H.263 Based Video Codec for Real-Time Visual Communications Over Wireless Radio Networks

Paramvir Bahl
Microsoft Corporation
Redmond, Washington, USA
bahl@microsoft.com

Imrich Chlamtac
University of Texas at Dallas
Dallas, Texas, USA
chlamtac@utdallas.edu

Abstract

With the emergence of the International Telecommunication Union's (ITU) H.263 video coding standard, real-time visual communications over low-bandwidth, low-error-rate wired telephone networks is now possible. Unfortunately, because of the nature of the algorithms employed within this compression standard and due to the inherent limitations of the radio environment, H.263 coded video performs poorly when employed over bandwidth-limited, error-prone wireless channels. In this paper we propose an intuitive, low-complexity, pre-compression scheme that improves the performance of H.263 coded video over radio channels. In our scheme video frames are spatially segmented into regions of different perceptual importance before being compressed independently with the H.263 encoder. This allows us to apply unequal error protection and prioritized transmission to achieve at least a minimum temporal resolution at the receiver. Additionally, with this technique, both spatial and temporal error propagation is limited and through intra-frame statistical multiplexing the reserved transmission bandwidth is utilized optimally. Simulation results demonstrate that in the presence of severe error conditions and severe bandwidth constraints, our modified H.263 video codec exhibits better error concealment, better temporal resolution, and better bandwidth utilization properties than the original H.263 video codec[1]

1. Introduction

Most current communications are carried out using source coders and channel coders that were designed independently of each other. The tradition of separating the source and channel coding tasks arises from the solid theoretical foundation of Shannon's celebrated *separation principle* of source and channel coding, which basically states that this separation is optimal. For example, in a point-to-point transmission using a known, time invariant channel, one can design the best possible channel coding method to approach channel capacity; i.e. achieve a rate R bits/second such that $R \leq C$ where C is the channel capacity in bits/second. Then the task of the source coder is simply to do the best job it can in compressing the input signal so that the compressed bit rate will match the rate of the channel. No scheme can do better theoretically than in this scenario.

There are, however, several reasons not to adhere blindly to the separation principle. For example, optimal codes such as the Huffman codes behave best when a two pass technique is applied to images, the first pass to establish probability of symbol occurrence and the second to code the data. In practice for video sequences this is never done as it is computationally too intensive and results in undesirable delay. Additionally, Shannon's work makes no assumptions on the error characteristics of the channel on which the data would traverse. Coding techniques that have been developed without regard to the channel characteristics can destroy the optimal codes in a manner that even with a few bits in error, the entire image signal is rendered useless. This is specially true for all of the current existing first generation video coding standards including MPEG-1, MPEG-2, H.261, and H.263 (coders that were designed without regard to the error characteristics of the channel) [1-3]. Another factor for considering joint source-channel coding, which is both a theoretical and practical one, has to do with optimizing the use of channel resources through statistical multiplexing.

Due to the large variations and unpredictability in the error characteristics of the radio channel, error resiliency in video applications via error detection, recovery, and concealment is a critical requirement. Error protection schemes that use sophisticated convolution or block codes may be employed to alleviate the error-induction problem but they aggravate bandwidth problems since several more bits have to be added to the video bit-stream. Similarly ARQ type re-transmission procedures while improving error recovery against burst errors aggravate latency (jitter) problems. While it is true that we cannot completely eliminate this overhead, we can reduce the dependence on such techniques while improving the final quality of the displayed video by letting the source encoder do its part.

[1] This research has been partially funded by ARO grants number DAAH04-95-1-0443 and DAAH04-96-1-0308

In this paper we propose a simple and intuitive image pre-compression step that segments video frames into regions of unequal importance before coding them independently. There are several motivating reasons for doing this: (1) It allows the transmitter to apply unequal error protection according to the importance of the region. This follows from the observation that image sections have unequal perceptual sensitivity to channel errors. Thus the decoded image quality can be improved by better protecting regions which have more impact on the image quality; (2) When bandwidth is limited, and when the bit error rates are high, it is possible to dynamically adjust the order and transmission priority of individual regions. The encoder thus works in harmony with the underlying network protocol, re-ordering the compressed bits before transmission in a manner that ensures optimum use of the available bandwidth resources; (3) It allows for improved temporal resolution at the receiver. The non-received (or corrupt) regions of the current frame are substituted by the regions from some previous frame and the full frame is reconstructed. This results in an improved perceived frame rate at the decoder; (4) It limits error spreading in the spatial and the temporal domain and (5) It reduces coding delays as transmission can begin as soon as the first region is compressed. Region segmentation can be employed as a pre-compression scheme relatively independent of the compression algorithm. In its degenerate form it adds minimally to the complexity to the overall process.

2. ITU Recommendation H.263 [3]

The ITU Recommendation H.263 is a motion-compensated, Discrete Cosine Transform-based inter-frame video coding standard developed primarily for very low bit-rate video communication (below 64 kilobits per second) over narrowband channels. While sharing considerable similarities with ITU's H.261 video coding standard [1], H.263 includes several improvements and is targeted for the Plain Old Telephone System (POTS) with modems that have the V.32 and the V.34 modem technologies. Compared to H.261, the number of available picture formats have been increased and the motion-compensation algorithm has been improved considerably. Some differences and highlights are: H.263 supports *unrestricted motion vector mode*, that is, unlike H.261, the block being referenced doesn't have to be completely inside the picture anymore. An arbitrary number of pixels from the block may be outside the picture, in which case the closest edge pixels are used. A second new feature is the *advanced picture mode*, where instead of using one vector pointing to a *16 by 16* block, prediction quality is improved by allowing the use of four motion vectors pointing to *8 by 8* blocks. Another improvement over H.261 is that in H.263 the resolution of the displacement vector has been increased to half-pixel. To calculate the referenced sub-pixels, the value of the surrounding pixels is interpolated. Finally, a new type of frame called *PB-frame* has been added to the H.261's I and P frame types. A PB-frame consists of a normal P-frame and a new bi-directional prediction frame. The idea is to be able to increase the temporal resolution at minimal expense for video playback. New and more efficient entropy encoding tables are also included in the standard. In general, improvements added in the H.263 codec provide good coding gains (reduced output bit rate and improved image quality) at the expense of slightly increased computationally complexity.

3. Effect of the Radio Environment on the Quality of Transmitted Video

The radio environment is characterized as having a limited spectrum (i.e. inadequate bandwidth) with high bit error rates (BER). These errors manifest themselves in the form of isolated random errors and in clusters in the form of burst errors. H.263 was originally developed for visual communications over POTS and for PSTN (Public Switched Telephone Network) multimedia terminals. PSTN is characterized by low delay, an error rate that is typically better than 10^{-6} and channel conditions that remain constant with time. For such an environment H.263 works reasonably well. However, wireless radio networks suffer from higher bit error rates (typically 10^{-2}) with channel characteristics that are time varying [4]. In such an environment we studied the performance of H.263 video and found that it performs poorly.

3.1 Simulation Methodology

Our transmitter model included a video compressor, a forward error correcting encoder (BCH encoder), a burst error correcting interleaver and a CRC error detecting coder (Figure 1). All components were implemented in software [5]. The test video sequences were compressed off-line while decompression was done in real-time. The video sequences used in our experiments were identical to the ones that have been adopted within ITU's Study Group-15 (SG-15) video compression committee. Each information block containing the compressed bitstreams was error protected using a invertible Read-Solomon (RS) code and a symbol based interleaver.

The H.263 compressed video data was fragmented into blocks of 48 octets, packaged and transmitted (stored) in packets of 53 octets. 5 octets were used for header information including 2 octets for CRC and 3 octets for miscellaneous information such as connection number, packet number, priority etc. (The packet size was chosen to be identical to the ATM cell size). To provide error protection, an interleaved shortened RS code over $GF(2^8)$ was used. Specifically, a distance-5 RS code with roots $1, \alpha, \alpha^2, \alpha^3$, where α is the primitive root of:

$$g(x) = x^4 + \alpha^{219} x^3 + \alpha^{56} x^2 + \alpha^{222} x + \alpha^6 \qquad (1)$$

was used. Encoding was done in a systematic manner; decoding was done using Berlekamp's iterative algorithm [6]. The interleaving technique used was a simple technique for turning burst errors to random errors. The RS code enabled these and other random octets and bit errors to be corrected readily.

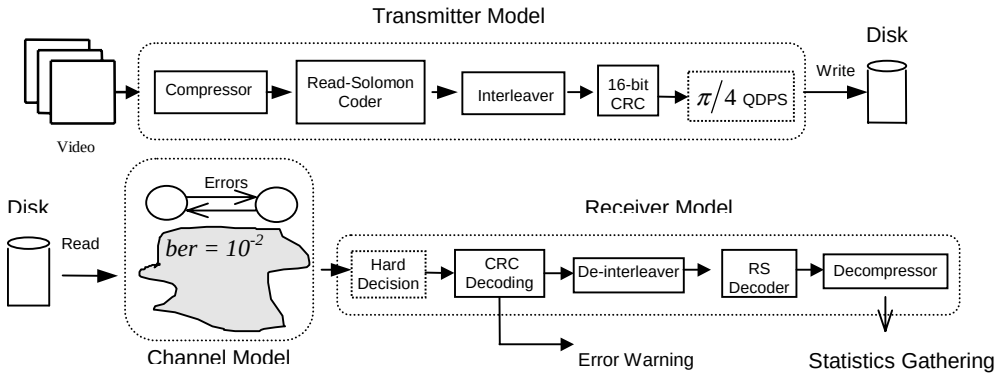

Figure 1: Simulation architecture for studying the performance of video over radio channels

The length of the code (n) was adjusted depending on the bit error rate of the radio channel being simulated[2].

Errors in the radio environments occur both as random bit errors and as clusters of errors in long and short bursts. These errors are generally attributed to phenomena such as multipath fading, shadowing, ground wave path loss, noise, and interference from other users. For the purposes of simulation, errors during transmission over a radio channel were modeled as a 2-state markov process with the two states representing the Burst Error State (BE) and the Random Error State (RE). In each state bit errors were distributed according to the Poisson distribution with a mean bit error rate of λ_{BE} and λ_{RE} respectively. The transition from the Random Error to the Burst Error state and from Burst Error to the Random Error state were also Poisson distributed with a mean transition rate of λ_{RB}/sec and λ_{BR}/sec respectively. Figure 2 illustrates the channel model used in our simulation.

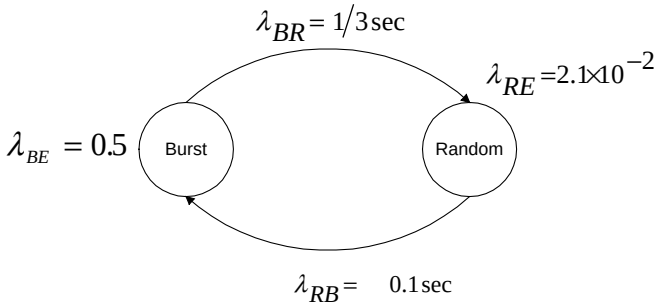

Figure 2: A 2-state model for errors

[2] More sophisticated error protection schemes are currently being researched including ones that include concatenation of RS and RCPC codes to provide even better error protection. The reader is referred to [6] and [7] for examples of such techniques.

3.2 Effect of Errors on H.263 Video

Based on our simulations and some recently reported results [7], the effects of transmission errors on H.263 video can be summed up as follows:

- Errors in packet headers or motion vectors can cause major damage
- A single bit error can destroy a major part of a video-frame in the spatial domain due to improper decoding of variable length codes
- Errors propagate among P and PB frame in the temporal domain, I-Pictures (independently coded pictures) stop error propagation in the temporal domain.
- Picture and Group of Blocks (GOB) headers stop error propagation in spatial domain.

In general, the impact of bit errors on video quality depends on the spatial and temporal location of the error. For inter-frame statistical redundancy removal coding techniques (like H.263) temporal and spatial error propagation problems are inherent and lead to reduced picture quality and reduced temporal resolution at the receiver. Specifically, entropy (or Huffman) coding causes spatial error spreading problems whereas motion compensation causes temporal error propagation problems. Protecting the entire bitstream is expensive because of bandwidth limitations and delay constraints. However, since we cannot entirely do away with error protection, a better approach is one where only the most critical information is error protected while the rest is sent as is.

4. The Region-Segmentation Approach

Applying simple heuristics that exploit prior knowledge of the characteristics and the geometry of the image contents within a video sequence, it is possible to improve the quality of the decoded video substantially [8]. Specifically, knowing that conferencing video is generally restricted to limited subject and camera movement and is usually characterized as head and shoulder scenes, video frames can easily be partitioned into regions of unequal importance. This is the basis of the technique described in this section.

In order to restrict the spreading of transmission errors in video sequences and to improve the temporal resolution, each video frame within a sequence is divided into *N* sub-frames (regions), each of which is coded independently and transmitted separately. Transmission errors due to channel imperfections can cause corruption in some of these sub-frames (regions) rendering them un-decodable or bandwidth unavailability can cause some of these not to reach the decoder. If this happens, at the receiver the complete frame is reconstructed from a combination of current and previous sub-frames that were received correctly. let S_{ij} represent the sub-frame j in frame i, then if all sub-frames are received error-free, the complete frame i is formed as follows:

$$S_i = \sum_{j=1}^{N} S_{ij} \qquad (2)$$

In the case when some of the S_{ij} are incorrectly received, S_i is formed by using the last corresponding j^{th} sub-frame that was received correctly. When used to reconstruct the image frame in this manner the temporal difference between the current sub-frame and the previous correctly received corresponding sub-frame dictates how good or bad the final picture looks. When this difference is large, visual quality is impaired by the tearing effect, however when the previous good sub-frame is from a preceding sequential frame, the quality generally tends to be acceptable. Infact, the tearing effect can be reduced considerably by having the receiver demand from the transmitter the immediate transmission of the sub-frames that it had to substitute in order to reconstruct the current frame. The transmitter would then switch the transmission priority of the requested regions to ensure that the requested sub-frames definitely reach the receiver. With such a scheme, the difference between the current sub-frame and the ones stored in the receiver's *Region Store* is never be too much and the tearing effect is mitigated. Figure 3 illustrates the concept for a simple horizontal partitioning scheme. The sub-frames stored in the receiver's Region Store, in general do not belong to the same image.

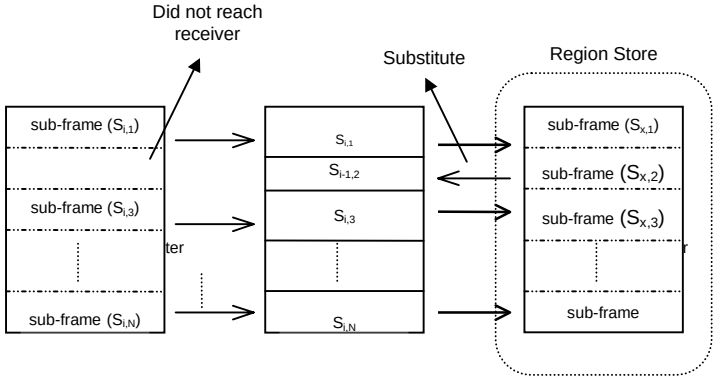

Figure 3: Reconstruction of a video frame

4.1 System Model

In the *region-segmentation* model the sub-frames (regions) are created before the compression process. To avoid complexity, the segmentation pattern is pre-determined and each image is broken up into five regions. In terms of implementation, this step is folded into the compression algorithm. It should be noted that sophisticated methods such as morphological filtering, wavelet and pyramidal decomposition can also be employed to determine different regions and subbands, but for video conferencing applications that are characterized by limited head and shoulder movements, and low subject and camera movement it is generally sufficient to segment the image according to simple heuristics in preference to increased complexity.

Once the image has been segmented into its various regions, an adaptive video codec compresses each region separately. In general, depending on the importance of the region, each region may be compressed differently. For example, the region designated the primary region may be compressed using an intra-frame compression algorithm such as the JPEG algorithm [9] while the other regions are compressed using a motion compensation, inter-frame algorithms. For our study, each region was compressed using the H.263 video codec. As soon as a region is compressed, the bitstream is error protected and interleaved. Once again, depending on the importance of the region, different levels of error protection and interleaving may be applied to the different regions. The final bitstream is then packetized, appropriate header information is added, and sent to the modulator. The decompression process is the reverse of the compression process. The demodulated signal is deinterleaved, detected errors are corrected (up to the limit of the error correcting code) and the regions are decompressed. The full frame is reconstructed from regions that were received correctly and by substituting the missing or incorrectly received regions available in the receiver's Region-Store. The Region-Store is updated to include the latest correctly received regions and the reconstructed image is sent to the application.

For our experiments, we segmented each frame of the test video sequence into five independent regions. Each region was then compressed using ITU's H.263 video codec. The H.263 bitstream syntax provided in the standard was modified to accommodate region-based coding. Specifically, regions were indicated through the PTYPE field in the picture header and the GOB layer was replaced by a "Region Layer" (BIR or Blocks in Region). Data for each region consisted of a region header followed by macroblocks, forming rectangular blocks. The region header had fields to indicate the position of the region within the frame; the type of coding employed, and the length or the number of bits in the region. Since the decoder already had the segmentation geometry, looking at the Region ID field it was able to figure out where to place the region within the frame. The Region ID was also used by the network layer to determine the degree of error protection to be employed and the transmission priority for each region.

4.2 Advantages of Region-Segmentation

All of the compression algorithms used in currently popular video coding standards (MPEG-1, MPEG-2, H.261, and H.263) can utilize this technique for wireless video communications.

4.2.1 Bounded Spatial and Temporal Error Propagation

One of major problem with the current video coding standards is that errors in one part of the image propagate both spatially and temporally. If the error corrector (e.g. and RS decoder or a ½ rate RCPC decoder) is unable to correct these errors, and the buffer size is limited, the GOB has to be re-transmitted (e.g. using ARQ techniques) or the video-frame has to be dropped. With the region-segmentation approach video-frames are broken up into independent regions and motion estimation and compensation is applied on a region basis as opposed to frame basis. Errors in one region do not propagate to other regions and consequently their effect is limited both spatially and temporally. If too many un-corrected errors occur in a region, the entire region is substituted by a previous correctly received region or when that is not available, a request for re-transmission is sent to the transmitter. Re-transmitting a region is faster than re-transmitting the entire frame and requires less bandwidth than a entire frame.

To illustrate the effect of limited spatial error propagation, we consider a video bitstream that has been corrupted with errors not caught by the error detector. For example, when Huffman codes in a H.263 bitstream are corrupted in a manner that changes them into different valid codes. Figure 5 illustrates this effect on the *Miss America* sequence. Without segmentation almost the entire image is rendered useless, but with segmentation only the region(s) corrupted by the error is effected while the rest of the video frame still looks acceptable.

4.2.2 Improved Temporal Resolution Under Severe Bandwidth Constraints

Another difficulty with current video coding standard's has to do with video frames not being transmitted when the available bandwidth is less than the amount required to transmit a full compressed frame. (For on-going connections, dynamic changes in bandwidth are possible if the total demand changes). While it is true that for non-real time video the amount of time needed to transmit a single video frame is not critical, in real-time video if the frame is not transmitted within a certain delay bound it is rendered useless and is generally dropped at the transmitter. In many instances it can happen that only part of the video frame is transmitted within the delay bound. If this happens, both the quality of the video and the system suffer. The quality suffers because the decoder has to skip the frame and the resulting video sequence looks choppy, looses synchronization with the accompanying audio and in general becomes unacceptable. The system suffers because valuable bandwidth was used up in transmitting data that never got used.

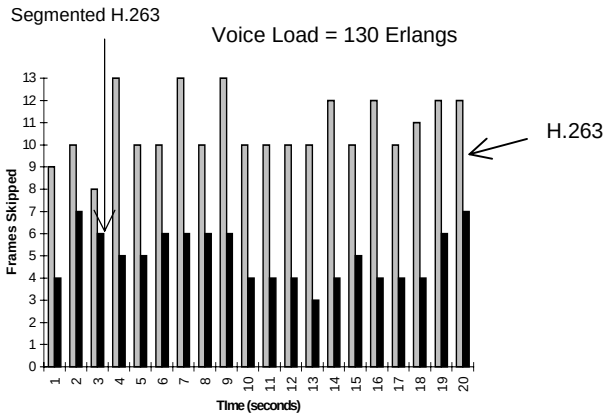

Figure 4: Number of the frames skipped with increase in load

In Figure 4, we have plotted the number of video-frames skipped at the receiver with time as the aggregate voice load in the system is increased. Clearly, the region-segmented H.263 video coder performs better than the one without it. The coded video bitstream in this case was the *Miss America* sequence coded at QCIF resolution with a bitrate of 23 kbits/second. The original video, available at 25 frames/second, was compressed and transmitted at 15 frames/second. The average PSNR for the luminance component was 31.2 dB and there were 75 frames in the sequence (used in loopback mode).

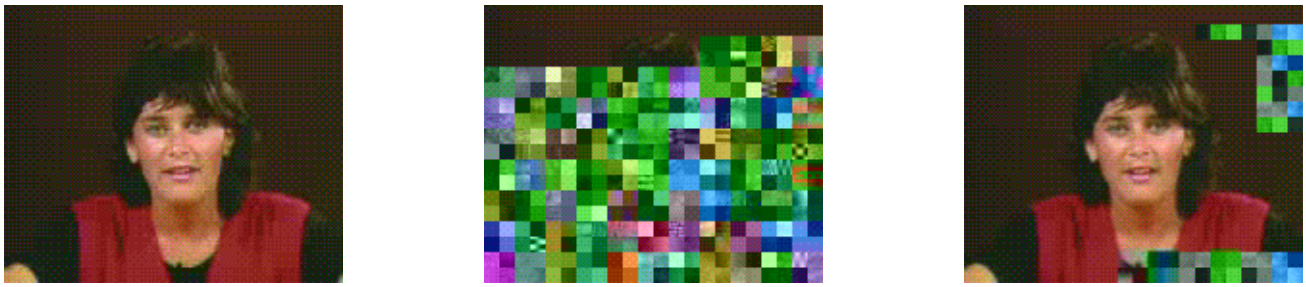

Figure 5: Example of error propagation containment with region segmentation

The WMAC protocol was a TDM-based protocol with built in support for voice, data, and video [11]. Time-frames were divided into 120 time-slots and traffic in one slot represents a load of one erlang. Voice was modeled as a 2-state on-off process with a CBR of 13 kbits/second when operating in the on-state (modeled after the RPE-LTP codec used in GSM). Errors were ignored for this simulation.

Figure 6 provided a long-term perspective on the performance of the region segmented H.263 video codec with differing loads. The PSNR for the H.263 video is higher than that for the segmented H.263 video. This can be explained on the basis that the PSNR for segmented H.263 is calculated from all video frames displayed, which include video frames that were reconstructed from regions belonging to other frames, whereas the only frames used in the PSNR of H.263 video are the ones that were received correctly.

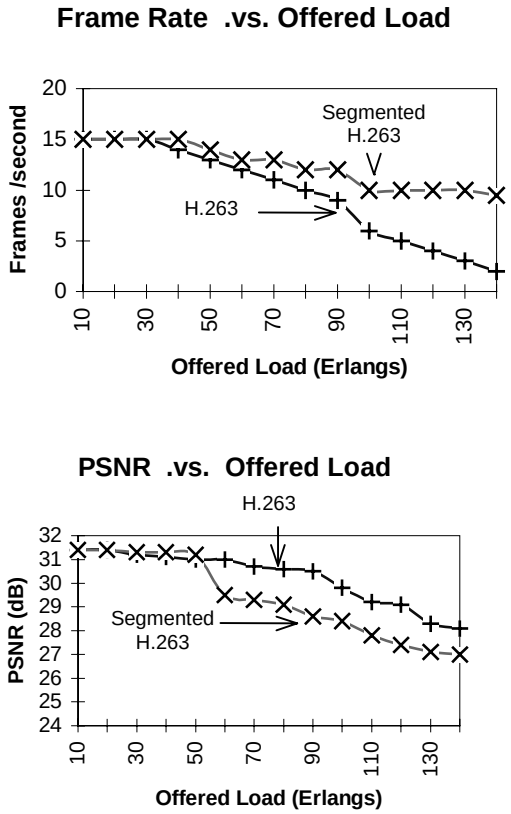

Figure 6: Comparison of temporal resolution with changing loads

4.2.3 Improved Temporal Resolution Under Severe Error Conditions

Temporal resolution for segmented H.263 is also improved when the BER is high or when the environment is characterized by frequent shadowing or multipath fading. When infected with serious errors that cannot be corrected, video data is lost, which in effect is same as not having enough bandwidth to transmit. As has been explained previously, the video-frame is constructed even when all regions haven't been received correctly. Using pieces of the regions from some previously received frame, a new frame can be constructed for display even when the amount of data received is incomplete.

Figure 7 illustrates the performance of the region-segmented H.263 video compared to the normal H.263 video codec under differing error conditions. In this simulation the average frame rate is plotted for the *Miss America* sequence.

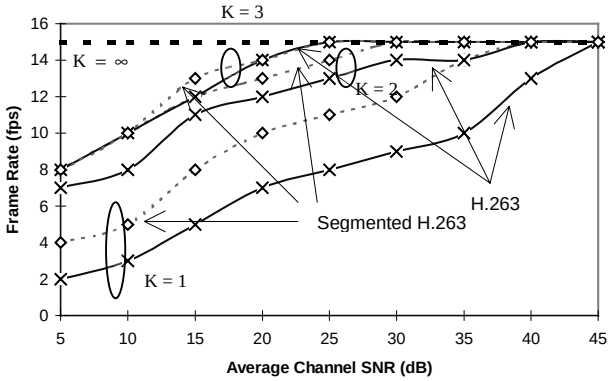

Figure 7: Comparison of temporal resolution with changing error characteristics

The parameter K in Figure 7 represents the number of re-transmissions allowed. As expected large values of K, result in a higher decoded frame rate. Looking at the figure, it is clear that the region segmented H.263 video performs better for different values of K.

4.2.4 Efficient Bandwidth Utilization

The region-segmentation approaches makes video codecs network friendly. To illustrate this, statistical characterization of similar video conferencing sequences is carried out [11] and the peak number of bits for the most important region in an image is determined. Then the amount of bandwidth reserved at connection establishment time is equal to the peak number of bits per second for the primary region. It is likely that most of the time the compressor will produce bits far below this peak number. To avoid under-utilizing and wasting allocated bandwidth, we introduce the notion of statistical multiplexing within a frame [8]. The bandwidth left over after the primary region has been transmitted is used for transmitting the remaining regions. Also packets whose re-transmission has been requested by the receiver (as in ARQ schemes) can be sent using the left over bandwidth that was reserved for primary region. In essence the idea is to combine statistical multiplexing at the system level with statistical multiplexing at the connection-level to achieve optimum bandwidth utilization. Bandwidth reservation or resource reservation is necessary if QoS is to be guaranteed to video connections.

To illustrate this concept we used the QCIF version of *Miss America* sequence. Each image in the video sequence was pre-processed and segmented into five regions and then each region was compressed using the H.263 compression algorithm. Bandwidth was reserved for the center region and was equal to its peak rate of 24 kbps. The average bit rate for this region was 8 kbps. The second, third, and forth regions had an average bit rate of 3, 2.5, and 2.8 kbps respectively. The main region was always transmitted and the left-over bandwidth was used to transmit the rest of the regions. The remaining bits of the compressed frame were transmitted using unreserved bandwidth. It should be noted, and as pointed out previously after steady state has been reached the video decoder is able to reconstruct the full frame as along as the main subband or main region is received. Figure 8 contains plots of the difference in the amount bandwidth reserved and the amount used.

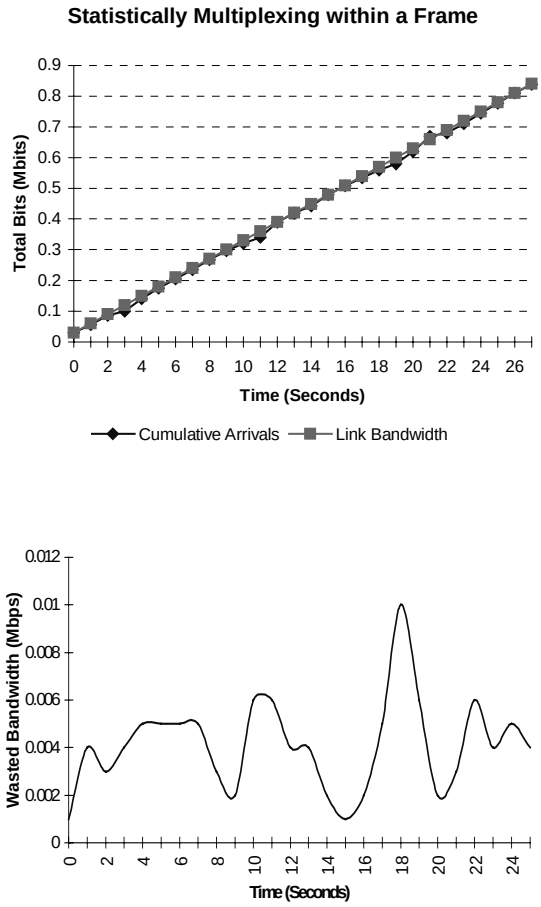

Figure 8: Intra-frame statistical multiplexing for region-segmented video codec

In [10] we have proposed and analyzed a novel bandwidth partitioning algorithm based on minimizing the maximum call blocking probability while working with the region-segmentation technique and intra-frame statistical multiplexing technique to provide guaranteed quality of service for real-time video connections in wireless radio networks.

5. Conclusions

Region-segmentation when used as a pre-processing approach achieves joint source/channel coding in the delay as well as loss and corruption dimensions. By allowing region-by-region reconstruction, the perceptual-delay of the video becomes less than the worst-case network delay, and the traffic capacity is increased because of the relaxing of the worst-case delay. The region bitstream model allows the application to independently control delay, loss, and corruption so that the part which requires rapid response and/or lower reliability is transmitted over appropriate low-delay/low-loss channels.

References

[1] *ITU Recommendation H.261*, "Video Codec for Audiovisual Services at p x 64 kbits/s," CDM XV-R 37-E, International Telegraph and Telephone Consultative Committee (1990).

[2] *ISO/IEC Standard 11172-2*, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at Upto about 1.5Mbits/s" (1993).

[3] ITU-T, SG15, *ITU Recommendation H.263*, "Video Coding for Narrow Telecommunication channels at < 64 kbit/s, " (April 1995)

[4] A. Puri, A. R. Reibman, R. L. Schmidt, and B. G. Haskell, "Robustness Considerations in ISO MPEG-4 and ITU-T Mobile Video Standards," *Proceedings of the 3rd International Workshop on Mobile Multimedia Communications*, Princeton, New Jersey (Sept. 1996)

[5] P. Bahl, P. Gauthier, and R. Ulichney, "Software-Only Compression, Rendering, and Playback of Digital Video," *Digital Technical Journal*, Vol. 7, No. 4 (March 1995): 52-75

[6] S. Lin and D. J. Costello, Error Control Coding: Fundamentals and Applications, Prentice-Hall, Inc, 1983

[7] H. Liu, and M. E. Zarki, "Performance of Video Transport over Wireless Networks Using Hybrid ARQ," *Proceedings of the 11th Annual IEEE Computer Communications Workshop*, Reston, Virginia, USA (Sept. 1996)

[8] P. Bahl and I. Chlamtac, "Strategies for Transmission of Compressed Video Over Error-Prone Radio Channels", *Proceedings of the 3rd International Workshop on Mobile Multimedia Communications*, Princeton, New Jersey (Sept. 1996)

[9] ISO/IEC IS 10918-1: "Information Technology - Digital Compression and Coding of Continuous Tone Still Images, Part 1: Requirements and Guidelines," (1994).

[10] P. Bahl, I. Chlamtac and A. Faragó, "Optimal Resource Utilization in Wireless Multimedia Networks," *Proceedings of the IEEE Conference on Communications*, Montreal, Canada (June 1997)

[11] P. Bahl, "Real-Time Visual Communications Over Narrowband Wireless Radio Networks," *Ph. D. Thesis*, University of Massachusetts, (1997)