基于累积时延的流媒体传输模糊拥塞控制

汪学舜[1,2] 余少华[2] 戴锦友[2] 罗 婷[1,2]

(华中科技大学计算机学院 武汉 430074)[1] (武汉邮电科学研究院 武汉 430074)[2]

摘 要 为保证网络流媒体传输质量,在流媒体的传输中需要采用有效的拥塞控制策略。结合流媒体数据对时延敏感的特点,提出了一种基于累积时延的模糊拥塞控制算法,该算法在流媒体数据流传输过程中检测和跟踪其时延,在转发分组数据前,根据容忍时延阈值,丢弃超时数据包,减少不必要的带宽浪费,并且对所到达的数据流按照累积时延进行优先级分类,把全局性缓冲区和各队列的局部性缓冲区按照正常、拥塞避免和拥塞的规则划分为3个具有交叉过渡域的阶段,然后采用整体和局部相结合的拥塞控制方法,实现队列调度过程中的模糊处理,从而对网络拥塞进行有效的控制。理论分析和实验结果表明,使用基于累积时延的模糊拥塞控制算法,能有效改善流媒体的传输性能,是解决流媒体传输拥塞控制的有效途径,并能对提高网络性能起到重要作用。

关键词 流媒体,拥塞控制,动态队列管理,累积传输延时

Fuzzy Congestion Control Algorithm Based on Accumulative Traffic Delay

WANG Xue-shun[1,2] YU Shao-hua[2] DAI Jin-you[2] LUO Ting[1,2]

(College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)[1]

(Wuhan Research Institute of Post and Telecommunication, Wuhan 430074, China)[2]

Abstract In order to guarantee the quantity of flow medium traffic delivered in the network, congestion control strategy is adopted in flow medium traffic. According to the delay sensitivity of flow medium data, this paper proposed a novel congestion control for multimedia transmission in networks known as Accumulative Delay Fuzzy Congestion Control (ADFCC). ADFCC improves the QoS of multimedia stream in network by detecting and discarding these packets that accumulated delay exceeds multimedia stream's delay tolerance so as to maintain high bandwidth utilization. All kinds of packets were firstly classified into queues according to their own priorities which calculated by accumulative delay. Then the buffer state was divided into three phases, including normal, congestion avoidance, and congestion according to their buffer usage ratio. The three phases are crossover each other because of their fuzziness. Then by combining the whole congestion control, with the part congestion control, the fuzzy congestion control algorithm was carried out. Simulation results show that the proposed scheme can effectively reduce the average received packet delay than that of using traditional congestion control algorithm. Moreover, it is suitable to be used in the ever-changing network environment and can improve utilization of the network resources.

Keywords Flow medium, Congestion control, Dynamic queue management, Accumulative traffic delay

随着通信技术和多媒体技术的飞速发展,网络上多媒体信息和实时任务的数量与日俱增。尤其是随着IPTV、视频监控、视频会议以及VoIP(Voice over IP)等应用的出现和普及,更加剧了这种趋势。越来越多的音、视频传输流媒体,造成了网络传输信息流的严重拥塞。媒体传输拥塞会引起过多的信元丢失和不可接受的端到端的信元传输延时,从而引起信息传输质量的下降。如何更好地预防和控制流媒体传输拥塞成为近年来网络研究领域的重要问题。解决流媒体拥塞所实施的拥塞控制的目的,一是确保网络用户得到预定的服务质量,二是优化网络资源的利用,以取得更好的网络利用效率。

1 引言

根据拥塞控制算法的实现位置,可以将拥塞控制算法分为两大类:链路算法(link algorithm)和源算法(source algorithm),近年来,已有许多前人开展了有益的工作。在源算法方面,使用最广泛的是TCP协议中的拥塞控制算法,出现了许多TCP拥塞控制机制的改进算法,包括慢启动(slow start)、拥塞避免、快速重传(fast retransmit)、快速恢复(fast recovery)、选择性应答(SACK)等;近年来对UDP协议中拥塞控制机制的研究也取得了较大的进展[1-4],这些研究对提高网络传输性能、适应网络的飞速发展起到了重要作用。

相对于源算法,链路算法中位于网络内部的路由器能更早地感知到拥塞。为此,人们提出了主动队列管理机制,其中最著名的要数RED算法,研究者们为克服RED的缺陷又研究了一系列RED的衍生算法,如Cisco公司在其7000系列和7500系列路由器产品中采用的WRED、基于稳定性考虑的自适应RED和SARED等;从资源管理的角度进行研究的

到稿日期:2009-07-08 返修日期:2009-09-24 本文受863国家重点基金项目(2005AA121410)资助。

汪学舜 男,博士生,主要研究方向为流媒体数据传输QoS保证、组播路由算法等,E-mail:wang_xueshun@163.com;余少华 男,博士生导师,主要研究方向为光通信和光传输网络;戴锦友 男,博士,主要研究方向为ATM电路传输;罗 婷 女,博士生,主要研究方向为软件可靠性。

AQM算法也引起了广泛的关注，如对系统缓冲资源进行监控与管理的DT算法；最近，又提出了基于PI控制器的PI[5]算法、多路径双拥塞控制[6]、结合时延和吞吐量的比例调度[7]算法等新思想。

然而对于多媒体的数据流来说，在传输过程中积累的时延如果超过用户的容忍阈值，那么即使正确地传送到客户端，它也将变得不可用。这是多媒体数据流和其他数据流之间的一个很重要区别。所以，在发生拥塞时，RED，DropTail等传统拥塞控制算法均无法解决该问题。本文根据流媒体数据的特点，提出了一种基于时延自适应的流媒体传输拥塞控制算法ADFCC。该算法通过检测和丢弃那些时延值超过用户容忍阈值的数据帧，并且自适应控制对流媒体数据进行分类传输，在不损害高优先级数据流量的情况下能较好地实时调整各优先级的门限值来减少网络资源的消耗，提高网络资源的利用率。通过实验分析和仿真研究表明，本文给出的算法在解决网络传输拥塞上性能是优越的，适合于当前的一般流媒体有效传输。

2 实现机制描述

2.1 拥塞控制模型

网络中到达路由器或者交换机同一转发端口的数据流，通常来自于不同的源端。如果对它们都采取同样的调度策略，那么，一些需要高速服务要求的数据流会因此得不到应有的服务，而一些低速的数据流会由于拥塞而占用更多宝贵的缓冲区资源。因此，在本文的拥塞控制算法中，首先根据数据流的累积时延，对各数据流进行分类，具体分类方法见2.2节。这样，要求由路由器某接口转发的所有数据包流都已经按照一定的分类标准被分类。

图1给出了结合令牌桶整形的拥塞控制算法的队列调度模型。其中，全局丢包模块在全局缓冲区占用率达到全局阈值时，进行全局性的随机丢包；而每种类型队列之前，数据包流首先经过一个令牌桶整形模块，对突发性流量进行整形，然后是局部丢包模块，它在该队列缓冲区占用率达到相应的局部阈值，且发生全局性拥塞避免或全局性拥塞时，对该队列进行局部丢包，也就是在网络性能变坏时，对过多占用缓冲区的长队列实行惩罚性丢包。下标 n 为当前的队列总数。C_j 为 j 类队列缓冲区的大小，l_j 表示相应队列的实时长度，并定义 $u_j=l_j/C_j$ 为该队列的缓冲区占用率。

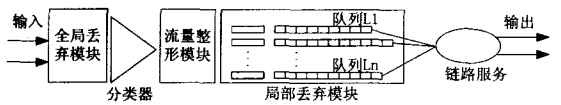

图1 拥塞控制模型

对于不同队列数据包的转发，采用加权轮询的方式进行，如果某一队列在轮到它时，没有待发的数据包，则轮到下一队列转发，并且在某一队列转发的过程中不允许中断，如果某一队列在属于它的服务时间段到达之前为空，同样也将轮到下一队列转发。而对于同一队列的数据包，则采取先入先出(FIFO)策略进行转发。

前面提到，各种不同类型的网络数据包的优先级要求是不一样的，因此对于不同的优先级赋予不同的权值，比如赋予最低优先级队列的优先级权值 r_1 为1，再高一级的优先级权值 r_2 为2，依次直到最高级别 r_n 为 n，而对于相同级别的数据流则都要进入到同一类型队列进行转发。

如果共享出口的链路带宽为 W，那么，某一级别类型的队列 j 就可以获得 $bw_j=W\cdot r_j/(\sum r_j)$ 部分的带宽，从而保证相应级别的数据流获得相应质量的服务。

2.2 流媒体数据优先级的设置与计算

流媒体应用的视频流或者音频流在分组交换的网络节点上传送时，一般假定数据分组之间维持着一个恒定的延迟关系，但事实上这是很困难的。在实际的传输过程中，由于拥塞的发生或者传输延迟的不同造成了数据分组在到达目的地时产生网络延迟的变化，造成了这些多媒体数据流服务质量的降低。当多媒体数据流穿越网络时，如果产生的时延过大，用户即使收到这些数据流，对用户来说它也将变得不可用，因为在用户端很难对一个数据流或多个数据流之间的数据分组再进行时序之间的调整。这是流媒体数据流和其他数据流之间一个很重要的区别。如果有大量的流媒体数据分组在网络传输过程中累计的时延超过了客户的容忍程度，那么这种情况将变得更加糟糕。这种情况下，这些多媒体数据虽然对用户来说已经不可用，但是仍然占用着网络的带宽，而传输这些无用的数据无疑增加了网络的拥塞程度。因此，在流媒体传输过程中，时延是极为重要的控制参数。

基于累积时延的模糊拥塞控制算法的基本思想是在流媒体数据流传输过程中检测和跟踪其时延，如果时延超过了用户的容忍阈值，那么就将该数据分组进行丢弃，这样可以减少不必要的带宽浪费。为了保持跟踪多媒体数据帧在端到端的传输过程中累积的时延，需要一个计数器。显然该计数器要在数据帧中的某个字段，来记录在整个传输过程中所经历的时延。当前的组网方式主要有两种：基于路由器的组网方式和基于以太网交换机的组网方式。下面详细说明时延计数器的实现方式。

基于路由器的组网方式中，路由器处理的是IP分组，所以我们选取在IP头中的ToS字段的3个bit作为时延值的计数器，同时ToS字段又代表了业务的等级。对于流媒体数据流来说，那些时延值较小的数据应该优先转发，因此它们对应较高的业务等级，而对于那些时延值已经很大的数据帧来说，可能对用户已经没有意义，在发生拥塞时应该首先丢弃，所以对应较低的业务等级。由此可以得出时延值和业务等级之间的对应关系是较大的时延值对应较低的优先级，而较低的时延值对应较高的优先级。本算法规定将时延值根据用户可以容忍的门限分为8个不同的等级，即从等级0到等级7。由于时延值越高优先级越低，而时延值越低优先级越高，因此等级0对应的二进制比特为111；等级1对应的二进制比特为110，以此类推。比如，如果时延值在等级3中，那么对应的优先级为100，即对应的优先级为4。从上面的分析可以看出，优先级的值正好为时延值等级的取反运算。具体的对应关系如图2所示。

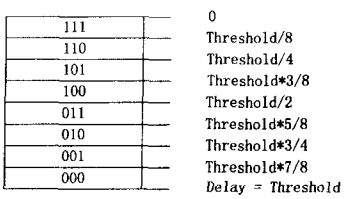

图2 优先级与时延值之间的对应关系

基于以太网交换机的组网方式大多应用在城域以太网中。在当前的城域网领域，以太网已经是最热门的解决技术之一。事实上，当前很多运营商都在城域网上实施以太网业务。然而，以太网技术向城域网延伸必然要面临和解决许多棘手的问题，例如突破 4096 个 VLAN 数的限制、透明 LAN 业务连接、服务质量保证等一系列问题。为了在城域以太网中突破 4096 个 VLAN 的限制和实现透明 LAN 业务传输，IEEE 在 2005 年定义了 Q-in-Q 的标准 IEEE802.1ad[8]（Provider Bridge）。Q-in-Q 技术中，外层的 VLAN 标签称为 S-VLAN，该 VLAN 是属于运营商网络范围内的 VLAN 空间的，而内层的 VLAN 标签称为 C-VLAN，是属于客户端的 VLAN 空间的。当运营商在 UNI 处收到用户发出的以太网帧后，就把一个 S-VLAN Tag 标签添加到用户 IEEE 802.1Q Tag 标签标记的以太帧。IEEE802.1ad 的 Q-in-Q 以太网帧格式如图 3 所示。

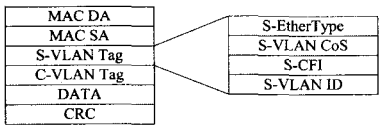

图 3　带有 Q-in-Q VLAN tag 标签的以太帧格式

S-VLAN Tag 标签是嵌在以太网目的 MAC 地址和源 MAC 地址之后，在 C-VLAN 之前。在 802.1Q VLAN 中，VLAN 中的优先级占用了 3 位，支持 8 个不同的优先级，用来区分不同的服务。在城域以太网中进行数据帧的转发时，通常只考虑 S-VLAN 中的优先级设置，S-VLAN ID 和 S-VLAN CoS 的设置常用于识别服务和服务性能（CoS）。而 C-VLAN Tag 标签通常不被改变，也不被访问。

C-VLAN Tag 标签中的优先级在运营商网络的传输过程中是不使用的，因此在基于以太网交换机的组网方式下，选用 C-VLAN Tag 中的优先级 bit 作为时延计数器。时延值和优先级的对应关系仍然采用和路由器组网方式下相同的对应关系，具体如图 2 所示。

2.3　基于队列的模糊拥塞控制

由于拥塞这一概念本身的模糊性，我们不可能用精确的数值来描述某一队列或者是全局的具体状态，只能说它们处于某一阶段的程度。比如说，某一队列当前的缓冲区占用率为 80%，那么我们会认为该队列当前发生拥塞的可能性极大，有可能还处于拥塞避免阶段，不可能正常。同样，对于全局缓冲区的占用情况也是一样的。因此，引入相关的模糊理论来描述这一模糊性的过程，充分利用模糊理论在处理不确定性问题上的优越性，以达到优化的拥塞控制性能。

根据 Zadeh 对模糊子集的定义，定义论域 U：缓冲区占用率（可以是队列缓冲区或全局缓冲区占用率）。取 U 到闭区间[0,1]上的 3 个映射 $\mu L: U \to [0,1]$，$u \to \mu L(u)$；$\mu K: U \to [0,1]$，$u \to \mu K(u)$；$\mu C: U \to [0,1]$，$u \to \mu C(u)$，确定出 U 的 3 个模糊子集：L, K, C. $\mu L(u)$，$\mu K(u)$，$\mu C(u)$，分别称为 u 对于 L，K，C 的隶属函数。其中，模糊子集 L，K，C 分别表示图 4 中所描述的 3 个阶段（同样也适用于全局缓冲区占用率情况），即正常、拥塞避免和拥塞 3 个阶段。

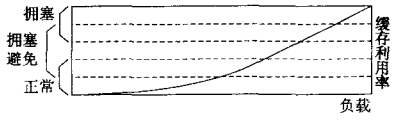

图 4　队列实时阶段的划分

关于隶属函数的确定，带有一定的主观性，不同的人对于同一问题可能会建立起不同的函数形式，但只要能够反映客观事实，就认为是正确的。根据文献[9]中的模糊统计方法，确定出队列缓冲区占用率 U 上的 3 个隶属函数 $\mu L(u)$，$\mu K(u)$，$\mu C(u)$，如图 5 所示。为了方便描述，将经过模糊统计方法得到的隶属函数进行简化，如图 6 所示。

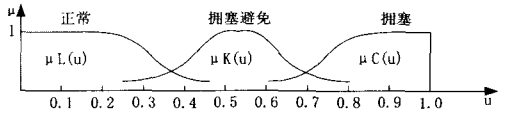

图 5　隶属函数

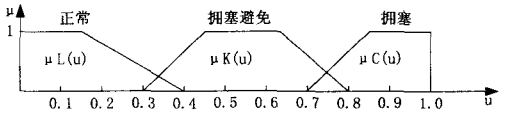

图 6　简化的隶属函数

队列的缓冲区占用率被划分为 3 个语言变量{Less Congestion，Knee Congestion，Cliff Congestion}。其划分的具体意义为：Less Congestion（简写为 L），表示该队列的缓冲区占用率处于正常状态；Knee Congestion（简写为 K），表示该队列的缓冲区占用率处于拥塞避免阶段；Cliff Congestion（简写为 C），表示该队列的缓冲区占用率就要达到饱和，处于拥塞状态。用决策集 V 来描述它们：$V=\{L,K,C\}$，以评价队列所处的实时状态。

上述是对拥塞程度描述为单个队列的情形。同样，这一过程描述也适应于整个队列的全局性缓冲区的占用情况。

为了得到各类型队列和全局性缓冲区占用的实时情况，采用计数的办法，随时跟踪缓冲区的情况，一旦各占用率变量处于相应隶属函数的不同区间时，就会触发相应动作，从而智能地控制拥塞，并取得更好的效果。表 1 描述了模糊拥塞处理的评判规则。在所有的队列当中，尽管各队列所对应的连接具有它们各自的权值和带宽，但总会有一些连接在某一时间段内出现持续的高速现象，这种高速到达是要超过该连接所拥有的缓冲区资源的。于是，相应的队列长度快速增长，它的缓冲区占用率很快进入拥塞范围，而此时，全局有可能还处于正常状态。但是，我们不能完全排除该队列是由于较大流量的突发性造成的，因此，如果此时全局缓冲区占用率还处于正常状态，这表明缓冲区资源还很丰富，于是我们就扩充分配给该队列的缓冲区容量，使队列退出拥塞状态，以免在网络负载较轻时，长的队列流过早地受到惩罚。

这种局部性拥塞处理与全局性拥塞处理是相互独立的，只要条件得到满足，二者就可以同时启动，也可以单独启动。

表 1　模糊拥塞处理的评判规则

区域	正常（整体）	拥塞避免（整体）	拥塞（整体）
正常（队列 j）	整体状态正常，队列 j 正常	整体状态正常，队列 j 正常	根据图 1 模型进行整体丢包
拥塞避免（队列 j）	整体状态正常，队列 j 正常	整体状态正常，队列 j 正常	根据图 1 模型进行整体丢包
拥塞（队列 j）	整体状态正常，扩大队列 j，避免拥塞	整体状态正常，根据图 1 模型进行部分丢包	根据图 1 模型进行整体丢包和部分丢包

2.4　算法处理步骤

为了实现的方便，用状态矩阵实时表示各类型队列的局部状态，定义状态矩阵：

$$R_{(3\times n)}=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ r_{31} & r_{32} & \cdots & r_{3n} \end{bmatrix}$$

它的元素 r_{ij} 表示队列 l_j 属于状态 L,K,C 的程度($i=1,2,3;j=1,2,\cdots,n$)。因此如果 $r_{ij}=1$,则表示队列 j 处于状态 $i(i=1,2,3;j=1,2,\cdots,n)$;反之,则不属于某一状态。同样地,类似地定义全局缓冲区占用率的实时状态设置状态矩阵 $R_{(3\times1)}$。

在检测到拥塞发生时,进行惩罚性丢包,惩罚性方法按照CHOKe方法进行。具体为:当一个分组到达拥塞的队列时,CHOKe从相应优先级队列中随机地挑出一个分组进行比较。如果它们属于同一个流,则这两个分组都被丢弃;否则,被挑出的分组依然留下,而刚到达的分组则以某种概率被丢弃。

综上所述,交换设备收到数据帧的算法处理步骤如下。

(1)计算累积时延

读取流媒体数据帧中携带的时延计数器对应的优先级值,假设读取的优先级为 j,变量 $delay$ 表示以太网数据帧在交换机中经历的时延,在交换机中的时延可以通过以下方式来估计:输出队列 j 的当前缓冲区占有率(u_j)以及队列 j 可用的链路带宽(bw_j):

$$delay=\frac{u_j * Packet\_size}{bw_j}$$

(2)更新累积时延并分级

由上一步计算的本地时延值加上流媒体数据帧中携带的时延计数器对应的值。如果 total_delay 超过了用户的容忍阈值,则该以太网帧就被丢弃。如果 total_delay 在用户容忍阈值之内,那么按照对应关系将时延计数器的值转换为数据帧中相应字段的计数,即确定该数据帧的优先级,更新时延计数器 counter。然后进入缓冲区等待,在队列长度历史表 Length 相应的项中记录此时的平均队列长度。

(3)当收到的数据帧数大于或等于 n 时,首先从高优先级的分组所在的缓冲区开始,计算这段时间该缓冲区的平均占有率。根据各类型队列缓冲区占用率设置状态矩阵 $R_{(3\times n)}$,根据全局缓冲区占用率设置状态矩阵 $R_{(3\times1)}$,同时修改相应队列长度上限。

(4) 根据队列 j 状态矩阵进行转发

如果状态矩阵 $R_{(3\times n)}$ 中 $r_{1j}=1$,正常转发;

如果状态矩阵 $R_{(3\times n)}$ 中 $r_{2j}=1$,则进行随机丢弃,若丢弃,按 CHOKe 进行惩罚性的丢包。

如果状态矩阵 $R_{(3\times n)}$ 中 $r_{3j}=1$,则进行丢弃,按 CHOKe 进行惩罚性的丢包。

(5)结束

上述方法中,第 3 步状态矩阵设置方法如下。

1)首先根据各类型队列缓冲区占用率的实时状态设置状态矩阵 $R_{(3\times n)}$,根据全局缓冲区占用率的实时状态设置状态矩阵 $R_{(3\times1)}$;

2)如果全局状态矩阵 $R_{(3\times1)}$ 中 $r_{11}=1$,则搜索队列状态矩阵 $R_{(3\times n)}$ 第 3 行中 $r_{3j}=1$ 的列,扩充相应队列 j 的缓冲区容量 C_j,使该队列缓冲区占用率不在拥塞状态;

3)如果队列 j 的缓冲区容量 C_j 进行了扩展,并且全局状态矩阵 $R_{(3\times1)}$ 中 $r_{31}=1$ 或者 $r_{21}=1$ 且 $r_{11}\neq1$,则恢复队列 j 的缓冲区容量 C_j 到原来的大小;

4)如果全局缓冲区占用率由一种状态转换到另一种状态,或者是同时处于两种状态的过渡阶段,此时,描述全局缓冲区占用率状态的全局状态矩阵 $R_{(3\times1)}$ 中相应的元素要进行 0 或者 1 的转换;

5)如果某类型队列的缓冲区占用率由一种状态转换到另一种状态;或者是同时处于两种状态的过渡阶段;此时;描述各种类型队列缓冲区占用率状态的队列状态矩阵 $R_{(3\times n)}$ 中相应列的元素要进行 0 或者 1 的转换;

6)结束。

3 实验与分析

时延对于多媒体业务是非常重要的一个参数指标,保证较小的时延是提高多媒体业务服务质量的一个重要手段。在试验中主要针对时延比较分析了 ADFCC 算法和传统拥塞控制算法 RED,DropTail 的性能。

本文采用 NS2 来进行算法的仿真试验,图 7 给出了仿真试验的拓扑图。在拓扑图中主机通过用户接入交换机(图中的 Customer switch)上,进而连到运营商网络中。从用户主机发出的数据流是不带 VLAN Tag 标签的,当数据流进入用户交换机后,在用户交换机上和用户主机的连的端口上会为数据流打上 C-VLAN Tag 标签,在这里根据端口来分配 C-VLAN。试验中用户主机 A1－A4 的 VLAN ID 分别为 100－400,但不带优先级。在用户交换机的上联端口上配置为 VLAN TRUNK 模式。这样从用户上联端口出来的数据流就带有 C-VLAN Tag 标签了。在进入 SP 交换机后,SP 交换机按照端口来为数据流分配 S-VLAN Tag 标签,用户数据流在进入 SP 网络之后就变成了 Q-in-Q 模式。试验中,两个用户交换机相连的 SP 交换机上的端口的 SP VLAN 分别为 1000 和 2000。由于本文的 ADFCC 算法是在 SP 网络中运行的,因此仿真试验涉及的主要是运营商网络部分,即图 7 中的 SP Networks 部分。

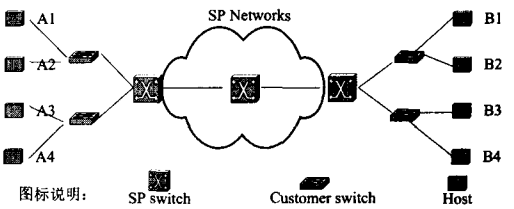

图 7 仿真试验拓扑图

在每个用户交换机连接的两个用户主机上分别发送 TCP(利用 FTP 流来代替)和 UDP(多媒体的流,比如 MPEG-2 的视频或音频,比特率为 6Mb/s)的数据流。设定用户交换机到 SP 交换机的链路容量为 10Mb/s,传输延迟为 3ms。MPEG-2 视频流的样本采用 VBR 格式。为了试验的简单,在下面的讨论中采用 UDP 流来代表视频流。实验主要从时延、有效吞吐量和 TCP 流的友好性 3 个方面进行。

3.1 时延的比较

在试验中数据帧从 A 端的主机发送,在进入用户端交换机后,将穿过一个有 3 个 SP 交换节点的链路,然后到达另一端的用户端交换机。SP 交换机之间的节点彼此之间通过 10Mb/s 的带宽链路相互联,它们之间的传播延迟为 2ms。这 3 个 SP 交换机之间的链路是整个链路上的瓶颈。RED 和 ADFCC 算法在 SP 交换节点上实现,用来对 TCP 流和 UDP 流进行拥塞控制。在试验中设定每个节点上队列中数据帧的限制为 150 个;试验中测试 256byte,512byte,1024byte 3 种帧长,测试时间为 3min。视频流的比特率设定为 6Mbps,而

FTP 的传输速率也设置为 6Mpbs。时延算法的门限设置为 0.5s;RED 算法中的参数设置为最大门限 100 个帧,最小为 30 个帧。

图 8,图 9,图 10 分别给出了 3 种算法在不同帧长情况下时延平均值的比较图。从图中可以看出,ADFCC 算法在各种帧长的情况下,平均时延值都要明显低于传统的 RED 和 DropTail 算法。从图中还可以看出 3 种算法,即 DropTail,RED 和 ADFCC 算法都有相同的曲线趋势,这种周期性的趋势主要是因为 TCP 协议本身的流量控制机制所导致的,而我们的算法呈水平趋势,说明了动态调节队列长度的效果。图中的试验结果表明本文提出的 ADFCC 算法相比于 DropTail 和 RED 算法可以达到更低的时延。

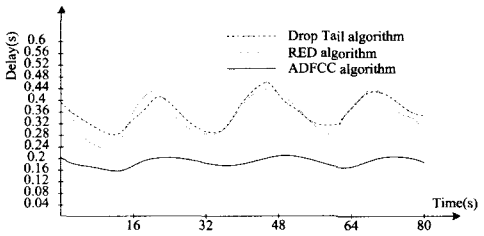

图 8 平均时延比较(帧长为 128 字节)

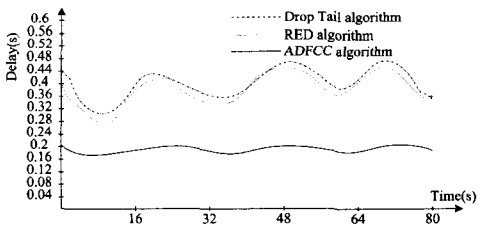

图 9 平均时延比较(帧长为 512 字节)

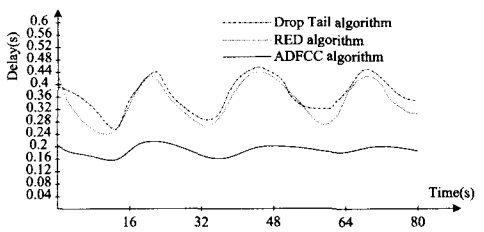

图 10 平均时延比较(帧长为 1024 字节)

3.2 有效吞吐量的比较

多媒体的 UDP 流对于时延是很敏感的。在分组的时延比用户的容忍阈值大的情况下,分组即使被成功地传送到客户端,实际上对于用户来说已经没有意义。而且如果没有足够的带宽,那么将可能增加发生拥塞的可能,在交换节点上转发这些时延较大的分组将减少传输信道的可用带宽,而且会导致其他的一些数据帧增加时延,致使更多的数据帧变成对用户来说没有意义的数据。所以在 ADFCC 算法中将超过用户容忍阈值的数据帧丢弃。

实际上在数据的传输过程中,对用户来说关心的是对其有用的输出而不单是吞吐量。这里的"有效"意味着多媒体数据帧到达客户端时满足时延容忍。"有效"的多媒体数据帧能满足在播放给用户之前到达它的位置。

实验仍然采用图 7 的拓扑图,每个用户交换机连接的两个用户主机上分别发送 FTP 流和 UDP 的视频数据流。设定用户交换机到 SP 交换机的链路容量为 10Mb/s,传输延迟为 3ms,MPEG-2 视频流的样本采用 VBR 格式。在试验中设定的参数与实验 1 中相同,仍然测试 256byte,512byte,1024byte 3 种帧长的情况,测试时间为 3min。视频流的比特率设定为 6Mbps,FTP 的传输速率也设置为 6Mpbs。时延用户容忍阈值在试验中设置在 0.1～0.5s 的范围内。试验结果在图 11、图 12、图 13 中显示。试验结果表明,ADFCC 算法相比于 DropTail 算法和 RED 算法可以达到较高的平均有效吞吐量。整个的趋势在图 11、图 12、图 13 中是可以很清楚地看到的,这说明试验结果和理论分析是一致的。

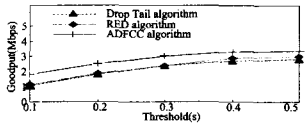

图 11 有效吞吐量的比较(帧长为 128 字节)

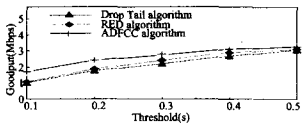

图 12 有效吞吐量的比较(帧长为 512 字节)

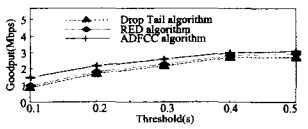

图 13 有效吞吐量的比较(帧长为 1024 字节)

3.3 TCP 流友好性的比较

UDP 是一种无连接协议,它没有相应的流量控制机制,不会对丢包做出相应处理。当 UDP 流和 TCP 流共享带宽时,UDP 流的这种特性会贪婪地占用大量的带宽,而降低 TCP 流的吞吐量。因此,本文提出的 ADFCC 算法有确定的 TCP 友好性是很重要的,它不能降低 TCP 流的吞吐量,要能达到和传统的 RED,DropTail 算法相同的友好性。在试验中,将 TCP 流和 UDP 流放入 FIFO 队列之前,针对它们分别应用不同的队列管理策略。

算法的 TCP 友好性是直接与 UDP 吞吐量和 TCP 吞吐量的比例相关的。图 14 给出了在不同帧长下的 UDP 和 TCP 吞吐量之间的比例。从试验中可以看出,ADFCC 算法和其他成熟的拥塞控制算法相比在 TCP 友好性方面是相接近的。

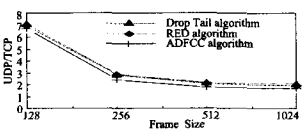

图 14 在不同帧长情况下的 UDP 和 TCP 吞吐量的比例

结束语 本文结合流媒体数据对时延敏感的特点,提出了一种基于累积时延的模糊拥塞控制算法,该算法在流媒体数据流传输过程中检测和跟踪其时延,在转发分组数据前,根据容忍时延阈值,丢弃超时数据包,减少不必要的带宽浪费,并且对所到达的数据流按照累积时延进行优先级分类,把全局性缓冲区和各队列的局部性缓冲区按照正常、拥塞避免和拥塞的规则划分为 3 个具有交叉过渡域的阶段,然后采用整体和局部相结合的拥塞控制方法,实现队列调度过程中的模糊处理,从而对网络拥塞进行有效的控制,并能提高网络资源利用率。实验结果表明该算法,该算法能有效改善流媒体的传输性能,实现有效的拥塞控制策略,并提高网络流媒体传输质量。

(下转第 90 页)

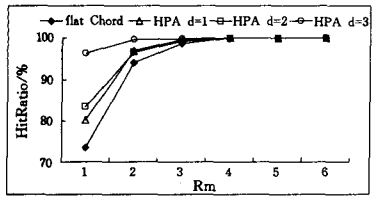

图4 不同 IR 和 d 下的查询命中率（IR=0.2）

然后，从单个查询的视角分析，得到8000个查询事件在 d=1时覆盖网和物理网的查询跳数，如图5和图6所示。d=2和3时的结果和结论类似，可以看出：1）物理网和覆盖网的查询跳数曲线的分布和相对位置是基本一致的；2）给定 IR，无论是物理查询跳数还是覆盖网中的查询跳数，在平面式 Chord 中的要比 HPA 覆盖网中的高很多；3）除去 R_m=1 的情况，对每根曲线而言，当 R_m=6 时可以获得最小的查询跳数；4）当 IR=0.0 时，查询跳数随 R_m 增大而单调减小；而当 IR=0.1 或 0.2 时，在 R_m=2 或 3 时获得查询跳数的最大值，这意味着当某个副本未被找到时，查询继续进行，查询消息将被发送到下一个离当前节点最近的副本提供者，因此查询跳数增加；当 $R_m \geqslant 4$ 时，所需平均查询跳数渐渐减小，这是因为可用副本增多，副本查找失败的几率相对下降。与 R_m=1 的情况相比，基于多 hash 函数的复制技术（R_m>1）能获得更高的查询命中率，不过会带来更长的查询路径。

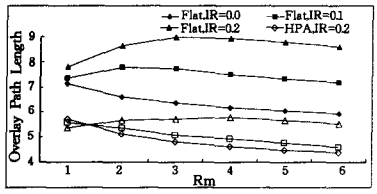

图5 覆盖网中的查询跳数（d=1）

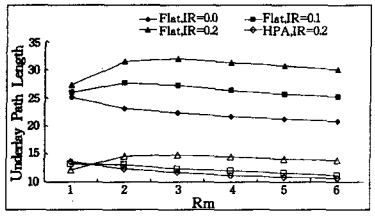

图6 物理网中的物理查询跳数（d=1）

结束语 本文讨论了一个结构化的、实现了拓扑优化的覆盖网，以及在此覆盖网中基于多 Hash 函数的复制发布与查询策略。本文提出的方法能有效地减少冗余流量，合理地发布副本，通过理论分析和仿真，证实了这些方法的有效性。深入的工作包括考虑节点的动态行为、负载均衡，对路由方法展开优化，在保证现有性能的基础上加快查询速度，以及设计分布式的 SP 选择算法等。

参 考 文 献

[1] Chen G H, Li Z H. Peer-to-Peer network structure, application and design[M]. Beijing: Tsinghua University Publishing House, 2007

[2] Wan K H, Chris L. An overlay network for replica placement within a P2P VoD network[J]. International journal of high performance computing and networking, 2005, 3(5/6): 320-335

[3] Mondal A, Madria S K, Kitsuregawa M. EcoRep: an economic model for efficient dynamic replication in mobile-P2P networks [C]// Proceedings of 13th international conference on management of data. Dec. 2006: 185-196

[4] Min C, Ann C. A peer-to-peer replica location service based on a distributed hash table[C]// Proceedings of ACM/IEEE conference on supercomputing. Nov. 2004: 56-67

[5] Lv Q, Li K, Cao P, et al. Search and replication in unstructured peer-to-peer networks[C]// Proceedings of the 16th international conference on supercomputing. New York, USA, June 2002: 84-95

[6] Zhou J, Zhang X. Clustered K-Center effective replica placement in peer-to-peer systems[C]// Proceedings of Global telecommunications conference 2007. Nov. 2007: 2008-2013

[7] Bassam A A, Chen W, Zhou B B, et al. Effects of Replica Placement Algorithms on Performance of structured Overlay Networks[C]// Proceedings of IEEE International Parallel and Distributed Processing Symposium. 2007: 460-467

[8] Marcel W, Paul H, Daniel B. Dynamic replica management in distributed hash tables[R]. RZ-3502. IBM, July 2003

[9] Xia Y, Chen S G, Cho C. Algorithms and performance of load-balancing with multiple hash functions in massive content distribution[J]. Computer Networks, 2009, 53(1): 110-125

[10] Lu K C, Lu H M. Graphics Theory and its Applications[M]. Beijing: Tsinghua University Press, 2005

（上接第61页）

参 考 文 献

[1] Larzon L-A, Degermark M, Pink S, et al. The Lightweight User Datagram Protocol (UDP-Lite)[S]. RFC 3828, Internet Engineering Task Force, July 2004

[2] Zhang X, Li Ch, Li X, et al. Multicast congestion control on many-to-many videoconferencing[C]// The Second International Conference on Future Generation Communication and Networking. 2008: 260-263

[3] Zhang Zaichen, Li O K. Network-Supported Layered Multicast Transport Control for Streaming Media[J]. IEEE Trransactions on PAarallel and Distributed Systems, 2007, 18(9)

[4] Vanit-Anunchai S, Billington J, Kongprakaiwoot T. Discovering chatter and incompleteness in the Datagram Congestion Control Protocol[C]// Proc. 25th IFIP International Conference on Formal Techniques for Networked and Distributed Systems. Oct. 2005

[5] Hong Yang, Yang W W. Design of Adaptive PI Rate Controller for Best-Effort Traffic in the Internet Based on Phase Margin [J]. IEEE Transactions on Parallel and Distributed Systems, 2007, 18(4)

[6] Voice T. Stability of Multi-Path Dual Congestion Control Algorithms[J]. IEEE/ACM Transactions on Networking, 2007, 15 (6)

[7] Kamal A E, Sankaran S. A Combined Delay and Throughput Proportional Scheduling Scheme for Differentiated Services[J]. Journal Computer Communications, 2006, 29: 1754-1771

[8] IEEE Std 802. 1ad™-2005[S]. IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks-Amendment 4: Provider Bridges. 2005

[9] 李士勇. 模糊控制. 神经控制和智能控制论[M]. 哈尔滨：哈尔滨工业大学出版社，1996：42-43