

一种高效的对等网络流媒体数据调度算法

任浩^{1,2,3}, 王劲林^{1,2}, 尤佳莉^{1,2}

(1. 中国科学院国家网络新媒体工程技术研究中心, 100190, 北京; 2. 中国科学院声学研究所, 100190, 北京; 3. 中国科学院研究生院, 100190, 北京)

摘要: 在基于无结构对等网络环境的 P2P 流媒体系统中, 针对现有数据调度算法不能充分利用节点带宽资源导致服务器负载较重的问题, 提出了一种基于数据块优先级的数据调度算法. 该算法根据邻居节点滑动窗口中数据需求信息和数据稀有性计算数据块优先级, 调度时优先请求高优先级的数据, 以提高节点间数据协作性. 当多个源节点可以同时提供所需数据时, 根据节点已上传下载数据量计算节点的贡献率, 向贡献率最低的源节点请求数据, 实现新加入节点上行带宽资源的快速利用. 仿真实验与实际系统测试表明, 该算法可以更充分地利用节点带宽资源, 较传统算法降低服务器负载 20% 以上, 提高了系统可扩展性.

关键词: 优先级; 数据调度; 贡献率; 流媒体; 对等网络

中图分类号: TP393 **文献标志码:** A **文章编号:** 0253-987X(2011)06-0020-07

An Efficient Data Scheduling Algorithm for Peer-to-Peer Streaming System

REN Hao^{1,2,3}, WANG Jinlin^{1,2}, YOU Jiali^{1,2}

(1. National Network New Media Engineering and Technology Research Center, Chinese Academy of Sciences, Beijing 100190, China; 2. Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China; 3. Graduate University of Chinese Academy of Sciences, Beijing 100190, China)

Abstract: A data-priority based scheduling algorithm is proposed to solve the problem that existing data scheduling algorithm cannot make full use of peers' bandwidth resources and causes heavy server load in unstructured overlay based peer-to-peer(P2P)streaming system. The algorithm calculates data block's priority based on data demand information and data scarcity of neighbors. Data blocks with high priority are requested first in scheduling. Moreover, when there are multiple source peers that can provide the desired data simultaneously, the peer with the lowest contribution ratio is selected to request data so that rapid use of newly joined peers' upstream bandwidth can be achieved. Peer's contribution ratio is calculated based on the amount of data that peer has already downloaded and uploaded. Simulation and experiments in real P2P streaming system show that the proposed algorithm can reduce server's load by 20 percent and improve system scalability effectively.

Keywords: priority; data scheduling; contribution ratio; streaming; peer-to-peer

随着宽带网络的迅速发展, 流媒体技术成为研究热点. 由于流媒体应用的数据量大、服务时间长, 传统 C/S 模式下服务器很容易成为系统瓶颈. 随着对等网络(P2P)的兴起, 由于其具有可扩展性高、部

收稿日期: 2011-01-04. 作者简介: 任浩(1983—), 男, 博士生; 王劲林(联系人), 男, 主任研究员, 博士生导师. 基金项目: 国家科技支撑计划资助项目(2008BAH28B04); 国家自然科学基金青年基金资助项目(60903218F0208); 国家高技术研究发展计划资助项目(2008AA01A317).

网络出版时间: 2011-03-24

网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20110324.1518.003.html>

署成本低等特点,使得基于 P2P 的流媒体技术得到了广泛推广^[1]. P2P 流媒体系统的重要特征是利用终端节点的上行带宽资源为用户提供流媒体服务,从而降低服务器的负载. 因此,如何通过合理的数据调度最大限度利用节点带宽资源是 P2P 流媒体研究中的关键问题之一.

针对现有调度算法大都以最大化系统吞吐率为目标^[2]而忽视如何尽量降低服务器负载的问题,本文提出以降低服务器负载为目标的数据调度(DPDS)算法. 该算法引入了数据块优先级的概念,将可提高节点间数据协作性的数据块赋予更高优先级并优先请求,同时根据邻居节点已上传数据量相对大小选择源节点,以更充分地利用节点带宽资源进行数据传输.

1 相关技术

当前流行的数据调度算法有如下 5 类:①随机调度^[3];②顺序调度^[4],也称最早截止期优先(NDF);③局部稀有优先^[5](LRF);④网络编码^[6](NC);⑤最小费用网络流^[7](MCNF).

随机调度是指所有数据块等概率获取,没有考虑节点间的协作性,降低服务器负载的效果不理想. NDF 算法是指每个节点按照数据块播放顺序请求数据,它能够提供较低的启动延迟,但会引发数据块共享率的逐步下降,无法充分利用节点带宽资源,导致服务器负载的加重. LRF 算法是一种被广泛使用的算法,它是一种“贪心”策略,主要思想是每个节点根据邻居节点的数据分布情况获取自己看来最稀少的数据,由于仅从节点自身角度出发获取数据,不利于节点间协作性的提高. 网络编码利用编码后的数据没有差别的特征来提高节点之间带宽的利用率,但算法复杂度较高. MCNF 算法利用网络流理论对系统吞吐率进行了优化,但没有对服务器负载进行优化,导致服务器负载较重.

文献^[8]从数据分发延迟的角度对服务器带宽消耗进行了分析,本文针对现有研究的不足从数据调度角度对最小化服务器带宽消耗进行建模,并提出 DPDS 算法. 实验结果表明,DPDS 算法在保证节点播放流畅度的前提下可以有效降低服务器负载.

2 服务器带宽消耗模型

2.1 前提假设

(1)设系统中共有 m 部等字节大小的影片,影片编号依次为 $1, 2, \dots, m$. 每部影片又被切割为 k 个

等字节大小的块,按照播放时间先后顺序对每个块依次编号为 $1, 2, \dots, k$. 于是,有如下数据分割矩阵

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1k} \\ f_{21} & f_{22} & \cdots & f_{2k} \\ \vdots & \vdots & & \vdots \\ f_{m1} & f_{m2} & \cdots & f_{mk} \end{bmatrix} \quad (1)$$

式中: f_{ij} 表示编号为 i ($1 \leq i \leq m$) 的影片中编号为 j ($1 \leq j \leq k$) 的数据块.

(2)节点周期性互相通告各自缓存中 f_{ij} 的存在信息,节点依据收到的信息以块为单位请求自己需要的数据,且每次至多请求 $R\tau/l$ 个数据块. 其中, R 为媒体流码率; τ 为调度周期; l 为每个块字节大小.

(3)每个节点维护一个最多可以存放 L 个数据块的滑动窗口用于请求需要的数据. 定义 $L = nR\tau/l$, 其中 n 为大于 1 的正整数. 滑动窗口以 R/l 的速率向节点播放时间增大的方向滑动.

(4)为保证节点播放的连续性,将滑动窗口分为紧急区和非紧急区. 紧急区为节点即将播放的数据;紧急区之后就是非紧急区,包括节点自身播放需要但播放时间并不紧急的数据. 进入紧急区中缺失的数据将采用顺序调度的方式优先请求,以保证节点播放的流畅度;非紧急区为节点与合作节点的数据交换区域,采用本文提出的 DPDS 算法进行调度. 节点滑动窗口如图 1 所示.



图 1 滑动窗口示意图

(5)在服务器和邻居节点同时拥有自己需要的数据时,优先向邻居节点请求.

2.2 模型表示

设决策变量 x_{is}^k 表示节点 i 是否从服务器请求编号为 k 的数据块, 1 为请求, 0 为不请求. 数据调度目标是 minimized 服务器带宽开销, 即

$$\min \sum_{i \in N} \sum_{k \in D_i} w_s^k x_{is}^k \quad (2a)$$

$$\text{s. t. } \sum_{j \in R_i^k} x_{ij}^k + x_{is}^k \leq 1, \forall i \in N, k \in D_i \quad (2b)$$

$$\sum_{i \in N} \sum_{k \in D_i} x_{is}^k l \leq \tau O_s$$

$$x_{is}^k \in \{0, 1\}, \forall k \in D_i, i \in N \quad (2c)$$

式(2a)保证每个数据块 k 仅从至多一个节点获得;式(2b)表示所有节点从服务器请求的数据量不能超过服务器上行带宽限制;式(2c)表明最小化服务器带宽消耗的数据调度是一个 0-1 规划问题. 由于优先向普通邻居节点请求数据,因而增加节点间数据互补性就可以有效降低服务器负载. 具体模型参数见表 1.

表 1 服务器带宽消耗模型参数

符号	意义
N	所有接收节点的集合,不包括数据源服务器,即 $N=\{1, \dots, N \}$
τ	数据调度周期
D_i	请求节点 i 在 $t_c \sim t_c + \tau$ 时间段滑动窗口内需要请求的数据块集合
x_{ij}^k	节点 i 是否从节点 j 请求数据块 $k(k \in D_i)$, 1 为请求, 0 为不请求
R_i^k	所有拥有节点 i 所需数据块 $k(k \in D_i)$ 的邻居节点集合 (不包括服务器)
O_s	服务器的上行带宽
w_s^k	从服务器获得数据块 $k(k \in D_i)$ 的服务器带宽开销
l	数据块大小

3 DPDS 算法

3.1 数据块优先级设定

DPDS 算法中各参数的意义见表 2. 优先级高的数据块将被优先请求,定义数据块 S_i^j 优先级如下所示

$$P_{ri}(S_i^j) = \frac{(\sum_{k=1}^m b_{ij}^k \overline{g_{ij}^k})^2}{F(S_i^j)}$$

(3)

其中幂数项表示同等条件下 $\sum_{k=1}^m b_{ij}^k \overline{g_{ij}^k}$ 占更大比重.

$F(S_i^j)$ 越小表示该数据块在网络中的备份数越少,数据越稀有,为了加速数据在网络中的扩散需要优先请求,因而被请求优先级越高. $\sum_{k=1}^m b_{ij}^k \overline{g_{ij}^k}$ 越大表示越多邻居节点也将请求该数据块,由于同等条件下优先向邻居节点请求数据,因而拥有越多其他节点需要的数据就可以提高节点间数据协作性,从而达到更充分利用普通节点上行带宽资源降低服务器负载的目的. 这样,可以克服传统调度算法由于仅考虑请求节点自身播放流畅性而忽视节点间数据协作性的缺点.

表 2 DPDS 算法参数

符号	意义
S_i^j	节点 i 滑动窗口中编号为 j 的数据块
m	节点 i 的邻居节点个数
b_{ij}^k	S_i^j 是否在邻居节点 k 的滑动窗口范围内, 1 为在, 0 为不在
$F(S_i^j)$	邻居节点滑动窗口中拥有 S_i^j 的邻居节点个数
g_{ij}^k	S_i^j 是否在邻居节点 k 的滑动窗口中存在, 1 为存在, 0 为不存在
$U_i(t)$	节点 i 在 t 时刻上传到 P2P 网络中的数据量
$D_i(t)$	节点 i 在 t 时刻从 P2P 网络中下载的数据量
$D_{ij}(t)$	t 时刻节点 j 从节点 i 获取的数据量

3.2 节点贡献率

节点 i 在 t 时刻的数据贡献数量为

$$C(i, t) = \int_0^t (U_i(t) - D_i(t)) dt$$

(4)

由于节点 i 上传量也等于 P2P 网络中其他节点从节点 i 的下载量,即

$$U_i(t) = \sum_{j=1}^N D_{ij}(t)$$

(5)

将式(5)代入式(4),得

$$C(i, t) = \int_0^t \left(\sum_{j=1}^N D_{ij}(t) - D_i(t) \right) dt$$

(6)

进而定义节点 i 在 t 时刻的贡献率为

$$E_i(t) = \frac{\int_0^t \left(\sum_{j=1}^N D_{ij}(t) - D_i(t) \right) dt}{\sum_{k=1}^N \int_0^t \left(\sum_{j=1}^N D_{kj}(t) - D_k(t) \right) dt}$$

(7)

$E_i(t)$ 作为源节点的选择依据, $E_i(t)$ 越小,越优先被选为提供数据的源节点. 新加入系统的节点由于没有上传或上传量相对很小,因而贡献率很小,通过将其优先选入传输会话贡献数据可以提高这部分节点的带宽资源利用,有利于服务器负载的降低.

3.3 算法流程与实例分析

DPDS 算法由每个节点分布的独立周期性执行. 算法的思想是节点首先计算滑动窗口中所有缺失数据块的优先级,然后将优先级最高的前 $R\tau/l$ 个数据块请求依次分配给拥有对应数据块的当前贡献率最低的邻居节点. 设当前节点为 i ,调度执行时刻为 t ,算法流程如图 2 所示.

在图 3 的调度场景中,请求节点 D 有 3 个普通邻居节点 A, B, C ,每个邻居节点的滑动窗口状态标识在节点旁边. 设 $R\tau/l=5, n=2$,则 $L=10$. 阴影部

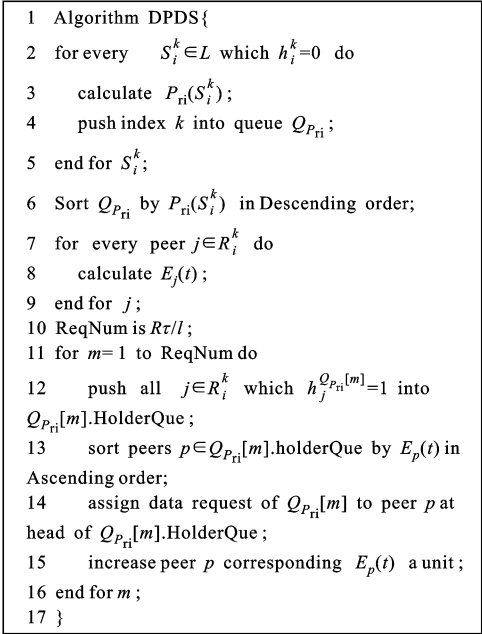


图 2 DPDS 算法流程

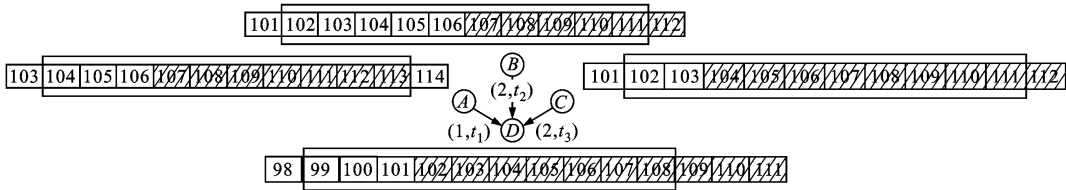


图 3 DPDS 算法的调度场景

表 3 NDF 算法的调度结果

数据编号	102	103	104
源节点	B	B	A

表 4 LRF 算法的调度结果

数据编号	102	103	104	105
源节点	B	C	A	B

表 5 DPDS 算法的调度结果

数据编号	104	105	106	102	103
源节点	B	B	A	C	C

3.4 算法时间复杂度分析

设邻居节点个数为 m . 算法流程中第 2~5 步复杂度为 $O(L)$, 第 6 步采用快速排序, 复杂度为 $O(L)+O(L\lg L)$, 第 7~9 步复杂度为 $O(m)$, 第 10

分表示节点滑动窗口中缺失的数据块.

节点间连线上的数字对的第 1 项表示邻居节点可以在 1 个调度周期内上载的最大数据块个数; 第 2 项表示节点已在线时长. 设定 $t_3 < t_2 < t_1$, 且 D 滑动窗口紧急区的数据已经填满, 此处只讨论如何利用 DPDS 算法调度非紧急区的数据.

根据前提假设, 节点 D 本次调度将请求 102~108 中的至多 5 个数据块. 根据式(3)数据优先级模型会依次请求 104、105、106、102、103, 再根据式(7)得到数据请求分配结果.

表 3~表 5 反映了 3 种算法在一个调度周期内数据块请求的顺序及对应请求源节点的分配结果. NDF 与 LRF 算法都优先向更长在线时间和更高带宽的节点请求数据.

从表 5 可以看出, DPDS 算法在一个调度周期内的节点带宽利用率至少提高了 20%, 有利于服务器负载的降低.

步复杂度为 $O(1)$, 第 12 步复杂度为 $O(m)$, 第 13 步同样采用快序排序, 复杂度为 $O(m)+O(m\lg m)$, 第 14、第 15 步复杂度为 $O(1)$. 因此, DPDS 算法总时间复杂度为

$$T = O(L) + O(L) + O(L\lg L) + O(m) + O(1) + O(L(2m + m\lg m + 1)) \tag{8}$$

一般情况下 $m < L$, 所以式(8)可简化为

$$T = O(Lm\lg m) \tag{9}$$

4 算法理论分析

NDF 算法仅考虑节点自身播放的连续性而优先请求距离播放点更近的数据, 导致所有节点都只缓存播放时间较早即索引编号较小的数据, 节点间数据缓存情况趋同, 降低了节点间的协作性, 不利于节点带宽资源的充分利用. Random 算法是无状态的调度策略, 也不利于节点带宽利用率的提高. 鉴于目前 LRF 算法的广泛应用, 本文重点说明 DPDS 算法相比 LRF 算法可以提高节点带宽资源利用率.

令 $U_i(S)$ 表示节点 i 的邻居节点中数据块 S 的数目, $\sigma(U_i(S))$ 为 $U_i(S)$ 的方差. 由于采用每个节点独立周期性的执行调度的方法, 因而由极大似然法则 i 具有一般性, 不妨令 $\sigma(U_i(S)) = \sigma(U)$. 文献[9]指出, $\sigma(U)$ 越小越有利于数据分发, 因此有如下命题

$$\sigma(U)_{\text{DPDS}} \leq \sigma(U)_{\text{LRF}} \quad (10)$$

证明: 假设共有 P 个节点合作从一个数据提供者请求数据, 设被请求的数据块总数为 Q , 每个节点最多能从同一个邻居节点请求 d 个数据块. 设 $Q = CQ' + \omega_1$, $P = CP' + \omega_2$. 所有 Q 个数据块被分为 C 个组, $C - \omega_1$ 个组每组含有 Q' 个块, ω_1 个分组每组含有 $Q' + 1$ 个块. 所有 P 个节点被分为 C 个组, $C - \omega_2$ 个组每组含有 P' 个节点, ω_2 个组每组含有 $P' + 1$ 个节点. P' 个节点分配到总数为 Q' 的组中, 令 (P', Q') 表示从该组中请求 d 个数据, $p(P', Q')$ 是事件 (P', Q') 的概率. 令 V_1 表示 (P', Q') 发生时数据片段 S 被请求次数为 i 的事件, $P(V_1)$ 是 V_1 的概率. 依此类推有

$$\left. \begin{aligned} P(V_1) &= P(V = i \mid (P', Q')) = \left[\begin{matrix} P' \\ i \end{matrix} \right] \left(\frac{d}{Q'} \right)^i \left(1 - \frac{d}{Q'} \right)^{P'-i} \\ P(V_2) &= P(V = i \mid (P' + 1, Q')) = \left[\begin{matrix} P' + 1 \\ i \end{matrix} \right] \left(\frac{d}{Q'} \right)^i \left(1 - \frac{d}{Q'} \right)^{P'+1-i} \\ P(V_3) &= P(V = i \mid (P', Q' + 1)) = \left[\begin{matrix} P' \\ i \end{matrix} \right] \left(\frac{d}{Q' + 1} \right)^i \left(1 - \frac{d}{Q' + 1} \right)^{P'-i} \\ P(V_4) &= P(V = i \mid (P' + 1, Q' + 1)) = \left[\begin{matrix} P' + 1 \\ i \end{matrix} \right] \left(\frac{d}{Q' + 1} \right)^i \left(1 - \frac{d}{Q' + 1} \right)^{P'+1-i} \end{aligned} \right\} \quad (11)$$

于是

$$\begin{aligned} \text{var}(U) &= E(U - EU)^2 = \\ &= \sum_{k=1}^4 \sum_{i=0}^{P'} (i - E(V_k))^2 P(V_k) \end{aligned} \quad (12)$$

式中

$$E(V_1) = \frac{P'd}{Q'}; \quad E(V_2) = \frac{(P' + 1)d}{Q'}$$

$$E(V_3) = \frac{P'd}{Q' + 1}; \quad E(V_4) = \frac{(P' + 1)d}{Q' + 1}$$

DPDS 算法优先请求邻居节点滑动窗口中也缺少的数据, 于是 $\sum_{k=1}^4 P_{\text{DPDS}}(V_k) < \sum_{k=1}^4 P_{\text{LRF}}(V_k)$, 由式 (12) 命题得证.

5 实验与分析

5.1 仿真环境设置及参数定义

利用 GT-ITM 生成 Transit-Stub 结构的双层物理网络拓扑, 分别代表核心路由层和边缘路由层. 核心路由层由 3 个核心域组成, 每个域包含 5 个核心路由节点; 边缘路由层由 15 个边缘域组成, 每个域平均包含 7 个边缘路由节点. 每个边缘域各附属 1 个核心路由节点, 每个边缘路由节点代表一个网络区域. 设定核心路由节点之间以及核心路由节点与边缘路由节点之间的带宽为 1 Gb/s, 边缘路由节点之间的带宽为 100 Mb/s, 路由路径选择由最短路径 Dijkstra 算法决定.

数据源服务器连接在一个固定的核心路由节点上, 上行接入带宽为 10 Mb/s, 媒体流码率为 280 kb/s, 数据块大小为 1 kB, 数据调度周期为 4 s. 仿真总节点数为 1 000 个, 终端节点以强度 1/8 的 Poisson 分布加入系统并随机连接在某个边缘路由节点上, 在线时间服从 $\lambda = 600$, $k = 2$ 的 Weibull 分布. 仿真运行 24 h, 所有实验结果均为运行 5 次的平均值.

定义系统上行带宽提供率 (U_{BSR}) 为系统中所有终端节点 (不包括服务器) 上行带宽之和与最小总带宽需求的比值, 最小总带宽需求为媒体流码率与系统中总节点数的乘积. 为了模拟节点异构性, 根据实际因特网测量结果^[10], 将用户节点上行接入带宽聚类为 3 类, 并规定各自所占比例, 如表 6 所示.

表 6 节点接入带宽及比例

U_{BSR}	不同上行/下行接入带宽节点占总节点的比例/%		
	1 Mb · s ⁻¹ 3 Mb · s ⁻¹	384 kb · s ⁻¹ 1.5 Mb · s ⁻¹	128 kb · s ⁻¹ 768 kb · s ⁻¹
1.1	10	45	45
1.15	10	50	40
1.2	15	39	46
1.25	16	40.5	43.5
1.3	18	40	42
1.35	20	40	40
1.4	20	45	35
1.5	25	41	34
1.7	35	30	35

5.2 系统平均吞吐率

定义系统平均吞吐率为各节点在数据播放截止

时间前收到的有效数据块的数量,该参数越大表明播放越流畅,最大值为媒体流码率。

图 4 给出了不同 U_{BSR} 下不同调度算法的吞吐率比较。可以看到,随着 U_{BSR} 的提高,DPDS 算法的吞吐率不断提高,由于 DPDS 算法能够充分利用节点带宽资源,随着系统中节点带宽能力的增大,可以使节点在播放截止时间前不能成功请求到数据的概率降低,从而使吞吐率得到改善。当 $U_{BSR} > 1.35$ 时,DPDS 算法的系统吞吐率已接近 MCNF 算法的最优吞吐率,即媒体流码率,表示系统中节点可以基本流畅播放视频。

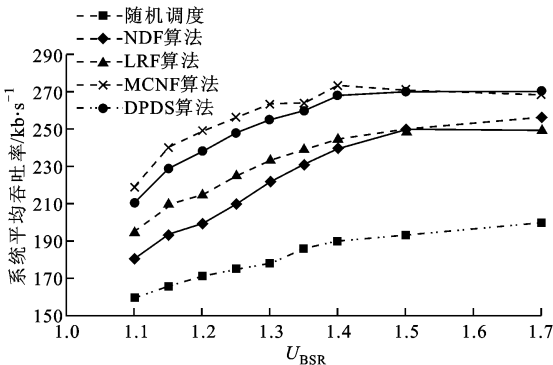


图 4 系统平均吞吐率比较

5.3 系统平均播放启动延迟

在流媒体系统中为了避免由于网络抖动造成播放中断,每个节点会维护一个播放缓冲区,当播放缓冲区中数据达到一定数量时才启动视频播放。定义播放启动延迟为节点加入系统到开始播放视频的时间间隔,显然播放延迟越小用户体验越好。仿真中设节点缓存 3 Mb 的数据后开始播放。

图 5 给出了不同调度算法的播放延迟比较。可以看到, $U_{BSR} = 1.3$ 时,DPDS 算法的播放启动延迟仅次于 NDF 算法,与 NDF 算法以牺牲服务器负载为代价降低播放启动延迟不同,DPDS 算法更注重通过提高节点带宽资源利用率来降低服务器负载。当 $U_{BSR} = 1.7$ 时,由于系统带宽能力的增强,DPDS 算法通过充分利用节点上传带宽资源,使节点能够在更短时间内获取播放需要的数据量,有效缩短播放延迟,同时不增加服务器的负载。由图 5 可以看到,90% 的节点播放延迟在 20 s 以内,可以提供更好的用户体验。

5.4 控制开销

P2P 流媒体系统中除了传输数据外,还需要传输缓存映像(BM)、数据请求、连接建立与维护等控制报文。

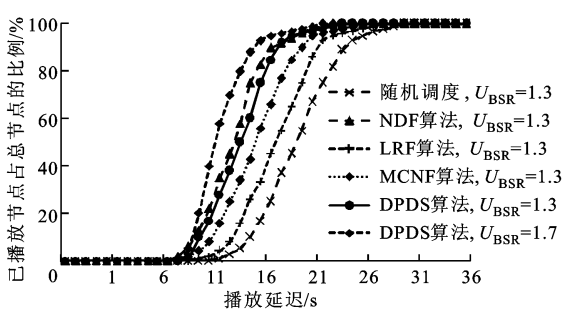


图 5 系统平均播放延迟累计分布

定义控制报文比率为控制报文传输速率与媒体流码率的比值。该参数用来衡量系统控制开销,显然控制报文比率越低系统可扩展性越高。图 6 给出了不同 U_{BSR} 下基于不同调度算法的控制开销比较。可以看到,当 $U_{BSR} > 1.3$ 时,DPDS 算法的控制开销低于其他算法。这是因为当系统中有更多高接入带宽节点时,由于节点带宽资源被充分利用因而数据请求失败概率低于传统算法,进而减少了数据重传请求报文数量,从而有效降低了系统控制开销。

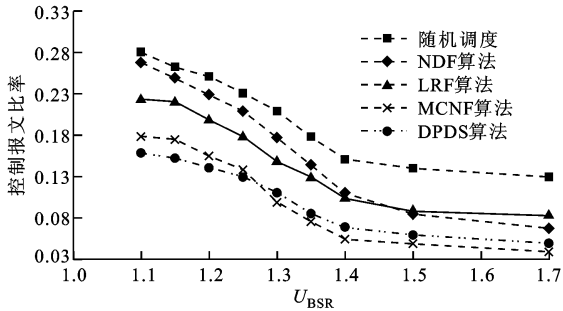


图 6 不同算法的控制开销比较

5.5 服务器负载

NOVA^[11]是由国家网络新媒体工程技术研究中心研发的 P2P 流媒体系统,可以同时支持直播和点播两种业务。实验中系统运行在 LAN 网络环境,节点上行物理带宽为 1 Mb/s,为实现节点异构性,利用 Bandwidth Controller Manager 带宽限制软件根据表 6 对节点可用上行带宽进行设置。运行节点总数为 50,其他参数与仿真一致。

定义归一化服务器带宽消耗(NCSB)为服务器实际出流量速率与最小服务器带宽需求(MSBN)的比值。由于仿真中 $U_{BSR} > 1$,故 MSBN 等于媒体流码率,即理想情况下媒体服务器只需要为每一个数据块推送一份拷贝到 P2P 网络中。从以上定义可以看到,NCSB 总是大于 1,且越接近 1 表示服务器负载越小,系统性能越好。

图7给出了不同调度算法的NCSB比较. 可以看到, DPDS算法服务器负载比LRF算法降低30%以上, 比MCNF算法也平均降低20%以上. 由于DPDS算法在确定数据请求顺序时除了考虑数据的稀有性和紧急性外, 还考虑数据被其他节点请求的可能性, 提高了节点间数据协作性, 因而有效降低了服务器负载. 同时, DPDS算法在选择请求源节点时优先选择已上传数据少的节点, 可以使刚加入系统的节点更多贡献上行带宽资源, 提高节点带宽资源利用率, 也有利于服务器负载的降低.

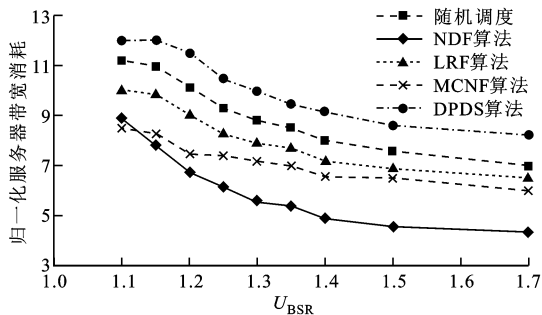


图7 不同算法的服务器负载比较

6 结 论

在研究P2P流媒体系统中节点带宽资源利用的问题时, 将服务器带宽消耗建模为最优化问题, 在此基础上引入数据优先级模型和节点贡献率模型, 提出一种基于数据优先级的调度算法. 通过优先请求邻居节点滑动窗口中也缺失的数据, DPDS算法可以有效提高节点间数据协作性, 以尽量降低服务器带宽消耗. DPDS算法在选择请求源节点时优先选择已上传数据量小的节点, 以充分利用刚加入系统节点的带宽资源. 仿真实验和实际系统运行表明, DPDS算法较传统算法可以有效降低服务器负载, 优化系统可扩展性. 在下一步的工作中, 将对DPDS算法在节点上行带宽受限时的性能进行优化, 进一步提高系统平均吞吐率.

参考文献:

- [1] 王锐, 钱德沛, 朱青林, 等. 一种高效的冗余编码 Mesh 流媒体覆盖网[J]. 西安交通大学学报, 2009, 43(10):56-60.
WANG Rui, QIAN Depei, ZHU Qinglin, et al. An

effective redundant coded mesh overlay streaming network[J]. Journal of Xi'an Jiaotong University, 2009, 43(10):56-60.

- [2] LI Jiaming, YEO Chai Kiat. Dynamic transmission scheduling for streaming applications via P2P overlay [J]. Journal of Signal Processing Systems for Signal Image and Video Technology, 2010, 59(3):355-366.
- [3] PAI V, KUMAR K, TAMILMANI K, et al. Chain-saw: eliminating trees from overlay multicast [J]. Lecture Notes in Computer Science, 2005, 36(3640): 127-140.
- [4] ZHOU Yiping, CHIU Dahming, LUI J C S. A simple model for analyzing P2P streaming protocols [C] // Proceedings of the IEEE International Conference on Network Protocols. Piscataway, NJ, USA: IEEE, 2007:226-235.
- [5] ZHENG Chen, YAN Jinyan. Optimizing resource scheduling in BitTorrent file distribution network [C] // Proceedings of the IEEE International Conference on Management and Service Science. Piscataway, NJ, USA: IEEE, 2009:1-4.
- [6] RAMASUBRAMONIAN A K, WOODS J W. Multiple description coding and practical network coding for video multicast [J]. Signal Processing Letters, 2010, 17(3):265-268.
- [7] ZHANG Meng, XIONG Yongqiang, QIAN Qian, et al. Optimizing the throughput of data-driven peer-to-peer streaming [J]. IEEE Transactions on Parallel and Distributed Systems, 2009, 20(1):97-110.
- [8] LU Yifeng, REN Hao, WANG Jinlin. Real-time performance vs. server bandwidth cost in peer-to-peer streaming system [C] // Proceedings of the International Conference on Computer and Electrical Engineering. Piscataway, NJ, USA: IEEE, 2008: 286-290.
- [9] BYERS J W, CONSIDINE J, MITZENMACHER M, et al. Informed content delivery across adaptive overlay networks [J]. IEEE Transactions on Networking, 2004, 12(5):767-780.
- [10] CIULLO D, GARCIA M A, HORVATH A, et al. Network awareness of P2P live streaming applications: a measurement study [J]. IEEE Transactions on Multimedia, 2010, 12(1):54-63.
- [11] LI X. NOVA[EB/OL]. [2010-11-30]. <http://dspnova.8800.org/>.

(编辑 杜秀杰 武红江)