Xstream-x264: Real-time H.264 Streaming with Cross-layer Integration

Golam Sarwar and Roksana Boreli
National ICT Australia (NICTA); University of New South Wales (UNSW)
{golam.sarwar,roksana.boreli}@nicta.com.au

Emmanuel Lochin
Université de Toulouse; UPS, INSA, INP, ISAE, LAAS; F-31077 Toulouse, France
emmanuel.lochin@isae.fr

Abstract

We present Xstream-x264: a real-time cross-layer video streaming technique implemented within a well known open-source H.264 video encoder tool x264. Xstream-x264 uses the transport protocol provided indication of the available data rate for corresponding adjustments in the video encoder. We discuss the design, implementation and the quality evaluation methodology utilised with our tool. We demonstrate via experimental results that the streaming video quality greatly improves with the presented cross-layer approach both in terms of lost frame count and the objective video quality metrics Peak Signal to Noise Ratio (PSNR).

Keywords video encoding, streaming, H.264, DCCP, MPEG4, cross layer

1. Introduction and Motivation

Online measurement company ComScore[1] estimates that on average, Internet users viewed 30 billion online videos per month in 2009, making video streaming one of the most pervasive applications on the Internet. Streaming video quality is highly dependent on the video codec data rate and impairments like frame losses [1], which may be due to congestion or change in the network used due to mobility.

The most commonly used protocol for multimedia communications is Real-time Transport Protocol (RTP) [2], used in conjunction with UDP to provide full transport protocol functionality, although it may also be used with other transport protocols like TCP. When used with Real-time Transport Control Protocol (RTCP) [2], RTP will also provide control and quality information. Additionally, RTP may have specific media related congestion control using UDP. Two other not widely used multimedia transport protocols include the Stream Control Transmission Protocol (SCTP) [3] and Datagram Congestion Control Protocol (DCCP) [4]. All of these transport protocols, with the exception of UDP, have their respective methods for reacting to losses and congestion on the link. Various link quality metrics may also be available in the lower layers. However, multimedia applications do not have any functional means to take advantage of this information and react by changing the encoding parameters. As a consequence, the application requiring a larger capacity than what is provided by the link and/or transport layers will experience packet losses. For the user, the visible or audible consequence will be a decrease of the perceived quality.

There has been a body of research in cross layer approaches which have demonstrated the potential of video streaming based on cross-layer information to improve the quality of end user's viewing experience [7–10]. Most of these proposals contain simulated results of potential improvements, due to the lack of freely available tools which can accept cross layer information, encode video according to that input and stream the encoded video to a receiver. EvalVid [11] and H.264/AVC JM Reference Software [12] are the most commonly used freely available video encoding evaluation tools today. MPEG4IP [13] also provides similar functionality. JM software supports only H.264 reference encoding, while EvalVid and MPEG4IP support off-line encoding and streaming, requiring the user to first encode the video and then, in the second step, stream it. However, none of these tools provides the crucial capability of encoding and streaming at the same time and more importantly, they do not allow the encoder to adjust the target encoding rate and other encoding quality control parameters based on external inputs. In this paper, we present a technique to utilize external information and a tool for a cross-layer adaptable H.264 video coding and streaming and demonstrate it's benefits to video quality on congested links.

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

2. Design of the Cross Layer Video Tool

H.264/MPEG-4 is a widely used standard for video compression today [5]. H.264 video, when streamed over congested communication channels, will incur a loss of quality if video frames are lost during transmission [1]. For real-time multimedia streaming, this creates a bigger problem as in-time delivery is crucial and reliability achieved using retransmissions may not satisfy the in-time requirement. To minimize losses and maximize end users' viewing experience, we propose a simple and efficient cross layer technique which adjusts the parameters of the ongoing video encoding process based on external information (eg. throughput, loss information about the link found in transport protocol).

The most common H.264 parameters specified for the encoding process are either a user-specified constant quality or a constant bitrate (CBR) [6]. We note that CBR encoding results in all frames being encoded with the same value of the maximum target bit rate, while constant quality, which effectively is VBR encoding will result in a data rate which is proportional to the degree of complexity and motion in the video. In our proposed technique, the encoder effectively encodes with both a variable quality and a variable rate by following an externally provided rate based on cross layer information. The encoding process changes the rate when a change in the available bandwidth is detected, due to loss or congestion. This allows the application to conform to the available bandwidth on the link resulting in a lower number of frame losses and provides an improved viewing experience.

We have implemented the proposed technique in a tool, by modifying an open source H.264 encoder x264 [6]. The implementation also includes a real time encoding and streaming capability and additional features which aid the evaluation of video quality when the received video has frame losses.

The functional components of our Xstream-x264 tool are shown in Figure 1.

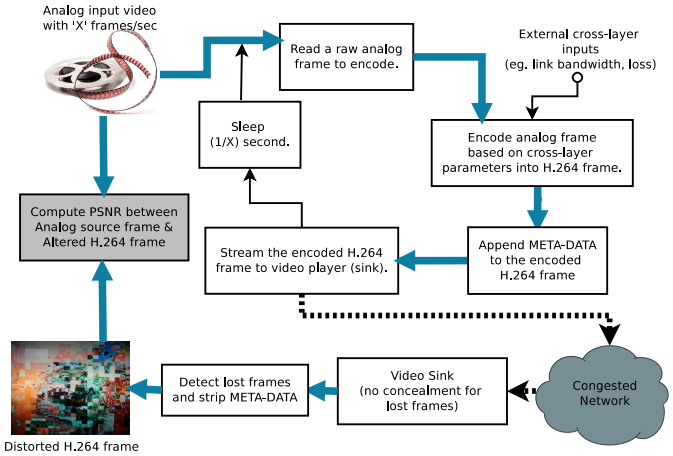

Figure 1: Design of Xstream-x264

The analog video source is encoded into H.264 frames based on throughput or link capacity information provided transport protocol (although other external parameters may also be used from other layers in OSI network stack). Meta-data, including frame sequence number and fragmentation sequence number is appended to the encoded frame (a total of 4 bytes per frame), which is then streamed with timing corresponding to the source video. The H.264 video is sent over the link of choice and captured by the video sink. This video is passed through the loss detection module we have also implemented, followed by the computation of video quality.

To evaluate the received video quality, we use Peak Signal to Noise Ratio (PSNR), the most widely used objective video quality metric standardized by ITU [14]. Although PSNR may not always accurately estimate the subjective video quality due to non-linear nature of the human visual assessment, studies have shown that it does provide a reasonable estimate of the subjective assessment if it is applied over the same video content under the same environment [15]. We use the *Compare* component of ImageMagick [16] tool-set for PSNR computation, in conjunction with our frame loss detection module. To enable PSNR computation using two streams which have an unequal number of video frames, for any detected lost frame, we insert a blank (all white) frame. The existing playback concealment techniques [17] can still be utilized in addition to our proposal and should further improve the results in terms of visual perception. We have however chosen not to use them in our experiments, in order to more clearly show the benefits of our proposal.

Our tool can currently be used with either UDP/RTP or DCCP with Congestion Control ID 3 (CCID3) [4]. DCCP was targeted as it is a good transport protocol candidate for future multimedia applications, with CCID3 specifically designed to work with data and video transmission [4].

3. Experiments and Analysis

To demonstrate the capability of Xstream-x264, in this section we present experimental results showing the effects of congestion. The simple network topology used in the experiments includes a source (video sender) and a sink (video receiver), connected by a router. Congestion is emulated in the router by using the Linux network emulator *NetEm* rather than additional traffic streams. The nominal link bandwidth is 1 Mbit/s and the Round-Trip-Time (RTT) is 10ms for all the experiments. All experiments are of 42 seconds duration and the artificial congestion is introduced by reducing the link bandwidth to 700 Kbit/s after 15 seconds; after the 30th second, the link bandwidth is further decreased to 400 Kbit/s. The Maximum Transmission Unit (MTU) of the network is set to 1400 bytes. If an encoded frame is larger than the MTU, the video frame is fragmented and transmitted in subsequent packets without any intermediate frame delay. Otherwise, the encoded frames are transmitted in real time, with respect to the intra-frame delay of the raw input video. The analog input video used in our experiment is a 41.66 seconds, 1000 frames long scene from a video clip of an action movie with a very large amount of motion. The input video plays at 23.976 frames per second (FPS).

We compare the calculated PSNR values for our proposed cross layer mechanism and a CBR encoded video stream, transmitted over the congested link. We use DCCP transport protocol to provide the cross-layer information about the link to the video application, we also use DCCP for the CBR experiments, with a constant bit rate (CBR) of 1 Mbit/s. To present a fair evaluation of how our proposed method utilizes the rate information provided by DCCP, we also perform experiments for an ideal scenario where the application knows the available rate on the link and changes the encoding parameters according to this presumed knowledge. The resulting PSNR values are shown in Figure 2, together with the congestion pattern used in the experiments. As can be expected, CBR (1 Mbit/s), which is not well suited to the available capacity of the link, experiences significant losses in the congested period and as a consequence, DCCP-CCID3 (which has a loss and RTT based congestion control mechanism [4]) severely restricts the rate available to the application. On the other hand, our proposed cross-layer VBR technique attempts to follow the available rate as advised by the transport protocol, thereby the result is much lower frame loss. The only loss is due to the frequency of DCCP rate change and the limitation of the application to only change the encoding rate on a per-frame basis. The ideal case where the application knows the exact rate rather than the DCCP information about the rate shows the highest PSNR.

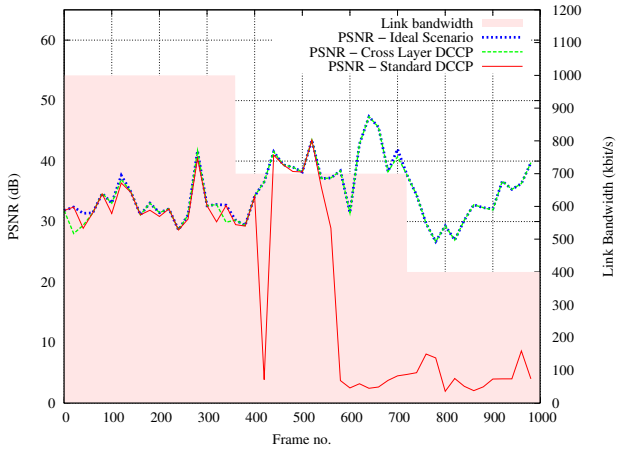

Figure 2: Comparison of PSNR for an ideal scenario, proposed cross-layer DCCP and standard DCCP streaming.

The analysis of the PSNR values for the for ideal scenario, proposed cross layer DCCP (averaged over all possible initial rates from 100 Kbit/s to 1 Mbit/s) and standard DCCP is shown in Table 1. We present the mean and coefficient of variation (CV, ratio of standard deviation to mean) for the full experiment and, separately, for the three congestion regions.

To further demonstrate the advantages of the adaptive mechanism, we also show the number of lost frames from the same experiments in Table 2. Please note that the ideal scenario has no losses, therefore it is omitted. Frame losses are separated into losses for the index (I), predictive (P) and bi-directional predictive (B) frames. In H.264, the I frames provide reference and necessary decoding information for the P and B frames and therefore lost I frames have a greater impact on quality. As can be seen in the tables, by adapting to our cross layer technique, not only we have reduced the overall number of lost frames, we have also cut down the number of lost I frames which guarantees improved video playback for the viewer.

		1 Mbit/s		700 Kbit/s		400 Kbit/s	
	Overall Mean (dB)	Mean (dB)	CV (dB)	Mean (dB)	CV (dB)	Mean (dB)	CV (dB)
IDEAL	35.20	33.60	0.08	38.68	0.11	32.77	0.11
Cross-layer DCCP	33.62	30.96	0.27	37.30	0.20	32.28	0.16
Standard DCCP	19.83	28.95	0.36	22.54	0.72	4.63	0.55

Table 1: Detailed analysis of PSNR from Figure 2.

Encoding Bit rate (Kbit/s)	% Total loss	% frame loss (I)	% frame loss (P)	% frame loss (B)
Cross-layer DCCP	6.21	0.20	2.30	3.70
Standard DCCP	47.75	0.60	25.13	21.82

Table 2: % Frame Loss for Cross Layer DCCP and Standard DCCP (1 Mbit/s CBR).

We further extend our work by emulating the standard YUV video sequences provided by Xiph Foundation [18] and Arizona State University [19] for multimedia research. Videos of different resolution and motion complexity are chosen for cross layer based DCCP and standard DCCP streaming. All videos are encoded at 23.976 frames per sec (FPS). Standard DCCP streaming is performed at 1 Mbit/sec. Emulations are performed under the same congestion environment shown in Figure 2. Performance of cross layer DCCP using Xstream-x264 and standard DCCP streaming in terms of number of lost frames are presented in Table 3. As can be seen in Table 3, cross-layer based DCCP streaming with Xstream-x264 significantly reduces number of lost frames and shows notable improvement in all cases.

In Table 4, we present detailed analysis of lost I, P and B frames for the experiments with standard YUV video sequences presented in Table 3. Cross-layer based streaming significantly reduces the number dropped I frames which substantially helps improve streaming video quality experience in a congested network environment.

We illustrate the visual quality difference in Figure 3 by comparison of a selected video frame from two sources: as decoded from the stream transmitted and received by our proposed cross layer approach and decoded from the 1 Mbit/s CBR received stream. It can be observed that the image in Figure 3b appears significantly more distorted than the image from Figure 3a.

Video Name	Resolution of the YUV Video Sequences	Motion Complexity	% Total Lost Frames - Standard DCCP	% Total Lost Frames - Cross Layer DCCP
Foreman	352 x 288	High	34.2	9.8
Football	352 x 288	High	37.1	9.2
Stefan	352 x 264	High	37	9.4
Highway	352 x 288	High	33.2	8.6
Bus	352 x 288	High	34.3	8.8
Coastguard	352 x 288	High	48.4	9.7
Carphone	176 x 144	Medium	17.3	5.1
Mobile	352 x 288	Medium	31.6	7.4
Paris	352 x 288	Medium	39.6	7.6
Suzie	176 x 144	Medium	23.1	4.1
Akiyo	352 x 288	Low	30	6.4
Bridge	352 x 288	Low	30.6	1.1
Container	352 x 288	Low	54.9	14
News	352 x 288	Low	42.6	11.3
Hall	352 x 288	Low	29.1	6.9

Table 3: % Frame loss of Cross Layer DCCP and Standard DCCP (1 Mbit/s CBR) for standard YUV video sequences.

(a) DCCP Cross-layer (b) Standard DCCP

Figure 3: Comparison of visual frame quality.

4. Conclusions and Future Work

We have proposed and implemented a novel cross layer based video encoding and streaming technique and demonstrated it's feasibility by Xstream-x264 based implementation. Our experimental results show that cross-layer based streaming significantly outperforms the traditional streaming method using DCCP transport protocol. As future work, we plan to integrate RTP transport protocol with our tool and make it available to a broader range of audience experimenting with multimedia encoding and streaming.

Acknowledgments

This research work has been supported by funding from National ICT Australia (NICTA). NICTA is a research organization funded by Australian Government research initiatives through Australian Research Council (ARC).

References

[1] A. Huszak, S. Imre, Analysing GOP Structure and Packet Loss Effects on Error Propagation in MPEG–4 Video Streams, ISCCSP 2010.

Video Name	Encoding Method	% I Frame Loss	% P Frame Loss	% B Frame Loss	% Total Loss
Foreman	Standard DCCP	0.2	17	17	34.2
	CrossLayer DCCP	0.0	4.2	5.6	9.8
Football	Standard DCCP	0.1	20.4	16.6	37.1
	CrossLayer DCCP	0.0	5.2	4.0	9.2
Stefan	Standard DCCP	0.5	15	21.5	37.0
	CrossLayer DCCP	0.0	4.2	5.2	9.4
Highway	Standard DCCP	0.3	9.1	23.8	33.2
	CrossLayer DCCP	0.0	2.5	6.1	8.6
Bus	Standard DCCP	0.4	14	20.9	35.3
	CrossLayer DCCP	0.1	3.4	5.3	8.8
Coast-gurad	Standard DCCP	0.4	28.6	19.4	48.4
	CrossLayer DCCP	9.7	0.0	6.1	3.6
Carphone	Standard DCCP	2.4	6.0	8.9	17.3
	CrossLayer DCCP	0.0	1.9	3.2	5.1
Mobile	Standard DCCP	0.3	10.2	21.1	31.6
	CrossLayer DCCP	0.0	2.5	4.9	7.4
Paris	Standard DCCP	0.3	23	16.3	39.6
	CrossLayer DCCP	0.1	4.6	2.9	7.6
Suzie	Standard DCCP	0.0	12.8	10.3	23.1
	CrossLayer DCCP	0.0	2.3	1.8	4.1
Akiyo	Standard DCCP	0.2	13.6	16.2	30
	CrossLayer DCCP	0.0	3.3	3.1	6.4
Bridge	Standard DCCP	0.1	9.0	21.5	30.6
	CrossLayer DCCP	0.0	1.6	5.7	7.3
Container	Standard DCCP	0.5	28	26.4	54.9
	CrossLayer DCCP	0.2	10.5	3.3	14
News	Standard DCCP	0.1	19.1	23.4	42.6
	CrossLayer DCCP	0.1	5.0	6.2	11.3
Hall	Standard DCCP	0.2	5.9	23	29.1
	CrossLayer DCCP	0.0	1.7	5.2	6.9

Table 4: Detailed Analysis of % I, P and B Frame Loss for the Experiments Presented in Table 3.

[2] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, RTP: A Transport Protocol for Real-Time Applications, RFC3550, July 2003.

[3] R. Stewart, Ed., Stream Control Transmission Protocol (SCTP), RFC 4960, September 2007.

[4] S. Floyd, E. Kohler and J. Padhye, Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC), RFC 4342, March 2006.

[5] T. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A., Overview of the H.264/AVC Video Coding Standard,ITCSVT 2003.

[6] x264, `http://www.videolan.org/developers/x264.html`.

[7] S. Benayoune, N. Achir, K. Boussetta, K. Chen, A MAC Centric Cross Layer Approach for H.264 Video Streaming over HSDPA, Journal of Communications, Oct 2009.

[8] Khan, S. Peng, Y. Steinbach, E. Sgroi, M. Kellerer, W., Application-driven cross-layer optimization for video streaming over wireless networks, 10.1109/MCOM.2006.1580942, 2006.

[9] Soni, R., Chilamkurti, N., Giambene, G., Zeadally, S., A Cross-Layer Design for H.264 Video Stream Over Wireless Local Area Networks, CSA 2008.

[10] Zhengye Liu, Hang Liu, Yao Wang, Cross Layer Adaptation for H.264 Video Multicasting Over Wireless Lan, ICME 2006.

[11] EvalVid - A Video Quality Evaluation Tool-set, `http://www.tkn.tu-berlin.de/research/evalvid/`.

[12] H.264/AVC JM Reference Software, `http://iphome.hhi.de/suehring/tml/`

[13] MPEG4IP, `http://www.mpeg4ip.net/`.

[14] ITU-T Rec. J.247 (08/08) Objective perceptual multimedia video quality measurement in the presence of a full reference, `http://www.itu.int/rec/T-REC-J.247/en`

[15] Q. Huynh-Thu1 and M. Ghanbari, Scope of validity of PSNR in image/video quality assessment, IET Electronics Letters, 2008.

[16] ImageMagick, `http://www.imagemagick.org/`

[17] Zhenyu Wu, Boyce, J.M., An Error Concealment Scheme for Entire Frame Losses for H.264/AVC, ISCAS 2006.

[18] Xiph. Foundation, `http://media.xiph.org/`

[19] Arizona State University, "YUV Video Sequences", `http://trace.eas.asu.edu/yuv/`