Computer Communications xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A distortion-minimizing rate controller for wireless multimedia sensor networks ☆,☆☆

Scott Pudlewski, Tommaso Melodia *

Wireless Networks and Embedded Systems Laboratory, Department of Electrical Engineering, State University of New York (SUNY) at Buffalo, Buffalo, NY, USA

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Multimedia sensor networks
Cross-layer design
Rate control

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.

A new cross-layer rate control scheme for WMSNs is introduced in this paper with a twofold objective: (i) maximize the video quality of each individual video stream; (ii) maintain fairness in the domain of video quality between different video streams. The rate control scheme is based on analytical and empirical models of video distortion 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 with respect to state-of-the art rate control algorithms, without sacrificing on fairness.

1. Introduction

The availability 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) [2], i.e., distributed systems of wirelessly networked devices deployed to retrieve video and audio streams, still images, and scalar sensor data from the environment. 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 [4,5]. First, embedded sensors are constrained in terms of battery, memory, processing capability, and achievable overall rate [2], while delivery of multimedia flows may be a resource-intensive task. Second, 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 networking 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 [6–10].

In this paper, we consider a multi-hop wireless network of video sensors deployed for surveillance applications and focus on reliable and real-time transport of video traffic. The objective is to design algorithms to *efficiently* and *fairly* share the common network infrastructure among the video streams 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) algorithm, a new decentralized cross-layer control algorithm that jointly regulates the end-to-end data rate, the video

☆ This work was supported by the US Air Force Research Laboratory under Grant FA8750-08-1-0063.

☆☆ A preliminary shorter version of this paper [1] appeared in the Proceedings of the IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS'2009).

* Corresponding author.
E-mail addresses: smp25@buffalo.edu (S. Pudlewski), tmelodia@eng.buffalo.edu (T. Melodia).

0140-3664/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2010.02.016

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 as in traditional rate control algorithms). Once the end-to-end data rate has been determined, the sender calculates 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 video distortion at the receiver.

Video distortion in wireless networks is mostly caused by lossy source coding, transmission errors originated by channel fading, buffer overflows and playout deadline misses. Intuitively, if the loss happens at a relay node due to congestion, then the video encoder rate should 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 should then be increased as the wireless channel errors decrease. In DMRC, the signal 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 [8,9,11–14], 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) [15], 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 regulating the end-to-end data rate based on the video compression rate and by jointly optimizing the video coding and channel coding rates, DMRC uses network resources more efficiently than state-of-the-art rate control protocols like the TCP-Friendly Rate Control (TFRC) [15], thus resulting in higher video quality at the receiver.

The remainder of this paper is structured as follows. In Section 2, we discuss previous work on related topics. In Section 3, we introduce the considered system model. In Section 4, we describe our solution to the determination of the video encoder rate and the channel encoder rate, and in Section 5 we extend the analysis to the case of predictive video encoders. In Section 6, we describe the DMRC rate controller in detail. In Section 8 we discuss performance evaluation results, while in Section 9 we draw the main conclusions.

2. Related work

The most common form of rate control is the well-known transmission control protocol (TCP) [16]. Because of the additive increase/multiplicative-decrease algorithm used in TCP, the rate transitions that it determines are not smooth enough for high quality video transfer [17]. In addition, TCP assumes that the main cause of packet loss is network 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 [18] it was experimentally shown 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 [15], 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 [16], and the Analytical Rate Control (ARC) [19], 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 [bit/s] represents the transmission rate, S [bit] is the packet size, R [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 [20], 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, [20] 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 [21], where the authors 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 focus on a solution that is independent of the underlying MAC protocol and of the specific video source encoding scheme, and consider a multi-hop network. 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 [21]. 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 [22]. However, transport-layer issues and video quality related metrics were not addressed.

Please cite this article in press as: S. Pudlewski, T. Melodia, A distortion-minimizing rate controller for wireless multimedia sensor networks, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.02.016

3. System model

3.1. 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 $\underline{R}_C = \left[R_C^1, R_C^2, \ldots, R_C^{MAX_\rho}\right]$, 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 $\mathscr{C}$. Different families of codes, such as BCH codes [23] or rate-compatible punctured codes (RCPC) [24], 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 [24]. 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 a given channel and network condition. Clearly, as these codes are punctured to reduce the redundancy, the effectiveness of the codes decreases along with the ability to correct bit errors. Therefore we are trading bit error rate for transmission rate.

3.2. Video distortion model

We evaluate video quality in terms of 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 video encoding at the highest possible rate 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 $m \times n$ 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 [bits in/bits out] is the *channel encoder rate* defined as the rate of the channel 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_{enc}^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.

4. 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 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 [25], Evalvid [26] and Evalvid-RA [27].

The term D_{enc}^m in (5) refers to the amount of distortion introduced by the video encoder. The video encoding scheme used in this paper is simple Motion Picture Experts Group (MPEG) [28]. 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) [29] 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 [28].

Please cite this article in press as: S. Pudlewski, T. Melodia, A distortion-minimizing rate controller for wireless multimedia sensor networks, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.02.016

Similar to [20], a three-parameter empirical model as in (8) can be used to represent the amount of distortion introduces by the MPEG encoder:

$$D_{enc}^{m} = D_0 + \frac{\theta}{R_{source}^{m} - R_0}, \tag{8}$$

where R_{source}^{m} is the video encoder rate and the parameters D_0, θ and R_0 can be estimated from empirical rate-distortion curves via a least square curve fitting [30]. The curves for three different videos, namely Akiyo, Foreman and Container from [31] are shown in Fig. 1. Although our methodology can easily be extended to higher order models, 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 1 min to the next, or from 1 h to the next. For such an application, the curve fitting could perhaps only be done once every 10 min. 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 easily 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 MHz with dynamic voltage scaling [3].

The term D_{loss}^{m} 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

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

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 [30] 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_{loss}^{m}, represents the packets lost due to channel errors. The second, PER_{delay}^{m}, represents losses due to packets arriving at the receiver after the playout delay deadline. We will examine each of these individually.

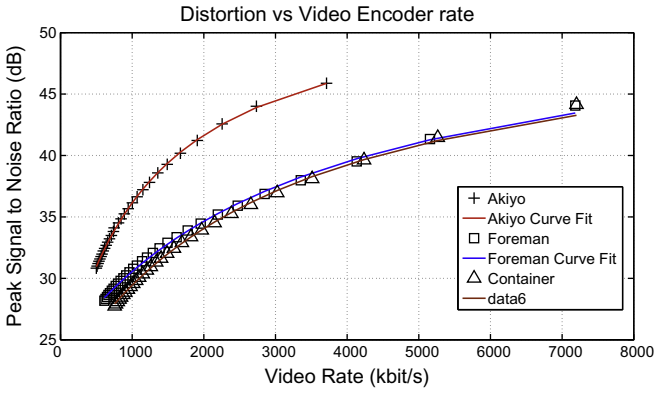

Fig. 1. Regression curves used to determine distortion due to JPEG encoding.

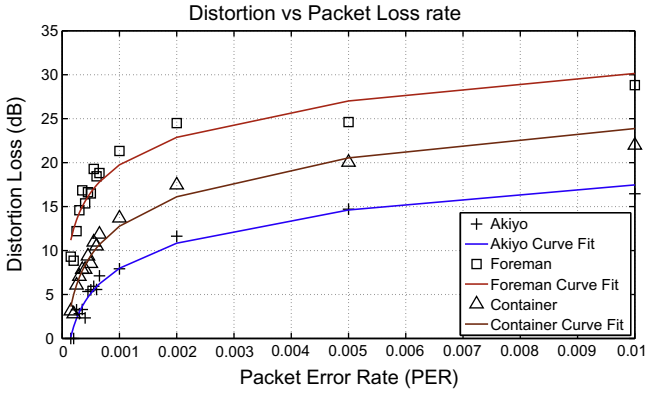

Fig. 2. Regression curves used to determine distortion due to packet error rate (PER).

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

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

where P_n is the probability of a successful transmission at the nth hop. PER_{loss}^{m} can be expressed as

$$PER_{loss}^{m} = 1 - P_{success}, \tag{11}$$

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 to transmit that packet. The effect of the channel coding can be taken into account by assuming that the code alters the apparent SNR at the receiver as in (12),

$$SNR = d_{free} \cdot R_{C,RCPC}^{m} \cdot \frac{E_b}{N_0}, \tag{12}$$

where d_{free} is the free distance of the code and $R_{C,RCPC}^{m}$ is the rate of the code ($R_{C,RCPC}^{m} < 1$) [32]. 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 7.

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 $PER_{delay}^{m} \sim \mathcal{N}(\mu_\epsilon, \sigma_\epsilon^2)$ where ϵ is the forward trip time, μ_ϵ is the mean of ϵ and σ_ϵ^2 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

$$PER_{delay}^{m} = Q\left(\frac{T_{playout} - \mu_\epsilon}{\sigma_\epsilon^2}\right) \tag{13}$$

which represents the standard error function, defined as

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

Even though there exist more accurate delay models, e.g., [33–35], these would be computationally infeasible for implementation on a WMSN node.

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

$$PSNR_{loss} = PSNR_{optimal} - PSNR_{achieved}, \tag{15}$$

against the product of R_V^m and R_C^m from (7). In (15), $PSNR_{optimal}$ is the $PSNR$ obtained using perfect channels, no buffer overflows and the

Please cite this article in press as: S. Pudlewski, T. Melodia, A distortion-minimizing rate controller for wireless multimedia sensor networks, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.02.016

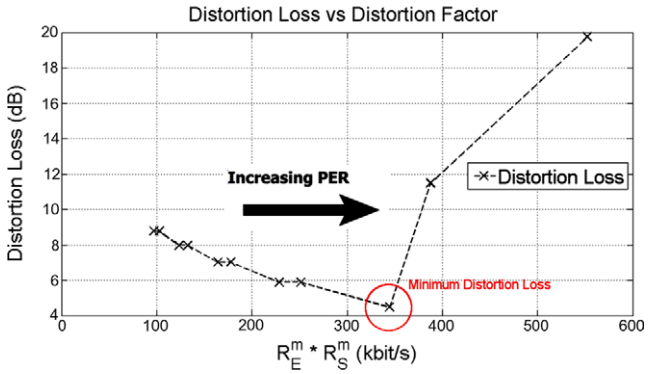

Fig. 3. Distortion loss vs. distortion factor.

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_V^m and R_C^m, we examine their product 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 defined in [24] and six of the videos defined in [31] to remove the dependency on these specific factors from the analysis.

Since the overall rate in (7) represents the ratio of the video and channel encoder rates, the product $R_V \cdot R_C = R \cdot R_C^2$ essentially gives us the ratio of the two rates. For a fixed end-to-end overall rate, 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 is derived for a fixed overall end-to-end rate of 6.4 Mbit/s. The figure clearly shows that there is a minimum distortion region along the curve. 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 identifies the optimal *PER* for given 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) $[\lambda_0, \lambda_1, \ldots, \lambda_{max}]$. The elements $\lambda_n \in \Lambda$ represent the possible values of $R_V^m \cdot R_C^m$. Since both R_V^m and R_C^m 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 $(D_{\lambda_n}^m)$ 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.

$$\begin{aligned} \mathbf{P1}: &Given: R^m,\ SNR,\ PER^m_{delay} \\ &Find: R_V^m,\ R_C^m \\ &Minimize: D^m_{enc} + D^m_{loss} \\ &Subject\ to: R_V^m \in \mathscr{R}_V,\ R_C^m \in \mathscr{R}_C \end{aligned}$$

Algorithm 1.

Distortion Minimization

1: $\Delta = -1,\ n = 0$
2: **while** $\Delta < 0$ do
3: **for** $\lambda_n \in \Lambda$ do
4: $\Delta = D^m_{\lambda_n} - D^m_{\lambda_{n+1}}$
5: $n = n + 1$
6: **end for**
7: **end while**
8: Return solution as $[R_V^m R_C^m]$ corresponding to λ_n

5. Effect of multiple frame types

While the above solution is valid in general, it does not take into account the frame structure introduced by predictive encoders such as MPEG [28] or H.264 [36] video compression. The effect of losing packets associated to different frames on the quality of the received video varies considerably.

A typical predictively encoded video is composed of three distinctly different types of frames; Intracoded (I) frames, predictive (P) frames and bidirectionally predicted (B) frames. Even though there are different possible ways to actually encode the video frames, a video with a group-of-pictures (GOP) of length 12 can be encoded as

$$I\,B\,B\,P\,B\,B\,P\,B\,B\,P\,B\,B\,I, \tag{16}$$

where the trailing *I* represents the beginning of the next GOP. The advantage to using the more complicated video encoder is that these more complex methods produce less data at the source encoder while maintaining the same or better video quality. However, the amount of dependency between the encoded frames increases as the complexity increases. For example, an *I* frame is, as its name states, entirely intra-coded and therefore it does not depend on any other frame to be properly decoded. The *P* frames are *predicted* from the previous *I* or *P* frame, and therefore have a direct dependence on the previous *I* or *P*. This means that a *P* frame can only be correctly decoded if the frame it depends on has also been correctly decoded. *B* frames are predicted both from the previous *I* or *P* frame and the subsequent *I* or *P* frame. Again, correctly decoding a *B* frame depends on first correctly decoding two other frames.

Another way to look at this structure is in terms of the effect the loss of each frame has on surrounding frames within a single GOP. For example, if a *B* frame is lost, only the data from that single frame is lost, and the resulting video will have lost only a single frame of information. In a 30 frame per second video, this may result in a small perceptual difference at the receiver in terms of video quality. If a *P* frame is lost, however, any frames that depend on that *P* frame will also be un-decodable at the receiver. In above case of a GOP of 12, this could result in anywhere from 4 (for the last *P* frame) to 10 (for the first *P* frame) additional frames being lost in the video. The effect of an *I* frame is even worse, where the entire GOP will not be decodable, and the video at the receiver will be frozen until the next *I* frame is correctly received. Within the MPEG standard, there are three types of encoding formats that can be considered.

5.1. Only intra-encoded frames

This is the case where only *I* frames are used in the video encoding. Even though the limited amount of compression available from this method means that it is rarely used in traditional application, the lower computational complexity of this type of encoder may sometimes make it desirable for low-cost embedded

Please cite this article in press as: S. Pudlewski, T. Melodia, A distortion-minimizing rate controller for wireless multimedia sensor networks, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.02.016

video sensors. In this case, there is no interdependence between different frames, so any frame lost will only affect the received video for that given frame. Therefore, the results from Section 4 can be implemented directly.

5.2. I and P frames used

In this type of video encoding, each GOP consists of an I frame followed by a series of P frames, where each P frame is dependent on the previous frame for correct decoding. Every lost frame will affect the rest of the frames in the GOP. This means that a packet lost will affect the rest of the GOP as if every following packet were lost.

In order to determine how this will affect the reconstructed video quality, we must first look at frame error rate (FER_{IP}) as opposed to PER. The frame error rate can be determined from the equation

$$FER_{IP} = 1 - (1 - PER)^L, \tag{17}$$

where L represents the mean number of packets in a frame. Since the effect of a loss of I or P frame is the same in terms of the effect on the rest of the GOP, we do not need to distinguish between them.

The ideal FER(FER_{ideal}) without taking this error propagation into account is determined from the PER determined with Algorithm 1. In average, the effect of the error propagation in this model can be expressed as

$$FER_{IP,ideal} = \frac{1}{2} \cdot \Psi \cdot FER_{ideal}, \tag{18}$$

where Ψ the length of the GOP, and the ideal PER can be determined from this by solving (17) for the PER and using $FER_{IP,ideal}$. This result can be intuitively understood by noticing that, on average, each frame loss will affect *half* of the GOP.

5.3. Addition of B frames

The addition of B frames complicates the analysis. In this case, each of the three types of packets will impact the total FER differently. The total FER_{IPB} can be expressed by

$$FER_{IPB} = FER_I + FER_P + FER_B, \tag{19}$$

where FER_I, FER_P and FER_B are the components of the frame error rate which occur form the loss of I, P or B frames, respectively. In order for (19) to be valid, FER_I, FER_P and FER_B must all be disjoint probabilities. Since this is not normally the case due to the interdependencies described above, those interdependencies must be removed from the calculations for the component frame error rates. This is done by excluding from each component frame errors that have already been accounted for in another component. For example, if a B frame is not received correctly and the I frame at the beginning of the GOP is also not received correctly, the frame error rate of the B frames must be decreased accordingly.

The individual frame error rates are calculated as

$$FER_i = (1 - (1 - PER)^{L_i}) \cdot W_i \cdot I_i \cdot P_i, \tag{20}$$

where $i \in \{I, P, B\}, W_i$ represents the *weight*, or impact of an error of type i on other frames within a GOP and I_i is the amount of *interdependence*, or the probability that a loss of type i was already accounted for by another type of error. P_i represents the probability that a given packet is a packet from a frame of type i.

W_i and I_i need to be calculated individually depending on the type of encoding used. As an example, we will show here how they could be calculated for the encoding scheme given in (16).

- $W_I \cdot W_I = \Psi$, or the weight of the loss of an I packet, will always be the size of the GOP. This is because the loss of an I frame always results in the loss of the entire GOP.
- I_I. Since I frames are decoded independently, $I_I = 1$.
- W_B. Since no other packets depend on B packets for decoding, $W_B = 1$.
- I_B. In order to correctly decode a B frame, the preceding I or B frame and the subsequent I or B frame must have been correctly decoded. Therefore the loss of either of these frames will have to be considered in I_B. For the given encoding pattern, the simplest way to do this is to consider each of the pairs of two B frames individually. For the first group, the loss of either frame will already be included if either the first I frame is lost or the first P frame is lost. This can be expressed as $I_{B,1} = 2 \cdot (FER_I + FER_P)$. The other groups can be calculated in the same way, resulting in

$$I_{B,1} = \frac{1}{4} \cdot (FER_I + FER_P)$$
$$I_{B,2} = \frac{1}{4} \cdot (FER_I + 2 \cdot FER_P)$$
$$I_{B,3} = \frac{1}{4} \cdot (FER_I + 3 \cdot FER_P)$$
$$I_{B,4} = \frac{1}{4} \cdot (2 \cdot FER_I + 3 \cdot FER_P).$$

  The overall value of I_B can be calculated to be

$$I_B = \sum_{i=1}^{4} I_{B,i} = \frac{1}{4} \cdot (5 \cdot FER_I + 9 \cdot FER_P). \tag{21}$$

- W_P. The weight of a P frame depends on where in the GOP it is. In this example, the loss of a P frame will affect 11, 8 or 5 frames, which includes all of the frames after the P frame, along with the two B frames before the P frame. Therefore, on average W_P will be 8 frames.
- I_P. A P frame cannot be correctly decoded if either the I frame was not correctly received, or if a previous P frame was not received. Since the number of preceding P frames depends on the location of the P frame, like the B frame we need to look at each of the frames individually and find the mean. This results in

$$I_{P,1} = \frac{1}{3} \cdot (FER_I)$$
$$I_{P,2} = \frac{1}{3} \cdot (FER_I + FER_P)$$
$$I_{P,3} = \frac{1}{3} \cdot (FER_I + 2 \cdot FER_P),$$

  and

$$I_B = \sum_{i=1}^{3} I_{P,i} = (FER_I + FER_P). \tag{22}$$

5.4. Unequal error protection

In Section 4, it was shown that there is an optimal tradeoff between the fraction of rate assigned to the video encoder and the rate given to the channel encoder. Since the video encoding scheme itself can cause additional frames to be lost at the receiver, more of the rate will have to be allocated to the channel encoder, which reduces the amount of the rate which can be allocated to the video encoder, which will in turn *reduce the received video quality for the same overall data rate*. One way to help offset this effect is to use Unequal Error Protection (UEP). Quite simply, by allocating more of the rate to more important frames (I frames) and less of the rate to less important frames (B frames), the overall quality of the received video can be increased without increasing the overall data rate.

Please cite this article in press as: S. Pudlewski, T. Melodia, A distortion-minimizing rate controller for wireless multimedia sensor networks, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.02.016

One simple way to accomplish this is to increase the rate of the *B* frames to the highest available rate by reducing the channel encoding to the lowest amount. Essentially, we would be considering the *B* frames as throwaway frames, which help if they are received, but are not as essential to the final decoding process. Then, within a GOP, we could decrease the rate of the *I* frames by increasing the redundancy. Although this method is clearly not guaranteed to be optimal, it is sufficient to show the benefit of using UEP. Also, this method will alleviate some of the cost of using the more advanced video encoding system. As can be seen in Fig. 4, even this simple unequal error protection method has the ability to increase the overall video quality, using a very small amount of calculation and no additional transmissions.

5.5. The effect of multiple packets on the optimal rate selection

The differing effects of *I*, *P* and *B* packets will directly influence the optimal tradeoff between the video encoder and the channel encoder. More specifically, the analysis of the video distortion as a function of packet loss will clearly be different, since the effects of a lost packet depend directly on the type of video encoder used. In Section 4, it was assumed that any packet loss only affected a single frame. This means that any packet lost from a frame results in that frame being missed at the receiver. Now we have shown that the loss of a single packet could result in multiple frame loss, which could include an entire GOP. In order to use this new information, the target PER from Algorithm 1 must be modified so that the overall *FER* of the received video is the same as if predictive video encoding were not used.

For example, the optimal *PER* is determined to be 10^{-3} according to the empirical Eq. (5). The *FER* in this case can easily be found with (17). If the *FER* is then modified to deal with the predictive video encoding, and a new $PER_{predictive}$ is found based on FER_{IPB}, the source node can again determine the amount of channel coding needed to achieve the *best performance* of the received video quality.

6. Distortion-minimizing rate control (DMRC)

In this section, we present a rate control scheme that regulates the overall end-to-end 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.

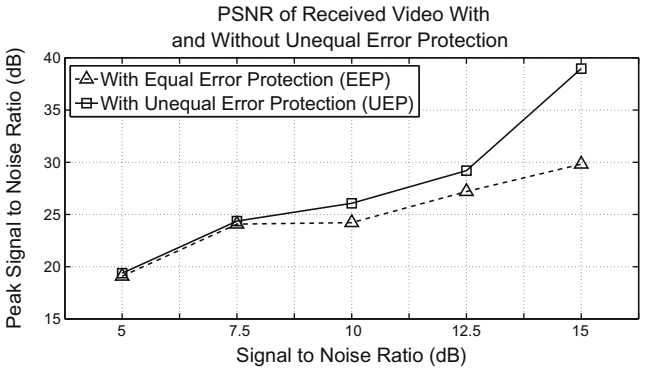

Fig. 4. Video quality with and without unequal error protection.

6.1. 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 that the network cannot 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 that would help solve or prevent a crime. If there is no fairness between video streams, this stream may have such poor quality that it cannot even be decoded. In most typical surveillance WMSN systems, there will only be a single camera to cover a given area, and if that camera's video is not able to be received, *that area will not be under surveillance*.

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 $R^m_{V,MAX}$ 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, $R^m_{V,MAX}$ 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* [31]. 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, 174×144 (QCIF) video sources, *Foreman* and *Akiyo*. Even though the raw data for both videos is 44,500 kbyte, the best quality for *Akiyo* can be obtained with a 4641 kbyte file with 45.88 dB PSNR, while *Foreman* needs 8980 kbyte to achieve 44.07 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 estimate if competing traffic is creating congestion. If the traffic is increasing, the *RTT* will increase and therefore the sender can decrease its own video encoder rate 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 follows this law:

$$R_i = \begin{cases} R_{i-1} - \frac{1}{\alpha \cdot \Delta \widetilde{RTT}} & \text{if } \widetilde{RTT}_t > \widetilde{RTT}_{t-1} \\ R_{i-1} + \frac{1}{\beta \cdot \Delta \widetilde{RTT}} & \text{if } \widetilde{RTT}_t \leqslant \widetilde{RTT}_{t-1}, \end{cases} \tag{23}$$

where

$$\alpha = \alpha_0 \cdot DR \tag{24}$$

$$\beta = \beta_0 \cdot DR. \tag{25}$$

In (23), $\widetilde{RTT}_t$ represents the weighted average of the previous *N RTT* measurements, and is defined as

Please cite this article in press as: S. Pudlewski, T. Melodia, A distortion-minimizing rate controller for wireless multimedia sensor networks, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.02.016

$$\widetilde{RTT}_t = \frac{\sum_{i=1}^{N} A_i \cdot RTT_{t-i}}{N \cdot \sum_{i=1}^{N} A_i}. \quad (26)$$

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 video coding rate and the channel coding rate and is defined as:

$$DR = \begin{cases} \Delta\rho + \sigma & \text{if } \widetilde{RTT}_t > \widetilde{RTT}_{t-1} \\ (MAX_\rho - \Delta\rho) + MAX_\sigma - \sigma & \text{if } \widetilde{RTT}_t < \widetilde{RTT}_{t-1}, \end{cases} \quad (27)$$

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 3.1) 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 that 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 (27), 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 (23). 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 (23) 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 starts 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 (27), which will therefore magnify the change in overall rate determined by (23). 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.

This can be seen in Fig. 5, which shows the data rate vs. time of two competing video transmissions. At 30 s, when the larger video begins to transmit, the smaller video immediately backs off. The speed of the data rate decrease is due to the very high data rate (and therefore the very high video quality) being used by the first video. Using this method, the videos are allowed to increase their transmission rate as needed to take advantage of the open channel, but still allow a new video the data rate it needs to transmit, even if that rate is more than what was needed by the original video.

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 as the *RTT* values indicate that another node is attempting to transmit a video.

7. 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

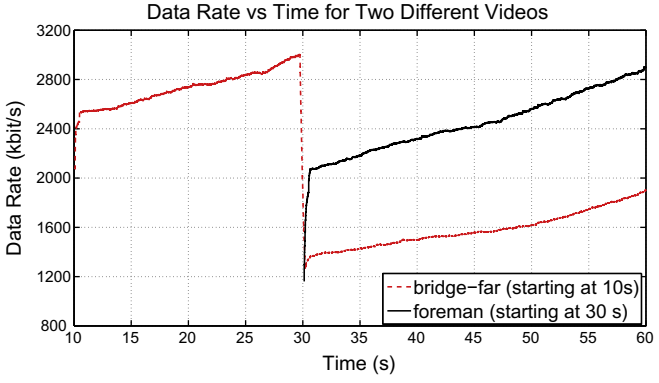

Fig. 5. The data rate of two different quality videos starting at different times.

Please cite this article in press as: S. Pudlewski, T. Melodia, A distortion-minimizing rate controller for wireless multimedia sensor networks, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.02.016

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.

8. Performance evaluation

The DMRC algorithm was evaluated using the ns-2 version 2.33 network simulator along with the Evalvid [26] tool-set. All simulations are done using 49 nodes in a 7×7 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 [37]. The average path length of three hops, however, varied greatly between simulations due to both the random selection of the video source and the number of interfering videos in a given simulation. A simple channel model was used based on a threshold method (i.e., packet is dropped if SINR is below a threshold, received if the SINR is above a threshold). Nearly all packet losses were due to interference between nodes or buffer overflow.

The frames were compressed at multiple compression rates using ffmpeg [25]. Because the ffmpeg encoder and decoder are compliant with the current MPEG standards, all videos generated here will conform with current video encoding standards. 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 [31]. These videos were chosen because they were each approximately of the same length, while their compressed sizes were different [38].

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. Fig. 6 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 4. 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

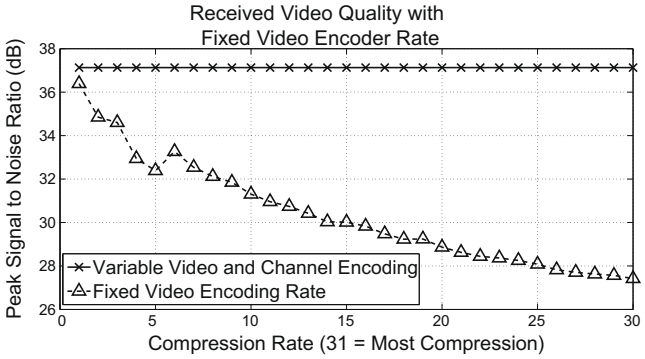

Fig. 6. Video quality of fixed video encoder vs. variable video and variable channel encoders.

are reported in Fig. 11. Again, the video quality is always higher when jointly selecting both video and channel encoder rates as described in Section 4 than with any fixed channel encoder rate. Then, the DMRC rate controller was compared to TFRC as described in [15] using two different sets of simulations. TFRC was chosen as a comparison because it is the most well-known equation based rate control protocol currently in use for multimedia networking.

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. 7. In this figure, the received video quality for all videos was averaged 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. 8, which traces the

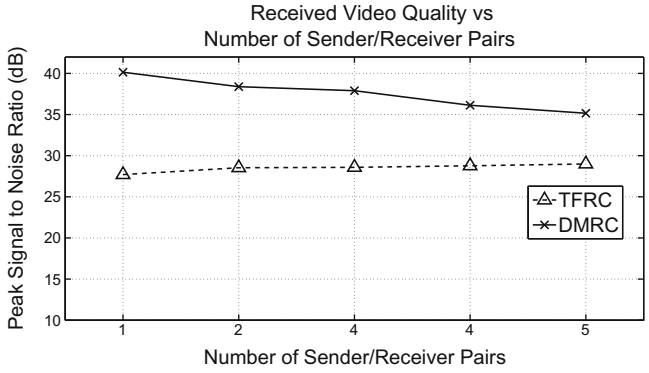

Fig. 7. Received video quality vs. number of sender/receiver pairs.

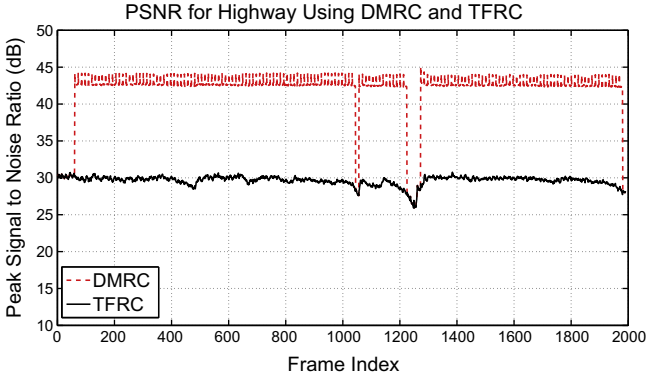

Fig. 8. Received video quality vs. time or *Highway* transmitted with DMRC and TFRC.

Please cite this article in press as: S. Pudlewski, T. Melodia, A distortion-minimizing rate controller for wireless multimedia sensor networks, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.02.016

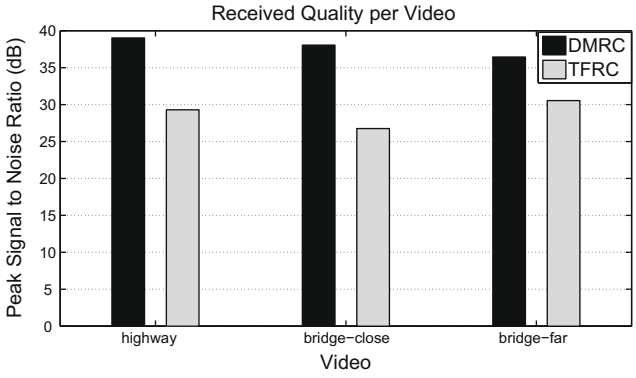

Fig. 9. Received video quality of different video for DMRC and TFRC.

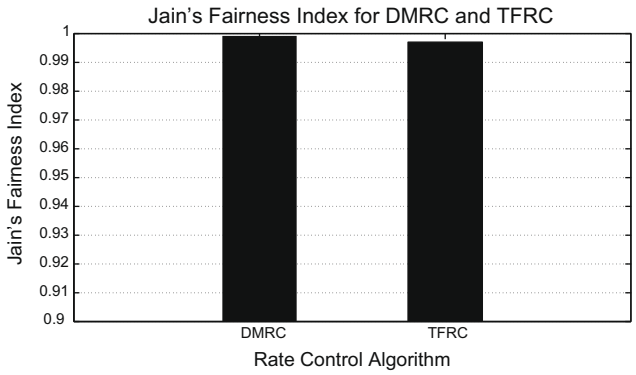

Fig. 10. Jain's fairness index for DMRC and TFRC.

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. 9. 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. 7. As the number of sender/receiver pairs increases, the video quality for the DMRC

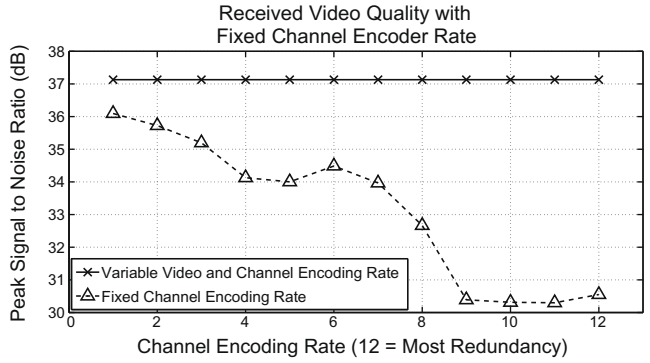

Fig. 11. Video quality of fixed channel encoder vs. variable video and variable channel encoders.

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.

Fig. 12 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 Index [39], defined as

$$f(x_1, x_2, \ldots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \cdot \sum_{i=1}^{n} x_i^2}, \tag{28}$$

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.

9. 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

Fig. 12. Snapshots of *Foreman* with 10 dB PSNR difference (DMRC at the top and TFRC at the bottom).

Please cite this article in press as: S. Pudlewski, T. Melodia, A distortion-minimizing rate controller for wireless multimedia sensor networks, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.02.016

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 a USRP2 software defined radio platform 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 [21] and UWB [22] multimedia sensors.

References

[1] S. Pudlewski, T. Melodia, DMRC: distortion-minimizing rate control for wireless multimedia sensor networks, in: Proceedings of the IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS), Macau S.A.R., PR China, October 2009.
[2] I.F. Akyildiz, T. Melodia, K.R. Chowdhury, A survey on wireless multimedia sensor networks, Computer Networks (Elsevier) 51 (4) (2007) 921–960.
[3] I. Akyildiz, T. Melodia, K. Chowdhury, Wireless multimedia sensor networks: applications and testbeds, Proceedings of the IEEE 96 (10) (2008) 1588–1605.
[4] O. Akan, Performance of transport protocols for multimedia communications in wireless sensor networks, Communications Letters IEEE 11 (10) (2007) 826–828.
[5] E. Gurses, O.B. Akan, Multimedia communication in wireless sensor networks, Annals of Telecommunications 60 (7–8) (2005) 799–827.
[6] T. Melodia, M.C. Vuran, D. Pompili, The state of the art in cross-layer design for wireless sensor networks, in: Proceedings of EuroNGI Workshops on Wireless and Mobility, Lecture Notes in Computer Science vol. 3883, Como, Italy, Springer, Berlin, July 2005.
[7] X. Lin, N. Shroff, R. Srikant, A tutorial on cross-layer optimization in wireless networks, IEEE Journal on Selected Areas in Communications 24 (8) (2006) 1452–1463.
[8] H.-P. Shiang, M. van der Schaar, Multi-user video streaming over multi-hop wireless networks: a distributed, cross-layer approach based on priority queuing, IEEE Journal on Selected Areas in Communications 25 (4) (2007) 770–785.
[9] S. Ci, H. Wang, D. Wu, A theoretical framework for quality-aware cross-layer optimized wireless multimedia communications, Advances in Multimedia 2008 (2008).
[10] P.V. Pahalawatta, A.K. Katsaggelos, Review of content-aware resource allocation schemes for video streaming over wireless networks, Wireless Communications and Mobile Computing 7 (2) (2007) 131–142.
[11] E. Setton, T. Yoo, X. Zhu, A. Goldsmith, B. Girod, Cross-layer design of ad hoc networks for real-time video streaming, IEEE Wireless Communications 12 (4) (2005) 59–65.
[12] Y. Andreopoulos, N. Mastronarde, M. van der Schaar, Cross-layer optimized video streaming over wireless multihop mesh networks, IEEE Journal on Selected Areas in Communications 24 (11) (2006) 2104–2115.
[13] A. Klein, J. Klaue, Performance study of a video application over multi-hop wireless networks with statistic-based routing, in: IFIP Networking 2009, Aachen, Germany, May 2009.
[14] D. Wu, H. Luo, S. Ci, A. Katsaggelos, Application-centric routing for video streaming over multi-hop wireless networks, in: Proceedings of the IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON), Rome, Italy, June 2009.
[15] M. Handley, S. Floyd, J. Padhye, J. Widmer, TCP Friendly Rate Control (TFRC): Protocol Specification, IETF RFC 3448, January 2003.
[16] M. Allman, V. Paxson, W. Stevens, TCP Congestion Control, IETF RFC 2581, April 1999.
[17] W.-T. Tan, A. Zakhor, Real-time internet video using error resilient scalable compression and tcp-friendly transport protocol, IEEE Transactions on Multimedia 1 (2) (1999) 172–186.
[18] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, S. Wicker, Complex behavior at scale: an experimental study of low-power wireless sensor networks, UCLA/CSD-TR 02-0013, Tech. Rep., February 2002.
[19] O.B. Akan, I.F. Akyildiz, ARC: the analytical rate control scheme for real-time traffic in wireless networks, Networking, IEEE/ACM Transactions 12 (4) (2004) 634–644.
[20] K. Stuhlmuller, N. Farber, M. Link, B. Girod, Analysis of video transmission over lossy channels, IEEE Journal on Selected Areas in Communications 18 (6) (2000) 1012–1032.
[21] L. Kondi, S. Batalama, D. Pados, A. Katsaggelos, Joint source-channel coding for scalable video over DS-CDMA multipath fading channels, in: Proceedings of the 2001 International Conference on Image Processing 2001, vol. 1, pp. 994–997, 2001.
[22] T. Melodia, I.F. Akyildiz, Cross-layer quality of service support for UWB wireless multimedia sensor networks, in: Proceedings of the IEEE Conference on Computer Communications (INFOCOM), Mini-Conference, Phoenix, AZ, April 2008.
[23] J. Massey, Shift-register synthesis and BCH decoding, Information Theory IEEE Transactions 15 (1) (1969) 122–127.
[24] J. Hagenauer, Rate-compatible punctured convolutional codes (RCPC codes) and their applications, Communications IEEE Transactions 36 (4) (1988) 389–400.
[25] F. Bellard. Available from: <http://www.ffmpeg.org>.
[26] J. Klaue, B. Rathke, A. Wolisz, EvalVid – a framework for video transmission and quality evaluation, in: Proceedings of the International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Urbana, Illinois, pp. 255–272, September 2003.
[27] A. Lie, J. Klaue, Evalvid-RA: trace driven simulation of rate adaptive MPEG-4 VBR video, Multimedia Systems 14 (1) (2008) 33–50.
[28] Generic Coding of Moving Pictures and Associated Audio Information: Video, ISO/IEC 13818-2, ITU-T Rec. H.262, 1995.
[29] JPEG2000 Requirements and Profiles, ISO/IEC JTC1/SC29/WG1 N1271, March 1999.
[30] P. Lancaster, K. Šalkauskas, Curve and Surface Fitting: An Introduction, Academic Press, London, 1986.
[31] Arizona State University Video Traces Research Group. Available from: <http://trace.eas.asu.edu/yuv/index.html>.
[32] S. Lin, D.J. Costello Jr., Error Control Coding: Fundamentals and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1983.
[33] L. Carvalho, J. Angeja, A. Navarro, A new packet loss model of the IEEE 802.11g wireless network for multimedia communications, IEEE Transactions on Consumer Electronics 51 (3) (2005) 809–814.
[34] V. Gungor, O. Akan, I. Akyildiz, A real-time and reliable transport protocol for wireless sensor and actor networks, IEEE/ACM Transactions on Networking 16 (2) (2008) 359–370.
[35] N. Shankar, M. van der Schaar, Performance analysis of video transmission over IEEE 802.11a/e WLANs, IEEE Transactions on Vehicular Technology 56 (4) (2007) 2346–2362.
[36] Advanced Video Coding for Generic Audiovisual Services, ITU-T Recommendation H.264, 2009.
[37] C.E. Perkins, E.M. Belding-Royer, S. Das, Ad hoc on demand distance vector (AODV) routing, IETF RFC 3561, July 2003.
[38] C.-H. Ke, C.-K. Shieh, W.-S. Hwang, A. Ziviani, An evaluation framework for more realistic simulations of MPEG video transmission, Journal of Information Science and Engineering 24 (2) (2008) 425–440.
[39] R. Jain, D.-M. Chiu, W. Hawe, A quantitative measure of fairness and discrimination for resource allocation in shared computer systems, DEC Research Report TR-301, September 1984.

Please cite this article in press as: S. Pudlewski, T. Melodia, A distortion-minimizing rate controller for wireless multimedia sensor networks, Comput. Commun. (2010), doi:10.1016/j.comcom.2010.02.016