基于 P2P 的多发送节点流媒体数据分派策略

Research on multi – sender data dispatching scheme based on P2P

张修如 阳天保 谭遥骋 朱光辉 （中南大学信息科学与工程学院 湖南 长沙 410075）

摘要：为解决 P2P 流媒体播放系统中多发送节点对一个节点同时提供服务的数据调度问题，提出了一种考虑网络变化、抖动、调度失败等情况下的动态数据分派策略。本文设计了基于 P2P 和 C/S 的混合流媒体系统三层结构，建立了数据分派的数学模型，并给出了一个具有最小缓冲延迟的动态数据分配算法。

关键词：P2P 流媒体 数据分派

1 引言

P2P 流媒体系统中数据调度问题在近年来得到许多研究者的关注。文献[1]研究了一个叫做 OTS_{p2p} 的媒体数据分配算法，在该算法中，假定媒体数据块为相同大小，请求节点 P_r 对集合 $\{P_s^1, P_s^2, \cdots, P_s^m\}$ 中的 m 个供应节点按照一定策略分配带宽，再根据给定的带宽大小分配媒体数据块。OTS_{p2p} 将计算最优媒体数据分配，它使得流会话中接收节点的缓冲延迟很小，该文证明了这个最小缓冲延迟是 $m\delta t$，其中 m 表示提供服务的节点数目，δt 表示单位时间。文献[2]提出了一个叫做固定长度槽调度方案（fixed – length slotted scheduling FSS）。与 OTS_{p2p} 不同，FSS 采用可变长度数据段，其大小视每个段传输时的带宽而决定。数据顺序分配，延迟主要依赖供应节点的最高带宽，但可通过扩大或缩小来槽长来使延迟成比例地扩大或缩小，在一个槽长内数据连续分配，发送简单。

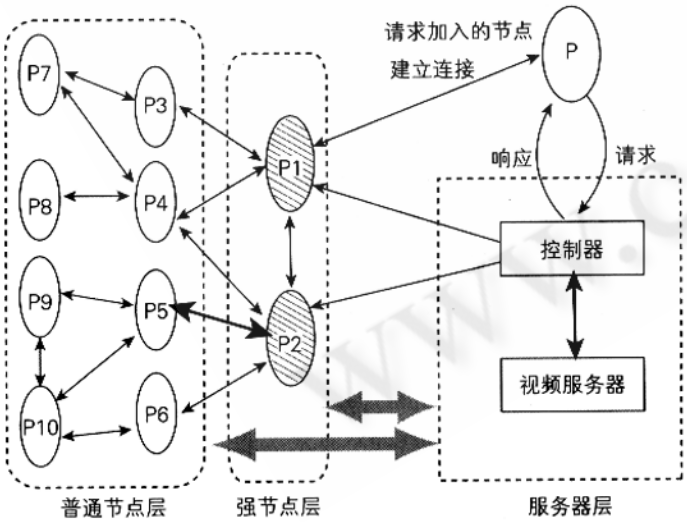

图 1 系统结构

本文设计和描述了一个基于 P2P 和 C/S 混合结构下多发送节点流媒体数据分配策略，并给出和比较了相关调度算法性能。

2 系统结构

2.1 系统结构图

系统采用 P2P 和 C/S 混合的三层结构模式，局部系统结构如图 1 所示，分别为服务器层，强结点（Super_peer）层，普通节点层。沿用簇的概念，整个系统将有 n 个簇组成，各角色功能描述如下：

控制器：负责创建强节点，并维护强节点列表信息，同时向它们发送整个视频文件流；作为每个节点加入系统的访问接入点，当收到节点的请求信息时，为每个请求节点创建唯一的节点号，同时返回合适的强节点信息列表。

视频服务器：存放视频文件，提供流媒体节目，如各种直播节目、电视节目、电影节目等，负责将解析文件数据包在网络中传输。

强节点：强节点的本质上是 P2P 系统中的普通节点，但是它又比普通节点性能好，有较大的网络带宽和负载。如图 1 中节点 P1、P2 均为每个簇的簇首节点。簇首节点响应节点加入请求，并维护本簇中节点信息列表。当簇内普通节点的媒体数据请求得不到满足时，簇首节点负责返回满足其要求的其他簇中节点信息。

普通节点：完成数据的调度与传输，缓冲区的管理，视频的播放，可在强节点的支持下向外簇或者服务器发出资源请求。

2.2 系统工作原理

当节点 P 提出媒体播放请求，控制服务器为 P 节点分配唯一的节点号，并向其返回满足该请求的强节点列表，请求节点与某个强节点建立连接，向强节点发送自身节点信息，并获取簇内节点信息列表，选择合适的服务节点进行流会话，这些服务节点提供的出口带宽能够满足该请求节点的播放需求。当节点的需求不能得到满足时，可以向服务器直接请求视频资源。底层通信协议，控制流采取 TCP 协议，数据流采取 UDP 协议，充分满足应用层的实时性需求。

3 数学模型

对于播放时长为 T 的一段节目，以节目的最小存储和传输单元为数据块，每块的播放时长为一个单位时间 δt，所有的数据块可以按照先后顺序从 1 到 T 编号。

设系统 D 中，p_i 代表节点，s_i 代表媒体资源数据片断，$P=\{p_i \mid 1\leq i\leq n\}$ 为节点集合，$S=\{s_i \mid 1\leq i\leq T\}$ 为流媒体资源数据片断集合，则产生一个系统 $D=(P, S)$。

节点 Pr 为请求节点，集合 $Ps=\{p_s^i \mid 1\leq i\leq n\}$ 为可提供媒体流给 Pr 的服务节点集合，R_0 为所请求节目的播放速率，R_{in} 为请求节点 Pr 的下载速率，并假定 $R_{in}\geq R_0$，集合 $Rs\{r(p_s^i) \mid 1\leq i\leq n\}$ 为集合 Ps 中相对应节点传输速率的组成的集合。

定义变量 $r_s^i=r(p_s^i)$，并对 r_s^i 进行降序排序得到 $Rs'\{r_s^i \mid 1\leq j\leq n, r_s^1\geq r_s^2\geq\cdots\cdots\geq r_s^n\}$，则对应 Rs' 的有序节点集合可以表示为 $Ps'\{p_s^i \mid 1\leq j\leq n, r_s^i\in Rs'\}$，并按传输速率将服务节点集合 Ps 分为两组集合 $P_s^{active}=\{P_s^i \mid 1\leq i\leq k\}$ 和 $P_s^{standby}=\{P_s^i \mid k+1\leq i\leq n\}$，其中 P_s^{active} 为当前数据服务节点集合，集合 $P_s^{standby}$ 为后备的服务节点集合，集合 P_s^{active} 的元素个数 k 满足 $r_s^1+r_s^2+\cdots\cdots+r_s^k\geq R_0$。当 $r_s^1+r_s^2+\cdots\cdots+r_s^k<R_0$ 时，集合 $P_s^{standby}$ 中的节点就有机会进入集合 P_s^{active} 中成为当前服务节点。

现在对节点 p_s^i 传输块 s_i 的过程进行描述：设 T_0 为启动传输的时刻，$t(p_s^i, s_i)$ 表示传输所需时间，$T(p_s^i, s_i)$ 为块 s_i 传输完成时刻，$T(p_s^i)$ 表示当前状态下 p_s^i 完成所有已经分配任务的时刻。$len(s_i)$ 表示块长度，则应有：

$$t(p_s^i, s_i)=len(s_i)/r(p_s^i)$$

$$T(p_s^i, s_i)=T(p_s^i)+t(p_s^i, s_i)$$

假定媒体数据块的大小是相同的，那么 $len(s_i)$ 取常量值，设为 L，则影响 $T(p_s^i, s_i)$ 的主要因素在于 $r(P_s^i)$，即服务节点 i 的有效上载带宽。

设调度周期一次分配有 k 个段，记 δ 为延迟（这里先设所有时间都以段的大小 δt 为单位），则第 j 段的实际播放时间是 $\delta+j-1$，$T(p_s^i, s_i)$ 为第 j 段的到达时间，则第 j 段允许播放的最早时间为 $T(p_s^i, s_i)-1$（假设一个段可边接收边播放），所以 $\delta+j-1\geq T(p_s^i, s_i)-1$，$j=1,2\ldots,k$。至此可知，延迟可表示为：

$$\delta=\max\{T(p_s^i, s_i)-j \mid i=1,2\cdots\cdots,n\ j=1,2\cdots\cdots,k\}$$，单位为 δt

4 调度策略

4.1 调度原则

在实时流媒体传输过程中，对任何 Pr 接收节点而言，均不能够预先准确地知道与其他 Peer 服务节点之间的网络带宽情况，并且这种带宽处于动态变化中。而为了减少网络上的拥塞以及为了与其它 TCP 流平等地共享网络带宽资源，通常需要在接收节点或发送节点上实施某种 TCP 友好的拥塞控制机制如 TFRC[3] 或 RAP[4]。

另外，网络传输路径上一般具有丢包、延迟、抖动等特性，从而会影响数据包在网络上的传输；对于从 Pr 接收节点到不同 Peer 服务节点之间的网络路径而言，它们在丢包率、往返延迟等方面又呈现动态异构的网络特性。因此，媒体数据调度策略必须要适应这种动态异构的网络特征，以平滑接收端的播放质量。

本文第 2 节中设计的 P2P 与 C/S 相结合的结构的优点主要在于：中央服务器不仅担当索引服务器的角色，还提供媒体源，既方便媒体资源管理又较好的融合了传统的 C/S 模式。当观看同个媒体节目的节点数越多，网络的节点存储的冗余数据也越多，节点之间就能够互相提供数据传输服务的能力就越强。基于以上的考虑，我们要求在调度分配数据时遵循如下原则：(1) 使得接收节点缓冲延迟最小；(2) 优先考虑向对等节点请求资源，减少服务器负载。(3) 考虑对等节点间

网络动态异构性特征，提高接收端播放质量。

4.2 带宽测量及带宽自适应调整

由本文第 3 节模型中可知，影响 T(psi, sj) 的主要因素在于 r(Psi)，即服务节点 i 的有效上载带宽。因此，若能够测量或者动态调整对等节点的有效网络带宽，则对于数据调度是十分有利的。

(1) 带宽测量。国内一个研究小组提出的随机发送单个小探测报文[5]的方法来实现端到端的带宽测量，这种方法基于蒙特卡洛随机抽样的思想，其测量范围不受源节点最大发送速率的限制。该方法不仅可以计算整条路径的可用带宽，也可以计算各段链路的容量和空闲率，进而分析得到各路由节点上的流量变化，以及各链路上对应的不同类型的背景流的分布。这种方法，对于我们进行动态测量网络带宽、确定调度分配有很大帮助。

(2) 带宽动态调整。我们考虑在上述带宽测量的基础上动态调整带宽，以便于适应流媒体实时传输调度的需要。给出“调度带宽”定义：Pr 节点调度数据时为多个服务节点假定的一个带宽值，而以这个“调度带宽”分配调度数据，从数据提供者获取数据的带宽应该大于视频的码率，以保证视频的正常播放。计算每个服务节点“调度带宽”的初始值：

初始调度带宽 = $r(P_s^i)$ 的初始值

其中 $r(P_s^i)$ 代表服务节点 i 的上载带宽，采用带宽测量方法获得初始数据。

我们给定动态调整策略：在每次调度周期内，增加调度带宽时，可以 Δr 为增幅提高调度带宽；降低调度带宽时，可根据前面调度周期返回的数据量计算出的调度带宽值作为下一调度周期内的调度带宽。假定服务节点提供的调度带宽都是 R_0/N 的整数倍，其中 N 是自然数，作为系统参数根据应用的实际情况适当选取，R_0 为所请求节目的播放速率，则可以取 $\Delta r = R_0/N$，这样系统中多个节点可以维持一个动态平衡。

4.3 调度算法

基于上述模型及描述，我们给出相应调度算法，称为 MBDS 算法(Multi – sender Minmum Buffering Delay Scheduling)。

输入：需要调度的数据集合 $S=\{s_i \mid 1\leqslant i\leqslant T\}$；当前数据服务节点集合 $P_s^{active}=\{P_s^i \mid 1\leqslant i\leqslant m\}$。

输出：一次分配映射，将 S 中数据块单射到 P_s^{active} 中，使得接收节点缓冲延迟最小。

算法描述：

(1) 定义数组 r[m]；

(2) 输入当前服务节点集 $P_s^{active}=\{P_s^i \mid 1\leqslant i\leqslant m\}$ 各节点的输出带宽到 r[1]，……r[m]；

(3) 按照带宽调整策略动态更新 r[m] 值；

(4) 从集合 S 取出一个未分配数据块 s_i，对所有服务节点 i，计算完成传输这个数据块的时刻

$T(p_s^i, s_i) = T(p_s^i) + t(p_s^i, s_i)$ 其中，$t(p_s^i) = L/r[i]$，L 为一个数据块长度；

(5) 计算 $T_{min} = Min\{T(p_s^i, s_i) \mid i=1,2\cdots\cdots m\}$，并记 min 为拥有最小 T_{min} 的服务节点下标值，初始值 $T_{min} = T(p_s^1, s_i)$，min = 1；

(6) 将块 s_i 分配给节点 P_s^{min}；

(7) 若 S 中还有未分配的调度数据块，则进入步骤 4 和 5，重复执行 4 ~7，直到集合 S 中的调度数据分配完毕。

4.4 调度失败恢复策略

在流媒体实时传输服务调度过程中，不能保证服务节点的数据调度总是成功的，比如服务节点网络中断，意外退出等等情况。为解决这些问题，我们采取以下策略：

(1) 设定数据调度返回时间上限 T_{up}，以此判断服务节点调度失败；

(2) 优先向其他服务节点及其候选服务节点提出调度请求；

(3) 当(2)不能响应时，可将数据调度到服务器，这时的数据都是比较紧急的，或是其他服务节点上没有的数据。

5 性能分析及测试结果

本文分析了多发送节点下数据调度问题，并提出了相应策略和方法。采用本文所述算法进行数据分配，并和 OTS_{p2p}[1]和 FSS[2]进行了比较。

测试环境：利用 OPNET 仿真工具生成由 500 个节点组成的 100M bit 总线以太网环境，各节点提供固定的出口带宽(25Kbit/s，50Kbit/s，100Kbit/ s，200 Kbit/ s，400 Kbit/s)，相应带宽节点数目分别为 50，100，200，100，50，数据包大小为 500 Kbyte，总共 100 个数据包，若媒体正常播放速率为 300 Kbit/s，将此作为接收节点数据包的消耗速度。节点选择采用静态分配策略，用随机变量 μ(t) 代表数据包的 S_i 传输的起始时

表 1　性能比较

	应用范围	延迟特性	算法复杂度
OTS	窄	优	一般
FSS	广	较差	简单
MBDS	广	优	复杂

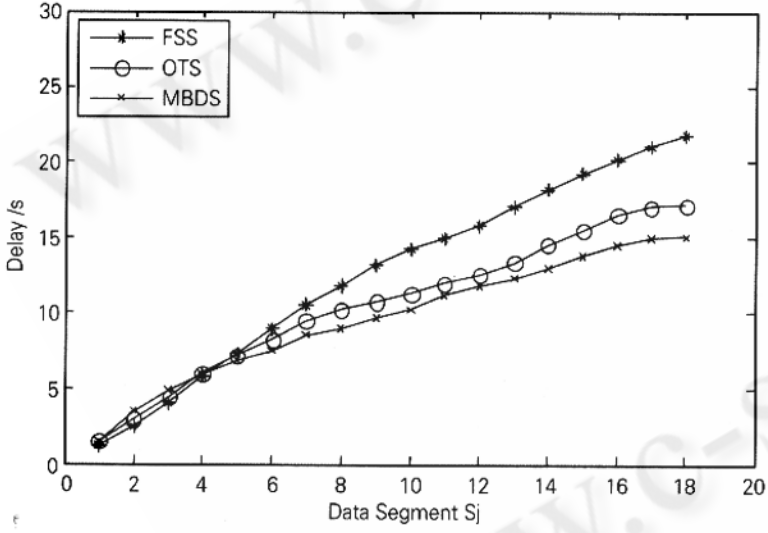

图 2　测试对比

间，记录不同算法在同一节点选择策略下各个数据块 s_j 的延迟时间 t，分析结果如表 1，测试结果如图 2 所示。结果表明 OTS 和 MBDS 的延迟性能都较 FSS 优良（如表 1），在一段时间后，延迟都将趋向稳定。

参考文献

1 D. Xu, M. Hefeeda, S. Hambrusch, and B. Bhargava. On peer – to – peer media streaming. In Proc. of IEEE ICDCS, Vienna, Austria, July 2002.

2 Jin B. Kwon and Hcon YYeom. Distributed Multimedia Streaming over Peer – to – Peer Networks, http://dcslab. snu. ac. kr/paper/europar. pdf.

3 S. Floyd, M. Handley and J. Padhye. Equation – based Congestion Control for Unicast Applications. Technical Report TR – 00 – 03, International Computer Science Institute, Berkeley, CA, March 2000.

4 R. Rejaie, M. Handley and D. Estrin. RAP: an End – to – end Based Congestion Control Mechanism for Realtime Streams in the Internet. In Proceedings of IEEE INFOCOM, New York, NY, March 1999: IEEE.

5 LIU Min, LI Zhong – Cheng, GUO Xiao – Bing, DENG Hui. An End – to – End Available. Bandwidth Estimation Methodology, Journal of