文献

[1] PPLive. http://www.pplive.com

[2] Coolstreaming. http://www.coolstreaming.org

[3] Li B, Xie S, Qu Y, et al. Inside the new coolstreaming: principles, measurements and performance implications. In: Proceedings of the 27th International Conference on Computer Communications, Phoenix, USA, 2008. 1031-1039

[4] Li B, Keung G Y, Xie S, et al. An empirical study of flash crowd dynamics in a P2P-based live video streaming system. In: Proceedings of the IEEE Global Communications Conference, New Orleans, USA, 2008. 1-5

[5] Hei X, Liang C, Liang J, et al. A measurement study of a large-scale P2P IPTV system. *IEEE Transactions on Multimedia*, 2007, 9(8):1672-1687

[6] Sripanidkulchai K, Ganjam A, Maggs B, et al. The feasibility of supporting large-scale live streaming applications with dynamic application end-points. In: Proceedings of the ACM Special Interest Group on Data Communication, Portland, USA, 2004. 107-120

[7] Wu C, Li B C, Zhao S Q. Magellan: charting large-scale peer-to-peer live streaming topologies. In: Proceedings of the 27th International Conference on Distributed Computing Systems, Toronto, Canada, 2007. 1-8

[8] Wu C, Li B C, Zhao S Q. Characterizing peer-to-peer streaming flows. *IEEE Transactions on Selected Areas in Communications*, 2007, 25(9):1612-1626

[9] Liu F, Li B, Zhong L, et al. How P2P streaming systems scale over time under a flash crowd? In: Proceedings of the 8th International Conference on Peer-to-Peer Systems, Boston, USA, 2009. 1-5

[10] Liu F, Li B, Zhong L, et al. Understanding the flash crowd in P2P live video streaming systems. In: Proceedings of 17th International Packet Video Workshop, Seattle, USA, 2009. 1-10

[11] Chen Z, Li B, Keung G Y, et al. How scalable could P2P live media streaming system be with the stringent time constraint? In: Proceedings of the 2009 IEEE International Conference on Communications, Dresden, Germany, 2009. 1-5

[12] Chen Y S, Zhang B X, Chen C J. Modeling and performance analysis of P2P live streaming systems under flash crowds. In: Proceedings of the 2011 IEEE International Conference on Communications, Kyoto, Japan, 2011. 1-5

[13] Yang X, Veciana G D. Service capacity of peer to peer networks. In: Proceedings of the 23th International Conference on Computer Communications, Hong Kong, China, 2004. 2242-2252

[14] Rubenstein D, Sahu S. Can unstructured P2P protocols survive flash crowds? *IEEE Transactions on Networking*, 2005, 13(3):501-512

A user access control method for P2P live streaming systems under flash crowds

Wu Haibo[*,**], Jiang Hai[*], Sun Yi[*], Li Jun[*], Li Zhongcheng[*]

([*]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

([**]Graduate University, Chinese Academy of Sciences, Beijing 100049)

Abstract

Flash crowds often make users suffer from long startup delays, high reject rate and bad service. A novel user access control algorithm based on capacity-aware is proposed to tackle the problems flash crowds bring. The algorithm controls users to enter at a proper rate to prevent the high user arrival rate to slow down the increase of system scale. Also, it gives high-bandwidth users the priority to enter the system to increase the system service capacity as quickly as possible. Moreover, it considers the waiting time of the low-bandwidth users and prevent these users from leaving the system for waiting too long. A model for evaluation of the new user access control algorithm is also given. The model analysis and related experiments reveal that the new algorithm can effectively deal with the flash crowd problems and improve the quality of service for users by reducing the startup delay and service reject rate.

Key words: P2P live, flash crowd, capacity-aware, user access control, startup delay