一种基于 TCP 的实时视频传输延时预测模型 *

熊永华[1], 吴 敏[1], 贾维嘉[2]
(1. 中南大学 信息科学与工程学院, 长沙 410083; 2. 香港城市大学 计算机科学系, 香港 999077)

摘 要: 针对 TCP 重传易导致延时抖动大、降低流媒体播放质量的问题,提出了使用 TCP 传输实时视频时,播放质量不受 TCP 重传影响的必要条件,即视频帧发送延时、播放缓冲区延时与网络往返时间需要满足一定的关系式。建立了视频帧发送延时的预测模型,该模型通过输入视频帧长度、丢包率、网络往返时间、TCP 拥塞窗口大小与 TCP 超时重传时间,预测视频帧的发送延时。NS2 模拟结果表明,可以通过发送延时的预测值来判断视频帧是否适合采用 TCP 传输。

关键词: 传输控制协议; 实时视频; 延时; 预测
中图分类号: TP393 **文献标志码**: A **文章编号**: 1001-3695(2010)06-2239-04
doi:10.3969/j.issn.1001-3695.2010.06.069

TCP-based real-time video transmission delay prediction model

XIONG Yong-hua[1], WU Min[1], JIA Wei-jia[2]
(1. *School of Information Science & Engineering, Central South University, Changsha* 410083, *China*; 2. *Dept. of Computer Science, City University of Hong Kong, Hong Kong* 999077, *China*)

Abstract: Aimed to larger end-to-end delays and jitters and worse playing quality posed by the TCP retransmission for the multimedia streaming transmitting via Internet using TCP, this paper proposed the necessary conditions to overcome effect on video playing quality caused by TCP retransmission, such as the formula describing the relationship between video frame sending-delays, play out buffer delays and round trip time (RTT). It presented a prediction model to predict the video frame sending-delays with input parameters as video frame lengths, network loss ratio, RTT, TCP congestion window size, and TCP retransmission timeout (RTO). Experiments using NS2 simulator show that the prediction value of video frame sending-delays can be used to determine whether it is feasible for a video frame to be transsmitted using TCP protocol.

Key words: TCP(transmission control protocol); real-time video; delay; prediction

TCP 是当前 Internet 上使用最为广泛的传输层协议,其最主要的特点是面向连接,采用加法增加乘法减少(AIMD)机制进行拥塞控制,采用滑动窗口协议进行流量控制,采用请求重传机制进行差错控制[1],这些使得 TCP 对于网络拥塞反应迅速、具有 TCP 友好性、容易被大多数防火墙所接受,但也导致了传输的吞吐率波动、端到端延时及抖动均大于同等条件下的 UDP 传输。

基于此,有研究人员认为 TCP 不适用于对于延时、抖动有严格要求的实时视频传输[2]。但由于 UDP 自身的简单、无连接和不可靠性,使得基于 UDP 的实时视频传输还需要重新设计拥塞控制、流量控制与差错控制机制,同时也需要考虑视频流的 TCP 友好性与防火墙穿透能力等,这些都增加了传输机制设计与维护的难度,并减少了其通用性能,商用的 Internet 实时视频传输系统,如 Skype、Realplayer 和 QQ 等,大都摈弃了设计维护较为复杂的 UDP 方式,而采用基于 TCP 的方式。

TCP 的优点及其在 Internet 与商业领域的广泛应用,也激励着研究者对 TCP 实时视频传输的研究与开发。文献[3]提出了一个接收方驱动的 TCP 连接带宽分配系统;文献[4]提出了一种扩展的 TCP 以支持实时流媒体应用;文献[5]设计了一种基于单个流多个 TCP 连接的实时多媒体传输系统;文献[6]提出了一个随机模型用于评估 TCP 实时与存储视频的传输性能;文献[7]提出了播放缓冲区自适应算法,用于随机丢包环境的无线网络视频流传输;文献[8,9]通过设置视频帧的优先级来控制帧率;文献[10]设计了一个具备联合缓冲区管理与拥塞控制机制的视频传输系统;文献[11]设计了一个基于多缓冲区调度算法的帧率自适应传输策略。

本文针对 Internet 上的 TCP 实时视频双向传输过程,分析推导了采用 TCP 传输实时视频的可行性与必要条件,设计了一种实时视频帧的发送延时预测模型,通过发送延时的预测值来判断视频帧是否适合采用 TCP 传输。

1 TCP 实时视频传输的局限性

实时流媒体传输在接收方通常需要设置播放缓冲区用于消除 IP 包的延时抖动,即接收方在 IP 包到达以后不是立即播放,而是将其放入播放缓冲区中等待一个时间段 D_{playout} 以后再进行播放,从而通过增加流媒体整体的端到端延时,扩大了单个包的延时抖动范围。文献[4]推导了 TCP 播放缓冲区延时 D_{playout} 与媒体包延时的关系为

收稿日期: 2009-11-19; **修回日期**: 2009-12-22 **基金项目**: 国家杰出青年科学基金资助项目(60425310);香港城市大学战略发展计划资助项目(7002102,7002214)

作者简介: 熊永华(1979-),男,湖北洪湖人,讲师,博士,主要研究方向为多媒体通信、无线网络(yhxiong@mail.csu.edu.cn);吴敏(1963-),男,广东化州人,教授,博士,主要研究方向为过程控制、鲁棒控制和智能系统;贾维嘉(1957-),男,贵州贵阳人,教授,博士,主要研究方向为多媒体通信、移动计算和分布式系统.

$$D_{\text{playout}} \geq \frac{3}{2}\text{RTT} + 3\text{period} + \text{delta} \tag{1}$$

其中：D_{playout}为播放缓冲区延时；RTT(round-trip time)为 TCP 的网络往返时间；period 为两个报文段的发送间隔时间，等于流媒体的采样间隔；delta 为补偿时间，如缓冲区的存取耗时等。在报文段丢失后，TCP 通过收到三次重复 ACK(acknowledgement)的快重传机制对丢失的报文段进行重传，如果 D_{playout}满足式(1)，则表明报文段的重传所造成的延时能被播放缓冲区所消除，即重传的媒体报文段能够在接收端按时播放，此时因 TCP 重传所造成的延时对于接收端而言是透明的，因而说明了 TCP 实时流媒体传输在满足式(1)的条件下是可行的。

文献[4]只考虑了一个视频帧只通过一个 TCP 报文段(segment)进行传输的理想情况，而实际传输过程中一个经过编码压缩后的实时视频帧往往大于 TCP 最大报文段(maximum segment size, MSS)，因此 TCP 需要将视频帧分为多个报文段进行发送，而当报文段数目大于 TCP 当前发送窗口大小时，视频帧需要经过多个 RTT 周期才能完成发送，如图 1 所示。

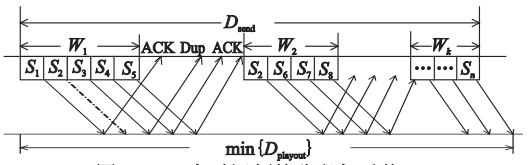

图1　TCP实时视频的分段与重传

设一个视频帧可以被分为 n 个 TCP 报文段(从 S_1 到 S_n)，在第 $i(1 \leq i \leq n)$个 RTT 开始时 TCP 发送窗口大小为 W_i 个报文段，则在第 i 个 RTT 内，发送端最多可以发送 W_i 个报文段，当 $W_i < n$ 时视频帧需要多个 RTT 才能完成发送。在出现报文段丢失时(如图 1 中 S_2 丢失)，根据 TCP Reno 与 TCP SACK 补充机制[12]，发送方收到三个以上重复 ACK(Dup ACK)后，TCP 将重传丢失的报文段(可能有多个)，同时发送数目为当前发送窗口大小一半的新的报文段，即第 i 个 RTT 内丢失的报文段可以在第 $i+1$ 个 RTT 内和新的报文段一起发送。因此设在第 k 个 RTT 时开始发送视频帧的最后一部分报文段(此时窗口大小为 W_k)，则有

$$\sum_{i=1}^{k} W_i \geq n \tag{2}$$

定义 D_{send}为视频帧的第一个报文段从 TCP 发送缓冲区进入网络，到其最后一个报文段进入网络的时间间隔(最后一个报文段可能是最末尾的一个报文段，也有可能是因丢包而重传的报文段)。当最后一个发送窗口无丢包时，$D_{\text{send}} = (k-1) \cdot \text{RTT}$；而当其有丢包时，由于需要一个额外的 RTT 时间来重传丢失的报文，$D_{\text{send}} = k \cdot \text{RTT}$。

根据文献[4]和式(1)的思想：如果 TCP 的播放缓冲区能够消除因为 TCP 重传而造成的延时抖动，则这时采用 TCP 传输实时流媒体才是可行的。而由图 1 可知，当视频帧的大小超过一个 TCP 最大报文段时，该视频帧需要多个 RTT 周期才能完成发送，此时的 D_{send}包括了在这些 RTT 周期中 TCP 重传所造成的延时抖动，因此根据式(1)的思想，此时的 TCP 播放缓冲区需满足

$$D_{\text{playout}} \geq D_{\text{send}} + \frac{1}{2}\text{RTT} \tag{3}$$

才能消除因为 TCP 重传所带来的延时抖动，由式(3)可得

$$D_{\text{send}} \leq D_{\text{playout}} - \frac{1}{2}\text{RTT} \tag{4}$$

当 D_{playout}为常量时，一个视频帧的发送延时需满足式(4)，否则该视频帧将错过其播放时间，造成严重的延时抖动，降低视频播放质量，即 TCP 实时视频的传输在满足式(4)时是可行的。

2　视频帧发送延时的特性

定义 $D_{\text{frame}}(i)$为第 i 个视频帧从采样生成，到接收端开始播放之间的端到端延时；$D_{\text{wait}}(i)$为第 i 个视频帧进入 TCP 发送缓冲区，到该视频帧的第一个报文段开始被发送的时间间隔；$D_{\text{network}}(i)$为第 i 个视频帧从发送端网络层到接收端网络层的传播延时。为便于研究，可认为 $D_{\text{network}}(i) = 0.5\text{RTT}$，则根据文献[11]，有

$$D_{\text{frame}}(i) = D_{\text{wait}}(i) + D_{\text{send}}(i) + C \tag{5}$$

其中：C 为常量。设 P 为视频帧的采样间隔时间，令

$$A(i-1) = D_{\text{wait}}(i-1) + D_{\text{send}}(i-1) - P \tag{6}$$

则当 $A(i-1) > 0$ 时，第 $i-1$ 个视频帧尚未全部从 TCP 发送缓冲区进入网络，而后续的第 i 个视频帧已经生成并进入了 TCP 发送缓冲区，因此有 $D_{\text{wait}}(i) = A(i-1)$；反之，若 $A(i-1) \leq 0$，则第 i 个视频帧生成时，第 $i-1$ 个视频帧已经完成发送，此时有 $D_{\text{wait}}(i) = 0$。因此有

$$D_{\text{wait}}(i) = \begin{cases} 0 & A(i-1) \leq 0 \\ A(i-1) & A(i-1) > 0 \end{cases} \tag{7}$$

令 $D_{\text{wait}}(1) = 0$，将式(6)代入式(7)并迭代，得到

$$D_{\text{airt}}(i) \geq \max\left\{0, \sum_{j=1}^{i-1} D_{\text{send}}(j) - (i-1) \cdot P\right\} \tag{8}$$

将式(8)代入式(5)有

$$D_{\text{frame}}(i) \geq \max\left\{0, \sum_{j=1}^{i-1} D_{\text{send}}(j) - (i-1) \cdot P\right\} + D_{\text{send}}(i) + C \tag{9}$$

可见，使用 TCP 传输实时视频时，视频帧的发送延时不仅决定了该视频帧自身能否在接收端按时播放，而且也将影响其后序视频帧的端到端延时。

发送延时 D_{send}的大小取决于 RTT 以及 TCP 拥塞窗口的大小及变化规律，这些是由 RTT、丢包率与 TCP 的拥塞控制、流量控制与差错控制机制共同决定的。因此，在不修改 TCP 的前提下无法对发送延时进行控制，但是可以利用影响发送延时的相关参数对发送延时进行预测。当预测的视频帧发送延时满足式(4)时，可认为该帧能正常播放，反之该帧将具有不合要求的端到端延时，并且还将加大其后续帧的端到端延时，不适合采用 TCP 传输。由此可见，视频帧发送延时的预测值能作为制定 TCP 视频传输机制的参考因素。

3　视频帧发送延时预测模型

3.1　假设条件

本文使用 TCP Reno 及其 SACK 补充版本，这是当前所有 Windows 操作系统默认的以及 Internet 上使用最为广泛的 TCP 版本，并作如下假设：

a)TCP 发送缓冲区的大小能够容纳多个连续的视频帧，TCP 的接收缓冲区能够及时将所收到的数据提交给应用层，而不会造成视频帧在 TCP 接收缓冲区溢出。TCP 发送与接收缓冲区大小主要由端主机的性能决定，因此该假设条件易满足。

b)不同 RTT 内的丢包事件相互独立，而同一个 RTT 内，若发送窗口有报文丢失，则该窗口中自该报文以后的报文也全部丢失。由于 TCP SACK 机制有能力一次重传同一窗口中所有丢失的报文，该假设不会影响报文的传输与重传时间。

c)根据 TCP，其发送窗口大小等于拥塞窗口(congestion

window)和发送方广告窗口(advertise window)的最小值,因此假设接收方广告窗口足够大(广告窗口大小由主机性能决定),则发送方窗口大小等于拥塞窗口大小。

d)假设 TCP 连接已经建立,并且关闭了 Nagel 算法,且不会产生糊涂窗口综合症。由于是双向的视频流,采用捎带(piggyback)ACK 方式,不适用延时 ACK。

根据上述假设,本文忽略了 TCP 的三次握手阶段和延时 ACK 补偿阶段,认为一个 TCP 报文段可处于慢开始、重传和拥塞避免三种阶段。定义三种阶段的发送延时分别为 D_{slow}、D_{re} 和 D_{con},而报文段处于三种阶段的概率分别为 P_{slow}、P_{re} 和 P_{con},则有期望的视频帧发送延时为

$$E[D_{\text{send}}]=P_{\text{slow}}D_{\text{slow}}+P_{\text{re}}D_{\text{re}}+P_{\text{con}}D_{\text{con}}=E[D_{\text{slow}}]+E[D_{\text{re}}]+E[D_{\text{con}}] \tag{10}$$

3.2　慢开始阶段的发送延时

设视频帧的长度为 S 个 TCP MSS,S_{slow} 为慢开始阶段结束以前已成功发送的报文段数目(即在 D_{slow} 内发送的报文段数目,不包括丢失的报文段)。设网络丢包率为 $p(p\geqslant 0)$,当 $p=0$ 时,可以期望在 D_{slow} 内发送完所有的 S 个报文段,即 $E[S_{\text{slow}}]=S$;当 $p>0$ 时,假设包丢失与包发送事件相互独立,则有离散随机变量 S_{slow} 的概率分布为 $P\{S_{\text{slow}}=k\mid k\in(0,S)\}=(1-p)^k\cdot p$因此有

$$E[S_{\text{slow}}]=\sum_{k=0}^{S-1}[(1-p)^k\cdot p\cdot k]+(1-p)^s\cdot S,p>0 \tag{11}$$

在慢开始阶段,接收方每收到一个 ACK 就将拥塞窗口大小增加一个报文段,设第 i 个 RTT 周期 TCP 拥塞窗口大小为 W_i,则有 $W_i=2\cdot W_{i-1}$。若在第 i 个 RTT 内能发送完 S_{slow},则有 $S_{\text{slow}}=W_1+2W_1+\cdots+2^{i-1}W_1=(2^i-1)\cdot W_1$,从而 $i=\log_2(S_{\text{slow}}/W_1+1)$,其中 W_1 为第一个 RTT,即视频帧开始被传输时的 TCP 拥塞窗口的大小,因此有

$$E[D_{\text{slow}}]=E[i\cdot \text{RTT}]=\text{RTT}\cdot\log_2(E[S_{\text{slow}}]/W_1+1) \tag{12}$$

其中:$E[S_{\text{slow}}]$取值如式(11)。设此时拥塞窗口大小为 W_{slow},则有 $W_{\text{slow}}=2^{i-1}W_1=(S_{\text{slow}}+W_1)/2$,从而得到

$$E[W_{\text{slow}}]=(E[S_{\text{slow}}]+W_1)/2 \tag{13}$$

3.3　重传阶段的发送延时

传输完 S 个报文段以后没有出现丢包现象的概率为$(1-p)^S$,因此在传输 S 个报文期间至少出现一个丢包现象,即 TCP 进入重传阶段的概率为 $1-(1-p)^S$。

TCP 通过两种方式判断是否进入重传阶段:收到三个以上的重复 ACK(Dup ACK)或者出现重传计时器超时(retransmission timeout, RTO)。文献[13]推导了 TCP 因为 RTO 而进入拥塞避免阶段的概率 $Q(p,W)$ 为

$$\min\left\{1,\frac{1+(1-p)^3\cdot(1-(1-p)^{W-3})}{(1-(1-p)^W)/(1-(1-p)^3)}\right\} \tag{14}$$

其中:p 为丢包率,W 为拥塞避免阶段开始时 TCP 拥塞窗口的大小。因此,TCP 因为收到三次重复 ACK 而进入拥塞避免的概率为 $1-Q(p,W)$。

定义 Z^{TO} 为从 RTO 恢复所需时间,Z^{TD} 为从重复 ACK 机制恢复所需时间,根据文献[13]有:

$$E[Z^{TO}]=G(p)\cdot T_0/(1-p) \tag{15}$$

其中:T_0 为首次出现 RTO 时重传计时器所设置的时间间隔,$G(p)$为

$$G(p)=1+p+2p^2+4p^3+8p^4+16p^5+32p^6 \tag{16}$$

$E[Z^{TD}]$的取值取决于同一发送窗口内丢包的数目及 TCP 的版本,本文针对 TCP Reno 及 SACK 选项,可以在一个 RTT 内重传一个窗口内所有丢失的报文,因此有 $E[Z^{TD}]=\text{RTT}$,从而

$$E[D_{\text{re}}]=(1-(1-p)^S)\cdot(E[Z^{TO}]\cdot Q(p,E[W_{\text{Slow}}])+\text{RTT}\cdot(1-Q(p,E[W_{\text{slow}}]))) \tag{17}$$

其中:p 为网络丢包率,S 为视频帧的长度,$E[Z^{TO}]$可由式(15)求得,$E[W_{\text{slow}}]$可由式(13)求得,p 和 $E[W_{\text{slow}}]$作为式(14)的输入参数。

在重传阶段,若是因 RTO 重传,则并未发送新的报文段,即 $S_{\text{RTO}}=0$;若是因三次重复 ACK 重传,则有窗口大小一半的新数据段被发送,即 $S_{TD}=0.5W_{\text{slow}}$,因此重传阶段期望发送的报文段长度

$$E[S_{\text{re}}]=(1-Q(p,E[W_{\text{slow}}]))\cdot 0.5E[W_{\text{slow}}] \tag{18}$$

3.4　拥塞避免阶段的发送延时

在对丢失报文重传以后,TCP 进入拥塞避免阶段,所发送的剩余报文段长度为

$$E[S_{\text{con}}]=S-E[S_{\text{slow}}]-E[S_{\text{re}}] \tag{19}$$

根据文献[13]所建立的拥塞避免阶段 TCP 拥塞窗口大小的模型,在开始发送剩余报文段时,TCP 拥塞窗口的期望值

$$E[W_{\text{con}}]=\frac{2}{3}+\sqrt{\frac{8(1-p)}{3P}+1} \tag{20}$$

利用文献[13]提出的 TCP 拥塞避免阶段的平均吞吐率模型,得到拥塞避免阶段吞吐率 B 为

$$\frac{\frac{1-p}{p}+\frac{E[W_{\text{con}}]}{2}+Q(p,E[W_{\text{con}}])}{\text{RTT}\cdot\left(\frac{1}{2}E[W_{\text{con}}]+1\right)+\frac{T_0\cdot Q(p,E[W_{\text{con}}])\cdot G(p)}{1-p}} \tag{21}$$

因此,有拥塞避免阶段的期望发送延时

$$E[D_{\text{con}}]=E[S_{\text{con}}]/B \tag{22}$$

将式(12)(17)(22)代入式(10)得到 $E[D_{\text{send}}]$的值为 p、S、W_1、RTT 和 T_0 的函数,记为 $E[D_{\text{send}}]=f(p,S,W_1,\text{RTT},T_0)$。

4　模拟实验与性能分析

4.1　模拟环境描述

在 NS2 下对发送延时性能与预测模型进行验证与分析,所采用的 NS2 模拟环境拓扑结构如图 2 所示。

TCP 发送端与接收端分别与路由器 R1 和 R2 连接,设置路由器 R1 队列长度为 50,链路最小 RTT 为 60 ms,MSS = 1 000 Byte。发送端输入一个自定义的视频流量(video traffic),该流量的生成来自于一个基于 H. 263 标准的视频追踪(video trace)文件。Video trace 文件由一段 H. 263 编码的视频生成,主要记录了视频帧的大小及其发送时间,如表 1 所示。

表 1　实时视频追踪文件格式

参数	帧编号			
	1	2	…	100
发送时间/s	0.2	0.4	…	20
帧大小/Byte	12 539	6 604	…	8 073

所使用的视频文件长度为 20 s,采用基于 H. 263 可变比特率编码标准压缩成 100 个大小不同的视频帧,帧间间隔为 200 ms。

4.2　模拟结果及性能分析

发送延时预测的目的在于利用预测值来判断实时视频帧是否适合采用 TCP 传输,据此制定相应的 TCP 传输策略。根据 $E[D_{\text{send}}]=f(p,S,W_1,\text{RTT},T_0)$,预测模型输出量为视频帧发送延时的期望值,输入量为丢包率 p,视频帧长度 S,发送时

刻 TCP 拥塞窗口大小 W_1、网络往返时间 RTT 和超时计时器 T_0。设 RTT = 100 ms,取 T_0 = 4RTT,采用同一段视频文件,并取 $D_{playout}$ = 400 ms,当路由器 R1 为尾部丢包策略(drop tail)时,设置 p = 0 和 p = 1.8%,当 R1 为随机早期丢包策略(random early detection, RED)时,设置 p = 2.1%,得到模型输入参数与预测结果分别如图 3 和 4 所示。

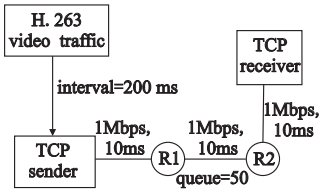

图2 NS2模拟环境的拓扑结构

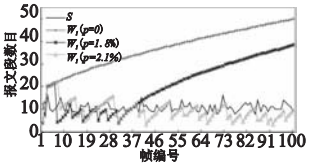

图3 不同丢包策略与丢包率下的输入参数

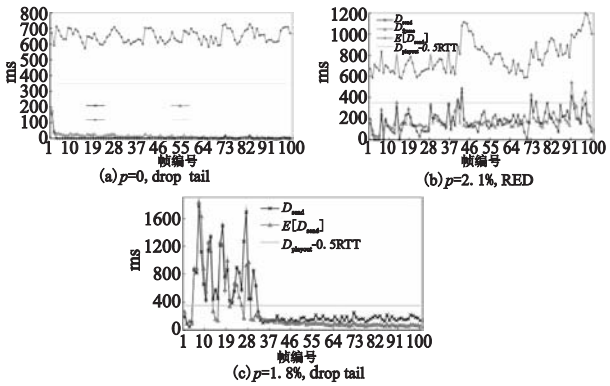

图4 不同丢包策略与丢包率下的预测结果

在图 3 中,当 p = 0 时,拥塞窗口 W_1 在第二帧以后开始超过帧大小 S,并且持续增加;当 p = 1.8% 时,由于路由器是采用 drop tail 策略,数据包的丢失具有爆发性(burst),丢包现象集中出现在第 7 ~ 27 帧间,且在第 7 帧时出现了 RTO,使得窗口大小直接跌到最小值,第 31 帧之后拥塞窗口开始线性增长,并在第 40 帧以后超过帧大小;当 p = 2.1% 时,路由器基于 RED 策略,丢失的数据包较为分散,且均采用快重传、快恢复机制,没有出现 RTO,从而使得拥塞窗口呈现锯齿状波动。

图 4(a)是 p = 0,drop tail 策略的情形。此时 D_{frame} 的值最大为 726 ms,平均值为 652 ms,所有视频帧都满足 Internet 实时视频的传输要求,这说明了式(4)必要条件的合理性。根据式(4),通过实际值 D_{send} 判断所有的视频帧都适合采用 TCP 传输,而通过预测值 $E[D_{send}]$ 也判断所有的视频帧都适合 TCP 传输,因此,预测判断的漏判率为 0,准确程度为 100%。

图 4(b)是 p = 1.8%,RED 策略的情形。D_{send} 的变化规律与 D_{frame} 基本一致,通过 D_{send} 判断为不适合的帧,其 D_{frame} 也较大,符合式(4)的必要条件。预测值对实际值波峰的跟随性较好,通过实际值可以判断在 100 个视频帧中不符合必要条件的视频帧有 5 个,编号集为 actual(40,42,73,75,91);通过预测值判断不符合条件的视频帧有 7 个,编号集为 prediction(13,40,42,73,91,96,97),其中 4 个帧(40,42,73,91)包含在 actual 中。因此采用预测判断时漏掉了 1 个不合要求的帧,而将 3 个符合要求的帧判断为不合要求,即预测判断的漏判率为 1%,准确程度为 96%。

图 4(c)是 p = 2.1%,drop tail 策略的情形。Actual 集有 19 帧,prediction 集有 26 帧,其中 18 帧在 actual 中,因此预测判断漏判率为 1%,准确程度为 92%。

5 结束语

本文基于 TCP Reno 和 TCP SACK 版本,通过分析 TCP 实时视频传输的可行性与局限性,提出了如果视频播放质量不受网络丢包及 TCP 重传影响,则视频帧发送延时所必须满足的约束条件。为了能在视频帧被发送以前就判断它的发送延时是否满足该约束条件,即判断该视频帧是否适合采用 TCP 传输,建立了一个视频帧发送延时预测模型,通过判断发送延时的预测值是否满足约束条件,来决定视频帧能否采用 TCP 传输。使用 NS2 模拟平台,在不同路由器丢包策略和丢包率环境下的实验表明,使用模型预测值进行判断,漏判率为 0% ~ 1%,准确程度为 92% ~ 100%,因此,可以通过发送延时的预测值来判断视频帧是否适合采用 TCP 传输。

笔者下一步将研究如何利用发送延时预测模型来制定新的 TCP 实时视频传输策略,如利用发送延时的预测值进行选择丢帧、帧率调整、播放缓冲区自适应调整等。

参考文献:

[1] 熊永华, 吴敏, 贾维嘉. 实时流媒体传输技术研究综述[J]. 计算机应用研究, 2009,26(10): 3615-3620.

[2] FLOYD S, HANDLEY M, PADHYE J, *et al*. Equation based congestion control for unicast applications[C]//Proc of ACM SIGCOMM-Mk. 2000: 43-56.

[3] MEHRA P, De VLEESCHOUWER C, ZAKHOR A. Receiver-driven bandwidth sharing for TCP and its application to video streaming[J]. IEEE Trans on Multimedia,2005,7(4): 740-752.

[4] LIANG S, CHERITON D. TCP-RTM: using TCP for real-time multimedia applications[C]//Proc of International Conference on Network Protocols(ICNP). 2002: 1-20.

[5] NGUYEN T, CHEUNG S. Multimedia streaming using multiple TCP connections[C]//Proc of the 24th IEEE International Conference on Performance, Computing, and Communications. 2005: 215-223.

[6] BOHACEK S. A stochastic model of TCP and fair video transmission [C]//Proc of INFOCOM. 2003: 1134-1144.

[7] ANTONIOS A. Improving the performance of TCP wireless video streaming with a novel playback adaptation algorithm[C]//Proc of IEEE International Conference on Multimedia and Expo. 2006: 1169-1172.

[8] SEELAM N, SETHI P, FENG W C. A hysteresis based approach for quality, frame rate, and buffer management for video streaming using TCP[C]//Proc of the Management of Multimedia Networks and Services Conference. 2001: 1-15.

[9] KRASIC C, WALPOLE J. Priority-progress streaming for quality-adaptive multimedia[C]//Proc of ACM Multimedia Doctoral Symposium. 2001: 463-464.

[10] BAJIC I V, TICKOO O, BALAN A, *et al*. Integrated end-to-end buffer management and congestion control for scalable video communications[C]//Proc of International Conference on Image Processing. 2003: 14-17.

[11] XIONG Yong-hua, WU Min, JIA Wei-jia. Efficient frame schedule scheme for real-time video transmission across the Internet using TCP [J]. Journal of Networks, 2009, 4(3): 216-223.

[12] FLOYD S, MAHDAVI J, MATHIS M, *et al*. IETF RFC 2883, An extension to the selective acknowledgement (SACK) option for TCP [S]. 2000.

[13] PADHYE J, FIROIU V, TOWSLEY D, *et al*. Modeling TCP Reno performance: a simple model and its empirical validation[J]. IEEE/ACM Trans on Networking, 2000, 8(2): 133-145.