

MobiStream: Error-resilient Video Streaming in Wireless WANs using Virtual Channels

Rajiv Chakravorty[†], Suman Banerjee[†] and Samrat Ganguly^{*}

[†]Department of Computer Sciences, University of Wisconsin-Madison, WI, USA. E-mail: {rajiv,suman}@cs.wisc.edu

^{*}NEC Laboratories America Inc., Princeton, NJ, USA. E-mail: samrat@nec-labs.com

Abstract—We present MobiStream—a video streaming system that exploits the perceptual value in video content and the characteristics of the link layer and physical layer channels to enable error-resilient video streaming over wireless wide-area networks (WWANs).

The key building block in MobiStream is the use of link-layer based, but application-controlled, *virtual channels* (ViCs) abstraction. Each virtual channel offers a level of reliability and statistical loss guarantees using ‘awareness’ of the characteristics of link-layer and physical layer channels. Video applications can dynamically instantiate new virtual channels, control their loss behavior, and/or flexibly switch video transmission across channels. MobiStream achieves fine-grained error-resilience by partitioning video frames into number of small, independently decodable, blocks of data (called ‘slices’) and assigns priority based on its perceptual (visual) usefulness. MobiStream augments a number of other enhancements for error-resilience: multiple description coding, perceptual slice-structured coding, low-delay inter-frame and intra-frame slice interleaving, dynamic unequal error protection, and priority-based video-data scheduling to enable error-resilient video streaming over wireless wide-area links.

MobiStream has been implemented and evaluated using loss distributions from tests over a commercial wide-area wireless (CDMA2000 EvDO) network. Results show that, even in stationary conditions, MobiStream, on average, improves video picture quality by at least 4 dB. We conclude that significant benefits to end-user experience can be obtained by deploying such a system.

I. INTRODUCTION

Advances in digital video coding techniques coupled with universal cellular network upgrades supporting broadband data-rates (e.g., CDMA2000 EvDO and UMTS) are driving considerable research into new video services and applications. However despite the promises of efficient video coding techniques, today’s video services have failed to take advantage of the advances in wireless. The fact that video services in many WWANs are limited to simple video-clips *downloads* as opposed to actual video streaming show the extent of difficulties and challenges that WWAN environments still pose.

There are several impediments facing efficient error-resilient video streaming over WWANs. Firstly, data transmission over a radio channel is prone to bit errors due to multi-path effects, shadowing, and interference. While link layer-based retransmissions and coding can improve performance to an extent, it still cannot avoid burst losses and results in high transmission delay and jitter. Secondly, there is an inherent tussle between voice and data users. Because available bandwidth is limited in a cell, each user is typically assigned a channel bandwidth that is a function of the signal strength and interference each user in

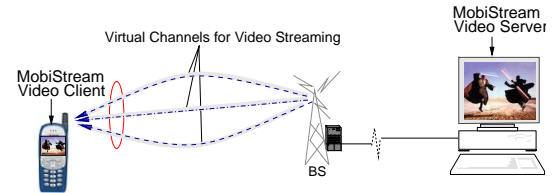


Fig. 1. Virtual Channels (ViCs) for Wireless Wide-area Networks.

that cell receives. If signal strength is too low or there is high interference in the channel then more processing gain (transmit power) or better channel coding (redundancy) is needed to protect the transmitted data. Thirdly, limited number of data channels are often shared by many users leading to high throughput variations. In some cases data transmissions can be completely impeded, e.g., during cell reselections/handoffs, resulting in transmission gaps ranging from a fraction of a second to several seconds. Such predicament in the available channels and bandwidth induces high delay jitter during data transfer. Finally, user mobility leads to varying signal strengths due to the physical radio propagation path loss and fading leading to different bandwidths being dynamically assigned to the end-user.

Because of these challenges video coding in conjunction with the commercial promise of broadband wide-area wireless technology, is attracting considerable research efforts particularly directed towards error-resilient video coding and transmission.

To this extent, we present MobiStream—a video streaming system that enables efficient and error-resilient video streaming over WWANs. The key building block in the MobiStream system is a link-layer based, but application-controlled, statistical loss-based *Virtual Channels* (ViCs) abstraction (fig. 1). From the perspective of (video) applications, ViC provides statistical loss guarantees to video traffic flowing between end points of a WWAN link in the face of varying channel conditions. Applications define and control the use of available bandwidth and loss allocations among each individual data flow within a ViCs. While ViCs may not be able to provide the spectrum of service guarantees for a range of other video-based applications, e.g., video telephony and digital video broadcasting, it can still provide adequate service guarantees necessary for efficient error-resilient video streaming. There are several advantages using the Virtual Channel (ViCs) abstraction in WWANs:

1) Smoothing burst losses: Burst packet losses can have severe negative impact on video streaming applications. Using

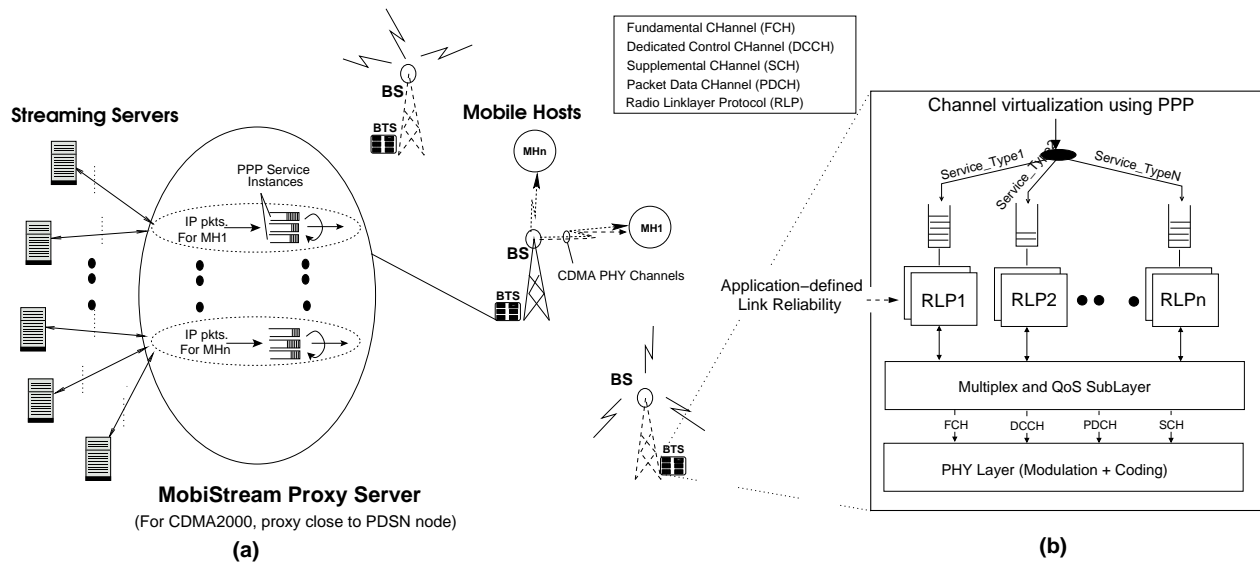


Fig. 2. Shows (a) MobiStream architecture for error-resilient video streaming over wireless wide-area links, and, (b) construction of virtual channels using point-to-point(PPP) protocol. MobiStream proxy server is located close to the cellular provider’s network (for CDMA2000 close to PDSN).

virtual channels MobiStream can reduce or even eliminate such burst loss effects leading to more efficient video streaming.

2) Statistical loss guarantees: Using link layer ‘awareness’ of loss distributions for a range of channel conditions, MobiStream can provide statistical loss guarantees for each virtual channel to serve part of its traffic.

4) Perceptual video data prioritization: Unlike voice-based applications, where voice samples can not be inherently prioritized, the nature of video content itself offers the possibility for very effective prioritization. MobiStream not only prioritizes video data streams over virtual channels, but also allows to express the *perceptual* importance of each data packet in a video stream, i.e., protect visually more important (video) data than less important ones.

3) Applications-defined link layer behavior: Video applications in MobiStream can dynamically instantiate new virtual channels, control their statistical loss behavior and/or flexibly switch video data transmission across such channels. This feature departs from the traditional observation of a wide-area wireless link as a single-hard-coded ‘fixed reliability’ logical data link. Instead, MobiStream views WWAN links as combination of multiple virtual channels each with an increasing order of statistical reliability.

In this work we present our practical experiences using the MobiStream system. We present its design and implementation, and perform an extensive evaluation. Using tests conducted over a commercial cellular CDMA2000 EvDO network, we show that MobiStream can provide statistical loss guarantees on the order of 0.12% of the total loss samples for a “best effort” reliable virtual channel and 0.53% for an unreliable virtual channel with bandwidth guarantees ranging from 64 Kbps to 256 Kbps.

Key Contributions

1) Our first contribution is the design, implementation and evaluation of MobiStream— a video streaming system that enables efficient and error-resilient video streaming over wireless wide-area networks. Mobistream has been evaluated using packet traces and loss distributions from stationary and drive tests taken over a commercial cellular (CDMA2000 EvDO) network. Our results demonstrate that MobiStream, on average, can improve video streaming (picture) quality in the presence of burst losses by at least 4 dB.

2) We introduce *virtual channels* as the building block for efficient video streaming in WWANs. Virtual channels depart than the traditional observation of WWAN link as a single hard-coded, fixed-reliability “best-effort” logical data link. Instead, it views a WWAN link as composed of multiple virtual channels. Each virtual channel is an independent logical entity that offers a level of link reliability with statistical loss guarantees. Virtual channels are flexible in the sense that they can be dynamically instantiated, their statistical loss behavior controlled and/or allow flexible switching in video data transmission across such channels. Video streaming applications particularly benefit from the flexibility offered in virtual channels by transmitting more useful data over more reliable channels and less important data over less reliable channels. However when the virtual channel is not reliable enough, it can still provide the flexibility to video applications using robust application-layer repair. MobiStream augments a number of techniques for error resilience using virtual channels: multiple description coding, fine-grained slice-structured video coding, low-delay slice interleaving, unequal error protection and smart video-data packet scheduling. We demonstrate the efficacy of virtual channels through these enhancements.

3) We exploit the perceptual value, i.e., visually more important data within the video data stream along with the characteristics of the WWAN link layer and channels to enable error-resilient video streaming. We argue (and show) that traditional metrics that measure video distortion like PSNR (Peak Signal to Noise Ratio) may not always capture the impact of losses on the visual perception of the streaming video content. To this extent, we present *perceptual slice-structured* video coding that assigns priority using region-of-interest based coding in video frames by partitioning them into number of small, independently decodable, blocks of data (called ‘slices’) to be then reconstructed with assigned priority based on its perceptual usefulness. Through real experiments, we demonstrate the effectiveness of perceptual slice-structured video coding.

Prior research work has investigated error-resilient video streaming in both wired and wireless networks. However our work differs from each of them in several ways. Section VI discusses related work.

Roadmap

Our paper is laid out as follows. The next section describes the ViCs abstraction. Section III motivates the case for video streaming over ViCs while Section IV proposes several techniques for error resilience using ViCs. Section V presents the implementation and evaluation of MobiStream. Section VI discusses related work and the last section concludes our paper.

II. VIRTUAL CHANNELS (ViC) ABSTRACTION FOR WIRELESS WIDE-AREA NETWORKS

Virtual Channels exploits two well-known principles in wireless video communications:

- 1) The first is that different parts of a video bit-stream consists of data with different importance, and hence needs to be protected via forward error correction (FEC) and automatic repeat request (ARQ) to *variable degrees*.
- 2) The second motivates from an important observation that, instead of link layer control wireless channels, let the applications take *partial control* of these channels using unequal FECs and/or retransmissions at the application layer itself.

Today’s WWAN networks however lack any mechanisms to support variable degrees of FEC protection or even retransmissions policies for the differently marked packets. For instance, the newly deployed CDMA2000 EvDO cellular system offers reasonably reliable transmissions over the radio link with a Radio Link Protocol (RLP) that provides, at best, “best effort” level of link-layer reliability set as network default.

Note that CDMA2000 standard however does specify data services to be flexibly defined and specified independently within the confines of the physical layer and the multiplex sub-layer interface. (see figure 2). The system has been designed to support multiple instances of the same service option commonly referred to as *service instances*. This option in CDMA2000 supports a maximum of six service instances per

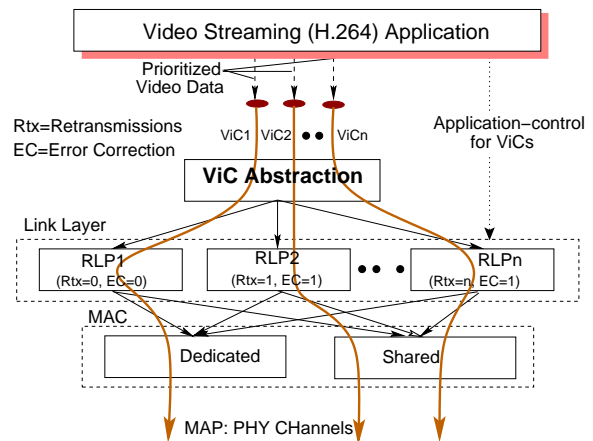


Fig. 3. Illustration of the Virtual Channels (ViC) Abstraction in WWANs.

mobile (MN), each of which can have a different associated radio link protocol (RLP) settings.

Despite these encouraging developments, applications have lacked to take advantage of the ways reliability can be exploited at the link-layer in these networks. Virtual Channels (ViCs) abstraction (see figure 3) aims to exploit this advantage offered by the link-layer (RLP) in CDMA2000 WWANs to allow video applications to flexibly change and control the levels of reliability available at the link-layer in these networks.

Video applications however still lack adequate control of the WWAN channels. These applications are rigidly tied to the “best-effort” levels of service guarantees based on the available link-layer FECs and cannot effectively prioritize transmission of the more important video data. Therefore, if video applications could take *partial control* of the channel, the best possible way to achieve this would be to use application layer FECs, i.e., by disabling link layer ARQ and error correction and letting the video applications take control.

Virtual Channels abstraction proposed in this paper not only benefits video applications by providing variable degrees of reliability, but it also empowers them to take partial control of the wireless channels by using application-layer FECs. Note however that the delays resulting from using FECs and retransmissions at the application layer may be too excessive for a full-scale applications layer channel control.

Virtual Channels and CDMA2000 Architecture. Virtual Channels and CDMA2000 QoS architecture can very well co-exist to the particular benefit of video streaming applications. In fact, Virtual Channels may even exploit many of features available from the CDMA2000 QoS architecture. For example, CDMA2000 QoS supports two subscriber modes— assured and non-assured mode. A mobile subscribed to an assured mode has the option of sending a set of QoS parameters— for example, the required air-link data rate, requested air-link frame error rate, acceptable air-link frame error rate. This set of QoS parameters in CDMA2000 is known to as a *QoS Blob*. Using this QoS information, the radio access bearer (RAB) service layer in CDMA2000, based on per-active session requirements from the BSC and the current RF conditions, decides the optimal QoS

multiplexing mechanism and the MAC sublayer multiplexing mechanism over the radio link. The eventual goal of the RAB service layer is to maximize the RF spectrum utilization, given the constraints of radio link quality and per-flow QoS requirements over-the-air.

Virtual Channels can similarly exploit use of the MAC layer in ways to efficiently multiplex traffic from multiple virtual channels (service instances) onto physical layer channels (the multiplex sub-layer).

Constructing Virtual Channels with PPP Protocol. Virtual Channels can be efficiently constructed using the PPP protocol. In the CDMA2000 architecture, IP packets received by the WWAN gateway (called PDSN) are sent over a GRE tunnel as frames to the BSC. At the BSC, data is converted to a byte stream which is then sent as RLP frames to the mobile node. Note that a single PPP (point-to-point) protocol session carries all data traffic in the form of PPP frames from the MN to the PDSN and from the PDSN to the MN.

There are two cases that arise in the way PPP is used.

Case 1: No service differentiation with Virtual Channels: When no service differentiation is required using Virtual Channels to and from a single user, then mapping of all the video data traffic onto a single RLP session is adequate. Thus, within the PPP frames, class based scheduling, for example, based on UDP/IP information, can be performed on the PPP frames. However once the PPP frames are converted to a byte stream and sent as RLP frames, these RLP frames are delivered to the MN in sequence. This is ensured by sequence numbers on the RLP frames.

Case 2: Data Prioritization using Virtual Channels: This is a special case of service differentiation using PPP and the procedure works as follows. A mobile node (MN) and a radio access network (RAN) can identify service specific instances with a service reference identifier (say sr_id1 to sr_idn). A single radio session is then maintained for all the connections (flows) associated with an MN and there is one PPP session per MN.

However a MN can establish multiple service instances for multiple such Virtual Channels within the network (RAN). To do this, the MN first establishes a primary PPP service instance before establishing secondary service instances. All service instances have to be of packet data, although different new service options (e.g. for VoIP) could also be defined. These new service options could also defer in the way RTP/UDP/IP headers are compressed, i.e., link-layer-assisted robust header compression (LLA-RoHC) used for transparent header compression and good enough header compression (GEHCO) for nontransparent header compression.

In this manner each RLP session in a service instance corresponds to a Virtual Channel that can have parameters such as the number of retransmissions applicable to all the RLP frames in that session. Thus, when retransmissions are not needed for certain portions of video data streamed over a Virtual Channel, it can be turned off just for that particular session. Other channels however can flexibly change the

number of retransmissions to benefit efficient streaming of the partitioned video content.

Mapping Virtual Channels onto PHY Layer Channels. Virtual Channel data in the form of link layer RLP data are multiplexed onto multiple PHY channels. An inherent advantage from this type of channels mapping is channel diversity.

As described earlier, CDMA2000 EvDO provides two types of logical channels in the current packet data implementations: Fundamental CHannel (FCH) and Supplemental CHannel (SCH). FCH maintains the physical layer connection and carries both signaling and a portion of packet data traffic, whereas SCH is allocated dynamically based on the traffic demand and carries high-speed packet data. The range of SCH data rate depends on the specified Radio Configuration (RC). There are 5 different radio configurations (RC1 through RC5) specified on the forward link (downlink). Most infrastructure and handset vendors implement RC3, which includes a 9.6 kb/s FCH. Therefore, the peak forward link data rate for a data user in CDMA2000 EvDO network can be as high as 2400 kb/s ($2SCH+1FCH+1DCCH$) at the physical layer.

The overall user throughput is bursty due to the dynamically assigned channel data rate. However the practical data rate allocated for a user at a given instant is determined by the network using vendor-specific algorithm. In a multi-data-user situation, the available resources is shared by all high-speed packet data users. The resource scheduler of the wireless network has control over when and what rate gets allocated to a virtual channel, i.e., rate assigned to a physical layer channel.

III. RESILIENCE TO BURST ERRORS IN VIDEO STREAMS

An inherent problem with any communications system is that information may be altered or lost during transmission due to channel noise. The effect of such information loss can have severe negative impact on the transport of compressed video streams because any damage to the compressed bit-stream may lead to objectionable visual distortion at the decoder.

To illustrate the visual artifacts caused by burst errors in a compressed video stream, in figure 4 we show a reconstructed video frame from the *akiyo* video sequence. The original sequence was encoded using the H.264 encoder [1] at an average rate of 256Kbps with a macroblock quantization parameter of 20¹. This is shown in the image in figure 4(a) (psnr of 44.6).

The image shown in figure 4(b) is a reconstructed frame (psnr of 34.1), where 5% packet loss leads to five damaged macroblocks in that particular frame. This causes visible discontinuity leading to perceptual degradation when the damaged block falls in a region of fast motion or in a region-of-interest within a video frame (as shown in a white box). This visible distortion degrades further when burst loss length is increased to 20% (leading to 20 damaged macroblocks) and shown in figure 4(c) (psnr of 23.7).

However when damaged macroblocks do not corrupt macroblocks that belong to a region-of-interest in a given frame, the overall perceptual distortion may not be as severe. This is evident in image shown in figure 4(d) (psnr of 24.1) where a

¹Quantization Parameter in H.264 has values between [1-51] range.

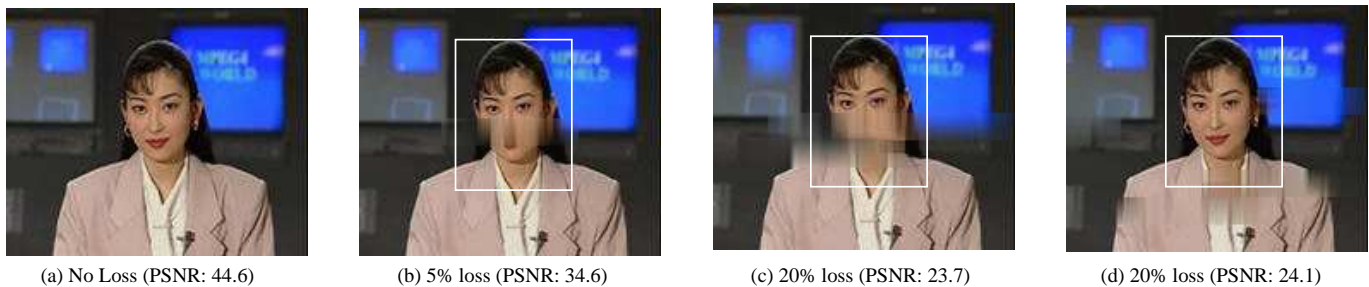


Fig. 4. Shows the impact of burst packet loss on video picture quality for the 3rd frame in *akiyo* video sequence. (The grey inner rectangle indicates the region-of-interest in this frame.)

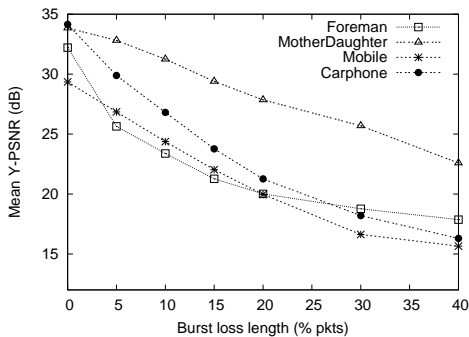


Fig. 5. Degradation in mean PSNR due to burst losses (in % of slice pkts) for different video sequences with decoder error concealment enabled.

burst loss of 20% (again leading to 20 lost macroblocks) shows significantly reduced *visual perceptual distortion* when compared to the image shown in figure 4(c). Note that burst loss in figure 4(c) and 4(d) as well as the mean degradation (in psnr) are nearly the same. However unlike the frame in figure 4(c) all the lost macroblocks in for the frame in figure 4(d) are concentrated outside the region-of-interest. Therefore, the reconstructed frame in this case looks far better even when it is compared to the 5% burst packet lost scenario depicted in frame of figure 4(b). This example clearly demonstrates the importance of perceptual coding in video frames.

Signal reconstruction and error-concealment techniques have been developed to alleviate the visible distortion in frames. These techniques strive to obtain a close approximation of the original signal or attempt to make the output signal at the decoder least objectionable to the human eye.

Using the latest H.264 implementation as an example [1] the process of error-concealment for a lost macroblock works as follows. If the lost macroblock was not intra-coded, then an estimate of its motion vector is computed by examining the motion vectors of its neighbors. The lost macroblock is then motion compensated using the estimated motion vector. If the lost macroblock was *intra-coded*, then its contents are spatially interpolated from adjacent macroblocks. If the adjacent intra macroblocks are lost too, then all lost macroblocks for that frame are assumed to be inter-coded. Finally, the encoder uses a macroblock refresh mechanism that intra-codes macroblocks according to a pre-determined “walk-around” pattern. This walk-around refresh mechanism is used to clean up residual encoder/decoder reference frame mismatches.

Despite efficient error-concealment techniques, when a reconstructed video is played back in real time, visible distortions still appear in more than one slice, because of the error propagation effects. These distortions are visually annoying and are certainly not acceptable for streaming-based video entertainment applications. Note also that the damaged blocks in an I-frame causes reconstructual errors in the following P-frames or B-frames.

Figure 5 plots video distortion with mean decoded PSNR vs. burst loss length (in % of macroblocks) for four video sequences encoded at similar data and frame-rates. The results shows that video distortion is not similar for different sequences. Some streams (e.g. mother and daughter) are inherently more resilient to bursty errors as also evident from their calculated mean PSNR shown as in figure 5. Nevertheless, such objective assessment of video streams using measured PSNR however may not always translate to a similar subjective assessment (as that perceived by a human eye) for reasons described above. .

IV. ERROR-RESILIENT TECHNIQUES USED IN MOBISTREAM

In this section we describe the different techniques used in MobiStream to improve error resilience using virtual channels. These techniques are: (1) multiple description video coding, (2) slice structure-based video coding, (3) inter-frame and intra-frame slice interleaving, and, (4) dynamic unequal error protection, and, (5) priority-based slice scheduling.

A. Multiple Description Coding

MobiStream exploits error-resilience using multiple independent description of the video over virtual channels. With Multiple Description Coding (MDC) a video stream is split into multiple independent video sub-streams (description), and each stream is decoded *independently* at a somewhat reduced quality. This independence in the video sub-streams leads to error-resilience when streaming over virtual channels by limiting the error propagation to only one description.

MobiStream constructs multiple descriptions of video that are adapted to the characteristics of the virtual channel. For example consider MobiStream used two virtual channels: a reliable virtual channel (with ARQ enabled) and an unreliable virtual channel (ARQ and error correction disabled). Using *temporal subsampling*, MobiStream will construct two independent

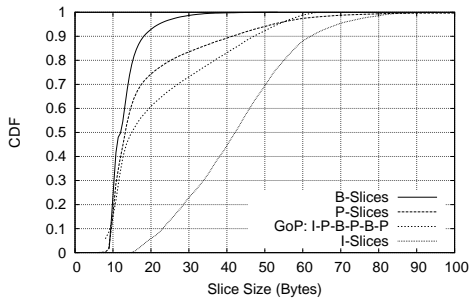


Fig. 6. Slice packet size distribution of an H.264 encoder in I, P and B-frame slices in the *Foreman* sequence. Each slice corresponds to a single macroblock.

video substreams out of original video sequence. For instance, a video stream encoded at a bit-rate of 144 Kbps at 15 fps can be constructed by MobiStream using MDC by splitting the same original video sequence now into two independently encoded video substreams, each encoded at a datarate of 72 Kbps at 7.5 fps. The resulting encoded video sub-streams are streamed over the virtual channels.

B. Slice-structured Video Coding

MobiStream uses slice-structured video coding as a fine-grained video coding mechanism to reduce video distortion due to burst packet losses.

A *slice* consists of one or more than one macroblocks (MBs) of a frame providing spatially distinct resynchronization points in the video data for that particular frame. Since no intra-frame prediction occurs across slice boundaries, the more the number of slices in a frame will also lead to increased error resilience at the expense of somewhat reduced coding efficiency.

Because intra-frame prediction cannot cross slice boundaries and because the probability of a short packet being corrupted or lost is lower than that for a larger packet, the overall packet loss probability will be reduced if these slice data-packets are made relatively small.

In figure 6 we plot the slice packet size distribution for I, P, and B-frame slices encoded at 144 Kbps 15fps for 300 frames of the *Foreman* video sequence with “ibpbbp” GoP structure. Note that each slice packet used in this distribution corresponds to one macroblock of a video frame. Therefore, for a QCIF format video sequence this corresponds to a total of $(9 \times 11) = 99$ macroblocks per frame or same number of slice data packets. We can see from the figure that majority ($> 80\%$) of these slice-data packets have size of less than 60 bytes.

MobiStream incorporates four different slice-group allocation schemes:

- *Allzero slices*: All-zero slice pattern consists of only one slice group, as typical scan-order slices (see fig. 7(a)).
- *Interleave slices*: In this pattern, macroblocks for even rows are assigned to slice group 0, while macroblocks in odd rows are assigned to slice group 1 (see fig. 7(b)).
- *Checker-Board slices*: A checker-board layout consists of the even rows, the even macroblocks are in slice group 0 and odd macroblocks are in slice group 1. (fig. 7(c))

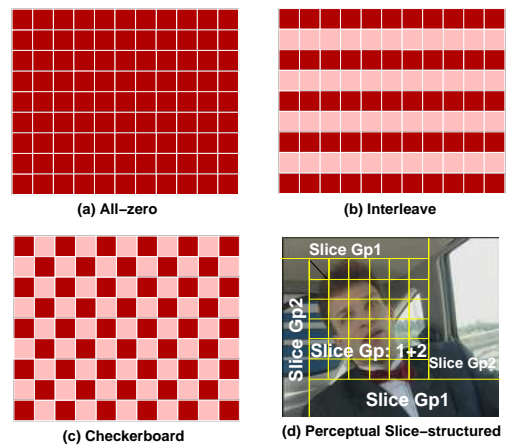


Fig. 7. Slice-group allocations using slice coding. Shows: (a) allzero slices, (b) interleaved slices, (c) checkerboard pattern slices, and, (d) proposed perceptual slice-structured coding.

Proposed Perceptual Video Coding. Slice groups can be efficiently utilized to code video frames based on ‘region-of-interest’. As described in Section IV, a region-of-interest coding technique mainly utilizes the human visual characteristics. For example, users typically pay more attention to a particular region-of-interest of a given video frame and they also tend to be less sensitive to the changes in the background (see figure 7(d)).

We exploit this inherent characteristic in video streams in our proposed *perceptual slice-structured* video coding. A region-of-interest in video frames can be encoded with a higher quality by choosing a better quantization parameter or by using a more error-resilient slice-group allocation. Figure 7(d) shows perceptual slice-structure coding scheme for a frame using different slice-group allocation scheme. In this example a frame from the ‘carphone’ sequence is slice-allocated in a way such that a scattered/checkerboard slice-group allocation is used to code the ‘region-of-interest’ whereas a less error-resilient ‘all-zero’ slice pattern is used for other less interesting regions of that frame.

An advantage of this type of slice-allocation is that priority of slices in each slice-group allocations can be easily specified. Therefore, slices from the region-of-interest of a frame will be prioritized more than slices that belong to other regions of that frame. Note that ‘region-of-interest’ for different video sequences can be dynamically detected using sophisticated frame object motion and edge detection mechanisms that are available for video.

C. Slice Interleaving

MobiStream uses slice interleaving technique as a tool to overcome the impact of burst losses in virtual channels. The objective is to *mitigate* the impact of such burst losses on video streams. Using slice interleaving adjacent lost slice-data packets of a frame are spatially distributed, which is less disturbing than concentrating the errors in one single region of the screen. When immediate neighbors of a lost macroblock are decoded successfully, the decoder can use the decoded information from

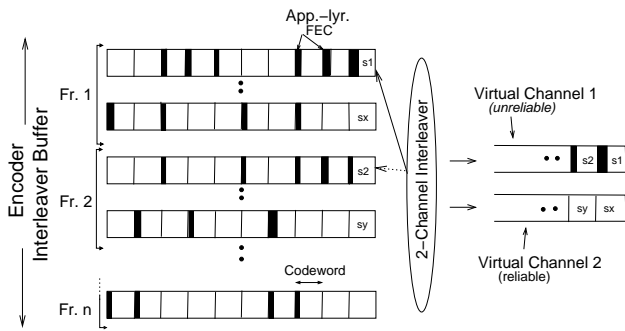


Fig. 8. Interframe slice interleaving over virtual channels.

neighboring macroblocks to predict the motion vectors and spatial content of the lost macroblock with better accuracy.

Two different levels of interleaving are used:

- (1) *Inter-frame slice interleaving*: In this scheme slices from multiple different frames are interleaved (fig. 8),
- (2) *Intra-frame slice interleaving*: In this scheme slices belonging to the same video frame are interleaved (fig. 9).

MobiStream implements a simple block-based interleaver for slice-data interleaving. In this approach, each individual slice data is first written as a codeword (including its FEC) row-by-row into an $m \times n$ matrix-array and read out column-by-column by the interleaver before it is streamed over virtual channels. However the reverse process is performed by the deinterleaver. Between successive symbols (bytes) of any given codeword there are $m - 1$ symbols that belong to the $m - 1$ other codewords being interleaved. If the interleaver has sufficient depth then channel fading that affects successive symbols belonging to the same codeword will be uncorrelated.

D. Unequal Error Protection

MobiStream protects data partitions in video streams using unequal error protection (UEP), but without adding significant overhead, according to their importance for reconstruction. UEP can significantly improve error resilience in streaming video over unreliable virtual channels.

MobiStream uses multiple shortened Reed-Solomon (RS) codes for dynamic Forward Error Correction (FEC) for use with slice-data packets. An $RS(n, k)$ reed soloman code represents an n symbol length code which contains k source symbols and $n - k$ protection symbols (as redundant data). $RS(n, k)$ code can correct up to $\lfloor (n - k) / 2 \rfloor$ symbols errors. Thus, if at least $n - k$ out of n symbols are correctly received, the underlying source information can be correctly decoded. Otherwise, none of the symbols can be recovered by the receiver.

In MobiStream there are different ways to apply UEP in the transport of priority slice-data packets. For instance, in the case of perceptual slice-structured coding, MobiStream protect slice-data packets (macroblocks) using UEP that belong to the region-of-interest (ROI). Thus slice-data packets from the ROI are protected better than slice-data packets available from other less interesting regions of a frame. In the same way, picture headers, motion vectors, picture parameter sets and I-frame slice-packets provide valuable video information and are offered strong protection.

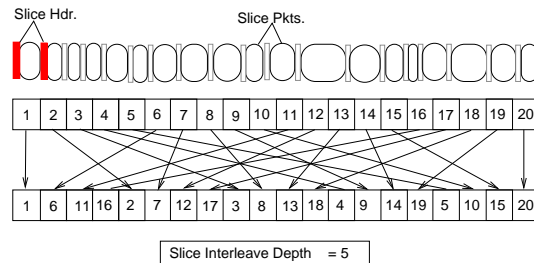


Fig. 9. Intraframe slice interleaving in MobiStream.

E. Slice Service Scheduling

MobiStream incorporates a scheduling strategy based on priority of the partitioned slice-data packets. Scheduling these slice-data packets over virtual channels however means that they are subjected to variable channel delays, burst packet losses and fluctuating bandwidths. This may lead to out-of-order arrival of slice data-packets at the receiver that are identified by the RTP [9] sequencing information. To overcome such problems of out-of-order slice-data packets arrivals over virtual channels, a receiver-end decoder can use a buffer to store and appropriately mark all the received (or lost) slice-data packet based on the information extracted from received RTP data packets.

Note that H.264 video decoder [1] used in MobiStream allows out-of-order decoding of slices. Hence reordering of slice-data packets in the jitter buffer is not required.

V. EXPERIMENTAL EVALUATION

In this section we report our experimental results. We first conduct tests over a cellular CDMA2000 EvDO network and show what levels of statistical loss guarantees video streaming applications can exploit over a WWAN network. Next, by replaying these traces for loss statistics with MobiStream, we show to what extent MobiStream can benefit video streaming performance using virtual channels. Finally, we also study the efficacy of the different error-resilient schemes used in MobiStream.

A. Links Tests with CDMA2000 EvDO WWAN

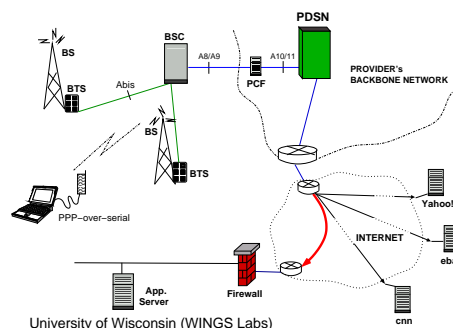


Fig. 10. Experimental Testbed Setup.

We conducted tests to obtain a realistic picture of the WWAN link performance and analyze what impact it can have on video

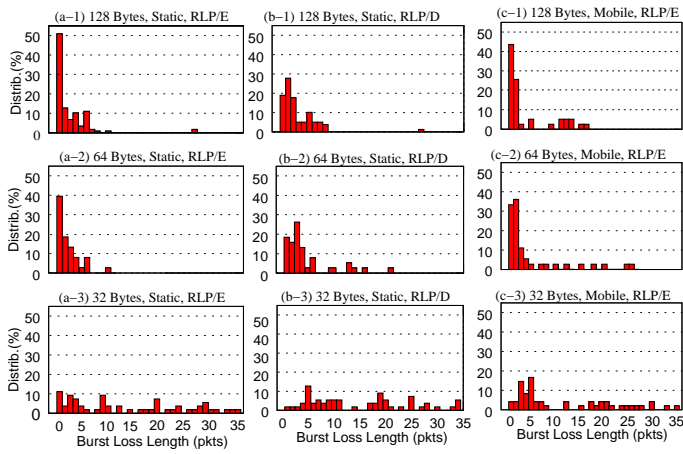


Fig. 11. Burst loss distributions for different packet sizes over a commercial CDMA2000 EvDO network. Each experiment consisted of 100000 back-to-back packet UDP samples at 144Kbps. *static* and *mobile* corresponds to stationary and mobile (drive) tests respectively. RLP/E and RLP/D indicates radio link layer as Enabled or Disabled by the client (Samsung SCH/a890).

streaming. Our tests had two important goals. First, we wanted to test what data-rate guarantees one can achieve in today's WWAN for efficient video streaming. Second, we wanted to evaluate the link loss statistics for different WWAN link-layer conditions. Finally, by using the statistical data made available from these tests, we measure its impact on MobiStream.

Testbed Setup: Our experimental setup consists of a commercial cellular CDMA2000 EvDO network testbed shown in figure 10. The client connects to the WWAN network using a PPP (point-to-point) link. In these tests we use a Samsung SCH-a1890 EvDO handset (PPP using serial link) with maximum data-rate of 2400 Kbps and 153 Kbps for forward(downlink) and reverse(uplink), respectively. Our CDMA EvDO operator provides a globally routable public IP address to its mobile hosts, sans firewall. Hence we are able to stream UDP test packets to-and-from a public server located in our Lab and our mobile host. In all these tests the error correction at the lower layer is kept enabled, i.e., packets that fail the checksum test are simply discarded.

We conducted streaming tests using UDP at three different data-rates of 64Kbps, 144Kbps, and 256 Kbps, respectively. For each experiment at these given data-rates, we used four different UDP sample packet size of 32 bytes, 64 bytes, 128 bytes and 256 bytes respectively. The choice of these packet size samples were made based on the typical slice packet-size distribution set for our encoder (figure 6).

Each experiment consisted of 100000 back-to-back UDP packet samples using `tcp+` installed in both the time synchronized mobile host (a laptop) and the server located in our lab. `tcp+` is modified to use a time-stamp and sequence number, hence we could detect lost packets or packets that arrived out-of-order. These tests were conducted in the following different scenarios: (1) stationary host (link-layer ARQ as network default), (2) mobile host while driving (link-layer ARQ as network default), and, (3) stationary host (link-layer ARQ disabled). We obtained a minimum of 5 sample sets for each individual test.

Trace Scenario	Streaming Tests (UDP pkt. trains at 64;144;256Kbps)					
	32 Bytes		64 Bytes		256 Bytes	
	tot.loss smpl.(%)	burst(%) len>5	tot.loss smpl.(%)	burst(%) len>5	tot.loss smpl.(%)	burst(%) len>5
Static-1(RLP/E)	0.12	79	0.21	21	0.28	26
Static-2(RLP/D)	0.53	86	0.51	34	0.88	37
Mobile(RLP/E)	0.36	83	0.53	23	0.64	29

TABLE I
PACKET LOSS STATISTICS FROM TESTS IN STATIONARY AND DRIVE (MOBILE) CONDITIONS OVER A CDMA2000 EvDO NETWORK.

Figure 11 show burst loss distributions from a sample set for different packet sizes at a datarate of 144 Kbps. We can observe that typical burst packet loss distributions for reasonably reliable “best-effort” channel (RLP/E) compared to unreliable channel (RLP/D) show burst loss length concentrations towards shorter burst loss lengths. Note that distributions for other data-rates were mostly similar, except that we observed less concentrated burst packets losses for the 256 Kbps streaming UDP tests.

Increasing the datarate significantly beyond 256 Kbps showed more such loss artifacts, hence we chose to use the more stable data-rates within this range. Since we have also observed out-of-order arrivals of packet samples in our tests, we conjecture that network default settings (RLP/E) of the link-layer is not set to achieve a very high level of reliability. Nevertheless, we can see that when packet sample size is increased, loss distributions show more concentrations towards smaller burst loss lengths.

Table I also summarizes loss statistics from each of these tests. It is clear from this table that for an unreliable channel, the total number of lost samples is higher and loss burst samples show more even (or flat) distributions, which also confirms that the probability of losing more successive data packets and total number of lost samples using an unreliable channel is more than a reasonably reliable “best effort” channel.

This result motivates our observation in MobiStream that video streaming applications can benefit from a range of the statistical loss guarantees available using virtual channels.

Parameters	Value
Video Sequence	Foreman
Sequence Length	300 frames
Format	QCIF (176 × 144 pixels)
Slice structure	1 slice per MB
Entropy Coding	CABAC
GOP Structure	I-B-P-B-P-B
Data Rate	144 Kbps
Frame Rate	15 fps
Data Partitioning	Disabled
Rate-Distortion Optimization	Disabled

TABLE II
H.264 ENCODER PARAMETERS USED IN MOBISTREAM EVALUATION.

B. Evaluating MobiStream with Virtual Channels

In this section we demonstrate how MobiStream improves error-resilience in video streaming using a range of techniques in different scenarios.

Experiment Setup: We have used the H.264/AVC encoder and decoder in MobiStream [1]. External to this codec software,

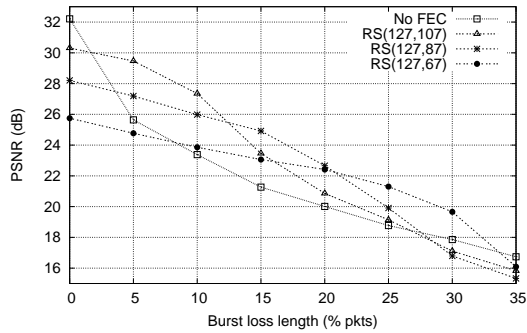


Fig. 12. Performance of dynamic application layer FECs using shortened Reed Solomon (RS) codes at different burst packet loss rates (% slice pkts) over an unreliable virtual channel. Vertical axis indicates the average luma PSNR (decoded) for *Foreman* video sequence.

we built all the necessary modifications for use with this codec and implemented the different techniques proposed to improve error resilience. In addition, we also instrumented the H.264 encoder with the ability to control the streams of a video sequence, the ability to statically induce burst packet losses by corrupting specific macroblocks within frames, and to then replay these damaged streams with an instrumented decoder that also logs video distortion.

Note that in our experiments we have used loss distributions obtained from the WWAN stationary and mobile tests reported in the previous section. We have used these loss statistics with the available software to induce burst losses and to replay these video streams with the decoder for each of the test scenarios as described. Unless stated otherwise, the H.264 encoder uses the parameters shown in table II.

The reported decoded PSNR is the arithmetic mean over the decoded luminance PSNR for frames of the encoded sequence. For all comparable results, the same starting position for a video sequence have been applied. It is also assumed that high-level syntax parameters (e.g. Picture Parameter Set) have been transmitted in advance and out-of-band using a reliable setup protocol. The RTP/UDP/IP/PPP overhead after RoHC (Robust header compression), and the link layer overhead is taken into account in the encoder's bit-rate constraints. The error concealment method in the decoder is based on that discussed in Section 4.

1) *MobiStream and Dynamic FECs with ViCs*: We demonstrate how *MobiStream* can use application-layer FECs to improve video streaming performance in the face of burst losses using Virtual Channels. *MobiStream* makes use of three different FEC codes: RS(127,107), RS(127,87) and RS(127,67) as specified in an increasing order of their ability to correct burst packet errors. Thus, dynamic FECs are applied in *MobiStream* using the shortened Reed Solomon (RS) codes to overcome burst losses over an unreliable channel.

Note that the choice of these codes are made based on packet loss distributions shown in the previous section. The codeword is of 1 byte size with multiple slice packets that can be aggregated to improve code efficiency and reduce header overhead. For example, two slice packets (macroblocks) in a B-frame may be combined to form a larger packet that also suits the selected code length.

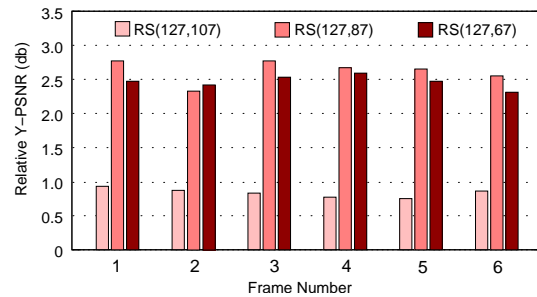


Fig. 13. Relative benefit in decoded PSNR using virtual channels for burst losses at 20% for different FEC used with unreliable virtual channels. Frames of a GoP (ipbpbpb) in the *Foreman* sequence are shown.

Figure 12 shows the performance trade-offs using different RS codes on mean decoded PSNR when packet burst loss rate is varied over an unreliable channel. It can be observed that, depending on the burst loss rate, RS codes can provide different degrees of benefit in their mean decoded PSNR. For example, when channel conditions are relatively good such that burst losses are in the range 0-10%, then use of RS(127,107) leads to optimal performance. However when losses are relatively high (20-30% burst losses) using a much heavier code like RS(107,67) provides the best gains in the decoded PSNR. Note however that the benefit of using a heavier code is at an expense of reduced coding efficiency.

Figure 13 plots the relative benefit in decoded PSNR using different RS codes for a burst loss of 20%. The figure shows the relative benefit in the PSNR for the first few frames of a group of pictures (GoP:ipbpbpb). It can be observed from this figure that using RS(127,87) code provides the best performance in decoded PSNR leading to improvements in the range between 2.4-2.8 dB whereas use of RS(127,67) increases coding overhead leading to a reduced PSNR benefit of just 0.6-0.9 dB.

2) *MobiStream with Multiple Virtual Channels*: We next evaluate the benefits using multiple virtual channels in *MobiStream*. In this experiment *MobiStream* makes use of two virtual channels. Each virtual channel provides a datarate of 72 Kbps, so a total available datarate of 144 Kbps. The first channel is a reliable virtual channel with link ARQ enabled (as network default). The second channel is an unreliable virtual channel (link layer ARQ disabled). For the unreliable virtual channel, *MobiStream* makes use of dynamic application layer FECs as described in the previous section. For these experiments we use a simple round-robin scheduler that schedules slice data packets across virtual channels.

Figures 14 (a) and (b) plot the mean decoded PSNR for the running 300 frames sequence in *Foreman* video when it is streamed over a WWAN link (without virtual channels). Hence for each burst loss event shown in figure 14(b) we can find that the PSNR for frames in the decoded stream in many instances drops below 22 dB. This leads to a poor overall picture quality of the decoded video stream that is also quite objectionable to the normal human eye.

In figure 14 (c) we plot the decoded PSNR using *MobiStream* when the same video sequence is streamed over two virtual channels. The channel in this case was subjected to stationary burst packet losses. But because *MobiStream* uses dynamic

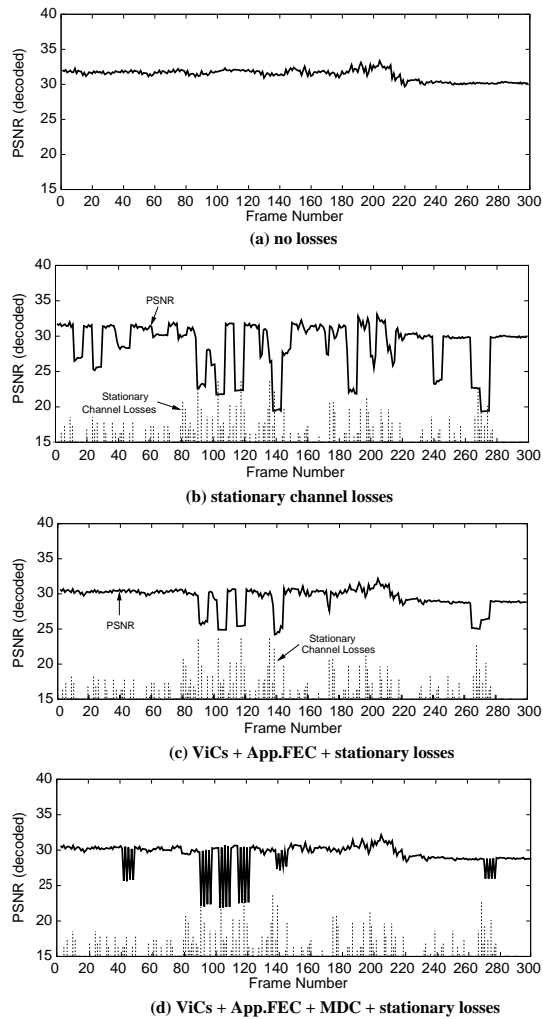


Fig. 14. Decoded PSNR for running 300 frames for *Foreman* video sequence. Shows (top-bottom): (a) decoded PSNR (no losses), (b) decoded PSNR in presence of stationary channel losses, (c) decoded PSNR using 2 virtual channels in presence of stationary losses, and, (d) decoded PSNR using 2 virtual channels with MDC in presence of stationary losses. Also shown are packet errors induced in different video frames (as dotted vertical lines).

FECs over the unreliable virtual channel, we can find that it can recover from most of these burst losses leading to much better and improved video picture quality. Figures 15 and 16 also plot the cumulative distributions of the decoded PSNR for *Foreman* sequence with stationary and mobile induced losses respectively. It is clear that by using multiple virtual channels and dynamic application-based FECs, MobiStream is able to significantly improve the video picture quality over WWANs.

3) *MobiStream and MDC with Virtual Channels*: The main difference in this scenario from that in the previous one is we split the video stream encoded at 144 Kbps (15 fps) into two independent video substreams using *temporal subsampling*. This means that for a given video sequence, MobiStream encodes two independent substreams for a maximum datarate of 72 kbps at 7.5 fps. Thus, combining these two independent descriptions gives an overall equivalent datarate of 144 kbps at 15 fps. Thus, MDC in MobiStream can create two completely independent video descriptions by intelligently skipping and coding the odd and even frames of a video sequence respectively.

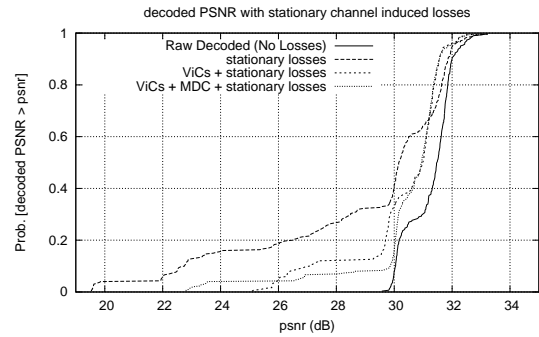


Fig. 15. Cumulative distribution of decoded PSNR for 300 frames in *Foreman* sequence. (stationary channel induced losses)

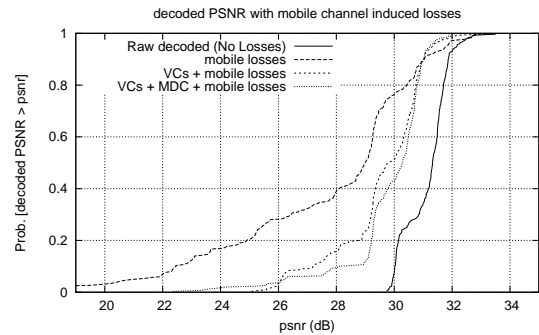


Fig. 16. Cumulative distribution of decoded PSNR for 300 frames in *Foreman* sequence. (mobile channel induced losses)

We conducted tests using MobiStream by streaming the two video substreams over virtual channels. Note that in these tests we disabled all other MobiStream enhancements, i.e., slice-structured coding, slice interleaving etc. For comparable results, we use the same starting positions and the channel statistics of mobile and stationary loss distributions that are taken from the experimental results of Section V-A.

Figure 14 (d) plots the decoded PSNR in MobiStream when video substreams are streamed over the respective virtual channels. We can see that MobiStream is again able to recover from most of the burst losses leading to further improvement in decoded picture quality. Figure 15 and 16 depicts the comparative benefit available using MDC over virtual channels and when not using MDC. Thus, we can see that MDC over virtual channels imparts extra error-resilience in video streaming and further improves the video picture quality in the presence of such burst losses.

4) *MobiStream and Slice-structured Coding with ViCs*: We evaluated the intrinsic error-resilience properties of the different slice-structure based coding schemes that are used in MobiStream. We compared four different slice-based coding techniques: (1) allzero slices, (2) interleaved slices, (3) checkerboard slices, and, (4) our proposed perceptual slice-structured coding scheme.

In these set of experiments we have used a channel bandwidth of 144 kbits/s. The target coding rate was however set to about 140 kbits/s; this rate was met by manually adjusting the quantization parameter in the encoder. Remainder of the bandwidth is left for different encoder-related constraints. In

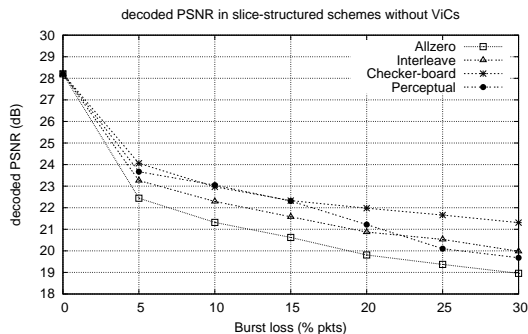


Fig. 17. Impact of burst packet loss (% slice pkts.) for various slice-structured coding schemes without using Virtual channels. The vertical axis indicates average luminance PSNR (decoded) for the *Foreman* video sequence.

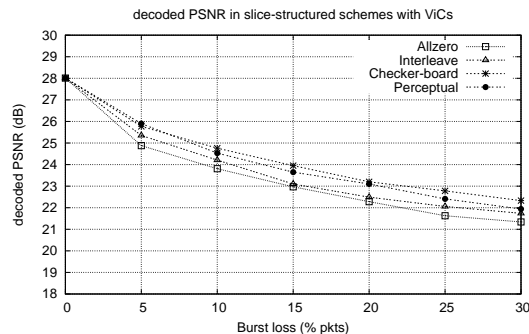


Fig. 18. Impact of burst packet loss rate (in % slice pkts.) for different slice-structured coding schemes using virtual channels. The vertical axis indicates average luminance PSNR (decoded) for the *Foreman* video sequence.

the following tests, the video streaming was assumed to start only after a setup interval during which the receiver buffer was filled. For these test a receiver buffer size of 5s is assumed.

We used MobiStream with and without the virtual channels. However for the scenario when the virtual channels are used, two such channels are considered each providing a datarate of 72 Kbps. The first is a reliable virtual channel and the second channel is an unreliable virtual channel (link layer ARQ disabled). Dynamic application-layer FECs are applied for data over the unreliable virtual channel. Except for the perceptual slice-based video coding scheme that uses a priority-based scheduler, rest of all our experiments make use of a simple round-robin scheduler that schedules slice data packets across virtual channels.

In figure 17 we plot the decoded PSNR for each of the slice-coding scheme when burst loss is varied. We can observe that, even when virtual channels are not used, slice-based coding improves the decoded PSNR of the video stream. We can see that interleaved-slices perform better than the allzero slices, while checkerboard-slices outperform all other slice-coding schemes w.r.t. mean decoded PSNR when burst loss rate is increased.

We conclude that video error concealment schemes perform very well when the lost packets as macroblocks are arranged in a checkerboard/scattered blocks fashion whereas interleave slice-pattern works relatively better. This is because as the distance between a corrupted block and the nearest error-free blocks increases, the distortion in recovered blocks grows. This arrangement is typically helpful in concealing the lost blocks by their surrounding blocks because images are generally smooth at block boundaries. Therefore, scattered slices are more easily concealed as compared to those concentrated in a small region. Subjective assessment of such decoded video streams shows that, at loss rates of up to 10%, the visual impact of the losses can be kept so low that only a trained eye can identify them.

Figure 18 plots the decoded PSNR when such slice-coding schemes used virtual channels. In this case we can observe that by using dynamic FECs over unreliable channels, MobiStream reduces the impact of burst losses to further improve the mean decoded PSNR (a 2-3 dB gain). Figure 20 show the relative benefits of the different slice-coding schemes using virtual channels for a GoP sequence in *Foreman*.

Note that *Foreman* sequence is known to be a very error concealment friendly video sequence, especially the later part of

second half with the camera pan. Subjectively, we do find that this part and the later stable picture looks significantly better with various slice-structured coding schemes.

Recall that, and as previously demonstrated in Section 4, objective improvements in the picture quality in video streams through measured PSNR may not always capture the ‘region-of-interest’ in frames that are perceptually (visually) more important in video sequences. Hence although checker-board/scattered slice-coding schemes shows better performance in the overall mean PSNR, the use of perceptual slice-structure coding still looks better to the human eye.

Figure 19 depicts each of these slice-coding coding schemes for the 55th frame of the decoded *Foreman* sequence that was subjected to burst loss rate of 20%. From each of these frame images we can clearly see (visually) how the slice-coding schemes perform (subjectively to a human eye), and therefore, perceptual slice-structured coding would be subjectively better than most other slice-coding schemes.

5) *MobiStream and slice interleaving with ViCs*: We conducted experiments to evaluate the gains using inter-frame and intra-frame slice interleaving over ViCs. Because of space constraints we only summarize the main result. The gains observed with slice interleaving in the face of burst loss depends on the slice interleaving depth. Using virtual channels and an interleave depth of 5, the relative benefit in PSNR observed is in the 2.0-2.7 dB range at a burst loss of 20%.

Slice-structured Coding Overheads: The overheads from slice-based coding, especially perceptual video coding and the use of slice interleaving, comes primarily from the broken in-picture prediction for the macroblock-line shaped slices and from the slice headers. For small picture sizes, this overhead may be substantial and can sometimes reach 10% or more of the coded bits, but the improvements in error resilience through error concealment is equally great.

The overhead comes from two sources: (i) the overhead of a compound packet is two bytes per carried data unit (called NAL unit in H.264), and, (ii) each macroblock line requires its own slice header, which is often somewhere between 2 and 3 bytes in size for QCIF (somewhat bigger for larger picture sizes).

Furthermore, there is an overhead associated with the fact that at least two RTP packets are used when slice interleaving is enabled (otherwise it could not help error concealment). The

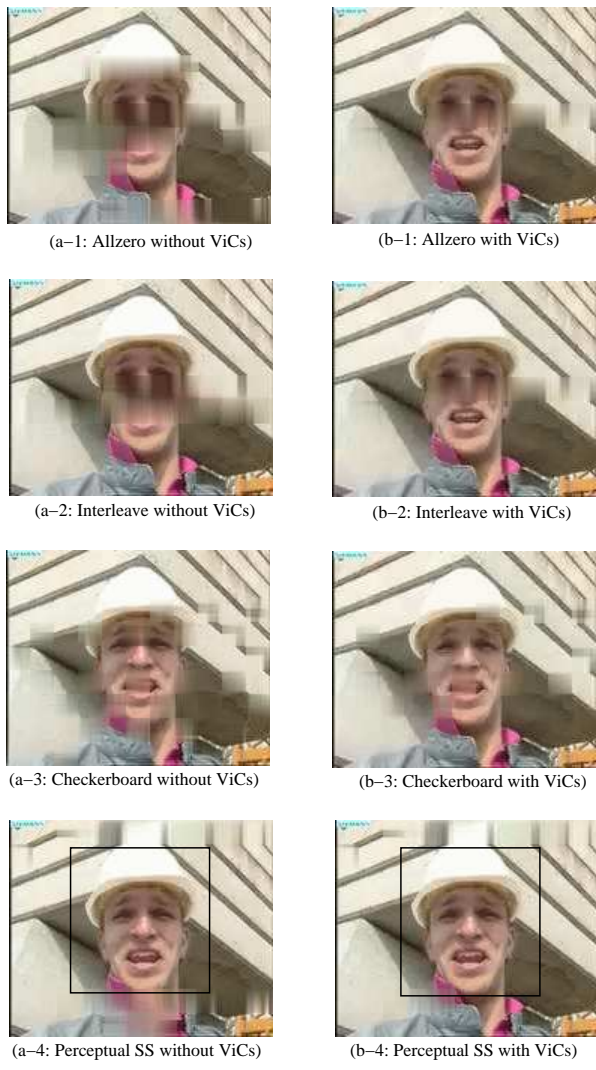


Fig. 19. Shows 55th frame of the *Foreman* video sequence using different slice-structured coding schemes. The left half of the figures shows these different slice-coding schemes without using virtual channels while right half shows the improvement resulting from use of virtual channels. Figures a-4 and b-4 show visually how perceptual slice-coding improves subjective video quality better than other slice-coding schemes.

overhead for an additional RTP packet is roughly 40 bytes (for the IP/UDP/RTP header). Note that with RoHC (robust header compression) this may be reduced to about 3 bytes or less.

A QCIF frame has 9 macroblock lines. With the above assumptions in mind, the overhead for slice interleaving is $9 \times 4 + 3$ bytes or 39 bytes per frame. At a frame rate of 15 fps, this results in 4.6 Kbit/s overhead at an operation point where 32 to 64 Kbit/s data rates are common. Thus, it is easy to see that the overhead for slice interleaving is roughly *less* than 10% of the bit rate (more for content where in-picture prediction is very beneficial and for I-frames).

VI. RELATED WORK

Prior research work has investigated issues related to video streaming in wired (e.g. [11][17]) and wireless environments (e.g. [4] [15],[18], [3]).

Our work in MobiStream differs from other prior works in several ways. In our work: (1) we introduce a link-layer based but application-controlled *virtual channels* abstraction for efficient streaming of video content over WWANs, (2) we exploit link ‘awareness’ based on statistical loss distributions to serve part of the video traffic, (3) we propose use of *applications-defined link-layer behavior* to allow (video) applications control the use of virtual channels, (4) we use perceptual video data prioritization to protect visually more important data than less important ones, and, (5) we provide fine-grained error-resilience using a range of available techniques over virtual channels.

In [18], Mao et al. use multiple description video coding with path diversity in wireless adhoc networks. In particular, they make use multiple description coding, reference picture selection in conjunction with layered coding and feedback. Miu et al. in [2] also performed a study of low-latency streaming video over WLAN networks exploiting path diversity.

Van der Shaar et al. in [15] proposed a cross-layer system for efficient streaming of video (MPEG-4) content over Wireless LANs. They show that an optimal packet size can be derived for efficient transmission based on channel (SNR) conditions in WLANs. Using this information, they further propose to dynamically change the MAC layer behavior (retransmissions) and use dynamic application-layer FECs for improving video streaming performance over Wireless LANs. MobiStream differs from the approach presented in [15] in that MobiStream exploits the link ‘awareness’ available from statistical packet loss distributions for virtual channels to enable efficient video streaming in WWANs. Related other studies like [6] and [23] have also investigated video streaming over wireless networks from a cross-layer design perspective.

In [13], Chesterfield et al. study streaming performance by exploiting diversity across multiple WWAN links and channels. They propose use of an application-layer repair strategy operating across multiple WWAN networks for streaming. Note that this is done by disabling the reliability (retransmissions) at the WWAN link-layer, and by allowing corrupted data be forwarded and then repaired by the application (this approach is similar to UDP-lite [3]).

MobiStream takes a different standpoint from that in [13] for video streaming. Instead of considering a WWAN link as either a hard-coded reliable link (retransmissions enabled) or an unreliable link (no link-layer retransmissions), MobiStream proposes the co-existence of multiple channel instances with variable degrees of reliability and partial channel control with application-layer FECs by using virtual channels. Thus, each virtual channel in MobiStream offers a different level of statistical loss guarantees, allowing video applications to adaptively prioritize and flexibly schedule video content across virtual channels.

In [12], Chakareski et al. compared layered coding vs. multiple descriptions for video over multiple paths. They show that, on one hand, layered coding over multiple paths provides scalable representation for video, on the other hand, multiple independent descriptions over such paths improves error resilience. Although MobiStream uses multiple descriptions for error resilience, it can very well leverage layered coding for scalable video representations over virtual channels. The primary con-

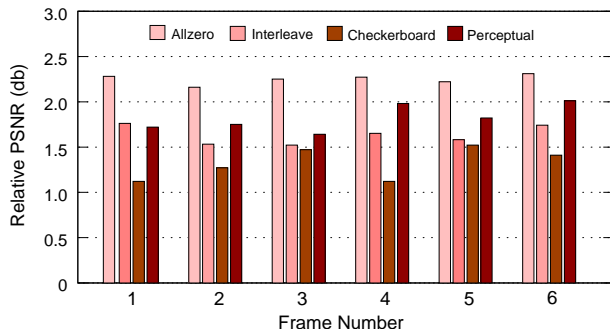


Fig. 20. Relative benefit in PSNR using virtual channels for different slice-based coding schemes at a burst loss rate of 20%. The different frames correspond to a GoP (ibppbb) in the *Foreman* video sequence.

cern using layered video coding is that the base layer needs to be sufficiently well-protected or streamed over a more reliable virtual channel.

Related other works have investigated improving video streaming over wireless links in other ways. For example, Cheung et al. in [7] present a double feedback streaming agent for improving video streaming over UMTS WWAN links while Zheng et al. in [10] use a UDP-based transport protocol for efficient media streaming in wireless networks.

In the recent years significant advances has been made within H.26x video standards. In fact, several research studies based on H.263/H.263+/H.264 have investigated different video coding techniques, e.g., [8],[14],[19],[22],[21] and [20]. Note that all of these works make use of RTP [9] as the transport protocol of choice for error-resilient video streaming. Reports on the performance of CDMA2000 networks, though not focussed on link packet loss statistics, is available in [16] and [5].

VII. CONCLUSIONS AND FUTURE WORK

We have presented MobiStream—a video streaming system that exploits: 1) the perceptual value of the video data, and, 2) the characteristics of the link layer and physical layer channels to enable error-resilient video streaming over WWANs. An initial prototype has been implemented and evaluated using loss distributions from tests conducted over a commercial wide-area wireless (CDMA2000 EvDO) network testbed. Results show that MobiStream is error-resilient in both stationary and mobile environments. We summarize the key features.

- **Virtual Channel (ViC) Abstraction:** ViC abstraction offers the levels of reliability and statistical loss guarantees to video applications using ‘awareness’ of the characteristics of the link layer and physical layer channels.
- **Fine-grained error-resilience:** By partitioning video frames into number of small, independently decodable, interleaved blocks of data (‘slices’), MobiStream can significantly improve video performance in presence of burst losses.
- **Perceptual Video Coding:** MobiStream associates priority to slices that have a higher perceptual value. This improves the overall subjective video quality.
- **Applications-defined link layer behavior:** Video applications can flexibly instantiate multiple virtual channels each with an

associated ‘soft’ reliability. Thus, partitioned video data can be very effectively prioritized over such channels.

We have planned further experiments using MobiStream for a spectrum of other video-based services, e.g., conversational services like cellular video telephony and digital video broadcasting services. Besides reusing many of the ideas proposed in MobiStream, we would also like to extend the concept of virtual channels abstraction in these different lossy environments that we feel would ultimately lead to fewer and less annoying video artifacts.

REFERENCES

- [1] H.264/AVC Software. <http://iphome.hhi.de/suehring/tml/>.
- [2] A. Miu, J. G. Apostolopoulos, W. Tan and M. Trott. Low-Latency Wireless Video Over 802.11 Networks Using Path Diversity. In *Proc. IEEE ICME*, July 2003.
- [3] A. Singh, A. Konrad and A. Joseph. Performance Evaluation of UDP lite for cellular video. In *Proc. of ACM Nossday*, 2001.
- [4] B. Girod and N. Farber. Wireless Video. *Compressed Video over Networks*, Marker Dekker, 2000.
- [5] E. Esteves, M. I. Gurelli, and M. Fan. Performance of Fixed Wireless Access with CDMA2000 1xEV-DO. In *Proc. of IEEE VTC*, 2001.
- [6] E. Setton, T. Yoo, X. Zhu, A. Goldsmith and B. Girod. Cross-layer Design of Ad Hoc Networks For Real-time Video Streaming. *IEEE Wireless Communications Magazine*, 2004.
- [7] G. Chueng, W-T. Tan and T. Yoshimura. Double Feedback Streaming Agent for Real-time Delivery of Media Over 3G Wireless Networks. *IEEE Transactions on Multimedia*, 6(2), 2004.
- [8] G. J. Sullivan and T. Weigand. Video Compression —From Concepts to the H.264/AVC Standard. *Proc. of the IEEE*, 93(1), 2005.
- [9] H. Schulzrinne et. al. RTP: A Transport Protocol for Real-Time Applications. In *IETF RFC – 1889*, January 1996.
- [10] H. Zheng and J. Boyce. An Improved UDP Protocol for Video Transmission Over Internet-to-Wireless Networks. *IEEE Transactions on Multimedia*, 3(3), September 2001.
- [11] I. Rhee. Error Control Techniques for Interactive Low-bit Rate Video Transmission over the Internet. In *Proc. of ACM Sigcomm*, 1998.
- [12] J. Chakareski, S. Han and B. Girod. Layered Coding vs. Multiple Descriptions for Video Streaming over Multiple Paths. In *Proc. of ACM Multimedia*, 2003.
- [13] J. Chesterfield, R. Chakravorty, I. Pratt, S. Banerjee and P. Rodriguez. Exploiting Diversity to Enhance Multimedia Streaming Over Cellular Links. In *Proc. of Infocom*, March 2005.
- [14] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer and T. Wedi. Video coding with H.264/AVC: Tools, Performance, and Complexity. *IEEE Circuits and Systems Magazine*, January 2004.
- [15] M. van der Schaar, S. Krishnamachari, S Choi, X. Xu. Adaptive Cross-layer Protection Strategies for Robust Scalable Video Transmission Over 802.11 WLANs. *IEEE Journal on Selected Areas of Communications (JSAC)*, 21(10), December 2003.
- [16] Q. Bi. A forward link performance study of 1xEV-DO system through field measurements. In *Bell-labs Technical Report*, March 2004.
- [17] R. Rejaie, M. Handley, D. Estrin. Layered Quality Adaptation for Internet Video Streaming. In *IEEE Journal on Selected Areas of Communications (JSAC), Special issue on Internet QoS*, November 2000.
- [18] S. Mao, S. Lin, S. Panwar, Y. Wang, and E. Celebi. Video Transport Over Ad Hoc Networks: Multistream Coding With Multipath Transport. *IEEE Journal on Selected Areas of Communications (JSAC)*, 21(10), 2003.
- [19] S. Wenger, M.M. Hannuksela, T. Stockhammer, M. Westerland and D. Singer. RTP Payload format for H.264 Video. In *IETF RFC – 3984*, February 2005.
- [20] T. Schierl, M. Kampmann and T. Weigand. H.264/AVC Interleaving for 3G Wireless Video Streaming. In *Proc. of ICME (under submission)*, July 2003.
- [21] T. Stockhammer, M. M. Hannuksela and T. Weigand. H.264/AVC in Wireless Environments. *IEEE Trans. on circuits and systems for video technology*, 13(7), July 2003.
- [22] T. Weigand, G. J. Sullivan, G. Bjontegaard and A. Luthra. Overview of the H.264/AVC Video Coding Standard. *IEEE Trans. on circuits and systems for video technology*, 2003.
- [23] Y. Shan and A. Zakhor. Cross Layer Techniques for Adaptive Video Streaming over Wireless Networks. In *Proc. of IEEE ICME*, 2002.