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 the video content and the characteristics of 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 in MobiStream offers a different level of reliability and statistical loss gaurantee using 'awareness' of the 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 to each such individual slice based on its perceptual (visual) usefulness. MobiStream augments a number of other enhancements for error-resilience: multiple description video 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 efficient and error-resilient video streaming over wireless wide-area links.

MobiStream has been implemented and evaluated using loss distributions from tests conducted over a commercial wide-area wireless (CDMA2000 3G) network. Results show that, MobiStream, in the face of burst packet losses can improve video picture quality by at least 4 dB. We conclude that significant benefits to end-user experience can be obtained by deploying such a video streaming system.

I. Introduction

Advances in digital video coding techniques coupled with universal cellular network upgrades supporting broadband data-rates (e.g., CDMA2000 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 rapid advances in wireless. The fact that video services in many WWANs are limited to simple video-clips *downloads* as opposed to actual video streaming shows the extent of difficulties and challenges that these WWAN environments still pose.

There are several impediments facing error-resilient video streaming over wide-area wireless cellular networks. First, data transmission over a radio channel is prone to bit errors or corruptions due to multi-path effects, shadowing, and interference. While medium or high persistence in the link layer and the use of channel coding in the physical layer can improve video performance to an extent, it still cannot avoid burst losses to occur

S. Banerjee is partially supported by NSF Award CNS-0520152

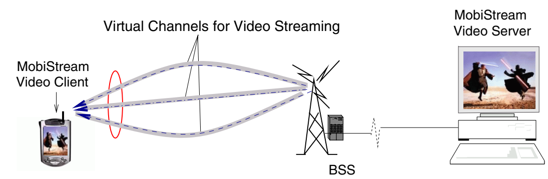


Fig. 1. **Virtual Channels (ViCs) for Wireless Wide-area Networks.**

(e.g, during fades) and may result in high transmission delay and jitter. Second, there is an inherent tussle between circuit voice and data users in many WWANs. Because available bandwidth is limited, each user is assigned a channel bandwidth that is a function of the signal strength and interference in 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. Third, limited number of data channels in many WWANs are often shared by many users resulting in 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 also 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 an user.

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

To this end, we present MobiStream— a video streaming system that enables efficient and error-resilient video streaming over WWANs. The key building block in MobiStream is a link layer based, but application-controlled, statistical loss-based *Virtual Channels* (ViCs) abstraction (Figure 1). From the perspective of (video) applications, ViCs provide statistical loss guarantees for *partitioned* video content flowing between the 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 ViC. While ViCs may not be able to offer the spectrum of service guarantees necessary for a range of other video-based applications, e.g., video telephony and

1-4244-0222-0/06/$20.00 (c)2006 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the Proceedings IEEE Infocom.

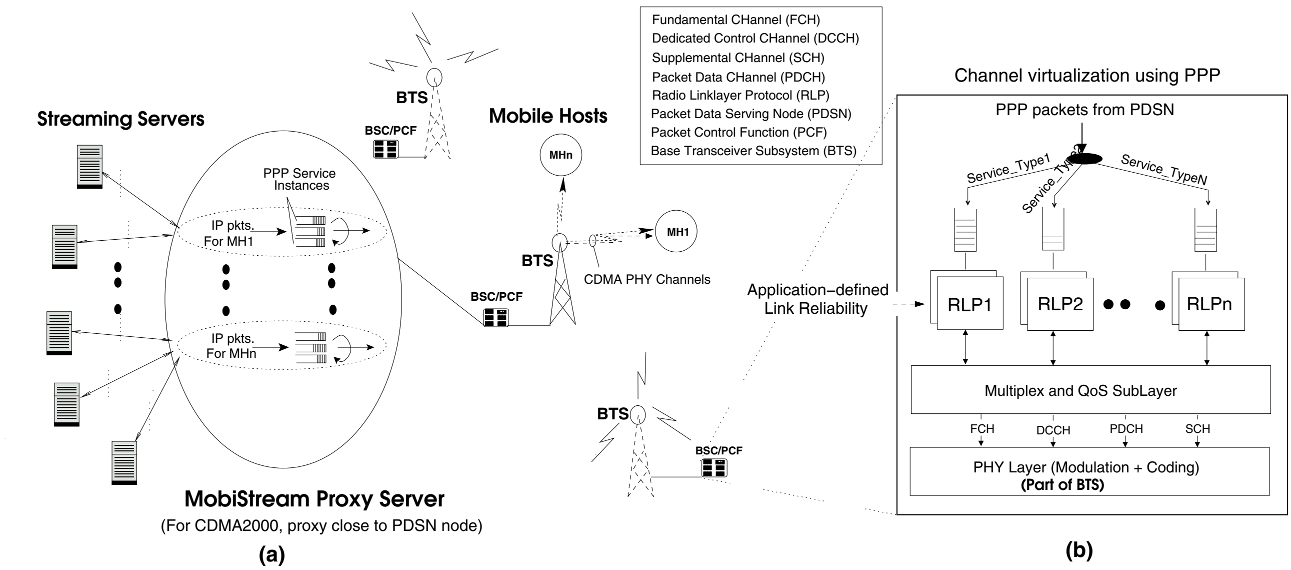


Fig. 2. **Shows (a) MobiStream architecture for error-resilient video streaming over wireless wide-area links, and, (b) construction of virtual channels using point-to-point (PPP) protocol. MobiStream proxy server is located close to the cellular provider's network. Example depicts a CDMA2000 cellular network with the MobiStream proxy located close to the PDSN node (not shown) or PCF.**

digital video broadcasting it can still provide adequate service guarantees for efficient and error-resilient video streaming. As we shall demonstrate in this work, there are several advantages of ViCs in WWANs:

1) Smoothing burst losses: Burst packet losses can have severe negative impact on video streaming applications. Using virtual channels MobiStream can reduce or even eliminate such burst loss effects leading to more efficient video streaming.

2) Perceptual video data prioritization: The inherent nature of the video content itself offers the possibility of very effective video data prioritization. MobiStream not only prioritizes video data streams over virtual channels, but also allows applications to 'express' the *perceptual* importance of each data packet (or slice) in a video stream, i.e., protect visually more appealing (video) data than less important ones.

3) Statistical loss guarantees: Using link and physical layer awareness of the loss distributions for a range of channel conditions, MobiStream can provide statistical loss control for each virtual channel to serve part of its traffic.

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

In this work we present our practical experiences using the MobiStream system. We first present its design and implementation, and then perform an extensive evaluation. Using tests conducted over a commercial cellular CDMA2000 3G 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 for bandwidths ranging from 64 Kbps upto 200 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 traces and packet loss distributions from stationary and drive tests taken from a commercial cellular (CDMA2000 3G) network. Our results demonstrate that MobiStream, on average, can improve video streaming (picture) quality even in the face of burst packet losses by at least 4 dB.

2) We introduce *virtual channels* as the key building block for video streaming in WWANs. Virtual channels depart than the traditional observation of WWAN link as a single hard-coded, fixed-reliability 'best-effort' data link. Instead, MobiStream views a WWAN link as composed of multiple virtual channels. Each virtual channel is an independent logical entity that offers a level of reliability with statistical loss guarantee. Virtual channels are flexible in the sense that they are dynamically instantiated, their statistical loss behavior controlled and/or allow flexible switching in video data transmission across channels. Video streaming applications particularly benefit from the flexibility and the awareness offered by virtual channels transmitting more useful data over more reliable channels and less important data over less reliable ones. However when a virtual channel is not reliable enough, it can still provide adequate flexibility to video applications for robust application-level

repair. MobiStream also augments a number of techniques for error resilience using virtual channels: multiple video 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 video frames along with the characteristics of the WWAN link layer and physical layer 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 burst packet loss on the visual perception of streaming video content. PSNR by itself cannot indicate the region-of-interest in a video frame sequence; it only measures bit-by-bit differences. To this end, we present *perceptual slice-structured* video coding that partitions video frames into number of small, independently decodable, blocks of data (called 'slices') and assigns priority to each individual slice using *region-of-interest* based coding. These frames of a video bitstream are then reconstructed with assigned priority based on their perceptual (visual) usefulness. Through real experiments, we demonstrate the effectiveness of perceptual slice-structured video coding.

Although prior work has investigated error-resilient video streaming in the context of both wired and wireless environments, our work significantly differs from each of them in several ways. In section VI we discuss related work.

Roadmap

This 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. In section VI we discuss 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 retransmission request (ARQ) to *variable degrees*.

2) The second motivates from an important observation that, instead of lower layers control wireless channels, let the video applications take *partial control* of these wireless channels using unequal FECs and/or retransmissions at the application layer itself.

Today's WWANs however lack mechanisms to support variable degrees of FECs or even retransmissions policies for

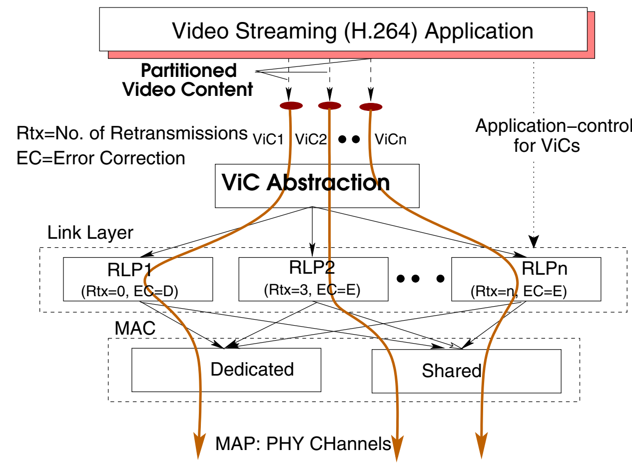


Fig. 3. **Illustration of the Virtual Channels (ViCs) Abstraction in WWANs.**

the *differently* marked packets. For instance, the newly deployed CDMA2000 3G cellular system offers reasonably reliable transmissions over the radio link meant for 'staple' Internet data with a Radio Link Protocol (RLP) that provides, at best, "best effort" level of link layer reliability and service guarantees set as network default.

The 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 such that it can 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 mobile (MN), each of which can have a different associated radio link protocol (RLP) setting.

Inspite of these encouraging developments, multimedia applications with diverse service requirements have failed to take advantage in ways reliability may be exploited from the link layer in these networks. Virtual Channels (ViCs) abstraction (see Figure 3) aims to exploit this advantage offered by the link layer in WWANs to allow video applications flexibly change and control the levels of reliability available at the lower link layer in these networks.

Video applications however still lack adequate control of the physical layer channels. This is because these network subsystems and terminals make use of the standard set of error detection, channel coding, puncturing, and interleaving schemes at the physical layer implemented in the hardware and meant primarily for circuit voice. In real cellular deployments this cannot be easily changed. As a result, applications other than circuit voice, having diverse set of service requirements are left with no other choice but to make use of the same physical layer channels that can offer only "best-effort" levels of service guarantees implemented through standard channel coding techniques (e.g., turbo or convolution codes) and optimised for circuit voice. Moreover, these cellular networks cannot effectively prioritize or flexibly schedule transmissions of the more important video data across wireless channels. Therefore, if video applications were able to take *partial control* of the physical layer channels, one way to achieve this is to let video applications implement

FECs at the application layer itself, i.e., disable link layer reliability and error correction/checksums at each of the lower layers and protocols and let video applications take control. In this circumstance, video applications implement FECs over Virtual Channels for 'outer channel coding' while the physical layer implements the standard 'inner channel coding'.

Note that in existing cellular deployment scenario it is neither easy nor economical to make BSS-wide software updates or hardware changes in order to implement new physical layer channel coding techniques for a spectrum of emerging multimedia applications. Instead, enabling partial channel control using Virtual Channels is rather straightforward and also very beneficial because it decouples the physical layer from the need to implement a new set of channel coding schemes whenever new multimedia applications are launched.

Virtual Channels abstraction proposed in this paper not only benefits video applications by providing variable degrees of channel reliability, but it also empowers them to take partial control of the wireless channels through implementation of FEC at the application layer itself. Note however that delays resulting from using FECs and retransmissions at the application layer may be excessive for a full-scale applications layer channel control. Evaluating this trade-off in real cellular deployment is an interesting topic for further study.

Virtual Channels and CDMA2000 Architecture. Virtual Channels and CDMA2000 Quality of Service (QoS) architecture can very well co-exist to the particular benefit of video streaming applications. In fact, Virtual Channels may even exploit many of the features available from the CDMA2000 QoS architecture [1]. 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: the required airlink data rate, requested airlink frame error rate, and acceptable airlink frame error rate. This set of QoS parameters in CDMA2000 is also 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 radio conditions, decides the optimal QoS multiplexing 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 interface.

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

Constructing Virtual Channels with PPP Protocol. Virtual Channels can be efficiently constructed using the point-to-point (PPP) protocol. As an example from the CDMA2000 architecture, IP packets received by the WWAN gateway called packet data serving node (PDSN) are sent over a tunnel as PPP packets to the BSC (see Figure 2). At the BSC, data is converted into 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 the data traffic in the form of PPP frames from the

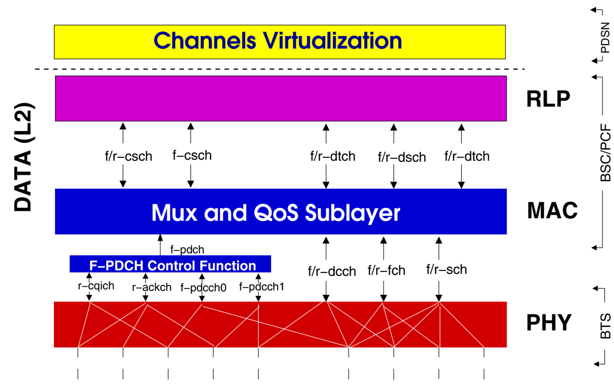


Fig. 4. **Illustration of how Virtual Channels (ViCs) in MobiStream can operate on top of the logical link layer (dedicated or shared) channels and physical layer data channels in CDMA2000 3G WWANs. Video content is scheduled over ViCs in a proxy at or close to the PDSN node.**

MN to the PDSN (reverse or the uplink) and from the PDSN to the MN (forward or the downlink).

Two cases arise in the way PPP is used.

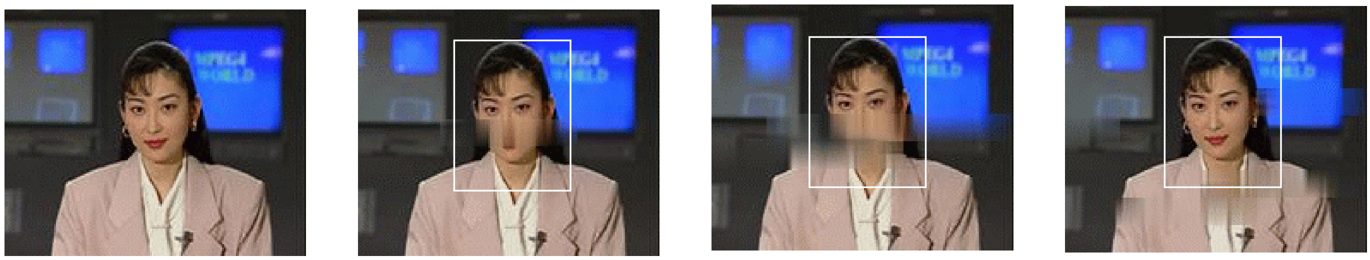
Case 1: No service differentiation. When no service differentiation is required using Virtual Channels to and from a mobile node or user, then mapping of all video data traffic onto a single RLP session is adequate. Thus, within the PPP frames, class based scheduling, for example, based on UDP/IP information, may 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 ViCs. 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 reference identifier (say from `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 may establish multiple service instances *dynamically* for multiple Virtual Channels with the radio access network (RAN). To do this, the MN first establishes a primary PPP service instance (with Service Option 33 or SO 33) before establishing multiple secondary service instances and specifying their QoS attributes (e.g., priority, min datarate, max delay, loss rate) with the PDSN node. The primary service instance is the 'best effort' non-assured mode, whereas the secondary service instances have assured mode service options (e.g., video, WWW) that could be defined. For example, Service Options 60 and 61 are already defined for Voice over IP (VoIP). Additionally new service options that are defined may also defer in the way RTP/UDP/IP headers are compressed, i.e., link layer assisted robust header compression (LLA-RoHC) for transparent header compression and good enough header compression (GEHCO) for nontransparent header compression.

In this manner each RLP session originating within a given service instance corresponds to a Virtual Channel that can have parameters such as the number of retransmissions (i.e.,

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the Proceedings IEEE Infocom.



(a) No Loss (PSNR: 44.6) (b) 5% loss (PSNR: 34.6) (c) 20% loss (PSNR: 23.7) (d) 20% loss (PSNR: 24.1)

Fig. 5. **Shows the impact of burst packet loss on video picture quality for the 3^{rd} frame in *akiyo* video sequence. (The grey inner rectangle indicates the *region-of-interest* in this frame.)**

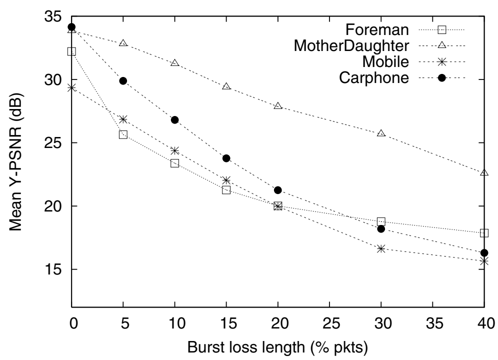


Fig. 6. **Degradation in mean PSNR due to burst losses (in % of slice pkts) for different video sequences with decoder error concealment enabled.**

link reliability) meant for all the RLP frames of that particular session. Thus, when retransmissions are not needed for certain portions of the video content streamed over a particular Virtual Channel, it could be turned off just for that particular session. Other Virtual Channels however may possibly change the number of retransmissions or instantiate new channels with required number of retransmissions (reliability) for efficient streaming of the partitioned video content.

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

There are two logical channels in CDMA2000: Fundamental CHannel (FCH) and Supplemental CHannel (SCH). These logical channels are then mapped to physical layer channels (Figure 5). 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 the RC3, which includes a 9.6 kbps FCH. Therefore, the peak forward link data rate for a user in CDMA2000 network is as high as that offered by the sum of all, *i.e.*, an aggregate datarate for 2SCH + 1FCH + 1DCCH can be obtained from 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 are 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 typically assigned to the 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 consequences on the transport of compressed video bitstreams because any damage to the compressed bit-stream may lead to objectionable visual distortion at the decoder.

To illustrate the visual artifacts caused by the impact of *burst packet errors* in a compressed video bitstream, in Figure 4 we show a reconstructed (decoded) video frame from the *akiyo* video sequence. The original video frame was encoded using the H.264 encoder [2] at an average bitrate of 256 Kbps with a macroblock quantization parameter (QP) of 20. This is shown in the image in Figure 4(a) (psnr of 44.6). Note that value of QP in H.264 range between 1 (highest) and 51 (lowest).

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 the video frame (shown in the white rectangular box). This visible perceptual distortion degrades further when burst loss length is increased to 20% (leading to 20 damaged macroblocks in that frame) shown in Figure 4(c) (psnr of 23.7).

However when burst loss events do not corrupt macroblocks that belong to a region-of-interest of a given frame, the overall perceptual distortion is not as severe. This is evident in the image shown in Figure 4(d) (psnr of 24.1) where a burst loss of 20% (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 lost macroblocks for the frame in Figure 4(d) are concentrated outside the region-of-interest. Therefore, the reconstructed frame

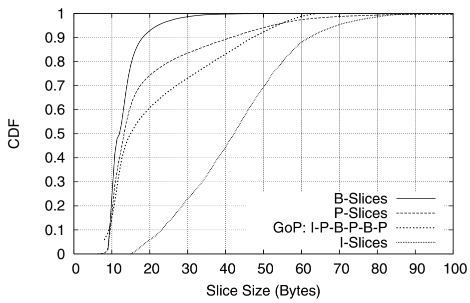


Fig. 7. **Slice packet size distribution for the H.264 encoder with the I, P and B-frame slices for the *Foreman* video sequence. Each slice corresponds to a single macroblock.**

in this case looks significantly better even when compared to the 5% burst packet loss scenario depicted earlier in the frame shown in Figure 4(b). This example clearly demonstrates the importance of perceptual region-of-interest based coding of video sequences.

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

Using the latest H.264 video codec implementation [2] as an example 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 the lost macroblocks for that frame are assumed to be inter-coded. Finally, the encoder also uses a mechanism that intra-codes macroblocks according to a pre-determined "walk-around" pattern. This mechanism is used to clean up residual encoder and decoder reference frame mismatches.

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

Figure 6 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. These results show that video distortion are not necessarily the same for different video sequences. Some streams (e.g., mother and daughter) appear inherently more resilient to burst errors as evident from their measured mean PSNR shown in Figure 6. Nevertheless, such objective assessment of video bitstreams using measured PSNR however may not always translate to a similar subjective assessment (as that perceived by a human eye) for reasons described above.

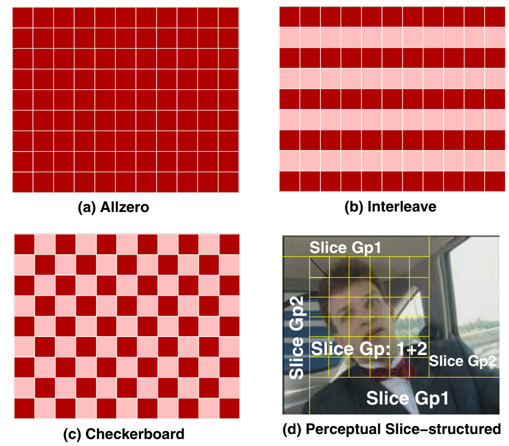


Fig. 8. **Slice-group allocations using slice coding. Shows: (a) allzero slices, (b) interleaved slices, (c) checkerboard pattern slices, and, (d) an example of proposed perceptual slice-structured coding.**

IV. Error-resilient Techniques used in MobiStream

In this section we describe the different techniques used in MobiStream that improve error resilience using virtual channels. We consider five different techniques: (i) multiple description video coding, (ii) slice structure-based video coding, (iii) inter-frame and intra-frame slice interleaving, (iv) dynamic unequal error protection, and, (v) priority-based slice scheduling. We describe how each of these techniques help improve video streaming performance in MobiStream using virtual channels.

A. *Multiple Description Video Coding*

MobiStream exploits error-resilience by streaming multiple independent descriptions of video over virtual channels. With Multiple Description Coding (MDC) a video bitstream is split into multiple independent video sub-streams (descriptions), and each sub-stream is decoded *independently* at a somewhat reduced quality. This independence in the video sub-streams leads to better error-resilience when streaming video descriptions over virtual channels having *little or no correlation*, thus, also possibly limiting the error propagation to only one channel or video description.

MobiStream constructs multiple descriptions of video adapted to the characteristics of the virtual channel. For example consider MobiStream used two virtual channels: a reliable virtual channel with link layer ARQ enabled (set as network default) and an unreliable virtual channel where link layer ARQ and error correction/checksums are disabled. Using *temporal subsampling*, MobiStream can construct two independent video substreams from the original video sequence. As an example, 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 into two independently encoded video substreams, by alternately skipping the even and odd frames in each substream that is encoded at a datarate of 72 Kbps and 7.5 fps. The resulting encoded video sub-streams are streamed over virtual channels.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the Proceedings IEEE Infocom.

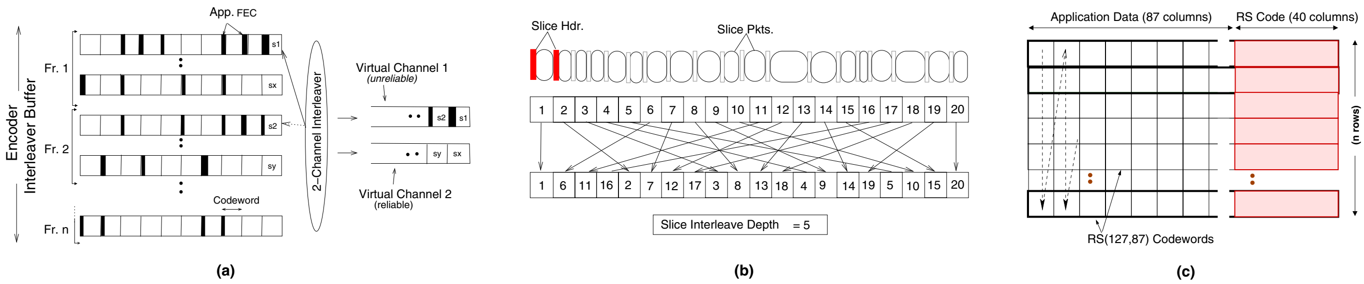


Fig. 9. **Shows (a) multi-channel interframe slice interleaving, (b) intraframe slice interleaving, and, (c) example RS(127,87) coding and interleaving combined into *virtual block-based interleaving*.**

B. Slice-structured Video Coding

MobiStream implements slice-structured video coding as a fine-grained video coding mechanism to alleviate the video distortion caused by the impact of burst packet loss.

A *slice* consists of one or more than one macroblocks 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 also leads 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 in a wireless channel 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 7 we plot the slice packet size distribution (including the headers) for I, P, and B-frame slices encoded at 144 Kbps 15fps for 300 frames of the *Foreman* video sequence with "ibpbpb" Group of Picture (GoP) structure. Note that each slice data 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 less than 60 bytes.

MobiStream implements four different slice-group allocation schemes:

- *Allzero slices:* Allzero slice pattern consists of only one slice group, as typical scan-order slices (Figure 8(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 (Figure 8(b)).
- *Checkerboard 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. (Figure 8(c))

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 based coding scheme 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 tend to be less sensitive to other changes in the background (see Figure 8(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 picture quality by choosing a better macroblock quantization parameter or by using a more error-resilient slice-group allocation.

A perceptually better and error-resilient slice-group allocation scheme can result in more effective and error-resilient streaming of the video content. Figure 8(d) shows perceptual slice-structured coding scheme for a frame using different slice-group allocation schemes. In this example a frame from the 'carphone' sequence is slice-allocated in a way such that a perceptually better 'checkerboard' slice-group allocation is used to code the 'region-of-interest' of that frame whereas a perceptually less error-resilient 'allzero' slice allocation is used for other less interesting regions of that frame.

Perceptual slice-structured video coding is however by no means limited to a few such slice-group allocation schemes. In fact, perceptual slice coding can support a range of error-resilient slice-group allocations with more number of slice groups to use for fine-grained error-resilience. However for QCIF size picture the limit on the slice-group allocation is 99 where each macroblock in a given QCIF frame is a slice.

An advantage of such a slice-group allocation is that priority of slice groups in a video frame can be very effectively specified. Therefore, slices from the region-of-interest of a frame are prioritized more than slices that belong to other regions of that frame. This also facilitates unequal error protection. Note that 'region-of-interest' for different video sequences can be dynamically determined using sophisticated frame object motion and edge detection mechanisms that are available for video.

C. Slice Interleaving

MobiStream uses slice interleaving technique in conjunction with application level coding (FEC) as a tool to overcome the impact of burst losses over virtual channels. The objective in interleaving slice data packets is to *mitigate* the impact of such burst loss events on the video bitstreams. Using slice interleaving adjacent lost slice data-packets of a frame are spatially distributed, which is less disturbing to a human user than concentrating the errors (or lost macroblocks) 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 neighboring macroblocks to predict the motion vectors and spatial content of the lost macroblock with better accuracy.

Two different levels of interleaving are possible in MobiStream:

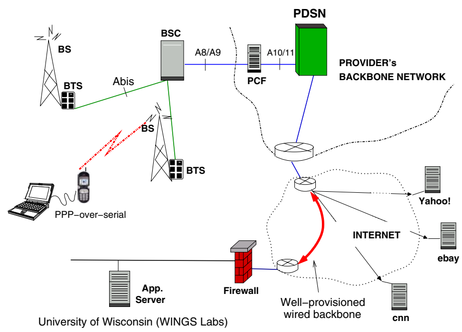


Fig. 10. **Experimental Testbed Setup.**

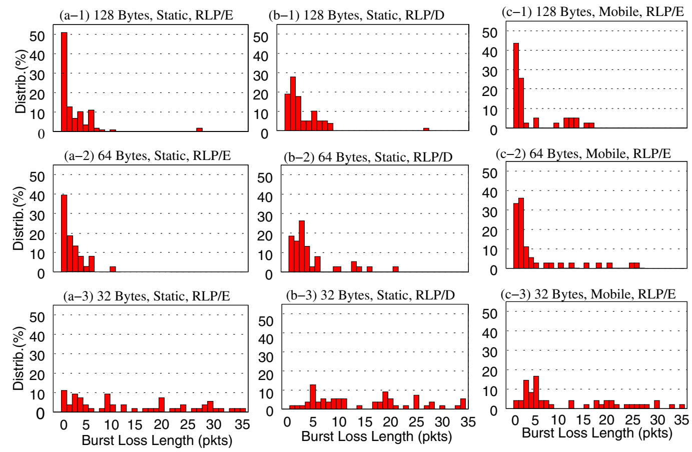


Fig. 11. Sample burst loss distributions for different packet sizes over a commercial CDMA2000 EvDO network. Each experiment consisted of 100000 back-to-back packet UDP samples at 144Kbps. `static` and `mobile` corresponds to stationary and mobile (drive) tests respectively. RLP/E and RLP/D indicates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

1) *Inter-frame slice interleaving:* In this scheme slices from multiple frames are interleaved (Figure 9(a)),
2) *Intra-frame slice interleaving:* In this scheme slices from a single video frame are interleaved (Figure 9(b)).

In inter-frame slice interleaving, individual slice data packets (including any FEC) from multiple video frames are systematically written into a hypothetical $m \times n$ matrix-array of depth d, where d is the number of frames in the encoder buffer that can be interleaved. Each matrix-array is then read out by the interleaver frame-by-frame after which these slice data packets are encoded and streamed over virtual channels. If the interleaver has sufficient depth to interleave a number of video frames then channel fading that may affect slice data packets across multiple video frames will be uncorrelated. The same is applicable when interleaving slice-data packets of a single video frame. Moreover, both these approaches can be easily adapted for unequal error protection (UEP) described next.

D. Unequal Error Protection

MobiStream enables unequal error protection (UEP) of video data partitions through dynamic application level FECs, but without adding significant overhead, according to their importance for reconstruction. In MobiStream there are two ways to apply UEP: 1) by applying slice interleaving and unequal coding on slice data-packets *independently*, or, 2) by combining the coding with a block-based data interleaver to perform *virtual block interleaving* (see Figure 9(c)). The latter approach is used in digital video broadcasting for handhelds .

Using the schemes above, MobiStream can make use of multiple shortened Reed-Solomon (RS) codes for dynamic application level coding to offer unequal protection to partitioned slice data packets. To illustrate this with a simple example, an RS(n,k) code represents an n symbol length code that contains n source symbols and n-k protection symbols (as parity symbols). RS(n,k) code can correct up to n-k errors in known positions (erasures) and [(n-k)/2] symbols received in errors in unknown positions. If at least k out of n symbols are received correctly, the underlying source information can be fully decoded. Otherwise, none of the error symbols can be recovered.

UEP can help improve error resilience in the streaming of partitioned video content especially over unreliable virtual channels. For instance, in the case of perceptual slice-structured video coding, MobiStream can code slice-data packets (macroblocks) with a more powerful RS code that belong to a region-of-interest (ROI). Thus, slice-data packets from a ROI are offered stronger protection as compared to slice-data packets available from other less interesting regions of the frame. In the same way, picture/slice headers, motion vectors, picture parameter sets, and I-frame slice-data packets provide valuable video information and are offered stronger protection.

E. Slice Service Scheduling

MobiStream incorporates a scheduling strategy based on the priority of the partitioned slice-data packets. Scheduling these slice-data packets over virtual channels however imply that these packets will be subjected to variable channel delays, burst losses and fluctuating channel bandwidths. This may lead to out-of-order arrival of slice-data packets at the receiver that are then identified by the RTP [3] sequencing information. Therefore, 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 corrupted/lost) slice-data packet based on the information extracted from received RTP data.

Note that H.264 video decoder [2] used in MobiStream allows out-of-order decoding of slices for several successive frames. Hence reordering of slice-data packets in the jitter buffer is typically not required.

V. Experimental Evaluation

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

TABLE I

LOSS STATISTICS FROM STATIONARY AND DRIVE (MOBILE) TESTS OVER A CDMA2000 EVDO NETWORK.

	Streaming Tests (UDP pkt trains at 64;144;256Kbps)					
	32 bytes		64 bytes		256 bytes	
Trace Scenario	loss (%)	burst(%) len>5	loss (%)	burst(%) len>5	loss (%)	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 II

H.264 PARAMETERS USED IN MOBISTREAM EVALUATION

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

A. Link Tests with CDMA2000 EvDO WWAN

We conducted tests to obtain a realistic picture of the WWAN link performance effects and to analyze what impact it can have on video streaming. Our tests had three important goals. First, we wished to evaluate what guaranteed datarate one can expect in todays WWANs for efficient streaming of video content. Second, we wanted to evaluate the link packet loss statistics under different WWAN conditions. Finally, by using the statistical loss information 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-a890 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 routeable 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 Laboratory and our mobile host. In all these tests the error correction/checksum offered by the lower layers is kept enabled, i.e., packets that fail the error checksum test in any layer 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 individual 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 for datarates set for our encoder (Figure 7).

Each experiment consisted of 100000 back-to-back UDP packet samples using `ttcp+` installed in both the time synchronized mobile host (a laptop) and the server located in our lab. `ttcp+` is modified to use a time-stamp and sequence number, hence we are able to detect lost packets or packets that arrived out-of-order. Note that in all our experiments loss events are due to wireless because: i) the wired part is sufficiently well-provisioned, ii) our tests were conducted at datarates far lower than required to overwhelm the wired part or even cause cross-traffic problems, and, more importantly, iii) our tests clearly reveal that loss events are wireless related, i.e., bigger size packets have higher loss probability than smaller size packets.

Our 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). For each test we obtained a minimum of 5 samples set.

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

Increasing the datarate significantly beyond 256 Kbps showed increase in such loss artifacts, hence we chose to use the more stable datarates within this range. Since we also observed some out-of-order arrivals of packet samples in our tests, we conjecture that network default setting (RLP/E) of the link layer is not set to achieve a very high level of reliability or in-order packet delivery. In any case, a local hardware configuration issue seems less likely. Nevertheless, we can see in Figure 11 that when sample packet size is increased, loss distributions show more concentrations toward smaller burst loss length.

Table I summarize 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 the total number of lost samples using an unreliable channel is more than a reasonably reliable "best effort" wireless channel.

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

B. Evaluating MobiStream with Virtual Channels

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

Experiment Setup: We have used the H.264/AVC encoder and decoder in MobiStream [2]. External to this codec software, 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

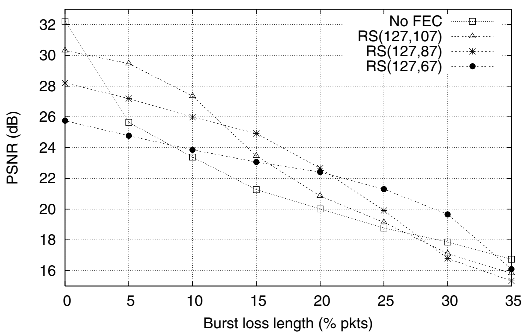


Fig. 12. Performance of dynamic application layer FECs using shortened Reed Solomon (RS) codes at different burst packet loss rates (% slice pkts) over an unreliable virtual channel. Vertical axis indicates the average luma PSNR (decoded) for *Foreman* video sequence.

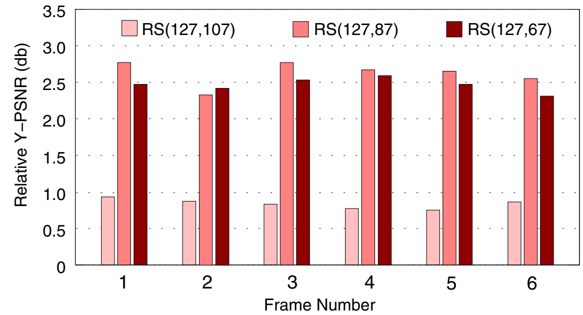


Fig. 13. Relative benefit in decoded PSNR using virtual channels with burst loss at 20% for different FEC used with unreliable virtual channels. Frames of a GoP (ibpbpb) in the *Foreman* video sequence are shown.

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 (Section V-A). We have used these loss statistics with the available software to induce burst loss events and to replay these video streams with the decoder for each of the test scenarios as described. Unless stated otherwise, the H.264 encoder will use the parameters shown in Table II and Virtual Channels show no correlation.

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 their 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.

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(127,67) provides better gains in the decoded PSNR. However the benefit of using a heavier code comes 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:ibpbpb). It can be observed from this figure that, at burst loss of 20%, using RS(127,87) code provides the best performance in mean decoded PSNR leading to improvements in the range between 2.4-2.8 dB whereas use of RS(127,67) increases coding overhead somewhat to marginally reduce the PSNR benefit by 0.2-0.4 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. It is assumed that each virtual channel offers 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. In 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 video sequence in *Foreman* 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 an overall poor picture quality of the decoded video bitstream 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 channels were uncorrelated and subjected to stationary burst packet losses. But because MobiStream uses dynamic FECs over the unreliable virtual channel, we find that it can easily recover from most of these burst losses leading

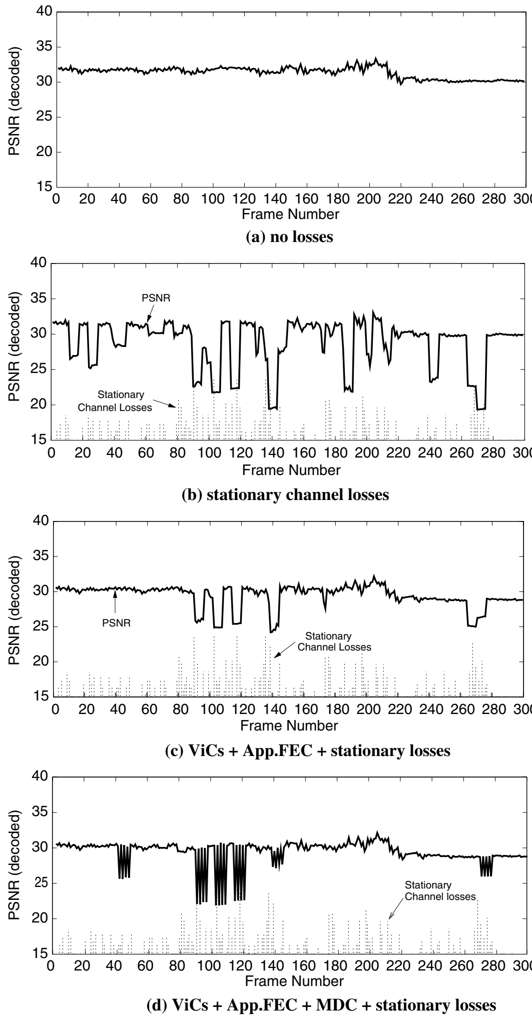


Fig. 14. Decoded PSNR for running 300 frames for *Foreman* video sequence. Shows (top-bottom): (a) decoded PSNR (no losses), (b) decoded PSNR in presence of stationary channel losses, (c) decoded PSNR using 2 virtual channels in presence of stationary losses, and, (d) decoded PSNR using 2 virtual channels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

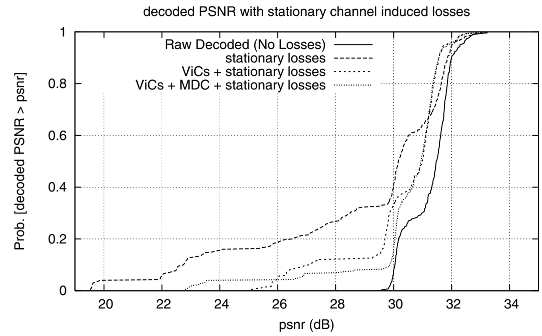


Fig. 15. Cumulative distribution of decoded PSNR for 300 frames in *Foreman* sequence. (stationary channel induced losses)

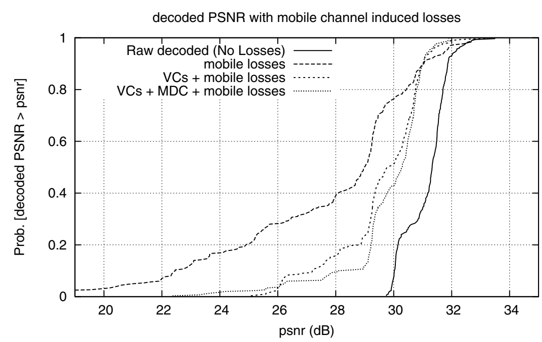


Fig. 16. Cumulative distribution of decoded PSNR for 300 frames in *Foreman* sequence. (mobile channel induced losses)

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 the previous one is that 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 video substreams for a maximum datarate of 72 kbps at 7.5 fps. Thus, combining these two independent descriptions give an overall equivalent datarate of 144 kbps at 15 fps. Hence, MDC in MobiStream can create two completely independent video description by intelligently skipping and coding the odd and even frames of a video sequence respectively.

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. Figures 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 kbps. The target coding rate was however set to about 140 kbps; 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 the following tests, the video streaming was assumed to start only

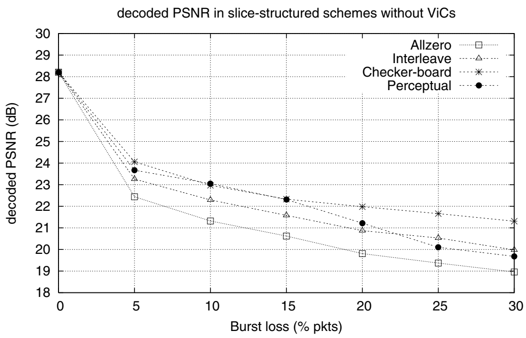


Fig. 17. Impact of burst packet loss (% slice pkts.) for various slice-structured coding schemes without using Virtual channels. The vertical axis indicates average luminance PSNR (decoded) for the *Foreman* video sequence.

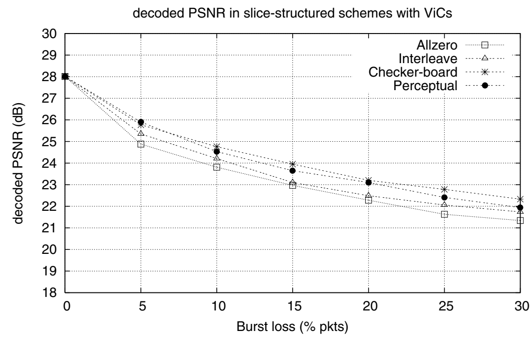


Fig. 18. Impact of burst packet loss (in % slice pkts.) for different slice-structured coding schemes using virtual channels. The vertical axis indicates average luminance PSNR (decoded) for the *Foreman* video sequence.

after a setup interval during which the receiver buffer was filled. For these tests a receiver buffer size of 5 seconds 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 one 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 over virtual channels.

In Figures 17 we plot the decoded PSNR for each of the slice-coding schemes when burst loss rate 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 or scattered blocks fashion whereas interleave slice-pattern works only relatively better than allzero. 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 in some cases show that, at loss rate of up to 10%, the visual impact of the such losses can be kept low that only a trained eye could identify them.

Figure 18 plots the decoded PSNR when such slice-coding schemes are used with virtual channels. For this case we can observe that by using dynamic FECs over unreliable channels, MobiStream reduces the impact of burst losses to further improve 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*.

The *Foreman* sequence is known to be an 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 other stable part in this video sequence looks significantly better with the various slice-structured coding schemes.

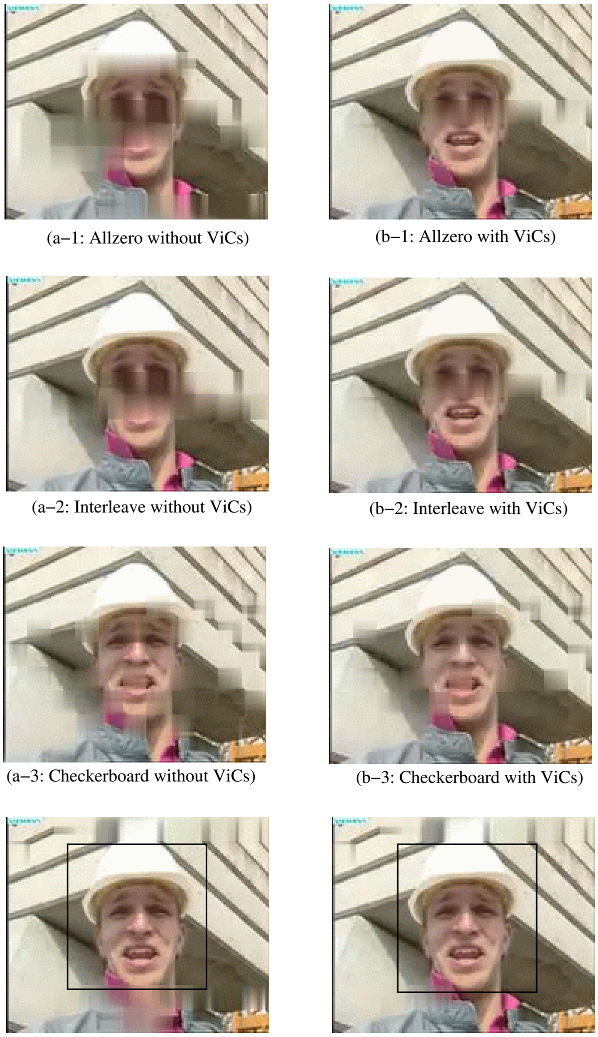
Recall that, and as previously demonstrated in Section 4, objective improvements in picture quality in video streams through measured PSNR may not always capture the 'region-of-interest' in frames that are perceptually (visually) more important in video sequences. Therefore, although checker-board/scattered slice-coding schemes shows better performance in measured mean PSNR, the decoded frame using perceptual slice-structured coding looks better to a human eye.

Figure 19 depicts the use of each of these slice-coding coding schemes for the 55th frame of the decoded *Foreman* sequence that was subjected to burst loss of 20%. From these frame images we can clearly see (visually) how each of these different slice-coding schemes perform (subjective to a human eye). Thus perceptual slice-structured coding leads to better subjective video picture quality than other slice-coding schemes.

5) MobiStream and slice interleaving with ViCs: We conducted preliminary experiments to evaluate gains from using inter-frame and intra-frame slice interleaving over ViCs. Because of space constraints we briefly summarize the main result. The gains observed with intra-frame slice interleaving in the face of burst loss depends on the slice interleave depth. As an example, using virtual channels and an interleave depth of 5, our results show that the relative benefit in PSNR is in the 2.0-2.7 dB range at burst loss of 20% for the *Foreman* sequence.

Slice coding overheads: The overhead from slice-structured coding, especially in the case of perceptual slice-structured video coding, comes primarily from the broken in-picture prediction and from the slice-group header itself. In other words, the more number of slice groups in a video frame, the better is the error concealment but more will be the slice coding overheads too. Depending on the picture sizes, this overhead may be substantial and can sometimes reach 10% or more of the coded bits, however 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 (also called network adaptation layer, i.e., NAL unit in H.264), and, (ii) the overhead of each slice-group in a frame requires its own slice header, which is somewhere between 2 and 3 bytes for QCIF



(a–1: Allzero without ViCs) (b–1: Allzero with ViCs)
(a–2: Interleave without ViCs) (b–2: Interleave with ViCs)
(a–3: Checkerboard without ViCs) (b–3: Checkerboard with ViCs)
(a–4: Perceptual SS without ViCs) (b–4: Perceptual SS with ViCs)

Fig. 19. Shows 55^{th} frame of the *Foreman* video sequence using different slice-structured coding schemes. The left half of the figures shows 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.

size picture but somewhat bigger for larger picture sizes.

Furthermore, there is overhead associated with RTP packet headers when slice coding is enabled, otherwise it may not help decoder error concealment. The overhead for an additional RTP packet is roughly 40 bytes (combined that for IP+UDP+RTP header). Note that with RoHC (robust header compression) this may be reduced to 3 bytes or less.

A QCIF video frame consists of 9 macroblock lines. With the above assumptions in mind, the overhead from *interleaved* slice-coding scheme, for example, is $9 \times 4 + 3$ bytes or 39 bytes per frame. At a frame rate of 15 fps, this results in 4.6 Kbps overhead for an operation point where 32 to 64 Kbps streaming video bitrates can be common. Hence it is easy to see that the overhead for an interleaved slice coding scheme is roughly *less* than 10% of the total bitrate and is somewhat more for video content where in-picture prediction is beneficial and for I-frames.

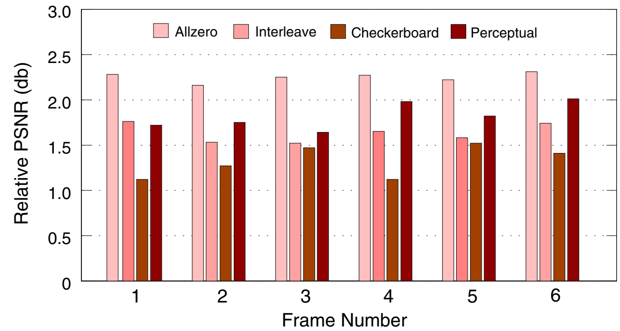


Fig. 20. Relative benefit in PSNR using virtual channels for different slice-based coding schemes at a burst loss of 20%. The different frames correspond to a GoP (ibpbpb) in the *Foreman* video sequence.

VI. Related Work

Prior research work has investigated issues related to video streaming in wired (e.g., [4]) and wireless environments (e.g., [5],[6] and [7]).

Our work in MobiStream significantly differs from all other prior works in several different 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 also propose *perceptual slice-structured video coding* and prioritization to protect visually more important video data than less important ones, and, 5) we provide fine-grained error-resilience using a range of available techniques over virtual channels.

In [6], 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 [8] also performed a study of low-latency streaming video over WLAN networks exploiting path diversity.

Van der Shaar et al. in [5] 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 [5] in that MobiSream exploits the link 'awareness' available from statistical packet loss distributions for virtual channels to enable efficient video streaming in WWANs. Related other studies like [9] investigated video streaming over wireless networks from a cross-layer design perspective.

In [10], Chesterfield et al. study (audio) streaming performance by exploiting diversity across multiple WWAN links and channels. They propose use of an application-layer packet repair strategy operating across multiple WWANs for streaming. This is done by disabling the reliability (retransmissions) and checksums of the lower layers, thus allowing corrupted portions

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the Proceedings IEEE Infocom.

of the received data be forwarded for possible use or repair by an application (this approach is similar to UDP-lite [7]).

MobiStream takes a different standpoint from that in [10] for video streaming. Instead of considering a WWAN link as either a hard-coded reliable link (link retransmissions set as default) or an unreliable link (link retransmissions and error checksums disabled), MobiStream proposes the co-existence of multiple *dynamic* channel instances with variable degrees of reliability and partial channel control that serve video applications 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 partitioned video content across such virtual channels.

In [11], 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 concern 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 [12] present a double feedback streaming agent for improving video streaming over UMTS WWAN links while Zheng et al. in [13] 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., [14],[15], [16] and [17]. Note that all of these works make use of RTP [3] 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 [18] and [19].

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) 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 content can be very effectively prioritized and flexibly scheduled 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 in these different lossy environments that we feel would ultimately lead to fewer and less annoying video artifacts.

References

[1] K. Guo, S. Rangarajan, M. Ali Siddiqui and S. Paul. Providing End-to-end QoS for Multimedia Applications in 3G Wireless Networks. In *Proceedings of the SPIE (ITCom'2003) Conference*, September 2003.

[2] H.264/AVC Software. `http://iphome.hhi.de/suehring/tml/`.

[3] H. Schulzrinne et. al. RTP: A Transport Protocol for Real-Time Applications. In *IETF RFC – 1889*, January 1996.

[4] 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.

[5] 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.

[6] 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.

[7] A. Singh, A. Konrad and A. Joseph. Performance Evaluation of UDP lite for cellular video. In *Proc. of ACM Nossdav*, 2001.

[8] 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.

[9] Y. Shan and A. Zakhor. Cross Layer Techniques for Adaptive Video Streaming over Wireless Networks. In *Proc. of IEEE ICME*, 2002.

[10] 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.

[11] J. Chakareski, S. Han and B. Girod. Layered Coding vs. Multiple Descriptions for Video Streaming over Multiple Paths. In *Proc. of ACM Multimedia*, 2003.

[12] 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.

[13] 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.

[14] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stoackhammer and T. Wedi. Video coding with H.264/AVC: Tools, Performance, and Complexity. *IEEE Circuits and Systems Magazine*, January 2004.

[15] 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.

[16] 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.

[17] 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.

[18] Q. Bi. A forward link performance study of 1xEV-DO system through field measurements. In *Bell-labs Technical Report*, March 2004.

[19] E. Esteves, M. I. Gurelli, and M. Fan. Performance of Fixed Wireless Access with CDMA2000 1xEV-DO. In *Proc. of IEEE VTC*, 2001.