Packet dropping for widely varying bit reduction rates using a network-based packet loss visibility model

Ting-Lan Lin[1], Jihyun Shin and Pamela Cosman
Department of Electrical and Computer Engineering,
9500 Gilman Drive, University of California, San Diego, La Jolla, CA - 92093-0407
Email: {tinglan,pcosman}@ece.ucsd.edu

Abstract

We propose a packet dropping algorithm for various packet loss rates. A network-based packet loss visibility model is used to evaluate the visual importance of each H.264 packet inside the network. During network congestion, based on the estimated loss visibility of each packet, we drop the least visible frames and/or the least visible packets until the required bit reduction rate is achieved. Based on a computable perceptually-based metric, our algorithm performs better than an existing approach (dropping B packets or frames).

1 Introduction and Background

In video transmission through a network, packet losses occur for different reasons. An intermediate router can drop packets because the incoming data rate is so high that the buffer overflows. With IPTV, a subscriber may want to watch a video in high resolution, but his access bandwidth is less than required. In this situation, a router should drop enough percentage of data to meet the access capabilities of the subscriber. The packet dropping policy in the router should be intelligent to minimize the video quality damage observed by the end user. The packet dropping rates required at the router can vary by a large amount.

In past work, dropping decisions made by an intelligent router depend on MSE induced by a packet loss. Especially in [1], an intermediate router with an optimization algorithm drops packets in a congested network from different streams to minimize the sum of cumulative MSE. The packets must have embedded information about the associated induced MSE. Also, the optimization process is too complex for most current routers. Furthermore, MSE does not correlate well with human perception [2]. The works [3, 4] are for No-Reference network monitoring, which can compute estimated video quality for a given packet loss pattern using only video bitstream information. However, they give an overall quality score for the sequence and do not tell us how to best drop packets to minimize the video quality degradation during network congestion.

In our prior work [5, 6], packet dropping methods based on perceptual video quality are discussed. The visual importance of each packet is evaluated in the encoder

[1]This work was supported by Futurewei Technologies, Inc. and by the Center for Wireless Communications at UCSD.

1068-0314/10 $26.00 © 2010 IEEE
DOI 10.1109/DCC.2010.47

by an *encoder-based* packet loss visibility model. Every piece of information available to the encoder can be used. Before the packet is sent to the network, a single bit of priority score is added to the header based on the estimated packet loss visibility. The router can then drop packets of lower priority during congestion. In [5, 6], we showed that the dropping policy that uses visibility-based packet prioritization performs well compared to DropTail, a widely-implemented method, and compared to a prioritization method that is based on the induced mean square error if that packet is lost [1]. In [5, 6], the dropping policy was only tested at low to moderate packet loss rates up to 20%. As we will show in this paper, that approach of dropping on a packet basis does not do well at high dropping rates. Another limitation of [5, 6] is that the priority score needs to be determined at the encoder and added as one bit to the packet header.

In the proposed work, we do not assume packets coming into the router are embedded with a visual priority; for each packet, the visual importance is obtained by the *network-based model* described in [7] which requires information only within one packet and no reference frame information. This is desirable because packets may be out of order or because there may be multiple streams and the network node cannot afford to decode and reconstruct them. The parameter extraction process can be made very efficient since it does not involve motion compensation (requiring reference frame), deblocking filter and frame reconstruction. Second, we devise a packet dropping method for widely varying packet loss rates including high rates. The algorithm drops the least visible *frames*, which incurs fewer blocky artifacts compared to dropping on a *packet* basis. This method is shown to be better than [5, 6] and a method used by industry which drops B packets, for different levels of packet loss rate.

This paper is organized as follows. In Section 2, the subjective tests for building the network-based packet loss visibility model are described. We also discuss self-contained factors that relate to packet loss visibility, and the models based on these factors. Section 3 discusses the proposed multiple packet loss algorithm using measurements obtained by the network-based packet loss visibility model. Simulation results are in Section 4, and Section 5 concludes the paper.

2 Network-based packet loss visibility model

In this section, we introduce the network-based packet loss visibility model. Most of this section is taken from our prior work [7], where we reported on a subjective experiment on packet loss visibility. The video encoder is H.264 JM9.3. The decoder is FFMPEG [8] due to its high efficiency and wide use in industry. The encoding settings are as follows: SDTV resolution (720×480), bitrate 2.5 Mbps, H.264 Main Profile Level 3, Viewing distance= $6 \times$ picture height, frame rate 30 fps, GOP structure = IBBPBBPBBPBBPBB 15/3. These settings adhere to ITU and DSL Forum Recommendations [9, 10]. Each Network Abstraction Layer (NAL) packet contains a horizontal row of macroblocks (16×16 pixels) in a frame.

When viewers see a visible artifact or a glitch, they respond by pressing the space

bar. To allow observers enough time to respond to each individual loss, only one packet loss occurs for every 4-sec non-overlapping interval. The loss occurs in the first 3 seconds of the interval, and the fourth second allows any error propagation to terminate. The losses are divided equally among I frames, P frames and B frames. Each loss is watched by 10 people. The ground truth packet loss visibility for a specific packet can be obtained as the number of people who see the loss artifact divided by 10. We obtain ground truth packet loss visibility for 900 losses from the experiment.

2.1 Features available within one packet for prediction

In this section, we first introduce candidate factors associated with a packet. Next, we build models using these parameters to predict, for each packet, the packet loss visibility results of our subjective experiment.

Content independent factors depend on, for example, the spatial location or frame type of the loss, but do not depend on the actual video content at the location of the loss. **Height** is the spatial location where the loss occurs; the top slice in a frame has Height=1, and the bottom slice in a frame has Height=N, where N is the number of packets in a frame. **DevFromCenter** = abs(Height-floor(N/2)) indicates how far away the loss occurs from the center (in the vertical direction) of the frame. **TMDR** is the maximum number of frames to which the error from this packet loss can propagate. TMDR=1 for non-reference frames. For reference frames, TMDR depends on the distance to the next I frame.

Content dependent factors depend on the actual video content at the location of the loss. The ones we use all involve taking a mean, maximum, or variance computed over all macroblocks in the packet. **MeanRSENGY** is the mean residual energy after motion compensation. **MaxRSENGY** denotes the maximal residual energy after motion compensation. Following the way these factors were used in [5], we used the above two terms after logarithm because they were shown to be more correlated with packet loss visibility (we add 10^{-7} before taking the log to avoid a log of zero problem). **MeanMotX** and **MeanMotY** are the mean motion vectors in the x and y directions. **MaxMotX** and **MaxMotY** are the maximal motion vectors in the x and y directions. **VarMotX** and **VarMotY** are the variances of the motion vectors in the x and y directions. **MotM** is $\sqrt{MeanMotX^2 + MeanMotY^2}$. To compute the factors related to phase of motion vectors, we only consider macroblocks with non-zero motion, for which the phase is well defined. **MeanMotA** is the mean phase. **MaxMotA** is the maximal phase. **MaxInterparts** is the maximal number of inter macroblock partitions in the packet.

In addition to these content independent and content dependent factors, we also consider the interactions between factors in one category and factors in the other, as well as between factors within the content independent category.

The motion information mentioned above is estimated by the network node where reference frames are not available. In some cases, the "true" values for those quantities require the reference frames. For example, the "direct" mode of coding a macroblock assumes that an object is moving with constant speed, so the motion vector for the current MB is copied from the previous co-located MB. Within a packet, we do not

have any information on the previous co-located macroblock. We instead compute the motion vector by averaging those of spatial neighboring blocks within the same slice. This way, the model is fully self-contained at the packet level, and can be implemented at a network node.

2.2 Building the visibility model

We choose a generalized linear model (GLM) with the logit function as link function, since it can predict a probability parameter in a binomial distribution. We want to know the probability p that a packet loss artifact will be observed when the packet is lost. A GLM with a logit function for the binomial distribution has the form

$$log(\frac{p}{1-p}) = \gamma + \sum_{j=1}^{P} x_j \beta_j \quad (1)$$

where $\beta_1, \beta_2, ..., \beta_P$ are the coefficients of the P factors considered for prediction, and γ is the constant term. Often the parameters of the GLM are estimated such that the resulting model has the least deviance (the deviance is a generalization of the residual sum of squares). This treats data points equally, no matter how far they are from the regression line. However, outliers may distort the results. To give unequal treatment to data points, we minimize the M-estimator [11]; data points farther from the regression line have smaller weights, and contribute less to the final modeling result. We chose the "Fair" function as the M-estimator function, shown in Figure 1. The M-estimator is computed as the sum of the weighted residual squares, where the weight of each data point is computed by the residuals in the previous iteration. The model developing procedure uses 4-fold cross validation to prevent the model overfitting the data, so an average M-estimator is produced for a set of factors. The factor which most reduces the average M-estimator goes next into the model. This procedure repeats until there is no improvement in the average M-estimator by including an additional factor. The best factors chosen for the SDTV model and their corresponding coefficients are listed in Table 1.

3 Multiple packet loss algorithm

In our prior work [5, 6], the packet dropping policy used the fact that each incoming packet to the router has a 1-bit prioritization bit to signal whether this packet is estimated to be of low or high packet loss visibility. The loss visibility estimation is done at the encoder. In this paper, we assume no prioritization bit and use the network-based model to estimate the visual importance of each incoming packet. This model predicts the packet loss visibility of each packet based only on the information within one packet (NAL in H.264). Also the model does not need information from the pixel domain, and therefore the tasks of motion compensation, deblocking filtering and frame reconstruction are not necessary. Therefore the complexity to obtain the factors for prediction is much lower than a full decoder and is more realistic for implementation in a router.

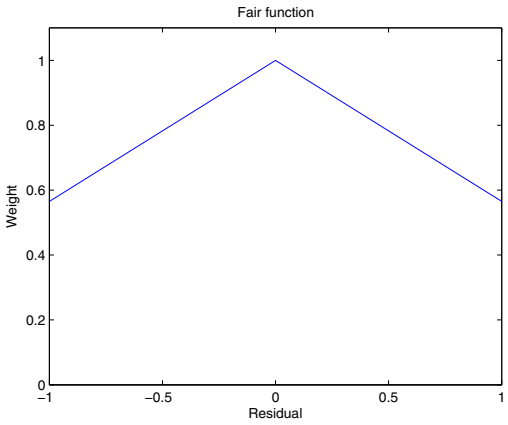


Figure 1: The Fair function against the residual.

We define bit reduction rate (BRR) as the percentage of bits that need to be dropped of the buffered packets to alleviate the congestion. Given the packet loss visibility scores, during network congestion, a router can straightforwardly drop packets with least estimated visibility until the required bit reduction rate is achieved and the congestion is relieved. We denote this method as **Vis-Pkt**. An intelligent dropping method that is implemented in a video-aware digital subscriber line access multiplexer (DSLAM) is discussed in [12]. It inspects the nal_ref_idc (NRI) bit in every NAL unit header. Packets which do not serve as reference pictures can be dropped during network congestion. We denote this method as **B-Pkt**.

In [13], subjective test results showed that in general, frame freezes are less noticeable than blockiness resulting from random packet loss. That is, when one portion of a frame is lost, and another part is not lost, the concealment may cause a spatial misalignment between objects/background in the intact portion of the frame and objects/background in the lost and concealed portion. This spatial misalignment in a frame often draws more attention from viewers than does a frame freeze, which has no spatial misalignment problem. Frame freeze can be produced by whole frame loss when the decoder conceals the lost frame by "frame copy". This motivated us to consider dropping packets on a *frame* basis, instead of dropping on a *packet* basis, as done in Vis-Pkt and B-Pkt. We design an algorithm, denoted as **Vis-Frame-Pkt**. The summed visibility over all packets in a frame is calculated for each frame. We drop the N least visible frames. N is chosen such that the total number of bits in the packets comprising these frames is under the total number of bits of BRR, but dropping the (N+1) least visible frames would put the total over BRR. Then we drop packets on a packet basis to reach the required number of bits of BRR. We design a similar algorithm, denoted as **B-Frame-Pkt**, which randomly drops B packets on a frame basis, and when dropping the next B frame would mean dropping more than BRR bits, it switches to dropping on a packet basis to reach the BRR bits.

Since the size of a packet is much less than that of a frame, these methods that switch to dropping on a packet basis try to meet very closely the goal of number of

Order	Factors	Coefficients
α	1	-2.6407
1	TMDR×MaxMotA	-4.7591e-3
2	DevFromCenter×MaxMotA	2.2996e-2
3	Height×MeanMotA	-8.8462e-4
4	TMDR×log(MeanRSENGY $+10^{-7}$)	3.5954e-3
5	TMDR×MeanMotY	-1.6431e-2
6	DevFromCenter×TMDR	-1.0164e-2
7	DevFromCenter×MeanMotY	5.3172e-3
8	TMDR	2.3680e-1
9	TMDR×MaxInterparts	-5.6283e-3
10	TMDR×MotM	4.9349e-3
11	Height×DevFromCenter	-3.1830e-3
12	Height×MaxInterparts	2.1661e-3
13	TMDR×VarMotY	5.1232e-4

Table 1: Table of factors in the order of importance for SDTV GLM model. The × symbol means interaction.

bits to be dropped. However, blockiness is introduced by this approach since some packets are dropped on a packet basis rather than on a frame basis. Another approach is to drop the (N+1) least visible frames all the way until the goal of number of bits to drop has been reached, even though this means that, in general, the goal will be exceeded by perhaps a large number of bits due to the granularity of dropping whole frames. We denote this method **Vis-Frame**. And the counterpart of this in dropping B packets is denoted **B-Frame**, which randomly drops B frames until the requirement is reached.

4 Experimental results

In this section, we compare the six methods for different videos and different levels of BRR. All the methods which relate to dropping B packets are implemented by randomly dropping B packets/frames in the buffer, and when running out of B packets/frames to be dropped, P packets/frames are dropped randomly. The performance is evaluated by averaging 50 random realizations.

The video encoder is H.264 JM9.3. The decoder used is FFMPEG [8]. The resolution is SDTV. The tested videos are encoded at 2.5Mbps, 30 fps using Main profile Level 3. The GOP (Group of Pictures) structure is IBBP (15 frames). Each NAL packet comprises one horizontal row of macroblocks. Therefore we have 30 packets in a frame.

For error concealment for non-whole-frame losses, the FFMPEG decoder begins by making a guess, for each lost macroblock, of whether it is more likely to have been intra coded or inter coded. For example, in P and B frames, the algorithm looks at

the coding mode of some or all of the macroblocks that are not lost, and if more than half of those have a coding mode which is intra, then the algorithm will guess that all the lost macroblocks in the frame were coded intra. Once the guess has been made of coding mode for each lost macroblock, the algorithm uses two different approaches. For the macroblocks which are guessed to be intra coded, for each 8×8 block in each MB, ffmpeg does a process called FillDC, which looks at the four directions surrounding the block (top, bottom, left, right) to find uncorrupted blocks. It then finds the pixel average of each uncorrupted neighboring block. Finally, it takes a weighted average (weighted according to distance) of the uncorrupted averaged blocks and the result is the block that is used for concealment. For the macroblocks which are guessed to be inter coded, the algorithm estimates the forward and backward motion vectors by scaling, in display distance, the colocated future and past motion vectors in the buffer. The obtained motion vectors are used to perform bi-directional motion estimation to conceal the lost MB. The whole frame losses are concealed by temporal interpolation of closest past and future frames.

We perform each dropping algorithm in a GOP, and the BRR is the percentage of bits to be dropped for this GOP. After the dropping policy is performed for a GOP, the FFMPEG decoding and error concealment are run, and then the Video Quality Metric (VQM) [14] is calculated to obtain the video quality score for this lossy GOP. VQM is a full-reference metric developed by the National Telecommunication and Information Administration. It ranges from 0 (excellent quality) to 1 (poorest possible quality). VQM scores have been shown to be better correlated with human perception than other full reference video quality metrics [15].

Two videos are tested in the simulation: *golf* is of slow movement, and *soccer* is of high motion and fast panning. The simulated BRRs are 0.5%, 5%, 10% and 20%. Note that BRR can be very different from packet loss rate (PLR). For example, 20% BRR can result in 50% PLR if the dropping algorithm drops B packets, which have much smaller sizes than I or P packets on average. Therefore, BRR ranging from 0.5% to 20% considers a very wide range of packet dropping levels. The two videos are subsets of the nine videos used for the subjective experiments, however, the losses injected for this section are very different from those in the subjective experiments because the latter used isolated slice losses.

Figure 2 shows the VQM performance versus GOP index for the six dropping methods for BRR = 0.5%, 5%, 10% and 20% for the *golf* video. Figure 3 shows VQM score averaged over GOPs versus BRR for the six packet dropping policies. From the figures, we observe the general trend that the -Frame-Pkt method is better than the -Pkt method for both the Vis and B methods. This means that dropping packets on a frame basis helps the video quality. We also observe that the -Frame method is better than the -Frame-Pkt method in general. This means even though -Frame drops more bits than BRR needs, the perceptual quality is improved due to having no spatial misalignment (blockiness) problem.

Note that the y-axis scale changes steadily as one looks at Figure 2(a), then 2(b), 2(c) and 2(d). This is because higher BRR makes the quality generally worse, so VQM scores become higher and have more variance. Note also that the y-axis scale for Figure 3(a) is not the same as for Figure 3(b). Because soccer is a high motion

video, and golf is a low motion video, losses are less concealable for soccer. Therefore, for a given BRR, the scores for soccer are worse (higher) than they are for golf. So, the average improvement in VQM score for the best dropping approach compared to the worst one is very much larger for soccer than for golf. While the average improvement in VQM score (provided by the best dropping method) shown in Figure 3(a) for the golf sequence is very small, and so might be considered perceptually not noticeable, if one looks at the VQM scores versus GOP index shown in Figure 2, many of the individual GOPs have substantial VQM improvements, which would be perceptually noticeable.

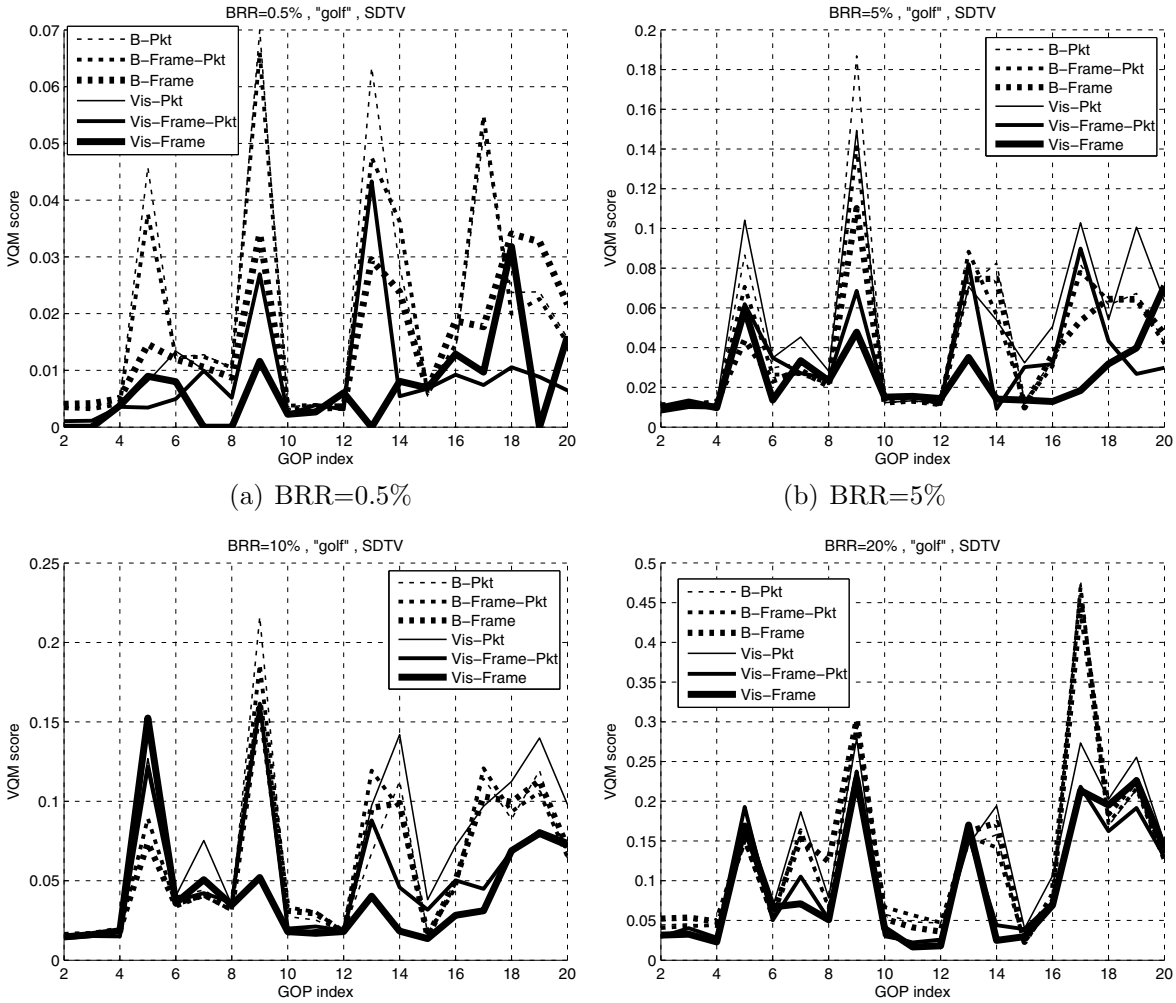
For comparisons between B- methods and Vis- methods, when the BRR is very low (0.5%), Vis-Pkt is better than B-Pkt. However, when the BRR increases, Vis-Pkt is not better than B-Pkt. It may be that Vis-Pkt does better at low dropping rates because the visibility model is developed from videos with isolated losses where the evaluated packets have their intact reference frames, which is not the case for much higher packet loss rate. However when we perform whole frame drop (Vis-Frame-Pkt or Vis-Frame methods), we can see from Figure 3 that for all BRRs, both Vis-Frame and Vis-Frame-Pkt are better than B-Pkt, B-Frame-Pkt, and B-Frame. Lastly, we can observe that Vis-Frame is always better than Vis-Frame-Pkt except at the lowest loss rates. This makes sense because for low loss rates, the fact that Vis-Frame exceeds the target BRR in order to maintain dropping of whole frames incurs a more severe penalty percentagewise in bits dropped.

5 Conclusions

We used a network-based packet loss visibility model to measure the visual importance of packets incoming to a router. The estimated visibility scores are then used by the router to perform intelligent packet dropping. For a very wide variety of bit reduction rates, the performance of the proposed algorithm outperforms both a visibility-based algorithm that drops packets on a packet basis, and an algorithm that drops B packets or B frames such as one currently implemented in a video-aware digital subscriber line access multiplexer. The contributions of this paper are (a) We showed that dropping whole frames and concealing by simple frame interpolation produces better quality video than dropping on a packet (slice) basis, (b) We showed that the visual advantage of dropping whole frames is sufficiently large that, except for low dropping rates, it pays to drop whole frames even when that means exceeding the target bit reduction rate for the GOP, (c) A simple visibility model that can be implemented inside the network provides a better basis for choosing frames to drop than just targeting B frames for dropping.

References

[1] J. Chakareski and P. Frossard. Rate-distortion optimized distributed packet scheduling of multiple video streams over shared communication resources. *IEEE Transactions on Multimedia*, 8:207 – 218, April 2006.



(a) BRR=0.5% (b) BRR=5%

(c) BRR=10% (d) BRR=20%

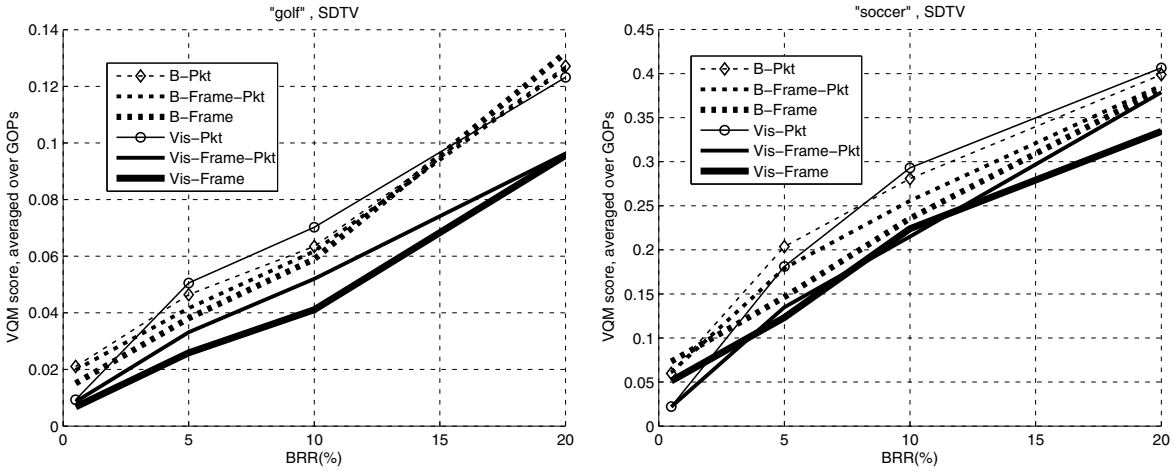
Figure 2: VQM performance vs. GOP index for the six packet dropping policies for *golf* for BRR = (a) 0.5% (b) 5% (c) 10% and (d) 20%. Lower VQM scores correspond to higher quality.

[2] B. Girod. *What's wrong with mean-squared error?* MIT Press, Cambridge, MA, USA, 1993.

[3] M. Naccari, M. Tagliasacchi, and S. Tubaro. No-Reference Video Quality Monitoring for H.264/AVC Coded Video. *IEEE Transactions on Multimedia*, 11(5):932 – 946, Aug. 2009.

[4] S. Tao, J. Apostolopoulos, and R. Guerin. Real-Time Monitoring of Video Quality in IP Networks. *IEEE/ACM Transactions on Networking*, 16(5):1052 – 1065, Oct. 2008.

[5] T.-L. Lin, Y. Zhi, S. Kanumuri, P. Cosman, and A. Reibman. Perceptual quality based packet dropping for generalized video GOP structures. *ICASSP*, 2009.



(a) *golf* (b) *soccer*

Figure 3: Average VQM score over GOPs vs. BRR for the six packet dropping policies for (a) *golf* and (b) *soccer*. Lower VQM scores correspond to higher quality.

[6] T.-L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. Cosman, and A. Reibman. A Versatile Model for Packet Loss Visibility and its Application to Packet Prioritization. *IEEE Transactions on Image Processing*, accepted for publication.

[7] T.-L. Lin and P. Cosman. Network-based Packet Loss Visibility Model for SDTV and HDTV for H.264 videos. *ICASSP*, 2010.

[8] The Official website of FFMPEG : http://ffmpeg.org/.

[9] ITU-R BT.710-4 Subjective Assessment Methods for Image Quality in High-Definition Television. Jan 1998.

[10] DSL Forum Technical Report TR-126: Triple-play Services Quality of Experience (QoE) Requirements. Dec 2006.

[11] W. J.J. Rey. *Introduction to Robust and Quasi-Robust Statistical Methods.* Springer, 1983.

[12] Alcatel-Lucent Technical Paper : Access Network Enhancements for the Delivery of Video Services. May 2005.

[13] N. Staelens, B. Vermeulen, S. Moens, J.-F. Macq, P. Lambert, R. Van de Walle, and P. Demeester. Assessing the influence of packet loss and frame freezes on the perceptual quality of full length movies. *International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM)*, 2009.

[14] VQM. http://www.its.bldrdoc.gov/n3/video/.

[15] M. H. Loke, E. P. Ong, W. Lin, Z. Lu, and S. Yao. Comparison of video quality metrics on multimedia videos. *IEEE ICIP*, October 2006.