

# 流控制传输协议传输流媒体性能研究

许林, 白光伟

XU Lin, BAI Guang-wei

南京工业大学 信息科学与工程学院, 南京 210009

College of Information Science and Engineering, Nanjing University of Technology, Nanjing 210009, China

E-mail: xlinlyg@163.com

XU Lin, BAI Guang-wei. Performance study of media streaming over SCTP. Computer Engineering and Applications, 2008, 44(1): 112-116.

**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. It is characterized by multi-homing and multi-streaming. In this paper, we designed and conducted a series of simulation experiments to demonstrate that SCTP may provide significantly advantages over the traditional TCP and UDP in real-time multimedia applications, in terms of delivery ratio, bandwidth cost, end-to-end delay, jitter, etc. On this basis, we explored a new strategy used to deliver video for streaming media applications with higher quality.

**Key words:** Stream Control Transmission Protocol (SCTP); media streaming; performance analysis

**摘要:** SCTP 协议 (Stream Control Transmission Protocol) 是一种面向报文的、可靠的传输层协议。它基于不可靠的、无连接的分组 IP 网络, 具有多宿 (multi-homing) 和多流 (multi-streaming) 等特性。通过仿真的方法分析, 证实 SCTP 在支持实时多媒体网络应用时的许多性能参数明显优于传统的 TCP 和 UDP, 包括数据包投递率、带宽开销、端到端延迟、延迟抖动等。在此基础上, 探讨一个适合在流媒体应用中, 能够有效提高流媒体传输质量的策略。

**关键词:** 流传输控制协议; 流媒体; 性能分析

**文章编号:** 1002-8331(2008)01-0112-05 **文献标识码:** A **中图分类号:** TP393.02

## 1 引言

近年来, 作为 Internet 最重要的应用技术之一的流媒体技术得到了迅猛的发展。流媒体的应用给用户带来了极大的便利。用户无需花费很长时间将多媒体数据全部下载到本地后再播放, 而只需将起始几秒的数据先下载到本地的缓冲区中就可以开始播放, 后面收到的数据会源源不断输入到该缓冲区, 从而维持播放的连续性。与传统的先下载后播放模式相比, 流媒体技术更复杂。这需要将多媒体的编、解码和传输技术很好地结合在一起, 才能确保用户能得到较好、较稳定的播放质量。目前多媒体的编解码技术如 MPEG-4、H264 等, 可以以较小的带宽开销实现较高质量的视频传输。而在 TCP/IP 协议簇中, 传输层协议长期以来主要包括 UDP 和 TCP。考虑到流媒体的实时性且用户容许一定程度上的错误的因素, 故通常采用 UDP 加 RTP/RTCP 的组合来传输流媒体。

然而无线网络流媒体的广泛应用, 给目前的传输技术带来了挑战。这是因为无线网络状况很不稳定, 设备的移动速度和所在位置都会严重地影响到传输。同时无线信道的环境也要比有线信道恶劣得多, 故造成很高的数据误码率。而高压缩的码流对传输错误非常敏感, 这会造成错误向后面的图像扩散。而

这是 UDP 加上 RTP/RTCP 的组合所无法解决的问题。因此无线流媒体应用需要更好的传输技术。

流控制传输协议 (SCTP: Stream Control Transmission Protocol) 是一种新型的传输层协议, 于 2000 年 10 月成为 IETF 的标准 (即 RFC2960)<sup>[1]</sup>。它兼具 UDP 和 TCP 的一些优点, 同时它还具有诸如面向消息的传输、多流 (Multi-streaming)、多宿 (Multi-homing) 等特性, 可以用来替代 UDP 传输流媒体<sup>[2]</sup>。另外 SCTP 的延伸版本 PRSCTP (即部分可靠的 SCTP)<sup>[3]</sup>, 它的方案是对数据包进行区分, 增加了前向确认块。这样可以为不重要的数据设置很短的生存期, 从而限制了它们的重传。这样的好处就是在带宽有限的条件下, 可以保证最重要的数据的传输。这种特性当然也可以应用于流媒体<sup>[4,5]</sup>。

影响流媒体传输的两大因素就是带宽和信道。且考虑到 SCTP 是在 TCP 的基础上发展起来, 继承了 TCP 的相当多的特性, 总体特性也强于 TCP<sup>[6-8]</sup>。故在下文只对 UDP 和 SCTP 进行性能比较。论文的第 2 部分是比较 UDP 和 SCTP 在理想条件 (即带宽足够大并且没有信道错误) 下传输流媒体的性能表现, 第 3 部分则比较只考虑带宽受限的因素下这两种传输层协议的性能, 第 4 部分则是只考虑信道出错情况下的 UDP 和 SCTP

**基金项目:** 国家自然科学基金 (the National Natural Science Foundation of China under Grant No.60673185)。

**作者简介:** 许林, 男, 硕士研究生, 主要研究领域为传输层协议, 网络性能分析和评价等; 白光伟, 男, 博士后, 教授, 主要研究领域为网络体系结构和协议, 无线 Ad Hoc 网络, 无线移动 Internet, 多媒体网络, 网络系统性能分析、评价等。

的性能比较,而第5部分是根据前面三组实验的结果得出的结论,以及在此基础上提出的一个更好的流媒体传输思路。

## 2 理想条件下的实验

本组实验是在带宽足够大并且没有信道错误环境下,比较这两种传输层协议的性能差距,并给后续实验提供参照值。实验采用 NS2 仿真的方法,场景如图 1 所示。

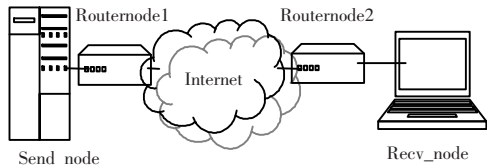


图1 实验拓扑结构图

图1中,节点 Send\_node 作为流媒体服务端发送视频数据,而节点 recv\_node 作为客户端接收数据,中间经过两个路由节点分别为 Routernode1 和 Routernode2,在这三段链路中带宽分别是 10 Mb/s、1 Mb/s 和 10 Mb/s,传输时延都是 1 ms。传输层协议分别采用 UDP 和 SCTP,包的大小控制在 1 024 字节以内。

采用电影“星球大战”的一个片断作为实时流媒体数据源(MPEG 视频压缩数据文件来自: <http://www.tkn.tu-berlin.de/research/trace/ltvt.html>, GOP 结构为  $N=12, M=3$ ); 该视频段文件播放时间为 1 个小时,共包括 89 998 个视频帧,总计为 126 386 573 字节。下面详细分析、研究实验结果。

### 2.1 投递率

当用 UDP 传输时,由于某些视频帧的大小超过了 1 024 字节,故 UDP Agent 要将它们分段,因此在这 1 个小时的视频传输中,节点 Send\_node 发送的数据包总共有 166 393 个。

当用 SCTP 传输时,节点 Routernode1 和 Routernode2 彼此发送的数据包总共有 240 657 个,其中节点 Routernode2 接收到的带数据块(data chunk)的数据包总共 166 393 个。

在这组实验中没有出现丢包现象。

### 2.2 带宽开销

图 2 显示的是两种传输层协议在带宽足够大且没有信道错误环境下传输流媒体的带宽开销变化。其中图 2(a)显示的是 1 个小时内的变化,而图 2(b)显示的是第 1 000 秒到 1 100 秒区间内的变化。图 2(c)反映的是带宽开销的概率密度函数(PDF),图 2(d)是概率分布函数(CDF)图。

通过图 2 发现,在 UDP 传输流媒体过程中,实际的带宽开销最高不到 450 Kb/s,正常情况下小于 300 Kb/s。而用 SCTP 传输流媒体,实际的网络开销最高也不到 500 Kb/s,正常情况下也在 300 Kb/s 左右。并且计算了仿真产生的数据,在一个小时的视频传输过程中,用 UDP 平均带宽开销为 295.8 Kb/s,其数学期望值为 289.93 Kb/s。而 SCTP 则为 312.14 Kb/s,其数学期望值为 311.16 Kb/s。

### 2.3 传输时延

图 3 显示的是两种传输层协议在理想条件下传输流媒体的时延变化。其中图 3(a)显示的是 1 个小时内的变化,而图 3(b)显示的是第 1 000 秒到 1 100 秒区间内的变化。同样图 3(c)是概率密度函数图,而图 3(d)是概率分布函数图。

在 UDP 传输流媒体过程中,实际的传输时延最高在 0.02 秒左右。而用 SCTP 传输流媒体,实际的传输时延最高要接近

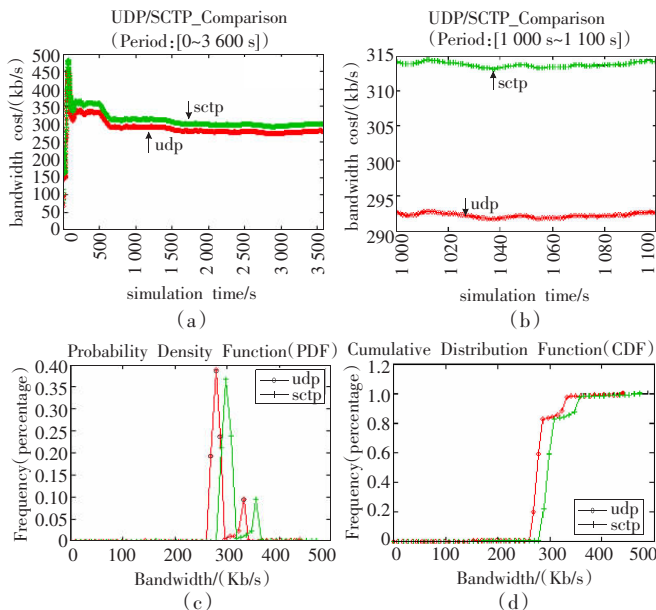


图2 带宽开销比较

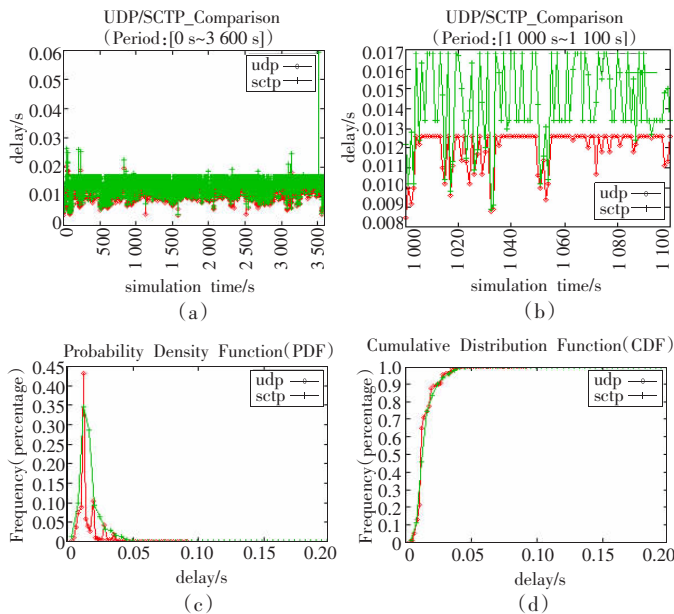


图3 端到端的传输时延比较

0.03 秒。经过计算,在整个视频传输过程中,用 UDP 平均时延只要 0.015 76 秒,其数学期望值为 0.015 59 秒。而 SCTP 平均时延 0.018 17 秒,其数学期望值为 0.017 85 秒。

### 2.4 延迟抖动

延迟抖动(Jitter)是指延迟时间变化量。Jitter 越大,则表示网络越不稳定。图 4 给出 SCTP 和 UDP 两种传输层协议在带宽足够大的条件下传输流媒体的 Jitter 比较。

图 4 中两种传输层协议的曲线几乎重叠。再进一步计算平均 Jitter 值,UDP 为 0.016 02 秒,而 SCTP 为 0.016 30 秒。这说明在带宽足够大的条件下传输流媒体,这两种传输层协议 Jitter 值没有差距。

通过这组实验得出如下结论:在理想条件下,SCTP 传输流媒体毫无疑问要比 UDP 需要更大一点的带宽开销。这是因为 SCTP 为了保证数据包的安全到达,要有确认机制,还要有拥塞和流量控制,同时建立联接也需要 4 次握手过程,故与 UDP 相

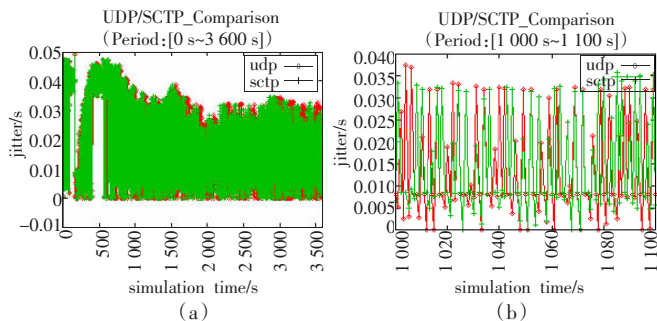


图4 延迟抖动比较

比,SCTP有很多控制机制和控制块。这些都占用网络资源的。不过由实验可知,增加的带宽开销并不大。同样的理由也可以用来解释时延差距。因此在这种理想的环境下,用UDP或用SCTP传输流媒体性能差不多。

### 3 只考虑带宽受限的因素

由第一组实验得出的结论可以知道,当网络的拓扑结构不变的话,传输该 trace 文件,用UDP和SCTP都需要大约300Kb/s的带宽开销,因此在这组实验设定的瓶颈带宽为250Kb/s,同样在这组实验中也没有考虑信道出错的因素。这组实验的目的就是验证在带宽受限下传输流媒体,用SCTP是否如我们想象的那样比UDP差得多。

#### 3.1 投递率

对UDP仿真产生的Trace文件进行分析,发送的数据包与实验1中链路带宽很大的数量是一样的,都是166393个数据包,而客户端收到的数据包为129196个,共有37198个数据包丢失,数据包丢失率高达22%,只有大约78%的数据到达了接收端。

而对SCTP仿真产生的Trace文件进行分析,服务端在1小时内共发送带数据块(data chunk)的数据包是140424个数据包,有25960个数据包没有及时发送出去,发送率仅达到84%。而在已发送的数据中,有7905个数据包丢失,丢失率为5.6%,这是由于网络拥塞,SCTP通过多次重传,仍然不能把这些数据包发送出去,最后由于超时或超过重传的次数而丢弃。

#### 3.2 带宽开销

图5显示的是两种传输层协议只考虑带宽受限因素下传输流媒体的带宽开销变化。其中图5(a)显示的是1个小时内的变化,而图5(b)显示的是第1000秒到1100秒区间的变化。图5(c)是概率密度函数图。图5(d)为概率分布函数图。

由于带宽比较小,所以在传送流媒体的时候,无论UDP还是SCTP都无法完整将数据传输到接收端。UDP采用的是丢的策略,链路正常就传输,一出现拥塞就丢包,然后就等着传输下一个数据包,所以它并没有完全占用整个链路的带宽,经过计算得出,它实际上只占用210Kb/s。而SCTP则不同,出现丢包,必须要重传,因此它几乎消耗掉整个链路的带宽,一直都在250Kb/s附近。

#### 3.3 传输时延

图6显示的是两种传输层协议在带宽受限下传输流媒体的时延比较。其中图6(a)显示的是1个小时内的变化,而图6(b)显示的是第1000秒到1100秒区间的变化。同样图6(c)是时延的概率密度函数图。图6(d)是概率分布函数图。

由图6所示,在带宽受限下,用SCTP传输流媒体的传输延

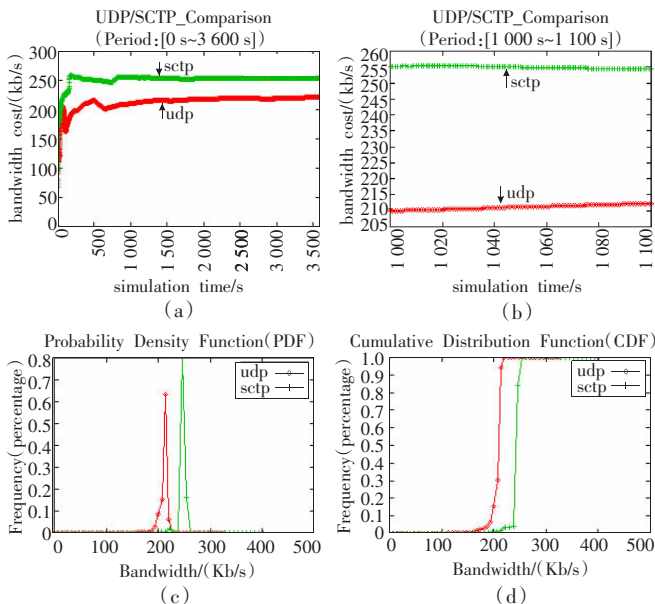


图5 带宽开销比较

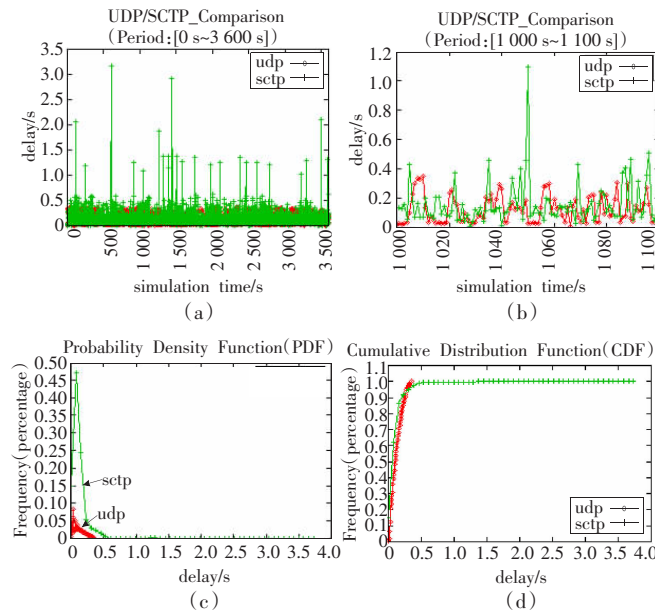


图6 端到端的传输时延比较

迟比用UDP的大了许多。UDP最大时延为0.36739秒,而SCTP则为3.80541秒;平均时延UDP为0.12713秒,SCTP则为0.16493秒。而且从图中可以看出SCTP的时延波动比较大。这都是因为用SCTP传输时,当出现丢包情况的时候,发送端要等待SACK包,要重传,有时甚至还要多次重传,自然传输时延会更长。

#### 3.4 延迟抖动

图7显示的就是这两种传输层协议在带宽受限的条件下传输流媒体的Jitter比较。

通过图6就可以看出在带宽受限的条件下,UDP的这个指标更好。图7验证这个结果。

通过上面几项指标的比较,我们认为在网络状况比较糟糕的情况下,用UDP传输流媒体时,发送端不管数据包是否能到达接收端,而只管把该发送的数据包全发送出去,因此丢包现象很严重。即不管播放质量如何,UDP发送端都把这一小时的流媒体数据全部按时地发送出去,作为观众不需要花更多的时

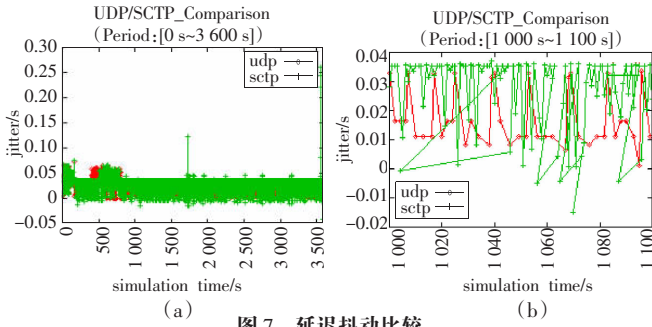


图7 延迟抖动比较

间来等待视频的传送,当然播放质量很糟糕。而用 SCTP 传输流媒体时,用户不得等待更长时间的数据传送,播放的效果则可能要不时停顿一下,等待后续数据的到达。因此在这种情况下,用 SCTP 和用 UDP 传输流媒体各有利弊。用 UDP 传输的流媒体播放质量要更差,而用 SCTP 传输则延迟要更大。

#### 4 只考虑信道错误的因素

这组实验的目的是验证在带宽足够大但存在信道错误的情形下,这两种传输层协议的性能表现。因此采用的是与实验 1 相同的场景,但是加入了错误模型。

##### 4.1 投递率

当用 UDP 传输时,在 1 个小时的时间里,服务端发送的数据包总共有 166 393 个。而接收端节点 Recv\_node 接收到的数据包共有 163 079 个。丢失的数据包共 3 314 个,即只成功接收到 97.98%的数据包。

当用 SCTP 传输时,1 个小时的时间里,节点 Send\_node 发送的数据包同样是 166 393 个,节点 Recv\_node 接收到的带数据块的数据包总共 166 252 个,共有 141 个包丢失。成功接收到 99.91%的数据包。

##### 4.2 带宽开销

图 8 显示的是两种传输层协议在带宽足够大且存在信道出错的环境下传输流媒体的带宽开销变化。图 8(a)显示的是 1 个小时内带宽开销的变化,图 8(b)显示的是第 1 000 秒到 1 100 秒区间内的变化,而图 8(c)是带宽开销的概率密度函数图,图 8(d)是概率分布函数图。

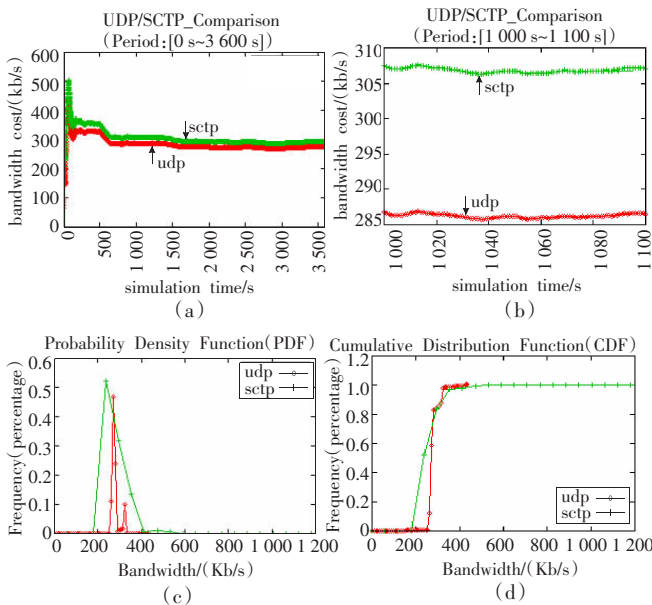


图8 带宽开销比较

用 UDP 传输的最大带宽开销为 437.31 Kb/s, 平均带宽开销为 284.82 Kb/s, 其数学期望值为 284.87 Kb/s。而用 SCTP 传输的话,这三个值分别是 1012.43 Kb/s、315.25 Kb/s 和 313.68 Kb/s。和第一组实验相比,这个指标对 UDP 没有任何影响,而 SCTP 则要受到一定的影响,主要表现在最大值有较大的变化,而平均带宽的占用情况略大一点。这是因为发生信道出错,就要丢包和重传,这样就会导致带宽会有较大的波动。

##### 4.3 传输时延

图 9 显示的是这两种传输层协议在存在信道错误下的传输时延对比。其中图 9(a)显示的是 1 个小时内的传输时延变化,图 9(b)显示的是第 1 000 秒到 1 100 秒区间内的变化。同样图 9(c)是概率密度函数图,而图 9(d)则是概率分布函数图。

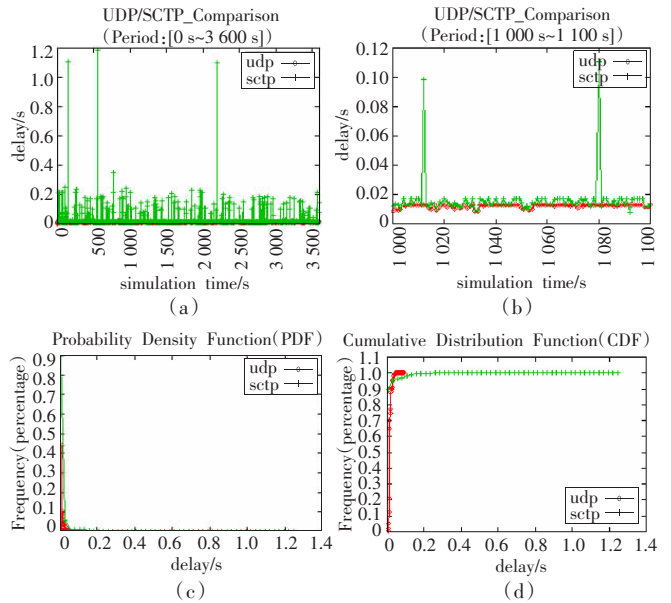


图9 端到端的传输时延比较

平均时延,UDP 是 0.015 65 秒,而 SCTP 是 0.022 98 秒;最大时延,UDP 是 0.091 584 秒,而 SCTP 是 1.271 35 秒;而它们时延的数学期望值分别是 0.015 60 秒和 0.022 20 秒。

##### 4.4 延迟抖动

图 10 显示的就是这两种传输层协议在存在信道出错的环境下传输流媒体的 Jitter 比较。

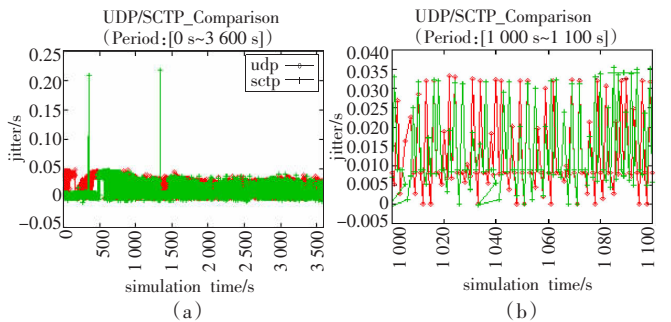


图10 延迟抖动比较

图 10 显示,这两种传输层协议在 Jitter 值这个指标上差距不大。只有当丢包情况发生时,SCTP 才会达到一个非常高的值。

通过上面几项指标的比较,在这组实验中,在丢包率方面,SCTP 要优于 UDP。从网络的带宽开销方面上看,UDP 无疑比 SCTP 好。在传输时延上,SCTP 似乎也比 UDP 更大。这是由于

信道出错,数据包就要丢失,对于 SCTP 就要重传,这时就会出现一个较大传输时延的数据包;而 UDP 已经把它们丢掉了,故 UDP 时延几乎跟第一组实验的结果一样。比较这几个指标,丢包率远比其它指标更重要。这因为在这组实验中,不管 UDP 还是 SCTP 都把所有的数据发送出去,而实际上流媒体的播放时间与数据的到达时间是分开的,也就是说接收端收到的数据有一部分保存在缓存里来防止播放的抖动性。所以用 SCTP 传输流媒体尽管出现有些数据包时延很长,但播放时并不会出现较大的抖动。相反,由于 UDP 丢了许多数据包,必然造成跳帧的现象。在本组实验所设定的场景下,最合适传输流媒体的是 SCTP,毕竟作为可靠的传输层协议,它的累积确认和重传策略,是目前解决丢包问题的唯一办法。

## 5 结束语

通过上述的三组实验的比较,作为新型的传输层协议的 SCTP 传输流媒体,在带宽很大且不存在信道错误的条件下,有不输于 UDP 的性能表现;而当带宽不足的情况下,则它的表现情况也并不比 UDP 差;而在带宽足够大且存在信道错误的情形下,用 SCTP 传输,它的丢包率则要远远优于 UDP。目前广泛采用 UDP 传输流媒体的主要原因是带宽的因素,但我们相信随着网络接入技术的不断发展,带宽影响将会有所好转,SCTP 替代 UDP 传输流媒体将是不错的选择。

然而还有一些问题值得继续探讨。由于目前带宽仍然有限,通过采用 SCTP 的延伸版本 PRSCTP 的方案来保证关键的 I 帧的可靠的传输,而让 P 帧和 B 帧采用不可靠的传输,这样可以在一定程度上减轻带宽受限的问题。但是 P 帧和 B 帧也很重要,特别是当场景快速变化的时候,丢掉了它们同样会对视频的播放质量造成很大的影响,故当带宽足够大的情况下,我们希望采用完全可靠的传输,而不采用部分可靠的方案。为此提出了一种解决方案,即将当前的带宽值作为一个激发 PRSCTP 策略的

判定条件。当目前的带宽远小于实验测出的理论值(这可以通过测量实时的带宽值与带宽理论值乘以一个权值进行比较)时,网络肯定是拥塞,就让 PRSCTP 策略起作用;而当带宽足够大的时候,可以认为这时的丢包原因只可能是信道出错,采用 PRSCTP 策略不能解决丢包的问题,而应该采用原来 SCTP 的策略,保证每个数据包的传输都能有效到达。如何实现这个方案将是近期研究的重点。(收稿日期:2007年9月)

## 参考文献:

- [1] Stewart R, Xie Q, Morneault K, et al. IETF RFC2960 Stream Control Transmission Protocol[S]. 2000.
- [2] Balk A, Sigler M, Gerla M, et al. Investigation of MPEG-4 video streaming over SCTP[J]. The International Journal of Computer and Telecommunications Networking, 2004, 44(4): 1022-1026.
- [3] Stewart R, Ramalho M, Xie Q, et al. IETF RFC3758 stream control transmission protocol(SCTP) partial reliability extension[S]. 2004.
- [4] Huang H, Ou J, Zhang D. Efficient multimedia transmission in mobile network by using PR-SCTP[C]//Communications and Computer Networks(CCN 2005), Oct 2005, Marina del Rey, USA, 2005.
- [5] Wang Hongtao. The performance comparison of PRSCTP, TCP and UDP for Mpeg-4 multimedia traffic in mobile network[C]//International Conference on Communication Technology, 2003, 1(2): 403-406.
- [6] Jungamaier A. Performance of the stream control transmission protocol[C]//Proceedings of the IEEE Conference 2000 on High Performance Switching and Routing, Heidelberg, Germany, 2000: 141-148.
- [7] Fu Shaojian, Atiquzzaman M. SCTP: state of the art in research, products and technical challenges [J]. IEEE Communications Magazine, 2004: 85-91.
- [8] Fracchia R.A. WiSE extension of SCTP for wireless networks[C]//Communications, ICC 2005 (IEEE International Conference), Seoul, Korea, 2005, 3: 1448-1453.

(上接 111 页)

签名,所以  $A_i$  不能否认自己的一个有效数字签名。同理可得  $B_i$  也不能否认自己的有效的代理盲签名。

### 4.4 签名的可区分性

若各签名方都正确执行方案,能将原始签名和代理签名分开。因为在对代理多重盲签名  $(e, s, U)$  的有效性进行  $e = H\left(m \parallel x \left( sG + e \sum_{i=1}^l R_{p_i} + U \right) \bmod N_n \right)$  验证时,要用到原始签名人公布的授权验证公钥  $R_{p_i}$ ,所以能很容易将原始签名和代理签名分开。

## 5 结束语

本文提出了一种基于环  $Z_n$  上的圆锥曲线的代理多重盲签名方案,其安全性主要基于  $n(n=pq, p, q$  为两个不同的大素数)的大数分解的困难性和环  $Z_n$  的  $C_n(a, b)$  离散对数问题。与基于有限域上离散对数的数字签名方案和环  $Z_n$  上的椭圆曲线的签名方案比较,除了保留原有的优点外,还具有明文嵌入方便、元素阶的计算及曲线上点的运算都比较容易、求逆元速度快以及更易于实现等优点,且在通过引进标准二进制计算群元素的整数倍的情况下,还能节约近 1/4 计算量,这些对于设计数字签名算法来说都具有较强的吸引力。

数字签名是实现信息保护的一种有效手段,目前数字签名技术研究已经比较成熟,但如何将各签名技术结合起来发挥更

有效的作用,满足实际需要,仍是一个值得研究的问题。本文提出了基于环  $Z_n$  上圆锥曲线的代理多重盲签名方案,该方案同时具备了代理签名、多重签名和盲签名的特点,且签名尺寸不会随着签名人数增加而增加。经过分析,证明该方案是安全高效的,具有一定的应用价值。(收稿日期:2007年9月)

## 参考文献:

- [1] Chaum D. Blind signature systems[C]//Proceedings of CRYPTO'83. New York: Plenum Press, 1984.
- [2] Mambo M, Usuda K, Okamoto E. Proxy signatures: delegation of the power to sign messages[J]. IEICE Trans Fundamentals, 1996, E79-A(9): 1338-1353.
- [3] 王标,朱文余,孙琦. 基于剩余类  $Z_n$  环上圆锥曲线的公钥密码体制[J]. 四川大学学报:工程科学版, 2005, 37(5): 112-117.
- [4] 肖龙,王标,孙琦. 基于环  $Z_n$  上的圆锥曲线数字签名和多重数字签名[J]. 西安交通大学学报, 2006, 40(6): 78-80.
- [5] 王洪涛,李大兴,周大水. 一种新的基于椭圆曲线的代理盲多重签名[J]. 微电子学与计算机, 2007, 24(2): 160-163.
- [6] Dai Zongduo, Pei Dingyi, Yang Junhui, et al. Crypta—analysis of a public key cryptosystem based on conic curves[C]//Proceedings of CryptEC'99: International Workshop on cryptographic techniques and ECommerce. Hong Kong: City University of Hong Kong, 1999: 259-261.