

Internet视频传输的拥塞控制策略

赵黔莉, 涂国防

(中国科学技术大学研究生院, 北京 100039)

摘 要: 网络拥塞是影响Internet视频通信质量的一个重要因素。该文分析了单播和组播中视频传输的各种拥塞控制策略, 并对各种拥塞控制策略的带宽利用率和服务灵活性进行了比较, 对提高Internet视频传输的质量有一定的意义。

关键词: Internet; 拥塞控制; 视频; 单播; 组播; RTP

Congestion Control Mechanism for Video Transport over Internet

ZHAO Qianli, TU Guofang

(Graduate School, University of Science & Technology of China, Beijing 100039)

【Abstract】 Network congestion is an important point for the video communication application. This paper analyses several congestion control methods of video transport for the unicast and multicast. It also compares the bandwidth efficiency and service flexibility of these congestion control methods. These works are quite significant for video communication.

【Key words】 Internet; Congestion control; Video; Unicast; Multicast; RTP

随着Internet的发展, 传输实时的多媒体数据流日益成为Internet上的重要应用。例如视频会议、远程教学等。这类多媒体应用的主要特点是: (1)带宽需求大; (2)实时性要求高, 对时延敏感, 但能容忍一定的数据丢失; (3)涉及到多点通信; (4)对服务质量有一定的要求。而现有的Internet无法为实时视频传输提供QoS保证。

网络拥塞是影响Internet视频通信质量的一个重要因素。拥塞控制的目的是保证网络传输的平稳性和公平性, 并尽量提高视频传输的服务质量。传统的TCP协议采用的拥塞控制算法主要有慢启动、拥塞避免、快速重传和快速恢复。由于TCP的面向连接性和重传机制不适合实时性较高的视频数据传输, 因此通常视频应用在传输层采用的是UDP协议。可以说TCP的拥塞控制算法并不适用于基于UDP协议的视频数据传输, 为此需要研究新的拥塞控制策略, 以尽可能避免网络拥塞, 提高视频传输的服务质量。

由于Internet上网络状态的随机性和复杂性, 视频传输中拥塞控制策略是非常复杂的。本文针对单播和组播两种数据传输方式, 详细分析了各种拥塞控制策略, 对提高Internet视频传输的质量有一定的意义。

1 拥塞控制策略考虑的因素

(1) TCP友好性

视频应用通常采用UDP协议。而现今的Internet中, TCP占90%的通信量, TCP通信量是网络的主流。当网络发生拥塞时, TCP协议端到端的拥塞控制机制使得TCP流发送的数据量迅速减小, 而UDP协议没有拥塞控制机制, 在带宽减少的情况下, UDP流将不会减少发送的数据。当TCP流和UDP流共享网络带宽时, UDP流将会进一步侵占带宽, 使得TCP流和UDP流之间存在不公平性。因此实时视频应用应该具有一定的拥塞控制机制来达到TCP友好性。TCP友好性的含义是TCP流和一个具有相似传输特征的自适应流如UDP流经过同样的路径, 又具有相同的RTT和包丢失率, 应该享有类似的带宽。

(2) 收敛性

由于网络的状态随机变化, 视频传输的速率也必须随时

改变。收敛性通常是用从拥塞的起始状态到达稳定状态的时间或速度来衡量的。有效的拥塞控制策略应能较快地收敛。

(3) 平稳性

有效的拥塞控制策略应对视频质量的影响较平稳, 减少视频质量的大量波动。

2 Internet视频传输中的拥塞控制策略

2.1 单播的拥塞控制策略

单播是IP最常用的数据传送方式, 它只涉及一个发送方和一个接收方, 如图1所示^[1]。单播传输方式下的拥塞控制算法通常是发送方根据接收方的反馈调整发送速度, 它分为基于探测和基于模型两种控制方式。然而不论哪种拥塞控制算法首先都需要得到网络状态的反馈信息。

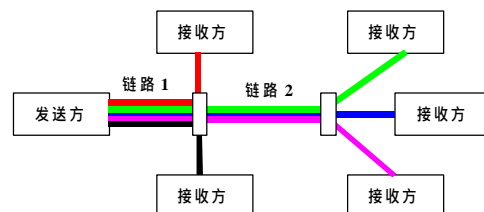


图1 单播

(1) 网络状态的判别

当前的Internet并不能显式地提供网络带宽的反馈信息(例如路由器通知发送方关于它自己的当前状态), 所以合理的拥塞控制过程只能通过端到端的间接信息(比如: 延时、包丢失率)来判断网络状态。这些信息都可以通过RTP/RTCP(实时传输与控制协议)得到。在一些文献中有采用包延迟作为反馈信息, 然而, 许多实验模拟结果都显示了: 基于包延迟的控制(Delay-based control)方法与基于包丢

基金项目: 国家自然科学基金资助项目(J60272056)

作者简介: 赵黔莉(1973—), 女, 硕士生, 主要研究方向: 互联网、多媒体等; 涂国防, 教授、博导

收稿日期: 2003-08-29

E-mail: zhaoql@cci.cn.net

失的控制 (Loss-based control) 方法相比, 当建立TCP连接时, 网络的吞吐量更低。换句话说: 在与TCP竞争时, 基于包延迟的控制会丢失一些带宽, 不具有TCP友好性。所以, 一般采用基于包丢失的控制方法^[3]。

1) 丢包的检测

丢包的判断没有统一的标准, 通常采用的方法为: 接收端通过检查接收到的RTP包的序号来检测是否有包丢失, 缓冲的上限为4个包, 即: 如果序号为 $n+1$ 、 $n+2$ 和 $n+3$ 的包都已收到, 而为 n 的包没有收到, 则认为第 n 个包丢了^[3]。

2) 平均丢包率的测量

在单播的环境中, 平均丢包率的测量较容易, 用某时间间隔内检测到的丢包数目除以时间间隔得到^[3]。

发送端根据接收端测量的平均丢包率来判断网络状态。

(2) 探索的拥塞控制

在基于探索的拥塞控制中, 发送方通过发送速率的调整来探索网络带宽, 在保持丢包率 p 低于某个丢包极限 P_{th} 的前提下, 调整发送速率。通常有两种调整方法: 一种是AIMD (Additive Increase Multiplicative Decrease), 即加法增加乘法减小。另一种是MIMD (Multiplicative Increase Multiplicative Decrease), 即乘法增加乘法减小。基于探索的拥塞控制是通过调整发送速率以适应网络带宽, 因此可以避免拥塞。

与RTP/RTCP标准一致, 我们让源端每隔 N_s 个RTP包周期性地发送一个RTCP控制包, 接收方每隔 N_r 个包或者至少5s一次发送一个RTCP反馈控制包, 反馈包中包含了在这个时间间隔内观察到的丢包率 P_{loss} 。编码器每收到一个RTCP包就执行一次速率调整, 因此, 源端进行两次速率调整的时间间隔大约等于返回的两次RTCP包的时间间隔^[3]。

下面是反馈控制算法中发送方和接收方的行为, 其中, IR 为初始速率(initial rate), MR 为最小速率(minimum rate), PR 为峰值速率(peak rate), AIR 为递增速率, r 为乘法递减因子, $P_{threshold}$ 是丢包率极限。算法如下:

(1) 发送方行为

- 1) 发送方开始发送数据, r 为输出速率, $r = IR$, 其中 $IR \geq MR$;
- 2) 发送方每发送 N_s 个RTP包就发送一个前向的RTCP控制包;
- 3) 发送方一收到反馈的RTCP包(其中携带了接收方检测到的丢包率 P_{loss}), 源端的输出速率开始按如下规则调整:

if($P_{loss} > P_{threshold}$)
 $r = \min\{(r + AIR), PR\}$
 else
 $r = \max\{(r \times r), MR\}$

(2) 接收方行为

- 1) 接收方记录接收到的RTP包的包头中序列号;
- 2) 每接收到 N_r 个包或者每隔至少5s, 接收方发送一个RTCP包给源端, 其中包含在这个时间间隔内观察到的丢包率 P_{loss} 。

该反馈控制算法调整视频编码器的输出速率 r 以维持丢包率 P_{loss} 在丢包极限 $P_{threshold}$ 之内。采用了AIMD控制算法, 当一个反馈的RTCP包指示没有拥塞的时候, 源端的发送速率加法增加。另外也有文献中采用MIMD的控制算法, 即: 没有拥塞时发送速率乘法增加。然而经验显示, 乘法增加速率调整算法与相对保守的加法增加速率调整算法相比, 通常带来源端速率的振荡, 从而引起更多的丢包。另一方面, 当发现返回的RTCP中 P_{loss} 大于 $P_{threshold}$ 时, 采用乘法递减能迅速降低发送速率, 从而缩短拥塞时间, 降低丢包率^[3]。

(3) 基于模型的拥塞控制

基于模型的拥塞控制不同于基于探索的拥塞控制: 基于

模型的需要明确地估计网络带宽, 而基于探索的是通过动态调整发送速率来对带宽进行探索。基于模型的是基于TCP连接的吞吐量模型来估计带宽的。TCP连接的吞吐量模型公式如下^[1]:

$$I = \frac{1.22 \times MTU}{RTT \times \sqrt{p}} \quad (1)$$

I TCP连接的吞吐量
 MTU 最大传输单元(连接中用的包的长度)
 RTT 传输往返时间
 P 包的平均丢失率

式(1)可以用于确定视频流的发送速率, 这样视频连接时的拥塞避免采用与TCP类似的方式, 实现与TCP流的公平竞争。也就是说, 具有前面提到的TCP友好性。

式(1)中用到的参数主要包括: RTT 和平均丢包率。

1) RTT 的测量

RTT 是反映网络状态一个重要参数, 它是TCP协议中用到的一个基本参数, 为了满足视频传输中的TCP友好性, 基于模型的拥塞控制方法也需要测量 RTT 。测量方法是利用RTCP中的SR或RR包中的时间戳, 在SR中的 $t_{rtt} = t_{now} - t_{delay} - t_{start}$, 其中 t_{now} 是发送方接收到RTCP反馈包的时间, t_{start} 是发送方发送RTCP包的时间, t_{delay} 是RTCP包在接收端的延时, 即: 接收端发送反馈RTCP包的时间减去接收端收到RTCP包的时间。

2) 平均丢包率的测量: 同前。

2.2 组播的拥塞控制策略

组播, 是为优化使用网络资源而产生的技术, 它涉及一个发送方和多个接收方, 如图2所示^[1]。

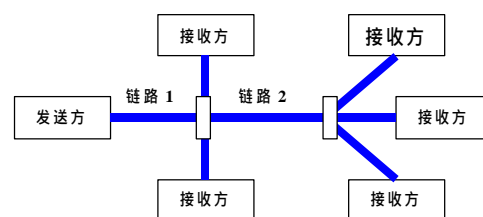


图2 组播

组播的基本思想是当一人要同时向一组人发送数据时, 他不必将数据向每一个接收者都发送一份, 而只需将数据发到一个组地址, 这样对于发送者而言, 数据只要发送一次就可, 所有加入该组的人都可以收到这份数据。无疑它减小了网络的负载和发送者的负担。目前, 组播已经得到了广泛的应用, 很多实时的多媒体会议及教育系统都是基于组播开发的。组播拥塞控制分为3种方式: 基于源端的, 基于接收端的和混合的。

(1) 基于源端的拥塞控制

由2.1节可以看出, 单播的拥塞控制基本都是采用基于源端的, 因为它都是根据接收方的反馈信息调整发送速率。同样, 组播基于源端的拥塞控制策略也是发送方根据接收方的反馈信息来调整发送速率。但是在组播的环境中, 由于有多个接收端, 各接收端的反馈信息是不相同的, 因此网络状态的判别较为复杂。

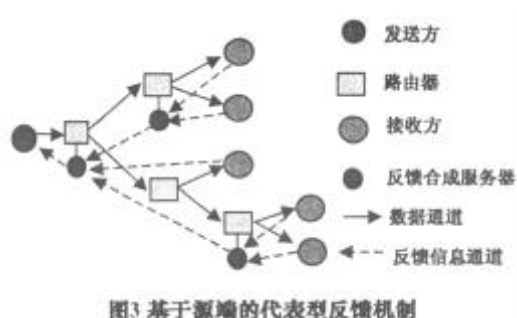
通常源端为了监控有多少视频包到达目的地, 应该从每个接收端得到信息。一种可能的方法是: 每个接收端一检测到丢包就发送一个否定应答NACK。然而, 在组播环境中, 可能有许多NACK包发过来, 另外, 大量的NACK包可能使网络更加拥塞, 这种问题称为“NACK爆炸问题”。解决这个问题的方法是采用一定的策略对反馈信息进行抑制或合

并,以减少接收方的反馈信息,同时仍能给源端提供足够的网络状态信息。这类策略主要有以下几种^[4-6]:

(1) 分布式策略:接收方周期性地向发送方反馈信息。根据组成员的数量确定反馈周期。这种方法适用于中小型的组规模。对于大规模的组,则由于反馈周期太长而使得反馈信息不能为发送方的调整策略提供参考。

(2) 抑制型策略:指如果一个接收方发现其他接收方已经反馈了与自己要反馈的信息类似的报告,那么就不反馈这个报告。

(3) 代表型策略:由于分布式的信息反馈不能很好地支持大的组播组,而抑制型的信息反馈方式往往只能获取部分接收方的信息而不是组的全部信息。因此很多算法力求通过中间节点来收集信息,从而为发送方提供一个“经过总结”的状态信息。一些文献中提出了组播路由树的中间路由器从组播的叶节点收集反馈信息,然后将这些信息总结成一个报告传送给组播树上的上一级路由器,如图3所示。这种方法需要改变网络节点的组播路由器的功能,因此,就目前来讲,这种方法并不现实。



(4) 抽样型:发送方从接收方中抽取反馈信息,这种方法是让接收方和发送方产生16位的随机钥匙,由接收方发送带有钥匙信息的控制信息,只有带有相同钥匙信息的接收方才被允许反馈信息。也有人采用概率方式在接收方生成令牌,只有得到令牌的接收方才被允许反馈信息。

在实际应用中,本文建议采用一种比较简单、实用的方法:在小的组播环境,接收者数量小于某个值(假定接收者数量小于10),接收者可以采用一检测到丢包就发送一个否定应答NACK的方式。在大的组播环境(接收者数量超过10时),接收者不是每次丢包都发送NACK,而是定期地发一个QoS监测值(T时间间隔内检测到的平均包丢失率,我们可以把T定义为接收到100个包的时间),根据RTP文档建议,我们也应确保每个接收者至少2分钟发一次反馈信息。我们注意到,在大多数情况下,接收方发反馈信息的频率将比每次丢包都发的频率低。因而,定期地发一个QoS监测值这种方法可以有效地减轻NACK爆炸问题^[2]。

具体实现是:每个接收方通过发RR(Receiver Report)包向源端发QoS监测值(反映平均包丢失率的数据),源端需将各个不同的QoS监测值转化为反映包是否很好到达接收端的一个“全局”值。一种方法是对每个接收端监测的值作平均得到总平均包丢失率(median_loss),然而,这种方法并不适合那种连接了不同带宽的网络,因为在一个组播树中,每个分支的丢包率差别都很大。另一种方法是对各个值作加权平均,可以根据重要程度设置权值,比较复杂。从实现的简单考虑可采用第一种方法^[2]。

这样在发送方就得到了一个反映网络状态的总平均包丢

失率值,同样,就可以采用单播中提到的基于探索和基于模型的拥塞控制算法来计算出源端的最大发送速率。

(2) 基于接收端的拥塞控制

基于接收端的拥塞控制通常针对发送视频是分层编码的,这样接收方可以通过接收不同的层来调整接收速率。

分层编码是把一个原始的视频序列压缩成一个基本层和多个增强层,基本层能被独立解码提供粗糙的视频质量,增强层能和基本层一起被解码,提供更好的视觉质量。接收端接收到的层越多,得到的视频质量越好。源端在完成分层编码后,将不同的层以不同的组播地址发送。接收端通过加入到不同的组来订购不同数量的层;而且,每一个接收端都力求能够在不引起拥塞的情况下订购尽可能多的组播组以提高视频质量。

与基于源端的拥塞控制类似,基于接收端的拥塞控制也分为两类:基于探索的和基于模型的。接收端基于探索的拥塞控制包括两部分^[1]:

1) 当一个接收端检测到网络没有拥塞出现时(通过判断自己的丢包率得到),接收端试着加入一个增强层,之后若仍然没有拥塞出现,表明这次加入是成功的;否则接收端放弃这次加入操作。当一次加入试验失败后,试验间隔延长一倍。

2) 当一个接收端检测到有拥塞时,丢掉一层以减少接收数据。

基于模型的拥塞控制与2.1节的类似,也是基于式(1)。

(3) 混合的拥塞控制

混合的拥塞控制是指发送端和接收端同时采用自己的拥塞控制策略。发送端可以根据接收端的反馈信息,采用基于源端的控制策略调整每个组播组的传输速率;而接收端可以通过采用基于接收端的控制策略,增加或去掉一层来调整接收的视频流的速率。混合拥塞控制的例子是DSO^[1](destination set grouping),源端的编码器输出多个不同质量和码率的码流,每个流被送到不同的IP组地址,接收端根据当前的网络状态和处理能力选择一个组播组加入,同时接收端将当前网络状态反馈给源端,源端还可以根据反馈信息来调整每个流的传输速率。

2.3 各种拥塞控制策略的比较

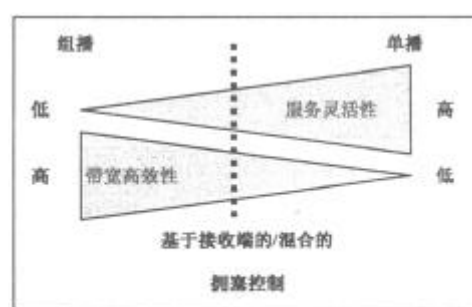


图4 带宽高效性与服务灵活性之间的权衡

2.1和2.2节介绍了视频传输中单播和组播的几种拥塞控制策略。可以看出,单播的拥塞控制方式比较简单,同时能够提供较好的服务灵活性,但是当用户较多时网络带宽的利用率很低。组播的视频传输中因为所有接收端共享一个通道,所以传输效率很高。在组播中,采用基于源端的拥塞控制针对不同接收端的连接带宽和应用需求不能提供各种灵活的服务,这样为了能够较好地平衡网络带宽的利用率和服务灵活性,提出了基于接收端的和混合的拥塞控制策略。图4

表明了单播、组播和网络带宽、服务灵活性的相互关系以及基于接收端的混合的拥塞控制在带宽高效性与服务灵活性之间的权衡^[1]。

3 结束语

在实际的具体应用中采用何种拥塞控制策略需要根据实际情况综合考虑。一般来说,单播适合采用基于源端的拥塞控制,组播适合采用基于接收端的或混合的拥塞控制,因为这样可以达到网络带宽的利用率和服务灵活性之间的平衡。最近基于源端的拥塞控制的许多研究焦点是基于模型的拥塞控制,认为在与TCP流竞争时能达到一定程度的公平性。然而,严格的基于模型的拥塞控制可能导致传输速率的骤减,并且可能得到不满意的视频质量。怎样在拥塞检测和反应的敏感性与对视频质量影响的平稳性之间权衡将有待于进一步的研究^[1]。

(上接第24页)

有效性,也就是智能化数据挖掘中问题识别层所做的工作。

(2) 辅助选择。辅助选择将数据挖掘过程转变满足求解条件的可操作过程,选择正确的数据分析功能。为了达到这一目标,辅助选择要完成3个主要任务:一是利用数据库表和实例信息确定属性和属性组;二是由单个的属性分类来确定属性组的分类;三是利用这一分类在数据分析包中确定功能。数据分析包的作用就在于能够分析属性组。

(3) 数据库交互平台。数据库交换能够输入数据库或数据仓库的数据用于数据分析。数据的输入有两种方式:(1)通过ODBC,(2)通过一种易于接受的数据库格式。既然数据库之间的数据交换目前已经为大众普遍了解和接受,如SQL,ODBA,ORACLE,RDA和JDBA等,数据库交换就不难实现。数据库交换与DBMS有关,也与源数据库和数据仓库有关。源数据库是指各种MIS数据库,在OLAP中是输入部分,数据仓库则是指数据挖掘和数据分析的主要数据库,汇集了来自关系数据库的各种数据。这样就可以保证数据的完整性和准确性,尤其是在互联网的环境中。

(4) 功能组合平台。功能组合平台可以调动数据分析包,通过运用数据,从知识库读取、解释包的输出结果来执行正确功能。为了与数据分析包进行高效的信息交换,功能组合平台要完成下面的任务:

1) 生成命令文件。命令文件(主要是ASCII文本)主要包括数据分析包使用的命令。

2) 准备数据文件。数据文件必须以数据分析包可识别的形式存在,数据文件的准备不仅包括以合适的文件格式输出文件,还要确保数据格式便于在模型库中进行分析。例如:若所有的实例都为数字型,许多数据分析包就不能对字符型数据进行分析。

3) 调用数据分析包。一旦执行了正确的文件,功能组合平台就在文件所在的目录下调用数据分析包。

4) 从数据分析输出文件中抽取相关结果。数据分析输出文件是半结构化的,功能组合平台拥有足够的关于统计包的知识才能运用知识库从输出数据中提取相关结果。

从上述功能中不难了解,中央处理单元是智能数据挖掘的核心,而功能组合平台则是智能数据挖掘成功的关键。中央处理单元主要通过agent来实现,即信息收集器收集信息,并将收集来的信息放入信息队列,而信息解释器取得信

参考文献

- 1 Wu Dapeng. Streaming Video over the Internet: Approaches and Directions. http://www.cs.cmu.edu/~dpwu/mypapers/streamingVideo_camera.pdf, 2001
- 2 Bolot J C, Turetti T. Experience with Control Mechanisms for Packet Video in the Internet. <http://citeseer.nj.nec.com/211830.html>, 1998
- 3 Wu Dapeng. A Framework of End-to-end Transport Architecture for Streaming MPEG-4 Video over the Internet. <http://citeseer.nj.nec.com/wu00endoend.html>, 2000
- 4 陈丹. 基于精细分层编码的视频通信技术研究. <http://www.cnki.net/>, 2003
- 5 Lee K W, Ha S, Bharghavan V. IRMA: A Reliable Multicast Architecture for the Internet. <http://citeseer.nj.nec.com/lee99irma.html>, 1999
- 6 Gao Yuan, Hou J C. RACOOM: A Rate Based Congestion Control Scheme for Multicast. <http://citeseer.nj.nec.com/217071.html>, 1999
- 7 刘俊. 组播中的拥塞控制技术探讨. 计算机应用研究, 2003, (1)

息队列中的信息,对之进行解释,然后根据自身状态调用合适的方法,执行相应的动作。功能组合平台成功与否取决于数据分析包、模型库和知识库三者之间的协调好坏。功能组合平台通过协调agent来实现,协调模块接受中央处理单元或辅助选择的请求,然后按照一定的规则检索知识库、模型库,提供一系列的知识资源或解决方案以供选择,在给定选择结果后,协调agent通过通信模块和数据库交互平台调用相关数据,并提交数据分析包求解,最后将结果返回中央处理单元。我们将中央处理单元、辅助选择、数据库交互平台和功能组合平台4个部分作为一个智能中间件。

4 结论

本文在层次模型的基础上提出了智能化数据挖掘的开发框架,并进一步提出了智能化DM模型。在这一框架和模型中,还需要有一个人工智能化的学习技术,以保证智能化DM系统能不断改进,更好地适应外界实体环境的不断变化。我们希望能更好地集成数据分析软件,进一步研究如何开发出一个更有用的能够不断适应新情况的智能中间件。

参考文献

- 1 陈晓红. 决策支持系统理论 and 应用[M]. 北京:清华大学出版社, 2000
- 2 Chen Xiaohong. DSS as an Application of General Systems Theory [J]. International Journal of General Systems, 1996, 25(2)
- 3 Cecil Eng Huang Chua, Roger H L Chiang, Ee-Peng Lim. An Intelligent Middleware for Linear Correlation Discovery [J]. Decision Support Systems, 2002, 32: 313-326
- 4 Shaw M J, Subramaniam C, Tan G W, et al. Knowledge Management and Data Mining for Marketing [J]. Decision Support Systems, 2001, 31: 127-137
- 5 Yasuhiko Takahara, Naoki Shiba, Liu Yongmei. General System Theoretic Approach to Data Mining System [J]. International Journal of General Systems, 2000, 31 (3): 245-264
- 6 李存华, 纪兆辉. 基于互联网络的决策支持系统模型 [J]. 计算机工程, 2000, 26(10): 48-50
- 7 高洪深. 决策支持系统(DSS): 理论·方法·案例[M]. 北京:清华大学出版社, 2000