P2P流媒体系统最大数据传输速率研究

吴国福[1],窦 强[2],温 俊[3],宋 磊[2],窦文华[2]
(1.国防科技大学航天与材料工程学院,湖南长沙 410073;
2.国防科技大学计算机学院,湖南长沙 410073;
3.总参第六十三所,江苏南京 210007)

摘 要: 个人计算机性能的提高和网络带宽的增加使得P2P流媒体应用系统迅速发展,本文对P2P流媒体系统模型和最大数据传输速率进行研究.提出一种P2P流媒体系统稳定状态下的系统模型,该模型使用较少的参数刻画系统在稳定状态下的属性.推导证明系统在稳定状态下支持的最大数据传输速率,并提出一种集中式算法WFSOT,快速构造支持最大数据传输速率的节点拓扑结构及分配节点带宽.对WFSOT算法进行分析,重点讨论节点在拓扑结构中深度的变化.

关键词: P2P流媒体;数据传输速率;拓扑结构;带宽分配;节点深度
中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2012) 03-0459-07
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2012.03.008

Research on Maximum Data Transmitting Rate of P2P Streaming System

WU Guo-fu[1], DOU Qiang[2], WEN Jun[3], SONG Lei[2], DOU Wen-hua[2]
(1. *College of Aerospace and Material Engineering, National University of Defense Technology, Changsha, Hunan* 410073, *China*;
2. *Computer School, National University of Defense Technology, Changsha, Hunan* 410073, *China*;
3. *The* 63*rd Institute of General Staff, Nanjing, Jiangsu* 210007, *China*)

Abstract: P2P streaming applications develop quickly with the enhancement of PC's performance and network bandwidth. In this paper the system model and the maximum data transmitting rate of P2P streaming system were studied. We presented a system model of P2P streaming system under stable state, which could capture the characteristic of the system by a few parameters. The maximum data transmitting rate of the stable system was derived and proven, and a centralized algorithm named WFSOT was designed to organize nodes into an overlay quickly which could support the maximum data transmitting rate and allocate nodes' practicable bandwidth. At last, we discussed several aspect of the WFSOT algorithm, especially the node's depth in the overlay topology.

Key words: P2P streaming; data transmitting rate; topology architecture; bandwidth allocation; node depth

1 引言

P2P流媒体技术充分利用客户端的闲置资源,提高系统可扩展性.现有的关于P2P流媒体的文献主要集中在构建合理的节点拓扑结构和媒体数据的调度方面[1~3].本文通过建立P2P流媒体系统模型,从理论上推导出P2P流媒体系统在稳定状态下支持的最大数据传输速率,文献[4]给出了类似的结论,与本文的不同之处在于:本文在证明过程中考虑了节点的下行带宽对最大数据传输速率的影响,同时证明过程中引入数据流树的概念,将内容数据分为分转发数据和广播数据,给出每种数据的分发模式.

2 P2P流媒体系统稳定状态下的系统模型

宏观上看,P2P流媒体系统数据传输部分由三大部分构成:源服务器、peer节点、Internet网络.下面首先对这三部分单独进行描述,抽取其关键部分进行建模,然后综合建立系统模型.

源服务器通常由多个服务器聚合起来对外提供服务.将这些服务器视为能力更强的聚合节点,整个系统只有一个虚拟源服务器.在P2P流媒体系统中,源服务器仅使用其上行带宽,因此我们用上行带宽 U_s 来描述源服务器.当系统中存在多个源服务器时,使用 U_s 表示虚拟服务器上行带宽,U_s 大小为各个源服务器的上行

收稿日期:2010-03-05;修回日期:2010-11-20
基金项目:国家自然科学基金(No.60603061,No.60603064)

带宽之和.

peer 节点的关键参数是端到端可用带宽.节点之间会因为瓶颈链路的不同而具有不同的端到端可用带宽.

首先我们考虑节点 A 和 B 之间的路径穿越 Internet 的情况,我们假定 A 和 B 之间的瓶颈链路只发生在 A 或者 B 的接入链路,这样 A 和 B 之间的可用带宽就取决于 A 或者 B 接入链路的上行带宽和下行带宽,仅使用 peer 节点接入链路的上行带宽和下行带宽就能够描述 peer 节点可用带宽.

其次我们考虑节点 A 和 B 处在同一个局域网络中的情况,A 与 B 之间的可用带宽很大.相比之下,A、B 和外部的其它节点之间的端到端可用带宽小得多,我们可以将 A、B 合并为一个超节点 C,超节点 C 的上行带宽和下行带宽分别设置为当该局域网络只存在节点 A 时,节点 A 对外的可用上行带宽和下行带宽.

上述两种情况覆盖了 Internet 中节点之间通路的绝大多数情况,我们仅用节点 i 的上行带宽 U_i 和下行带宽 D_i 来描述节点之间的端到端可用带宽,例如节点 A 到节点 B 的端到端可用带宽可以表示为 $\min\{U_A, D_B\}$.

Internet 网络提供源服务器、peer 节点之间的连接通路,可以将 Internet 视为一个超级虚拟节点,该节点具有无穷大的上行带宽和下行带宽.源服务器通过上行链路与 Internet 相连,peer 节点通过上行链路和下行链路与 Internet 相连接.这样 P2P 流媒体系统可以简化为图 1 所示.

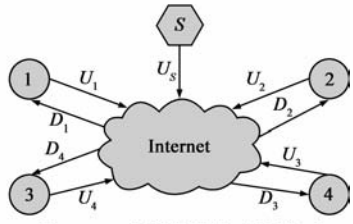


图1 P2P流媒体传输系统模型

表 1 常用符号及其意义说明

符号	意义说明
S	源服务器
N	peer 节点数量
U_i	peer 节点 i 的上行带宽
D_i	peer 节点 i 的下行带宽
$r_{\max}$	最大数据传输速率
$D_{\min}$	peer 节点中最小下行带宽,$\min\{D_i, i=1\cdots N\}$
U_{all}	系统可用上行带宽之和,$U_s+\sum_{i=1}^{N}U_i$
U_s	源服务器上行带宽
P	peer 节点集合,$\{P_i, i=1\cdots N\}$
U	peer 节点的上行带宽集合,$\{U_i, i=1\cdots N\}$
D	peer 节点下行带宽集合,$\{D_i, i=1\cdots N\}$
U_{sum}	peer 节点可用上行带宽之和,$\sum_{i=1}^{N}U_i$
U_{avg}	peer 节点平均可用上行带宽,$\frac{U_s+\sum_{i=1}^{N}U_i}{N}$

根据系统模型图,我们使用六元组$\langle S, P, U_s, U, D, N\rangle$来描述 P2P 流媒体系统,其中 S 表示源服务器,$P=\{P_i, i=1\cdots N\}$ 表示 peer 节点集合,U_s 表示源服务器的上行带宽,$U=\{U_i, i=1\cdots N\}$表示 peer 节点的上行带宽集合,$D=\{D_i, i=1\cdots N\}$表示 peer 节点的下行带宽集合,N 表示 peer 节点的数量.

表 1 列出文中常用符号表示及其意义说明.

3 P2P 流媒体系统最大数据传输速率

本节我们推导证明 P2P 流媒体系统支持的最大数据传输速率.

定义 1 P2P 流媒体系统数据传输速率 r 定义为 peer 节点接收到不重复数据的速率,或者视频数据编码后的码流速率.

要达到系统支持的最大数据传输速率,直观上认为应该尽可能地利用所有参与节点的可用带宽,同时 peer 节点接收的数据不能重复,因为重复数据相当于浪费可用带宽.每个 peer 节点要接收到相同速率的数据,那么最大数据传输速率必然受限于 peer 节点的最小下行带宽.综合上述认识,我们给出 P2P 流媒体系统在稳定状态下支持的最大数据传输速率表达式.

定理 1 P2P 流媒体系统在稳定状态下支持的最大数据传输速率 $r_{\max}$为

$$r_{\max}=\min\{U_s, U_{\text{avg}}, D_{\min}\} \tag{1}$$

先给出证明的基本思路.数据分发主要涉及三个方面:拓扑结构、节点的带宽分配和数据调度.证明过程中我们采用同一拓扑结构:源服务器与所有 peer 节点连接,peer 节点采之间采用全互连方式连接,图 2(a)给出具有 4 个 peer 节点的拓扑结构图.根据数据的流动方向,我们可以将上述拓扑结构转化为数据流树结构,如图 2(b)是 2(a)等效的数据流树.数据流树很直观地显示节点之间的数据传输及带宽分配,同时最大延迟等同于数据流树的高度.节点的带宽分配如下:源服务器分配给每个 peer 节点的带宽根据 peer 节点上行带宽的不同而有所变化,每个 peer 节点的上行带宽平均分成$(N-1)$份,分配给其它 peer 节点.数据调度在证明过程中描述.证明过程先推导出整个系统支持的数据传输速率上限,然后证明数据传输速率可以达到该上限,从而定理得到证明.

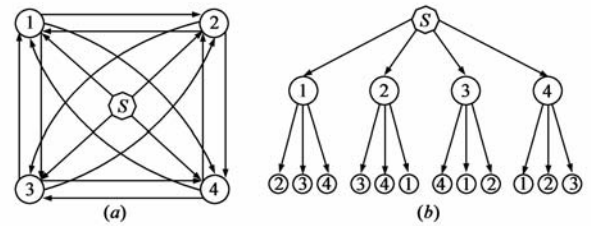


图2 P2P系统拓扑结构及其数据流树

证明 1　$r_{\max}(\min\{U_s, U_{\text{avg}}, D_{\min}\}$　(2)

源服务器作为数据源,新产生的数据都需要从源服务器发送给 peer 节点,如果码流速率 $r > U_s$,那么必然有部分数据无法发送给 peer 节点,所以 $r \leqslant U_s$,即

$$r_{\max} \leqslant U_s \tag{3}$$

peer 节点得到的数据均来自源服务器和其它 peer 节点,节点发送的数据也都被系统中的 peer 节点接收,因此系统对数据流是封闭的,系统所有节点的发送速率之和 T 与 peer 节点接收速率之和 R 相等(源服务器不接收数据),即 $T = R$. 系统中节点能够提供的发送速率之和最大为所有节点的上行带宽之和,即 U_{all},故 $T \leqslant U_{\text{all}}$. 系统中 peer 节点可能接收到重复数据,故 $N * r \leqslant R$. 由不等式的传递关系可得 $N * r \leqslant U_{\text{all}}$, $r \leqslant U_{\text{avg}}$,即

$$r_{\max} \leqslant U_{\text{avg}} \tag{4}$$

peer_i 能够接收的最大数据传输速率受限于其下行带宽,数据传输速率 r 不可能高于其下行带宽 D_i,可得 $r \leqslant D_i$,即

$$r_{\max} \leqslant \min\{D_i, i = 1 \cdots N\} \tag{5}$$

由不等式(3) ~ (5)得不等式(2)成立.

证明 2　$r_{\max} \geqslant \min\{U_s, U_{\text{avg}}, D_{\min}\}$　(6)

证明过程中我们将数据分为广播数据和转发数据,当 peer 节点接收到广播数据时,将其锁定在本地,不再向外转发;当接收到转发数据时,将类型修改为广播数据,发送给其它($N-1$)个 peer 节点. 下面分 4 种情况证明不等式(6).

情况 1　当$(N-1)U_s \geqslant U_{\text{sum}}$并且 $D_{\min} \leqslant U_{\text{avg}}$时,源服务器分配给每个 peer 节点 i 的带宽为$\frac{U_i}{N-1} + \frac{1}{N}(U_s - \frac{U_{\text{sum}}}{N-1})$,其中前一部分用来发送转发数据,刚好为节点 i 上行带宽的$\frac{1}{N-1}$,peer_i 收到转发数据后广播给其它($N-1$)个 peer 节点,充分利用 peer_i 的上行带宽;后一部分用来发送广播数据. 当单位数据产生后,源服务器将数据分成 2 大部分,一部分为转发数据,另一部分为广播数据. 其中转发数据大小为$\frac{\frac{U_{\text{sum}}}{N-1}}{\frac{U_{\text{sum}}}{N-1} + \frac{1}{N}(U_s - \frac{U_{\text{sum}}}{N-1})}$,分配给 peer_i 的转发数据大小为$\frac{\frac{U_i}{N-1}}{\frac{U_{\text{sum}}}{N-1} + \frac{1}{N}(U_s - \frac{U_{\text{sum}}}{N-1})}$;广播数据大小为$\frac{\frac{1}{N}(U_s - \frac{U_{\text{sum}}}{N-1})}{\frac{U_{\text{sum}}}{N-1} + \frac{1}{N}(U_s - \frac{U_{\text{sum}}}{N-1})}$.

peer 节点从源获得转发数据和广播数据,从其它($N-1$)个 peer 节点处获得广播数据. peer_i 接收数据的速率为

$$\begin{aligned} r_i &= \frac{U_i}{N-1} + \frac{1}{N}(U_s + \frac{U_{\text{sum}}}{N-1}) + \sum_{j=1, j\neq i}^{N} \frac{U_j}{N-1} \\ &= \sum_{j=1}^{N} \frac{U_j}{N-1} + \frac{1}{N}(U_s + \frac{U_{\text{sum}}}{N-1}) \\ &= \frac{U_s + U_{\text{sum}}}{N} = U_{\text{avg}} \end{aligned}$$

这些数据均没有重叠成分,因此 peer 节点接收数据的速率就是数据传输速率. 由于 $D_{\min}(U_{\text{avg}}$,每个节点的下行带宽都不小于节点的数据接收速率,数据可以完全接收. 系统支持的最大数据传输速率 $r_{\max} \geqslant r_i$,即

$$r_{\max} \geqslant U_{\text{avg}} \tag{7}$$

$N = 1$ 的情况我们最后处理.

情况 2　当$(N-1)U_s \geqslant U_{\text{sum}}$并且 $D_{\min} < U_{\text{avg}}$时,如果仍旧按情况 1 中的情况进行带宽分配和数据调度,必然存在 peer 节点由于下行带宽的限制无法接收全部数据. 我们只使用节点上行带宽的 $D_{\min}/U_{\text{avg}}$,剩下部分闲置. 对于部分可用上行带宽,仍旧按照情况 1 中情况进行处理,每个 peer 节点接收数据的速率为

$$r_i = U_{\text{avg}} * (D_{\min}/U_{\text{avg}}) = D_{\min}$$

节点的下行带宽 $D_i \geqslant D_{\min} = r_i$,节点可以按照速率 r_i 接收. 系统支持的最大数据传输速率 $r_{\max} \geqslant r_i$,即

$$r_{\max} \geqslant D_{\min} \tag{8}$$

情况 3　当$(N-1) * U_s < U_{\text{sum}}$并且 $D_{\min} \geqslant U_s$ 时,将源服务器的上行带宽按照 U_i/U_{sum}的比例分配给 peer_i,即分配给 peer_i 的带宽为$(U_i/U_{\text{sum}}) * U_s$;peer 节点将其上行带宽平均分配给其它($N-1$)个 peer 节点. 当源产生单位数据后,源服务器将数据按照其带宽分配的比例进行划分,作为转发数据发送给 peer 节点,peer 节点收到源服务器发送的转发数据后,将数据变换为广播数据,发送给其它($N-1$)个 peer 节点. 每个 peer 节点都接收到来自源服务器的转发数据和来自其它($N-1$)个 peer节点的广播数据. 源服务器发送数据的速率为 $\sum_{i=1}^{N} (U_i/U_{\text{sum}}) * U_s = U_s$,没有超出源服务器上行带宽限制;$\text{peer}_i$ 发送数据的速率为$(N-1) * (U_i/U_{\text{sum}}) * U_s < (U_i/U_{\text{sum}}) * U_{\text{sum}} = U_i$,没有超出其上行带宽的限制. peer_i 接收数据的速率为

$$\begin{aligned} r_i &= \frac{U_i}{U_{\text{sum}}} * U_s + \sum_{j=1, j\neq i}^{N} \frac{U_i}{U_{\text{sum}}} * U_s = \sum_{j=1}^{N} \frac{U_i}{U_{\text{sum}}} * U_s \\ &= \frac{U_s}{U_{\text{sum}}} * U_{\text{sum}} = U_s \end{aligned}$$

接收数据速率没有超出其下行带宽 $D_{\min} \geqslant U_s$ 的限制,同时收到的数据没有重复,peer_i 接收数据的速率就

是系统的数据传输速率,系统支持的最大数据传输速率 $r_{\max} \geqslant r_i$,即

$$r_{\max} \geqslant U_s \tag{9}$$

情况 4 当$(N-1)U_s < U_{\text{sum}}$并且 $D_{\min} < U_s$ 时,如果仍按情况 3 中的情况进行带宽分配和数据调度,必然存在 peer 节点由于下行带宽的限制无法接收完整数据.我们只使用节点上行带宽的 $D_{\min}/U_s$,剩余部分空闲置.对于部分可用上行带宽,仍旧按照情况 3 中情况进行处理,每个 peer 节点接收数据的速率为

$$r_i = U_s * D_{\min}/U_s = D_{\min}$$

节点的下行带宽 $D_i \geqslant D_{\min} = r_i$,节点可以按照速率 r_i 接收.系统支持的最大数据传输速率 $r_{\max} \geqslant r_i$,即

$$r_{\max} \geqslant D_{\min} \tag{10}$$

上述四种情况涵盖了除 $N=1$ 外的所有状态,当 $N=1$,系统中只存在一个 peer 节点,这时的最大数据传输速率受到 U_s 和 D_1 限制,$r_{\max} = \min\{U_s, D_1\}$,即

$$r_{\max} = \min\{U_s, D_{\min}\} \tag{11}$$

由不等式(7)~(10)和等式(11)可得不等式(6)成立.

由不等式(2)、(6)可得等式(1)成立,定理 1 证明完毕.

4 WFSOT 算法设计与证明

定理 1 的证明过程给出一种实现最大数据传输速率的集中式算法,但是该算法采用 peer 节点全互连的方式,随着节点数量增加,系统控制开销急遽增大,可扩展性差.那么是不是存在其它形式的拓扑结构,在合理的带宽分配和数据调度下也能达到最大数据传输速率呢?下面我们给出一种新的拓扑结构支持 $r_{\max}$.

上一节证明过程讨论了 4 种不同情形,我们认为情况 1 在实际中最具代表性.在后面的讨论中,我们主要针对情况 1 进行讨论并且认为系统中 peer 节点的数量 $N>1$,其余三种情形可做类似的讨论.

情况 1 中最大数据传输速率为 $r_{\max} = U_{\text{avg}}$,要达到该速率必须满足以下 2 个必要条件:

条件 1 源服务器和 peer 节点的上行带宽完全得到利用.

条件 2 peer 节点收到的数据无重复部分.

证明 在达到数据最大传输速率的情况下,考虑到节点可能接收到重复数据,系统需要的总上行带宽 $\Omega \geqslant N * r_{\max} = U_{\text{all}}$,即不小于 U_{all};如果节点的上行带宽没有得到完全利用,即参与数据传输的总上行带宽小于 U_{all},相互矛盾,所以条件 1 是必要的.如果 peer 节点接收到重复数据,那么在达到最大数据传输速率的情况下,系统需要的上行带宽 $\Omega > N * r_{\max}$,即大于 U_{all};系统所能提供的最大上行带宽为 U_{all},矛盾,故条件 2 是必要的.

下面我们提出一种集中式的算法用来求解支持最大数据传输速率的拓扑结构及其对应的带宽分配.源服务器集中服务上行带宽最大的几个 peer 节点,使得每个节点都尽可能得到最大带宽.源服务器上行带宽分配结束后依次分配已经获得最大带宽节点的上行带宽给后续节点,上行带宽最小的节点最后得到服务.如果一直按这种方式操作,最后节点的上行带宽无法利用,所以最后几个节点需要相互贡献上行带宽.我们称该算法为带宽优先、顺序满足的拓扑构造算法,简称为 WFSOT.WFSOT 算法形式化描述如表 2.

表 2 WFSOT 算法形式化描述

输入:N 节点数量,$U[N+1]$节点上行带宽,假设源服务器的上行带宽最大

输出:allocate$[N+1, N]$节点间的连接关系及带宽分配,depth$[N+1]$节点在数据流树中的深度

WFSOT:

步骤 1 按节点上行带宽降序排列 U

步骤 2 求满足$(k-1) * r_{\max} \geqslant \sum_{j=N-k+1}^{N} U[j]$并且 $k * r_{\max} \geqslant \sum_{j=N-k}^{N} U[j]$最小 k 值$(k \geqslant 2)$

步骤 3 循环 $i = 1:(N-k)$

步骤 4 $r = r_{\max}$

步骤 5 循环 $j = 0:(i-1)$

步骤 6 如果 $U[j] >= r$,则 $U[j] = U[j] - r$, allocate$[j, i] = r$, depth$[i]$ = depth$[j]$ + 1;

步骤 7 否则 $r = r - U[j]$, allocate$[j, i] = U[j]$, $U[j] = 0$, depth$[i]$ = depth$[j]$ + 1;

步骤 8 最后 k 个节点全互连,每个节点将上行带宽平均分配给其它$(k-1)$个节点,更新 allocate

步骤 9 循环 $i = (N-k+1):N$

步骤 10 $r = r_{\max}$ − sum(allocate$(i, :)$)

步骤 11 循环 $j = 0:(N-k)$

步骤 12 如果 $U[j] >= r$,则 $U[j] = U[j] - r$, allocate$[j, i] = r$, depth$[i]$ = depth$[j]$ + 1;

步骤 13 否则 $r = r - U[j]$, allocate$[j, i] = U[j]$, $U[j] = 0$, depth$[i]$ = depth$[j]$ + 1;

步骤 14 循环 $i = (N-k+1):N$ depth$[i]$ = depth$[i]$ + 1;

下面我们证明 WFSOT 算法的正确性,即证明下面的定理成立.

定理 2 WFSOT 算法结束后,每个 peer 节点都分配得到最大数据传输速率的带宽

证明 首先证明存在 k 值满足表 1 步骤 2 中的条件.当 $k=N$ 时,满足步骤 2 中的条件,

因为 $r_{\max} = \dfrac{U_0 + \sum_{j=1}^{N} U[j]}{N} \leqslant \dfrac{U_0 + (N-1)U_0}{N} = U_0$

所以$(N-1)*r_{\max}=U_0-U_0+N*r_{\max}-r_{\max}$

$$=(U_0-r_{\max})+(U_0+\sum_{j=1}^{N}U[j])-U_0$$

$$\geqslant\sum_{j=1}^{N}U[j]$$

$$=\sum_{j=N-N+1}^{N}U[j] \tag{12}$$

$$N*r_{\max}=U_0+\sum_{j=1}^{N}U[j]=\sum_{j=0}^{N}U[j]$$

$$=\sum_{j=N-N}^{N}U[j] \tag{13}$$

由不等式(12)、(13)可得,存在 k 同时满足步骤 2 中的不等式,所以存在最小 k 值($k\geqslant 2$).

循环步骤 3 ~ 步骤 7 依次为节点 1 ~ $(N-k)$分配最大数据传输速率带宽,若要使得带宽分配可行,只要满足下列不等式(14)即可

$$\sum_{j=0}^{i-1}U[j]\geqslant i*r_{\max}(1\leqslant i\leqslant(N-k)) \tag{14}$$

假设不等式(14)不成立,即 $\sum_{j=0}^{i-1}U[j]<i*r_{\max}$,由于 U 是按降序排列的,故有 $i*r_{\max}>\sum_{j=0}^{i-1}U[i-1]=i*U[i-1]$,即

$$r_{\max}>U[i-1] \tag{15}$$

由步骤 2 知 $k*r_{\max}\geqslant\sum_{j=N-k}^{N}U[j]$,可得 $\sum_{j=0}^{N}U[j]-\sum_{j=N-k}^{N}U[j]\geqslant\sum_{j=0}^{N}U[j]-k*r_{\max}=(N-k)*r_{\max}$,即

$$\sum_{j=0}^{N-k-1}U[j]\geqslant(N-k)*r_{\max} \tag{16}$$

由假设与不等式(16)可得 $\sum_{j=0}^{N-k-1}U[j]-\sum_{j=0}^{i-1}U[j]>(N-k)*r_{\max}-i*r_{\max}$,即 $\sum_{j=i}^{N-k-1}U[j]>(N-k-i)*r_{\max}$,$U$ 按降序排列可得 $\sum_{j=i}^{N-k-1}U[j]\leqslant\sum_{j=i}^{N-k-1}U[i]=(N-k-i)*U[i]$,即

$$U[i]>r_{\max} \tag{17}$$

由不等式(15)、(17)可知当 $i<(N-k)$时,$U[i]>U[i-1]$,出现矛盾,不等式(15)成立. 当 $i=(N-k)$,由不等式(16)可知不等式(14)成立.

循环步骤 9 ~ 步骤 13 为剩下的 k 个节点分配带宽,该循环将前$(N-k)$个节点剩余的上行带宽分配给这 k 个节点. 由于在步骤 8 中已经将剩余的 k 个节点上行带宽相互分配,所以要完全利用节点的上行带宽,对于节点 $i((N-k+1)\leqslant i\leqslant N)$必须满足

$$r_{\max}-\sum_{j=N-k+1,j\neq i}^{N}\frac{U[j]}{(k-1)}\geqslant\frac{U[i]}{(k-1)} \tag{18}$$

也就是节点 i 从前$(N-k)$个节点获得带宽必须不小于自身分配给其它节点的带宽,否则分配给其它节点的上行带宽不能完全利用. 不等式(18)等价于

$$r_{\max}-\sum_{j=N-k+1}^{N}\frac{U[j]}{(k-1)}\geqslant 0 \tag{19}$$

由步骤 2 中的条件可知不等式(19)成立.

每个 peer 节点分配得到的最大带宽为 $r_{\max}$,所有节点的上行带宽都可得到完全利用. 当所有 peer 节点得到最大带宽时恰好所有节点上行带宽都得到完全利用,故 peer 节点在算法结束时能够得到最大带宽.

至此,我们证明了 WFSOT 算法运行中的条件都得到满足,算法结束时每个节点都获得最大数据传输速率带宽 $r_{\max}$,即定理 2 成立.

拓扑结构和带宽分配完成后,需要进行数据调度使得系统在运行的时候每个节点都获得最大数据传输速率. 假设节点接收到单位数据后才进行转发. 对于前$(N-k)$个 peer 节点,不存在相互交换数据,假设节点 $i(1\leqslant i\leqslant N-k)$由 m 个父节点提供数据,每个父节点提供的带宽为 $c_j(j=1,\cdots,m,\sum_{j=1}^{m}c_j=r_{\max})$,则节点 n_j 将单位数据中的$(\sum_{p=0}^{j-1}c_p,\sum_{p=0}^{j}c_p]$部分发送给节点 i. 对于后 k 个节点,它们之间要相互交换数据,由条件 2 知相互交换的数据不能存在重复部分,因此在数据调度的时候就要保证这一点. 由 WFSOT 步骤 2 中的条件$(k-1)*r_{\max}\geqslant\sum_{j=N-k+1}^{N}U[j]$可得$\frac{1}{(k-1)}\sum_{j=N-k+1}^{N}U[j]\leqslant r_{\max}$,这保证相互交换的数据在合理分配后不存在重复部分. 从前$(N-k)$个节点分配给节点 $i(N-k+1\leqslant i\leqslant N)$的交换部分数据为$(\frac{1}{(k-1)}\sum_{j=N-k+1}^{i-1}U[j],\frac{1}{(k-1)}\sum_{j=N-k+1}^{i}U[j]]$,相互交换后节点 i 得到的数据为$(0,\frac{1}{(k-1)}\sum_{j=N-k+1}^{N}U[j]]$,对于剩下$(\frac{1}{(k-1)}\sum_{j=N-k+1}^{N}U[j],r_{\max}]$部分继续从前$(N-k)$个节点处获得. 假设节点 $i(N-k+1\leqslant i\leqslant N)$由前$(N-k)$个节点中的 m 个提供数据,每个节点提供的带宽为 $c_j(j=1,\cdots,m)$,假设 $\sum_{p=1}^{q}c_p\geqslant\frac{U[i]}{(k-1)}$,则节点 $n_j(0<j<q)$将单位数据中的$(\frac{1}{(k-1)}\sum_{p=N-k+1}^{i-1}U[p]+\sum_{p=1}^{j-1}c_p,\frac{1}{(k-1)}\sum_{p=N-k+1}^{i-1}U[p]+\sum_{p=1}^{j}c_p]$部分发送给节点 i,这部分为转发数据;节点 n_q 将$(\frac{1}{(k-1)}\sum_{p=N-k+1}^{i-1}U[p]$

$+\sum_{p=1}^{q-1} c_p, \frac{1}{(k-1)}\sum_{p=N-k+1}^{i} U[p]]$ 和 $(\frac{1}{(k-1)}\sum_{p=N-k+1}^{N} U[p], \frac{1}{(k-1)}\sum_{p=N-k+1}^{N} U[p]+\sum_{p=1}^{q} c_p-\frac{U[i]}{(k-1)}]$ 发送给 i,其中前一部分为转发数据,后一部分数据不需要进一步转发;节点 n_j ($q<j\leqslant m$) 将单位数据中的 $(\frac{1}{(k-1)}\sum_{p=N-k+1}^{N} U[p]+\sum_{p=1}^{j-1} c_p-\frac{U[i]}{(k-1)}, \frac{1}{(k-1)}\sum_{p=N-k+1}^{N} U[p]+\sum_{p=1}^{j} c_p-\frac{U[i]}{(k-1)}]$ 部分发送给节点 i,这部分数据也不需要进一步转发.这样节点 i 获得所有数据,而且数据无重复.

5 WFSOT 算法分析

本节中我们对 WFSOT 算法的复杂性和运行结果进行分析,以加深对 P2P 流媒体系统的理解.

5.1 WFSOT 算法的时间复杂性

步骤 2 最多需要 $O(N^2/2)$时间完成,循环步骤 3 ~ 步骤 7 是双重循环,最长所需时间为 $O(N^2/2)$,循环步骤 9 ~ 步骤 13 最多需要 $O(N^2/2)$的执行时间,故整个算法的执行时间为 $O(N^2)$.

5.2 WFSOT 算法的通信开销

WFSOT 主要通信开销发生在中心节点,中心节点获取 peer 节点可用带宽需要 $O(N)$的通信量,通知节点建立父子关系需要 $O(2dN)$的通信量(假设每个节点平均有 d 个父节点). d 的值较小,因此算法的通信开销为 $O(N)$.

5.3 拓扑结构中节点深度的分析

WFSOT 算法生成的节点拓扑结构可以转换为等效的数据流树,节点在数据流树中的深度表示节点接收到数据的最大延迟,即节点的播放延迟.下面通过具体的例子来说明节点深度的分布及其相关影响因素.在下面的实验中,假设 peer 节点的上行带宽在 5 ~ 20 之间均匀分布,使用二元组〈U_s, N〉表示一组实验.图 3 显示的是在〈200,2000〉情况下节点的深度分布图,从图中可以看出,68.5% 的节点深度在 7 ~ 11 之间,节点深度平均为 8.65.图 4 显示了节点的平均深度随节点数量的变化,节点的平均深度随着节点数量的增多而增大,但是增加的趋势放缓.

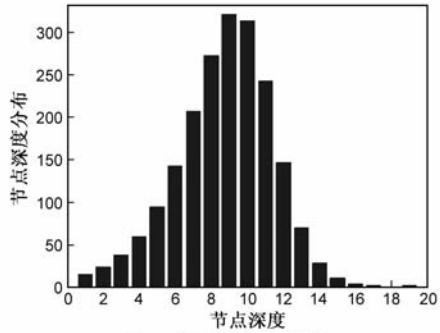


图3 节点深度分布图

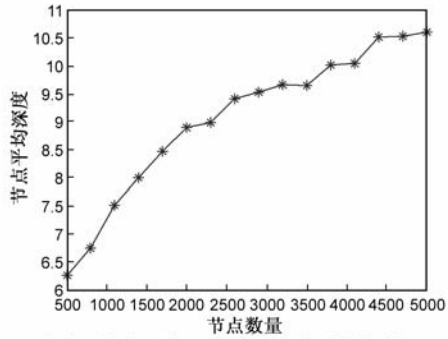


图4 节点平均深度随节点数量的变化

6 相关工作

R.Kumar、Y.Liu 等采用统计流理论对 P2P 流媒体系统进行建模[4].D.Qui、S.Srikant 对类 BT 系统建立了流模型[5],引入节点不稳定因素,将种子的数目作为流的数量,为系统建立了差分方程并求解系统稳定状态下的方程解.黄红兵等基于涌现视角给出一个有效的类 BitTorrent 系统分析模型[6].Kumar[7]针对具有有限的下载速率的带宽异构的文件共享流系统,推导出最小下载时间表达式.N.Maghar 在文献[8]中指出,在随机连接的 mesh 拓扑网络中为了最大化利用节点的可用带宽,需要满足所有连接具有相同带宽这一条件.D.Stutzbach 等[9]研究了 P2P 内容分发系统的可扩展性.

7 结论

本文的主要贡献在于以下四个方面:(1)提出了 P2P 流媒体系统模型,该模型使用较少的参数刻画了系统的属性;(2)推导出系统支持的最大数据传输速率,该速率可以用来作为参考值来衡量 P2P 流媒体系统的性能;(3)提出了集中式算法 WFSOT,该算法可以快速地计算出系统达到最大数据传输速率的节点拓扑结构及节点带宽分配,并且从理论上证明了算法的正确性.

参考文献

[1] Zhang X Y, Liu J C, Li B, Yum P. Cool streaming: A datadriven overlay network for peer-to-peer live media streaming[A]. Proc IEEE INFOCOM[C]. Miami, USA: IEEE Press, 2005. 2102 – 2111.

[2] Jin H, Deng D. Anysee: Mulitcast-based peer-to-peer media streaming service system[A]. Proc Asia Pacific Conference on Communications[C]. Australia: IEEE, 2005. 274 – 278.

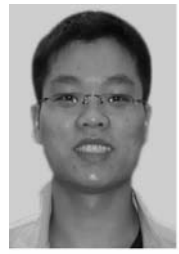
[3] Zhang M, Tang Y, Zhao L, Luo J. Gridmedia: A multi-sender based peer-to-peer multicast system for video streaming[A].

Proc IEEE International Conference on Multimedia and Expo [C]. Netherlands: IEEE Press, 2005. 614 – 617.
[4] Kumar R, Liu Y, Ross K. Stochastic fluid theory for P2P streaming systems[A]. Proc IEEE INFOCOM[C]. AnChorage, AK: IEEE Press, 2007. 919 – 927.
[5] Qiu D, Srikant S. Modeling and performance analysis of bit torrent-like Peer-to-Peer networks[A]. Proc ACM SIGCOMM [A]. Portland: ACM Press, 2004. 367 – 378.
[6] 黄红兵,任传俊,金士尧. 基于涌现视角的类 bit torrent 性能分析模型[J]. 电子学报,2010,32(2):307 – 313.
Huang Hong-bing, Ren Chuan-jun, Jin Shi-yao. A performance analysis model for BitTorrent-Like Peer-to-Peer systems based on the notion of emergence[J]. Acta Electronica Sinica, 2010, 32(2):307 – 313. (in Chinaese)
[7] Kumar R, Ross K W. Peer assisted file distribution: The minimum distribution time[A]. Proc HOTWEB[C]. Boston, USA: IEEE, 2006.
[8] Magharei M, Rejaie R. PRIME: P2P receiver drIven MEsh-based streaming[J]. IEEE/ACM Transactions on Networking, 2009, 17(4):1052 – 1065.

作者简介

吴国福　男,1980 年生于山东青岛,博士,主要研究方向为 P2P 计算.
E-mail: gfwu@nudt.edu.cn

窦　强　男,1973 年生于湖南长沙,研究员,主要研究方向为高性能计算、实时系统.



温　俊　男,1979 年生于安徽肥东,博士,主要研究方向为移动计算、无线传感器网络.

宋　磊　男,1976 年生于山东济南,博士研究生,主要研究方向为传感器网络、无线局域网安全.

窦文华　男,1946 年生于山西平定,教授,博士生导师,主要研究方向为高性能计算、高级计算机网络.