Citation for the published Journal paper:
Title:
In press: Eliminating the Effects of Freezing Frames on User Perceptive by Using a Time Interleaving Technique

Author:
Hussein Muzahim Aziz, Markus Fiedler, Håkan Grahn, Lars Lundberg

Journal:
Multimedia Systems

Year:
2011
Vol.

Issue:

Pagination:

URL/DOI to the paper:
10.1007/s00530-011-0235-z.

Multimedia Systems
DOI 10.1007/s00530-011-0235-z

REGULAR PAPER

Eliminating the effect of freezing frames on user perceptive by using a time interleaving technique

Hussein Muzahim Aziz · Markus Fiedler · Håkan Grahn · Lars Lundberg

Received: 11 November 2010 / Accepted: 18 May 2011

Abstract Streaming video over a wireless network faces several challenges such as high packet error rates, bandwidth variations, and delays, which could have negative effects on the video streaming and the viewer will perceive a frozen picture for certain durations due to loss of frames. In this study, we propose a Time Interleaving Robust Streaming (TIRS) technique to significantly reduce the frozen video problem and provide a satisfactory quality for the mobile viewer. This is done by reordering the streaming video frames as groups of even and odd frames. The objective of streaming the video in this way is to avoid the losses of a sequence of neighbouring frames in case of a long sequence interruption. We evaluate our approach by using a user panel and mean opinion score (MOS) measurements; where the users observe three levels of frame losses. The results show that our technique significantly improves the smoothness of the video on the mobile device in the presence of frame losses, while the transmitted data are only increased by almost 9% (due to reduced time locality).

Communicated by Thomas Haenselmann.

H. M. Aziz (✉) · M. Fiedler · H. Grahn · L. Lundberg
School of Computing, Blekinge Institute of Technology, Karlskrona, Sweden
e-mail: hussein.aziz@bth.se

M. Fiedler
e-mail: markus.fiedler@bth.se

H. Grahn
e-mail: hakan.grahn@bth.se

L. Lundberg
e-mail: lars.lundberg@bth.se

Keywords Streaming video · Frozen pictures · Interleaving · Switching frames · MOS

1 Introduction

With the rapid development of video coding and wireless communication technologies, video streaming has become very popular to mobile users [11]. The H.264/AVC coding standard has been introduced to achieve high compression of the video stream. H.264/AVC is a lossy video compression system that removes subjective redundancy, i.e. elements of the video sequence that can be removed without significantly affecting the end viewer's perceived quality [22].

The User Datagram Protocol (UDP) is a transport protocol often used for streaming video. UDP does not guarantee packet delivery; the receiver needs to rely on another protocol like the Real-Time Protocol (RTP) to detect packet losses [9]. RTP is a standard protocol used over UDP for streaming videos and it is designed to provide end-to-end transport functions for supporting real-time applications. When the network is congested, the UDP sender continues to send packets at a constant rate resulting in a number of unavoidable packet losses. As a result, these packet losses between the video sender and the receiver significantly degrade the quality of the received video [2].

Real-time video streaming is particularly sensitive to frame losses and delays since the frames must arrive at the mobile device before their playout time with enough time to decode and display the frames [8]. The available bandwidth of a wireless channel varies with time, and that can potentially result in loss of frames, called outage [1, 23]. If the receiver buffer runs out of packets, the playout of the video is interrupted until the buffer has received as many frames that are required to decode and continue to display the video sequence on the mobile device [13]. Too long outages (e.g. several seconds) will result in user dissatisfaction. Therefore, outages should be maintained at an acceptable level to satisfy the quality of service (QoS) [13, 16]. Even a low amount of frame losses can result in a severe degradation of quality as perceived by the users [12, 24]. Streaming video over wireless networks with high error rates will affect the video quality and the viewer perceives frozen pictures for a certain duration followed by a more or less abrupt change in the picture content due to frame losses [19, 21].

To handle the frozen picture problem, the Time Interleaving Robust Streaming (TIRS) approach is proposed in this paper. The purpose of TIRS is to eliminate frozen pictures by avoiding a sequence of neighbouring frames to be lost and allowing at least every second frames to be present in the mobile device. TIRS is applied before the video stream is compressed by the H.264 encoder. H.264 will compress the video by removing (subjectively) redundant data. The benefit of compressing the video stream by H.264 is the time locality, i.e. the difference between two adjacent frames is often relatively small.

In the case of interleaving, the relative difference between two adjacent frames in the video stream will increase, thus reducing time locality and increasing the amount of the data needed to encode the video. Therefore, the size of the streaming video may increase due to interleaving. Consequently, interleaving will enhance the user perceived quality by reducing the problem with frozen pictures, but on the other hand it will increase the amount of transmitted data since the time locality is reduced.

Due to interleaving, adjacent frames in the original video will not arrive at the receiver one after the other, i.e. there is a time delay between the arrivals of adjacent frames. Therefore, the frames will be stored in a buffer at the receiver side before they can be played on to the display screen in the original order. This means that we need to introduce extra delay time due to interleaving.

In this paper, we quantify the gain in user-perceived quality caused by frame interleaving for video transmissions over wireless channels with potential frame losses. We also quantify the increased sized of the data encoding the video due to reduced time locality. Finally, we calculate the relation between the maximum outage that can be handled and the size of the receiver buffer. It turns out that TIRS on average can handle network outages that are longer than the (jitter) buffer on the receiver side.

The remainder of this paper is organized as follows. Section 2 provides background and related work to video streaming. Section 3 discusses the Interleaving Distance Algorithm (IDA) proposed by Claypool and Zhu [7]. In Sect. 4, a detailed explanation of the proposed TIRS technique for streaming the video frames in different positions is presented. Section 5 discusses the effect of frame losses on the interleaved video stream. Section 6 studies the effect of the interleaving on the video file size for the IDA and the proposed TIRS techniques. Section 7 compares our proposed TIRS technique to the IDA technique and Reed–Solomon coding. The analyses of the subjective data from the subjective evaluation are reported in Sect. 8. Section 9 provides the experimental results and discussions. Finally, we conclude this study with Sect. 10.

2 Background and related work

Frame interleaving is used in video streaming to reduce the effects of a sequence of frame losses. The sender reorders the frames before transmitting them to the receiver, so that originally adjacent frames are separated by a distance that may vary over time. Interleaving will reduce the effect of frame losses by dispersing the occurrence of errors in the original stream [6, 7, 18].

Cai and Chen [5] proposed an interleaving approach to improve the performance of video streaming over unreliable networks. The interleaving is applied to the compressed bitstream before the channel coding. The channel coding used was Reed–Solomon to generate channel-coded block symbols and transmit these blocks over the network with the consideration of packet losses, which will cause interruption at the receiving side. The packet loss may cause damages on a channel-coding block so that the error recovery capability of the channel coding may be exceeded. The authors proposed the forward error correction (FEC)-based pre-interleaving error control scheme for video streaming over unreliable networks. They declare that the proposed pre-interleaving greatly improved the performance of video streaming with packet losses on ATM networks.

Schierl et al. [23] presented a streaming system that utilizes interleaved transmission for real-time H.264/AVC video and audio in wireless environments. In their approach, they consider the audio as the highest priority. The interleaved transmission is carried out by using priority-based scheduling (PBS) with client feedback about the current fill level of each priority class in the client buffer and for retransmission on the link layer for different error rates.

An error control protocol for robust MPEG-4 video multicast over wireless channels, called unequal interleaved FEC, is proposed by Nafaa et al. [17]. The protocol combines features of MPEG-4 data partitioning, FEC, and interleaving techniques. Adaptive FEC transmission based on efficient feedback for 1-to-n multicast communication scenario is also considered in their study.

Tsai et al. [25] proposed a technique to disperse burst losses to different FEC blocks. When sending the data packets of FEC blocks over multiple paths, the proposed technique changes the transmission order of FEC blocks and sends them using path interleaving. The receiver has a packet buffer to handle the impact of packet disordering. Path interleaving aims at striping two or more FEC blocks to multiple paths to spread the burst of packet losses to different blocks.

The above authors used feedback techniques and retransmission mechanisms for error recovery for lost packets. Retransmission due to lost packets will increase the server overhead of fetching the needed frames from the storage and sending them to the mobile client over the wireless network. There will be overhead on the server side as well as in the wireless network, and the overhead will depend on the network condition and the amount of packets lost. Therefore, we propose the TIRS technique to avoid the use of feedback techniques and retransmission mechanisms.

3 The interleaving distance algorithm

Claypool and Zhu [7] proposed video interleaving as a repair technique to ameliorate the effect of frame losses from the streaming video. They re-sequence the frames in a video stream at the sender side and stream them through the wireless network to the receiver side. They proposed an Interleaving Distance Algorithm (IDA) for recording television video programs before they are encoded by using MPEG-1. The decoding tool used is the Berkeley MPEG-2 player. In their study, the interleaving distances 2 and 5 were chosen: 2 for short distance interleaving and 5 for long distance interleaving. The interleaving distance determines how long time consecutive frames are spread out in the video stream.

The Group of Pictures (GOP) length [7] was used as a basis for their interleaving technique. For a GOP of 9 frames (used in their paper) and an interleaving distance of 2, they first encode and transmit the first 9 even frames, 2, 4, 6, 8, 10, 12, 14, 16, 18, in the original video, and then they encode and transmit the first 9 odd frames, 1, 3, 5, 7, 9, 11, 13, 15, 17, as shown in Fig. 1. After that they continue with frames 20, 22, 24, 26, 28, 30, 32, 34, 36 and then frames 19, 21, 23, 25, 27, 29, 31, 33, 35, and so on. The same mechanism is applied for 5 distance interleaving of 9 frames per GOP.

The interleaving of IDA is shown in Fig. 1 for different interleaving distances. In this case, we interleave 30 frames. Figure 1e corresponds to the case with interleaving distance 5 and GOP length 9.

The interleaving mechanism in IDA changes the frame positions in the video before it is compressed and streamed over the wireless network. The number of buffers on the

Fig. 1 Frame sequences with different interleaving distances and GOP length 9

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 1819 20 21 22 23 24 25 26 27 28 29 30 …

a Original sequence

1 3 5 7 9 11 13 15 17 2 4 6 8 10 12 14 16 18 …

b Interleaving distance 2

1 4 7 10 13 16 19 22 25 2 5 8 11 14 17 20 23 26 3 6 9 12 15 18 21 24 27 …

c Interleaving distance 3

1 5 9 13 17 21 25 29 33 2 6 10 14 18 22 26 30 34 3 7 11 15 19 23 27 31 35 4 8 12 16 20 24 28 32 36 …

d Interleaving distance 4

1 6 11 16 21 26 31 36 41 2 7 12 17 22 27 32 37 42 3 8 13 18 23 28 33 38 43 4 9 14 19 24 29 34 39 44 5 10 15 20 25 30 35 40 45 …

e Interleaving distance 5

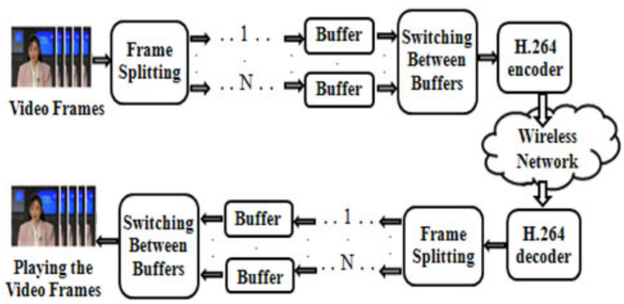


Fig. 2 Streaming video as IDA over wireless network

sending and receiving side needed to hold the frames is equal to the interleaving distance, as shown in Fig. 2. Frames in the video sequence will be split according to the interleaving distance N. If $N = 3$, as shown in Fig. 1c, the frames will be forwarded to three different buffers. The first buffer will hold frames number 1, 4, 7, 10, 13, 16, 19, 22, 25, the second buffer will hold frames number 2, 5, 8, 11, 14, 17, 20, 23, 26, and the third buffer will hold frames number 3, 6, 9, 12, 15, 18, 21, 24, 27. The size of each buffer is equivalent to the length of the GOP.

The video frames will be compressed and streamed based on switching between the buffers to create the frame sequence interleaving order. For interleaving distance $N = 3$, it sends the frame group from the first buffer, followed by the frame group from the second buffer, followed by the frame group from the third buffer, followed by the frame group from the first buffer and so on. For instance, we transmit the first 27 frames in the following order: 1, 4, 7, 10, 13, 16, 19, 22, 25, 2, 5, 8, 11, 14, 17, 20, 23, 26, 3, 6, 9, 12, 15, 18, 21, 24, 27.

In the normal case, the mobile device will start playing the video as soon as the frames are received (with the possible exception of jitter buffers). When using interleaving, frames will arrive in a different order according to IDA. When the mobile device starts receiving the interleaved video stream, it will decode the video using the H.264 ffmpeg decoder [27]. The received frames will be forwarded to the appropriate buffers, i.e. the odd buffer or the even buffer when using interleaving distance 2. The general case with interleaving distance N is shown in Fig. 2.

The latency time for playing the video is based on the interleaving distance and the GOP. The waiting time will be (GOP $\times$ $(N - 1)$)/transmission rate**,** as an example, if we assume that the IDA interleaving distance is 3 as shown in Fig. 1c, and then the waiting time before start playing the video is 0.6 s given a video with 30 frames per second. The longer the interleaving distances applied, the more is the waiting time needed before we can start playing the video on the mobile device. In Sect. 6 we investigate the video size using IDA and H.264 for interleaving distances 2, 3, 4 and 5.

4 The time interleaving robust streaming technique

In this section, we introduce the TIRS technique to stream video frames. The idea behind TIRS is to avoid frozen pictures in the mobile device when streaming over unreliable networks. This can be done by streaming the video as groups of even and odd frames. The TIRS technique will be applied before the video has been encoded and compressed by the H.264 encoder. TIRS can be implemented for different interleaving times (Δt) to distribute frames to different positions in the streaming sequence and to avoid a consecutive loss of frames in the streaming video.

In the following example $\Delta t = 1$ s. In TIRS, a sequence of even frames is followed by a sequence of odd frames. We assume 30 frames per second and since $\Delta t = 1$ s we get a frame group (FG) of 30 frames (Δt multiplied with the transmission rate). For each FG, we have two subgroups: Fe and Fo, which are the even and odd frames belonging to the same FG as shown in Fig. 3.

The second step is to create the streaming video based on the interleaving technique. The even and odd frames will be grouped and distributed in different positions in the streaming sequence; we start with Fe followed by Fo.

The even frames (Fe) from FG-1 and FG-2 will be grouped and streamed followed by a sequence of odd frames (Fo) from FG-1, FG-2, FG-3 and FG-4, and followed by a sequence of even frames (Fe) from FG-3, FG-4, FG-5 and FG-6 and so on. For example, the frames will be transmitted in the following order: 2, 4,..,58, 60, 1, 3,…,117, 119, 62, 64,…,178, 180, 121, 123,…, as shown in Fig. 4. The reason why we send four groups of odd frames instead of two groups is that we want to minimize the reduction of time redundancy. This will increase the compression rate when using H.264. There are two buffers: the even buffer contains the Fe frames and the odd buffer contains the Fo frames.

To implement the TIRS technique, a switching between buffers is used to switch between groups of even and odd frames according to the time interleaving parameters and before it has been encoded by the H.264 encoder. After that, the time interleaving streaming sequence will be streamed through the wireless network as shown in Figs. 4 and 5. When the mobile device starts receiving the video, the H.264 decoder will decode the video and the frames will be split into even and odd frames and forwarded to two buffers.

At start-up, the even frames will be delayed by Δt waiting for the odd frames to arrive to the odd buffer. The number of frames that has arrived to the even buffer is $V = \Delta t \times R$, where R is the number of frames per second received by the buffer. This means that the size of each buffer is V.

Figure 6a shows how the frames are stored in the even buffer, and Fig 6b shows the odd buffer. At time Δt, the

Fig. 3 Video streaming sequence

FG-1	FG-2
1 2 3 4 … 18 19 20 21 … 28 29 30	31 32 33 34 … 48 49 50 51 … 58 59 60
1 second	1 second

a Original stream

FG-1		FG-2	
Fe1	Fo1	Fe2	Fo2
2 4 6 ... 26 28 30	1 3 5... 25 27 29	32 34 36 ... 56 58 60	31 33 35... 55 57 59
0.5 second	0.5 second	0.5 second	0.5 second
1 second		1 second	

b Frames interleaving within the same frame group (FG)

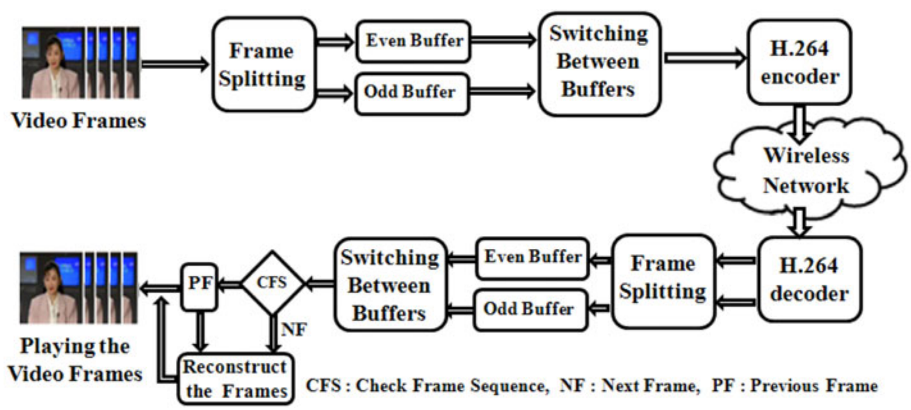
Fig. 4 The proposed TIRS technique, where $\Delta t = 1$ s

FG-1 FG-2 FG-3 FG-4 FG-5 FG-6 FG-7 FG-8 …

Fe1 Fo1 Fe2 Fo2 Fe3 Fo3 Fe4 Fo4 Fe5 Fo5 Fe6 Fo6 Fe7 Fo7 Fe8 Fo8 …

Fe1 Fe2	Fo1 Fo2 Fo3 Fo4	Fe3 Fe4 Fe5 Fe6	Fo5 Fo6 Fo7 Fo8	…
2 4 ... 58 60	1 3 5 ... 115 117 119	62 64 66 ... 178 180	121 123 125... 237 239	…
1s	2s	2s	2s	…

Fig. 5 Streaming video using TIRS over wireless networks



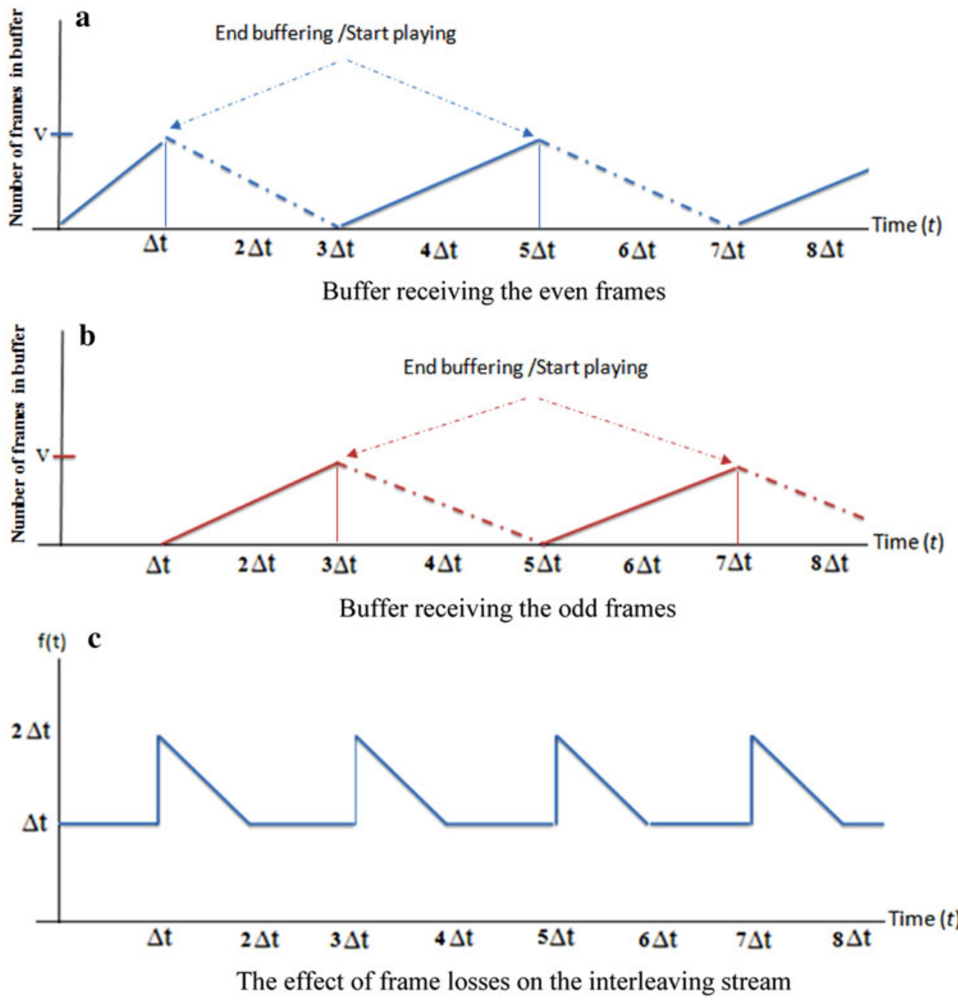
playout of the video starts. Figure 6a, b shows that after the initial start-up, i.e. after time Δt, the sum of the number of frames in the two buffers is always V.

A check frame sequence (CFS) will be used to check the missing frames from the streaming video [3]. If there is a missing frame, a reconstruction mechanism will be applied to create the missing frame based on linear interpolation between the next and previous frames [20].

5 The effect of losses on the interleaving frames

Streaming video over unreliable network could have a significant effect on the video quality, especially when a consecutive sequence of frames is lost. The playout of a video starts when there are available frames in the buffer. In the TIRS case, we need to compensate for the delay due to frame group (FG) interleaving. Interleaving

Fig. 6 The buffering behaviour corresponding to the frame types and frame lost



has the advantage that we can use frame interpolation to tolerate loss of frame sequences up to a certain length. By using long interleaving intervals (e.g. $\Delta t = 2$ s instead of $\Delta t = 1$ s), we can obviously tolerate losses of longer frame sequences, which is good in case of long outages. On the other hand, long interleaving intervals means that the delay due to buffering gets longer, which is a disadvantage in many cases, e.g. when looking at live events and when changing from one channel to another when watching TV. Consequently, we would like to have short buffer delays while at the same time being able to tolerate losses of long frames sequences (long outages).

Figure 6c shows *f(t)* which is the maximum outage length that we can handle at time t given that no frames outside of the outage are lost; $t = 0$ marks the start of the video. This means that the video will be shown at the receiver at Δt after it has been transmitted from the sender. The sender will start by sending even frames based on the value of Δt and then it will send odd frames for $2\Delta t$ and then even frames for $2\Delta t$, and then odd frames for Δt, and so on.

If the outage starts at $t = 0$ until Δt, then all the even frames are lost and if the odd frames are received by the mobile device (assuming that no additional frames are lost), then the lost even frames can be recreated by interpolating the odd frames from Δt to $2\Delta t$.

If the outage starts at Δt until $3\Delta t$, then the lost odd frames can be recreated by interpolating the receiving even frames from 0 to Δt.

If the outage starts at $t = 0$ until $1.5\Delta t$, then the outage duration is longer than Δt and in this case we face a frozen video.

We can observe that if the outage occurs in the initial stage of the streaming video and its greater than Δt, then we will face a frozen video, but if the outage occurs between Δt and $3\Delta t$ then there is the possibility to interpolate the missing frames. If the outage occurs between $2\Delta t$ and $4\Delta t$ and it is greater than Δt, we will face a frozen picture.

6 The effect of interleaving on the file size

Compressing the video frames based on IDA and TIRS by using the H.264 ffmpeg encoder [27] is affected by the amount of redundancy that can be removed. When the frame positions change, the temporal locality decreases which leads to increased size of the video stream for both techniques. IDA and TIRS have been applied on the video test sequences Akiyo, Foreman, News, Waterfall and Football [26]. The chosen videos are well known as professional test sequences that have different characteristics, with a transmission rate of 30 frames per second, and each video has a 10 s duration with QCIF (176 $\times$ 144 pixels) resolutions.

6.1 The effect of IDA on the file streaming size

As shown in Table 1, the sizes of the compressed video clips increase differently for different interleaving distances. The reason for this is that the frames are moved to different scene positions, which affects the temporal locality and thus also the ability to compress the video using H.264.

The size increase varies from one video to another since each video has different features which affect the temporal locality, and thus also the size, differently. As the interleaving distance in IDA increases the temporal locality is in most cases reduced, and the size of the video therefore increases. The exception is the Foreman video, where the size for IDA-5 is somewhat smaller than the size using IDA-4; this is a coincidence.

It is worth noticing that the increase of size for the highly dynamic Football video is not bigger than the increase of size for the more static videos. It seems that even if the dynamic videos clearly contain less temporal locality, this does not mean that the relative reduction of temporal locality (and thus relative increase of size) due to interleaving is larger for dynamic videos than for more static videos.

6.2 The effect of TIRS on the file streaming size

The effect of different time interleaving on the streaming file size is shown in Table 2, where the values of Δt considered are 1, 2 and 3 s. For $\Delta t = 1$ s, interleaving the initial stream is based on streaming the even frames from two groups, followed by odd frames from four groups and so on, as shown in Fig. 4. For $\Delta t = 2$ s, interleaving the initial stream is based on streaming the even frames from 4 groups followed by odd frames from 8 groups and so on, while for $\Delta t = 3$ s interleaving the initial stream is based on streaming the even numbers from 6 groups followed by odd frames from 12 groups and so on.

The $\Delta t = 1$, 2 and 3 s interleaving is applied before the videos are compressed by the H.264 ffmpeg encoder [27]. The sizes of the compressed video clips increase differently and for different videos, as frames are moved to different scene position, affecting the temporal locality as shown in Table 2.

The changes in the streaming file size for the three interleaving times are rather similar to each other except the News video as the 2 s interleaving show much less increases in the streaming size. This is probably due to a coincidence, i.e. the frames happen to be moved to a position where there is similar data and the encoder will hence be able to remove more redundant data.

It is again worth noticing that the increase in size for the highly dynamic Football video is not bigger than the

Table 1 The compressed video size (bytes) and the number of packets for IDA

Videos	Akiyo	Foreman	News	Waterfall	Football
Original	2,886,120	7,151,874	4,115,786	6,679,052	8,524,180
No. of packets	1,961	4,859	2,796	4,537	5,791
IDA-2	**3,093,878**	**7,348,202**	**4,465,012**	**7,243,318**	**8,597,744**
No. of packets	2,102	4,992	3,033	4,921	5,841
Increase	**7.19%**	**2.73%**	**8.48%**	**8.44%**	**0.86%**
IDA-3	**3,227,560**	**7,429,644**	**4,637,112**	**7,474,446**	**8,622,794**
No. of packets	2,193	5,047	3,150	5,078	5,858
Increase	**11.83%**	**3.88%**	**12.66%**	**11.90%**	**1.16%**
IDA-4	**3,295,714**	**7,461,896**	**4,726,700**	**7,617,466**	**8,634,452**
No. of packets	2,239	5,069	3,211	5,175	5,866
Increase	**14.19%**	**4.33%**	**14.84%**	**14.05%**	**1.29%**
IDA-5	**3,385,274**	**7,450,040**	**4,778,734**	**8,711,264**	**8,667,018**
No. of packets	2,300	5,062	3,246	5,918	5,888
Increase	**17.29%**	**4.16%**	**16.10%**	**30.42%**	**1.67%**

Table 2 The compressed video size (bytes) and the number of packets for TIRS

Videos	Akiyo	Foreman	News	Waterfall	Football
Original	2,886,120	7,151,874	4,115,786	6,679,052	8,524,180
No. of packets	1,961	4,859	2,796	4,537	5,791
$\Delta t = 1$	**3,124,282**	**7,365,602**	**4,498,224**	**7,237,768**	**8,617,482**
No. of packets	2,122	5,004	3,056	4,916	5,854
Increase	**8.25%**	**2.98%**	**9.29%**	**8.36%**	**1.09%**
$\Delta t = 2$	**3,122,252**	**7,379,988**	**4,261,158**	**7,238,698**	**8,598,686**
No. of packets	2,121	5,013	2,895	4,918	5,841
Increase	**8.18%**	**3.18%**	**3.53%**	**8.37%**	**0.87%**
$\Delta t = 3$	**3,121,174**	**7,364,790**	**4,491,916**	**7,234,486**	**8,620,228**
No. of packets	2,120	5,003	3,052	4,915	5,856
Increase	**8.14%**	**2.97%**	**9.13%**	**8.31%**	**1.12%**

increase in size for the more static videos. It seems that even if the dynamic videos clearly contain less temporal locality, this does not mean that the relative reduction of temporal locality (and thus relative increase of size) due to interleaving is larger for dynamic videos than for more static videos.

7 Comparison between IDA, Reed–Solomon and TIRS

In this section, we will first compare TIRS and IDA and then TIRS and Reed–Solomon-based approaches, e.g. approaches like the one suggested by Cai and Chen [5].

The compressed size of the video stream will increase for both the IDA and TIRS. The increases of the video size will be different from one video to another. For the IDA, the compressed size of the video increases as the interleaving distance grows. The reason for this is that for long interleaving distances, the frames are moved longer distances, thus reducing the temporal locality. The compression rate using TIRS seems to be rather independent of Δt.

To handle long outages using IDA, we need to increase the interleaving distance. However, even if we use long interleaving distances, two or more consecutive frames can be lost as soon as the length of the outage exceeds the GOP length. For instance, if we use interleaving distance 5 and suffer from an outage of two times the length of the GOP, we will lose two consecutive frames at 9 places (GOP length is 9). Each sequence of two lost frames will be separated by three correct frames (due to using interleaving distance 5). Losing a sequence of two or more consecutive frames will obviously decrease the quality of the video. In TIRS we can handle outages in the range of Δt to $2\Delta t$, without losing two consecutive frames in the video (see Fig. 6c for details).

For short interleaving, e.g. IDA-2 and $\Delta t = 1$ s in TIRS, the size increase is more or less the same for IDA and TIRS. This is not so surprising since TIRS is IDA-2, but instead of using the GOP length we use a frame group length of Δt. The only other difference between TIRS and IDA-2 is that in TIRS we reorder the frame groups so that we minimize the number of switches between odd and even frames, thus avoiding unnecessary loss of temporal locality.

For long interleaving that can handle long outages, e.g. IDA-5 and $\Delta t = 3$ s in TIRS, the size increase is significantly larger in IDA, as the neighbouring frames are separated with a distance of five times the GOP length, i.e. $5 \times 9 = 45$ frames. In TIRS with $\Delta t = 3$ s, neighbouring frames are separated with three times the transfer rate, i.e. $3 \times 30 = 90$ frames. This means that TIRS with $\Delta t = 3$ s can handle longer outages than IDA-5. In general, the outage length that can be tolerated by IDA is less than that of TIRS, since TIRS can increase the value of Δt without any significant additional overhead.

The number of buffers needed for IDA depends on the interleaving distance (N). For TIRS, we only need two buffers, which reduce the implementation complexity. The initial waiting time for the video for IDA is increased as the interleaving distance (N) is increased. For TIRS approach, the initial waiting time is Δt.

An alternative to interleaving approaches like IDA and TIRS is to use FEC. A well-known FEC scheme is Reed–Solomon [4]. Cai and Cheng [5] discuss different ways of interleaving symbols (frames), and thus spreading out the errors in an optimal way before applying Reed–Solomon coding. By spreading out the errors in an optimal way, one gets maximum benefit from the parity symbols. The way that Cai and Cheng use interleaving, however, does not change the fundamental properties of their FEC-based approach. The fundamental difference between the interleaving (such as that used in TIRS and IDA) and FEC approaches is that FEC approaches try to transmit all frames correctly by correcting the errors using parity symbols, while the interleaving approaches approximate missing frames by interpolating neighbouring frames. To handle long outages in a FEC-based approach, we either need to spread out the information over very long time intervals, thus generating a significant start-up delay, and/or use a lot of the bandwidth for parity. For instance, if we want to handle a 1 s outage with a Reed–Solomon code with 10% parity overhead, we need to code the data for the missing second in a 110/5 = 22 s block (using the standard notation for Reed–Solomon, we assume that $k = 100$, $t = 5$ and a data transmission rate of five symbols per second). This means that to handle outages up to 1 s, an FEC-based approach would cause a 22 s delay instead of a 1 s delay, which we get with TIRS (which has less than 10% overhead). If we increased the parity overhead in Reed–Solomon to 20%, we would, at least, need a 60/5 = 12 s delay for the 1 s outage case (here we assume that $k = 50$, $t = 5$ and a data transmission rate of five symbols per second). If we increased the parity overhead in Reed–Solomon to 40%, we would need at least 35/5 = 7 s delay for the 1 s outage case (here, we assume that $k = 25$, $t = 5$ and a data transmission rate of five symbols per second). In general, we need a $2\Delta t\ (100 \pm x)/x$ second delay in order to handle a Δt outage using x % parity overhead, This means that there is a very significant cost in terms of overhead and/or delay due to exactly recreating the missing frames with FEC, instead of interpolating them with interleaving approaches such as TIRS and IDA.

8 Subjective viewing test

8.1 Testing methods

It is well known that the peak signal-to-noise ratio (PSNR) does not always rank the quality of an image or video sequence in the same way as a human being would. There are many other factors considered by the human visual system and the brain [14].

One of the most reliable ways to assess the quality of a video is subjective evaluation using mean opinion score (MOS). MOS is a subjective quality metric obtained from a panel of human observers. It has for many years been regarded as the most reliable form of user-perceived quality measurements [15].

MOS measurements are used to evaluate the video quality in this study. We follow the guidelines outlined in the BT.500-11 recommendation of the radio communication sector of the International Telecommunication Union (ITU-R). The score grades in this method range from 0 to 100. These grades are mapped to quality ratings on the five-grade category scale labelled: Excellent (5), Good (4), Fair (3), Poor (2), and Bad (1). The physical laboratory environment used, with controlled lighting and set-up, conforms to the ITU-R recommendation [10].

The subjective experiment was conducted at Blekinge Institute of Technology in Sweden. The users observed the video clip with frozen picture and the proposed TIRS technique, with the participation of 30 non-expert test subjects, 26 males and 4 females. They were all university staff and students and within an age range from 23 to 37 years.

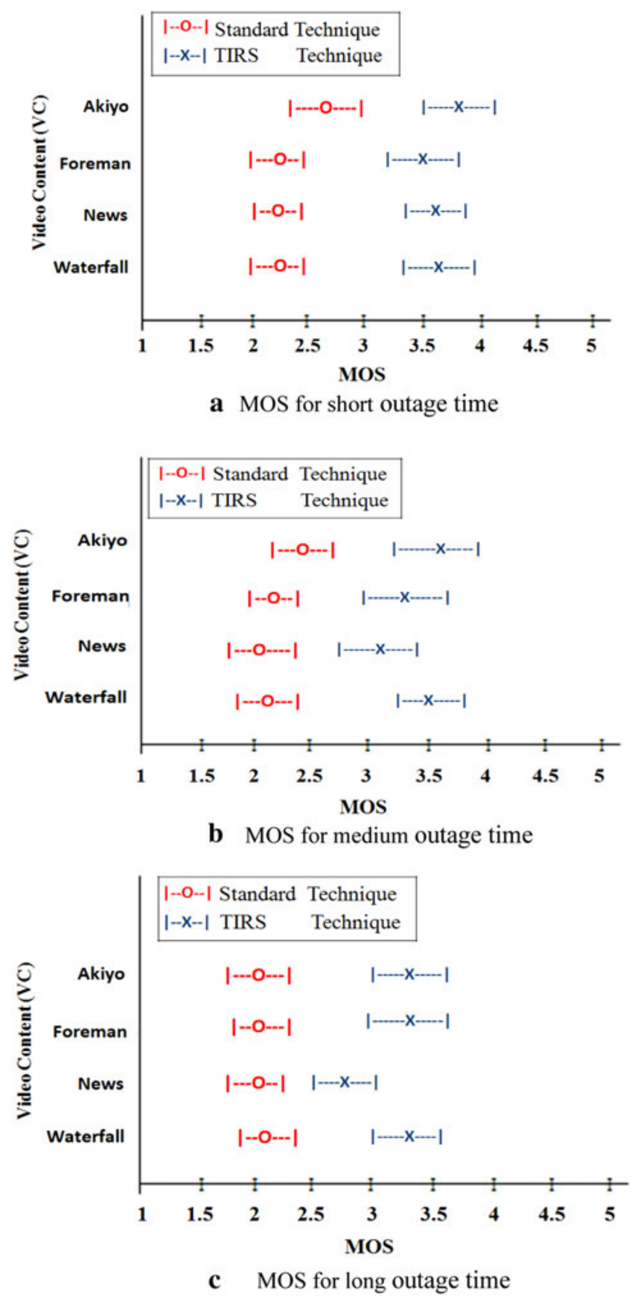


Fig. 7 The MOS for different video contents and for different outage time, showing the average and the standard deviation

8.2 Testing materials and environments

The study is done by coding and compressing the video test sequences Akiyo, Foreman, News and Waterfall [26] using the H.264 ffmpeg codec [27] for the proposed interleaving technique. The chosen videos are coded with a resolution of 176 × 144. The transmission rate is 30 frames per second and the number of frames transmitted is 1,800 (corresponding to 60 s).

The outage duration times in this study are 650, 1,300 and 2,000 ms for short, medium and long outages, respectively. The video sequences are shown on a 17-inch EIZO FlexScan S2201 W LCD computer display monitor with a resolution of 1,680 × 1,050 pixels. The video sequences for the standard technique with no interleaving and TIRS are displayed with a resolution of 176 × 144 pixels in the centre of the screen with a black background.

9 Experimental results

For the TIRS technique, we use $\Delta t = 1$ s. This means that we will be able to handle all short outages, some medium outages and maybe some long outages depending on when the outage occurs. We calculated the conventional statistics such as the average (Avg), the standard deviation (StD), coefficient of variation (CV) and the 95% confidence interval (CI) for the scores. The statistical analysis of the data from the subjective experiments is shown in Fig. 7 and Table 3.

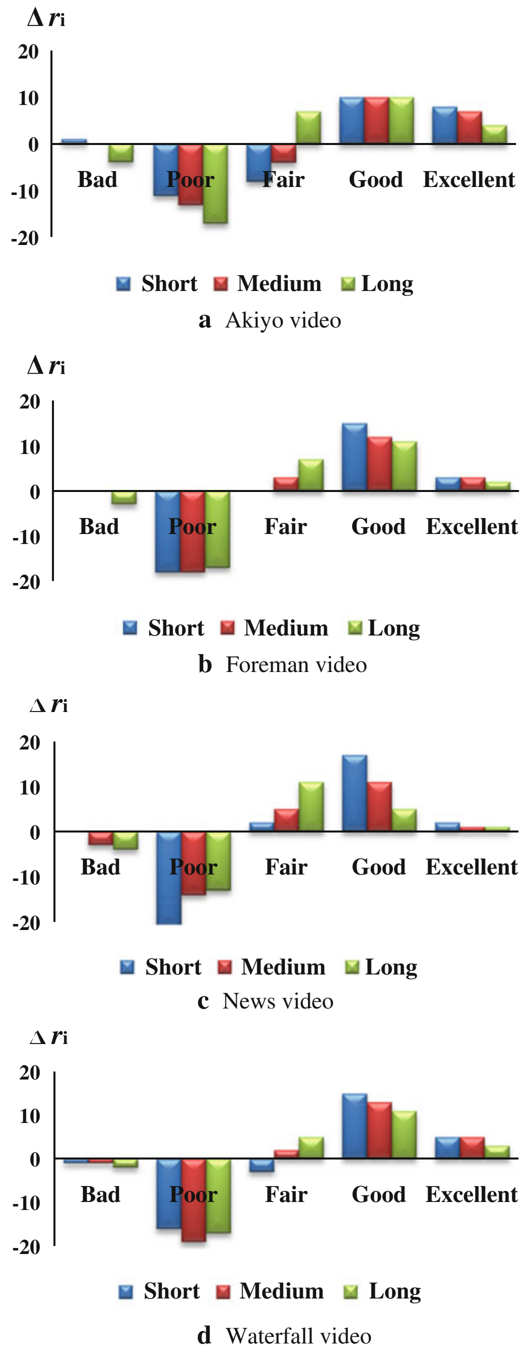
The MOS for TIRS is significantly better than the MOS for the standard technique for all videos. For the standard technique with no interleaving, it is clear that the viewers are disturbed in the frozen pictures in the test videos. For the short outages, the MOS is lower than 3 for all videos, except for the Akiyo video which had a MOS lower than 3.5, on the five-level quality scale. The reason for a higher score for Akiyo is that the video contains less motion than the other videos. For medium and long outages, the MOS is lower than 3, due to the higher percentage of missing frames, which the viewer notices as frozen pictures.

For medium and long outages, the MOS for TIRS is somewhat lower than for the case with short outages. The reason for this is that $\Delta t = 1$ s cannot handle all of the medium and long outages. Figure 6c shows that the length of the outage that can be handled depends on when the outage occurs. In our case ($\Delta t = 1$ s), we can handle outages of at least 1 s and at best 2 s.

Let r denote the vector of the number of ratings in each category provided by the users in a certain experiment, i.e. $r = [r_{\mathrm{Bad}(1)}, r_{\mathrm{Poor}(2)}, r_{\mathrm{Fair}(3)}, r_{\mathrm{Good}(4)}, r_{\mathrm{Excellent}(5)}]$. For each outage duration l, we obtain one vector of ratings r^l without taking advantage of interleaving, and a corresponding vector $r^{l,\mathrm{TIRS}}$ when using TIRS. The difference vector $\Delta r = r^{l,\mathrm{TIRS}} - r^l$ illustrates the change in the number of user rankings per category: if $\Delta r_i < 0$, then the number of corresponding user ratings in category i has decreased when employing the TIRS technique, and vice versa. If TIRS yielded an improvement, one would actually expect a decrease in the number of negative rankings ($\Delta r_{\mathrm{Bad}(1)} \leq 0$, $\Delta r_{\mathrm{Poor}(2)} \leq 0$) and on the other hand a growth in the number

Table 3 The statistical data analysis for different videos and for different outage time

	Frozen Technique				Proposed Video			
	Akiyo	Foreman	News	Waterfall	Akiyo	Foreman	News	Waterfall
a. Statistical data analysis for short outage time								
Avg	2.70	2.27	2.27	2.27	3.87	3.57	3.67	3.70
StD	0.65	0.52	0.45	0.52	0.97	0.90	0.66	0.92
CV	24%	23%	20%	23%	25%	25%	18%	25%
95% CI	9.0%	8.5%	7.4%	8.5%	9.3%	9.4%	6.7%	9.3%
b. Statistical data analysis for medium outage time								
Avg	2.40	2.20	2.03	2.17	3.63	3.40	3.13	3.63
StD	0.56	0.48	0.61	0.53	1.07	0.97	0.97	0.96
CV	23%	22%	30%	24%	29%	29%	31%	26%
95% CI	8.7%	8.1%	11.2%	9.1%	11.0%	10.6%	11.5%	9.8%
c. Statistical data analysis long outage time								
Avg	2.00	2.03	1.93	2.10	3.43	3.30	2.87	3.37
StD	0.59	0.56	0.58	0.55	0.97	0.92	0.90	0.96
CV	30%	28%	30%	26%	28%	28%	31%	28%
95% CI	11.0%	10.3%	11.2%	9.8%	10.5%	10.4%	11.7%	10.6%



a Akiyo video

b Foreman video

c News video

d Waterfall video

Fig. 8 Difference in the number of user ratio per category when employing the TIRS technique and for different outage times

of positive rankings ($\Delta r_{\text{Good}(4)} \geq 0$, $\Delta r_{\text{Excellent}(5)} \geq 0$). Figure 8 supports this assumption and also shows that the biggest change due to the TIRS technique are in the categories Poor (2) and Good (4), whereas the change in the categories Bad (1) and Excellent (5) is more limited.

The videos presented to the viewers resulted in a wide range of perceptual quality ratings in the experiments. When we analyse the scores we feel that TIRS is a satisfactory technique to distribute the missing frames in the streaming video, to eliminate the frozen pictures and provide a smooth video on the mobile screen.

10 Conclusions

Wireless network transmission errors in forms of frame losses could have a major impact on the end user experience of real-time videos. A Time Interleaving Robust Streaming (TIRS) technique is proposed to minimize the effect of outages and frame losses. Streaming the video frames according to TIRS makes it possible to significantly reduce negative effects when a sequence of consecutive video frames is lost during the transmission.

The advantage of TIRS is that lost frame sequences will be spread out on the streaming video with the ability to reconstruct it at the receiver side. This means that if 30 consecutive frames are lost, we do not get a 1 s freezing (assuming 30 frames per second); the 30 frames are distributed to two frame groups related to another two frame groups. Because of this, the lost frames can be interpolated by using the surrounding frames. The disadvantage of interleaving is that the size of the video stream is increased slightly (less than 10%).

TIRS offers some important advantages compared to the previous IDA interleaving technique when we have long outages (more than 1 s). The first advantage is that the nature of the interleaving scheme in IDA is such that even if we use long interleaving distances, two or more consecutive frames can be lost as soon as the length of the outage exceeds the (Group of Pictures) GOP length. In TIRS, we can handle outages in the range of Δt to $2\Delta t$, without losing two consecutive frames in the video as shown in Fig. 6c. The second advantage of TIRS compared to IDA is that in TIRS the interleaving length (i.e. Δt) can be increased without affecting the size overhead. In IDA, the size overhead increases significantly when the interleaving distance grows.

Our evaluation was based on user panel tests and showed that there was a significant quality improvement when using TIRS.

Acknowledgments We would like to thank the staff and students at Blekinge Institute of Technology, Sweden, for participating in the subjective experiments. We also would like to thank the Swedish Knowledge Foundation for sponsoring a part of this work through the project QoEMoS (21601420).

References

1. Apostolopoulos, J.: Reliable video communication over lossy packet networks using multiple state encoding and path diversity. In: Proceedings of Visual Communications and Image Processing, pp. 392–409 (2001)

2. Argyriou, A., Madisetti, V.: Streaming H.264/AVC video over the internet. In: IEEE Consumer Communications and Networking Conference (CCNC'04), pp. 169–174 (2004)
3. Aziz, H.M., Fiedler, M., Grahn H., Lundberg, L.: Streaming video as space-divided sub-frames over wireless networks. In: the Proceedings of the 3rd Joint IFIP Wireless and Mobile Networking Conference, (WMNC'10) (2010)
4. Benslimane, A.: Multimedia multicast on the internet. Wiley-ISTE publisher (2007)
5. Cai, J., Chen, C.W.: FEC-based video streaming over packet loss networks with pre-interleaving. In: Proceeding for the International Conference on Information Technology: Coding and Computing, pp. 10–14 (2001)
6. Chen, L.-J., Sun, T., Sanadidi, M.Y., Gerla, M.: Improving wireless link throughput via interleaved FEC. In: Proceedings of the 9th IEEE Symposium on Computers and Communications, pp. 539–544 (2004)
7. Claypool, M., Zhu, Y.: Using interleaving to ameliorate the effects of packet loss in a video stream. In: Proceedings of the International Workshop on Multimedia Network Systems and Applications (MNSA) (2003)
8. Cranley, N., Davis, M.: Video frame differentiation for streamed multimedia over heavily loaded IEEE 802.11e WLAN using TXOP. In: Proceedings of 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07) (2007)
9. Dapeng, W., Hou, Y.T., Zhu, W., Zhang, Y.-Q., Peha, J.M.: Streaming video over the internet: approaches and directions. IEEE. Tran. Circ. Syst. Video. Tech. **11**(3), 282–300 (2001)
10. International Telecommunication Union. Methodology for the subjective assessment of the quality of television pictures. ITU-R, Rec. BT.500-11, (2002)
11. Jianhua, W., Jianfei, C.: Quality-smoothed encoding for real-time video streaming applications. In: Proceedings of the 9th International Symposium on Consumer Electronics (ISCE'05), pp. 445–449 (2005)
12. Lo, A., Heijenk, G., Niemegeers, I.: Performance evaluation of MPEG-4 video streaming over UMTS networks using an integrated tool environment. In: Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'05), pp. 676–682 (2005)
13. Lei, H., Fan, C., Zhang, X., Yang, D.: QoS aware packet scheduling algorithm for OFDMA systems. In: IEEE 66th Vehicular Technology Conference, pp.1877–1881 (2007)
14. Martinez-Rachm, M., Lopez, O., Pinol, P., Malumbres, M.P., Oliver, J., Calafate, C.T.: Quality assessment metrics vs. PSNR under packet loss scenarios in MANET wireless networks. In: Proceedings of the International Workshop on Mobile Video (MV '07), pp. 31–36 (2007)
15. Martinez-Rach, M., Lopez, O., Pinol, P., Malumbres, M.P., Oliver, J., Calafate, C.T.: Behavior of quality assessment metrics under packet losses on wireless networks. XIX Jornadas de Paralelismov (2008)
16. Mancuso, V., Gambardella, M., Bianchi, G.: Improved support for streaming services in vehicular networks. In: Proceedings of ICC 2004 Conference, pp. 4362–4366 (2004)
17. Nafaa, A., Ahmed, T., Mehaoua, A.: Unequal and interleaved FEC protocol for robust MPEG-4 multicasting over wireless LANs. In: Proceedings of the IEEE International Conference on Communications (ICC'04), pp. 1431–1435 (2004)
18. Nafaa, A., Taleb, T., Murphy, L.: Forward error correction strategies for media streaming over wireless networks. IEEE. Commun. Mag. Featur. Top. New. Trends. Mob. Internet. Tech. Appl. **46**(1), 72–79 (2008)
19. Ong, E.P., We, S., Loke, M.H., Rahardja, S., Tay, J., Tan, C.K., Huang, L.: Video quality monitoring of streamed videos. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '09), pp. 1153–1156 (2009)
20. Peng, Y.-C., Chang, H.-A., Chen, C.-K., Kao, L., Kao, H. C.-J.: Integration of image stabilizer with video codec for digital video cameras. In: IEEE international symposium on circuits and systems (ISCAS'05), pp. 4871–4874 (2005)
21. Quan, H.-T., Ghanbari, M.: Asymmetrical temporal masking near video scene change. In: Proceedings of the IEEE International Conference on Image Processing (ICIP'08), pp. 2568–2571 (2008)
22. Richardson, I.E.G.R.: H.264 and MPEG-4 video compression video coding for next-generation multimedia. John Wiley & Sons Ltd (2003)
23. Schierl T., Kampmann M., Wiegand T., 2005. H.264/AVC interleaving for 3G wireless video streaming. In: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '05), pp. 868-871
24. Szymanski, H.T., Gilbert, D.: Internet multicasting of IPTV with essentially-zero delay jitter. IEEE transactions on broadcasting **55**(1), 20–30 (2009)
25. Tsai, M.-F., Ke, C.-H., Kuo, C.-I., Shieh, C.-K.: Path dependent adaptive forward error correction with multipath interleaving control scheme for video streaming over wireless networks. In: IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 1236–1239 (2009)
26. http://trace.eas.asu.edu/yuv/index.html (visited, 1/11/2009)
27. http://www.ffmpeg.org (visited, 4/03/2011)