DMRC: Distortion-Minimizing Rate Control for Wireless Multimedia Sensor Networks

Scott Pudlewski and Tommaso Melodia
Wireless Networks and Embedded Systems Laboratory
Department of Electrical Engineering
State University of New York (SUNY) at Buffalo
e-mail: {spudlewski, tmelodia}@eng.buffalo.edu

***Abstract*—The availability of inexpensive CMOS cameras and microphones that can ubiquitously capture multimedia content from the environment is fostering the development of Wireless Multimedia Sensor Networks (WMSNs), i.e., distributed systems of wirelessly networked devices that can retrieve video and audio streams, still images, and scalar sensor data. WMSNs require the sensor network paradigm to be re-thought in view of the need for mechanisms to deliver multimedia content with a pre-defined level of quality of service (QoS).**

A new rate control scheme for WMSNs is introduced in this paper with a two-fold objective: i) maximize the video quality of each individual video stream; ii) maintain fairness in video quality between different video streams. The rate control scheme is based on both analytical and empirical models and consists of a new cross-layer control algorithm that jointly regulates the end-to-end data rate, the video quality, and the strength of the channel coding at the physical layer. The end-to-end data rate is regulated to avoid congestion while maintaining fairness in the domain of *video quality* rather than data rate. Once the end-to-end data rate has been determined, the sender adjusts the video encoder rate and the channel encoder rate based on the overall rate and the current channel quality, with the objective of minimizing the distortion of the received video. Simulations show that the proposed algorithm considerably improves the received video quality without sacrificing fairness.

I. Introduction

The availability[1] of inexpensive hardware such as CMOS cameras and microphones that can ubiquitously capture multimedia content from the environment has fostered the development of Wireless Multimedia Sensor Networks (WMSNs) [1], i.e., distributed systems of wirelessly networked devices deployed to retrieve video and audio streams, still images, and scalar sensor data [2]. WMSNs will enable new applications such as multimedia surveillance, traffic enforcement and control systems, advanced health care delivery, structural health monitoring, and industrial process control [3]. Many of these applications require the sensor network paradigm to be re-thought in view of the need to deliver multimedia content with predefined levels of quality of service (QoS).

QoS-compliant delivery of multimedia content in sensor networks is a challenging, and largely unexplored task [4] . First, embedded sensors are constrained in terms of battery, memory, processing capability, and achievable overall rate [1], while delivery of multimedia flows may be a resource-intensive task. Secondly, in multi-hop wireless networks the attainable capacity of each wireless link depends on the interference level perceived at the receiver. Hence, capacity and delay attainable at each link are location dependent, vary continuously, and may be bursty in nature, thus making QoS provisioning a challenging task. Lastly, functionalities handled at different layers of the communication protocol stack are inherently and strictly coupled due to the shared nature of the communication channel. Hence, different functionalities aimed at QoS provisioning should not be treated separately when efficient solutions are sought, i.e., a cross-layer design approach is needed [5], [6], [7], [8], [9].

In this paper, we consider a multi-hop wireless network of video sensors deployed for surveillance applications and turn our attention to reliable transport of video traffic. The objective is to devise mechanisms to *efficiently* and *fairly* share the common network infrastructure among the flows generated by different video sensors, to deliver high-quality video on resource-constrained devices. To achieve this objective, we propose the Distortion-Minimizing Rate Control (DMRC), a new distributed cross-layer control algorithm that jointly regulates the end-to-end data rate, the video quality, and the strength of the channel coding at the physical layer to minimize the *distortion* of the received video. The end-to-end data rate is chosen to avoid congestion while maintaining fairness in the domain of *video quality* rather than data rate. Once the end-to-end data rate has been determined, the sender determines the optimal proportion of video encoder rate and channel encoder rate based on the overall rate available and on the current quality of the wireless channel on the source-destination path, with the objective of minimizing the distortion of the received video.

Video stream distortion in wireless networks is mostly caused by lossy source coding at the source, transmission errors caused by channel fading, buffer overflows and playout deadline misses. If the loss happens at a relay node due to congestion, then the video encoder rate has to be decreased smoothly to reduce congestion. In case packets are being lost due to correlated fading on the wireless link, the video encoder rate should remain unchanged and the channel encoder rate can be reduced. The channel encoder rate can then be increased as the wireless channel errors decrease. In DMRC, the signal

[1]This work was supported by the U.S. Air Force Research Laboratory under Grant FA8750-08-1-0063.

978-1-4244-5113-5/09/$25.00

to noise ratio (SNR) and the round trip time (RTT) are used to determine what is causing the distortion at the receiver. By using feedback packets, the receiver updates the sender with the current forward channel information including SNR and RTT. This allows the sender to correctly react to the cause of packet errors.

Unlike other current cross-layer optimization techniques [7], [8], [10], [11], [12], [13], the proposed scheme minimizes the video distortion by finding the optimal ratio of video encoder rate to channel encoder rate. Furthermore, the control algorithm finds the best possible transmission rate for a network sending primarily video. Differently from previously proposed schemes such as TCP-Friendly Rate Control (TFRC) [14], we will not take fairness towards TCP as a key design principle. In a resource-constrained WMSN, priority must be given to the delay-sensitive flows, possibly at the expense of other delay-tolerant data. Furthermore, TCP traffic is unlikely to be simultaneously transmitted in a sensor network. By basing the data rate on the video compression rate and by jointly optimizing video coding and channel coding, DMRC uses network resources more efficiently than TFRC, resulting in higher video quality at the receiver.

The remainder of this paper is structured as follows. In Section II, we discuss previous work on related topics. In Section III, we introduce the considered system model. In Section IV, we describe our solution to the determination of the video encoder rate and the channel encoder rate, and in Section V we describe the workings of the DMRC rate controller. In Section VII we discuss performance evaluation results while in Section VIII we draw the main conclusions.

II. Related Work

The most common form of rate control is the well-known transmission control protocol (TCP) [15]. Because of the additive increase/multiplicative-decrease algorithm used in TCP, the rate that it determines is not smooth enough for high quality video transfer [16]. In addition, TCP assumes that any lost packets are due to congestion. Although this assumption is reasonable for wired networks, for wireless networks channel errors must be taken into account if an accurate prediction of the network congestion is needed. For example, in [17] it was experimentally showed how in sensor networks packets are frequently dropped because of channel errors even on short-distance links. Because of this, very few links can be considered error-free and the effect of packet drops due to channel loss has a large impact on the video quality.

These considerations have led to a number of equation-based rate control schemes. Equation-based rate control analytically regulates the transmission rate of a node based on measured parameters such as the number of lost packets and the round trip time (RTT) of the data packets. Two examples of this are the TCP-Friendly Rate Control [14], which uses

$$X = \frac{S}{R\sqrt{\frac{2\pi}{3}} + \left(4R\left(3\sqrt{\frac{2\pi}{8}} \cdot \pi \cdot (1 + 32 \cdot \pi^2)\right)\right)}, \tag{1}$$

i.e., the throughput equation of TCP Reno [15], and the Analytical Rate Control (ARC) [18], which uses

$$X = \frac{S}{4 \cdot R} \cdot \left(3 + \sqrt{25 + 24\left(\frac{1-\omega}{\pi - \omega}\right)}\right). \tag{2}$$

In (1) and (2), $X\,[\mathrm{bit/s}]$ represents the transmission rate, $S\,[\mathrm{bit}]$ is the packet size, $R\,[\mathrm{s}]$ is the round trip time (RTT), π is the loss event rate, and ω is the probability of being in a lossy state. Both of these schemes attempt to determine a source rate that is fair to any TCP streams that are concurrently being transmitted in the network. 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 [19], the authors analyze the video distortion over lossy channels of MPEG encoded video with both inter-frame coding and intra-frame coding. A factor β is defined as the percentage of frames that are an intra-frame, or I frame, i.e., a frame which is independently coded. The authors then derive the value β that optimizes distortion at the receiver. Similar to our work, [19] investigates optimal strategies to transmit video with minimal distortion. However, the authors assume that the I frames are received correctly, and that the only loss is caused by the inter-coded frames. We take the idea a step further and assume that any packet can be lost. Also, we jointly optimize the video coding and the channel coding, which will lead to a better overall performance.

Cross layer design techniques to transmit video over wireless networks are also addressed in [20], where the authors attempt to minimize the video distortion by optimizing the code division multiple access (CDMA) coding parameters, the video encoder rate, and the channel encoder rate. This paper focuses specifically on CDMA channels and uses the *operational rate-distortion functions* (ORDF) for each scalable layer to determine the distortion. Conversely, we provide a more general solution that is independent of the underlying MAC protocol and of the specific video source encoding scheme. To address the multiple rates at the source, the originating node will alter the number of bits per pixel used in the video encoder, thereby changing the rate at the expense of the received video distortion. If a more specific transmission technology were to be considered, our approach could be extended to include characteristics of the receiver as in [20]. Finally, our previous work has investigated cross-layer joint routing, scheduling, and channel coding for WMSNs based on the time-hopping impulse radio ultrawide band (UWB) transmission technique [21]. However, transport-layer issues and video quality related metrics were not addressed.

III. System Model

A. Channel Coding

The proposed system includes a *channel encoder* block that adds redundancy to combat channel fading. The channel encoder at node i receives a block of L uncoded bits, selects the encoding rate $R_{C,i}$, which represents the number of data bits per encoded bit, among the set $\underline{R}_C = [R_C^1, R_C^2, \cdots, R_C^{MAX_\rho}]$, with $R_{C,i}^1 = 1$ (i.e., transmitting uncoded data), $R_C^1 > R_C^2 > \cdots > R_C^{MAX_\rho}$ and where MAX_ρ is the total number of available codes in the family. Hence, when code $R_C^{MAX_\rho}$ is selected, i.e., when $R_{C,i} = R_C^{MAX_\rho}$, the encoder produces a block of coded bits of length $L/R_C^{MAX_\rho}$. The set of available codes $\underline{R}_C$ depends on the chosen family of codes $\mathcal{C}$. Different families of codes, such as BCH codes [22] or rate-compatible punctured codes (RCPC) [23], have different performance and different levels of complexity.

In this paper, we consider RCPC codes. Specifically, we use the $\frac{1}{4}$ mother codes discussed in [23]. Briefly, a $\frac{1}{4}$ convolutional code is punctured to decrease the amount of redundancy needed for the encoding process. These codes are punctured progressively so that every *higher rate* code is a subset of the lower rate codes. For example, any bits that are punctured in the $\frac{4}{15}$ code must also be punctured in the $\frac{1}{3}$ code, the $\frac{4}{9}$ code, and so on down to the highest rate code, in this case the $\frac{8}{9}$ code. Because of this setup, the receiver can decode the entire family of codes with the same decoder. This allows the transmitter to choose the most suitable code for the given data. 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 bit error rate for transmission rate.

B. Video Distortion Model

We define distortion as the average peak signal-to-noise ratio (PSNR) of the received video. Since we are interested in the amount of distortion introduced by the video transmission, we use the difference between the PSNR actually obtained from the video transmission scheme and the best possible PSNR obtainable from the technology (i.e. no packets are lost and the highest possible rate MPEG encoder is used to encode the video frames). PSNR is then defined as

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right), \tag{3}$$

where MAX_I is the maximum possible pixel value for each frame. MSE is the mean squared error, which is defined as

$$MSE = \frac{1}{mn}\sum_{i=1}^{m-1}\sum_{j=1}^{n-1}\|I(i,j) - K(i,j)\|^2 \tag{4}$$

for any two $m \,\mathrm{x}\, n$ images I and K where one of the images can be considered to be a noisy approximation of the other. To extend this to video distortion rather than image distortion, we take the PSNR measurement for each frame and average over all of the frames in the video. For any frames that are dropped or unable to be decoded at the receiver, the previous frame in the received video is measured against the current frame in the ”good” video (i.e. the video before it was transmitted) and this value is used in the average.

For real-time video streaming applications, video packets have to reach the decoder within a predefined latency bound, referred to as the *playout deadline*. The video quality at the receiver is therefore affected by two major factors, i.e., distortion caused by lossy encoding and distortion caused by loss of packets. Hence, the distortion D^m of a video stream m can be expressed as

$$D^m = D_{enc}^m + D_{loss}^m, \tag{5}$$

where D_{enc}^m represents distortion introduced by lossy encoding while $D_{loss}^m = f(PER^m)$ is the distortion introduced by loss of packets, and is thus a function of the end-to-end packet error rate PER^m for video stream m. The latter is in turn the sum of several components, i.e.,

$$PER^m = PER_{loss}^m + PER_{delay}^m. \tag{6}$$

In the above expression, PER_{loss}^m represents the percentage of packets lost due to impairments of the wireless channel, along with packets dropped at intermediate relay nodes caused by congestion (buffer overflows). PER_{delay}^m represents packets dropped since they miss the playout deadline. Now, the physical layer data rate R^m for a video stream m can be expressed as:

$$R^m = \frac{R_V^m}{R_C^m}, \tag{7}$$

where R_V^m, R_C^m and R^m are defined as follows.

Definition 1: R^m [kbit/s] is the *total overall rate* available for a video stream m as decided by the rate control algorithm. R_V^m [kbit/s] is the *video encoder rate* defined as the rate of the compressed video generated by the video source. R_C^m [bitsin/bitsout] is the *channel encoder rate* defined as the rate of the RCPC encoder.

Given a fixed data rate R^m, we need to determine the video encoder rate and the channel encoder rate. Clearly, a lower video encoder rate increases D_{enc}^m, i.e., the portion of distortion introduced by lossy encoding. However, it leaves room for a more redundant channel encoder rate, which may decrease losses due to channel impairments and hence reduce D_{loss}^m. Conversely, a higher video encoder rate may decrease D_{loss}^m but increase D_{loss}^m. Hence, the objective of the proposed cross-layer controller is, for each stream m, to jointly determine the data rate at the physical layer R^m, the rate of the source coder R_V^m, and the channel coding rate R_C^m in such a way as to minimize the perceived distortion at the receiver.

IV. Optimal Video and Channel Encoder Rates

The optimal ratio of the video encoder rate and the channel encoder rate must be determined to minimize video distortion. First, we will examine the effect of each of the rates on the video distortion by presenting an analytical model of the system. Then, we will present an algorithm to determine how to divide the overall rate between the video encoder

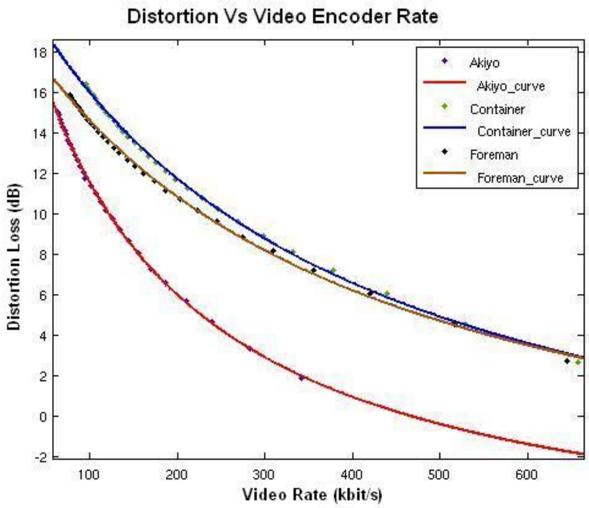

Fig. 1. Regression curves used to determine distortion due to JPEG encoding

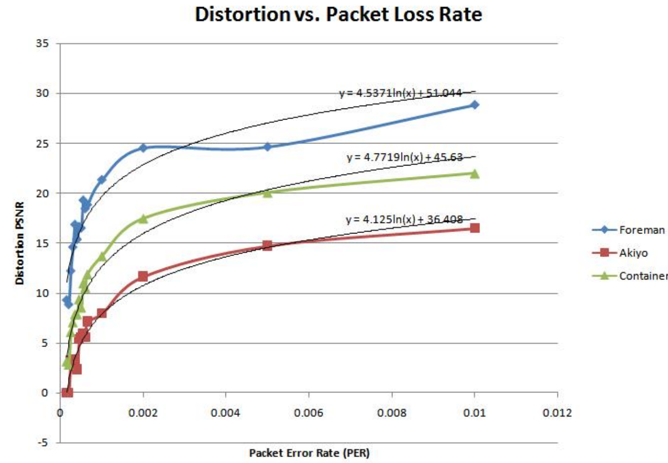

Fig. 2. Regression curves used to determine distortion due to packet error rate (PER).

and the channel encoder. Finally, we will find the scheme for transmitting video that will result in the least amount of distortion at the receiver. The tools used to do this are ffmpeg [24], Evalvid [25] and Evalvid-RA [26].

The term D^m_{enc} in (5) refers to the amount of distortion introduced by the video encoder. The type of video encoding used in the paper is simple Motion Picture Experts Group (MPEG) video encoding [27]. Our methodology is however general and can be extended to any video encoding scheme. The first frame of the video is encoded according to a Joint Photographic Experts Group (JPEG) [28] lossy compression scheme. The amount of loss introduced by this portion of the compression is determined by the amount of compression applied to the frame, and can be set by the node in order to alter the rate of the video. For a group of pictures (GOP) after the first frame, the encoder uses motion estimation to determine the difference between the first frame and the preceding frame. It then only encodes the difference between the two frames and transmits this difference, which is generally much less data than is needed to transmit the entire frame. For more information about MPEG video compression, the reader is referred to [27].

Similar to [19], a three parameter empirical model as in (8) can be used to represent the amount of distortion introduces by the MPEG encoder:

$$D^m_{enc} = D_0 + \frac{\theta}{R^m_{source} - R_0}, \tag{8}$$

where R^m_{source} is the video encoder rate and the parameters D_0, θ and R_0 can be estimated from empirical rate-distortion curves via a linear least squared curve fitting [29]. The curves for three different videos, namely Akiyo, Foreman and Container from [30] are shown in Fig. 1. Our methodology can be extended to higher order models, but the three parameter model was chosen because of its relatively low complexity and good accuracy. To use this empirical model for real-time control of video streaming, this curve fitting has to be done in real time at the sender. Although the sender would not have information about the entire video on which to base its measurements, it can use past video information. How often the sender would have to make these measurements depends on the variation in the video being sent. For example, most surveillance videos have very little variation from one minute to the next, or from one hour to the next. For such an application, the curve fitting could perhaps only be done once every ten minutes. In something which varied more often, such as aerial video from a search and rescue operation, the curve fitting would have to be done more often. Note that this operation can fastly be performed on current high-performance platforms for multimedia sensing such as the iMote2, which is built around an integrated wireless microcontroller consisting of the new low-power 32-bit PXA271 Marvell processor, which can operate in the range $13 - 416\,\text{MHz}$ with dynamic voltage scaling [3].

The term D^m_{loss} in (5) refers to the amount of distortion introduced into the video by packet loss, and is more difficult to determine. First, we assume that the mean-squared error (MSE) varies linearly with the packet loss. Since the dependence between distortion and MSE is logarithmic as in (3), we will express the distortion caused by the packet loss as

$$D^m_{loss} = a \cdot \ln(PER^m) + b, \tag{9}$$

where the parameters a and b can be estimated from empirical rate-distortion curves via regression techniques. In this case they were determined through linear least square [29] curve fitting. Three curves showing the derivation of a and b are shown in Fig. 2. The packet error rate PER^m is shown in (6) to be made up of two different components. The first, PER^m_{loss}, represents the packets lost due to channel errors. The second, PER^m_{delay}, represents losses due to packets arriving at the receiver after the playout delay deadline. We will examine each of these individually.

Assuming a single sender and receiver, we can assume that the probability of a packet successfully reaching the destination in an n hop path is

$$P_{success} = P_n(1 - P_{n-1})(1 - P_{n-2}) \cdots (1 - P_1) \quad (10)$$

Where P_n is the probability of a successful transmission at the n^{th} hop. PER^m_{loss} can be expressed as

$$PER^m_{loss} = 1 - P_{success}, \quad (11)$$

which represents the packets lost along the entire forward path. The success probability P_n in (10) depends on the bit error rate of the link, which in turn depends on the modulation scheme, on the (SNR) and on the packet length. Furthermore, P_n will be affected by the error correcting capabilities of the RCPC code being used for that packet. The effect of the channel coding can be found by assuming that the code alters the apparent SNR at the receiver by (12),

$$SNR = d_{free} \cdot R^m_{C,RCPC} \cdot \frac{E_b}{N_0} \quad (12)$$

where d_{free} is the free distance of the code and $R^m_{C,RCPC}$ is the rate of the code ($R^m_{C,RCPC} < 1$) [31]. This equation allows the sending node to use the RCPC codes to directly affect the packet error rate for each video packet being sent, as long as it receives channel state information about the forward path. This can be done by embedding this information in the rate control packets. This will be discussed further in Section VI.

The packet loss caused by the delay is found by assuming that the end-to-end delay of the packets can be modeled as a Gaussian random variable $PER^m_{delay} \sim \mathcal{N}(\mu_\epsilon, \sigma^2_\epsilon)$ where ϵ is the forward trip time, μ_ϵ is the mean of ϵ and σ^2_ϵ is the variance of ϵ. Then, the ϵ information can be used to calculate the distribution of the delay at the current time in the network for a specific video flow using

$$PER^m_{delay} = Q\left(\frac{T_{playout} - \mu_\epsilon}{\sigma^2_\epsilon}\right) \quad (13)$$

which represents the Q function, defined as

$$Q(\alpha) \triangleq \frac{1}{\sqrt{2\pi}} \int_\alpha^\infty e^{-\frac{x^2}{2}}\, dx. \quad (14)$$

Though there are other more accurate delay models available, such as [32], [33], [34], they would be computationally infeasible for implementation on a WMSN node.

Figure 3 shows the distortion loss $PSNR_{loss}$, as defined as

$$PSNR_{loss} = PSNR_{optimal} - PSNR_{achieved}, \quad (15)$$

against the product of R^m_V and R^m_C from (7). In (15), $PSNR_{optimal}$ is the $PSNR$ obtained using perfect channels, no buffer overflows and the best available video encoder rate. $PSNR_{achieved}$ is the PSNR of the video using the rate constraints and including channel losses. Since the overall rate in (7) is defined as the ratio of R^m_V and R^m_C, we examine the product in order to determine the relative importance of the two factors. To generate the curves in Fig. 3, the distortion loss as in (15) was plotted for a three-hop path. The SNR, overall data rate and RTT were all set to fixed values. This makes it possible to isolate the effect that the proportion of the overall rate allowed for the two encoders (video encoder and channel encoder) has on the received video distortion. The distortion was then averaged over a number of different RCPC codes and videos to remove the dependency on these specific factors from the analysis.

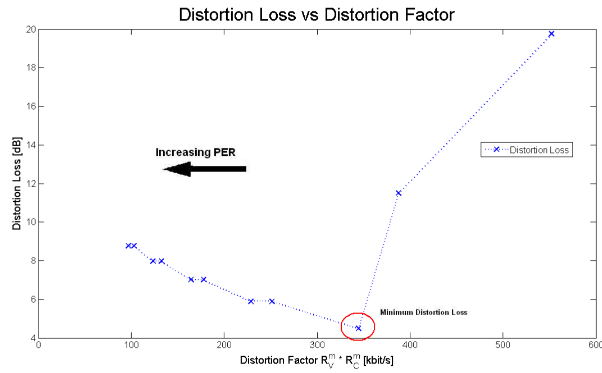

Fig. 3. Distortion Loss vs Distortion Factor.

Since the overall rate in (7) is a fraction of the video and channel encoder rates, the product $R_V \cdot R_C = R \cdot R^2_C$ essentially gives us the ratio of the two rates where the far left of the x-axis corresponds to the majority of the rate assigned to the video encoder, and the far right corresponds to the majority of the rate assigned to the channel encoder. The curve shown in Fig. 3 represent a fixed overall rate of $6.4\,\mathrm{Mbit/s}$. The figure clearly shows that there is a minimum distortion region along the curves. As the packet error rate is decreased beyond a certain point, the gain from the extra packets at the receiver does not make up for the distortion created by the loss of data rate available for the video encoder. Essentially, this minimum point shows us the ideal PER for this specific set of SNR, R^m, and RTT. Also, it is clear that moving right from the minimum point results in a more dramatic increase in distortion compared to moving to the left. This is due to the logarithmic nature of the relationship between distortion and PER^m in (9), compared to the near-polynomial relationship to the video encoder rate in (8). The problem of determining the minimum point in Fig. 3 can be formulated as an optimization problem, solved with Algorithm (1) below. In this problem, it is assumed that the SNR and the overall rate R^m are given. It is also assumed that there are no local minima to the right of the minimum point, which is supported by our experimental results. With R^m set to a fixed value, we find the optimal value for the vector Λ where Λ is the ordered vector (ascending) $[\lambda_0, \lambda_1, \cdots, \lambda_{max}]$. The elements $\lambda_n \in \Lambda$ represent the possible values of $R^m_V \cdot R^m_C$. Since both R^m_V and R^m_C can only take on discrete values, Λ will have finite dimension. In Algorithm (1), the variable Δ is used to determine whether the distortion loss ($D^m_{\lambda_n}$) obtained by using two consecutive values λ_i and λ_{i+1} is increasing or decreasing. As soon as Δ is found to be increasing, the values of R_V^m and R_C^m corresponding to that value of λ are returned.

$$\begin{array}{rl} \mathbf{P1}: Given: & R^m,\ SNR,\ PER_{delay}^m \\ Find: & R_V^m,\ R_C^m \\ Minimize: & D_{enc}^m + D_{loss}^m \\ Subject\ to: & R_V^m \in \mathcal{R_V},\ R_C^m \in \mathcal{R_C} \end{array}$$

Algorithm 1 Distortion Minimization

$\Delta = -1,\ n = 0$
while $\Delta < 0$ **do**
  for $\lambda_n \in \Lambda$ **do**
    $\Delta = D_{\lambda_n}^m - D_{\lambda_{n+1}}^m$
    $n = n + 1$
  end for
end while
Return solution as $[R_V^m R_C^m]$ corresponding to λ_n

V. Distortion-minimizing Rate Control (DMRC)

In this section, we present a rate control scheme that adjusts the overall data rate at each node in order to obtain fairness in the distortion of all video flows transmitted through the network. The main objectives of our cross-layer rate controller are: i) maximize the video quality of each individual video stream; ii) maintain fairness in video quality between different video streams. This is done by using the estimated receiver video quality (calculated at the sender) as the main factor in the rate decision rather than the data rate of the video.

A. Transmission Rate

Any video source in a WMSN needs to take two factors into account when determining its transmission rate. First, the sender has to make sure that it is allowing any other nearby transmissions at least as much bandwidth as it needs to attain a comparable video quality as itself. Second, the sender needs to make sure that packet losses due to buffer overflows and channel errors are reduced to an acceptable level, which can be done by reducing the overall data rate if it increases to a level which the network can not handle.

To understand why fairness is important, imagine that this system is used in a video security system. Since there is no way to determine ahead of time where an interesting event may be occurring, any of the transmitted videos may be recording something which would help solve or prevent a crime. If there is no fairness between video streams, this stream may have such poor quality that it isn't good enough to even decode, let alone be used for anything useful.

The transmission rate control decision will be based on the measured round trip time (RTT), the current overall transmission rate R^m and the distortion caused by the current video encoder rate. The maximum rate $R_{V,MAX}^m$ is the rate at which the video is encoded with the least possible distortion, and the channel encoder is using the highest rate RCPC code available. Because the video compression is based on the amount of distortion in the video rather than the size of the resulting video frames, $R_{V,MAX}^m$ can be different for each video, motivating us to use the amount of distortion caused by the video encoding as the rate control criteria rather than the rate of the video. This can be seen with the videos *Akiyo* and *Foreman* [30]. Because *Foreman* has much more variation than *Akiyo*, *Akiyo* can be compressed much more while keeping the same video quality. For example, we used ffmpeg to encode raw YUV video with the highest quality MPEG compression using only inter-frame encoding directly from two same sized, 174 x 144 (QCIF) video sources, *Foreman* and *Akiyo*. Even though the raw data for both videos is $44,500\,\mathrm{kbyte}$, the best quality for *Akiyo* can be obtained with a $4,641\,\mathrm{kbyte}$ file with $45.88\,\mathrm{dB\,PSNR}$, while *Foreman* needs $8,980\,\mathrm{kbyte}$ to achieve $44.07\,\mathrm{dB\,PSNR}$.

As the traffic in the network changes, the average RTT will also change with it. By measuring RTT dynamics, it is possible for a sender to predict how much other traffic is in the network which could interfere with its video transmission. If the traffic is increasing, the RTT will increase and therefore the sender can decrease its own video encoder rate in order to keep the network stable. Similarly, if the traffic is decreasing, the RTT will decrease and the sender can increase the overall rate. Finally, packet losses, both due to channel loss and buffer overflow, are taken into account by using the value RTT_{lost_pkt} for the RTT of the missing packet, where RTT_{lost_pkt} is an arbitrarily high number (e.g., the playout deadline). This will cause the average RTT measurement to increase and the overall data rate at the sender to decrease.

To determine the overall rate R_i at any decision period i, the video source uses

$$R_i = \begin{cases} R_{i-1} - \frac{1}{\alpha \cdot \Delta RTT} & \text{if} \quad \widetilde{RTT_t} > \widetilde{RTT_{t-1}} \\ R_{i-1} + \frac{1}{\beta \cdot \Delta RTT} & \text{if} \quad \widetilde{RTT_t} \leq \widetilde{RTT_{t-1}}, \end{cases} \tag{16}$$

where

$$\alpha = \alpha_0 \cdot DR \tag{17}$$

$$\beta = \beta_0 \cdot DR. \tag{18}$$

In (16), $\widetilde{RTT_t}$ represents the weighted average of the previous N RTT measurements, and is defined as

$$\widetilde{RTT_t} = \frac{\sum_{i=1}^{N} A_i \cdot RTT_{t-i}}{N \cdot \sum_{i=1}^{N} A_i} \tag{19}$$

The value N is the number of RTT measurements that are considered, and A_i represents the weight of the $(t-i)^{th}$ measurement. By weighting the average, more emphasis can be given to the more recent RTT values allowing the rate to adapt faster to changes in the network. DR stands for the *distortion ratio*, defined below.

Definition 2: The *distortion ratio* is a measurement of video distortion based on the source coding rate and the channel coding rate and is defined as:

$$DR = \begin{cases} \Delta\rho + \sigma & \text{if} \quad \widetilde{RTT_t} > \widetilde{RTT_{t-1}} \\ (MAX_\rho - \Delta\rho) + MAX_\sigma - \sigma & \text{if} \quad \widetilde{RTT_t} < \widetilde{RTT_{t-1}}, \end{cases} \tag{20}$$

where $\Delta\rho$ represents the difference between the index of the current RCPC code and the index of the RCPC code required to achieve the desired packet loss ratio and MAX_ρ is the number of RCPC codes available. Similarly, σ represents the index of the current coding rate. The maximum index of the coding rate (i.e., the index corresponding to the worst possible rate) is represented by MAX_σ. Constants α_0 and β_0 are used to enforce a smooth change in rate and to help the network reach a steady state that will allow equal video quality for all video streams.

For example, consider a variable rate channel encoder with MAX_ρ available channel codes (as in Section III-A) where the lower index indicates the more redundant channel code. In this scenario, the optimal index ρ_{opt} will be the index of the channel code which minimizes the curve in Fig. 3. If for any reason the currently used channel coding rate ρ is higher than ρ_{opt} (i.e. the sender is using a lower rate code), then $\Delta\rho$ would indicate the distance $\rho_{opt} - \rho$. After ρ_{opt} is chosen, the optimal value for σ is always going to be the index corresponding to the highest video encoding rate allowed by the overall rate R^m. Again, assume that there are a set of video encoding rates $R_V^1 > R_V^2 > \cdots > R_V^{MAX_\sigma}$ where R_V^1 represents the highest video encoder rate (lowest amount of compression) and MAX_σ the lowest video encoder rate (most compression). Using (20), there are two cases where DR will be high. If $\widetilde{RTT_t} > \widetilde{RTT_{t-1}}$, the overall rate will be decreasing according to (16). If the source node is sending video where both $\Delta\rho$ and σ are low, meaning the video is being sent at a high rate with very little compression and at or near the optimal channel coding, the distortion ratio DR will be high. Also, if $\widetilde{RTT_t} > \widetilde{RTT_{t-1}}$ and the video is being sent with a high $\Delta\rho$ and σ, DR will also be high. In the opposite cases, DR will be low. When DR is large, the rate changes according to (16) will be small. These cases are when the source node is sending video at optimal encoding parameters and the rate should increase, or when the source node is sending video with bad encoding parameters and the rate is decreasing.

By using the distortion ratio as the basis for DMRC, the rate changes will be based on the estimated received video quality along with the actual rate of the video. For example, if a node is sending video in a network with very little traffic, it will send video at or near the maximum rate. If another sender/receiver pair start sending another video nearby, the original sender will be able to detect this by an increase in the value of $\widetilde{RTT_t}$. Since the first sender is already sending video at a very high rate, it can afford to lower the sending rate of its video stream without severe loss in received video quality. In this case, the distortion ratio will be low according to (20), which will therefore magnify the change in overall rate determined by (16). If, however, the original sending node is sending video at a very low rate originally, the distortion ratio will be high, and the rate will only decrease by a small amount. The opposite is true if a node detects a decrease in $\widetilde{RTT_t}$. If the video quality for this node is already high, there is no reason to increase the overall rate by a large amount. However, if the video is being sent with very poor quality, the sender will take advantage of the decrease in $\widetilde{RTT_t}$ and raise the overall rate much more dramatically.

The main advantage to this algorithm is seen in the case where one node is sending a video with a much higher data rate than another node. If the rate control system is based simply on the bit rate of the video, then the smaller video (e.g. *Akiyo*) will always be received with higher quality than the larger video (e.g. *Foreman*). However, by using the video quality as a decision metric, the sender of the smaller video will not continue to increase its rate beyond the point of achieving a reasonably high distortion is the RTT values indicate that another node is attempting to transmit a video.

VI. Cross Layer Implementation

This scheme depends on two sets of information being available at the sending node; SNR values measured along the forward path, and the RTT. Both of these sets of information can be easily made available at the source by adding information to the data packets sent to the receiver and the response packets sent at the transport layer.

The SNR can be simply measured at the physical layer along the forward path. The receiver along the path with the lowest SNR measurement records that information in the packet (i.e., if the current SNR is lower than what is already recorded in the packet, replace it with the current measurement). When the data reaches the receiver, this information is stripped out of the packet at the transport layer and sent back to the sender in the next feedback packet. The receiver adds the most recent forward path SNR measurement to the packet and sends it back to the source. By keeping the most recent SNR readings, the sender can get an accurate estimation of the forward path channel. The worst SNR is used because the hop with the worst signal strength is the most likely place for the packet to get dropped. The RTT measurements are calculated by adding time stamps to both the data packet and the response packet.

When a new data packet is created, the video source can read the current information about the channel and the delay from the transport layer. The probability of packet loss due to delay can be calculated from the RTT values. Then, the type of coding necessary to reach the ideal PER^m can be easily calculated based on the type of modulation currently in use at the physical layer, the SNR readings, and the loss already expected from delay. After that, the video source encodes each frame with the highest possible encoding rate given the overall transmission rate allowed at the node and the amount of that rate needed for the channel encoder. The video packets which are then encoded and transmitted will give the lowest delay possible given the current network conditions.

VII. Performance Evaluation

The DMRC algorithm was evaluated using the ns-2 version 2.33 network simulator along with the Evalvid [25] toolset. All simulations are done using 49 nodes in a 7 x 7 grid. The sender/receiver pairs are chosen randomly from the set based on the random number generator in ns-2, and all simulations were run 20 times with different random number seeds. Considering a 95% confidence interval, a relative error within 10% was observed for all data points. A CSMA/CA medium access control was considered and end-to-end routes were established based on AODV [35].

The frames were compressed at multiple compression rates using ffmpeg [24]. A text trace of each video file was created including the size of each frame and the time from the start of the video that the frame occured. This text information was then included within the data field of the packets in the simulation, based on which level of compression was decided upon at the sending node. As each packet was sent and received, a trace file was generated indicating the time segment number along with which compressed video was being sent for each frame. After each simulation, the video file was reconstructed using the Evalvid software, expanded into uncompressed video using ffmpeg, and compared to the original uncompressed source file again using the Evalvid software. The files used for simulation were *Highway*, *Bridge-Close*, *Bridge-Far*, and *Foreman* from [30]. These videos were chosen because they were each approximately of the same length, while their compressed sizes were different [36].

The first set of simulations were carried out to determine the advantage of joint simultaneous selection of both the video encoding and the channel encoding. This was done by setting the video encoder rate to a fixed value while allowing the channel encoder rate to vary, and then repeating for all available video compression rates. Thus, all available static video compression rates were tested to determine whether any of them performed better than our scheme selecting jointly both rates. Figure 4 shows the results for these simulations. The solid horizontal line represents the average PSNR of all simulations using the same topology and source/destination pairs but using the scheme described in Section IV. It can clearly be seen that on average the video quality is better with joint selection of video encoder rate and channel encoder rate regardless of what static level the video encoder is set at. The same simulations were performed keeping the channel encoder fixed and varying the video encoder rate, the results of which are reported in Fig. 5. Again, the video quality is always higher when jointly selecting both video and channel encoder rates as described in Section IV than with any fixed channel encoder rate. Then, the DMRC rate controller was compared to TFRC as described in [14] using two different sets of simulations. First, the number of simultaneous sender/receiver pairs was varied between two and six. This was done to compare the performance of the two protocols as the amount of congestion in the network varied. The results are shown in Fig. 6. In this figure, the received video quality for all videos was averaged

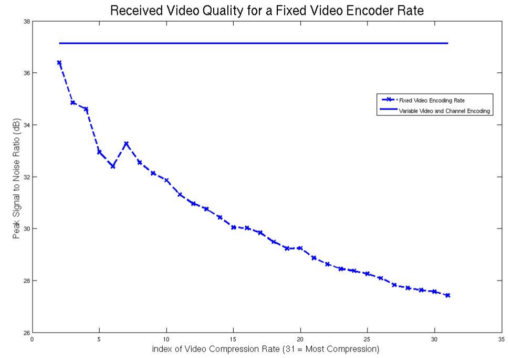

Fig. 4. Video Quality of Fixed Video Encoder vs Variable Video and Variable Channel Encoders.

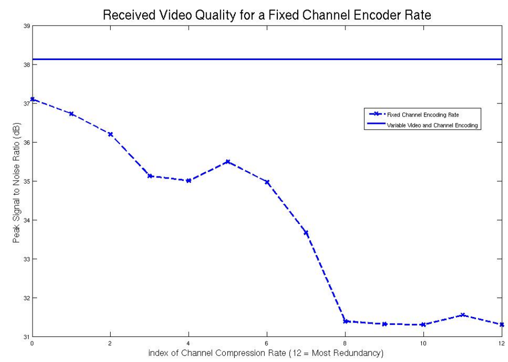

Fig. 5. Video Quality of Fixed Channel Encoder vs Variable Video and Variable Channel Encoders.

at the receiver. It can be seen that in all cases, DMRC results in higher received video quality than TFRC. This is confirmed by Fig. 7, which traces the PSNR against time for the video *Highway* through one simulation. Clearly, DMRC consistently outperforms TFRC.

Second, three different videos were streamed at the same time and the distortion of each video was observed, as shown in Fig. 8. In both cases, the video quality was higher for the simulations using the distortion minimizing rate control (DMRC) when compared to TFRC. This is because DMRC is less conservative than TFRC, so the average rate is higher allowing both stronger channel coding and higher video coding rate. This can also be seen in the changes in the DMRC distortion curve in Fig. 6. As the number of sender/receiver pairs increases, the video quality for the DMRC curve decreases, while the distortion for TFRC remains nearly constant. This is because the conservative rate determined by TFRC still leaves room for other videos to be sent, while the DMRC rate is closer to the physical maximum allowed by the channel. As more videos are added into the network, the rate for each video in DMRC must decrease.

Figure 9 shows four screen shots from the *Foreman* video received using DMRC (top) and TFRC (bottom), i.e., at roughly 10 dB difference in PSNR. Finally, Jain's Fairness

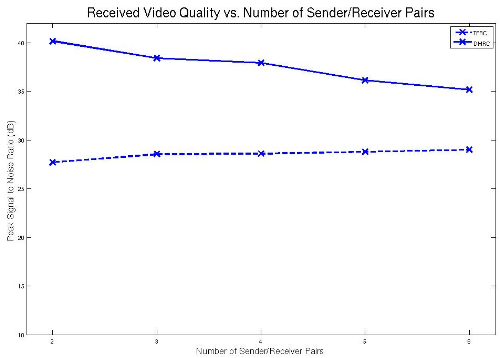

Fig. 6. Received Video Quality Vs. Number of Sender/Receiver Pairs.

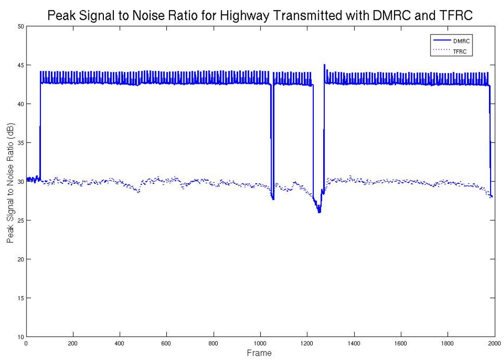

Fig. 7. Received Video Quality Vs. Time or *Highway* Transmitted with DMRC and TFRC.

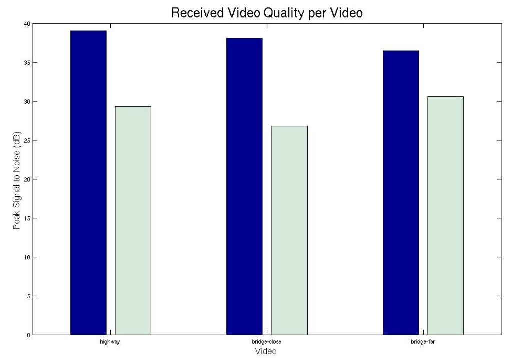

Fig. 8. Received Video Quality of Different Video for DMRC and TFRC.

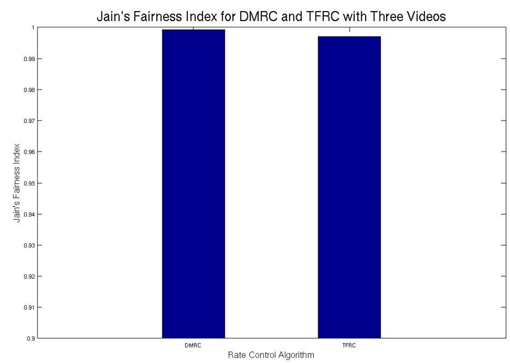

Fig. 10. Jain's Fairness Index for DMRC and TFRC.

Index [37], defined as

$$f(x_1, x_2, \ldots, x_n) = \frac{\left(\displaystyle\sum_{i=1}^{n} x_i\right)^2}{n \cdot \displaystyle\sum_{i=1}^{n} x_i^2}, \qquad (21)$$

was used to assess the fairness in terms of *received video distortion*. The results are reported in Fig. 10. In both cases, the index was near one, which indicates very high fairness between the three videos, with DMRC slightly outperforming TFRC.

VIII. Conclusions and Future Work

In this paper, a rate control scheme is introduced, along with a joint video and channel encoder rate allocation scheme. The rate allocation scheme is based on both analytical and empirical models, and finds the combination of video and channel encoding that results in the best quality video at the receiver. The algorithm presented is simple enough to run in real time on a WMSN node. Simulation results support that the proposed system results in better quality video than varying either of the encoders individually Furthermore, the DMRC rate control scheme is presented, which bases rate control decisions on the quality of the video being sent. This was compared to TFRC in terms of the received video quality. Our results show that the rates decided on by DMRC result in higher-quality video than those selected by TFRC.

We are currently working on implementing the algorithms presented in this paper on the iMote2 platform [38] with multimedia sensing board to experimentally verify our simulation results and to ensure that the scheme can be implemented in real-time on a resource-constrained device. We also intend to extend our scheme to account for multi-layer video encoding. Finally, our scheme will be specialized for CDMA [20] and UWB [21] multimedia sensors.

References

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

[2] M. Rahimi, R. Baer, O. Iroezi, J. Garcia, J. Warrior, D. Estrin, and M. Srivastava, "Cyclops: In Situ Image Sensing and Interpretation in Wireless Sensor Networks," in *Proc. of the ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, San Diego, CA, November 2005.

[3] I. Akyildiz, T. Melodia, and K. Chowdhury, "Wireless multimedia sensor networks: Applications and testbeds," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1588–1605, Oct. 2008.

[4] Akan, O.B., "Performance of Transport Protocols for Multimedia Communications in Wireless Sensor Networks," *Communications Letters, IEEE*, vol. 11, no. 10, pp. 826–828, October 2007.

[5] T. Melodia, M. C. Vuran, and D. Pompili, "The State of the Art in Cross-layer Design for Wireless Sensor Networks," in *Proceedings of EuroNGI Workshops on Wireless and Mobility. Springer Lecture Notes in Computer Science 3883*, Como, Italy, July 2005.

Fig. 9. Snapshots of *Foreman* with 10 dB PSNR difference (DMRC at the top and TFRC at the bottom).

[6] X. Lin, N. Shroff, and R. Srikant, “A tutorial on cross-layer optimization in wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1452–1463, Aug. 2006.

[7] H.-P. Shiang and M. van der Schaar, “Multi-user video streaming over multi-hop wireless networks: a distributed, cross-layer approach based on priority queuing,” *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 4, pp. 770–785, May 2007.

[8] S. Ci, H. Wang, , and D. Wu, “A Theoretical Framework for Quality-Aware Cross-Layer Optimized Wireless Multimedia Communications,” *Advances in Multimedia*, vol. 2008, no. , 2008.

[9] P. V. Pahalawatta and A. K. Katsaggelos, “Review of content-aware resource allocation schemes for video streaming over wireless networks,” *Wireless Communications and Mobile Computing*, vol. 7, no. 2, pp. 131–142, Jan 2007.

[10] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, and B. Girod, “Cross-Layer Design of Ad Hoc Networks for Real-Time Video Streaming,” *IEEE Wireless Communications*, vol. 12, no. 4, pp. 59–65, August 2005.

[11] Y. Andreopoulos, N. Mastronarde, and M. van der Schaar, “Cross-Layer Optimized Video Streaming Over Wireless Multihop Mesh Networks,” *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 11, pp. 2104–2115, Nov. 2006.

[12] A. Klein and J. Klaue, “Performance Study of a Video Application over Multi-Hop Wireless Networks with Statistic-Based Routing,” in *Networking 2009*, Aachen, Germany, may 2009.

[13] D. Wu, H. Luo, S. Ci, and A. Katsaggelos, “Application-Centric Routing for Video Streaming over Multi-hop Wireless Networks,” in *submitted for publication*, 2009.

[14] M. Handley, S. Floyd, J. Padhye, and J. Widmer, “TCP Friendly Rate Control (TFRC): Protocol Specification,” IETF RFC 3448.

[15] M. Allman, V. Paxson, and W. Stevens, “TCP Congestion Control,” IETF RFC 2581.

[16] W.-T. Tan and A. Zakhor, “Real-time Internet video using error resilient scalable compression and TCP-friendly transport protocol,” *Multimedia, IEEE Transactions on*, vol. 1, no. 2, pp. 172–186, Jun 1999.

[17] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, “Complex behavior at scale: An experimental study of low-power wireless sensor networks,” UCLA/CSD-TR 02-0013, Tech. Rep., February 2002.

[18] O. B. Akan and I. F. Akyildiz, “ARC: the analytical rate control scheme for real-time traffic in wireless networks,” *Networking, IEEE/ACM Transactions on*, vol. 12, no. 4, pp. 634–644, Aug. 2004.

[19] K. Stuhlmuller, N. Farber, M.Link, and B. Girod, “Analysis of video transmission over lossy channels,” *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 6, pp. 1012–1032, Jun 2000.

[20] L. Kondi, S. Batalama, D. Pados, and A. Katsaggelos, “Joint source-channel coding for scalable video over DS-CDMA multipath fading channels,” *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 1, pp. 994–997 vol.1, 2001.

[21] T. Melodia and I. F. Akyildiz, “Cross-layer Quality of Service Support for UWB Wireless Multimedia Sensor Networks,” in *Proc. of IEEE Conference on Computer Communications (INFOCOM), Mini-Conference*, Phoenix, AZ, Apr. 2008.

[22] J. Massey, “Shift-register synthesis and BCH decoding,” *Information Theory, IEEE Transactions on*, vol. 15, no. 1, pp. 122–127, Jan 1969.

[23] J. Hagenauer, “Rate-compatible punctured convolutional codes (RCPC codes) and their applications,” *Communications, IEEE Transactions on*, vol. 36, no. 4, pp. 389–400, Apr 1988.

[24] F. Bellard, http://www.ffmpeg.org.

[25] J. Klaue, B. Rathke, and A. Wolisz, “EvalVid - A Framework for Video Transmission and Quality Evaluation,” in *In Proc. of the Intl. Conf. on Modelling Techniques and Tools for Computer Performance Evaluation*, Urbana, Illinois, September 2003, pp. 255–272.

[26] A. Lie and J. Klaue, “Evalvid-RA: trace driven simulation of rate adaptive MPEG-4 VBR video,” *Multimedia Systems*, vol. 14, no. 1, pp. 33–50, June 2008.

[27] “Generic Coding of Moving Pictures and Associated Audio Information: Video,” ISO/IEC 13818-2, ITU-T Rec. H.262, 1995.

[28] “JPEG2000 Requirements and Profiles,” ISO/IEC JTC1/SC29/WG1 N1271, March 1999.

[29] P. Lancaster and K. Šalkauskas, *Curve and Surface Fitting: An Introduction*. London: Academic Press, 1986.

[30] Arizona State University Video Traces Research Group, http://trace.eas.asu.edu/yuv/index.html.

[31] S. Lin and D. J. Costello, Jr, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1983.

[32] L. Carvalho, J. Angeja, and A. Navarro, “A new packet loss model of the IEEE 802.11g wireless network for multimedia communications,” *IEEE Transactions on Consumer Electronics*, vol. 51, no. 3, pp. 809–814, Aug. 2005.

[33] V.C. Gungor and O.B. Akan and I.F. Akyildiz, “A Real-Time and Reliable Transport Protocol for Wireless Sensor and Actor Networks,” *IEEE Transactions on Networking*, vol. 16, no. 2, pp. 359–371, April 2008.

[34] N. Shankar and M. van der Schaar, “Performance Analysis of Video Transmission Over IEEE 802.11a/e WLANs,” *IEEE Transactions on Vehicular Technology*, vol. 56, no. 4, pp. 2346–2362, July 2007.

[35] C. E. Perkins, E. M. Belding-Royer, and S. Das, “Ad Hoc On Demand Distance Vector (AODV) Routing,” IETF RFC 3561.

[36] C.-H. Ke, C.-K. Shieh, W.-S. Hwang, and A. Ziviani, “An Evaluation Framework for More Realistic Simulations of MPEG Video Transmission,” *Journal of Information Science and Engineering*, vol. 24, no. 2, pp. 425–440, March 2008.

[37] 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*, September 1984.

[38] “Crossbow,” http://www.xbow.com.