Simple Packet Scheduling Method for Data-Partitioned Video Streaming Over Broadband Wireless

Laith Al-Jobouri, Martin Fleury, M. Ghanbari
School of Computer Science and Electronic Engineering
University of Essex
Colchester, U.K.
{lamona, fleum, ghan}@essex.ac.uk

Ismail A. Ali
Dept. of Electrical and Computer Engineering
University of Duhok
Duhok, Iraq
iaali@uod.ac

ABSTRACT

Source-coded data partitioning of a compressed video stream protects more important data, reducing the risk of error corruption in wireless networks. However, because the data partitions of a video slice are each assigned to different network packets, in congestion-prone networks the increased number of packets per slice and their size disparity may increase the packet loss rate from buffer overflows. This paper recommends packet-size dependent scheduling as a relatively simple way of alleviating the problem, which problem remains real whenever there is an adverse distribution of packet sizes entering an access network. The paper also contributes an analysis of partition and packet sizes as a prelude to scheduling regimes.

Index Terms— broadband wireless**,** access network congestion, data partitioning, multimedia networking, packet scheduling

1. INTRODUCTION

Despite increases in capacity, broadband wireless systems still suffer from limitations in bandwidth capacity [1], leading to congestion at base-station (BS) buffers. To reduce the risk of buffer overflow, this paper proposes an enhanced packet scheduling regime for video streams encoded using data partitioning [2]. Forward Error Correction (FEC) and error control [3] together with error resilience and concealment methods [4] can offer video streams protection from wireless channel errors. However, they are powerless against packet drops through buffer overflow. Unfortunately also, due to the predictive nature of video coding [5], most packet losses have an effect that extends in time until the decoder is reset (intra refreshed).

One source-coded error resilience scheme, data partitioning, in its current form is vulnerable to congestion due to the number of small packets that are produced. Data partitioning, which is a form of layered error resilience, can provide graceful degradation of video quality and as such has found an application in mobile video streaming [2]. In an H.264/AVC (Advanced Video Coding) codec, when data partitioning is enabled, every slice is divided into three separate partitions: partition-A has the most important data, including motion vectors (MVs); -B contains intra coefficients; and -C contains inter coefficients, the least important data in terms of reconstructing a video frame at the decoder. These data are packed into three types of Network Abstraction Layer units (NALUs) output by the codec. The importance of each NALU-bearing packet is identified in the NALU header. Though it is possible to aggregate (or segment) NALUs) before encapsulation in IP/UDP/RTP packets this would be to neglect the advantages of retaining smaller partition-A and -B bearing packets [3] because smaller packets have a lower probability of channel error.

To reduce the risk of buffer overflow for data-partitioned video streams, this paper proposes a simple packet-scheduling method, which can work in addition to other channel error protection. Such packet-scheduling schemes, e.g. [6] [7] [8], act irrespective of physical-layer data scheduling and may be independent of data-link medium access control (though some schemes intervene at the Medium Access Control (MAC) sub-layer). The proposed packet scheduling method works by smoothing the packet-scheduling times across one or more frame intervals, according to allowable latency. Packets are allocated a scheduling time for output to a BS buffer in proportion to their size.

As such the scheduling method is relatively simple to implement, which is one of its attractions owing to the need to reduce latency for interactive services such as mobile video-conferencing. In the paper, the potential value of the application-layer scheduling approach is demonstrated in simulation which takes account of physical later packetization and scheduling. Currently, our investigations are at a preliminary stage. However, the paper also contributes a video-content-dependent of packetization issues suitable for others planning application packet scheduling or video smoothing algorithms [9].

The application-layer packet scheduling is demonstrated for IEEE 802.16, which is the standardized version of WiMAX wireless broadband technology [10]. WiMAX continues to be rolled out in many parts of the world that do not benefit from existing wired infrastructures or cellular networks. WiMAX is also cost effective in rural and suburban areas in some developed countries. It is designed to provide effective transmission at a cell's edge by the allocation to a mobile user of sub-channels with separated frequencies to reduce co-channel interference. Time Division Duplex (TDD) and effective scheduling of time slots between mobile stations (MSs) through Time Division Multiple Access (TDMA) (not to be confused with application-layer packet scheduling) increases spectral efficiency, while the small frame size of 5 ms selected by the WiMAX Forum can reduce latency for applications such as video conferencing. The transition to the higher data rates of IEEE 802.16m indicates the technological route that WiMAX will respond to the technological advances of its competitors, especially Long Term Evolution (LTE). However, we have experimented with version 802.16e, as this is backwards compatible with fixed WiMAX, which remains the most widely deployed version.

The remainder of this paper is organized as follows. Section 2 considers related packet scheduling schemes for video streaming. Section 3 then describes three components of our streaming methodology. Section 4 outlines the simulation model, which is applied in Section 5. Finally, Section 6 considers future work after the early findings of Section 5.

2. RELATED WORK

In [6], the authors combined scheduling at the transmitter with control of playout speed at the receiver, across a time-varying wireless link. This proposal required video-content-dependent decisions with the aim of maximizing video quality at the receiver. Decisions were driven by a Markov process informed by the motion activity of the transmitted video frames. In the scenario of [6], the scheduler discards late packets and catches up with delayed playout during poor wireless channel conditions. The receiver adapts its playout rate (the rate video frames are displayed at), slowing it down when video with low motion and, hence, low coding complexity occurs. The aim was to avoid playout interruptions, which are thought to be less disconcerting to the viewer than small slowdowns in the playout rate. The authors concluded by identifying the need for practical heuristics that reduce the complexity of their proposed scheme due to the need to inspect the content type, which remains a problem for real-time operation and which problem our simplified scheme seeks to address.

The optimal scheduling decision is again computed through a Markov process in [7]. However, the authors are aware that burst errors are a threat and, in particular, may disrupt feedback signals. Therefore, modeling is designed to take into account incomplete information about the receiver's state. The end result is improved video quality but delays to video packet transmission during poor channel states. In this early work, the impact on playout delay was not taken into account and the authors also acknowledge the need to develop heuristic decision-guiding rules.

In [8], the video content is taken into account to vary the packet scheduling decision at the transmitter. Inspection of the video content does imply an increase in end-to-end delay, as does the need to await protocol acknowledgments. The position of a packet's data within a Group-of-Pictures weights its importance to the playout process and, hence, determines its transmission schedule deadline. Similarly, the texture- and motion-complexity governed scheduling decisions when adapting the packet transmission order. Simplified wireless conditions were assumed to test the scheme. However, the authors note that channel coding is insufficient in itself during 'bursty' error conditions, requiring intelligent Automatic Repeat ReQuests (ARQs) and/or packet scheduling. The result of packet scheduling was improved objective video and a more even distribution of temporal distortion. Data partitioning was also turned on in some of the evaluation tests but with the MPEG-4 part 2 form of data partitioning in which each video packet is internally divided between shape/motion data and less important texture data.

The work in [11] is aimed at IEEE 802.11 WLANs running the Distributed Coordination Function (DCF) at the MAC layer, i.e. retransmitting data packets that are lost. By determining the importance of the video data within a packet, differing re-try limits occur. The codec includes the importance of the video data as side information with the video data. However, to achieve significant gains in video quality requires cross-layer intervention in the wireless protocol stack, as well as modification of the video codec. Research in [12] introduced another scheduling scheme that relies on video content side information placed in advance within the bitstream by the encoder. It is designed in a practical fashion for a multi-user, TDMA wireless environment such as High Speed Downlink Packet Access (HSPDA).

LTE leaves choice of packet scheduler to the vendor, resulting in many research proposals [13]. These are system-level schedulers, of which [14] combines the popular proportional fair algorithm with a higher-level scheduler that allocates the amount of data that each real-time flow can send within one time epoch.

3. METHODOLOGY

As a whole the WiMAX video-streaming scheme, Fig. 1, comprises three components: 1) source-coded data partitioning of the video bitstream; 2) adaptive rateless channel coding; and 3) application-layer packet scheduling. This Section now describes these three components.

A. *Data partitioning*

We have employed data partitioning. As previously remarked, data-partitioned H.264/AVC compressed video can be an effective means of placing more important data (as far a video decoding is concerned) in smaller, less error-prone packets in a wireless channel, provided low-quality

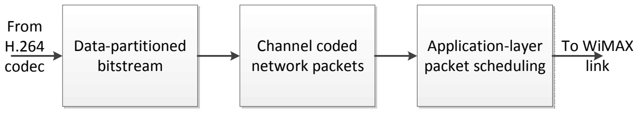


Figure 1. Overview of video-streaming scheme.

video (quantization parameter (QP) greater than 30) is not transmitted, as then the less important packets diminish in size. As is common in mobile video communication, in this paper an IPPPP… frame coding structure is employed, that is an intra-coded frame followed thereafter by predictively coded P-frames. Notice that B-slices are not permitted in the H.264/AVC Baseline profile, aimed at reducing the complexity of bi-predictive coding on mobile devices. Random Intra Macroblock Refresh (RIMR) (forcibly embedding a given percentage of randomly placed intra-coded macroblocks (MBs) in a P-frame) was turned on to counteract spatio-temporal error propagation that would otherwise occur in the absence of I-frames.

When data partitioning is enabled, every slice is divided into up to three separate partitions and each partition is encapsulated in either of type 2 to type-4 NALUs. DP-A, carried in an NALU of type 2, comprises the MB addresses and types, their motion vectors (MVs), and QPs. If any MBs in the frames are intra-coded, their frequency-transform coefficients are packed into a type-3 NALU or DP-B. DP-C contains the transform coefficients of the motion-compensated inter-coded MBs and is carried in an NALU of type-3. (DP-B and DP-C also contain Coded Block Patterns, compact maps indicating which blocks within each MB contain non-zero coefficients.)

Reconstruction of the other partitions is dependent on the survival of partition-A, though that partition remains independent of the other partitions. Constrained Intra-Prediction (CIP) [15] was set in order to make partition-B independent of partition-C. When only DP-A survives then its motion vectors can be employed in error concealment at the decoder using motion copy. When DP-A and DP-B survive then error concealment can combine texture information from DP-B when available, as well as intra concealment when possible. When DP-A and DP-C survive, to reconstruct it is necessary to partially discard DP-C data reliant on missing DP-B MBs and then reconstruct either using DP-A MVs or DP-A with DP-C texture data, when it is available.

The relative size of the data-partition packets is determined by the quality of the video, which in turn is governed by the QP of the MBs. The QP is set in the configuration file of the H.264/AVC codec, prior to compression. Fig. 2 is a comparison between the relative sizes of the partitions according to QP for the two diverse reference video clips used later in Section 5. For the purposes of assessing the impact of the QP, Variable Bit-Rate (VBR) was encoded in this Section. In VBR video the QP remains constant in order to preserve video quality. However, most broadcasters prefer Constant Bit-Rate (CBR) video as it allows transmission bandwidth and storage to be predicted. For that reason, CBR video is tested in Section 5, though the QP value may vary a little in order to maintain a constant bit-rate.

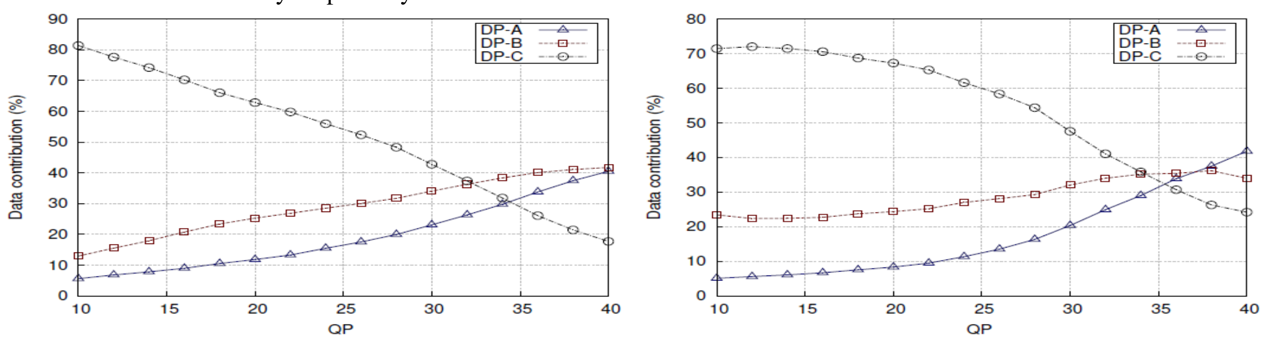
B. Adaptive channel coding

Rate adaptation of application-layer FEC, i.e. rateless Raptor code [16], is performed in order to match wireless channel conditions. The code was applied at the level of bytes within a packet in the interests of reduced latency rather than at the usual packet level. The probability of channel byte loss (BL) serves to predict the amount of redundant data to be added to the payload. In an implementation, BL is found through measurement of channel conditions, which is anyway mandatory in a WiMAX mobile station. If the original packet length is L, then the redundant data is given simply by

$$R = L \times BL + (L \times BL^2) + (L \times BL^3) \ldots \qquad (1)$$
$$= L/(1-BL) - L,$$

which is arrived at by adding successively smaller additions of redundant data, based on taking the previous amount of redundant data multiplied by BL. The following statistical model models the Raptor code reconstruction properties with high reliability [17] for k > 200

$$P_f(m,k) \begin{cases} = 1 & \text{if } m < k \\ = 0.85 \times 0.567^{m-k} & \text{if } m \geq k \end{cases} \qquad (2)$$



(a) (b)

Figure 2. Percentage contributions of data partitions A, B and C over a range of QPs for (a) *Paris* and (b) *Stefan*.

where $P_f(m,k)$ is the decode failure probability of the code with k source symbols, if m symbols have been successfully received.

Packets with ‘piggybacked’ repair data are also sent. Suppose a packet cannot be decoded, despite the provision of redundant data. It is implied from (2) that if less than k symbols (bytes) in the payload are successfully received then a further k - m + e redundant bytes can be sent to reduce the risk of failure. This reduced risk arises because of the exponential decay of the risk that is evident from equation (3) and which gives rise to Raptor code’s low error probability floor. In practice, e = 4, reducing the probability of failure to decode to 8.7%. Only one retransmission over a WiMAX link is allowed to avoid further increasing latency.

C. Packet scheduling method

Consider Fig. 3 in which a single video frame has been assigned equal bit length slices. The geometric space taken up by any slice is dependent on the coding complexity of the content. Shaded MBs in Fig. 3 intra-coded macroblocks (MBs), i.e. inserted as a result of RIMR. Each slice was further partitioned in source coding space into up to three partitions. It is possible that partition-B may occasionally be absent (the top slice in Fig. 3) if no RIMR MBs are allocated to a slice and if no naturally intra-coded MBs are assigned to the slice by the encoder. (Naturally encoded intra MBs are inserted if an encoder can find no matching MB in a reference frame or as a way of improving the quality.) Similarly, it is possible that if (say) an extremely high bit rate was allocated to the stream, that partition C might not be present. However, the simple packet scheduling scheme is independent of source coding allocations.

Fig. 4 demonstrates the results of two alternative packet scheduling regimes. A frame interval is shown, which is 1/30 s at 30 frame/s. In the default case, scheduling is at equal intervals in time. We are aware that packet scheduling may never follow this ideal regime in a processor even with a real-time operating system present. However, the regime acts as a point of comparison. In the simple scheduling scheme proposed, packets are allocated a scheduling point according to their relative size within a frame’s (or multiple frames’) compressed size. Thus, for any one packet indexed as j with length l^j, its scheduling time allocation is:

$$t^{j}_{alloc} = \frac{l^{j}}{\sum_{i=1}^{n} l^{i}} f \qquad (1)$$

where f is the fixed frame interval, and there are n packets in that frame. The denominator of (1) sums the lengths of the packets within frame j and allocates a time interval relative to its size (numerator of (1)) relative to the total length of the packets within frame j.

The proposed scheduling method obviously preserves the original average bit-rate, though there is a one frame video frame latency, while the frame’s packet sizes are assessed and the scheduling takes place. In [18], it is pointed out that IEEE 802.21 Media Independent Handover (MIH) services (IEEE 802.21 WG, 2008) provides a general framework for cross-layer signaling that can be used to achieve the scheduling. In IEEE 802.21, a layer 2.5 is inserted between the level 2 link layer and the level 3 network layer. Upper-layer services, known as MIH users or MIHU communicate through this middleware to the lower layer protocols. For mobile WiMAX and later versions of WiMAX, another WiMAX-specific set of standardized communication primitives exists as IEEE 802.16g.

4. SIMULATION MODEL

To model the video communication over WiMAX, the well-known ns-2 simulator was augmented with a module from the Chang Gung University, Taiwan [19] that has proved an effective way of modeling IEEE 802.16e’s behavior. In the evaluation, transmission over WiMAX was carefully modeled. The IEEE 802.16e TDD frame length was set to 5 ms, as only this value is supported in the

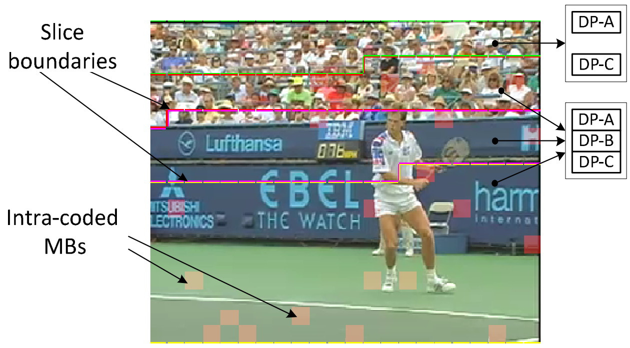


Figure 3. Example video frame from *Stefan* split into equal bit length slices, showing each slice partitioned into three partitions (A, B, C). Notice that it is possible in some circumstances for partitions B (or C) to be missing.

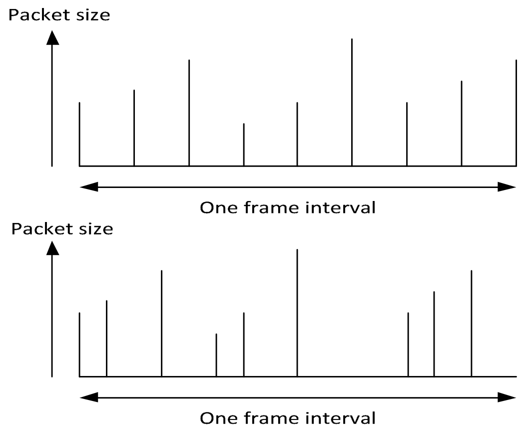


Figure 4. Example of packet scheduling regimes: (top) Equally-spaced in time, (bottom) Packet ordering preserved but scheduling time allocated according to packet size.

WiMAX Forum simplification of the standard. The raw downlink data rate of 10.67 Mbps results from the use of one of the mandatory WiMAX coding modes [10] for a TDD downlink/uplink sub-frame ratio of 3:1. The WiMAX BS was assigned more bandwidth capacity than the uplink to allow the BS to respond to multiple MSs. All buffers were set to 50 packets and all packets were directed to WiMAX's real-time Polling System (rtPS) class of service.

Apart from video streaming to a WiMAX MS, three other MSs received continuous CBR data, Fig. 5, with two receiving at 1 kbps, and one at 840 kbps. Packet size was set at 1 kB, the WiMAX maximum transport unit. It is these streams that inject cross-traffic into the scenario modeled and provide any congestion experienced by the video stream in the experiment reported in Section 5.

To model adverse channel conditions across the WiMAX link 'bursty' errors were modeled by a Gilbert-Elliott two-state hidden Markov model [3]. In consequence, around 20% of packets required retransmission of repair data. However, this contributes to packet latency rather than packet loss from congestion. Packet loss itself was at the level of 2% for the CBR streams described in Section 5. However, as the demonstration of Section 5 shows, this is enough to considerably reduce the video quality from a good level of over 30 dB, using the approximate mapping from PSNR to subjective scoring in [20].

5. FINDINGS

Figs. 6 and 7 show the packet size distribution as a result of coding two test video clips, *Paris* and *Stefan*, with Common Intermediate Format (CIF) (352×288 pixels/frame) respectively at 30 frame/s. The sequences were coded with a CBR target of 500 kbps. 5.6% intra-coded MBs were randomly added to each frame, equivalent to one row of MBs in CIF. (In the JM reference software of H.264/AVC used, this results in a gradual decoding refresh within 18 frames, because there is no replication of previously allocated random MB positions in successive frames.) From Fig. 6 for *Paris*, about 63% of packets are moderately sized, with 11% relatively large. The remaining 26% are small. The average absolute length of motion vectors in the Stefan sequence is 5.5, while for *Paris* it is only 2.2, implying that there is much more motion activity in *Stefan*. Partially as a result, the more active *Stefan* sequence has fewer smaller packets than *Paris* at this bitrate.

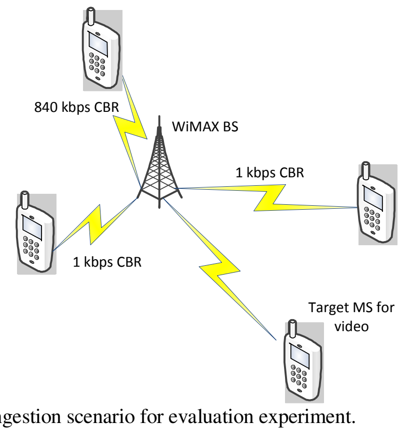


Figure 5. Congestion scenario for evaluation experiment.

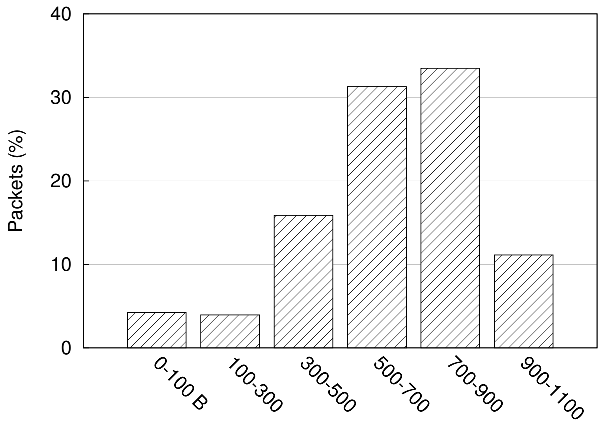


Figure 6. Packet size (bytes) distribution for *Paris* CIF @ 30 Hz, coded at a CBR of 500 kbps, with 5.6 % RIMR.

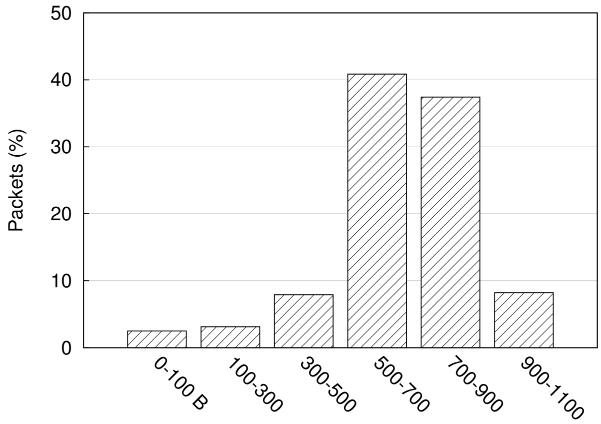


Figure 7. Packet size (bytes) distribution for *Stefan* CIF @ 30 Hz, coded at a CBR of 500 kbps, with 5.6 % RIMR.

Figs. 8 and 9 present the result of streaming *Paris* and *Stefan* respectively across the WiMAX link, before and prior to turning on the congesting sources mentioned at about frame number 125. From the Figures it is apparent that there is a content-dependent effect, as the more active sequence, *Stefan*, does not benefit from the scheme as much as *Paris* does. The reason for this is apparent from looking back to the packet size histograms of Figs. 6 and 7. As *Stefan's* packet sizes are predominantly in the range 500–900 B, they are more likely to pose a risk of overflow at the WiMAX BS output buffer. However, for both types of content there is a positive gain from turning on the packet scheduling and in the case of *Paris* there is a very definite gain in video quality, which remains stable.

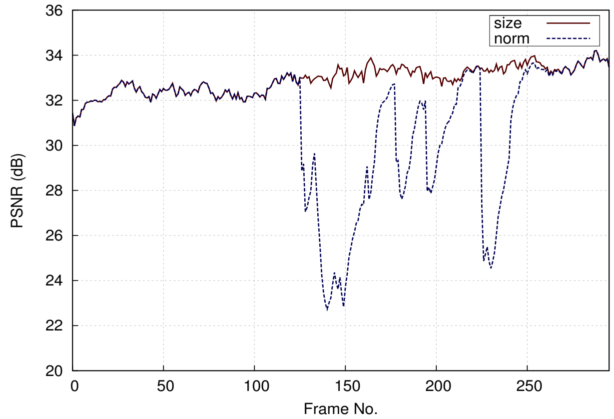


Figure 8. Objective video quality for *Paris* when using both scheduling approaches, by 'size' and the equi-spaced 'norm'.

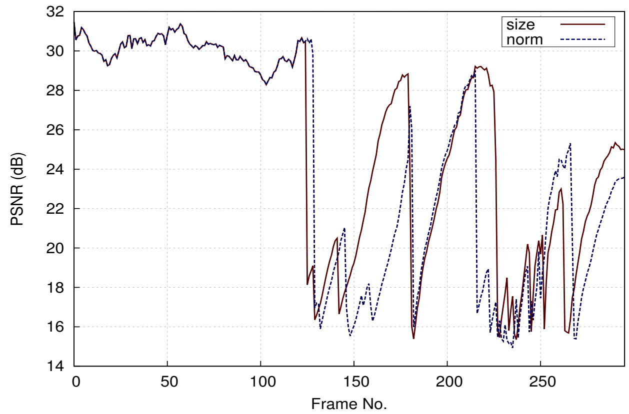


Figure 9. Objective video quality for Stefan when using both scheduling approaches, by 'size' and the equi-spaced 'norm'.

6. CONCLUSION

The proposed sized-based scheduling solution, specialized to data-partitioned video streams, will certainly benefit the types of content favored by mobile viewers according to quality-of-experience studies, which are studio scenes with limited motion activity. Active sports clips, especially those with small balls, are less attractive but may become more so as the move to higher resolutions on mobile devices continues (e.g. full VGA format (640 × 480 pixels/frame at 30 Hz) for streaming in Apple's FaceTime). However, in that case the problem of packet scheduling regimes may become more acute. Future work will require a robust investigation of appropriate packet scheduling for congested broadband links for a range of content genres.

REFERENCES

[1] O. Oyman, J. Foerster, Y.-j. Tcha and S.-C. Lee, "Towards enhanced mobile video services over WiMAX and LTE," *IEEE Commun. Mag.,* vol. 48, no. 8, pp. 68–76, 2010.

[2] T. Stockhammer, and M. Bystrom, "H.264/AVC data partitioning for mobile video communication," in *IEEE Int'l Conf. on Image Processing*, pp. 545-548, 2004.

[3] L. Al-Jobouri, M. Fleury, and M. Ghanbari, "Protecting H.264/AVC data-partitioned video streams over broadband WiMAX," *Advances in Multimedia*, vol. 2012, Article ID 129517, 11 pages, 2012.

[4] M. Fleury, S. Moiron, and M. Ghanbari, "Innovations in video error resilience and concealment," *Recent Patents on Signal Processing*, vol. 1, no. 2, pp. 1-11, 2011.

[5] M. Ghanbari, *Standard codecs: Image compression to advanced video coding*, IEE, London, UK, 2003.

[6] Y. Li, A. Markopoulou, J. Apostolopoulos, and N. Bambos, "Content-aware playout and packet scheduling for video streaming over wireless links," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 885-895, 2008.

[7] D. Tian, X. Li, G. Al-Regib, B. Altunbasak, and J.R. Jackson, "Optimal packet scheduling for wireless video streaming with error-prone feedback," in *IEEE Wireless Common. and Networking Conf.,* pp. 1287-1292, 2004.

[8] S.H. Kang, and A. Zakhor, "Packet scheduling algorithm for wireless video streaming," in *Int'l Packet Video Workshop,* 2002.

[9] E. Jammeh, M. Fleury, and M. Ghanbari, "Smoothing transcoded MPEG-1 video streams for Internet transmission," in *IEE Proc.: Vision, Image & Signal Processing*, vol. 151, no. 4, pp. 298-306, 2004.

[10] J.G. Andrews, A. Ghaosh, & R. Muhamed, *Fundamentals of WiMAX: Understanding broadband wireless networking*, Prentice Hall, Upper Saddle River, NJ, 2007.

[11] C.-M. Chen, C.-W. Lin, and Y.-C. Chen, "Packet scheduling for video streaming over wireless with content-aware packet retry limit," in *IEEE 8th Workshop on Multimedia Signal Processing,* pp. 409-414, 2006.

[12] G. Liebl, M. Kalman, and B. Girod, "Deadline-aware scheduling for wireless video streaming," in *IEEE Int'l Conf. on Multimedia and Expo*., 2005.

[13] F. Capozzi, G. Piro, L.A. Grieco et al., "Downlink packet scheduling in LTE cellular networks: Key design issues and a survey," IEEE Communs. Surveys & Tutorials, vol. 15, no. 2, pp. 678-700, 2013.

[14] G. Piro, L.A. Grieco, G. Boggia, R. Fortunat, and P. Camarda, "Two-level downlink scheduling for real-time multimedia services in LTE networks," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 1052-1065, 2011.

[15] Y. Dhondt, S. Mys, K. Vermeirsch, and R. Van de Walle, "Constrained Inter Prediction: Removing dependencies between different data partitions," in *Advanced Concepts for Intelligent Visual Systems*, pp. 720-731, 2007.

[16] A. Shokorallahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551-2567, 2006.

[17] M. Luby, T. Gasiba, T. Stockhammer, and M. Watson, "Reliable multimedia download delivery in cellular broadcast networks," *IEEE Trans. Broadcast*., vol. 53, no. 1, pp. 235-246, 2007.

[18] P. Neves, S. Sargneto, K. Pentikousis, and F. Fontes, "WiMAX cross-layer system for next generation heterogeneous environments," in U.D.D. Dalal & Y.P. Kosta (eds.), *WiMAX, New Developments*, In-Tech, Vukovar, Croatia, pp. 357-376, 2009.

[19] F.C.D. Tsai et al. "The design and implementation of WiMAX module for ns-2 simulator," in *Workshop on ns2: the IP network simulator*, article no. 5, 2006.

[20] J. Klaue, B. Rathke, and A. Wolisz, "Evalvid: A framework for video transmission and quality evaluation," in *13th Int'l Conf. on Modeling Techniques and Tools for Computer Performance*, pp. 255-272, 2003.