Compressed-Sensing-Enabled Video Streaming for Wireless Multimedia Sensor Networks

Scott Pudlewski, *Student Member*, *IEEE*, Tommaso Melodia, *Member*, *IEEE*, and Arvind Prasanna

Abstract—This paper presents the design of a networked system for joint compression, rate control and error correction of video over resource-constrained embedded devices based on the theory of Compressed Sensing (CS). The objective of this work is to design a cross-layer system that jointly controls the video encoding rate, the transmission rate, and the channel coding rate to maximize the received video quality. First, compressed sensing-based video encoding for transmission over Wireless Multimedia Sensor Networks (WMSNs) is studied. It is shown that compressed sensing can overcome many of the current problems of video over WMSNs, primarily encoder complexity and low resiliency to channel errors. A rate controller is then developed with the objective of maintaining fairness among different videos while maximizing the received video quality. It is shown that the rate of Compressed Sensed Video (CSV) can be predictably controlled by varying only the compressed sensing sampling rate. It is then shown that the developed rate controller can be interpreted as the iterative solution to a convex optimization problem representing the optimization of the rate allocation across the network. The error resiliency properties of compressed sensed images and videos are then studied, and an optimal error detection and correction scheme is presented for video transmission over lossy channels. Finally, the entire system is evaluated through simulation and test bed evaluation. The rate controller is shown to outperform existing TCP-friendly rate control schemes in terms of both fairness and received video quality. The test bed results show that the rates converge to stable values in real channels.

Index Terms—Compressed sensing, optimization, multimedia content, congestion control, sensor networks.

1 Introduction

Wireless Multimedia Sensor Networks (WMSNs) [2], [3] are self-organizing systems of embedded devices deployed to retrieve, distributively process in real-time, store, correlate, and fuse multimedia streams originated from heterogeneous sources [4]. WMSNs are enablers for new applications including video surveillance, storage, and subsequent retrieval of potentially relevant activities, and person locator services.

In recent years, there has been intense research and considerable progress in solving numerous wireless sensor networking challenges. However, the key problem of enabling real-time quality-aware video streaming in large-scale multihop wireless networks of embedded devices is still open and largely unexplored. There are two key shortcomings in systems based on sending predictively encoded video (e.g., MPEG-4 Part 2, H.264/AVC [5], [6], [7], H.264/SVC [8]) through a layered wireless communication protocol stack, i.e., *encoder complexity* and *low resiliency to channel errors* [9].

- **Encoder complexity.** Predictive encoding requires complex processing algorithms, which lead to high energy consumption. New video encoding paradigms are therefore needed to reverse the traditional balance of complex encoder and simple decoder, which is unsuited for embedded video sensors. Recently developed *distributed video coding* [10] algorithms (aka Wyner- Ziv coding [11]) exploit the source statistics at the decoder, thus shifting the complexity to the decoder. While promising [2], most practical Wyner-Ziv codecs require end-to-end feedback from the decoder [12], which introduces additional overhead and delay. Furthermore, gains demonstrated by practical distributed video codecs are limited to 2-5 dBs PSNR [13], [12]. Distributed video encoders that do not require end-to-end feedback have been recently proposed [14], but at the expense of a further reduction in performance. In addition, all of these techniques require that the encoder has access to the *entire video frame* (or even multiple frames) before encoding the video.
- **Limited resiliency to channel errors**. In existing layered protocol stacks based on the IEEE 802.11 and 802.15.4 standards, frames are split into multiple packets. If even a single bit is flipped due to channel errors, after a cyclic redundancy check, the entire packet is dropped at a final or intermediate receiver.[1] This can lead to the video decoder being unable to decode an independently coded (I) frame, thus leading to loss of the entire sequence of video frames. Instead, ideally, when one bit is in error, the effect on the reconstructed video should be unperceivable, with minimal overhead. In addition, the perceived

• *The authors are with the Department of Electrical Engineering, State University of New York (SUNY), Buffalo, New York 14260. E-mail: {smp25, ap92}@buffalo.edu, tmelodia@eng.buffalo.edu.*

Manuscript received 10 Nov. 2010; revised 6 Apr. 2011; accepted 22 Apr. 2011; published online 19 Apr. 2012.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2010-11-0516. Digital Object Identifier no. 10.1109/TMC.2011.175.

1. No Forward Error Correction (FEC) is used in either IEEE 802.11 or 802.15.4, and hence a faulty bit corrupts the whole packet.

1536-1233/12/$31.00 © 2012 IEEE Published by the IEEE CS, CASS, ComSoc, IES, & SPS

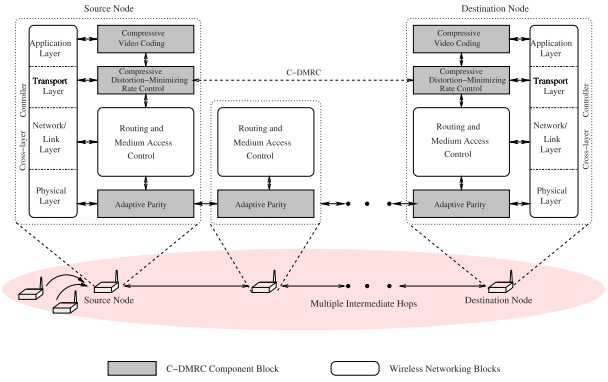

Fig. 1. Architecture of the C-DMRC system.

video quality should gracefully and proportionally degrade with decreasing channel quality.

In this paper, we show that a new cross-layer optimized wireless system based on the recently proposed Compressed Sensing (CS) paradigm [15], [16], [17], [18] can offer a convincing solution to the aforementioned problems. Compressed sensing (aka "compressive sampling") is a new paradigm that allows the faithful recovery of signals from $M \ll N$ measurements, where N is the number of samples required for the Nyquist sampling. Hence, CS can offer an alternative to traditional video encoders by enabling imaging systems that sense and compress data simultaneously *at very low computational complexity for the encoder*. Image coding and decoding based on CS has recently been explored [19], [20]. So-called single-pixel cameras that can operate efficiently across a much broader spectral range (including infrared) than conventional silicon-based cameras have also been proposed [21]. However, transmission of CS images and video streaming in wireless networks, and their statistical traffic characterization, are substantially unexplored.

For this reason, we introduce the Compressive Distortion-Minimizing Rate Control (C-DMRC), a new distributed cross-layer control algorithm that jointly regulates the CS sampling rate, the data rate injected in the network, and the rate of a simple parity-based channel encoder to maximize the received video quality over a multihop wireless network with lossy links. The cross-layer architecture of our proposed integrated congestion control and video transmission scheme is shown in Fig. 1. By jointly controlling the compressive video coding at the application layer, the rate at the transport layer, and the adaptive parity at the physical layer, we leverage information at all three layers to develop an integrated congestion-avoiding and distortion-minimizing system. Our work makes the following contributions:

- **Video transmission using compressed sensing.** We develop a video encoder based on compressed sensing. We show that, by using the difference between the CS samples of two frames, we can capture and compress the frames based on the temporal correlation at low complexity without using motion vectors.
- **Distortion-based rate control.** C-DMRC leverages the *estimated received video quality* as the basis of the rate control decision. The transmitting node controls the quality of the transmitted video directly. Since the compression of the video is linearly dependent on the video quality, this effectively controls the data rate. By controlling congestion in this way, fairness in the quality of the received videos is maintained even over videos that have very different compression ratios.
- **Rate change aggressiveness based on video quality.** With the proposed controller, nodes adapt the *rate of change* of their transmitted video quality based on an estimate of the impact that a change in the transmission rate will have on the received video quality. The rate controller uses the information about the estimated received video quality *directly* in the rate control decision. If the sending node estimates that the received video quality is high, and Round Trip Time (RTT) measurements indicate that current network congestion condition would allow a rate increase, the node will increase the rate less aggressively than a node estimating lower video quality and the same round trip time. Conversely, if a node is sending low-quality video, it will gracefully decrease its data rate, even if the RTT indicates a congested network. This is obtained by basing the rate control decision on the *marginal distortion factor*, i.e., a measure of the effect of a rate change on video distortion.
- **Optimality of rate control algorithm.** We finally show that the proposed rate control algorithm can be interpreted as an iterative gradient-descent-based solution to the optimal rate allocation problem (i.e., finding the rates that maximize the sum of video qualities).

To evaluate the system presented in this paper, the video quality is measured as it is perceived at the receiving node. For most measurements and simulations, Structural Similarity (SSIM) [22] is used to evaluate the quality. The SSIM index is preferred to the more widespread Peak Signal to Noise Ratio (PSNR), which has been recently shown to be inconsistent with human eye perception [22], [23], [24].[2]

The remainder of this paper is structured as follows: In Section 2, we discuss related work. In Section 3, we introduce the C-DMRC system architecture. In Section 4, we describe the proposed video encoder based on compressed sensing (CSV). In Section 5, we introduce the rate control system. Section 6 discusses channel coding issues.

2. SSIM considers three different aspects to determine the similarity between two images. If one image is considered the original, then the measure can be viewed as the relative quality of the second image. The SSIM index first calculates the luminance difference between the two images. Then, it subtracts the luminance components out and measures the contrast difference between the two images. Finally, the contrast is divided out and the structural difference is measured as the correlation between the two remaining signals. These three measurements are then combined to result in the overall SSIM index, which is a normalized value between 0 and 1. SSIM is a more accurate measurement of error because the human visual system perceives structural errors in the image more than others. For example, changes in contrast or luminance, although mathematically significant, are very difficult to discern for the human eye. Structural differences such as blurring, however, are very noticeable. SSIM is able to weight these structural differences better to create a measurement closer to what is visually noticeable than traditional measures of image similarity such as Mean Squared Error (MSE) or PSNR. These results have been shown for images [22] and for videos [23], [24] in the LIVE database.

The performance results are presented in Section 7. In Section 8, we show how the proposed rate control subsystem can be interpreted as the solution algorithm to a rate optimization problem. Finally, in Section 9, we draw the main conclusions and discuss future work.

2 Related Work

The most common rate control scheme is the well-known Transmission Control Protocol (TCP) [25], [26]. Because of the additive increase/multiplicative-decrease algorithm used in TCP, the variation in the rate determined by TCP can be very distracting for an end user, resulting in poor end user perception of the video quality [27]. In addition, TCP assumes that the main cause of packet loss is congestion [28], and thus misinterprets losses caused by channel errors as signs of congestion. These considerations have led to a number of equation-based rate control schemes, which analytically regulate the transmission rate of a node based on measured parameters such as the number of lost packets and the round trip time of the data packets. Two examples of this are the TCP-Friendly Rate Control [29], [28], which uses the throughput equation of TCP Reno [25], and the Analytical Rate Control (ARC) [30], [31]. Both of these schemes attempt to determine a source rate that is fair to TCP streams. However, in a WMSN, priority must be given to the delay-sensitive flows at the expense of other delay-tolerant data. Therefore, both TCP and ARC result in a transmission rate that is more conservative than the optimal rate. For this reason, in an effort to optimize resource utilization in resource-constrained WMSNs, our scheme does not take TCP fairness into account.

Recent work has investigated the effects of packet loss and compression on video quality. In [32], Stuhlmuller et al. analyze the video distortion over lossy channels of MPEG-encoded video with both interframe coding and intraframe coding. A factor β is defined as the percentage of frames that are an intraframe, or I frame, i.e., a frame that is independently coded. The authors then derive the value β that minimizes distortion at the receiver. In [32], Stuhlmuller et al. investigate optimal strategies to transmit video with minimal distortion. However, the authors assume that I frames are received correctly, and that the only loss is caused by the intercoded frames. In this paper, we assume that any packet can be lost, and rely on properties of CS video and on an adaptive parity mechanism to combat channel impairments and increase the received video quality.

Quality of Service (QoS) for video over the Internet has been studied in [33] and [34]. Both of these works deal with QoS of video over the Internet using TCP or a TCP-Friendly rate controller. In general, a WMSN will not be directly connected to the Internet, so requiring fairness to TCP may result in significant underestimation of the achievable video quality.

Several recent papers take a preliminary look at video encoding using compressed sensing [35], [36], [37]. Our work is different in the following sense: 1) we only use information that can be obtained from a single-pixel camera [21] and do not use the original image in the encoding process at the transmitter. Hence, C-DMRC is compatible with direct detection of infrared or terahertz wavelength images, along with the ability to compress images during the detection process, avoiding the need to store the entire image before it is compressed; 2) we look at the problem from a networking perspective, and consider the effect of joint rate control at the transport layer, video encoding, and channel coding to design an integrated system that maximizes the quality of wirelessly transmitted CS video.

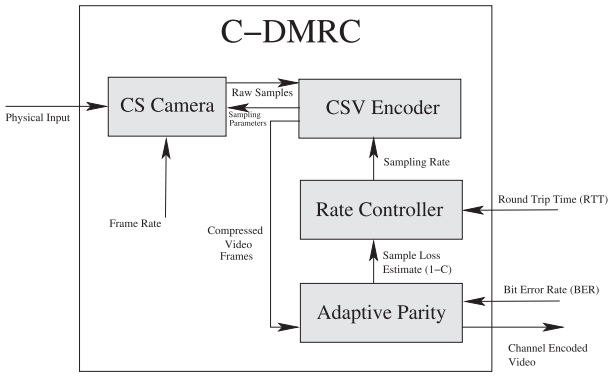

Fig. 2. Architecture of the C-DMRC video rate control system.

Finally, video encoding algorithms based on compressed sensing are also presented in [38] and further refined in [39], [40], and [41]. In these works, the authors present a Distributed Compressive Video Sensing (DCVS) system which, like the encoder presented in this paper, does not require the source node to have access to the raw video data. The rate allocation is examined in [39]. However, unlike [39], the rate allocation scheme presented in this paper is intended to maximize the rate over multiple videos sharing a network, while [39] looks to determine the optimal rate of a single video session based on the sparsity of that raw video.

3 System Architecture

In this section, we describe the overall architecture of the Compressive Distortion-Minimizing Rate Controller (C-DMRC). The system takes a sequence of images at a user-defined number of frames per second and wirelessly transmits video encoded using compressed sensing. The end-to-end round trip time is measured to perform congestion control for the video within the network, and the Bit Error Rate (BER) is measured/estimated to provide protection against channel losses. The system combines functionalities of the application layer, the transport layer and the physical layer to deliver video through a multihop wireless network to maximize the received video quality while accounting for network congestion and lossy channels. As illustrated in Fig. 2, there are four main components to the system, described in the following.

3.1 CS Camera

This is the subsystem where the compressed sensing image capture takes place. The details of the CS-based video representation are discussed in detail in Section 4.1.1. The camera can be either a traditional CCD or CMOS imaging system, or it can be a single-pixel camera as proposed in [21]. In the latter case, the samples of the image are directly obtained by taking a linear combination of a random set of the pixels and summing the intensity through the use of a

photodiode. The samples generated are then passed to the video encoder.

3.2 CSV Video Encoder

The CSV video encoder is discussed in Section 4.2. The encoder receives the raw samples from the camera and generates compressed video frames. The compression is obtained through properties of CS and by leveraging the temporal correlation between consecutive video frames. The number of samples, along with the sampling matrix (i.e., which pixels are combined to create each sample, as discussed in more detail in Section 4.1) are determined at this block. The number of samples, or *sampling rate*, is based on input from the rate controller, while the sampling matrix is preselected and shared between sender and receiver.

3.3 Rate Controller

The rate control block takes as input the end-to-end RTT of previous packets and the estimated sample loss rate to determine the optimal sampling rate for the video encoder. This sampling rate is then fed back to the video encoder. The rate control law, which is designed to maximize the received video quality while preserving fairness among competing videos, is described in detail in Section 5. The CS sampling rate determined by the C-DMRC block is chosen to provide the optimal received video quality across the entire network, which is done by using the RTT to estimate the congestion in the network along with the input from the adaptive parity block to compensate for lossy channels.

3.4 Adaptive Parity

The Adaptive Parity block uses the measured or estimated sample error rate of the channel to determine a parity scheme for encoding the samples, which are input directly from the video encoder. The Adaptive Parity scheme is described in Section 6.

4 CS Video Encoder (CSV)

In this section, we introduce the video encoder component of the compressive distortion-minimizing rate control system.

4.1 Video Model

4.1.1 Compressed Sensing Preliminaries

We consider an image signal represented through a vector $\mathbf{x} \in R^N$, where N is the vector length. We assume that there exists an invertible $N \times N$ transform matrix $\mathbf{\Psi}$ such that

$$\mathbf{x} = \mathbf{\Psi}\mathbf{s}, \tag{1}$$

where $\mathbf{s}$ is a K-sparse vector, i.e., $\|s\|_0 = K$ with $K < N$, and where $\|\cdot\|_p$ represents p-norm. This means that the image has a sparse representation in some transformed domain, e.g., wavelet. The signal is measured by taking $M < N$ measurements from linear combinations of the element vectors through a linear measurement operator $\mathbf{\Phi}$. Hence,

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{s} = \tilde{\mathbf{\Psi}}\mathbf{s}. \tag{2}$$

We would like to recover $\mathbf{x}$ from measurements in $\mathbf{y}$. However, since $M < N$ the system is underdetermined. Hence, given a solution $\mathbf{s^0}$ to (2), any vector $\mathbf{s}^*$ such that $\mathbf{s}^* = \mathbf{s^0} + \mathbf{n}$, and $\mathbf{n} \in \mathcal{N}(\tilde{\mathbf{\Psi}})$ (where $\mathcal{N}(\tilde{\mathbf{\Psi}})$ represents the null space of $\tilde{\mathbf{\Psi}}$), is also a solution to (3). However, it was proven in [16] that if the measurement matrix $\mathbf{\Phi}$ is sufficiently incoherent with respect to the sparsifying matrix $\mathbf{\Psi}$, and K is smaller than a given threshold (i.e., the sparse representation $\mathbf{s}$ of the original signal $\mathbf{x}$ is "sparse enough"), then the original $\mathbf{s}$ can be recovered by finding the sparsest solution that satisfies (2), i.e., the sparsest solution that "matches" the measurements in $\mathbf{y}$. However, the problem above is in general NP-hard [42]. For matrices $\tilde{\mathbf{\Psi}}$ with sufficiently incoherent columns, whenever this problem has a sufficiently sparse solution, the solution is unique, and it is equal to the solution of the following problem:

$$P_1 : \ minimize \ \|\mathbf{s}\|_1$$

$$subject\ to : \|\mathbf{y} - \tilde{\mathbf{\Psi}}\mathbf{s}\|_2^2 < \epsilon, \tag{3}$$

where ϵ is a small tolerance. Note that problem P_1 is a convex optimization problem [43]. The reconstruction complexity equals $O(M^2N^{3/2})$ if the problem is solved using interior point methods [44]. Although more efficient reconstruction techniques exist [45], the framework presented in this paper is independent of the specific reconstruction method used.

4.1.2 Frame Representation

We represent each frame of the video by 8-bit intensity values, i.e., a grayscale bitmap. To satisfy the sparsity requirement of CS theory, the wavelet transform [46], [47] is used as a sparsifying base. A conventional imaging system or a single-pixel camera [21] can be the base of the imaging scheme. In the latter case, the video source only obtains random samples of the image (i.e., linear combinations of the pixel intensities). In our model, the image can be sampled using a scrambled block Hadamard ensemble [48]

$$\mathbf{y} = \mathbf{H_{32}} \cdot \mathbf{x}, \tag{4}$$

where $\mathbf{y}$ represents image samples (measurements), $\mathbf{H_{32}}$ is the 32×32 Hadamard matrix and $\mathbf{x}$ the matrix of the image pixels. The matrix $\mathbf{x}$ has been randomly reordered and shaped into a $32 \times \frac{N}{32}$ matrix, where N is the number of pixels in the image. Then, M samples are randomly chosen from $\mathbf{x}$ and transmitted to the receiver. The receiver then uses the M samples along with the randomization patterns for both randomizing the pixels in $\mathbf{x}$ and choosing the samples out of $\mathbf{y}$ to be transmitted (both of which can be decided before network setup) and recreates the image solving P_1 in (3) through a suitable algorithm, e.g., GPSR[3] [49], StOMP [50].

4.2 CS Video Encoder (CSV)

The CSV video encoder uses compressed sensing to encode video by exploiting the spatial and temporal redundancy within the individual frames and between adjacent frames, respectively.

4.2.1 Intraframe (I) Encoding

As stated above, each of the I frames are encoded individually, i.e., as a *single image* that is *independent of the*

3. GPSR is used for image reconstruction in the simulation results presented in this paper.

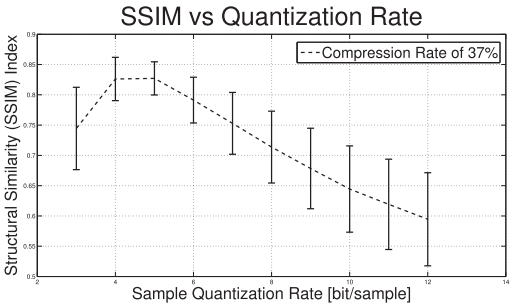

Fig. 3. Structural similarity index [22] for images with a constant bit rate of 37 percent of the original image size for varying quantization levels.

surrounding frames. Two variables mainly affect the compression of I frames; the sample quantization rate (Q), and the ratio of samples per pixel (γ), referred to as the sampling rate.

Sample quantization rate. The sample quantization rate refers to the number of bits per sample used to quantize the data for digital transmission. We conducted empirical studies to test the effect of quantization of samples generated from linear combinations of pixels as in (4) over a set of reference images with a constant overall compression rate. The results are reported in Fig. 3, which shows the SSIM index [22] of a set of reference images for multiple quantization levels. The reference images used are 25 grayscale images varying in size between 256×256 pixels to $1{,}024 \times 1{,}024$ pixels. All images are from the USC Signal and Image Processing Institute image repository [51]. As Q decreases and less bits are being used to encode each sample, more samples can be obtained for the same compression rate. There is a clear maximum value at $Q = 5$.

Sampling rate γ. The sampling rate γ is the number of transmitted samples per original image pixel. An empirical study was performed on the images in [51] to determine the amount of distortion in the recreated images due to varying sampling rates, and is reported in Fig. 4.

The proposed CSV encoder is designed to: 1) encode video at low complexity for the encoder; 2) take advantage of the temporal correlation between frames. While the proposed method is general, it works particularly well for security videos, in which the camera is not moving, but only the objects within the Field of View (FOV) are moving. Because of the still camera, there will often be a large amount of redundancy from one frame of the video to the next due to the unchanging background.

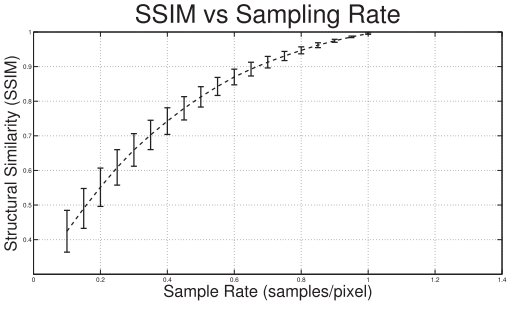

Fig. 4. Structural similarity index [22] for images with varying levels of sampling rate γ.

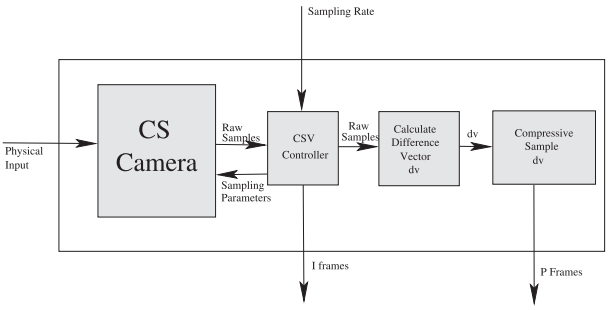

Fig. 5. Block diagram for CS Video Encoder.

The key intuition behind the encoder is to exploit this redundancy within the framework of compressed sensing. To this aim, we consider the *algebraic difference between the CS samples*. Then, this difference is *again compressively sampled* and transmitted. If the image being encoded and the reference image are very similar (i.e., have a very high correlation coefficient), then this difference image (represented as difference between compressed samples) will be sparse (in the domain of compressed samples) and have less variance than either of the original images. The main compression of the difference frames comes from the above properties and is exploited in two ways. First, because of the sparsity in the difference frame, it can be further compressed using CS resulting in fewer samples. The number of samples is based on the sparsity the same as is done in the CS sampling of the initial frame. Second, the lower variance allows us to use fewer quantization levels to accurately represent the information, and therefore fewer bits per sample.

4.2.2 Video Encoding

The video encoding process is determined by the type of frame (I frame or P frame) being encoded, as shown in Fig. 5. The pattern of the encoded frames is $IPPP \ldots PIPPP \ldots$, where the distance between two I frames is referred to as the group of pictures (GOP).

I frames are encoded using (4). The number of samples to include is determined as $\gamma_I \cdot N$, where N is the number of pixels in the unencoded frame and γ_I is the I frame sampling rate. The rate control law to determine the current value for γ_I is discussed in Section 5. The samples are then quantized at the optimal rate ($Q = 5$ in our experiments) and then transmitted.

To construct a P frame, the image is first sampled using CS using the same scheme as an I frame (i.e., using (4)). However, to take advantage of temporal correlation, additional processing is required. First, the difference vector ($\mathbf{dv}$) for frame t is calculated as

$$\mathbf{dv} = \mathbf{s^*_{t-1}} - \mathbf{s^*_t}, \tag{5}$$

where $\mathbf{s^*_t}$ is a vector containing all of the samples of the tth frame. The $\mathbf{dv}$ is then compressed using (4), quantized and transmitted. The number of samples m needed to represent $\mathbf{dv}$ after it is compressed is proportional to the *sparsity* K of $\mathbf{dv}$ and defined as $m \approx K\, log(N)$, where N is the length of $\mathbf{dv}$. For videos with very high temporal correlation such as security videos, the $\mathbf{dv}$ will also have very low variance, allowing for a lower quantization rate Q. In this work, we used $Q = 3$.

TABLE 1
Compression Improvement Using P Frames

Amount of Motion	low	medium	high
Improvement	556%	455%	172%

In terms of compression ratio, the effectiveness of this scheme depends on the temporal correlation between frames of the video. The compression of each of these schemes (at the same received video quality) was compared to basic CS compression (i.e., only I frames) for three videos. The videos chosen were foreman (representing high motion) and two security videos; one monitoring a walkway with moderate traffic (moderate motion) and one monitoring a walkway with only light traffic (low motion), and the percentage improvement calculated as $\frac{\text{Size without P frames}}{\text{Size with P frames}} \times 100$ is presented in Table 1. While the high-motion video can be compressed increases by 172 percent, the moderate- and low-motion security videos (which are the type of videos that our encoder was designed for) show far more improvement by using the P frames.

4.2.3 Video Decoding

The decoding process, illustrated in Fig. 6, solves the problem in (3) to reconstruct the $\mathbf{dv}$ (in the case of a P frame) and the original frame. For I frames, the frame can be directly reconstructed from the received samples. For P frames, the $\mathbf{dv}$ must first be reconstructed from the received samples. Once this vector is reconstructed using (3), the samples for the tth P frame are found by $\mathbf{s_t^*} = \mathbf{dv} + \mathbf{s_{t-1}^*}$. The tth frame is then reconstructed using (3) from $\mathbf{s_t^*}$.

5 Rate Control Subsystem

In this section, we introduce the congestion-avoiding rate control mechanism for use with the Compressed Sensed Video encoder (CSV) described in Section 4.2. The rate control subsystem both provides fairness in terms of video quality and maximizes the overall video quality of multiple videos transported through the network.

To avoid network congestion, a sending node needs to take two main factors into account. First, the sender needs to regulate its rate in such a way as to allow any competing transmission at least as much bandwidth as it needs to attain a comparable video quality as itself. Note that this is different from current Internet practice, in which the emphasis is on achieving fairness in terms of data rate (not video quality). Second, the sender needs to regulate its rate to make sure that packet losses due to buffer overflows are reduced, which can be done by reducing the overall data rate if it increases to a level that the network cannot sustain.

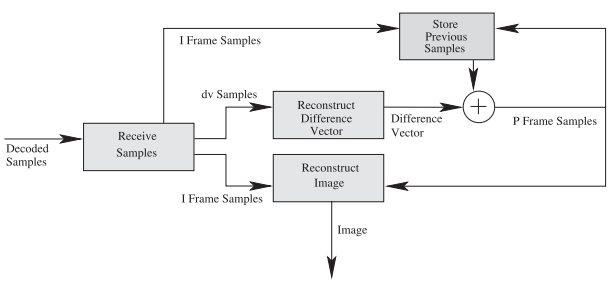

Fig. 6. Block diagram for CS Video Decoder.

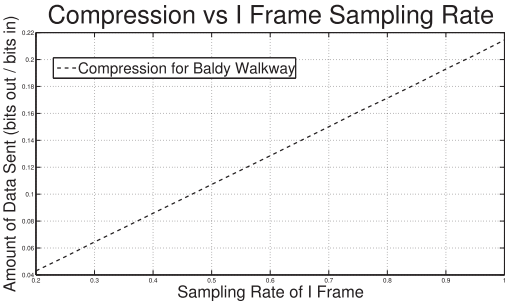

Fig. 7. Ratio of Encoder Output to Encoder Input versus I Frame Sampling Rate.

To determine congestion, the round trip time (RTT) is measured for the transmitted video packets, where RTT is defined as the amount of time it takes for a packet to go from the source to the destination and a small reply packet to go from the destination back to the source. The change in RTT is measured as

$$\Delta\widetilde{RTT_t} = \frac{\sum_{i=0}^{N-1} a_i \cdot RTT_{t-i}}{N \cdot \sum_{i=0}^{N-1} a_i} - \frac{\sum_{i=1}^{N} a_i \cdot RTT_{t-i}}{N \cdot \sum_{i=1}^{N} a_i}, \tag{6}$$

which represents the difference of the weighted average over the previous N received RTT measurements with and without the most recent measurement. The weights a_i are used to low-pass filter the round trip time measurements, to give more importance to the most recent RTT measurements and to make sure that the protocol reacts quickly to current network events, while averaging assures that nodes do not react too quickly to a single high or low measurement.

The video encoder described in Section 4 generates two types of video frames; the I frame, which is an intraencoded frame, and the P frame, which is an interencoded frame. The I frames are *independently encoded*, i.e., they are encoded using only the data contained within a single frame allowing these frames to be decoded independently of the previous frame. However, I frames do not take advantage of correlation between frames resulting in lower rate-distortion performance. P frames on the other hand are encoded based on previous frames by leveraging the temporal correlation between frames. Although this results in smaller frame sizes, it also allows errors to propagate from one frame to the next [32].

We present a novel approach in which the sampling rate γ_I of the video is used to control the data rate. Since γ_I is linearly proportional to the compression of the video (as seen in Fig. 7), controlling γ_I controls the compression rate of the entire video and therefore the data rate of the video transmission. Because of this linear relationship, we can control the compression of the entire video by varying only the I frame video quality.

We model the quality of the received video stream with a three-parameter model [32]

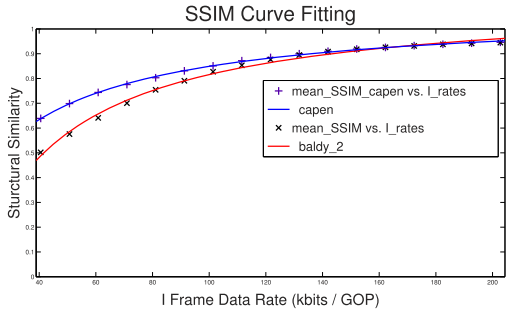

Fig. 8. Linear least-squares curve fitting for two videos using CS Video Encoder.

$$D_I = D_0 + \frac{\theta}{\gamma_I - R_0}, \tag{7}$$

where D_I represents the distortion of the video. The parameters D_0, θ, and R_0 depend on the video characteristics and quantization level Q and can be estimated from empirical rate-distortion curves via a linear least-squares curve fitting.

The rate control is based on the marginal distortion factor δ, which is defined by

$$\delta = \frac{\theta}{(\gamma_I - R_0)^2}, \tag{8}$$

i.e., the derivative of (7) with respect to γ_I.

At the source node of each video transmission, the amount of data generated by the video source for the tth group of pictures is controlled through

$$\gamma_{I,t+1} = \begin{cases} \gamma_{I,t} - (1-\delta)\cdot\beta\cdot\Delta\widetilde{RTT}_t & \text{if } \Delta\widetilde{RTT}_t > \alpha \\ \gamma_{I,t} - \delta\cdot\kappa\cdot\Delta\widetilde{RTT}_t & \text{if } \Delta\widetilde{RTT}_t < -\alpha \\ \gamma_{I,t} & \text{else,} \end{cases} \tag{9}$$

where $\beta > 0$ and $\kappa > 0$ are both constants used to scale δ to the range of the sampling rate. α is a constant used to prevent the rate from oscillating with very minor changes in $\Delta\widetilde{RTT}_t$. The marginal distortion factor is used in (9) to promote fairness in terms of distortion. If there are two nodes transmitting video and both observe the same negative value for $\Delta\widetilde{RTT}_t$, the sending node with the lower current video quality will take advantage of the decreased network congestion more than the node that is transmitting at a higher rate by increasing its sampling rate more aggressively. The inverse is true for positive values of $\Delta\widetilde{RTT}_t$. This can be seen in Fig. 8. At lower compression levels, a change in the rate has a higher impact on the received image quality than an equal change will have at a higher rate. Similarly, $1-\delta$, results in a function which has low values at low rates, and higher values at higher rates. This $1-\delta$ is used to prevent a node from decreasing the rate significantly when the rate is already low, but encourage the node to decrease the rate when the data rate is already high.

Channel errors are accounted for through the use of the adaptive parity scheme, described in Section 6. The adaptive parity scheme provides feedback to the C-DMRC rate controller indicating the expected sample delivery success rate c.

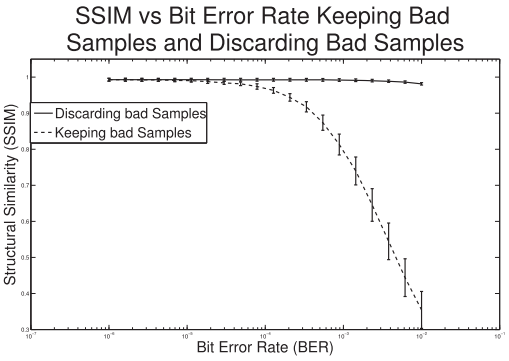

Fig. 9. SSIM for images with and without errored samples.

6 Adaptive Parity-Based Transmission

For a fixed number of bits per frame, the perceptual quality of video streams can be further improved by dropping errored samples that would contribute to image reconstruction with incorrect information. This is demonstrated in Fig. 9 which shows the reconstructed image quality both with and without including samples containing errors. Although the plots in Fig. 9 assume that the receiver knows which samples have errors, they demonstrate that there is a very large possible gain in received image quality if those samples containing errors can be removed.

We studied adaptive parity with compressed sensing for image transmission in [52], where we showed that since the transmitted samples constitute an unstructured, random, incoherent combination of the original image pixels, in CS, unlike traditional wireless imaging systems, no individual sample is more important for image reconstruction than any other sample. Instead, the number of correctly received samples is the only main factor in determining the quality of the received image. Because of this, a sample containing an error can simply be discarded and the impact on the video quality, as shown in Fig. 9, is negligible as long as the error rate is small. This error detection is realized by using even parity on a predefined number of samples, which are all dropped at the receiver or at an intermediate node if the parity check fails. This is particularly beneficial in situations when the BER is still low, but too high to just ignore errors. To determine the amount of samples to be jointly encoded, the amount of correctly received samples is modeled as

$$c = \left(\frac{Q\cdot b}{Q\cdot b + 1}\right)(1 - BER)^{Q\cdot b}, \tag{10}$$

where c is the estimated amount of correctly received samples, b is the number of jointly encoded samples, and Q is the quantization rate per sample. To determine the optimal value of b for a given BER, (10) can be differentiated, set equal to zero and solved for b, resulting in

$$b = \frac{-1 + \sqrt{1 - \frac{4}{log(1-BER)}}}{2Q}. \tag{11}$$

The optimal channel encoding rate can then be found from the measured/estimated value for the end-to-end BER and used to encode the samples based on (10). The received video quality using the parity scheme described was compared to different levels of channel protection using Rate Compatible Punctured Codes (RCPCs). Specifically, we

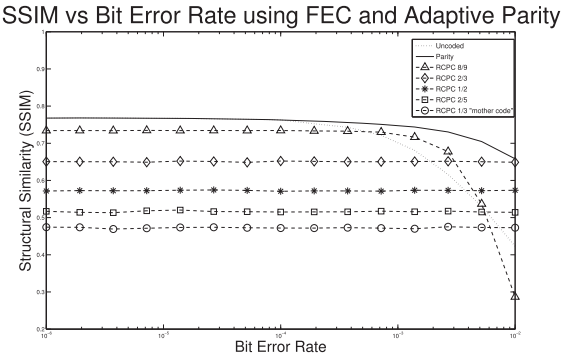

Fig. 10. Adaptive parity versus RCPC encoding for variable bit error rates.

use $\frac{1}{4}$ mother codes discussed in [53]. Clearly, as these codes are punctured to reduce the redundancy, the effectiveness of the codes decreases as far as the ability to correct bit errors. Therefore, we are trading BER for transmission rate.

Fig. 10 illustrates the performance of the adaptive parity scheme compared to RCPC codes. For all reasonable values of the bit error rate, the adaptive parity scheme outperforms all levels of RCPC codes. The parity scheme is also much simpler to implement than more powerful Forward Error Correction (FEC) schemes. This is because even though the FEC schemes show stronger error correction capabilities, the additional overhead does not make up for the video quality increase compared to just dropping the samples that have errors.

7 Performance Evaluation

We perform two sets of experiments to verify the performance of the C-DMRC system. First, the rate controller is simulated using ns-2 version 2.33, modified to simulate transmission of CS video. In addition, to evaluate the effect of a real wireless channel, CS video streaming with the adaptive parity-based channel encoder is tested using a multihop test bed based on USRP2 software defined radios.

7.1 Evaluation of Rate Controller

The rate control algorithm of C-DMRC is compared directly to TFRC [29], [28] to verify: 1) that the received video quality is higher than that obtained by using TFRC rate control for a single video transmission; 2) that the overall fairness (as measured with Jain's Fairness Index) is higher between multiple videos than that obtained from TFRC; 3) that the overall received video quality over multiple video transmissions is higher for the C-DMRC rate controller than for TFRC. The topology of the network is a Manhattan Grid consisting of 49 nodes (7×7). The senders and sink are chosen randomly for 10 random number seeds. All senders transmitted video to a single sink node. Routing is based on AODV [54], and MAC on IEEE 802.11b. The model used for the wireless radios is based on the 914 MHz Lucent WaveLAN DSSS radio. The physical channel is modified to return the bit error rate of each packet, which is needed for the adaptive parity calculation.

We use real video traces recorded at the University at Buffalo to simulate video traffic within the ns-2 simulator. Initially, trace files are obtained from the CSV video encoder for multiple values of γ_I. These trace files are input into ns-2, where the rate control decisions are made at simulation time. The network simulator determines the sampling rate γ_I, and derives the video size based on this value. After network simulation, a trace of the received samples is fed back into the CSV video decoder and the resulting received video frames are recreated using compressed sensing. These video frames are then compared to the original uncompressed and untransmitted video.

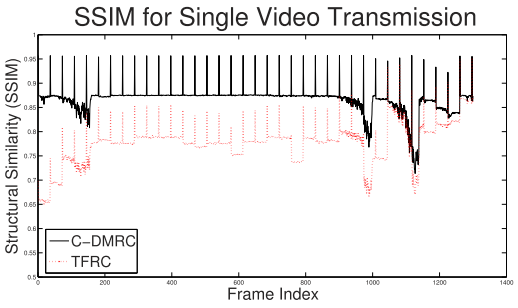

Fig. 11. SSIM of a single video transmission using C-DMRC and TFRC.

The first result is for a single video transmission within the 49 node network. This is done to compare C-DMRC and TFRC in a best case scenario (i.e., no interflow interference and sufficient available capacity). Fig. 11 shows the instantaneous SSIM at each frame of a 1,300 frame video. Clearly, C-DMRC results in a higher SSIM value for basically the entire video. The portions of the video where both SSIM values decrease are due to variations in the video content. At these points, traffic originating from the video increases, resulting in an increase of RTT and a decrease in the sampling rate. Both C-DMRC and TFRC respond to this traffic increase quickly, but C-DMRC recovers faster than TFRC.

The second set of simulations compare C-DMRC and TFRC with multiple videos simultaneously transmitted in the network. The number of video transmissions is varied from 1 to 5, with each video starting 10 seconds (120 frames) after the previous. The varying starting rates assure that videos starting at different times are treated fairly.

Fig. 12 shows the results from this simulation. For each of the simulations, C-DMRC results in a higher average SSIM than TFRC. The fairness is shown in Fig. 13, where Jain's Fairness Index [55] is used to measure the fairness among multiple senders. Again, C-DMRC clearly outperforms TFRC.

Finally, in Fig. 14, we show the quality in SSIM and PSNR of the Akiyo video [56]. Unlike the security videos tested until now that have brief periods of moderate

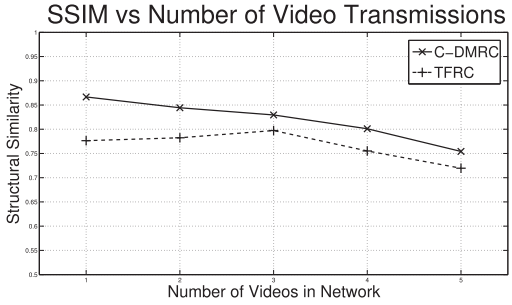

Fig. 12. SSIM of a multiple video transmissions using C-DMRC and TFRC.

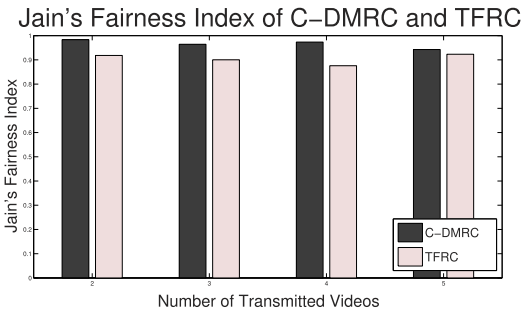

Fig. 13. Jain's Fairness Index of a multiple video transmissions using C-DMRC and TFRC.

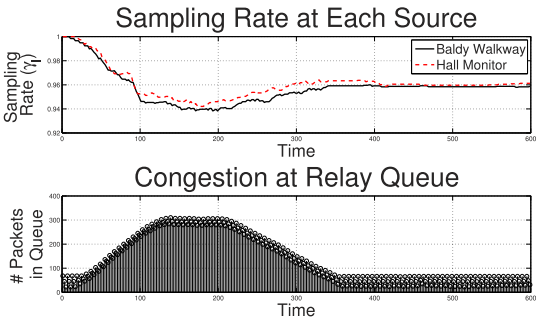

Fig. 15. Test bed results showing the sampling rates of the two videos and the length of the queue at the relay.

motion, Akiyo has constant low motion. Similar to the other tests, C-DMRC outperforms TFRC.

7.2 Software Defined Radio Test Bed Experiments

The proposed C-DMRC scheme is tested on a Universal Software Radio Peripheral 2 (USRP2) [57] platform running the GNU radio [58] software radio protocol stack. There are three nodes in the network, with one node acting as a video source, one as both a source and a relay and a third as the common destination. The MAC protocol is IEEE 802.11b and the modulation scheme employed is DQPSK to provide a physical layer data rate of 2 Mbit/s. The radios are placed 6 m apart, and the frequency selected for transmission is 2.433 GHz. Each frame output from the CS encoder is packetized and transmitted as a single burst. The GOP size of the video is three with each GOP consisting of one I frame followed by two P frames. One acknowledge packet is sent for each frame, and $\Delta\widetilde{RTT}$ is measured over six frames.

Two different videos are transmitted through the network. The videos are input into the source nodes as a series of bitmap images. The source nodes then use the C-DMRC scheme to encode the video using CSV, add parity bits for error correction, and transmit the video through the network. The sampling rate at each source is determined using the rate control law (9). To determine the performance of the system, the sampling rate γ_I is measured at each of the sources, and the queue length is measured at the relay. As Fig. 15 shows, the sampling rates vary based on the queue length, and converge to a stable solution. When the sum encoding rate is higher than the capacity of the link from the relay to the destination, there is congestion as can be seen by the increase in the queue lengths. The source nodes are able to detect this congestion from the increase in $\Delta\widetilde{RTT}$ and reduce their sampling rates by as determined by (9). As the queues empty and $\Delta\widetilde{RTT}$ decreases, the sampling rate is increased until a stable value is reached. The resulting sampling rate is low enough that the bottleneck link can handle the resulting traffic, and the sampling rates remain nearly constant for the remainder of the video transmission.

Finally, a test is done to demonstrate the fairness of the protocol across different videos from different sources. In this test, two significantly different videos are considered. One is the Pamphlet video and the other is the Baldy Walkway video, which is shot at the University at Buffalo. The Pamphlet video has sudden rapid movements and the Baldy Walkway video has low-temporal movement. The figure shows the reaction of the sampling rate of the Baldy Walkway video with changes in the Pamphlet video. As seen in Figs. 16 and 17, the sampling rate for both the videos is nearly the same and responds to changes in either of the videos.

7.3 Comparison with H.264

In this section, we present a brief qualitative comparison between the proposed CSV encoder and H.264. Based only on rate distortion performance, H.264, which uses the most sophisticated motion estimation techniques, far outperforms CSV. However, when the error resilience and low-complexity encoding of CSV are considered, we can make a case that CSV requires less power for both encoding a video and for transmitting each frame for a target level of received video quality. Though we present her some measurements between the two encoders, a full comparison between the two encoders requires a new power-rate-distortion framework that takes error resilience into account, which is beyond the scope of this paper and saved for future work.

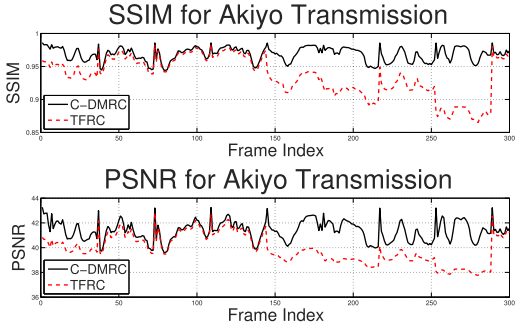

Fig. 14. SSIM and TFRC of Foreman transmissions using C-DMRC and TFRC.

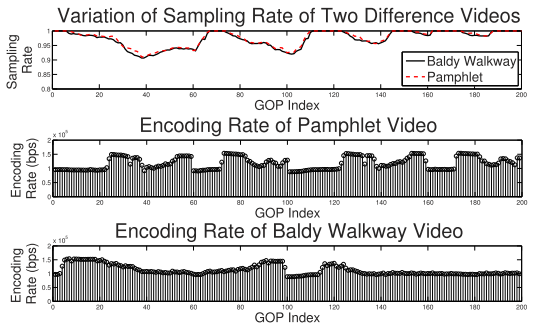

Fig. 16. Sampling rate and encoding rates of significantly different videos.

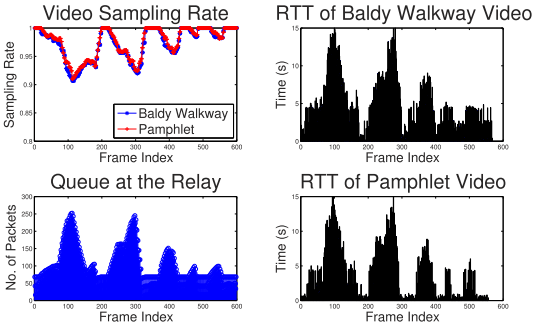

Fig. 17. Sampling rate and round trip times of significantly different videos.

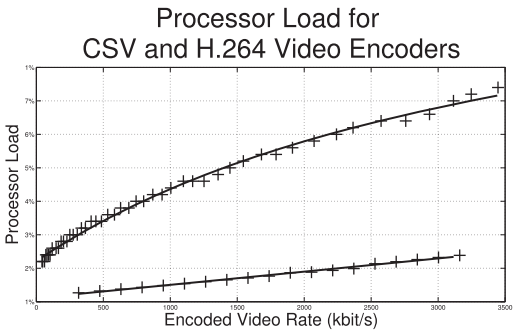

Fig. 19. Processor load for a software implementation of H.264 and CSV.

We first look at a comparison between H.264 and CSV in terms of error resilience. As previously discussed in Section 6, CS measurements can tolerate a fairly large amount of bit errors before the received video quality is affected. This is certainly not the case for predictive video encoding, and not even more transform-based image compression standards such as JPEG. Fig. 18 shows the effect of bit errors on the received video quality for CVS and H.264. This data is found by transmitting each video through a binary symmetric channel, and decoding the errored video. No channel coding was used for either encoder, and the H.264 video is decoded using the implementation in ffmpeg. The SSIM is modeled empirically as a low pass filter. The H.264 quality decreases when the BER increases above 10^{-4}. However, CSV quality does not decrease until the BER increases above 5×10^{-3}. Clearly, the CSV encoder can tolerate much higher error rates than H.264 before the video quality is significantly affected. This could result in significant transmission power savings or a significant decrease in the amount of forward error correction coding for a video sensor node.

Another important factor in WMSNs is the complexity of encoding CSV compared to H.264. To measure the complexity difference, the CSV encoder was compared to the libx264 encoder as implemented in ffmpeg [59] using only intra (I) frames. The processor load for encoding the video was compared, and is presented in Fig. 19. Clearly, the processor load is significantly lower for CSV than for H.264. This leads to a decrease in the energy needed to encode the video. To determine an overall comparison between the two encoders, the energy savings gained by the error resilience and lower encoding complexity need to be compared to the energy cost of transmitting more bits to represent each frame.

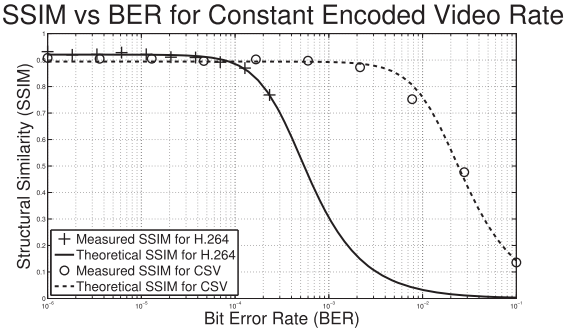

Fig. 18. SSIM of H.264 and CSV with channel errors.

8 The Optimality of the C-DMRC Rate Controller

Last, we analyze the performance of the rate controller presented in Section 5. We represent the network as a set $\mathcal{N}$ of nodes. The set $\mathcal{L}$ represents the set of all links in the network. We indicate the set of video source nodes as $\mathcal{S}$, with $\mathcal{S} \subset \mathcal{N}$. We define $\mathcal{L}(s)$ as the set of links in $\mathcal{L}$ used by source s. For each link l, let $\mathcal{S}(l) = \{s \in \mathcal{S} \mid l \in \mathcal{L}(s)\}$ be the set of sources that use link l. By definition, $l \in \mathcal{L}(s)$ if and only if $s \in \mathcal{S}(l)$. The problem can then be formulated as follows:

$$\begin{aligned} \underset{\boldsymbol{\gamma_I}}{\text{maximize}} \quad & \sum_{s \in \mathcal{S}} U_s(\gamma_{I,s}) \\ \text{subject to} \quad & \sum_{i:\ell \in \mathcal{L}(i)} \tau \gamma_{I,i} \leq c_\ell, \quad \forall \ell \in \mathcal{L} \end{aligned} \tag{12}$$

where $\boldsymbol{\gamma_I} = [\gamma_{I,1}, \gamma_{I,2}, \ldots, \gamma_{I,|\mathcal{S}|}]$ is the vector of sampling rates for all sources, τ is a constant that maps sampling rates to data rates, i.e., $x_i = \tau \gamma_{I,i}$, and $U_i(\gamma_{I,i}) = D_{0,i} + \frac{\theta_i}{\gamma_{I,i} - R_{0,i}}$ is the quality of video source i at sampling rate $\gamma_{I,i}$. The set $\mathcal{L}(i)$ contains all links used by video stream i, and c_ℓ represents the capacity on link ℓ. Since $U_i(\gamma_{I,i})$ as defined in (7) is a concave function over the feasibility region and the constraints are affine in the rate variables, the problem is convex. We modeled the problem in (12) with CVX [60] and solved it as a semidefinite program using SeDuMi [61].

We can now prove the following lemma:

Lemma 1. *The rate control equation update (9) converges to a distributed solution to (11).*

Proof. The Lagrangian of (11) is defined as [43]

$$\begin{aligned} L(\boldsymbol{\gamma_I}, \boldsymbol{\lambda}) &= \sum_{s \in \mathcal{S}} U_s(\gamma_{I,s}) - \sum_{l \in \mathcal{L}} \lambda_l \left(\sum_{s \in \mathcal{S}(l)} \tau_s \gamma_{I,s} - c_l \right) \\ &= \sum_{s \in \mathcal{S}} \left(U_s(\gamma_{I,s}) - \tau_s \gamma_{I,s} \sum_{l \in \mathcal{L}(s)} \lambda_l \right) + \sum_{l \in \mathcal{L}} \lambda_l c_l, \end{aligned} \tag{13}$$

where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_{|\mathcal{L}|}]$. The Lagrange dual function is then given by

$$\begin{aligned} g(\boldsymbol{\lambda}) &= \max_{\boldsymbol{\gamma_I}} L(\boldsymbol{\gamma_I}, \boldsymbol{\lambda}) \\ &= \sum_{s \in \mathcal{S}} \max_{\gamma_{I,s}} \left[U_s(\gamma_{I,s}) - \tau \gamma_{I,s} \lambda^s \right] + \sum_{l \in \mathcal{L}} \lambda_l c_l, \end{aligned} \tag{14}$$

where $\lambda^s = \sum_{l \in \mathcal{L}(s)} \lambda_l$.

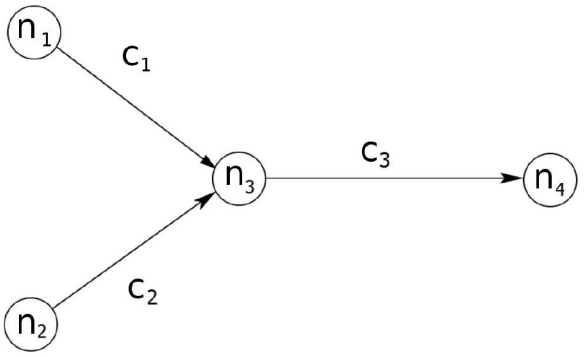

Fig. 20. Simple network topology demonstrating congestion between two video streams.

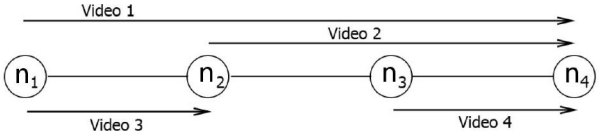

Fig. 21. Linear network topology demonstrating varying congestion between four video streams.

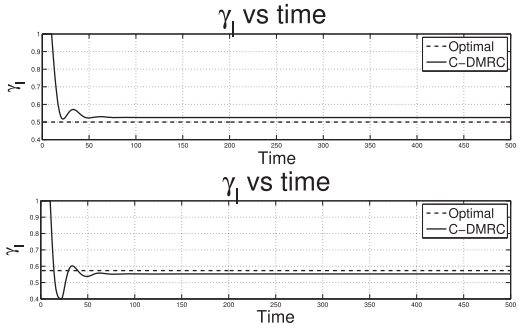

Fig. 22. Sampling rate from the C-DMRC rate controller compared to the optimal sampling rate allocation for Topology 1.

Each Lagrange multiplier λ_l, which can be interpreted as the delay at link l [62], is implicitly updated as

$$\lambda_l(t+1) = \left[\lambda_l(t) - \alpha\left(c_l - x_l^*(\lambda(t))\right)\right]^+, \tag{15}$$

where $x_l^*(\lambda(t)) = \sum_{s \in S(l)} \tau_s \gamma_{I,s}^*$ represents the total optimal rate offered at link l and α denotes the step size. We define $[\cdot]^+$ as $max(0, \cdot)$.

Since $U_s(\gamma_{I,s}) - \tau_s \gamma_{I,s}$ is differentiable, $\max_{\gamma_{I,s}} \left[U_s(\gamma_{I,s}) - \tau_s \gamma_{I,s} \lambda^s\right]$ is obtained when

$$\frac{dU_s(\gamma_{I,s})}{d\gamma_{I,s}} = \tau_s \lambda^s, \tag{16}$$

which states that the derivative with respect to the sampling rate should be equal to a scaled version of the delay. Since $U_s(\gamma_{I,s})$ (as defined in (7)) is a concave monotonically increasing function in $\gamma_{I,s}$, $\frac{dU_s(\gamma_{I,s})}{d\gamma_{I,s}}$ is decreasing in $\gamma_{I,s}$. Therefore, as λ^s varies, the optimal update direction of $\gamma_{I,s}$ is the *negative of the direction of the change in round trip time*.

The simplest interpretation of $\Delta RTT^i(t+1)$ as calculated in (6) and used in (9) for source i is the difference between consecutive delay measurements $\lambda^i(t) - \lambda^i(t-1)$. The update direction of $\gamma_{I,s}$ is then given by $(-\Delta RTT)$, which is the direction of the update in (9). Finally, it was shown in [63] that, given a small enough step size, a gradient projection algorithm

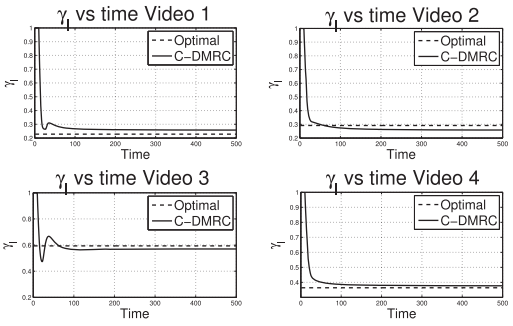

Fig. 23. Sampling rate from the C-DMRC rate controller compared to the optimal sampling rate allocation for Topology 2.

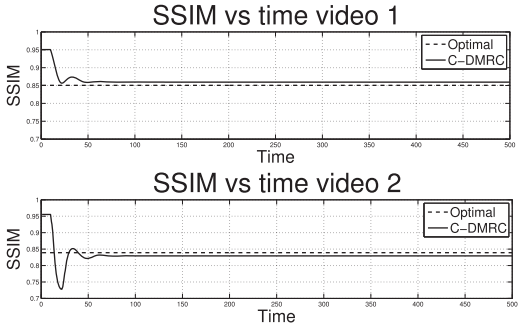

Fig. 24. Video quality from the C-DMRC rate controller compared to the optimal sampling rate allocation for Topology 1.

such as (9) will converge to the optimal sampling rate allocation. □

Numerical simulations were run to support this interpretation. Two simple networks were tested as shown in Figs. 20 and 21, respectively, where C_i represents the capacity on link i and N_i represents node i. The arrows represent video streams. In both cases, the optimal rate allocation was determined by solving the optimization problem directly as a semidefinite program using SeDuMi [61] with the convex optimization toolbox CVX [60], and the same problem was solved using the iterative algorithm (9).

These two topologies were chosen because they verify two important requirements of a distortion-based rate controller. The network in Fig. 20 has two video streams with a single point of congestion. This topology can be used to assure that two different videos with different rate-distortion properties achieve the same received video quality. The other topology, shown in Fig. 21, was used to show that the rate controller will take advantage of unused capacity. Video 3 in this network is only contending with a single other video, while the other three videos are contending with each other resulting in a higher optimal rate for Video 3.

The results from these tests are shown in Figs. 22, 23, 24, and 25. Figs. 22 and 23 show the I frame sampling rate of the videos compared to the optimal value, and Figs. 24 and 25 show the actual video qualities. In all cases, the rate found by the iterative algorithm was within 5 percent of the optimal value as determined by the convex solver. The 5 percent difference between the optimal rates and the rates obtained from the iterative algorithm are due to the step size of the algorithm. If the step size were decreased, the resulting rate would be closer to the optimal. However, making the step size too small results in an algorithm which is infeasible to implement because of the amount of updates

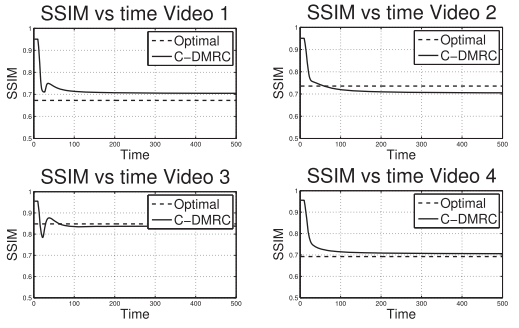

Fig. 25. Video quality from the C-DMRC rate controller compared to the optimal sampling rate allocation for Topology 2.

needed. Finally, to avoid the trivial solution of all rates and qualities being equal, different videos were transmitted. The simulations show that the iterative algorithm achieved all requirements, and was nearly optimal for both networks.

9 Conclusions and Future Work

This paper introduced a new wireless video transmission system based on compressed sensing. The system consists of a video encoder, distributed rate controller, and an adaptive parity channel encoding scheme that take advantage of the properties of compressed sensed video to provide high-quality video to the receiver using a low-complexity video sensor node. The rate controller was then shown to be an implementation of an iterative gradient descent solution to the optimal rate allocation optimization problem. Simulation results show that the C-DMRC system results in a 5-10 percent higher received video quality in both a network with a higher load and a small load. Simulation results also show that fairness is not sacrificed, and is in fact increased, with the proposed system. Finally, the video encoder, adaptive parity and rate controller were implemented on a USRP2 software defined ratio. It was shown that the rate controller correctly reacts to congestion in the network based on measured round trip times, and that the system works over real channels. We intend to implement the remaining portions of the C-DMRC system on the USRP2 radios, including image capture and video decoding. We will also measure the performance and complexity of this system compared to state-of-the-art video encoders (H.264, JPEG-XR, MJPEG, MPEG), transport (TCP, TFRC) and channel coding (RCPC, Turbo codes).

Acknowledgments

A preliminary shorter version of this paper [1] appeared in the Proceedings of IEEE SECON 2010, Boston, Massachusetts, June 2010. This paper was based upon work supported in part by the US National Science Foundation under grant CNS1117121 and by the US Office of Naval Research under grant N00014-11-1-0848.

References

[1] S. Pudlewski, T. Melodia, and A. Prasanna, "C-DMRC: Compressive Distortion-Minimizing Rate Control for Wireless Multimedia Sensor Networks," *Proc. IEEE Seventh Ann. Comm. Soc. Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON),* June 2010.

[2] I.F. Akyildiz, T. Melodia, and K.R. Chowdhury, "A Survey on Wireless Multimedia Sensor Networks," *Computer Networks,* vol. 51, no. 4, pp. 921-960, Mar. 2007.

[3] S. Soro and W. Heinzelman, "A Survey of Visual Sensor Networks," *Advances in Multimedia,* vol. 2009, article number 640386, 2009.

[4] Y. Gu, Y. Tian, and E. Ekici, "Real-Time Multimedia Processing in Video Sensor Networks," *Signal Processing: Image Comm. J.,* vol. 22, no. 3, pp. 237-251, Mar. 2007.

[5] "Advanced Video Coding for Generic Audiovisual Services," ITU-T Recommendation H.264, 2010.

[6] T. Wiegand, G.J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 13, no. 7, pp. 560-576, July 2003.

[7] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammar, and T. Wedi, "Video Coding with H.264/AVC: Tools, Performance, and Complexity," *IEEE Circuits and System Magazine,* vol. 4, no. 1, pp. 7-28, Jan.-Mar. 2004.

[8] T. Wiegand, G.J. Sullivan, J. Reichel, H. Schwarz, and M. Wien, "Joint Draft 11 of SVC Amendment," Doc. JVT-X201, July 2007.

[9] Y. Wang, S. Wenger, J. Wen, and A. Katsaggelos, "Error Resilient Video Coding Techniques," *IEEE Signal Processing Magazine,* vol. 17, no. 4, pp. 61-82, July 2000.

[10] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed Video Coding," *Proc. IEEE,* vol. 93, no. 1, pp. 71-83, 2005.

[11] A. Wyner and J. Ziv, "The Rate-Distortion Function for Source Coding with Side Information at the Decoder," *IEEE Trans. Information Theory,* vol. 22, no. 1, pp. 1-10, Jan. 1976.

[12] A. Aaron, E. Setton, and B. Girod, "Towards Practical Wyner-Ziv Coding of Video," *Proc. IEEE Int'l Conf. Image Processing,* Sept. 2003.

[13] A. Aaron, S. Rane, R. Zhang, and B. Girod, "Wyner-Ziv Coding for Video: Applications to Compression and Error Resilience," *Proc. IEEE Data Compression Conf. (DCC),* pp. 93-102, Mar. 2003.

[14] T. Sheng, G. Hua, H. Guo, J. Zhou, and C.W. Chen, "Rate Allocation for Transform Domain Wyner-Ziv Video Coding without Feedback," *Proc. ACM Int'l Conf. Multimedia,* Oct. 2008.

[15] D. Donoho, "Compressed Sensing," *IEEE Trans. Information Theory,* vol. 52, no. 4, pp. 1289-1306, Apr. 2006.

[16] E. Candes, J. Romberg, and T. Tao, "Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information," *IEEE Trans. Information Theory,* vol. 52, no. 2, pp. 489-509, Feb. 2006.

[17] E.J. Candes, J. Romberg, and T. Tao, "Stable Signal Recovery from Incomplete and Inaccurate Measurements," *Comm. Pure and Applied Math.,* vol. 59, no. 8, pp. 1207-1223, Aug. 2006.

[18] E. Candes and T. Tao, "Near-Optimal Signal Recovery from Random Projections and Universal Encoding Strategies?" *IEEE Trans. Information Theory,* vol. 52, no. 12, pp. 5406-5425, Dec. 2006.

[19] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, "Compressive Imaging for Video Representation and Coding," *Proc. Picture Coding Symp.,* Apr. 2006.

[20] J. Romberg, "Imaging via Compressive Sampling," *IEEE Signal Processing Magazine,* vol. 25, no. 2, pp. 14-20, Mar. 2008.

[21] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk, "Single-Pixel Imaging via Compressive Sampling," *IEEE Signal Processing Magazine,* vol. 25, no. 2, pp. 83-91, Mar. 2008.

[22] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Processing,* vol. 13, no. 4, pp. 600-612, Apr. 2004.

[23] S.S. Hemami and A.R. Reibman, "No-Reference Image and Video Quality Estimation: Applications and Human-motivated Design," *Signal Processing: Image Comm.,* vol. 25, no. 7, pp. 469-481, 2010.

[24] S. Chikkerur, V. Sundaram, M. Reisslein, and L.J. Karam, "Objective Video Quality Assessment Methods: A Classification, Review, and Performance Comparison," *IEEE Trans. Broadcasting,* vol. 57, no. 2, pp. 165-182, June 2011.

[25] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," IETF RFC 2581, 2010.

[26] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A Compound TCP Approach for High-Speed and Long Distance Networks," *Proc. IEEE INFOCOM,* pp. 1-12, Apr. 2006.

[27] W.T. Tan and A. Zakhor, "Real-Time Internet Video Using Error Resilient Scalable Compression and TCP-Friendly Transport Protocol," *IEEE Trans. Multimedia,* vol. 1, no. 2, pp. 172-186, June 1999.

[28] M. Handley, S. Floyd, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification," IETF RFC 3448, 2010.
[29] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-Based Congestion Control for Unicast Applications," *SIGCOMM Computer Comm. Rev.*, vol. 30, pp. 43-56, Aug. 2000.
[30] O.B. Akan and I.F. Akyildiz, "ARC: The Analytical Rate Control Scheme for Real-Time Traffic in Wireless Networks," *IEEE/ACM Trans. Networking*, vol. 12, no. 4, pp. 634-644, Aug. 2004.
[31] O. Akan, "Performance of Transport Protocols for Multimedia Communications in Wireless Sensor Networks," *IEEE Comm. Letters*, vol. 11, no. 10, pp. 826-828, Oct. 2007.
[32] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of Video Transmission over Lossy Channels," *IEEE J. Selected Areas in Comm.*, vol. 18, no. 6, pp. 1012-1032, June 2000.
[33] Q. Zhang, W. Zhu, and Y. Zhang, "End-to-End QoS for Video Delivery over Wireless Internet," *Proc. IEEE*, vol. 93, no. 1, pp. 123-134, Jan. 2005.
[34] D. Wu, Y.T. Hou, and Y.Q. Zhang, "Transporting Real-Time Video over the Internet: Challenges and Approaches," *Proc. IEEE*, vol. 88, no. 12, pp. 1855-1877, Dec. 2000.
[35] V. Stankovic, L. Stankovic, and S. Cheng, "Compressive Video Sampling," *Proc. European Signal Processing Conf.*, Aug. 2008.
[36] J. Park and M. Wakin, "A Multiscale Framework for Compressive Sensing of Video," *Proc. Picture Coding Symp.*, May 2009.
[37] R. Marcia and R. Willett, "Compressive Coded Aperture Video Reconstruction," *Proc. European Signal Processing Conf.*, Aug. 2008.
[38] L.W. Kang and C.S. Lu, "Distributed Compressive Video Sensing," *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1169-1172, 2009.
[39] H.W. Chen, L.W. Kang, and C.S. Lu, "Dynamic Measurement Rate Allocation for Distributed Compressive Video Sensing," *Proc. SPIE Visual Comm. and Image Processing, Special Session on Compressed Sensing and Sparse Representation*, 2010.
[40] H.W. Chen, L.W. Kang, and C.S. Lu, "Dictionary Learning-Based Distributed Compressive Video Sensing," *Proc. Picture Coding Symp.*, pp. 210-213, 2010.
[41] T. Do, Y. Chen, D. Nguyen, N. Nguyen, L. Gan, and T. Tran, "Distributed Compressed Video Sensing," *Proc. IEEE 16th Int'l Conf. Image Processing (ICIP)*, pp. 1393-1396, 2009.
[42] A. Bruckstein, D. Donoho, and M. Elad, "From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images," *SIAM Rev.*, vol. 51, pp. 34-81, rapid post, 2007.
[43] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ., Mar. 2004.
[44] I.E. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, 1994.
[45] M. Zhu and T. Chan, "An Efficient Primal-Dual Hybrid Gradient Algorithm for Total Variation Image Restoration," Technical Report UCLA CAM 08-34, 2008.
[46] A. Graps, "An Introduction to Wavelets," *IEEE Computational Science and Eng.*, vol. 2, no. 2, pp. 50-61, 1995.
[47] S. Rein and M. Reisslein, "Performance Evaluation of the Fractional Wavelet Filter: A Low-Memory Image Wavelet Transform for Multimedia Sensor Networks," *Ad Hoc Networks*, vol. 9, pp. 482-496, http://www.sciencedirect.com/science/article/B7576-50S8PP0-1/2/fb2518d9e84aa1d10f8f88b15a857fb5, 2010.
[48] L. Gan, T. Do, and T.D. Tran, "Fast Compressive Imaging Using Scrambled Block Hadamard Ensemble," *Proc. European Signal Processing Conf.*, rapid post, 2008.
[49] M.A.T. Figueiredo, R.D. Nowak, and S.J. Wright, "Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems," *IEEE J. Selected Topics in Signal Processing*, Special Issue on Convex Optimization Methods for Signal Processing, vol. 1, no. 4, pp. 586-598, Dec. 2007.
[50] D.L. Donoho, Y. Tsaig, I. Drori, and J.L. Starck, "Sparse Solution of Underdetermined Linear Equations by Stagewise Orthogonal Matching Pursuit," *Discovery*, vol. 114, pp. 1-39, Mar. 2006.
[51] USC Signal and Image Processing Inst., http://sipi.usc.edu/database/index.html, 2012.
[52] T. Melodia and S. Pudlewski, "A Case for Compressive Video Streaming in Wireless Multimedia Sensor Networks," *IEEE COMSOC MMTC E-Letter*, vol. 4, no. 9, Oct. 2009.
[53] J. Hagenauer, "Rate-Compatible Punctured Convolutional Codes (RCPC Codes) and their Applications," *IEEE Trans. Comm.*, vol. 36, no. 4, pp. 389-400, Apr. 1988.
[54] C.E. Perkins, E.M. Belding-Royer, and S. Das, "Ad Hoc on Demand Distance Vector (AODV) Routing," IETF RFC 3561, 2010.
[55] R. Jain, D.M. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems," DEC Research Report TR-301, Sept. 1984.
[56] Arizona State Univ. Video Traces Research Group, http://trace.eas.asu.edu/yuv/index.html, 2012.
[57] M. Ettus, "Building Software Defined Radios: The USRP Product Family," Product Brochure, July 2009.
[58] The GNU Radio Project, http://gnuradio.org/trac, 2012.
[59] F. Bellard, http://www.ffmpeg.org, 2012.
[60] M. Grant and S. Boyd "CVX: Matlab Software for Disciplined Convex Programming, Version 1.21," http://cvxr.com/cvx, 2010.
[61] J. Sturm, "Using SeDuMi 1.02, a MATLAB Toolbox for Optimization over Symmetric Cones," *Optimization Methods and Software*, vol. 1, pp. 625-653, 1999.
[62] S.H. Low, L.L. Peterson, and L. Wang, "Understanding TCP Vegas: a Duality Model," *J. ACM*, vol. 49, pp. 207-235, Mar. 2002.
[63] S. Low and D. Lapsley, "Optimization Flow Control, I: Basic Algorithm and Convergence," *IEEE/ACM Trans. Networking*, vol. 7, no. 6, pp. 861-874, Dec. 1999.

Scott Pudlewski received the BS degree in electrical engineering from the Rochester Institute of Technology, New York, in 2008, and the MS degree in electrical engineering from the University at Buffalo, The State University of New York (SUNY), Buffalo, in 2010. Currently, he is working toward the PhD degree in electrical engineering while working in the Wireless Networks and Embedded Systems Laboratory under the guidance of Dr. Tommaso Melodia. His main interests include compressed sensing for video encoding and transmission, video distortion-based networking, and in wireless multimedia sensor networks (WMSNs) in general. He was the recipient of the SAP America Scholarship in 2008. He is a student member of the IEEE.

Tommaso Melodia received the Laurea (integrated BS and MS) and doctorate degrees in telecommunications engineering from the University of Rome La Sapienza, Italy, in 2001 and 2006, respectively. He received the PhD degree in electrical and computer engineering from the Georgia Institute of Technology in 2007. He is currently an assistant professor with the Department of Electrical Engineering at the University at Buffalo, The State University of New York. He coauthored a paper that was recognized as the Fast Breaking Paper in the field of computer science for February 2009 by Thomson ISI Essential Science Indicators, and a paper that received an Elsevier Top Cited Paper Award. He is an associate editor for the *IEEE Transactions on Mobile Computing*, *Computer Networks* (Elsevier), *IEEE Communications Surveys and Tutorials*, and *ACM/Springer Wireless Networks*. He has served on the technical program committees of several leading conferences in wireless communications and networking, including IEEE INFOCOM, IEEE SECON, ACM MobiCom, and ACM MobiHoc. He is the technical program committee vice chair for IEEE GlobeCom 2013 and the technical program committee vice chair for information systems for IEEE INFOCOM 2013. His current research interests are in modeling, optimization, and experimental evaluation of wireless networks, with applications to cognitive and cooperative networking, multimedia sensor networks, and underwater networking. He is a member of the IEEE.

Arvind Prasanna received the BEng degree from the Visvesvaraya Technological University, Bangalore, India, in 2008 and the MS degree in electrical engineering from The State University of New York, Buffalo, in 2011. Here, he was a research assistant at the Wireless Networks and Embedded Systems Laboratory. His research and career interests include multimedia systems design and development and in wireless sensor networks.