EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH ARTICLE Open Access

SCTP as scalable video coding transport

Jordi Ortiz*, Eduardo Martínez Graciá† and Antonio F Skarmeta†

Abstract

This study presents an evaluation of the Stream Transmission Control Protocol (SCTP) for the transport of the scalable video codec (SVC), proposed by MPEG as an extension to H.264/AVC. Both technologies fit together properly. On the one hand, SVC permits to split easily the bitstream into substreams carrying different video layers, each with different importance for the reconstruction of the complete video sequence at the receiver end. On the other hand, SCTP includes features, such as the multi-streaming and multi-homing capabilities, that permit to transport robustly and efficiently the SVC layers. Several transmission strategies supported on baseline SCTP and its concurrent multipath transfer (CMT) extension are compared with the classical solutions based on the Transmission Control Protocol (TCP) and the Realtime Transmission Protocol (RTP). Using ns-2 simulations, it is shown that CMT-SCTP outperforms TCP and RTP in error-prone networking environments. The comparison is established according to several performance measurements, including delay, throughput, packet loss, and peak signal-to-noise ratio of the received video.

Keywords: H.264/SVC, SCTP, CMT-SCTP, Adaptive video streaming, ns-2

Introduction

Quick developments in network infrastructure, processing power, and storage capability in recent years are making possible the growth of multimedia services through the Internet, giving rise to a bunch of applications that include video streaming, video conference, and high-definition broadcasting. Nevertheless, most of these multimedia services are currently implemented using technologies that are not properly designed to cope with the strong variability in the quality of transmission, which is experienced preeminently in wireless and mobile networks. Equally neglected in communications nowadays is the heterogeneity of receiver devices when providing ubiquitous multimedia services.

Regarding source video compression, it is still common to find MPEG-2 and H.264/AVC as the video codecs in use. However, the main profiles of these codecs were designed to cope with a fixed space-temporal video signal because they were conceived to be employed with guaranteed resources for transmission and decoding. Although the extended profile of MPEG-2 permits some degree of scalability, it is seldom used because of the strong processing power that it compels. In video streaming, nowadays, the prevalent choice to handle the adaptation to varying network conditions comprises the generation of several versions of the compressed video at different bit rates and spatial dimensions. During the streaming session, the server monitors the connection to decide which is the best version to use. The server can jump from one to another version at periodic points in the video timeline. Less frequent is the use of real-time video transcoding to perform a fine-grain adaptation. Nevertheless, both solutions involve a prohibitive amount of storage or processing resources at the server side.

Concerning the transport technology, it has been extensively demonstrated that traditional protocols like the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) are not well suited for the transmission of multimedia information when network conditions are poor, due to the strong delay and robustness requirements imposed by multimedia communications for an adequate user experience. The usual alternative to the lack of a multimedia-friendly transport protocol is the adoption of an end-to-end model in which the Realtime Transmission Protocol (RTP)[1] is encapsulated in UDP, and any auxiliary apparatus for the video transmission is performed in a non-standard way at the application level,

*Correspondence: jordi.ortiz@um.es
†Equal contributors
Facultad de Informática, University of Murcia, Campus de Espinardo, Murcia 30100, Spain

including flow and congestion control, retransmission, and redundancy, if any.

The combination of H.264/SVC [2] and the Stream Transmission Control Protocol (SCTP) [3] could change substantially the current panorama. This paper is the first study, to the best of our knowledge, focused on demonstrating the benefits of this synergy.

The first point of interest in this study is the scalable video codec (SVC), a standardized extension of the well-known H.264/AVC [4] video codec. SVC enables the generation of a video coded bitstream from which a set of different video representations, defined by operational points in the spatial, temporal, and quality video dimensions, can be extracted. Each of them is characterized by a bandwidth requirement. For streaming applications, the server can select the appropriate video representation to be delivered according to an updated description of the streaming context that includes the available bandwidth and the end device features (screen size, for example). The selected video representation is structured as a set of layers dependent in an incremental way. For video transmission, the set of video layers can adequately be split in different streams according to their importance in the hierarchical relationship.

The second focus of interest is the SCTP protocol, a general-purpose transport protocol which is able to provide a reliable full-duplex transmission with flow and congestion control, multi-streaming, and multi-homing. The latter is a technique used to improve the reliability of the connection between two endpoints: a peer process involved in an SCTP association can employ several Internet Protocol (IP) addresses corresponding to different network interfaces. The SCTP baseline uses a primary interface for transmission and the others for backup, but the concurrent multipath transfer (CMT)-SCTP extension [5] can use load balancing between all the interfaces. Regarding multi-streaming, the data can be divided into streams delivered with an independent flow control, thus reducing the impact of the head-of-line blocking TCP problem.

SCTP and SVC can be combined through the use of multi-streaming and multi-homing mechanisms to deliver appropriately the set of video layers to the receiver. These new features have a very promising application in mobile devices with multiple network interfaces, cellular and wireless. This paper examines different strategies to transmit SVC with SCTP and shows how they exceed the performance obtained with TCP and UDP in the same network conditions. The study is carried out with the ns-2 network simulator [6], using an implementation of SCTP provided by the University of Delaware [7].

The remainder of this work is structured in the following sections. The section 'H.264/SVC background' contains a brief review of SVC design principles and main features. The section 'SCTP background' contains the corresponding description of SCTP and its CMT extension used in this study. The next section 'Related study' presents a short review of previous papers that study SVC video transmission based on the availability of multiple network paths between sender and receiver. In the section 'Evaluation scenarios', there is an explanation of the ns-2 simulation scenarios that have been used. The section 'Evaluation results' exposes the outcomes obtained. Finally, conclusions are presented in the last section.

H.264/SVC background

This section contains a brief technical description of SVC, the scalable extension of the H.264/AVC [4] video standard. For additional information on the fundamentals of SVC, the reader is referred to [2,8]. SVC defines a scalable video compression with backward compatibility with advanced video coding (AVC). In this study, we assume that the reader is familiar with AVC.

Scalable coding has a long tradition in compression standards. In a certain way, progressive modes of JPEG were a form of scalable coding, and MPEG-2 SNR, spatial, and high profiles were designed to permit the generation of an enhanced video.

A scalable bitstream consists of a number of layers, including a base layer and one or more enhancement layers [9]. The base layer can be decoded independently and provides the elementary video quality. A higher quality can be obtained by decoding the base layer plus enhancement layers, improving the perception of the decoded sequence as more enhancement layers are used. There are three modes of video scalability: spatial scalability, temporal scalability, and quality or signal-noise-ratio (SNR) scalability.

When using spatial scalability, the base layer is coded at a low spatial resolution, and enhancement layers give progressively higher spatial resolution. With temporal scalability, layers improve the frame rate. Take into account that temporal scalability is available in any video codec that enables a hierarchical structure of reference frames in a group of pictures, but SVC includes control headers to identify temporal layers. Quality scalability generates layers with progressively higher picture fidelity. The base layer contains a strongly compressed version of each picture, and enhancement layers incorporate more information to increase the SNR value. These three scalability dimensions can be combined together in a three-dimensional coordinate space. Note that a scalable video can use any combination of the three dimensions. For instance, it is possible to use quality and temporal scalability, and no spatial scalability at all.

SVC inherits from AVC the division of the codec design in two main blocks: the video coding layer (VCL),

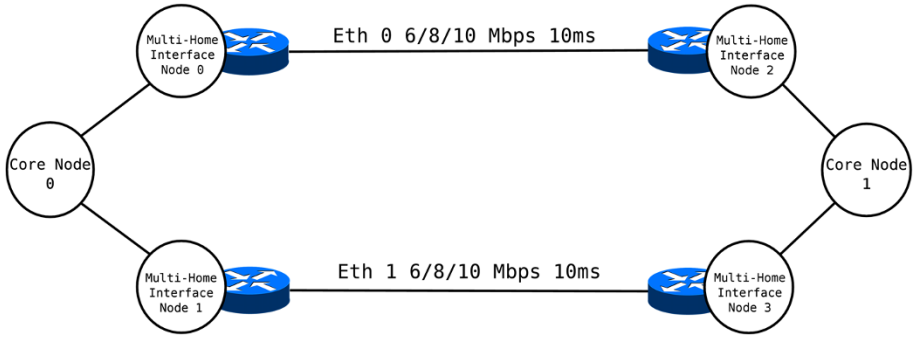


Figure 1 Ns-2 simulation scenario.

which is in charge of the source video coding, and the network abstraction layer (NAL), which is dedicated to the adaptation of the bitstream produced by the VCL to networking or storage. In this study, we can limit our attention to the NAL subsystem, as it provides enough information to identify the data pertaining to each layer. This is, precisely, the advantage of the VCL and NAL separation.

The elementary concept at the NAL level is the NAL unit (NALU). An AVC NALU contains a header followed by a raw byte sequence payload. This header is made of a unique byte and includes a field to identify the type of NALU. A complete picture can be segmented into multiple NALUs, each carrying usually a coded slice or control information. The set of NALUs needed to decode a complete picture is called access unit. AVC introduces a special type of access unit called instantaneous decoding refresh (IDR), which can be identified through the inspection of the NALU header. A sequence of access units starting with an IDR can be decoded independent of any previous pictures in the bitstream. The maximum NALU size can be configured as a parameter of the encoder in order to be adjusted to the maximum transmission unit (MTU).

SVC is backwards compatible with AVC. The base layer of a scalable video is represented with a set of NALUs that can be decoded by any compatible AVC decoder. Additional NALU types are defined by SVC. The main characteristic of these new NALUs is the use of a sequence of 3 bytes, following the AVC header, that contain three identifiers: temporal identifier (TID), dependency identifier (DID), and quality identifier (QID). These identifiers represent a point in the temporal, spacial, and quality scalable dimensions, respectively. The inspection of these fields permits to identify NALUs belonging to a specific enhancement layer. SVC access units start with base layer NALUs followed by enhancement layer NALUs, which are organized with increasing values in the (DID, QID, TID) triple identifier. The exact number of NALUs per access unit depends among others on the number of enhancement layers.

A special SVC control NALU at the beginning of the

Table 1 Strategy overview

Strategy	Multi-streaming	Reliable	Unreliable	Mixed
CMTMultiReliable	√	√	○	○
CMTMultiUnreliable	√	○	√	○
CMTMultiMixed	√	○	○	√
SCTPMultiMixed	√	○	○	√
SCTPMultiReliable	√	√	○	○
SCTPMultiUnreliable	√	○	√	○
SCTPReliable	○	√	○	○
SCTPUnreliable	○	○	√	○
RTPMulti	√	○	√	○
RTP	○	○	√	○
TCP	○	√	○	○

Table 2 No error

Strategy	30 Mb		45 Mb		60 Mb	
	Avg diff time	PSNR	Avg diff time	PSNR	Avg diff time	PSNR
CMTMultiReliable	0.0057	34,4653	0.0056	34,4653	0.0055	34,4653
CMTMultiUnreliable	0.0057	34,4653	0.0056	34,4653	0.0055	34,4653
CMTMultiMixed	0.0057	34,4653	0.0056	34,4653	0.0055	34,4653
SCTPMultiReliable	1.6108	34,4653	0.0172	34,4653	0.0061	34,4653
SCTPMultiUnreliable	1.6108	34,4653	0.0172	34,4653	0.0061	34,4653
SCTPMultiMixed	1.6108	34,4653	0.0172	34,4653	0.0061	34,4653
SCTPReliable	1.6105	34,4653	0.0171	34,4653	0.0060	34,4653
SCTPUnreliable	1.6105	34,4653	0.0171	34,4653	0.0060	34,4653
RTPMulti	0.0061	33,4609	0.0052	34,4653	0.0052	34,4653
RTP	0.0094	29,8153	0.0059	33,6131	0.0052	34,4653
TCP	2.4409	34.4653	0.2865	34.4653	0.0583	34.4653

bitstream contains metadata about the scalable structure of the video. It is the scalability info, a description of the enhancement layers of the video that includes the resolution of the spatial layers, the frame rate of the temporal layers, and the average bandwidth required to transmit the video up to each layer. This information can be used at different control points in the streaming system in charge of doing rate shaping of the video. It could be the streaming server or a media-aware network element located at the frontier between two network domains, one with a big bandwidth and the other with a more limited one. No transcoding is necessary but a simple discarding of NALUs. Dynamic switching between enhancement layers is possible during the streaming session at IDR access units. Scalable information can also be used when doing forward error correction or any other technique to improve the robustness of the more sensitive data. The base layer can be transmitted with more care than any other layer as it is the more important piece of information required to perform a correct decoding.

SCTP background

SCTP is a general-purpose transport protocol initially conceived for conveying telephony signaling messages over IP best-effort networks. The baseline SCTP, defined in RFC 4960 [3], offers a transport service that aims to solve some well-known problems of TCP that affect the performance of highly delay-constrained services. This section is dedicated to the description of the main features of SCTP and CMT-SCTP. For a complete description of the protocol, the reader is referred to [10]. An excellent survey about the SCTP research can be found in [11].

Protocol bases

SCTP is located at the transport level in the Internet network architecture and works directly over IP. It permits to

Table 3 Uniform 10^{-4}

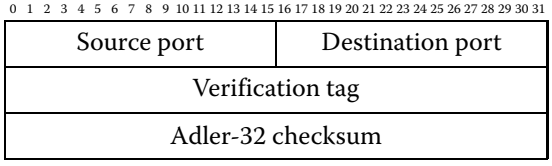
Strategy	30 Mb		45 Mb		60 Mb	
	Avg diff time	PSNR	Avg diff time	PSNR	Avg diff time	PSNR
CMTMultiReliable	0.0057	34,4653	0.0056	34,4653	0.0055	34,4653
CMTMultiUnreliable	0.0057	34,4653	0.0056	34,4653	0.0055	34,4653
CMTMultiMixed	0.0057	34,4653	0.0056	34,4653	0.0055	34,4653
SCTPMultiReliable	1.6146	34,4653	0.0174	34,4653	0.0062	34,4653
SCTPMultiUnreliable	1.6146	34,4653	0.0174	34,4653	0.0062	34,4653
SCTPMultiMixed	1.6146	34,4653	0.0174	34,4653	0.0062	34,4653
SCTPReliable	1.6149	34,4653	0.0172	34,4653	0.0061	34,4653
SCTPUnreliable	1.6149	34,4653	0.0172	34,4653	0.0061	34,4653
RTPMulti	0.0061	33,4229	0.0052	34,4272	0.0052	34,4272
RTP	0.0094	29,7677	0.0059	33,5600	0.0052	34,4159
TCP	2.4581	34,4653	0.2865	34,4653	0.0583	34,4653

use a connection-oriented service, similar to that of TCP but bypassing some of its limitations such as the head-of-line blocking or the restriction of using single-homed connections. Hence, SCTP offers a standardized alternative to the implementation of application-dependent reliable data transfer protocols on top of UDP.

The protocol employs the concept *association* to manage the relationship between two *endpoints*. At a given time, only one SCTP association can exist between two SCTP endpoints. The association is established with a four-way handshake based on a cookie mechanism that protects against synchronization attacks. An SCTP endpoint on a multi-homed host is a combination of transport addresses from which SCTP packets can be received. All transport addresses involved in an SCTP endpoint have the same port but can use multiple IP addresses. Participating addresses are notified to the other end during the association setup. This multi-homing feature is used as a fault-tolerant mechanism for the association. An SCTP peer monitors which of the addresses at the other end are available for receiving user messages by means of special control messages called *heartbeats*. Responding addresses are considered *active* and thus available for communication. One of them will be used as the *default* destination so that there will be a *primary path* between both ends, but if there is a failure detection - the primary address stops sending back acknowledgments - a different destination address can be used to generate a backup path, and the communication can continue.

Additionally, SCTP permits to bind user messages to *streams* in the context of an association. During the association setup, both ends negotiate the number of streams that will be used. A stream is a unidirectional sequence of messages that SCTP must deliver in order to the upper layer at the receiving end. Streams are implemented with stream identifiers and stream sequence numbers that are attached to user messages. When a transmission error occurs, the negative effects will be restricted to the streams involved in the packet loss. As a consequence, the increase in the delay due to retransmissions (head-of-line blocking) will not affect other streams. SCTP permits sending messages that bypass the normal ordered delivery. These messages are delivered as soon as they are received.

SCTP packets contain a common SCTP header, followed by one or more *chunks*. The fields of the common SCTP header are the following:



The verification tag provides protection against blind attackers and a mechanism to discriminate packets from a previous association between the same pair of endpoints. Chunks following the common header may carry user data or control signaling, and a single SCTP packet may transport user data associated with multiple streams. We restrict our attention to some of the fields of data chunks:

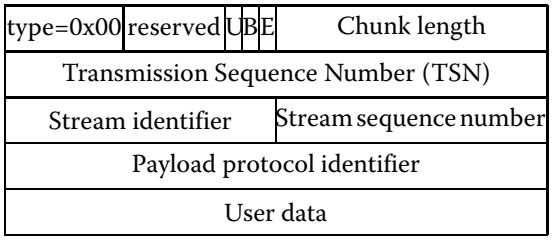


Table 4 Uniform10^{-3}

Strategy	30 Mb		45 Mb		60 Mb	
	Avg diff time	PSNR	Avg diff time	PSNR	Avg diff time	PSNR
CMTMultiReliable	0.0244	34,4653	0.0101	34,4653	0.0057	34,4653
CMTMultiUnreliable	0.0300	34,4596	0.0101	34,4653	0.0057	34,4653
CMTMultiMixed	0.0300	34,4596	0.0101	34,4653	0.0057	34,4653
SCTPMultiReliable	1.9877	34,4653	0.0458	34,4653	0.0556	34,4653
SCTPMultiUnreliable	1.9877	34,4653	0.0458	34,4653	0.0556	34,4653
SCTPMultiMixed	1.9877	34,4653	0.0458	34,4653	0.0556	34,4653
SCTPReliable	3.0093	34,4653	0.0466	34,4653	0.1124	34,4653
SCTPUnreliable	2.0130	34,4653	0.0466	34,4653	0.1124	34,4653
RTPMulti	0.0061	32,9780	0.0052	33,9753	0.0052	33,9753
RTP	0.0094	29,2786	0.0059	33,0309	0.0052	33,8717
TCP	2.9667	34,4653	0.4165	34,4653	0.1325	34,4653

Table 5 Uniform 10^{-2}

Strategy	30 Mb		45 Mb		60 Mb	
	Avg diff time	PSNR	Avg diff time	PSNR	Avg diff time	PSNR
CMTMultiReliable	1.3272	34,4653	0.0644	34,4653	0.6181	34,4653
CMTMultiUnreliable	0.4039	34,4372	0.7470	34,4417	0.4098	34,4367
CMTMultiMixed	1.7904	34,2869	0.7470	34,4417	0.4026	34,4427
SCTPMultiReliable	41.1406	34,4653	20.3130	34,4653	3.5188	34,4653
SCTPMultiUnreliable	17.2343	33,9208	12.4975	34,1825	13.0626	34,1603
SCTPMultiMixed	17.2308	33,9577	12.4961	34,1896	13.0586	34,1666
SCTPReliable	44.7756	34,4653	19.2490	34,4653	9.7437	34,4653
SCTPUnreliable	18.2282	33,9311	12.1949	34,1579	11.1983	34,0951
RTPMulti	0.0061	29,5685	0.0052	30,5368	0.0052	30,5368
RTP	0.0094	24,7738	0.0059	28,5860	0.0052	29,3284
TCP	37.1808	34,4653	25.5925	34,4653	23.5196	34,4653

Bit U indicates that the user data carried in the chunk is unordered, meaning that the receiver can deliver the data to the application without reordering. In order to implement acknowledgments and loss or duplicate detection, data chunks have a unique transmission sequence number that is maintained at the association level. Regarding the size of user messages, SCTP may fragment a message that causes the packet to exceed the path MTU. Bit B indicates the beginning part of a user message that is fragmented, and bit E indicates the ending part. The receiving end will reassemble fragments in order to keep the message boundaries when data are finally delivered to the upper layer. As a consequence, SCTP is a *message-oriented* protocol, similar in this aspect to UDP, instead of a stream-oriented protocol such as TCP.

SCTP employs a *flow control* mechanism that is nearly the same as the one used in TCP. The algorithm is based on the use of a *receiver window* (rwnd) variable at the sender side. This variable gives an indication of the available buffer space at the receiver end. During the association setup, each endpoint notifies its reception buffer size, and rwnd is initialized with this value. When a data chunk is transmitted, the sender endpoint subtracts the size of

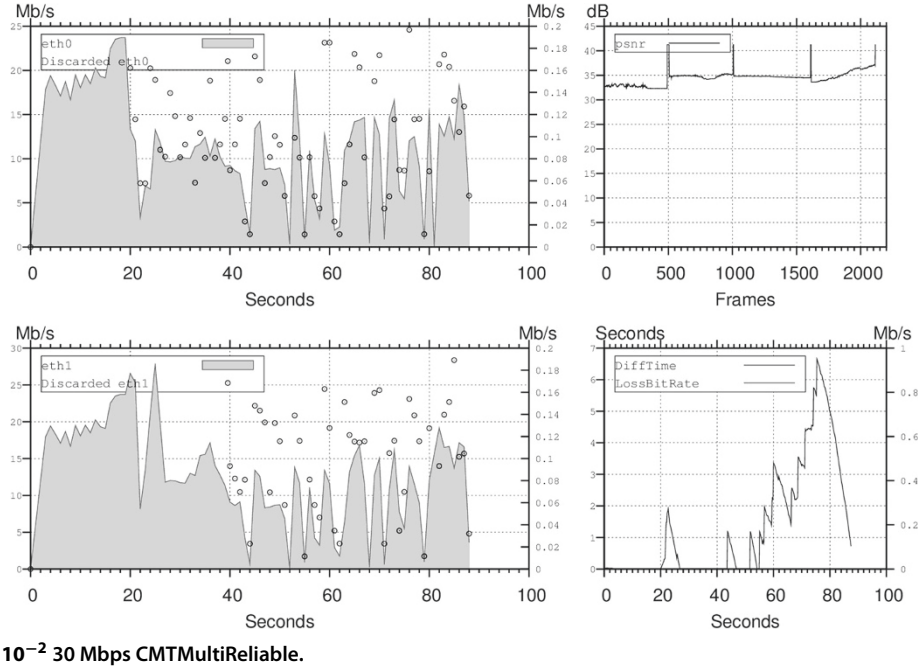


Figure 2 Uniform 10^{-2} 30 Mbps CMTMultiReliable.

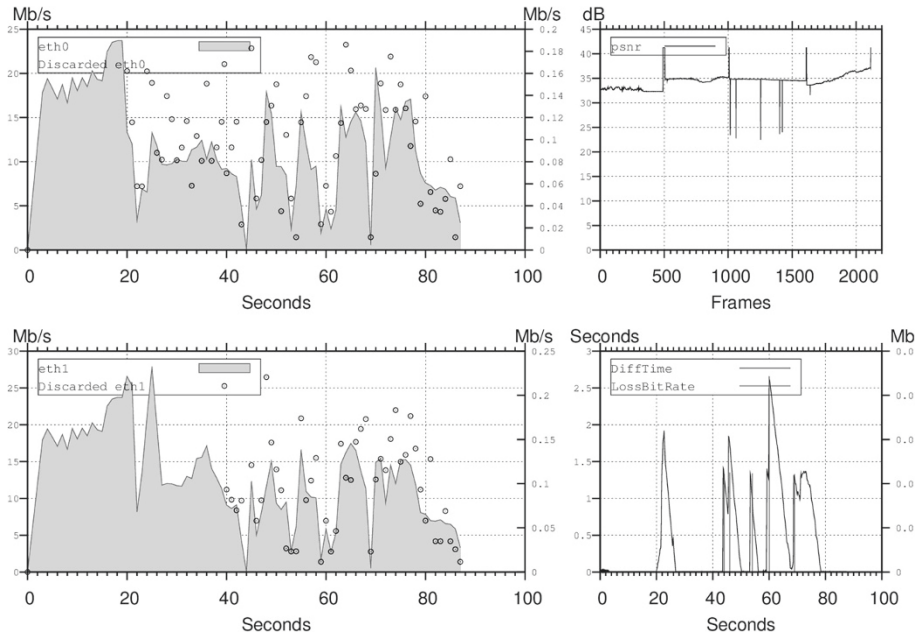


Figure 3 Uniform 10^{-2} 30 Mbps CMTMultiUnreliable.

the data from the current value of rwnd. A special type of control chunk, called selective acknowledgement (SACK), is used by the receiver to inform about data reception. At the sender end, SACKs are processed to check if the rwnd must be increased. Three pieces of information included in the SACK are used to update the rwnd value. The first one is called *advertised receiver window* (a_rwnd), and it indicates the available space in the reception buffer. As the receiver buffer is filled with data chunks, the a_rwnd variable is decreased, and it is increased when chunks are delivered to the upper level. The second one is the cumulative TSN ack, that is, the largest TSN received in sequence. The third piece of information is a list of blocks of consecutive data chunks that have been received after a gap of missing chunks. When a SACK is received, the cumulative TSN and the list of gap blocks are used to calculate the amount of *outstanding bytes*, that is, the total size of chunks already sent and not yet acknowledged. The rwnd variable is updated with the a_rwnd value minus the outstanding bytes. At any time, if rwnd equals zero, the sender stops sending data. Nevertheless, if there is no congestion, the sender can have one outstanding data chunk per round trip time to force the reception of a_rwnd updates.

The *congestion control* algorithm used in SCTP employs the well-known strategy called *additive increase and multiplicative decrease* used in TCP, but with some modifications needed due to its multi-homing nature. The congestion control uses three variables to regulate the transmission: the *congestion control window* (cwnd), the *slow-start threshold* (ssthresh), and the *partially acknowledged bytes* (partial_bytes_acked). The main difference between the congestion control algorithms used in TCP and SCTP stems from the fact that SCTP uses separate cwnd and ssthresh variables for each of the destination addresses, and as a consequence, the congestion control is applied independently to each path. The algorithm has two main states per destination: *slow-start* and *congestion avoidance*. The initial slow-start state probes the path to determine the available capacity. The cwnd variable keeps an approximation of the amount of data that can be injected into the path before causing congestion. This value is used to limit how much outstanding data can fly to the destination address. The amount of outstanding data is called *flightsize*. At any given time, the sender must not transmit new data if the flightsize is greater or equal to the cwnd of the destination. During the slow-start phase, the cwnd has a minimal value of 1 MTU, and the ssthresh could be initialized to the receiver window. The value of cwnd increases when an incoming SACK advances the cumulative TSN point. Take into account that SCTP associations can have multiple destination addresses, and as a consequence, a SACK received from one destination address may acknowledge data sent to other addresses. If some of the acknowledged data were sent to a destination address and the congestion window is being fully utilized (the flightsize is greater or equal to the cwnd), the sender increases cwnd proportionally to the amount of acknowledged data sent to that destination. When the cwnd reaches the ssthresh, there is a

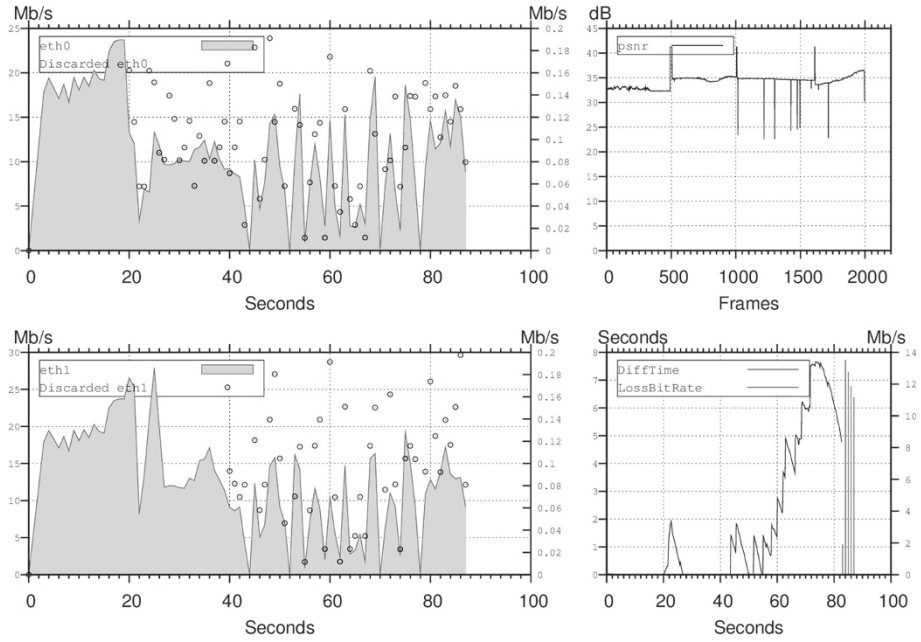


Figure 4 Uniform 10^{-2} 30 Mbps CMTMultiMixed.

transition to the congestion avoidance state. During congestion avoidance, the cwnd is incremented only by 1 MTU when the congestion window is fully utilized and a SACK increases the cumulative TSN point. The auxiliary variable partial_bytes_acked helps in the implementation of this mechanism, counting the amount of bytes acknowledged since the last update of the cwnd variable.

The multiplicative decrease of the congestion window is activated when a packet loss is detected. One of the mechanisms to detect packet losses is based on the retransmission timers. Each time a data chunk is sent to a destination address, the retransmission timer of that destination is initialized with the value of the estimated retransmission timeout, computed with the smoothed round-trip time and variation of the corresponding path. If the retransmission timer expires, SCTP assumes a severe congestion problem causing the congestion control to go back to the slow start state. Additionally, the ssthresh is reduced to a half of the cwnd, and the cwnd is reset to 1 MTU. On the other hand, if the loss is detected with the inspection of gap blocks in a received SACK, both the ssthresh and the cwnd are reduced to a half of the previous value of cwnd. If three consecutive SACKs report the same missing TSNs, *fast retransmission* is activated. In this case, the sender determines how many chunks marked for retransmission fit in a single packet, and this packet is sent ignoring the value of cwnd and without delay. Then, the transmission enters in *fast recovery* mode, and the highest outstanding TSN is marked as the fast recovery exit point. While in this mode, the ssthresh and cwnd should not be reduced due to subsequent fast recovery events. The fast recovery mode is exited when all TSNs up to and including the exit point are acknowledged.

Concurrent multipath transfer

CMT is a proposed extension for SCTP designed to enable simultaneous data delivery through all the available paths between a pair of source and destination endpoints. The extension, proposed by researchers from the University of Delaware, is fully described in [5]. The goal pursued by CMT is the increase in fault tolerance and in global transmission performance with respect to the baseline SCTP, where only one path is used at a time. The extension is designed under the assumption that the bottleneck queues on the end-to-end paths are independent. This assumption is important to maintain a TCP-friendly flow and congestion control over all the available paths.

CMT schedules the transmission of new data through the available paths as soon as the corresponding cwnds allow it. When there is space for transmission simultaneously for several destinations, data are sent uniformly through all of them. However, taking into account that each path has different delay and bandwidth characteristics, this simultaneous transmission incurs in a significant packet reordering at the receiver end. This effect, connatural to CMT, provokes a substantial degradation in the performance of the basic algorithms of SCTP. In [5], three negative effects of packet reordering are identified: unnecessary fast retransmissions triggered by an increase in the number of gap reports, a slow increase of cwnd due

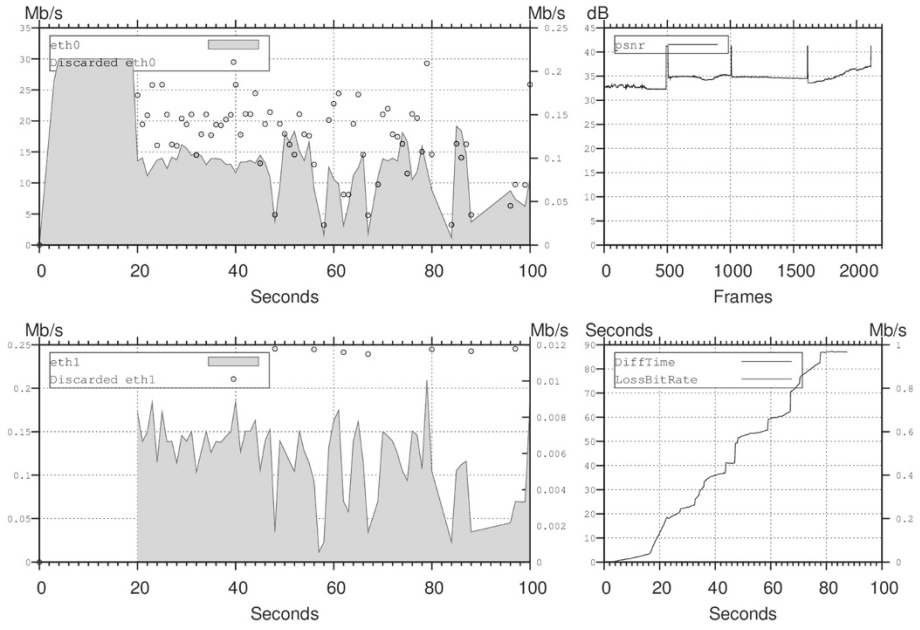


Figure 5 Uniform 10^{-2} 30 Mbps SCTPMultiReliable.

to the lack of cumulative TSN acknowledgements, and an increase of acknowledgment traffic due to immediate delivery of SACKs when out-of-order data are detected at the receiver end.

CMT uses three new algorithms to cope with these problems. The *split fast retransmission* (SFR) algorithm keeps an independent virtual queue for each path within the retransmission buffer. Additional variables associated with each virtual queue permit to apply the fast retransmission procedure on a per destination basis, keeping track of the path that was used to transmit each TSN. The *Cwnd Update for CMT* (CUC) algorithm allows to increase the destination cwnd with SACKs that do not advance the cumulative TSN point. The algorithm tracks the earliest outstanding TSN per path and updates the path's cwnd just with the information contained in gap blocks, even if the cumulative TSN is not modified. The *Delayed Ack for CMT* (DAC) algorithm is designed to reduce the acknowledgment traffic. The baseline SCTP protocol is designed with the assumption that data received out-of-order indicates possible loss, and accordingly, the receiver should immediately send a SACK with a gap report to trigger fast recovery as soon as possible. The DAC algorithm specifies a different receiver and sender behavior to infer when a gap report is caused by loss and not reordering. In [5], the authors indicate that the combination of the three mentioned algorithms improves the aggregate cwnd growth in comparison with multiple SCTP associations (one per path) between the sender and the receiver. The same article explores different retransmission policies and concludes that it is beneficial to send retransmissions to the destination with the largest cwnd or ssthresh instead of using the same destination employed in the initial transmission.

Related study

During the last years, there is an increase in the amount of research done to explore the delivery of scalable video to multi-homed networks or end devices. Most of the papers found investigate scheduling algorithms that select which path should be used to transmit SVC data according to the dynamic network conditions in order to optimize the quality of the received video. In [12], the authors describe a packet scheduling algorithm for the delivery of SVC to multi-homed mobile networks. The transmission protocol used is RTP, and the scheduling is done according to the priority signaling in NAL units. The algorithm proposed offers an improvement in peak signal-to-noise ratio (PSNR) over the generic packet selection and scheduling for multi-path video streaming proposed in [13]. Another work that employs RTP as the transport protocol is [14]. In this work, it is described as a complete prototype implementation in which a control and signaling architecture improves the delivery of SVC layers according to the output provided by an adaptation and decision engine that employs cognitive techniques. The adaptation mechanism is integrated in a complete streaming architecture following the classical IETF standards for video streaming. An experimental evaluation using a client with two wireless

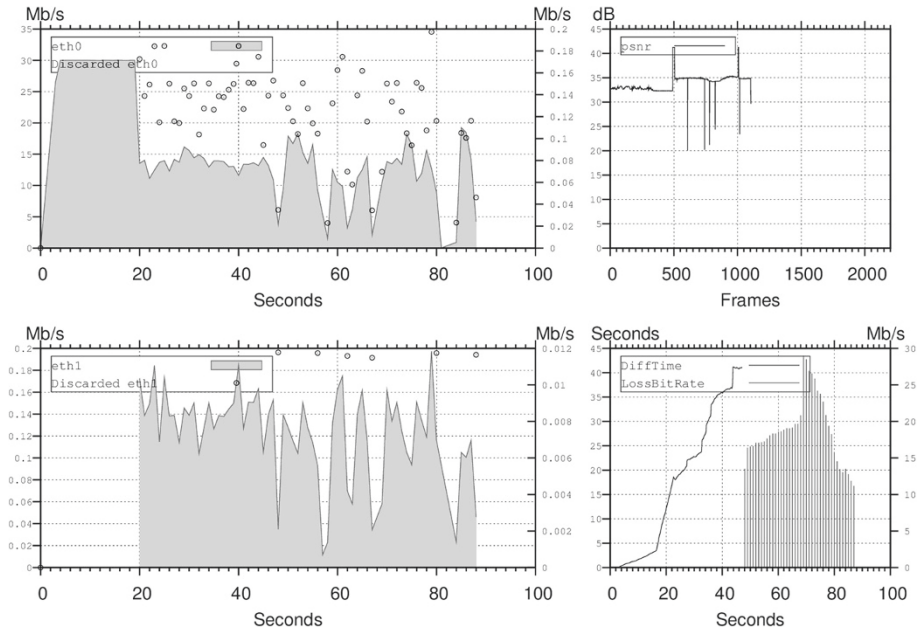


Figure 6 Uniform 10^{-2} 30 Mbps SCTPMultiUnreliable.

interfaces is presented. In [15], a SVC streaming system over mobile *ad hoc* networks is proposed in which multiple routes between source-destination pairs permit to select backup paths in case the current one is broken. The transport protocol used is UDP. However, UDP and RTP are not the unique transport protocols proposed in the literature about SVC and multi-path networks. The work presented in [16] describes an algorithm that optimizes resource allocation in a streaming scenario with multiple clients and access networks. The algorithm determines the streaming rate over each network, the video packets to be transmitted, and the access network to send each video packet. Tests are performed using the DCCP protocol [17], a transport protocol that provides bidirectional

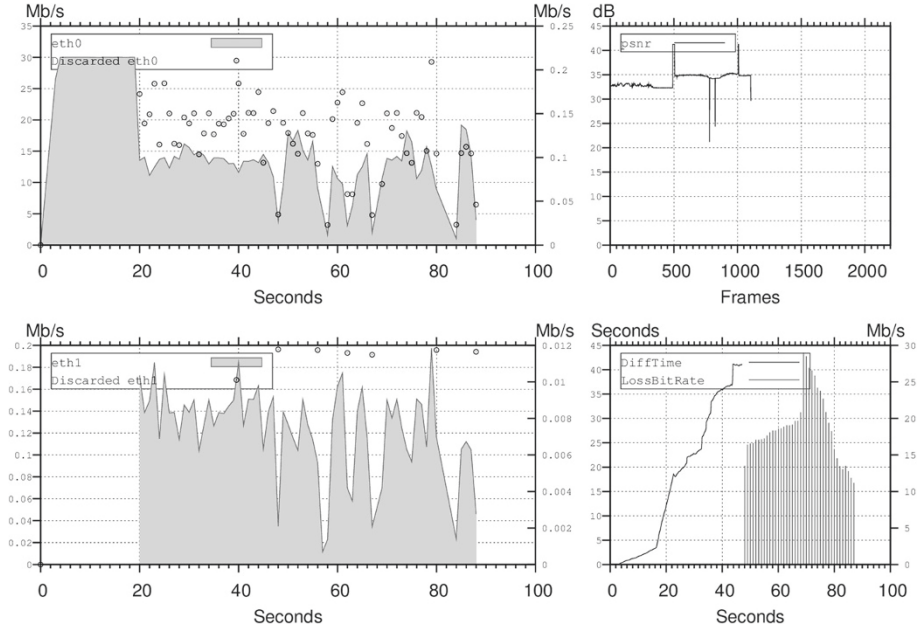


Figure 7 Uniform 10^{-2} 30 Mbps SCTPMultiMixed.

unicast connections of congestion-controlled unreliable datagrams but does not support multi-streaming or multi-path delivery.

Evaluation scenarios

In this section, we aim to describe the simulation environment employed to demonstrate the advantages of using SCTP as the SVC video transport protocol in comparison with other protocols (RTP and TCP) currently used. The main strategy to show the outperformance of SCTP is based on sending SVC layers as SCTP streams over a SCTP association while also using the multi-homing features of the protocol. The scenario in which the comparison is established employs extremely lossy networks. The simulations are carried out using ns-2 (v. 2.34), which includes the implementation of the SCTP transport protocol with Delaware's University extensions.

In order to implement our source agent, we use trace files generated from real mp4 files created with a modified version of *mp4creator* developed in the SCALNET project[18]. This tool permits to employ the mp4 file format as the container of SVC video, according to the ISO standard [19]. We use mp4 files that contain, in addition to the actual bitstream, some extractor and RTP hint tracks that ease the streaming process done by the server. The generated traces contain the size of each RTP packet, the packet number within the frame, the frame number, the sequence number, and a stream identifier that is used within SCTP. We build two types of video source, one taking only one stream as input and the other taking multiple streams. We have also developed a video sink that generates an output trace of the received video events. This trace permits us to generate, using the original mp4 file, SVC files in which all the lost NALUs are removed. Instead of generating one SVC file to represent the received video for the whole transmission, we produce one SVC file per frame. With this approach, we try to have a fine grain control over the decoding process to know whether a particular frame has been successfully decoded, or there is not enough information to decode it at all. In case a particular frame is not decoded, we replicate the last decoded frame. With this, we can calculate the PSNR value of each frame with the certitude that it is aligned with the original video sequence. Obviously, this decision lead us to encode our test stream with I frames exclusively.

We employ a common video sequence for the complete set of transmission scenarios. The sequence is the union of *Parkrun*, *Shields*, *Stockholm*, and *Mobcalm* subsequences. We have encoded them using the Joint Scalable Video Model (JSVM) reference software, version 9.19. There are two layers, the base layer and one additional quality enhancement layer, with average bit rates of 6.8 and 23.5 Mbps, respectively. The video is VBR encoded, causing some severe variations in the transmission rate that globally rise over 45 Mbps at some segments of the video. The duration of the encoded stream is 2,116 frames streamed at 25 frames per second, permitting a test of 88.64 s. The generated SVC file has an average PSNR-Y of 34.46 dB. The sequence is encapsulated in an mp4 file and hinted with an MTU limit of 1,460 bytes. Transmission using TCP is treated as a special case because it is not a message-oriented protocol. The TCP agent collocated with the video sink agent produces reception events indicating amounts of received bytes which do not respect NALU blocks. In this case, the agent waits until all the bytes for a NALU have been received before including it in the reception trace.

The simulated scenario is represented in Figure 1. It is a simple network with two end systems connected by two duplex links. These systems are actually modeled with three nodes each (a core node and two nodes for the network interfaces), as proposed by CMT-SCTP module authors [20]. Each core node employs a transport agent. In order to establish a comparison, we use SCTP, SCTP/CMT, TCP/FullTCP, and RTP transport agents. At the sender side, the transport agent is attached to a video source application that reads information from the pre-generated video trace. At the receiver end, the transport agent connects to a video sink that generates the reception trace. We left most of the SCTP parameters in ns-2 with their default values. Nevertheless, there are some exceptions that have been modified. The *pathMaxRetrans_* and *changePrimaryThresh_* govern the decision of which interface to use in case a retransmission has to be done. We set both parameters to 1, implying that after a packet loss, the SCTP agent will resend it on the secondary interface. Additionally, we set the *Reliability_* parameter to 0 when testing the use of the SCTP protocol in unreliable mode, which is basically a best effort approach similar to UDP, but using congestion control. Nevertheless, even in the unreliable mode, we have configured the SCTP protocol to send at least one retransmission for each loss. The results of each simulation include the following information, averaged per second:

- *Diff. time*, that is, the delay experimented by video packets in the transit from source to sink.
- *Bandwidth* used in each interface.
- *Discarded packets* detected in each interface, due to router congestion or link failure.
- *Loss bandwidth* detected at the video sink, after packet retransmissions.
- *PSNR-Y*, objective measure of the received video quality with respect to the original file. The value is obtained using the *PSNRStatic* tool included in the JSVM reference software.

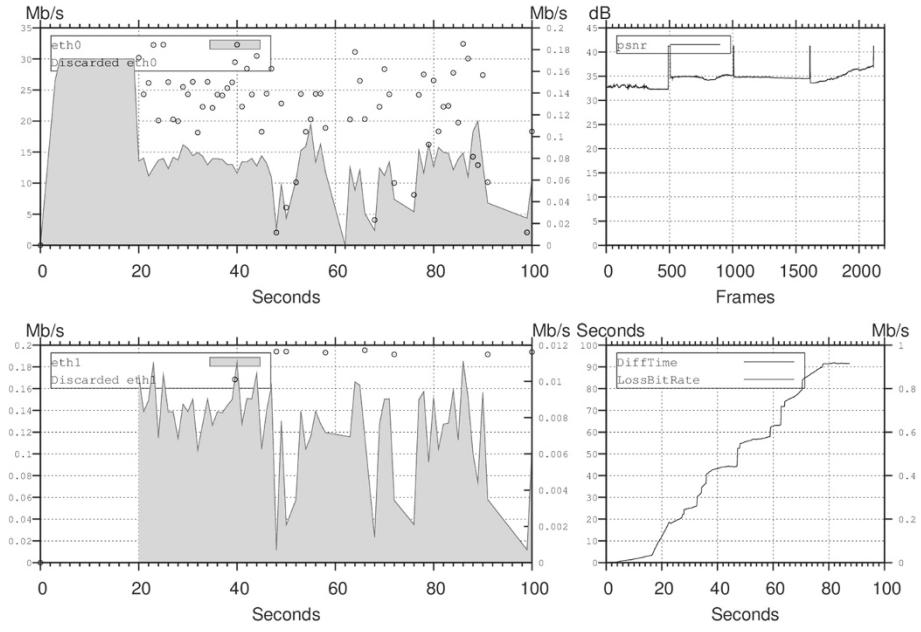


Figure 8 Uniform 10^{-2} 30 Mbps SCTPReliable.

Simulations use three different link bandwidths: 30, 45, and 60 Mbps. Both links have the same bandwidth. Each bandwidth configuration is tested with four different packet error rates (PER): no loss and uniform loses with 10^{-4} probability, with 10^{-3} probability, and with 10^{-2} probability. In a uniform error model, we consider the following relationship between packet error rate and bit error rate (BER):

$$\mathrm{PER} = 1 - (1 - \mathrm{BER})^{\mathrm{MTU}}.$$

With an MTU of 1,500 bytes, the PER values corre spond, roughly speaking, to BER values of 10^{-8}, 10^{-7}, and 10^{-6}. Losses start at the 20th second in the first link and at the 40th second in the second. Both links are configured with a constant delay of 5 ms. For each combination of network bandwidth and error distribution, we have tested 11 different transmission strategies. Due to the amount of

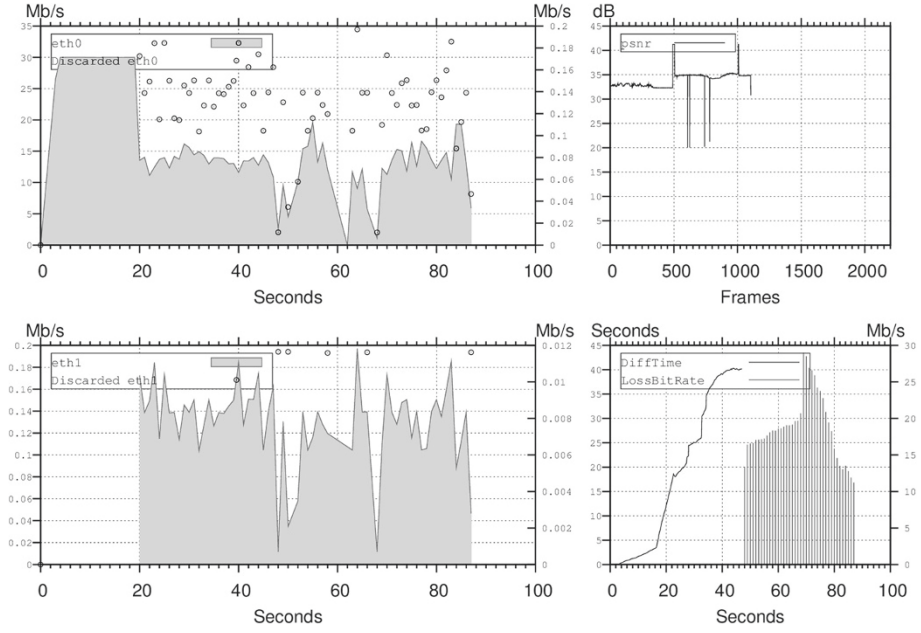


Figure 9 Uniform 10^{-2} 30 Mbps SCTPUnreliable.

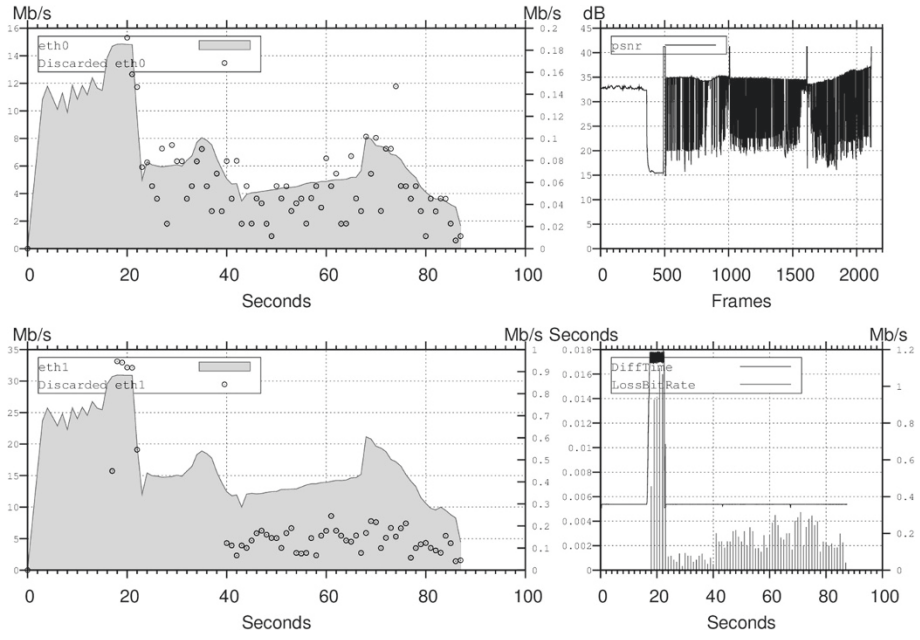


Figure 10 Uniform 10^{-2} 30 Mbps RTPMulti.

combinations, we describe each one with a label that is used in the rest of the paper:

- *CMTMultiReliable*. CMT-SCTP transmission, with base and enhancement layers using different streams, both with reliable transmission
- *CMTMultiUnreliable*. Similar to the previous one but using unreliable transmission for both layers
- *CMTMultiMixed*. Similar to the previous one but using reliable transmission for the base layer and unreliable for the enhancement layer
- *SCTPMultiReliable*. Baseline SCTP transmission, with base and enhancement layers using different streams, both with reliable transmission

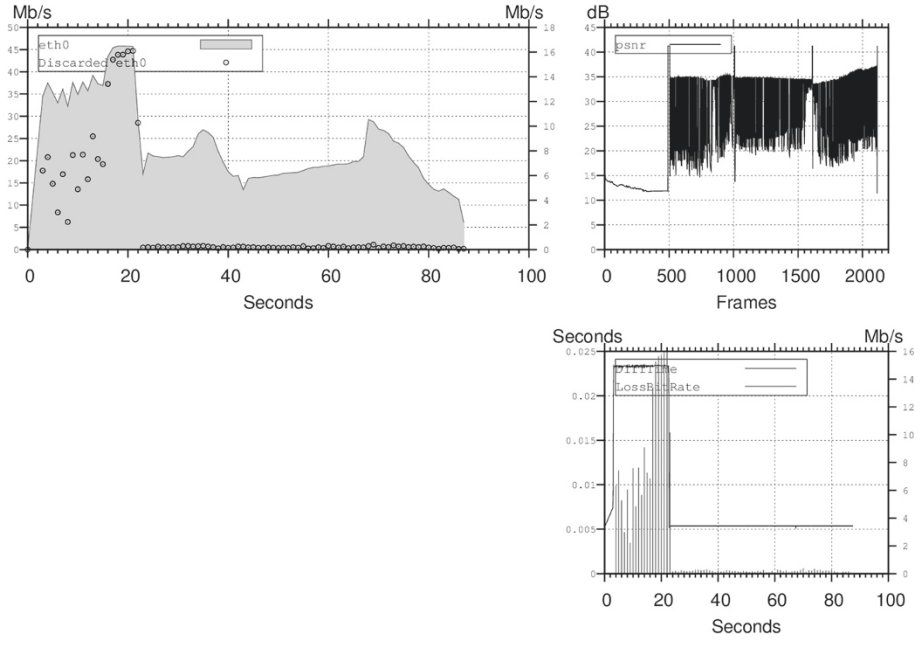


Figure 11 Uniform 10^{-2} 30 Mbps RTP.

- *SCTPMultiUnreliable*. Similar to the previous one but using unreliable transmission for both layers
- *SCTPMultiMixed*. Similar to the previous one but using reliable transmission for the base layer and unreliable for the enhancement layer.
- *SCTPReliable*. Baseline SCTP transmission, with base and enhancement layers using the same stream with reliable transmission
- *SCTPUnreliable*. Similar to the previous one but using unreliable transmission
- *RTPMulti*. RTP transmission, with base and enhancement layers being sent through the first and second interfaces, respectively
- *RTP*. Similar to the previous one but using only the first interface
- *TCP*. TCP transmission with both layers sent through the first interface

Table 1 summarizes the characteristics of these eleven strategies.

Evaluation results

This section shows the results of the simulations and describes the obtained values. There is an explosion in the combinations of the 11 transmission strategies, four error configurations, and three bandwidth configurations. As a result, only the average values of all the combinations are summarized in Tables 2, 3, 4, and 5, each one corresponding to a loss probability. These tables contain values that measure the average delay and PSNR-Y for each transmission strategy. In the case of TCP, the PSNR-Y value corresponds to that of the complete error-free sequence, as a reference to compare with the best possible case. In addition to this, Figures 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12 show more detailed information for the 30 Mbps bandwidth and 10^{-2} uniform loss rate combination. This scenario represents the worst case of all those studied, with a strong packet loss rate. Although the uniform packet loss is an oversimplification of an error model, it permits to check the situation corresponding to a constant bit error rate of 10^{-6} due to transmission errors, a parameter similar to that of a wireless connection with strong jamming. The objective is to push the transmission to the limit, instead of doing the simulation of a more realistic network model. The scenarios present a congestion situation up to the 20th second, due to peaks in the video bandwidth that surpass the link bandwidth.

Figures 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12 represent a mosaic of three or four graphs for each transmission strategy. The top left plot shows data about the eth0 interface. Similarly, the bottom left plot presents the same information for the eth1 interface (some strategies use only the first one). Both plots present the bandwidth usage on the scale of the left axis of the plot, while the amount of discarded packets is represented with reference to the right axis. The use of two axes tries to ease the practical representation of data that would be impossible in some cases with just one axis. This dual vertical axis is also used in the bottom right plot, in which the average delay of packets is shown using the left axis, whereas the loss bandwidth (data that did not arrive to the video receptor) is shown using the right axis. Finally, the top right plot represents the PSNR-Y of the received video.

As shown in Figures 2, 3, and 4, CMT transmissions use both interfaces fully. All of the other strategies present a clear unbalance in the use of both interfaces. Most of them employ preeminently the first interface, with the exception of RTPMulti, shown in Figure 10, where the second interface is used to send the enhancement layer that has a bigger bandwidth consumption than the base layer. SCTP transmission strategies (Figures 5, 6, 7, 8, and 9) make a sparse use of the second interface, which is reduced to the retransmission of packets that have been lost through the first interface.

Taking a look at Figure 11 corresponding to the RTP scenario, it is possible to see how congestion affects strongly the quality of the transmission when the sending rate exceeds the available bandwidth. When the video bandwidth exceeds 30 Mbps, the loss bandwidth at the reception rises clearly. As expected, RTP has the worst PSNR value but the best average delay value. We have to take into account that packet losses are not retransmitted, so there is no penalization to the delay; however, the negative influence in the quality of the received video is excessive.

Looking at Figure 12, we observe that TCP performs well during the initial phase of the simulation, obtaining the expected result due to the congestion control. However, as the number of transmission errors rises up, the average delay grows prohibitively. On the other hand, TCP has a better delay average than SCTP in scenarios where there is no error but there are congestion periods exclusively.

The SCTP reliable approach outperforms TCP on lossy scenarios and scenarios where the bandwidth is sufficient. Nevertheless, we have observed that the SCTP implementation in ns-2 has a problem regarding the unreliable mode. As can be seen in Figures 6, 7, and 9, the receiver video agent stops receiving data from the SCTP agent approximately at the middle of the simulation. When the receiver window is full, the SCTP agent drops all incoming data chunks, causing the starvation of the video agent. The pathmaxretrans_ parameter set to 1 produces, when the available bandwidth is high and there is a big error rate, an increment in disordered packets for which the baseline SCTP is not well prepared. This is precisely the motivation of the three algorithms SFR, CUC, and DAC employed by the CMT extension. Sending each layer in a different

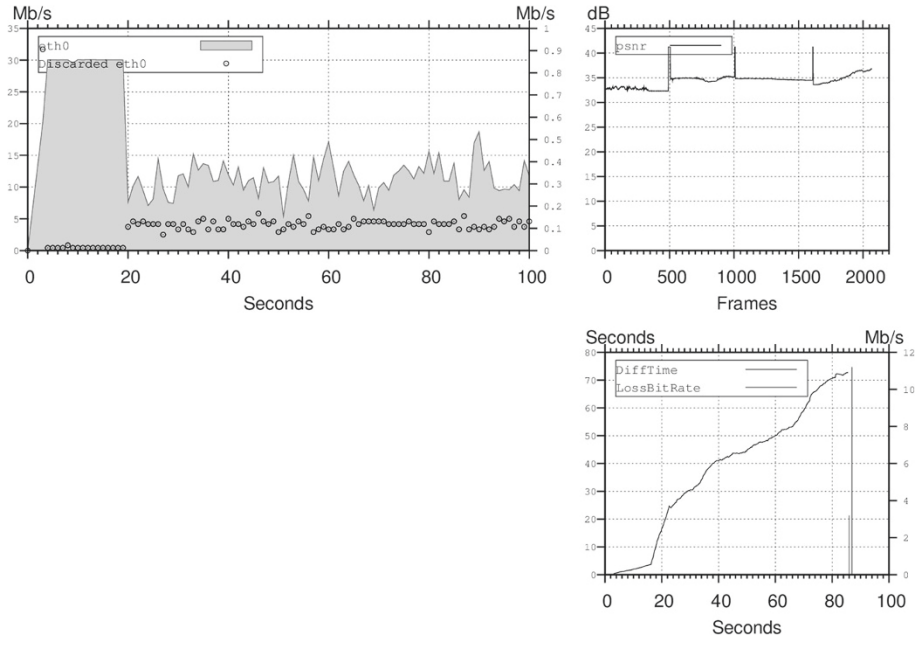


Figure 12 Uniform 10^{-2} 30 Mbps TCP.

stream (SCTPMultiReliable) enhances the transmission performance on lossy scenarios in comparison with the use of a unique stream (SCTPReliable), as can be seen in Tables 2, 3, 4, and 5.

Regarding the unreliable SCTP approach, the results show that there is nearly no difference in the value of the average delay in comparison with the reliable approach. Of course, the big difference comes in relation with the PSNR value. The unreliable approach provokes a reduction in the quality of the video, which is traduced in black square artifacts due to complete video slices lost.

To get a compromise between error resilience and delay, we test scenarios in which the base layer is protected using a reliable transmission while the enhancement layer is sent with an unreliable approach. The results exposed in the SCTPMultiMixed columns in Tables 2, 3, 4, and 5 show that this mixed approach globally reduces the average time with respect to the only-reliable approach while obtaining a greater PSNR than the only-unreliable one. The enhancement of this approach could be measured not only in terms of the delay and objective quality, but also in the quality of experience results for the user, as only packets of the enhanced layer will be affected by losses, reducing the overall impact on the quality of the image.

We studied transmission mechanisms with CMT and the three alternatives in relation with the reliability of the streams (reliable streams, unreliable streams, and a mixed combination of both), and our conclusion is that, thanks to the good performance of using reliable streams in CMT, CMTMultiReliable is always the best choice of all because the delay is kept low and the PSNR is always the maximum.

Conclusions

SCTP is one of the most promising protocols for data transmission in the field of multimedia communications. In addition, the CMT extension is a key technology to take profit of the actual multi-homing environments (networks and devices). On the other hand, SVC is a very interesting codec that, although still not broadly adopted by the industry, has already demonstrated its capabilities through a series of research articles and projects.

This article shows empirically how mixing SCTP streams with SVC layers is a natural and profitable approach for video delivery in high error rate transmissions. This first experimental study will be followed by a deeper study of the multiple configurable parameters of SCTP such as the reliability level for error prone connections or the path and association retransmission limits, among others. The investigation could be improved by incorporating other SCTP extensions like the Potentially Failed SCTP-PF [21], which offers a non-duplicate transport service with tunable loss recovery. In addition to this, a more realistic video approach should be taken using HD video with at least temporal scalability and a more intense use of a number of layers to study the preferred approach for each enhancement type.

Finally, a real implementation of this transmission techniques could permit to contrast the simulation results obtained in this study.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
This work was partially funded by the Spanish Interministerial Comission of Science and Technology (CICYT) and the EU-ICT OPENLAB (FP7-287581) project.

Received: 30 October 2012 Accepted: 29 April 2013
Published: 31 May 2013

References

1. H Schulzrinne, S Casner, R Frederick, V Jacobson, *RFC 3550. RTP: A Transport Protocol for Real-Time Applications*. (Internet Engineering Task Force, Fremont, 2003)
2. T Wiegand, G Sullivan, H Schwarz, M Wien, *ISO/IEC 14496-10:2005/Amd3: Scalable Video Coding*. (International Standardization Organization, Geneva, 2007)
3. R Stewart, *RFC 4960, Stream Control Transmission Protocol*. (Internet Engineering Task Force, Fremont, 2007)
4. T Wiegand, G Sullivan, G Bjntegaard, A Luthra, Overview of the H.264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol. **13**(7), 560–576 (2003)
5. J Iyengar, P Amer, R Stweart, Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths. IEEE/ACM Trans. Netw. **14**(5), 951–964 (2006)
6. Information Sciences Institute, The network simulator ns-2. (University of Southern California). http://www.isi.edu/nsnam/ns/. Accessed 29 March 2013
7. Protocol Engineering Laboratory, ns-2 SCTP module. (University of Delaware). http://pel.cis.udel.edu. Accessed 29 March 2013
8. H Schwars, D Marpe, T Wiegand, Overview of the scalable video coding extension of the H.264/AVC standard. IEEE Trans. Circuits Syst. Video Technol. **17**(9), 1103–1120 (2007)
9. IEG Richardson, *Video Codec Design*. (Wiley, Hoboken, 2002)
10. R Stewart, Q Xie, *Stream control Transmission Protocol. A Reference Guide*. (Addison-Wesley, Boston, 2002)
11. L Budzisz, J Garcia, A Brunstrom, R Ferrús, A taxonomy and survey of SCTP research. ACM Comput. Surv. **44**(4) (2012)
12. J Nightingale, Q Wang, C Grecos, Empirical evaluation of H.264/SVC streaming in resource-constrained multihomed mobile networks. Multimedia Tools Appl (2012). doi:10.1007/s11042-012-1219-5
13. D Jurca, P Frossard, Video packet selection and scheduling for multipath streaming. IEEE Trans. Multimedia. **9**(3), 629–641 (2007)
14. T Ojanperä, M Luoto, J Ortiz, M Myllyniemi, Integrating adaptive video streaming service with multi-access network management. Mobile Netw. Appl. **17**(4), 492–505 (2012)
15. C Lal, V Laxmi, MS Gaur, in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics. ICACCI '12*. A rate adaptive multipath routing protocol to support video streaming in MANETs (ACM New York, 2012)
16. NM Freris, C hsin Hsu, X Zhu, JP Singh, in *Proceedings of IEEE International Symposium on Multimedia (IMS'10)*. Resource allocation for multihomed scalable video streaming to multiple clients (IEEE Taichung, December 2010)
17. E Kohler, M Handley, S Floyd, *RFC 4340, Datagram Congestion Control Protocol*. (Internet Engineering Task Force, Fremont, 2006)
18. M Ransburg, E Martínez Graciá, T Sutinen, J Ortiz, M Sablatschan, H Hellwagner, in *Mobile Multimedia Communications*. Scalable video coding impact on networks (Springer New York, 2010), pp. 571–581
19. P Amon, T Rathgen, D Signer, File format for scalable video coding. IEEE Trans. Circuits Syst. Video Technol. **17**(9), 1174–1185 (2007)
20. P Amer, A Caro Jr, J Iyengar, P Natarajan, N Ekiz, ns-2 SCTP Readme file (2009). http://nsnam.cvs.sourceforge.net/viewvc/nsnam/ns-2/sctp/sctp.README. Accessed 29 March 2013
21. R Stewart, M Ramalho, Q Xie, M Tuexen, P Conrad, *RFC 3758: Stream Control Transmission Protocol Partial Reliability Extension*. (Internet Engineering Task Force, Fremont, 2004)

doi:10.1186/1687-6180-2013-115
Cite this article as: Ortiz *et al.*: SCTP as scalable video coding transport. *EURASIP Journal on Advances in Signal Processing* 2013 **2013**:115.