Using SCTP with Partial Reliability for MPEG-4 Multimedia Streaming

[Published in Proc. of BSDCon Europe 2002]

M. Molteni and M. Villari*
Cisco System Technology Center
06410 Sophia Antipolis, France.
e-mail: {mmolteni,mvillari}@cisco.com

October 24, 2002

Abstract

The MPEG-4 standard encodes a video stream in 3 different frames, I, P and B, where the P and B frames depend on the I frame. Loosing an I frame is especially bad. In this paper we exploit the Partial Reliability features of the SCTP transport protocol to selectively retransmit I frames in presence of congestion, obtaining a better quality of the decoded stream.

1 Introduction

The Stream Control Transmission Protocol (SCTP) is a relatively new IP transport protocol. It is reliable, connection and message-oriented, and has a set of new features that make it well suited for a wide class of applications. SCTP can provide ordered or unordered delivery, and when both sides implement Partial Reliability SCTP (PR-SCTP), the sender can choose the retransmission behavior on a per packet basis, in a continuous spectrum from TCP-like reliability with multiple retransmissions to UDP-like unreliability with no retransmission at all, always retaining TCP-friendly congestion control and congestion avoidance.

The MPEG-4 standard is becoming a popular format for streaming multimedia on the Internet. MPEG-4 encodes the video bitstream in groups of different frame types (I, P and B frames), where the I frame is independent, while the P and B frames depend on the I frame in the group. This means that loosing an I frame (for example due to network congestion) causes a noticeable worsening of the video quality of all the frames in the group. The transport protocol utilized by MPEG is RTP/UDP. RTP, being concerned with real-time traffic, does not provide reliable delivery.

Lately a Cisco implementation of SCTP has been imported in the IPv6/IPsec stack developed by the KAME project, providing kernel-level SCTP for the operating systems derived from BSD Unix.

This paper describes our preliminary work on FreeBSD to modify two open source programs, a MPEG-4 streamer and a MPEG-4 player, to utilize PR-SCTP as transport instead of RTP/UDP, enforcing differentiated Partial Reliability per frame type.

Our preliminary results show that under congestion scenarios there is improved playout quality of the video stream, due to the increment of the number of I-frames that PR-SCTP allows to salvage from congestion.

*on leave from University of Messina.

2 Background

In this section we briefly describe the various fields involved in our work, highlighting key concepts. The reader is referred to the bibliography for deeper treatment.

2.1 SCTP

The Stream Control Transmission Protocol (SCTP) [14] is a relatively new IP transport protocol. It is reliable and connection-oriented as TCP, and has a TCP-friendly congestion control.

While TCP is bytestream-oriented, SCTP is message-oriented (i.e. it preserves message boundaries). This means that an application that needs message boundaries doesn't have to provide its own framing.

SCTP can multiplex multiple "data streams" into one SCTP *association*. Each message is associated with a stream number, and messages belonging to the same stream are delivered in order. However, while one stream may be blocked waiting for the next in-sequence message, delivery from other streams may proceed, avoiding head-of-line blocking. Also, the packet delivery can be ordered or unordered, with stream granularity.

Further, when both endpoints implement Partial Reliability SCTP (PR-SCTP) [13], the sender can choose the retransmission behavior on a per message basis. As of today the only partially reliable service specified is the *timed reliability* service, but different notions of partial reliability can be introduced and the niceness of the extension is that the receiver doesn't have to know which kind of partial reliability is performed by the server.

Timed reliability means that the user can specify the *lifetime* of a message: when the lifetime is expired and the message hasn't been acked yet, the sender stops the retransmission efforts and drops the packet. In the protocol control plane, the sender will send a forward TSN (Transmission Sequence Number), telling the receiver to move its cumulative ack point forward. The effect of moving the ack point forward is to consider the skipped messages as received and acked.

2.2 MPEG

The Motion Pictures Experts Group (MPEG) released in 1998 the MPEG-4 standard [1]. Looking at the previous standards, MPEG-1 can encode up to 1.5 Mbps (low bit rate), whereas MPEG-2 can go up to 15 Mbps (high bit rate). MPEG-2 is used in applications such as DVD video and cable or satellite broadcast. MPEG-4 embodies several video codecs, such as Dvix and Xvid, which are capable of a ten times reduction of the bit rate in both areas (low and high bit rate) keeping the same quality. The MPEG-4 visual standard has been explicitly optimized for three bit rate ranges: below 64 kbps, 64–384 kbps and 384–4 Mbps. MPEG-4 provides also features for the animation of faces and synthetic bodies.

The MPEG-4 scene consists of a number of audio and video media objects. Several of them are typically background, like audio clips or static images. The information for each streaming media object are brought within one or more elementary streams.

The entire MPEG-4 standard includes specifications on hundreds of features, but no particular application needs to support all of those features. *Profiles* and *Levels* define what an application supports. A Profile defines the features and qualitative functionality, and the Level specifies the quantitative complexity of the functions within a Profile.

The basic object in MPEG-4 is a Video Object Plane (VOP), which can have any shape. A conventional video frame is represented by a VOP with a rectangular shape, and in this paper we use the term *frame* and *VOP* equivalently.

The MPEG encoding considers three kinds of frames: Intra-VOP (I), Predicted (P)

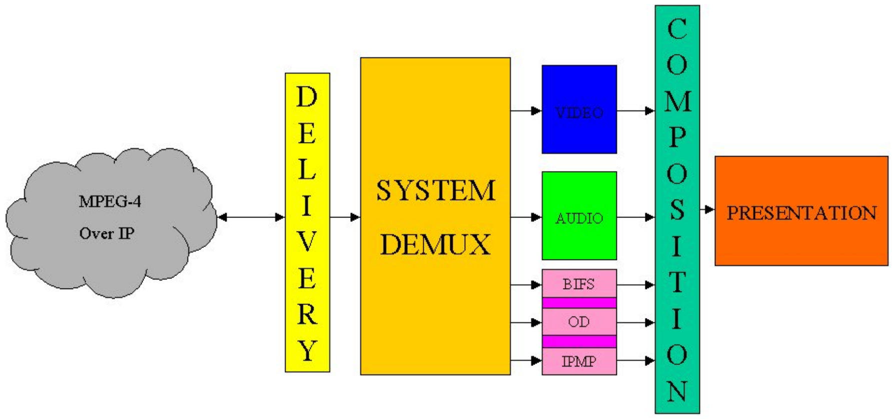

Figure 1: MPEG-4: The data streams corresponding to audio/video signals are stored separately. They are composed in an audiovisual and integrated presentation only to the receiver.

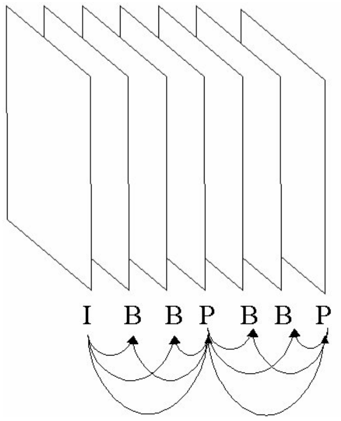

Figure 2: Group of frames (GOV), showing the dependency between frames

and Bidirectional interpolated (B) frames. I frames are coded independently from other frames; the compression is typically spatial. P frames are coded having as reference the time-preceding P or I frame. B frames are coded having as reference the time-adjacent I or P frames. B frames are never a reference for other frames. P and B frames are temporal compressed, and used to perform motion estimation and compensation.

A GOV (Group of VOPs) is a set of I, P and B frames; its sequence and length can vary during encoding. A GOV, as depicted in Figure 2, always begins with a I frame, which is also the only I frame in the GOV. The next I frame in the stream is the beginning of the next GOV. An important part of the information in a GOV resides in the I frame. It has to be noted that the encoding in I, P and B frames is done by the MPEG-4 ISO codec. Other codecs widely used, as for example Xvid, at least today only use I and P frames.

3 Related Work

There is a lot of research going on to improve the quality of video streaming over best effort networks, trying to introduce intelligent retransmission.

The Internet Draft [8] describes new RTP payload formats to enable multiple and optional selective retransmissions in RTP. These are especially applicable to environments where enhanced RTCP feedback is available. These payload formats can be used to separate the media stream according to prioritization of packets or according to the status of the transmission (i.e. transmission or retransmission).

Feamster [6] realizes a complete environment to test Selective Reliability RTP (SR-RTP). He uses RTP over UDP with an extension to enable a feedback communication between server and client. He has developed a streaming application to delivery high quality video. He introduces also a post-processing phase to recover some of the packet loss, and analyzes the results with peak signal-to-noise ratio (PSNR).

The work by Raman [10] is an implementation and evaluation of the Image Transport Protocol (ITP) for image transmission over lossy or wireless networks. They verify the quality at Application Level Framing (ALF); as measure they consider the evolution of the PSNR. ITP runs over UDP and incorporates receiver-driven selective reliability. ITP enables a variety of new receiver post-processing algorithms such as error concealment that further improve the interactivity and responsiveness of reconstructed images.

4 MPEG-4 over PR-SCTP

RFC [7] describes how MPEG-4 in an IP network is transported over the real-time transport protocol (RTP) [11]. RTP itself is normally transported over UDP[1].

UDP is best effort as IP, and since RTP concerns itself with real-time data, a RTP packet lost in the network will not be retransmitted by the sender. This is normally what is wanted, because there is no point in receiving a time sensitive message after its useful lifetime has expired, and so retransmission would be useless for the receiver and potentially bad for the overall network congestion.

Applications like streaming video use time-sensitive messages, but to work around temporary network delays they employ a few seconds buffering. MPEG-4 encodes the video stream in three different frames types, I, P and B, as seen in section 2.2. Loosing an I frame is strongly worse than loosing a P or a B frame. The idea at the basis of our and other works is to somehow give reliability (by means of retransmission) to what is really important, I frames.

[1] It is also possible to transport RTP over TCP, in the so-called interleaved mode of RTSP (Real Time Streaming Protocol) [12].

In our case we obtain I frames reliability by exploiting a native feature of the SCTP transport protocol, namely partial reliability, instead of having the application adding this on top of RTP/UDP as is done in other approaches.

As in other works we analyze the signal quality information from the application level with PSNR, as detailed in Section 6.

Nonetheless, reliability by itself is useless if the message is received after its validity is expired, and bandwidth-heavy if used for all messages without regards to the importance. SCTP naturally solves both problems, because

- the granularity of the retransmission is per message, and so only messages containing I frames can be tagged for retransmission.
- the efforts in retransmissions are tunable, and in our case the *timed reliability* maps directly to the time-sensitiveness of video streaming.

Also, since MPEG-4 takes care of message reordering, we asked SCTP for unordered delivery.

From an implementation point of view, as detailed in section 5, we took an existing streaming server and player and replaced the UDP sockets with SCTP ones, trying to perturb as little as possible the surrounding code.

When the server was to send an I frame, we set the time to live of the message to a tunable value, to be determined by experimentation. As default we used 2000 ms. Conversely, when the server was to send a P or B frame, we set the time to live to the special value `SEND_EXACTLY_ONCE`.

We open the SCTP socket in the so-called UDP-style (a UDP-style SCTP socket has

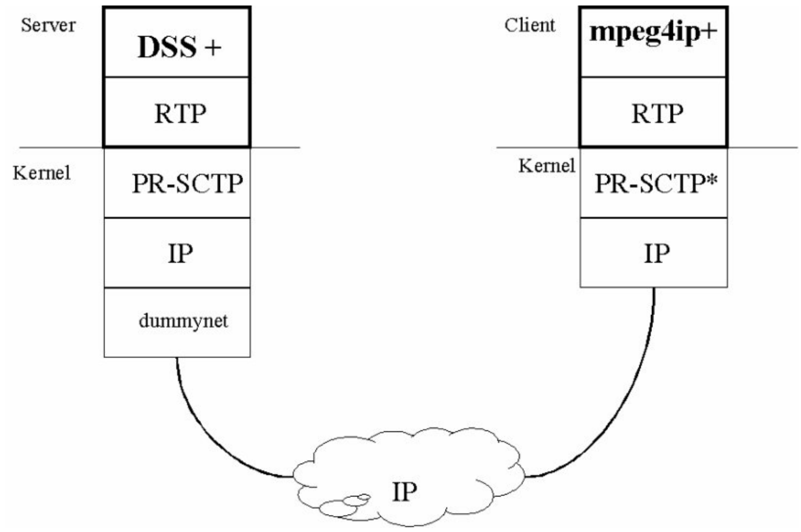

Figure 3: Client Server scenario

nothing to do with UDP unreliability, it just refers to one of the two models offered by the SCTP socket API[2]) because it maps well to the UDP interface that the server expects and because it is the model that allows us fine-grained control over the per message SCTP behavior.

Since we used SCTP instead of UDP to encapsulate RTP, the payload available for RTP was smaller, and we had to take this into consideration when creating the *hint* track of the MPEG-4 file to be read by the server.

Figure 3 depicts the components of our scenario. On the server side, DSS+ is the Darwin Streaming Server as modified to use SCTP. Dummynet is used to introduce random packet losses. On the client side, mp4player+ is the mp4player as modified to use SCTP. PR-SCTP* on the client is to remember that the SCTP receiver doesn't have to know which reliability algorithm is used by the sender.

5 FreeBSD Implementation

We used a FreeBSD 4.6-RELEASE machine plus various KAME snapshots. The original intent was to use one machine as the server and various machines as the clients, but due to time constraints and a few bugs we finally settled to use only one machine and the loopback interface.

We used dummynet [9], a network emulator and traffic shaper included in FreeBSD, to introduce random packet drops in the test network.

5.1 KAME/SCTP

Randall Stewart and Peter Lei, both from Cisco, wrote a SCTP kernel implementation [5] for the various BSDs, which is now part of the KAME source code [3]. Installing a KAME snapshot and adding the line `options SCTP` to the kernel configuration file is enough to enable SCTP support, providing the SCTP socket API as defined in [15].

Since the SCTP kernel implementation is actively developed, we kept following both the KAME snapshots and the SCTP patches against the snapshot, until October 2002.

5.2 Darwin Streaming Server

The Apple Darwin Streaming Server (DSS) [2] is mostly written in C++. It starts

[2]See section 2.1 for further details.

from a generic socket class, from which a UDP socket class is inherited. On top of that, since RTP/UDP requires two sequential ports (even and even + 1), it has "socket pairs" and various other gadgets. We replaced the UDP socket used by RTP with a SCTP socket, leaving the companion RTCP socket as UDP.

We then tought to the SCTP socket that it had to send its packets with Partial Reliability, depending on the frame type: P and B frames had to be send just once (as plain RTP), while I frames[3] had to be eventually resent. Since the socket class didn't had the notion of frame type, we had to find out where in the code we could grab this information and how to pass it all the way down to the SCTP socket class.

5.3 MPEG4IP mp4player

The Cisco MPEG4IP [4] mp4player is written in C++ and relies on the UCL (University College London) RTP library, written in C. As in the Darwin Streaming Server case, we replaced the RTP UDP socket with a SCTP socket, both in mp4player and in the RTP library. We also extended the RTP library to accept already initialized sockets instead of doing the initialization by itself.

We also modified some of the decoder plugins used by the player, as detailed in Section 5.4.

5.4 Tools

The mp4player uses plugins for the decoding, among them MPEG-4 ISO and Xvid. We modified both plugins to gather and log various informations related to timings, sequence number, frame size, etc, and to write to file a full dump of the raw video stream (in YUV format) just decoded.

We also wrote a tool, `psnr`, to analyze two raw video dumps, generated by the plugins, and to calculate the peak signal-to-noise ratio (PSNR) between each corresponding frame, as described in section 6. The same tool is also able to extract a frame form the YUV dump and export it in PPM format.

[3] called Key Frames in DSS.

6 Experimental Results

In this section we provide an analysis of the experimental results we obtained. We would like to point out that these results are preliminary and require further testings and analysis. We made tests from both high quality (MPEG-2 DVD tracks converted to medium quality 700 kbps) and low quality (200 kbps) video streams, and we present here the results for the low quality case.

We used peak signal-to-noise-ratio (PSNR) in Equation 1 as method to evaluate in an objective manner the quality of video streams. PSNR is widely utilized in literature to evaluate the quality of a generic image before and after a lossy compression. Although it is not the best technique to synthesize the human visive perception, it provides a reasonable level of objectivity.

We compared the same video sequence with and without packet drops in the UDP and SCTP case. The dummynet traffic shaper [9] allowed us to vary the drop rate from 2^{-8} (ca 0.004) to 2^{-3} (ca 0.125).

Figure 4 shows the differences in PSNR for UDP (upper picture) and SCTP (lower picture) in the case of a 2^{-6} drop rate. On the x axis there is the frame number, on the y axis there is the PSNR in dB. The 100 dB peak actually means infinity, when the two frames are the same, thus the more the peaks, the higher the quality of the video stream. The graphs show that the SCTP case has a higher number of peaks.

Also in our experiments we noted that for high packet drop rates (2^{-3}) the SCTP case is actually worse than UDP. Our gut feeling is that the SCTP congestion control mechanism gets triggered, while UDP happily keeps sending packets at full throttle worsening the congestion. Further experimenta-

$$PSNR = 20\log_{10}\frac{255}{\left(\frac{1}{N_1N_2}\sum_{x=0}^{N_1-1}\sum_{y=0}^{N_2-1}[f(x,y)-f'(x,y)]^2\right)^{1/2}} \tag{1}$$

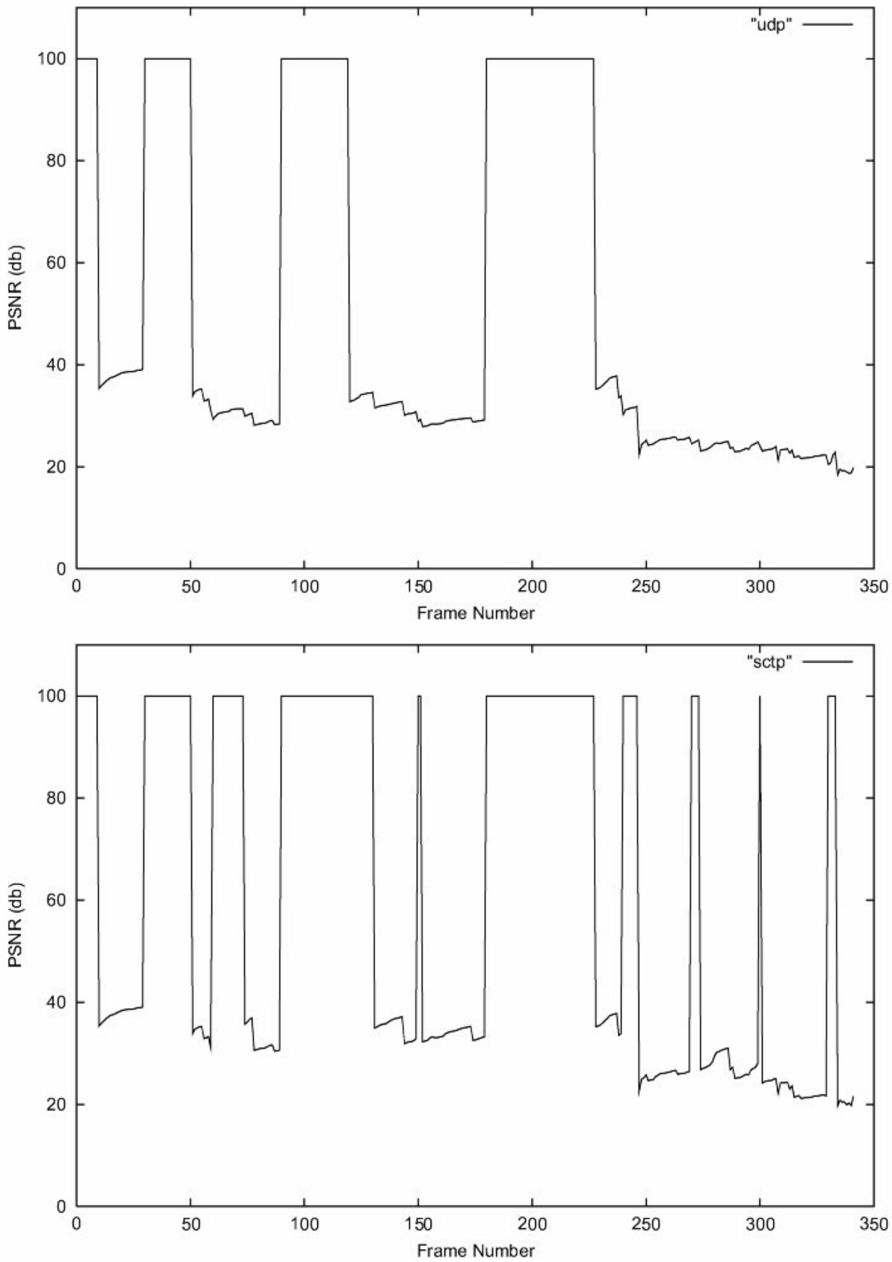

Figure 4: PSNR for UDP and SCTP.

tion is needed to replicate the behavior and to give a proper explanation.

7 Conclusions and Future Work

In this paper we showed that using Partial Reliability SCTP to optionally retransmit MPEG-4 I frames may result in improved quality of the decoded video stream. Further work is required to better identify and qualify the congestion scenarios that might be improved by the usage of PR-SCTP.

For the future, we plan to address some of the shortcomings and limitations of the present work, among them we will utilize different hosts for the server and the clients, and we will do more tests with deeper analysis.

Some interesting ideas we have are to better integrate RTP with SCTP instead of just replacing the UDP encapsulation with the SCTP one, and to exploit the native SCTP stream multiplexing.

We also plan to make the various components we used more robust. We had many crashes, and while some of them may be explained by the perturbations introduced by our modifications, some others seems to be more general, and related to insufficient input validation.

References

[1] Iso/iec 14496-1 final draft international standard mpeg-4 systems (oct. 1998).

[2] `http://www.developer.apple.com/darwin/projects/streaming/`.

[3] `http://www.kame.net/`.

[4] `http://www.mpeg4ip.net/`.

[5] `http://www.sctp.org/`.

[6] Nicholas Feamster. Adaptive delevery of real-time streaming video. Master's thesis, Massachusetts Institute of Technology, 2000.

[7] Y. Kikuchi, Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, and H. Kimata. RFC 3016: Rtp payload format for mpeg-4 audio/visual streams, 2000.

[8] A. Miyazaki et al. *RTP Payload Formats to Enable Multiple Selective Retransmissions*. draft-ietf-avt-rtp-selret-05.txt, 2002.

[9] L. Rizzo. Dummynet and forward error correction. in proc. freenix 98, 1998.

[10] M. Srinivasan S. Raman, H. Balakrisman. An image transport protocol for the internet. In *2000 International Conference on Network Protocols.*, 2000.

[11] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RFC 1889: RTP: A transport protocol for real-time applications, 1996.

[12] H. Schulzrinne, A. Rao, and R. Lanphier. RFC 2326: Real time streaming protocol (rtsp), 1998.

[13] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, and P. Conrad. *SCTP Partial Reliability Extension*. draft-stewart-tsvwg-prsctp-00.txt, 2002.

[14] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. RFC 2960: Stream control transmission protocol, 2000.

[15] R. Stewart, Q. Xie, L. Yarroll, J. Wood, K. Poon, K. Fujita, and M. Tuexen. *Sockets API Extensions for Stream Control Transmission Protocol (SCTP)*. draft-ietf-tsvwg-sctpsocket-05.txt, 2002.