An Adaptive Multi-Purpose Transmission Scheme for H.264 Encoded Video in Wireless Networks

Markus Quaritsch*, Mario Wiesinger†, Bernhard Strobl‡, Bernhard Rinner*
*Klagenfurt University, Institute of Networked and Embedded Systems, 9020 Klagenfurt, AUSTRIA
†Graz University of Technology, Institute for Technical Informatics 8010 Graz, AUSTRIA
‡Austrian Research Centers GmbH, Smart Systems Division, 1020 Wien, AUSTRIA
Markus.Quaritsch@uni-klu.ac.at, wiesinger@iti.tugraz.at, Bernhard.Strobl@arcs.ac.at, Bernhard.Rinner@uni-klu.ac.at

Abstract— **Transmitting live video data is crucial in almost any surveillance application. Although the emerging research field of smart cameras promises to automate video analysis, human operators are still required to initiate the according actions in the case of an alarm. Therefore, it is important to have high-quality real-time video of the corresponding scenes. Usually, video data also has to be archived for later analysis.**

This paper presents a novel transmission scheme for encoded video data which allows for low-latency transmissions as it is required for live watching and reliable transmissions for archiving simultaneously in a single connection. The proposed transmission scheme uses SCTP as transport layer protocol and is designed for H.264 encoded video, but can be adapted to other video encoding standards as well.

The real-world evaluation shows that this novel transmission scheme is competitive to the widely used RTP/UDP method. When incorporating limited bandwidth and packet-loss our approach outperforms RTP/UDP regarding the number of received frames as well as quality of the perceived video.

I. Introduction

In recent years there is an increasing demand on visual surveillance systems. The applications cover a wide area, ranging from monitoring public places such as train stations or airports, over traffic monitoring up to assisted living.

There is also a lot of research effort in the field of smart cameras [1]. These cameras are equipped with high-performance on-board computing and communication. Smart cameras are thus capable to analyze the acquired video data and provide abstract information on a scene or generate alarms when detecting suspicious actions. The ultimate goal is to unburden human operators from observing dozens of cameras. However, in the case of an alarm human interaction is still inevitable and this requires visual information on the scene. Additionally, video footage has to be archived for later investigation.

Smart surveillance cameras thus have to provide a high quality video stream of a scene to support the human operator. Currently employed solutions are based on RTP/UDP for live watching and TCP based streams for archiving. This paper reports on a novel transmission scheme which addresses two conflicting issues, namely low-latency and reliability. The described approach is based on the stream control transmission protocol (SCTP) and can be used for live video streaming and archiving simultaneously. This allows to have only a single video stream instead of a RTP/UDP live video stream and a TCP stream to the video archive. The transmission method is designed for H.264 encoded video because this is an emerging encoding standard with increased performance. But the proposed method can be easily adapted to other video formats as well. The implementation of the proposed method has been stressed in streaming H.264 encoded video data from a smart camera [2]. Evaluation results show that this approach is competitive to RTP/UDP under optimal network conditions and outperforms RTP/UDP when incorporating limited bandwidth as well as packet-loss. Hence, the proposed transmission scheme is perfectly suited for wireless environments where the available bandwidth is limited and the packet loss is considerable and also a good choice for wired environments.

The reminder of this paper is organized as follows. Section II gives a short overview of the used technologies and surveys prior art in this area. Section III then describes the proposed transmission scheme and Section V summarizes the obtained results. Finally, Section VI concludes this paper with a brief discussion.

II. Background and Related Work

A. Video Compression

H.264/AVC [3], also known as MPEG-4 part 10, is a new standard for encoding video data. Compared to previous video encoding standards, H.264 provides enhanced video compression performance for interactive applications like video conferences as well as non-interactive applications like television broadcasting and storage. Compared to previous standards (e.g. MPEG-2), H.264 shows an increased encoding efficiency of up to 50 %, while the complexity of the decoder quadrupled [4].

H.264/AVC consists of two conceptual layers. The video coding layer (VCL) is responsible for efficient video encoding/decoding while the network adaption layer (NAL) handles the transmission of encoded data over a variety of communication channels (either circuit-switched or packet-switched) or stores the video footage. Details of the VCL are not relevant for further considerations as its main objective is efficient encoding. However, one feature is interesting for low-latency video transmission. With H.264 it is possible to split the image-region into multiple slices whereas the individual slices can be decoded independently. This allows to transmit parts of an image (i.e. a slice) while the VCL is still processing other slices of the same image. Thus, the delay from acquiring an image to displaying it at a remote site can be significantly reduced.

The VCL generates NAL units for packetizing and transmission by the NAL. Each NAL unit can be decoded independently. NAL units are classified into VCL and non-VCL NAL units. The VCL NAL units consist of encoded video data while the non-VCL NAL units comprise associated additional information such as parameter sets. As non-VCL NAL units do not contain video data, these units have no tight timing constraints but they are required for decoding.

The transmission of NAL units is not part of the H.264 standard. Thus, the NAL may use several different transport mechanisms. In IP-based networks, RTP/UDP is typically used to stream H.264 encoded video while MPEG-2 transport streams are common for broadcasting.

B. Stream Control Transmission Protocol

The stream control transmission protocol (SCTP) [5], [6] is among the most recent transport layer protocols (OSI layer 4) of the Internet protocol (IP) suite. SCTP offers a connection-oriented and basically reliable communication channel on top of a connection-less packet-based network such as IP. It combines the benefits of TCP and UDP while cutting their drawbacks and further introduces a set of new features. In the following, a selection of features interesting for our proposed approach are presented.

Message oriented delivery: While TCP is a byte-stream oriented transport protocol with strict in-order delivery, SCTP is message oriented and preserves boundaries of application-layer messages, similar to UDP. Messages are encapsulated in chunks whereas a SCTP packet may be comprised of multiple chunks. This allows to decouple reliable delivery from message ordering.

Multiple logical streams: One of the most valuable new feature of SCTP is its support for multiple logical streams within a single association (an *association* is what is called *connection* in TCP). A stream is a unidirectional logical data flow and an arbitrary number of streams can be used in an association in both directions. For each stream, the message order is preserved while the message order between streams is not preserved. Using multiple streams avoids the head-of-line (HOL) blocking known from TCP. HOL blocking occurs when a TCP receiver is forced to re-sequence packets that arrive out of order because of network reordering or packet loss. In SCTP, if a packet is lost, only the according stream is blocked waiting for re-transmission while all other streams are not affected.

Congestion Control: A major benefit of SCTP is its TCP-friendly congestion control. As SCTP and TCP are used together in the same network, it is important to evenly share the available bandwidth between TCP and SCTP. This is achieved by a congestion control mechanism similar to TCP.

Multiple delivery modes: In addition to ordered delivery of messages within a stream, SCTP also supports unordered messages which are passed to the application immediately. These unordered messages are comparable to UDP datagrams. The partial reliability extension [7] further allows to transmit partially reliable or even unreliable messages. With *timed reliability*, the sender can define a lifetime for each message. Lost messages are retransmitted during their lifetime and discarded afterwards.

C. Related Work

SCTP is well suited for transmitting different kinds of data including encoded video data. However, at the time of writing there exists no standard which exploits the advanced features of SCTP for streaming H.264 or otherwise encoded video data (c.f. RFC 3984 [8] for streaming H.264 over RTP/UDP).

Despite this, several different approaches for transporting encoded video data via SCTP have been evaluated. Lifen et al. [9] investigated in transmitting MPEG-4 video using SCTP. In their work, I-frames are transmitted with higher reliability (these are retransmitted at most once) than P- and B-frames which are not retransmitted. Additionally, unordered delivery is used so that the receiver passes data to the application as it arrives. Simulations with the ns-2 network investigate on the size of the playback buffer under different conditions of the network; the latency of the video transmission is not considered.

Wang et al. [10] use the partial reliability extension of SCTP (PRSCTP) and compare the transmission of MPEG-4 video data with TCP and UDP. The clients are connected to the network via mobile-IP. Similar to the previous work, I-frames are assigned a longer lifetime than P- and B-frames. The simulation focuses on the delay during handoff in the mobile-IP network. Results show that the performance of PRSCTP is between that of UDP and TCP. The advantages of PRSCTP are its congestion control and the ability to drop unimportant data while retransmitting important data.

Argyriou describes in [11] an architecture for streaming H.264 encoded video. The transport protocol used is Media-SCTP, a modified (not standardized) version of SCTP with additional support for prioritization of streams. Separate streams are used for different kinds of NAL units. Each NAL unit is classified according to its importance and transmitted via the corresponding stream. I-frames and P-frames are considered more important than B-frames. An additional stream is used for receiver feedback. This allows for dropping frames at the sender and thus manage the bandwidth. The proposed architecture has been stressed with the ns-2 network simulator.

The approach presented in this paper significantly differs from the work described above as it address two conflicting issues of video streaming, namely low-latency streaming as required for live watching and reliability in order to archive the video footage. Furthermore, this approach is solely based on standardized mechanisms and protocols. While the results in the papers cited above are all founded on simulations, the evaluations of the presented transmission scheme have been conducted in a real network environment by streaming encoded live video data.

III. Multi-Purpose Transmission scheme

The transmission scheme presented in this paper addresses two conflicting issues in transmitting encoded video data, namely low-latency transmission for live viewing and simultaneously reliable message transport for archiving in a single video stream.

The transmission scheme is based on the assumption of H.264 encoded video data where encoded slices of each frame are available as individual NAL units. Though, this method can be adapted to other kinds of encoded video data es well. Furthermore, the SCTP protocol [5] with the partial reliability extension [7] is used as transport layer protocol. This allows to define a lifetime for each individual NAL unit.

The basic idea of this novel approach is to use multiple SCTP streams and multiplex the individual NAL units on the streams in a round-robin manner. Using multiple streams avoids blocking of subsequent slices (packets) which clearly improves latency as well as jitter. Moreover, each SCTP packet has assigned a lifetime, depending on transmission parameters. This prevents head-of-line blocking within a single stream. Either the packet can be transmitted within its lifetime or the packet is dropped if its lifetime exceeds. In both cases the transmission queue of a stream is empty when the next slice for this stream is being sent.

The number of required streams (n_{streams}) depends on parameters of the encoded video as well as the anticipated reliability and can be calculated according to (1).

$$n_{\text{streams}} = t_l \cdot \text{fps} \cdot n_{\text{slices}} + 1 \qquad (1)$$

The higher the lifetime t_l of the individual NAL units is chosen, the more reliable is the transmission. The video frame-rate (fps) and the number of slices (n_{slices}) depend on the configuration of the encoder. An additional stream is used for non-VCL NAL units.

The first $n_{\text{streams}} - 1$ streams are used in a round-robin manner to send VCL NAL units while the additional stream is dedicated to non-VCL NAL units. Fig. 1 illustrates the mapping of VCL NAL units to streams. The frame-rate is 25 fps using a single slice per frame for easier illustration and a lifetime of 2 s. A stream is not used until the lifetime of the last packet exceeded. For instance, stream 0 is used to send frame nr. 1 and then idle for 2 s until frame nr. 50 is sent.

The reliability of the transmission is configured at the sender side, independent of subsequent use at the receiver side. Depending on the application, the receiver can use different strategies to handle the incoming slices.

A. Most recently delivery

Applications for displaying the live video are typically interested in the most recent data. In this case the application opts for *most recently delivery* to get the slices

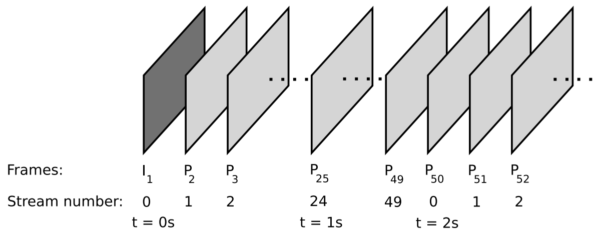

Fig. 1. Mapping of frames to streams.

immediately when they arrive. This allows to decode the video with low latency as well as low jitter.

In case of a lost packet the retransmitted packet either arrives in time and thus can be used by the decoder. Or, the retransmitted packet arrives too late and stresses the decoder's error-correction mechanisms which typically cause a degraded quality as the packet is not available for decoding. However, slices arriving too late can be stored in a buffer and when watching a scene again (e.g. for further investigation), the full image quality can be perceived.

B. Ordered delivery

The *ordered delivery* mode is intended for applications which have to receive the video data in the correct order, typically for archiving. Thus, the receiver has to reorder the incoming packets before passing on to the application. The increased latency and jitter in the case of packet loss is tolerable. As the transmission is only partially reliable, some frames may get lost. However, reliability of transmissions can be tuned via the lifetime of each slice in order to alleviate the probability of lost frames.

IV. System Architecture

The architecture of a streaming solution based on the proposed transmission scheme is sketched in Fig. 2. On the sender-side the source video is encoded and then passed on to the streaming-server. In the case of H.264 the streaming-server can be seen as a network abstraction layer for SCTP. The encoded slices are streamed to the client using the transmission scheme described in Section III. The video source must not necessarily be an encoder providing live video data but can also be pre-encoded footage for video-on-demand applications.

On the receiving side the streaming-client picks up and demultiplexes the slices from the server. Applications on the client may use different strategies to handle the incoming data. For decoding a live video stream the *most recently delivery* mode may be preferred to reduce latency and jitter. When the objective is to archive the video the *ordered delivery* mode should be chosen to get all frames into the archive. An application is not forced to choose between one of these two delivery modes but may use both simultaneously.

The streaming-server and streaming-client are depicted as individual modules in Fig. 2. Concerning implementation details, both may be implemented either as standalone application or as library to be integrated in applications.

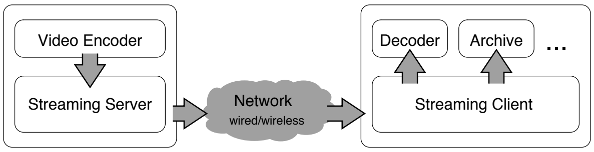

Fig. 2. Streaming architecture.

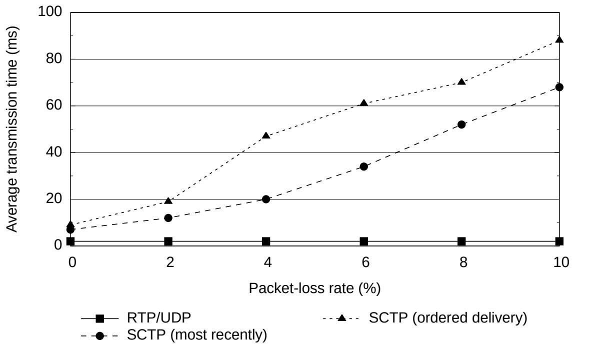

Fig. 3. Comparison of the average transmission time of RTP/UDP and SCTP (*most recently delivery*).

V. Experimental Results

The presented approach for transmitting H.264 over SCTP has been stressed in a real-world application. The evaluation investigates the use of the proposed transmission scheme in wired and also wireless Ethernet networks. Encoded video data is streamed from a smart camera [2] over a 100 MBit wired Ethernet to a standard PC. The wireless network is emulated by limiting the bandwidth and increasing the packet loss in the network. Encoding is done on the digital signal processor of the smart camera using an encoder from Ateme[1] while the SCTP streaming server runs on the ARM-based host processor. Video data is encoded at 20 fps with a GOP parameter of 20 and a single slice per frame[2]. According to practical evaluations a lifetime of 2 s for SCTP packets has shown to yield good results.

The proposed transmission scheme is compared to the widely-used RTP/UDP method for streaming video. TCP-based streaming solutions are hardly used and thus not of great interest.

First, the average transmission times of SCTP and RTP/UDP are analyzed under different packet-loss conditions. Fig. 3 illustrates the results. As expected, UDP offers fast transmission independent of the packet-loss rate while the transmission times of SCTP increase with the packet-loss rate. In case of no or only little packet-loss, SCTP is competitive regarding the transmission times. However, this experiment does not consider the number of lost packets and thus the perceived quality at the receiver.

The second evaluation focuses on the number of lost respectively late frames. Fig. 4 shows the obtained results for different packet-loss rates. When using RTP/UDP the number of lost frames increases significantly with the packet-loss rate up to 30 % and more. In contrast to this, the SCTP based transmission does not lose any frames and even under high packet-loss conditions only about 10 to 15 % of the frames are received too late.

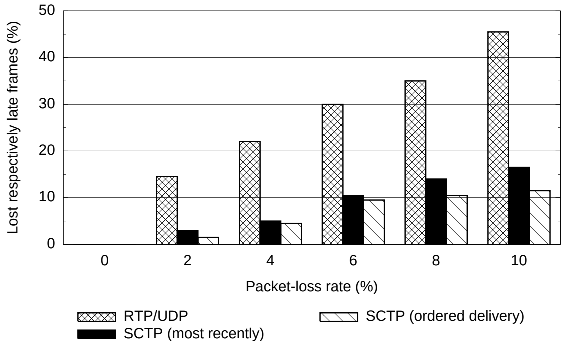

Fig. 4. Comparison of lost respectively late frames under different packet-loss conditions.

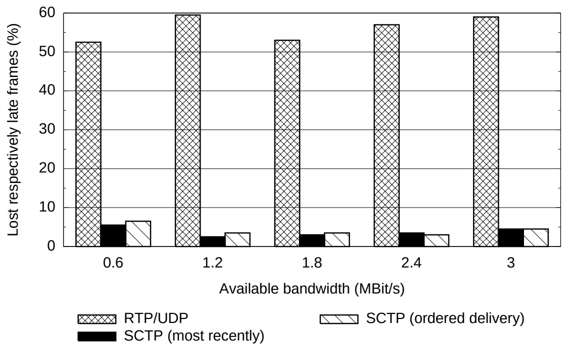

Fig. 5. Comparison of lost respectively late frames under different bandwidth constraints.

Fig. 5 illustrates the number of lost respectively late frames under a constant packet-loss rate of 4 % varying the available bandwidth from 0.6 MBit/s up to 3 MBit/s. Under this conditions more than 50 % of the frames are lost in all cases when using RTP/UDP. As expected, the SCTP based approach does not loose any frames and the number of late frames (less than 7 %) is tolerable because H.264 includes several error-resilience features to cope with lost frames. Of course, the average transmission time for SCTP is higher, but less than 100 ms with an available bandwidth of 1.8 MBit/s is also acceptable.

The intention of the last evaluation is to rate the quality of the received video stream independent of the decoder's performance. Therefore, the number of frames at the receiver are counted and each frame is weighted according to its type and whether it has been received on time, the frame was late or even lost (c.f. Table I). Significantly more weight is put on the I-frames as the following P-frames rely on them. For the evaluation 200 consecutive frames of a video in CIF resolution encoded with a GOP of 20 have been examined. Hence, the sum of weighted frames is 580 when all frames are received in time (10 I-frames and 190 P-frames). Table II lists the number of received frames and rates the quality of the received video when streaming the video over a link with 6 % packet-loss

[1] http://www.ateme.com/

[2] The parameters have been chosen due to limitations of the available encoder.

TABLE I

EVALUATION SCHEME FOR THE QUALITY OF THE RECEIVED VIDEO.

Frame type	*Reception status*	*Weight*
I-frame	in-time	20
	late	10
	lost	0
P-frame	in-time	2
	late	1
	lost	0

TABLE II

COMPARING THE QUALITY OF THE RECEIVED STREAM.

Frame type	*Frame count*	*SCTP*	*RTP/UDP*
I-frames	in-time	10	0
	lost frames	0	10
	late frames	0	0
P-frames	in-time	179	84
	late frames	0	106
	late frames	11	0
Rating		569	168
Relative rating		98 %	29 %

and a limited bandwidth of 1 MBit/s.

The results show, that the proposed transmission scheme is significantly better than the RTP/UDP method. When using SCTP, only a negligible number of P-frames is received too late while the majority of frames is received on time. By contrast, RTP/UDP shows a very poor performance. All I-frames and more than 50 % of the P-frames are lost. Decoding this stream would yield, if at all, very unsightly results.

VI. CONCLUSION AND FUTURE WORK

In this paper a novel approach for transmitting H.264 encoded video data is presented that allows for life watching as well as archiving video footage in a single stream. The proposed method is based on SCTP, a transport layer protocol. Multiple logical streams within an SCTP association are used together with the partial reliability extension of SCTP. Hence, this approach is more efficient regarding bandwidth requirements because instead of two individual streams (i.e. one stream for live video data and one stream for archiving) this can be handled with a single stream fulfilling the different requirements of both applications.

The evaluation in a real-world environment shows that this approach outperforms the widely used RTP/UDP streaming method. The average transmission times of the proposed approach are somewhat higher than with RTP/UDP. But when taking into account the number of lost frames under different packet-loss conditions, this method performs significantly better. Also under network conditions with limited bandwidth and considerable packet-loss rates which are typical for WLAN networks, this approach performs considerably better than RTP/UDP.

Future work includes to adapt this novel transmission scheme to other video standards. Further, it is desirable to adapt the lifetime of SCTP packets to the current network conditions autonomously.

REFERENCES

[1] H. Aghajan and R. Kleihorst, Eds., *Proceedings of the First ACM/IEEE International Conference on Distributed Smart Cameras*, Vienna, 2007. IEEE.

[2] M. Bramberger, A. Doblander, A. Maier, B. Rinner, and H. Schwabach, "Distributed Embedded Smart Cameras for Surveillance Applications," *Computer*, vol. 39, no. 2, pp. 68–75, 2006.

[3] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 560–576, 2003.

[4] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," *Circuits and Systems Magazine, IEEE*, vol. 4, no. 1, pp. 7–28, 2004.

[5] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, and M. Kalla, "Stream Control Transmission Protocol," RFC 2960, Internet Engineering Task Force, Oct. 2000.

[6] R. Stewart and C. Metz, "SCTP: new transport protocol for TCP/IP," *IEEE Internet Computing*, vol. 5, no. 6, pp. 64–69, 2001.

[7] R Stewart, M. Ramalho, Q. Xie, M. Tuexen, and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension," RFC 3758, Internet Engineering Task Force, May 2004.

[8] S. Wenger, M. Hannuksela, Thomas Stockhammer, M. Westerlund, and D. Singer, "RTP Payload Format for H.264 Video," RFC 3984, Internet Engineering Task Force, Feb. 2005.

[9] Z. Lifen, S. Yanlei, and L. Ju, "The performance study of transmitting MPEG4 over SCTP," in *Neural Networks and Signal Processing. Proceedings of the International Conference on*, 2003, vol. 2, pp. 1639–1642 Vol.2.

[10] H. Wang, Y. Jin, We. Wang, J. Ma, and D. Zhang, "The performance comparison of PRSCTP, TCP and UDP for MPEG-4 multimedia traffic in mobile network," in *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on*, 2003, vol. 1, pp. 403–406 vol.1.

[11] A. Argyriou, "A Novel End-to-End Architecture for H.264 Video Streaming over the Internet," *Telecommunication Systems*, vol. 28, no. 2, pp. 133–150, February 2005.