

基于带宽管理的实时流媒体传输策略

许 林, 白光伟

(南京工业大学信息科学与工程学院, 南京 210009)

摘 要: 流控制传输协议(SCTP)是基于不可靠的、无连接的分组 IP 网络上的一种面向消息的、可靠的传输层协议, 它更适合在较高的网络带宽环境下的应用。而作为一种部分可靠版本的 PRSCTP 在带宽受限的网络环境下有更好的性能表现。该文提出一种适合实时流媒体网络应用的方案 BMSCTP, 它通过带宽估计机制, 实现自适应地在 SCTP 和 PRSCTP 之间进行切换。出于性能分析和评估的需要, 设计了一组仿真实验。实验结果显示, 动态网络环境下传输实时流媒体时, 在投递率和传输延迟等性能指标方面, BMSCTP 要优于 SCTP 和 PRSCTP。
关键词: 流控制传输协议; 带宽管理; 流媒体

Bandwidth Management Based Transmission Strategy for Real-time Streaming Media

XU Lin, BAI Guang-wei

(College of Information Science and Engineering, Nanjing University of Technology, Nanjing 210009)

【Abstract】 Stream Control Transmission Protocol(SCTP) is a message-oriented, reliable transport protocol operating on top of a potentially unreliable connectionless packet service such as IP. However, it requires high network bandwidth. As a partially reliable version of SCTP, PRSCTP achieves better performance in limited bandwidth network environments. This paper explores a new strategy Bandwidth Management based SCTP(BMSCTP) which tries to switch between SCTP and PRSCTP adaptively by using a bandwidth estimation mechanism. For performance analysis and evaluation, a series of simulation experiments are designed and conducted in this work. Simulation results demonstrate that the BMSCTP may provide significantly advantageous over the SCTP and PRSCTP for real-time multimedia applications in dynamic network environments, in terms of delivery ratio, end-to-end delay, etc.

【Key words】 Stream Control Transmission Protocol(SCTP); bandwidth management; streaming media

流媒体是现今最流行的网络应用之一。然而无线网络自身所具有的信道带宽有限、误码率较高和信道质量不稳定等特性,使得在无线网络环境下为实时流媒体应用提供 QoS 保证面临新的、严峻的挑战。无线流媒体的应用需要一套比有线网络更可靠的机制来保证数据的传输,其中最重要的就是需要一种兼具 UDP 实时性和 TCP 可靠性的新型传输层协议。本文在分析研究新型传输层协议 SCTP 及其扩展版本 PRSCTP 基础上,提出一种适合在无线网络环境下传输流媒体的方案。

1 研究背景

传统的 Internet 传输层协议主要包括 TCP 和 UDP。TCP 协议是一种面向连接的传输协议,提供端到端可靠的、有序的数据传输机制。但 TCP 协议采用的流量控制和拥塞控制机制容易引起流媒体过大的速率波动,其报文的逐个确认、超时重发机制也会带来较大的传输时延;UDP 协议提供无连接、不可靠的数据报服务,而且不具备拥塞机制的 UDP 流在网络环境中将大量抢占信道资源,带来系统拥塞崩溃的潜在危险,故这 2 种协议都不能很好地满足流媒体传输的要求。SCTP 作为一种新型完全可靠的传输层协议,它兼具 UDP 和 TCP 的一些优点,可以用来传输流媒体^[1]。而部分可靠的流控制传输协议 PRSCTP 对重要数据采用与 SCTP 一样的可靠传输,对一般数据采用不可靠的传输,可有效缓解带宽因素造成的对流媒体传输的影响^[2]。

1.1 流控制传输协议

流控制传输协议(Stream Control Transmission Protocol, SCTP)是面向连接、端到端、全双工、带有流量和拥塞控制的可靠传输层协议,并于 2000 年 10 月成为 IETF 的标准(RFC2960)^[3]。SCTP 能实现 TCP 及其各种主流改进版本的所有功能,并且弥补了 TCP、UDP 的一些不足,同时还具备一些诸如支持多路径传输、支持多流等许多更适宜数据传输的新特点。所以,它在很多方面开始得到较为广泛的应用。为此专门做了多组实验,论证用 SCTP 替代 UDP 传输流媒体的可行性。

1.2 部分可靠的流控制传输协议

部分可靠的流控制传输协议(PRSCTP)是 IETF 于 2004 年制定的传输层协议标准(RFC3758)^[4]。PRSCTP 与标准版本的 SCTP 一样也支持多流传输,而 PRSCTP 还允许为每个流定义重传次数来保证其不同级别的可靠性。对于可靠性高的流(即重要的数据),采用完全可靠的传输策略;而对于那些可靠性较低的流,PRSCTP 通过新增加的控制块类型 FORWARD TSN,让发送方通知接收端把累积确认点强制性向前推移,发送方不用再重传该点之前丢失的分组,从而实现不可靠传

基金项目: 国家自然科学基金资助项目(60673185)

作者简介: 许 林(1971 -),男,硕士研究生,主研方向:传输层协议,网络性能分析和评价;白光伟,教授、博士后

收稿日期: 2008-08-02 **E-mail:** xlinlyg@163.com

输。这样可减少重传次数，从而提高PRSCTP的实时性能。

1.3 MPEG4

MPEG4 是由国际电信联盟 ITU 和国际标准化组织 ISO 制定发布的视频、音频数据的压缩标准。为了追求更高的压缩效率，注重除去图像序列的时间冗余度，同时又要满足多媒体等应用所必需的随机存取要求，MPEG 压缩算法将视频图像序列划分为帧内图像(即 I 帧)、预测图像(即 P 帧)、双向预测图像(即 B 帧)。和静止图像压缩一样，I 帧采用独立的压缩编码，去除图像空间冗余信息，因此可以独立解码；P 帧压缩采用对过去的(最靠近的)I 帧或 P 帧进行前向预测编码，这种运动补偿方法去除了相邻图像帧之间时间上的冗余信息，因而获得比 I 帧更高的压缩比；而 B 帧的压缩类似于 P 帧，所不同的是 B 帧压缩同时考虑前后相邻帧之间的信息冗余，即双向运动补偿，因此 B 帧的压缩比最高。也就是说 I 帧影响了整个 GOP(Group Of Picture)的所有帧，故丢掉了 I 帧，则整个 GOP 里面的其他帧都不能正确解码，P 帧只影响后续的 B 帧，B 帧不影响任何帧。因此在这 3 种帧中，I 帧最重要。

2 基于带宽管理的流媒体传输策略

在前文中提到，SCTP 和 PRSCTP 都可以支持流媒体传输。但是它们在不同的网络环境下，性能表现上也各有千秋。因此结合两者的优点，提出了一种基于带宽管理的流媒体传输策略(Bandwidth Management based SCTP, BMSCTP)。

2.1 BMSCTP 原理

在研究过程中进行了多组传输 MPEG4 流媒体的仿真实验，传输层协议分别采用了 SCTP 和 PRSCTP。在实验中发现，当只考虑信道带宽受限情况时，用 SCTP 传输流媒体的时延非常大，1 h 的视频片段需要更长的时间传输，因此流媒体实际的播放效果比较差，要不时停下来等待后续数据的到来。而如果这时采用 PRSCTP 方案，并且只保证关键的 I 帧可靠传输，P 帧和 B 帧采用不可靠传输，这样可以有效减少所需的带宽开销，从而播放比较流畅。故在带宽受限的网络环境下，PRSCTP 传输流媒体的性能表现要优于 SCTP。而存在较为严重的信道出错情况时，并且这时带宽能够保证所有数据包都能正常传输和重传，用 SCTP 传输显然丢包情况很好，即发送的视频数据基本上都能顺利到达接收端。虽然端到端的时延仍然有点大，但考虑到流媒体的播放时间与数据的到达时间是不一样的，即为了防止播放的抖动性，实际上接收端接收到的数据有一部分保存在缓存里，因此认为这点时延不会对视频的播放质量造成太大的影响。相反，如果这时采用 PRSCTP 作为传输协议的话，由于 PRSCTP 采用的是部分可靠的策略，尽管时延情况要好一点，但由于丢掉一些“不太重要的”数据包，造成跳帧现象比较严重，播放质量不如 SCTP。故在带宽足够大的网络环境下，SCTP 传输流媒体的性能表现要优于 PRSCTP。最后又在实验中验证，当带宽受限并且同时存在较为严重的信道出错情况下，无论 SCTP 还是 PRSCTP 都不能比较好地传输流媒体，但用 PRSCTP 表现仍要好于 SCTP。

根据上面的分析表明，用 SCTP 和 PRSCTP 传输流媒体，各有利弊，它们各自适合不同网络环境下的流媒体传输，因此提出了这种基于带宽管理的流媒体传输方案(BMSCTP)。其原理是：通过实时计算端到端的可用带宽估计值和拥塞控制机制下允许发送端的最大发送速率，将这 2 个值作为激发 SCTP 和 PRSCTP 之间传输策略切换的判定条件。当链路带

宽估计值小于最大发送速率值时，认为这时候网络处于拥塞状态，而自动切换到 PRSCTP 的传输策略，即只保证对关键的 I 帧的可靠传输，而对 P 帧和 B 帧采用不可靠的传输，这样可有效地减轻拥塞状况；而当可用带宽估计值大于最大发送速率时，可以断定这时丢包的原因只可能是信道出错，这时通过丢 P 帧、B 帧，不能解决问题，采用 SCTP 的策略来保证每个数据包的可靠传输。

实现上述策略，就需要实时计算链路中端到端路径可用带宽。

2.2 可用带宽估计值算法

目前存在着许多种端到端带宽测量算法。归结起来看，如果按是否向网络内部注入探测包来进行划分，这些算法可分为 2 种类型，即被动测量技术和主动测量技术^[5]。

在主传输路径上，采用的是被动测量技术，这是因为传输路径中有正在传输的数据，只需用 2 个连续的 SACK 包就能够进行带宽估计，这种技术实际上已经在 TCP Westwood 上成功应用^[6]。将这种技术引入到 BMSCTP 上，认为应该可以得到更好的效果。因为 SCTP 与 TCP 不同，它完全采用累积确认(即 SACK)，而非 ACK。而一个 SACK 包通常可以声明多个数据包已经被正确接收，间隔的时间更长，资源的开销也会更小一点。

如果在 t_k 时刻，一个 SACK 包到达发送端，其确认的数据包总长为 d_k ，则从理论上讲， t_k 时刻的实时可用带宽可以通过下式确定：

$$B_k = \frac{d_k}{(t_k - t_{k-1})} = \frac{d_k}{\Delta t_k} \quad (1)$$

其中， t_{k-1} 是前面一个 SACK 的接收时间；而 Δt_k 为连续 2 个 SACK 包到达的时间间隔。

式(1)计算出来的是实时可用带宽测量值，其中 B_k 值与数据包总长 d_k 成正比，与 2 个 SACK 包的时间间隔 Δt_k 成反比。而由于 SACK 确认的数据包长度 d_k 经常发生变化，有时可能很小， Δt_k 也会受到网络因素的影响，这些都会造成计算出来 B_k 值精度不会太高，并且 B_k 值波动也很大。因此如果拿 B_k 作为激发策略切换的判定条件，则必然导致设计的传输策略变化过于频繁，从而增加了许多不必要的开销。为此采用一个近似值 \hat{B}_k 来代替实时可用带宽值 B_k ，作为可用带宽的估计值，它可由下式确定：

$$\hat{B}_k = \alpha_k \hat{B}_{k-1} + (1 - \alpha_k) \frac{B_k + B_{k-1}}{2} \quad (2)$$

其中， $\alpha_k = \frac{2\tau - \Delta t_k}{2\tau + \Delta t_k}$ 是在时间 t_k 的过滤系数，并且 τ 是个大于 RTT 的常量，它可以通过实验的方式获取一个较好的值。这样当 2 个 SACK 包的时间间隔 Δt_k 减少，过滤系数 α_k 会随着 Δt_k 减少而增加，则带宽估计值 \hat{B}_k 就不会随着 Δt_k 变化而变化太大。

考虑到 SCTP 支持多路径传输，故除了主传输路径外，还可能会存在着其他的备用传输路径。对于这些备用路径来说，一般情况下是没有数据传输，而要估计它们的可用带宽的话，只能采用主动测量技术，如报文对技术(packet pair)等。这时可以通过主动测量技术估计所有路径的可用带宽。当主传输路径出现拥塞，就将数据的传输路径切换到可用带宽估计值最大的那个备用路径，并且确定采用 SCTP 还是 PRSCTP 作为初始传输策略。但是由于多路径传输不是本文研究的重

点,另外采用这些思路必然增加很多资源开销,因此在这里不详细介绍。

2.3 传输策略切换的判定条件

通过上面方法,得到了端到端实时可用带宽的估计值。与 TCP 相类似, SCTP 也有一套拥塞控制机制,它定义了一个拥塞窗口值($cwnd$),这个值控制发送端发送数据包的速率。而 2 倍的拥塞窗口值($cwnd$)除以往返时延 RTT (即 $2cwnd/RTT$),可以认为是拥塞控制机制下允许发送端的最大发送速率。当式(2)计算的带宽估计值 $\hat{B}_k > 2cwnd/RTT$,则说明网络的实际可用带宽值很大,目前的发送数据的速率并没有超过实际可用带宽值,即出现了丢包事件必然不是拥塞造成的,只可能是信道出错等原因,故必须采用 SCTP 的传输策略,保证数据的完整可靠传输;相反如果 $\hat{B}_k < 2cwnd/RTT$,说明网络的可用带宽很小,则不得不采用 PRSCTP 策略,保证关键数据的传输。

3 性能仿真分析

本节通过 NS2 仿真方法验证提出的基于带宽管理的流媒体传输策略,分析研究其性能。仿真采用的网络拓扑结构如图 1 所示。

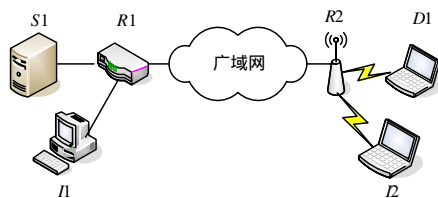


图 1 实验拓扑结构图

在图 1 中,节点 $S1$ 作为流媒体服务端发送视频数据,而节点 $D1$ 作为客户端接收数据。2 个路由节点分别为 $R1$ 和 $R2$,另外还有 2 个节点 $I1$ 和 $I2$ 作为干扰源,用于发送 CBR 数据流,并且在链路上添加了信道错误模型。 $S1$ 发送的视频文件为 Starwars 片段,传输时间为 1 h,传输层协议分别是 SCTP, PRSCTP 和 BMSCTP。设定节点 $I1$ 从第 1 800 s 开始向节点 $I2$ 发送 CBR 流。

因为其他实验验证,通过 SCTP 正常传输上述视频文件所需要的带宽开销大约是 315 Kb/s,同时考虑到链路还存在信道出错,所以在 2 个路由节点之间设定了瓶颈带宽为 750 Kb/s。本组实验可分为 2 个阶段。第 1 阶段,即前 1 800 s,实验只考虑信道出错的因素。第 2 阶段为后 30 min,由于链路上出现 CBR 流,它们抢占了大量的带宽资源,从而不足以保证视频源的正常传输,即实验中又增加了对带宽受限因素的考虑。这时设计的 BMSCTP 就应该自动从 SCTP 策略切换到 PRSCTP 策略。下面从数据包投递率和端到端传输时延 2 个方面深入分析研究本文提出的基于带宽管理的流媒体传输策略,评价其性能。

3.1 投递率

在第 1 阶段,用 SCTP 传输共计发送了 78 285 个携带数据块(data chunk)的数据包。仿真实验结果显示由信道出错造成的丢包数量为 3 194 个。但是通过一次或多次重传后,实际上真正丢失的数量仅为 7 个;用 BMSCTP 发送的数据包为 78 283 个,其真正丢失的包为 8 个;而用 PRSCTP, 30 min 共传输了 81 668 个数据包,而信道出错丢包 4 083 个,造成的最终丢包数量高达 3 742 个。通过上述数据分析,表明在只考虑信道出错的因素下,用 3 种协议传输流媒体发送数据

包的个数差距不大。但丢包情况差距很大,尽管带宽还很充足,用 PRSCTP 仍然没有重传丢失了的 P 帧和 B 帧数据,这必然会造成视频播放时的跳帧现象。

在第 2 阶段,用 SCTP 传输共计发送了 11 235 个携带数据块的数据包,由信道出错等原因造成的丢包数量为 570 个,通过重传后,真正丢失的数量为 463 个,这是由于经过多次重传后,客户端仍不能成功接收而造成的;而用 BMSCTP 发送的数据包为 37 735 个,其真正丢失的包为 1 733 个;而用 PRSCTP, 30 min 共传输了 37 953 个数据包,而信道出错丢包 1 886 个,造成的最终丢包为 1 738 个。在这个阶段,由于既存在信道出错,又受到带宽受限的影响, SCTP 几乎已经无法正常传输数据了,而后 2 种传输协议仍然能继续传输部分数据,因此在这种情况下, PRSCTP 和 BMSCTP 的性能表现显然要好于 SCTP。这是因为它们在牺牲了一些不太重要的数据前提下,缓解了带宽的拥塞状况,从而尽可能传输更多的数据,观众仍可以较为流畅地观看视频。

3.2 传输时延

图 2 显示 3 种传输层协议传输流媒体的端到端时延比较。其中图 2(a)显示的是 1 h 内的变化,图 2(b)显示的是 1 700 s ~ 1 900 s 之间的变化。

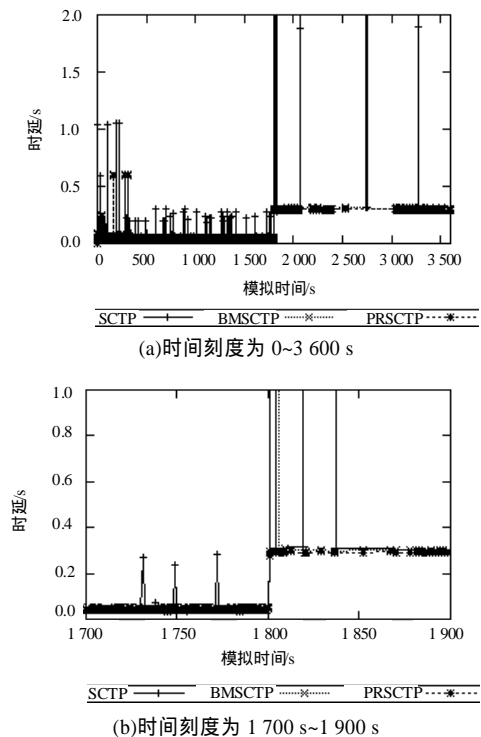


图 2 3 种传输层协议的端到端时延比较

在图 2(a)中显示,前 1 800 s 中, SCTP 和 BMSCTP(其实这时就是 SCTP)比 PRSCTP 多了一些突发值,这是因为信道出错的数据包可能是 I 帧数据,也可能是 P 帧和 B 帧,前两者协议是采用完全可靠传输,全部丢失的数据都要重传,而 PRSCTP 则只需要重传 I 帧数据,故出现了一些差异。通过上述分析表明,在第一阶段上,3 种协议传输时延差距不大。图 2(b)中显示,由于链路上出现大量 CBR 流,它们抢占了很大带宽资源,使得链路可用带宽不足以保证视频源的正常传输。这时不仅有信道出错的丢包外,还增加了超时丢包。因

(下转第 113 页)