Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2009-005 February 7, 2009

SoftCast: One Video to Serve All Wireless Receivers

Szymon Jakubczak, Hariharan Rahul, and Dina Katabi

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

SoftCast: One Video to Serve All Wireless Receivers

Szymon Jakubczak Hariharan Rahul Dina Katabi
szym@mit.edu rahul@csail.mit.edu dk@mit.edu

Abstract

The main challenge in wireless video multicast is to scalably serve multiple receivers who have different channel characteristics. Current wireless transmission schemes, however, cannot support smooth degradation. Specifically, each packet is transmitted at a particular bitrate and is decodable only by receivers that support the chosen bitrate. Broadcasting a video stream to all receivers requires transmitting at the lowest bitrate, and hence reduces everyone to the performance of the worst receiver in the multicast group.

This paper introduces SoftCast, an alternative design for wireless video multicast, in which a sender broadcasts a single stream and each receiver watches a video quality that matches its channel quality. SoftCast achieves this by making the magnitude of the transmitted signal proportional to the pixel value. Hence, channel noise directly translates to a small perturbation in pixel values, allowing graceful degradation with increasing noise. SoftCast introduces a novel power allocation scheme that allows the transmission of real-valued video signals in a compact and resilient manner. We implement SoftCast in the WARP radio platform. Our results show that SoftCast improves the average video quality across multicast receivers by 3-7dB over the current approach. Further, it stays competitive with the current approach even for regular unicast.

1 Introduction

Emerging applications of wireless video multicast include mobile TV, media sharing, and the broadcast of sporting events, lectures, promotional clips, and security videos in hotspots, universities, malls, airports, and train stations [14, 8, 40]. The main challenge in wireless video multicast, however, is to serve multiple receivers who have different channel characteristics. These receivers differ in the bit rate they can support, the throughput they obtain, and the packet loss they experience, but are all interested in the same video.

Ideally, one would like a source to broadcast a single video stream to all receivers, yet deliver to each receiver a video resolution that matches its channel. This is however not possible in the current network design, which exhibits a cliff effect in video performance. Specifically, the quality of a received video is constant above a certain SNR, and dramatically degrades when the receiver channel is below this SNR. This is due to decisions both at the PHY and the video codec. First, packets today are transmitted by the PHY at a particular 802.11 bitrate, and hence receivers that have channels that cannot support this bitrate are unable to receive these packets, and decode the video. Thus, broadcasting a video stream to all receivers requires transmitting at the lowest bit-rate. This restriction then forces the video codec to encode the stream at a low quality, and hence receivers that have channels that can support higher qualities do not experience any advantage. Thus, all receivers are essentially reduced to the performance of the worst receiver in a multicast group.

Today, there are two approaches to tackle this problem, but neither of them is satisfactory. The prevalent approach unicasts an independent stream to each receiver [7, 44, 43]. This approach does not scale beyond a handful of receivers. The alternative approach, known as multiple resolution coding (MRC), divides the video into a base layer and multiple enhancement layers. The base layer is sent at the lowest 802.11 bitrate to make it decodable by all receivers, and the enhancement layers are sent at higher bitrates decodable only by receivers with good channels [20, 10]. This approach is useful for wired multicast, where a receiver with a congested link can download only the base layer, and avoid packets from other layers. With wireless, all layers share the medium. The existence of the enhancement layers reduces the bandwidth available to the base layer and further worsens the performance of poor receivers. Additionally, the approach is complex because it requires deciding on how to divide the wireless capacity between the layers and which bitrate to use for each layer.

This paper introduces SoftCast, an alternative approach to wireless video multicast where a sender simply broadcasts its packets without specifying a bitrate or a video resolution, but each receiver sees a video resolution that matches its channel. Receivers with good channels extract a higher information rate from the transmitted signal and hence obtain a better video resolution. Receivers with worse channel extract fewer bits of information and can watch a lower resolution of the transmitted video. All this happens naturally without any receiver feedback or varying the rate of the video codec.

SoftCast achieves this goal by encoding a video using fixed-precision real numbers and maintaining that representation through the whole stack, from the video codec all the way to the physical layer. This is a significant departure from the current approach in which a transmitted video goes through a series of video coding, FEC codes, and modulation schemes that convert pixel values to codewords that lack the numerical properties of the original values. Specifically, two values that are numerically far apart, *e.g.*, 37 and 143, can be mapped to adjacent codewords, say, 01001000 and 01001001, and as a result, a single bit flip can cause a dramatic change in the rendered video. This creates a cliff effect when the chosen video and bit rates do not match the channel. In contrast, SoftCast pushes the real representation all the way to the channel by making the magnitude of the transmitted signal proportional to the pixel value. Hence, channel noise translates to a small perturbation in pixel values, *e.g.*, transforming a value like 37 to 38, which has a small impact on video quality. This allows a video to degrade smoothly as channel noise increases.

However, the naive approach of directly transmitting real pixel values is wasteful because it provides neither the *compression* performed by the video codec, nor the *error protection* offered by the PHY forward error-correcting code (FEC). Traditional compression and error-correcting schemes operate over finite-field codewords [6], and hence cannot be applied to real values. SoftCast develops a unified approach for real numbers that addresses both these challenges using power allocation. Specifically, it recognizes that the result of traditional compression and error correction is to redistribute the bits in a data stream, *i.e.* compression removes redundant bits, whereas error correction adds redundancy to allow the correction of erroneous bits. Since current wireless communication schemes are designed to give equal power to each bit, this redistribution of bits translates to a redistribution of power in the transmitted signal. SoftCast achieves the analogous effect instead by a direct redistribution of power among the real values in the data stream. A SoftCast source can compute the optimal power allocation to produce a compact and resilient representation of the video signal, subject to a hardware power budget. The SoftCast source can perform this optimization purely based on the video data, without any need for receiver feedback. This allows SoftCast to achieve the graceful degradation property of real numbers, while maintaining compactness and resilience.

We implemented a prototype of SoftCast using the WARP radio platform. We evaluated our design in comparison with two baselines: a single-layer MPEG video, and a canonical two-layer MRC. We evaluate these schemes using the Peak Signal-to-Noise Ratio (PSNR), a standard metric of video quality [38, 28].[1] Our results show the following findings:

- SoftCast delivers on the promise of smooth degradation with channel quality. Its performance scales linearly with channel SNR.
- For a diverse multicast group, SoftCast improves the average receiver's PSNR by 3.6-7.3 dB over an MPEG baseline and by 3.1-7 dB over MRC.
- While SoftCast is targeted for multicast, its performance is competitive even for a single receiver. In our testbed, SoftCast is at most 1.5 dB worse, and can be 4 dB better than MPEG for unicast.

To the best of our knowledge, SoftCast is the first system that exploits power allocation to achieve a compact and resilient representation of video using real numbers, and demonstrates its benefits via an implementation.

2 Related Work

This paper builds on foundational work in rate distortion theory, video coding, signal processing, and wireless networks to synthesize a comprehensive design for wireless video multicast.

The terms layered video, scalable video, and multiple resolution coding (MRC), all refer to an encoding technique that fragments a video stream into a base layer and a number of enhancement layers [37, 30, 47, 26]. The base layer is necessary for decoding the video stream, whereas the enhancement layers improve its quality. No enhancement layer, however, can be applied unless all lower layers are fully decoded. This strategy works well for streaming video to multiple receivers over the wired Internet, where each receiver can subscribe to the layers that match its bottleneck capacity [10, 17]. In a wireless environment, however, a receiver cannot choose which layer to receive; it receives equally from all layers. Packets received from a layer that a receiver cannot fully decode are wasted [39]. Additionally, this scheme requires solving a complex optimization problem to assign 802.11 bitrates, codec rates, and fraction of airtime to each layer. Furthermore, the optimal solution changes with group and channel dynamics. SoftCast is intrinsically different from this approach – it provides a single stream that is useful to all receivers, and has no parameters to tune.

Multiple Description Coding (MDC) encodes a single video stream into multiple independent sub streams called descriptions. The receiver can use any description to decode the video; the quality of the decoding improves with the number of received descriptions. MDC is typically used to counter packet loss due to congestion [12] or failure [13]. MDC however is unsuitable for broadcasting packets to receivers with diverse channels. Applying MDC to a wireless multicast means transmitting packets at a specific bit-rate, and hence they would reach only receivers that can support the chosen bit-rate. SoftCast is similar to MDC in that a receiver can recreate the transmitted video frame at a resolution that scales with the number of received packets. SoftCast however differs from prior work on MDC in that it is a cross-layer design that provides scalable video coding while eliminating the bitrate selection problem.

Superposition coding [6] is a PHY technique that allows a base station to send independent streams to different users. The signal of one stream is modulated with the signal of the other stream. Decoding is done using

[1] Improvements in PSNR of magnitude larger than 0.5–1 dB are visually noticeable [33, 28].

successive interference cancellation. Superposition coding however is purely a channel code, and does not address the issue of video coding. Further, it requires a highly complex receiver and a major change to the hardware [42]. In contrast, SoftCast provides an integrated scheme for both video and channel coding, without changing receiver complexity.

Similarly to analog television, SoftCast exhibits graceful degradation and can support diverse receivers. SoftCast however, is fundamentally a digital scheme both at the PHY and codec layers. In particular, SoftCast performs power allocation in software that allows it to provide a compact and resilient representation of the video signal that matches the efficiency of traditional digital schemes.

WHDI (Wireless Home Digital Interface) is a proposal for wireless high-definition video connectivity. The standard eventually aims to provide uncompressed video between cable boxes and TV screens to allow low-complexity, unbuffered HDTV [2]. Its design [1] is based on splitting the video data into important components, which are sent using a standard digital scheme, and refinements, which are sent in a raw form on the channel. In contrast, a chief goal of SoftCast is to compact the video representation before transmission, which it achieves using a novel power allocation technique.

Finally, there are many papers from both the system and multimedia communities that address various aspects of wireless video [8, 41, 4, 21], some of which have also explored multicast applications [26, 23, 36, 22]. These schemes focus on network and application layer optimizations. They however rely on the PHY to pick a particular 802.11 bit rate. Yet, any choice of bit rate works only for those receivers whose channel can support the chosen bit rate, and degrades sharply below a certain SNR. In contrast, our work recognizes that a scalable video design must span the entire stack, addressing the cliff effect both at the video encoding layer and the PHY.

3 Video Primer

A video is a stream of frames, where each frame can be represented as a matrix of pixels. A colored frame is nothing but a gray frame, a red frame, and a blue frame. The red and blue frames are one quarter the resolution of the gray frame. In this paper, we focus our description on gray frames. Red and blue frames are treated similarly to gray frames, both by MPEG and our scheme.

A raw video file is extremely large and needs to be compressed for most practical purposes. The state-of-the-art compression technique is based on the MPEG standard [19].

MPEG performs intra-frame compression to remove the redundancy within each frame. It divides a frame into small blocks of, say, 8×8 pixels and then takes a 2-dimensional Discrete Cosine Transform (DCT) of the luminance in each block. DCT is commonly used in image compression because of its energy compacting property [32]. Specifically, since images are relatively smooth, most of their energy is in the low frequency components and most of the high frequencies are close to zero. MPEG then throws away "less important" information by quantizing the DCT components, in order to achieve further compression. These quantized values are then encoded using Huffman coding, which is a variable length code that exploits the non-uniform distribution of the DCT components to expend few bits on common values and more bits on rare values.

MPEG also performs inter-frame compression to eliminate redundant information across frames [11]. In particular, it uses differential encoding, which compares a frame against a prior reference frame and only encodes the differences. It also uses motion compensation to predict the movement of a particular block across time and compress that information as short commands.

Once the compressed video is generated, it is then transmitted over the medium using an 802.11 bitrate (a joint choice of modulation and coding) usually determined by the link layer.

4 Why Is Wireless Video Multicast Problematic?

This section describes the implications for video multicast, of choices made in video coding and digital transmission.

4.1 Video Coding

The techniques described in §3 were designed to produce good compression ratios, but produce several highly undesirable effects for wireless multicast.

- The quantization of DCT components already decides the video fidelity for all receivers, and hence it forces all multicast receivers to the same quality. Receivers with good links are forced to have the same quality as the receiver with the worst quality link.
- Huffman coding is highly fragile to bit errors and packet loss. Specifically, since it uses variable length encoding, a single bit error can confuse the receiver about symbol boundaries, which makes the whole frame unrecoverable [45]. While a human user may be perfectly fine with a few pixel errors, Huffman coding requires a video to be delivered in a lossless manner [18]. Thus, a video signal, which is intrinsically robust to noise and errors, becomes highly fragile.
- The differential encoding and motion compensation that MPEG performs are far from ideal for wireless transmission. They create dependencies between different packets in a coded video, and hence the loss of some packets can affect the decoding of correctly received video packets.

4.2 Digital Transmission

It is widely known that today's digital communication systems, including 802.11, have a cliff effect [46], i.e., above a certain SNR the bit error rate stays relatively low, whereas below it the received data becomes virtually random [3]. One major reason for such behavior is forward error correction (FEC) codes used in 802.11 PHY. FEC codes can correct for a certain number of bit errors. However, when the number of errors exceeds the pre-allocated redundancy, they not only fail, but also spread the error around to correctly demodulated bits. This happens because of the convolutional nature of the code, which creates dependencies between the decoding of consecutive bits.[2]

Even, if one does not use any FEC, the underlying modulation can still create a cliff-effect. Digital modulation is a form of coding; it maps a chunk of bits to a point in a constellation space. This mapping operates in a finite field and is designed to minimize the number of bit errors. However, all bits in a video application are not equal, *e.g.*, flipping the most significant bit of a real number can have a much more dramatic impact on the video than flipping the least significant bit.

5 SoftCast Overview

We want to broadcast a single video to diverse receivers and enable each receiver to watch a video quality that is commensurate with the quality of its channel. To tackle this goal, we first need to understand the implications of diverse channels on video delivery. Two metrics can vary across receivers: 1) the subset of the transmitted packets received at that node, 2) and the signal-to-noise ratio (SNR) in the received packets. Ideally, we would like the resolution of the received video to increase proportionally to the number of captured packets and the SNR of these packets. To achieve this goal, SoftCast relies on two principles:

1. *Video packets should be created equal and independent.* At the transmitter, all coded packets of a particular video frame should have more or less equally important information. Hence, one does not need to worry about losing "special" packets that prevent decoding, and the more packets one receives, the better the quality of the decoded frame.

2. *The transmitted signal should be proportional to the fixed-precision real values representing the original video signal.* This would allow the noise in the channel, which perturbs the transmitted signal, to translate to small deviations in the video signal. When the transmitted signal is received with higher SNR (i.e., it is less noisy), the video is naturally received at a higher resolution.

SoftCast achieves these principles through a cross-layer design that retains the real-number representation of the video all the way from the video codec down through the PHY, scaling and combining these real values to ensure that all transmitted data packets are of equal importance.

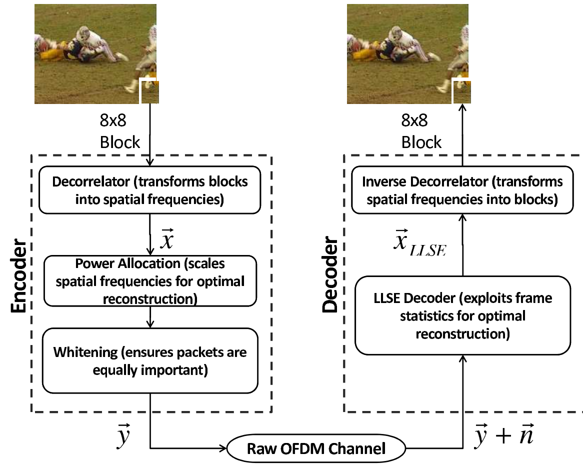


Figure 1: **Block diagram of SoftCast's codec.**

SoftCast addresses three challenges as part of its design. First, directly using the real pixel values is inefficient because of the high degree of redundancy common to video frames. However, one cannot apply traditional compression and error-correcting schemes to compact the video without losing the real value representation, because these schemes output codewords in a finite field, which do not have the graceful degradation properties of real values. SoftCast's encoder has to achieve a compact and resilient representation of the video signal while still retaining the properties of real numbers. Second, SoftCast's decoder needs to optimally reconstruct the transmitted video from the noisy received signal, and from any subset of received packets in the face of packet loss. Finally, the PHY needs to be able to transmit real values while continuing to support unmodified legacy applications. The rest of this section provides an overview of the design of SoftCast's components, which are described further in §6, §7, and §8.

(a) Encoder: Both compression and error-correction reallocate bits in a data stream to maximize the information flow from sender to receiver. Compression removes redundant bits from a data stream, while error-correcting codes add back redundancy in a manner that protects the transmitted data. With current schemes, each bit in a data stream effectively consumes equal transmission power on the medium, and hence, a reallocation of bits within a bitstream corresponds to a reallocation of power in the transmitted signal. Since SoftCast's data stream consists directly of real values rather than bits, SoftCast can achieve the same goal of maximizing the information flow (which translates to maximizing fidelity of the received video) by directly controlling the power allocation for the different real values. Specifically, given a hardware power budget, SoftCast finds the optimal scaling of the real values in the data stream that minimizes frame reconstruction errors. [3]

[2]In fact, all strong codes exhibit a cliff effect.

[3]The energy associated with a real value is the square of its magnitude.

This produces the most compact and resilient representation of the input data stream.

Scaling the magnitude of a transmitted signal provides resilience to channel noise, consider the following example. Consider a channel that introduces an error of ± 0.1. If a value of 2.5 is transmitted directly over this channel, it results in a received value of $2.4 - 2.6$. However, if the transmitted signal scales the value by $10x$, the received signal varies between $24.9 - 25.1$, and hence the scaled down received value experiences an error only of ± 0.01. However, since there is a fixed power budget, scaling up and therefore expending more power on some items translates to expending less power on other items. SoftCast's optimization finds the optimal scaling factors that balance this tension.

§6 describes the architecture of an encoder that implements this power based framework. The encoder has three modules as shown in Fig. 1. The *decorrelation* module changes the representation of the video frame to the frequency domain, which naturally distinguishes important spatial frequencies that carry the structure of the image from frequencies that provide refinement details. The output of the decorrelation model is then processed by the *power allocation* module which identifies the optimal scaling of the different frequency values. Finally, the *whitening* module transforms the signal and packetizes it to ensure that all transmitted packets carry equally important information.

(b) Decoder: The decoder, described in detail in §7, has to invert the operations of the encoder to recover the original frame. SoftCast employs a smart decoder that exploits the correlation of the video signal to reduce reconstruction errors. This is particularly important for receivers with poor channels. When the channel is noisy, the received signal values are not accurate, and our estimate of the original signal can be improved by knowing its correlation statistics. Also, SoftCast's decoder stays effective in the presence of lost packets. Since the encoder ensures that all packets have equally important information, the decoder can reconstruct the frame from any number of received packets, allowing a graceful degradation of the video signal with increasing packet loss.

(c) PHY: SoftCast's PHY layer, described in detail in §8, directly transmits the packetized real values generated by the video encoder. To do so, it leverages the fact that modern transmission systems based on OFDM use pilot signals to perform channel estimation and calibration, and hence can send the data as real numbers without affecting performance. SoftCast augments the PHY with the option to bypass FEC and modulation, and enhances the traditional socket interface used by existing applications with a raw OFDM socket interface used to transmit real-valued signals.

Lastly, before we describe each of the three pieces in detail, it's important to note that this paper focuses on intra-frame video coding, i.e., coding individual video frames, leaving inter-frame coding for future work. Our current codec exploits the correlation across blocks within a frame to perform optimal power allocation, providing a compact representation of each frame. One way to extend the method to perform inter-frame coding is to have a second level of hierarchy that applies our linear codec across frames, using similar ideas for optimal power allocation. Inter-frame coding, however, is a major area of video coding [11, 5, 15] that deserves its own study, and is beyond the scope of this paper.

Nonetheless, our evaluation compares SoftCast against MPEG with and without inter-frame coding. Our results show that, for multicast, SoftCast outperforms MPEG even when the latter is allowed to exploit inter-frame coding (see details in §10).

6 SoftCast's Video Encoder

As in MPEG, SoftCast operates on small blocks of pixels. Each frame is divided into 8×8 blocks. For instance, a 352×240 frame is converted into 1320 such blocks. SoftCast employs *a linear codec over these image blocks*.

(a) Decorrelation: The first step in encoding the frame aims at removing correlation across pixels in the same block. Since, images are relatively smooth, most of their energy is concentrated in the low spatial frequency components (slow-changing gradients), while the high spatial frequency components (small details) are close to zero. Thus, as in MPEG, SoftCast exploits this property by taking a 2-dimensional Discrete Cosine Transform (DCT) of pixel luminance in each block. The DCT has two nice properties: 1) it decorrelates the video signal since it projects it on an orthogonal basis, and 2) it redistributes the energy (the information) in a block to compact it in a few components, which typically refer to the low spatial frequencies.

As shown in Fig. 2, DCT is a linear transform that takes as input an 8×8 image block, multiplies it by the known DCT matrix, and produces an 8×8 spatial frequency representation of the block. Note that these frequencies refer to the periodicity in the image pixels, and are not related to wireless frequencies. The spatial frequency representation is then traversed in a zig-zag order (as in MPEG) to produce a 64×1 column vector, $\vec{x}$. The first row (element) of $\vec{x}$ represents the DC (zero frequency) component of the block, *i.e.* the average value of its pixels. The other rows refer to the projection of a block on some spatial frequency band, with the low spatial frequencies in the top rows and the high spatial frequencies in the bottom rows. As said above, most images are smooth and hence have most of their energy in the low spatial frequency bands, i.e., the first few rows of vector $\vec{x}$. If a row of $\vec{x}$ is consistently zero across all blocks in a frame, we ignore that row.

(b) Power Allocation: It is important to generate a compact and resilient representation of $\vec{x}$ before transmis-

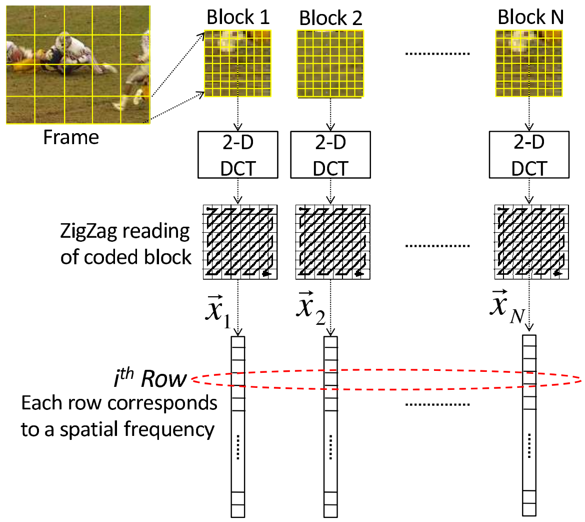


Figure 2: A frame is divided into 8x8 blocks. Each block is transformed via a 2-dimensional DCT to a 64x1 vector of spatial frequencies. The vectors corresponding to different blocks are then tiled together.

sion on the channel. Power allocation scales the magnitudes of $\vec{x}$ to provide the representation that minimizes the metric of interest, which is reconstruction errors.

To find the optimal scaling factors, we will rely on the correlation of the blocks in a frame. We consider an image block as a random vector derived from some distribution over all image blocks in the video frame. The covariance matrix of $\vec{x}$, $\Lambda_x = E[\vec{x}\vec{x}^T]$ is the average correlation between different rows in the random vector $\vec{x}$. An entry, λ_{ij} in this matrix refers to the correlation between row i and row j of $\vec{x}$, computed across all blocks in the frame. Since DCT is highly effective at decorrelating most images, the correlation between different rows is effectively zero in practice, and the only relevant elements of this covariance matrix are the diagonal elements. Given these diagonal elements, SoftCast can compute the optimal scaling matrix as described in the lemma below, which we prove in the appendix.

LEMMA 6.1. *Let $\vec{x}$ be a decorrelated signal with covariance matrix Λ_x. Let λ_i be the i^{th} diagonal element of the covariance matrix. Assume that we have a total power P, and that the channel is additive white Gaussian noise. The linear encoder that minimizes the mean square reconstruction error is:*

$$\vec{y} = G\vec{x},$$

where G is a diagonal matrix with elements

$$g_i = \lambda_i^{-1/4}\left(\sqrt{\frac{P}{\sum_{j=1}^{64}\sqrt{\lambda_j}}}\right).$$

Observe that this scaling is analogous to traditional compression and error-correction in the sense that it finds the representation that maximizes the information flow from sender to receiver, *i.e.*, minimizes the error.

(c) Whitening: We now need to assign these scaled block vectors to packets. It is unlikely that a whole frame could fit in one packet, hence we must decide how to split the blocks in a frame between multiple packets. We do not want a split that assigns full blocks to a packet because such a split means that a packet loss would cause the loss of entire blocks, and hence create patchy frames. We want a split that ensures that every packet contributes equally to the reconstruction of all blocks in the frame.

One might think that this could be addressed by assigning rows (spatial frequency bands) in blocks to packets. The problem, however, is that these rows still differ significantly in their power, and hence some packets would have higher power than others. This unequal distribution is undesirable in the face of packet loss, because the loss of high-power packets would have far greater impact on the quality of the reconstructed signal than the loss the low power packets. Thus, we want a linear transform that changes the representation of the vectors, $\vec{x}$'s, to a new representation, where all rows have equal average power. This transform also must maintain the power allocation done in the above section. Typically, communication systems use the Hadamard matrix for this purpose [27]. The Hadamard matrix is an orthogonal transform composed entirely of +1s and -1s. Multiplying by this matrix creates a whitening effect because it creates a new representation where the original rows are smeared across all dimensions.

Putting this together, we obtain:

$$\vec{y} = HG\vec{x} \tag{1}$$

Where $\vec{x}$ is the DCT representation of a block, G is the scaling matrix in lemma 6.1, H is the Hadamard matrix, and $\vec{y}$ is the encoded version of the block.

Thanks to this whitening, we can now assign rows of the $\vec{y}$'s to packets (*e.g.*, in round-robin fashion). These packets will have equal power, and hence offer better packet loss protection.

These packets are delivered directly to the PHY (via a raw socket), which interprets their data directly as the signal to be sent on the medium, as described in §8.

In addition, the encoder sends a small amount of metadata, specifically the diagonal elements, λ_i, of the covariance matrix to assist the decoder in inverting the received signal. The overhead of this metadata is low (0.005 bits per pixel in our implementation, *i.e.* less than 2% overhead), and it can be sent using the traditional mechanism at the lowest 802.11 bitrate to allow reliable reception by all receivers. The matrix G can be computed directly from this covariance metadata. Note that the DCT and Hadamard matrices are fixed and known, and do not need to be communicated.

7 SoftCast's Video Decoder

On the receive side, and as will be described in §8, the PHY estimates and corrects the attenuation and phase of the received signal using pilot signals. The end result is that for each value y_i that we sent, we receive a value y_i+n_i, where n_i is a random noise. It is common to assume the noise is additive, white and Gaussian. While this is not exact, it provides good insight and works reasonably well in practice. Hence, we can say that the received block $\vec{\hat{y}}$ satisfies the following:

$$\vec{\hat{y}} = \vec{y} + \vec{n}, \tag{2}$$

where $\vec{y}$ is the transmitted coded block, and $\vec{n}$ is a noise vector whose entries are i.i.d Gaussian variables. The driver delivers the received packets to the video application, which needs to decode them and reconstruct the original video signal.

(a) LLSE Decoder: Our goal is to decode the received frame in a manner that minimizes reconstruction errors. For the time being, let us assume that our receiver has received all packets. We can re-write the received signal as

$$\begin{aligned} \vec{\hat{y}} &= HG\vec{x} + \vec{n} \\ &= C\vec{x} + \vec{n}, \end{aligned} \tag{3}$$

where C is the encoder matrix that combines the effect of power allocation and whitening, and $\vec{n}$ is additive white Gaussian noise. Given the received signal, and the encoding matrix C (which can be computed from the received metadata), we want to compute our best estimate of $\vec{x}$.

The solution to this problem is widely known as the Linear Least Square Estimator (LLSE) [24]. Specifically, if the decoder knows the covariance matrix of the source signal Λ_x, and the covariance matrix of the channel noise, Σ, then the Linear Least Squares Estimator (LLSE) [24] estimates the original signal as:

$$\vec{x}_{LLSE} = \Lambda_x C^T (C\Lambda_x C^T + \Sigma)^{-1} \vec{\hat{y}}, \tag{4}$$

where $\vec{x}_{LLSE}$ refers to the LLSE estimate of block $\vec{x}$, C^T is the transpose of the encoder matrix C from Eq. (3). Note that the diagonal Λ_x matrix is transmitted as metadata by the encoder. The noise covariance matrix, Σ, is also a diagonal matrix, where each entry corresponds to the noise variance experienced by the packet in which the respective row was transmitted. We estimate this noise variance from the packet RSSI as reported by the PHY.[4]

Consider how the LLSE estimator changes with SNR. At high SNR, the noise is small, (i.e., $\|\Sigma\|_2 \approx 0$), and Eq. 4 becomes:

$$\begin{aligned} \vec{x}_{LLSE} &\approx \Lambda_x C^T (C\Lambda_x C^T)^{-1} \vec{\hat{y}} \quad (5) \\ &= C^{-1}\vec{\hat{y}}. \quad (6) \end{aligned}$$

Thus, at high SNR, the LLSE estimator becomes the inverse of the measurements. This is because at high SNR we can trust the measurements and do not need the correlation of the blocks Λ_x. In contrast, at low SNR, when the noise power is high, one cannot fully trust the measurements and hence it is better to readjust the estimate according to the correlation statistics of the original blocks. In §10, we present experimental results that support this intuition, namely that incorporating video correlation statistics at the decoder is beneficial for low SNR receivers.

The final steps are simple: reorganize $\vec{x}_{LLSE}$ into an 8×8 DCT block by undoing the zigzag traversal in Fig. 2, multiply by the inverse of the DCT matrix to generate the original video block, and tile the blocks into a video frame.

(b) Dealing with Packet Loss: What if a receiver experiences packet loss? How can such a receiver decode? Recall that we spread an encoded block across multiple packets to ensure that the loss of a packet reduces the resolution of all blocks in a frame rather than create a patch on the screen.

Say that our blocks are 8×8 pixels, then $\vec{y}$ has 64 elements. Depending on the size of a video frame (the format of the video), a packet may contain one or more rows from each block. For ease of exposition, let us assume that the lost packet contains the i^{th} row of the blocks. Then this loss would eliminate row y_i for $i \in \{1, 2, \ldots, 64\}$. Define $\vec{y}_{*i}$ as $\vec{y}$ after removing the i^{th} row, and similarly C_{*i} and n_{*i} as the encoder matrix and the noise vector after removing the i^{th} row. Effectively:

$$\vec{y}_{*i} = C_{*i}\vec{x} + \vec{n}_{*i}. \tag{7}$$

The LLSE decoder becomes:

$$\vec{x}_{LLSE} = \Lambda_x C_{*i}^T (C_{*i}\Lambda_x C_{*i}^T + \Sigma_{(*i,*i)})^{-1} \vec{y}_{*i}. \tag{8}$$

Note that we remove a row and a column from Σ. Eq. 8 gives the best approximation of the 64 elements in $\vec{x}$ using only 63 measurements, i.e., the approximation that minimizes the mean square errors.

The same approach extends to any number of lost packets. Clearly, this approximation becomes worse for receivers which lose more packets, and hence they experience a lower resolution video.

8 SoftCast's PHY Layer

SoftCast's PHY layer then directly transmits the real values generated by the video codec over the medium. This allows channel noise, which manifests as small perturbations of the transmitted signal, thus directly translate to

[4]The RSSI is the received signal power, which is the transmitted signal power multiplied by the square of the attenuation plus the noise power. The receiver knows the transmitted signal power. It can also estimate the channel attenuation by inserting a few known values at the beginning of each packet and see how much they get attenuated. Once you have these values and the RSSI, you can estimate the noise power.

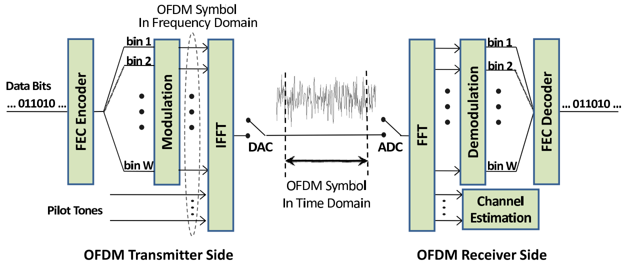


Figure 3: Diagram of a Typical OFDM System.

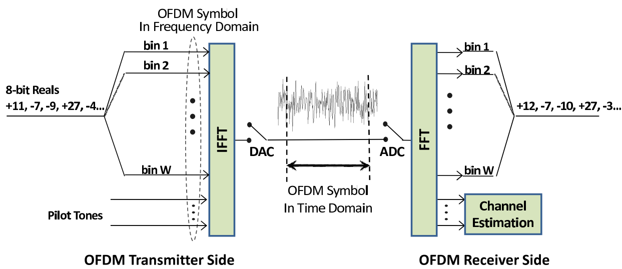


Figure 4: Diagram of SoftCast's Raw OFDM System.

Figure 5: The WARP radio platform.

Figure 6: Topology with node locations.

small perturbations of the transmitted real value (as in Eq. 2). In this section, we describe how we can modify the PHY to support transmission of real values while still maintaining compatibility with the traditional networking stack.

Our approach leverages the characteristics of Orthogonal Frequency Division Multiplexing (OFDM), which is already part of the 802.11 PHY layer. OFDM divides the 802.11 spectrum into many independent narrow subcarriers, called OFDM bins. Fig. 3 shows a schematic of a typical OFDM channel. A data stream is FEC encoded and striped across the OFDM bins. The bits in each bin are modulated to generate a signal with in-phase and quadrature components. The signals across all bins are then converted to the time domain using an Inverse Fast Fourier Transform (IFFT) and sent on the medium by the transmitter. The receiver reverses the process to retrieve the transmitted data stream.

SoftCast adds an option to the PHY interface to allow OFDM to bypass FEC and modulation, presenting a raw OFDM channel to higher layers. The application (i.e., the socket) can then decide whether to use a standard OFDM channel, or to have its data transmitted without FEC and digital modulation. Streaming media applications can choose the raw OFDM option if they want to prevent the cliff effect and allow their signal resolution to scale with the SNR of the channel. In this case, the OFDM channel will have the schematic in Fig. 4. File transfer applications and any other applications where the data can be expressed only at a single resolution, will pick the standard OFDM option, and hence will experience the standard OFDM channel in Fig. 3.

In the raw mode, the PHY does not know whether a packet contains video, audio, or other source signals. From the perspective of the PHY, it receives a stream of values that it needs to put on the raw OFDM channel. It assigns a pair of values to each data bin, mapping one to the in-phase, and the other to the quadrature component, of the signal. As before, these signals are then input to an IFFT module to generate a time-domain signal which is transmitted across the channel. At the receiver, the time signal is sent to an FFT to retrieve the values in each of the OFDM bins. These values are scaled and calibrated using standard OFDM processing. At this point, the traditional OFDM pipeline of demodulation and FEC is bypassed, and the raw values are transmitted to the receive socket. Note that these values are not subjected to the usual checksum test, since it is acceptable for the received signal to deviate from the transmitted signal.

Note that SoftCast leverages OFDM's use of a preamble and pilot bins separate from data bins [16] to perform the traditional functions of synchronization, carrier frequency offset (CFO) estimation, channel estimation, and phase tracking.

9 Experimental Environment

We implemented a prototype of SoftCast using the WARP radio platform [35], and evaluated it in comparison with the following baselines: single-layer MPEG with and without inter-frame compression, and two-layer multi-resolution coding (MRC) with inter-frame compression.

(a) Hardware: Our experiments use nodes connected to the WARP radio platform. The WARP platform, shown in Fig. 5, is similar to an actual 802.11 card, in that it has a dual-band radio that supports both the 2.4 GHz (802.11b/g) and 5 GHz (802.11a) ISM bands. Its FPGA code implements standard 802.11 transmit and receive chains, including OFDM over BPSK, QPSK, 16QAM and

64QAM modulations. Further, it allows us to modify the PHY to support the transmission of raw OFDM packets.

(b) Video Quality Metric: We compare the four schemes using the Peak Signal-to-Noise Ratio (PSNR). PSNR is a standard measure of video/image quality [38, 28]. It is a function of the mean squared error (MSE) between a decoded video frame and its original version and is computed as follows:

$$PSNR = 10\ log_{10} \frac{2^L - 1}{MSE} \quad [dB],$$

where L represents the number of bits used to encode pixel luminance, typically 8 bits, and the MSE between a decoded frame I and its original version K is:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i,j) - K(i,j)\|^2,$$

Where n and m are the number of columns and number of rows in a frame. The PSNR of a video is the average PSNR of its frames. Table 7 from [48] maps PSNR values to user perception of video quality. Note that a PSNR below 16 dB is effectively a random noise, whereas a PSNR above 37 dB is excellent quality.

Rating	PSNR Range
Excellent	> 37
Good	31- 37
Fair	25 - 31
Poor	20 - 25
Bad	< 20

Figure 7: Table provides a map between PSNR and user perception of video.

(c) Tested Videos: We use standard reference videos available at the Center for Image Processing Research (CIPR).[5] In particular, we use the SIF format which has 352×240 pixels. We experiment with two standard videos: 150-frame *tennis* and 125-frame *football*. The two videos differ in their compressibility. *tennis* at 32 dB PSNR is 2.86 times more compressible with inter-frame compression (one I frame per 12 video frames) than with intra-frame compression alone. In contrast, *football* at 32 dB PSNR is 1.59 times more compressible with inter-frame compression (one I frame per 12 video frames) than with intra-frame compression alone. For each experimental point we transmit the video once.

(d) Compared Schemes: We compare the following:

- **SoftCast:** This is our implementation of SoftCast which follows the description in §6, §7 and §8
- **MPEG-Inter:** This represent the common design in current wireless environments. Specifically, we use the open source implementation of MPEG-2 video codec provided by `FFmpeg` [9]. We select MPEG-TS as the stream format which is the common choice for streaming. The scheme uses interframe compression with the default option of one reference frame (I-frame) per 12 video frames.
- **MPEG-Intra:** This is the same as above except that it does not use inter-frame compression and hence all frames are reference frames (I-frames).
- **MRC:** This scheme uses multiresolution coding to encode the video into two layers: a base layer and an enhancement layer. It is implemented using `FFmpeg` [9], by applying the standard scheme for encoding MRC [15], which first quantizes the video to a coarse quality generating the base layer. Next, it decodes the base layer and subtracts it from the original video to obtain the residual values, which are then encoded using standard MPEG encoding and become the enhancement layer. We enabled inter-frame compression for both layers.

Comparing these schemes requires implementing an evaluation infrastructure that is suitable for running MPEG, MRC, and SoftCast.

- *MPEG and MRC:* The current version of the WARP PHY does not implement 802.11's convolutional codes (FEC). Thus, for comparison with MPEG and MRC, we implemented the 802.11 convolutional codes in software and applied them to the bit-stream before passing it to WARP. Our implementation uses Matlab reference implementation of 802.11 convolutional codes which is part of the communication toolbox [29]. It includes the scrambler, the convolutional coder, and the interleaver.
- *SoftCast:* We have implemented both SoftCast's video coder and the raw OFDM channel. The raw OFDM channel is implemented by augmenting the FPGA code on the WARP board to allow it to bypass modulation (WARP doesn't have FEC as mentioned earlier). The video coder is implemented in software according to the description in §6 and §7.

(e) Optimizing the Compared Schemes: To ensure that the comparison between these schemes is fair, we make them equal in terms of channel use, which includes average power and total airtime. Specifically, we allow each scheme to use the medium for 20% of the time, leaving 80% of the time for other applications. We also allow each scheme to optimize its parameters.

- *SoftCast:* It has no parameters to tune, and this is part of its appeal.
- *MPEG with and without interframe compression:* MPEG has to pick an 802.11 bit rate and also a codec-rate (i.e., the output rate of the codec). The codec rate is always equal to the bit rate multiplied by the fraction of air time MPEG gets. This choice ensures that the encoded video fits exactly in the available wireless throughput. As for the choice of bit rate, we allow an MPEG sender to try all possible 802.11 bit rates and

[5]http://www.cipr.rpi.edu/resource/sequences/sif.html

Parameter	Considered values
Airtime Ratio	4:1, 2:3, 3:2
Bit Rate for Base Layer	All 802.11g bit rates
Bit Rate for Enhancement Layer	All bit rates higher than base rate

Figure 8: Parameters used to optimized MRC.

pick the rate that maximizes the average PSNR across all receivers in its multicast group.

- *MRC:* Similarly to MPEG, MRC has to pick a codec-rate and an 802.11 bit-rate for each layer. Additionally, it has to pick how to divide airtime between its two layers. As in MPEG, the codec-rate for each layer is set to the layer's bit rate multiplied by its fraction of airtime. As for the rest of the parameters, we try all combinations in Table 8 and allow an MRC sender to pick the combination that maximizes the average PSNR for the receivers in its multicast group.

The above description shows that we are very conservative in our comparison, allowing the baselines schemes to optimize their parameters after knowing all information about the receivers in a multicast group.

(f) Testbed: Fig. 6 shows our node locations. Since we have only three WARP boards, we move the boards between the locations indicated in the figure to cover all possible sender receiver pairs. (Only a small number of boards have been produced at the time of writing and each board costs USD $12,000$.) Given all sender-receiver pairs, we then create multicast groups by picking one sender and various receivers. This is an accurate representation of a multicast group except in that the receivers do not receive concurrently. However, many prior measurements [31, 34] have shown that receptions at different multicast receivers are independent. Thus, the fact that we do the experiments at each receiver successively instead of concurrently should not affect the results.

For each sender receiver pair, we run the compared schemes: SoftCast, MPEG-Inter, MPEG-Intra, and MRC. As stated above, we let the two MPEG schemes and the MRC scheme optimize their parameters offline after they are given access to the multicast groups' statistics. Thus, for each location, we run MPEG-Inter with all possible 802.11 bit rates and allow it pick for evaluation the bit rate that maximizes the average PSNR for the receivers in each group, as explained. The same applies to MPEG without interframe compression. For MRC, we run all combinations in Table 8 for each location, and allow an MRC sender to pick offline the combination that maximizes the average PSNR for the receivers in each group. Note that SoftCast does not need to optimize any parameters.

10 Evaluation

We compare SoftCast with MPEG and MRC for a variety of conditions.

10.1 Video Quality in a Multicast Group

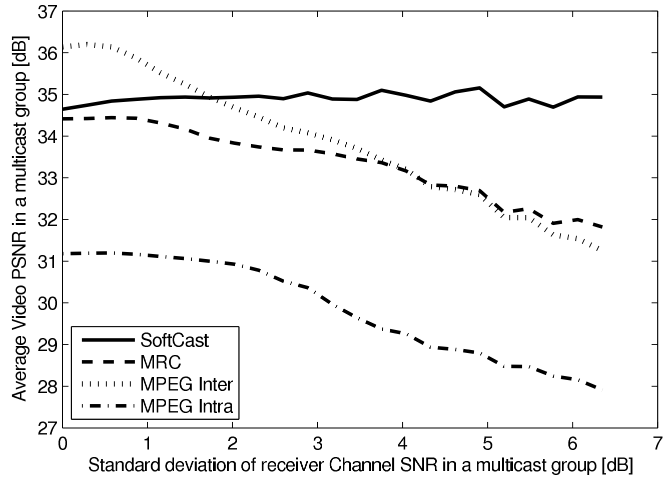
This section demonstrates that SoftCast can improve the performance of a group of diverse multicast receivers, as compared to the alternatives.

Method. In this experiment, we create 100 different multicast groups by picking a random sender and different subsets of receivers in the testbed. Each multicast group is parametrized by the standard deviation of receiver SNRs. We hold the average SNR of all multicast groups constant at 15 (± 1) dB, which is the average SNR of our testbed, and vary the standard deviation from 0-6 dB. Each multicast group has up to 20 receivers, with multicast groups with zero standard deviation having only one receiver.

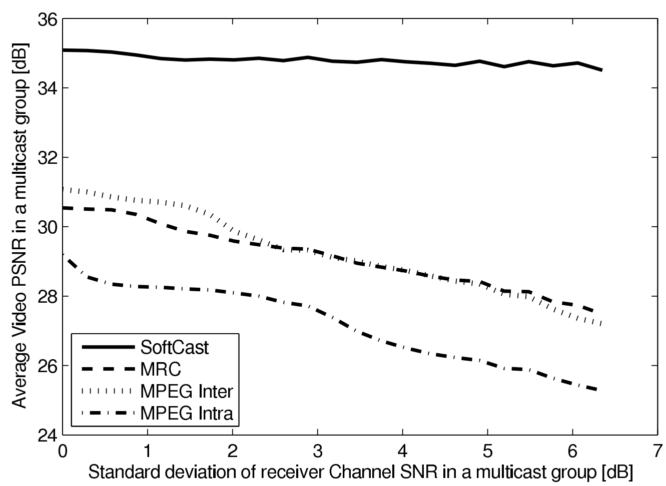
For each group, we run each of the four compared schemes. As explained earlier, for MPEG-Inter, MPEG-Intra, and MRC, we allow them to pick the optimal parameter settings for the specific group.

Results. Figs. 9(a) and (b) plot the average PSNR in a multicast group as a function of its standard deviation for two videos, *tennis* and *football*. It shows that SoftCast delivers a gain of 3-7 dB over the best baseline for diverse multicast groups. Further, SoftCast continues to deliver the same average performance for groups with increasing SNR standard deviation, in contrast to all 3 baselines whose performance degrades with increasing diversity in the multicast group. MPEG-Inter outperforms the other baselines for most of the range. This is because it takes advantage of more compression opportunities in comparison to MPEG-Intra. It might seem surprising that a single-layer MPEG outperforms MRC despite the fact that they both use inter-frame compression. This is because MRC splits wireless bandwidth between a base and an enhancement layer, and hence compromises the base quality received by the poorest receivers.

While it is expected that MPEG-Inter outperforms SoftCast for *tennis*, which has significant opportunities for inter-frame compression, one may be surprised that SoftCast can outperform all baselines even for a single receiver in the case of *football*. This effect is attributed to the following reasons: The most important reason is packet loss, which has a drastic impact on MPEG, and fairly minor impact on SoftCast. The second reason stems from the fact that MPEG has to be transmitted over a fixed 802.11 bitrate, of which there are only a few discrete options, none of which might be optimal for the particular channel instance. Similarly, the video codec is also forced to make discrete decisions by quantizing the data to match it to a particular 802.11 bitrate. Note that all of these inefficiencies come from making thresholded decisions (*e.g.*, dropping a packet for a few bit errors) which might not be optimal for the specific channel instance. SoftCast avoids

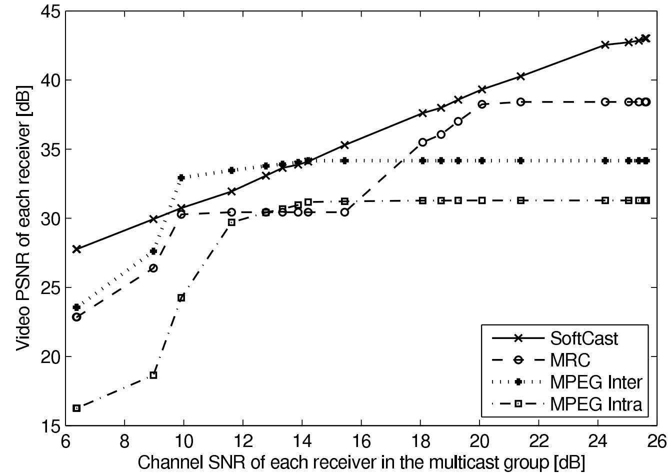


(a) *tennis*

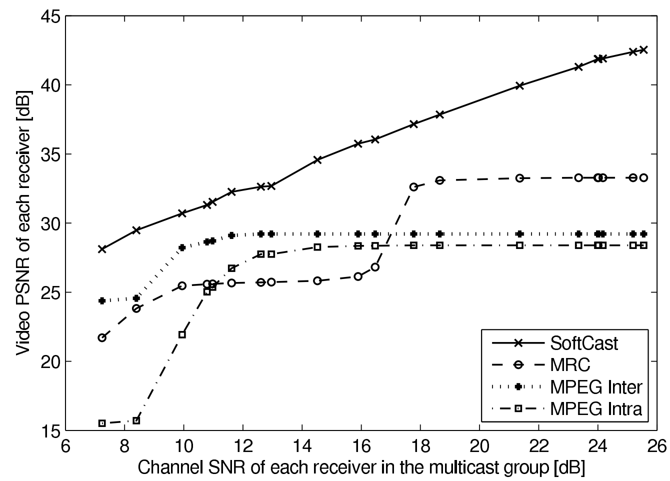


(b) *football*

Figure 9: Average PSNR across receivers in a multicast group as a function of standard deviation of receiver channel SNRs. The mean channel SNR is kept at 15dB.



(a) *tennis*



(b) *football*

Figure 10: PSNR of individual receivers in a sample group of mean channel SNR 15dB and standard deviation 6dB. Each marker represents one receiver.

these inefficiencies by maintaining the real-valued representation throughout the entire communications stack.

10.2 Zooming in on a group

In order to understand the source of SoftCast's gains for diverse receiver groups, we zoom in on a typical diverse group for both videos.

Method. The experimental setup is the same as before, but we focus our attention on a single representative multicast group. The group is composed of 20 receivers, and has an SNR standard deviation of 6dB. Each datapoint is the average of 10 experimental runs.

Results. Figs. 10(a) and (b) show the per-receiver PSNR as a function of its channel SNR. They demonstrate that SoftCast delivers on the promise of smooth degradation of received video quality with SNR. In contrast, the baseline schemes, though optimized for these specific receiver instances for this group, are unable to provide good performance to all receivers. Specifically, they show a sharp cliff effect. MRC suffers at the low end as compared to MPEG because it wastes bandwidth on the enhancement layer for these receivers. On the flip side, MPEG is inferior to MRC at the high end, since the latter has the ability to partition the receivers into two categories.

10.3 Resilience to packet loss

A key feature of SoftCast is its resilience to packet loss, due to its whitening of the data in the packets. This gain is manifested even for a single sender-receiver pair, as we demonstrate in the results below.

Method. In this experiment, we pick 10 sender-receiver pairs with an average SNR of 15dB. For each sender-receiver pair, we take an originally lossless video transfer and introduce packet losses uniformly at random with

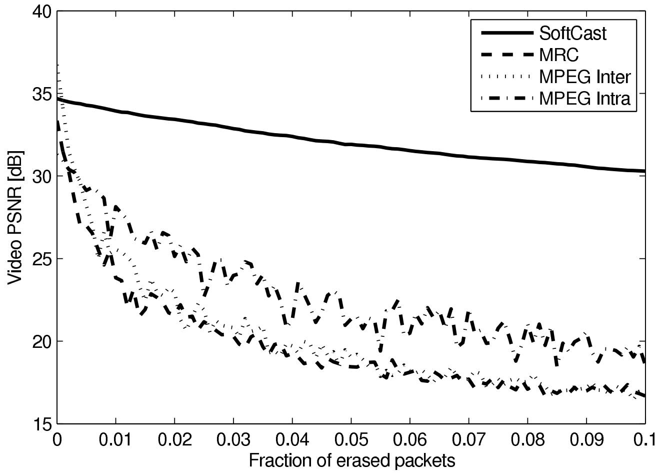


Figure 11: PSNR as a function of packet loss for ***tennis***

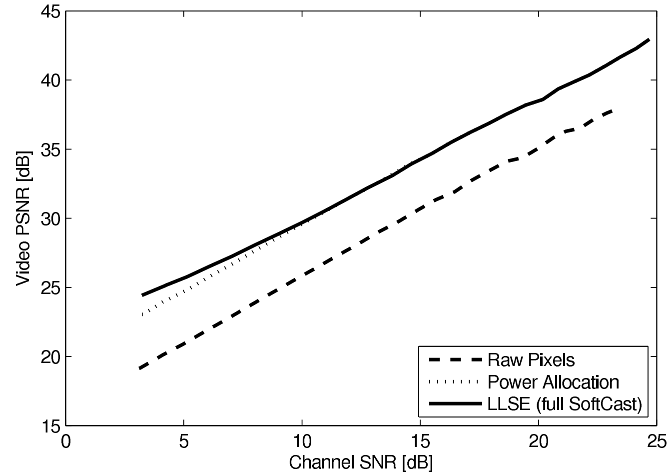


Figure 12: Microbenchmarks: The figure shows the contribution to received video PSNR from different SoftCast components for ***tennis*****.**

increasing probability.

Results. Fig. 11 plots the PSNR of the decoded video stream as a function of packet loss probability. As expected, SoftCast's performance degrades only gradually as packet loss increases, in comparison to the baselines which have a steep performance drop even for as low as 0.2% packet loss rate. SoftCast achieves this gain for two reasons: first, the whitening module ensures that all packets are equally important, and second, the decoder can recover an approximation of the frame from any subset of received packets.

It is important to note that MPEG-Inter which had the best performance for lossless unicast (as can be inferred from the PSNR for low diversity groups in Fig. 9(a)) suffers more than MPEG-Intra in the presence of packet loss. This is due to the fact that interframe compression causes dependencies between packets, and hence increases their sensitivity to packet loss. While it is possible to alleviate this sensitivity by adding erasure codes to an MPEG stream [36], this adds additional complexity, whereas SoftCast comes with natural resilience to packet loss, and such resilience can be obtained without receiver feedback or precomputed redundancy.

10.4 Understanding SoftCast

We will now explore the contribution to SoftCast's performance from each of its components.

Method. We pick 30 sender-receiver pairs that span an SNR range from 3.5 to 25 dB. We turn on different combinations of SoftCast's components (raw OFDM, power allocation, and LLSE decoding), and compute the PSNR of the received video individually for each sender-receiver pair.

Results of using Raw OFDM. We first turn off all of SoftCast's video coding functionalities (including the DCT) and focus on the raw OFDM channel. Specifically, we set the encoder and the decoder matrices to the identity matrix, and broadcast the video blocks on the raw OFDM channel without any coding.

The dashed line in Fig. 12 demonstrates that transmitting the video signal over a raw OFDM channel without any modulation or FEC, allows the received video quality to scale smoothly with receiver channel quality. This is in contrast to the well-known cliff effect of digital transmission [3], which can be also seen in the PSNR graphs of Fig. 10.

(b) Results of Power Allocation. Next we turn on the decorrelation and power allocation modules, but instead of using the optimal LLSE decoder, the receiver simply uses the inverse of the encoding matrix to decode the received signal. The dotted line in Fig. 12 shows that power allocation provides a significant PSNR gain (about 4 dB) over raw video signals throughout the entire SNR range.

(c) LLSE Decoder. Now we turn on a full-fledged SoftCast– *i.e.*, we use the entire encoder described in §6, as well the full LLSE decoder described in §7. The solid line in Fig. 12 shows the PSNR in this case. It shows that the full-fledged SoftCast maintains all of the benefits of the raw OFDM channel and the optimal encoder, but also adds a new gain at low SNRs. Recall from (4) that the LLSE estimator takes advantage of the knowledge of the correlation in the blocks of the original signal in order to improve the quality of the decoding. This information is particularly helpful at low SNR receivers where the measurements are noisy and cannot be fully trusted. It is exactly at these receivers that these gains are most valuable, because the video quality they receive is on the boundary of acceptability, and small increases in PSNR can lead to big improvements in perception.

11 Conclusion

This paper introduces SoftCast, a new approach to wireless video multicast, that significantly outperforms traditional schemes for groups with diverse receivers, while still providing competitive performance for individual receivers. It does this through a cross-layer design that retains the real valued representation of the video signal throughout the network stack from the codec to the PHY. SoftCast develops a novel power allocation scheme that provides a compact and resilient real-valued representation of the video signal, and an enhancement to the PHY to directly transmit real values on the channel. Put together, SoftCast provides an elegant framework that has no parameter tuning, requires no receiver feedback, and scales gracefully with receiver SNRs and packet delivery rates.

Acknowledgments: We thank Nate Kushman for several insightful discussions, and Grace Woo for her work on an earlier version of the design. This work is supported with a Cisco gift.

References

[1] Amimon Inc. Enabling wireless uncompressed HDTV connectivity with a unique video-modem approach: A technical overview.

[2] Amimon Inc. Wireless HDTV – Compressed or Uncompressed? That is the question.

[3] Audioquest Technologies. Understanding HDMI Eye Pattern, BER and Cliff Effect.

[4] J. Byers, M. Luby, and M. Mitzenmacher. A digital fountain approach to asynchronous reliable multicast. *Selected Areas in Communications, IEEE Journal on*, 20(8):1528–1540, Oct 2002.

[5] R. J. Clarke. Image and video compression: A survey. Technical report, Department of Computing and Electrical Engineering, Heriot-Watt University.

[6] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley and Sons, 1991.

[7] K. Elleithy, T. Sobh, A. Mahmood, M. Iskander, and M. Karim. Efficient support of wireless video multicast services in 3G and beyond. In *Advances in Computer, Information, and Systems Sciences, and Engineering*, 2005.

[8] M. Etoh and T. Yoshimura. Advances in wireless video delivery. *Proceedings of the IEEE*, 93(1):111–122, Jan. 2005.

[9] FFMPEG. http://www.ffmpeg.com.

[10] S. Floyd, V. Jacobson, C.-G. Liu, S. McCanne, and L. Zhang. A reliable multicast framework for light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, 5(6):784–803, 1997.

[11] D. L. Gall. Mpeg: a video compression standard for multimedia applications. *Commun. ACM*, 34(4):46–58, 1991.

[12] V. Goyal. Multiple description coding: compression meets the network,. *IEEE Signal Processing Magazine*, 18(5):74–93, 2001.

[13] V. K. Goyal, J. Kovaeevi, and J. A. Kclncr. Quantized frame expansions with erasures. *Applied and Computational Harmonic Analysis*, 10:203–233, 2001.

[14] L. Hanzo, P. Cherrimana, and J. Streit. *Wireless Video Communications: Second to Third Generation and Beyond*. John Wiley, 2001.

[15] B. G. Haskell, A. Puri, and A. N. Netravali. *Digital Video: An Introduction to MPEG-2*. Springer, 1997.

[16] J. Heiskala and J. Terry. *OFDM Wireless LANs: A Theoretical and Practical Guide*. Sams Publishing, 2001.

[17] D. Hoffman and M. Speer. Hierarchical video distribution over internet-style networks. volume 1, pages 5–8, Sep 1996.

[18] Huifang Sun and Xuemin Chen and Tihao Chiang. *Digital Video Transcoding for Transmission and Storage*. CRC Press, Piscataway, NJ, USA, 2005.

[19] International Organization for Standardization. *Overview of the MPEG-4 Standard*. 2002.

[20] Jozsef Vass and Xinhua Zhuang. Multiresolution-multicast video distribution over the Internet. In *Wireless Communications and Networking Conference (WCNC)*, 2000.

[21] J. Karlsson, H. Li, and J. Eriksson. P2P Video Multicast for Wireless Mobile Clients, 2006.

[22] D. Koutsonikolas, S. Das, Y. C. Hu, and I. Stojmenovic. Hierarchical geographic multicast routing for wireless sensor networks. *Sensor Technologies and Applications, International Conference on*, 2007.

[23] R. Kumar, A. Paul, and U. Ramachandran. Fountain broadcast for wireless networks. In *Second International Workshop on Networked Sensing Systems*, 2005.

[24] C. Lawson and R. Hanson. *Solving Least Squares Problems*. Society for Industrial Mathematics, 1987.

[25] K.-H. Lee and D. P. Petersen. Optimal linear coding for vector channels. *IEEE ToC*, Dec 1976.

[26] A. Majumdar, D. Sachs, I. Kozintsev, K. Ramchandran, and M. Yeung. Multicast and unicast real-time video streaming over wireless lans. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6):524–534, 2002.

[27] Martin Schnell. Hadamard codewords as orthogonal spreading sequences in synchronous DS CDMA systems for mobile radio channels. In *IEEE Third International Symposium on Spread Spectrum Techniques and Applications*, 1994.

[28] M. O. Martínez-Rach, O. López, n. Pablo Pi M. P. Malumbres, J. Oliver, and C. T. Calafate. Quality assessment metrics vs. PSNR under packet loss scenarios in MANET wireless networks. In *MV '07: Proceedings of the international workshop on Workshop on mobile video*, pages 31–36, New York, NY, USA, 2007. ACM.

[29] Mathworks Inc. MATLAB IEEE 802.11a WLAN model. MATLAB Communications Toolbox.

[30] S. McCanne, M. Vetterli, and V. Jacobson. Low-complexity video coding for receiver-driven layered multicast. *Selected Areas in Communications, IEEE Journal on*, 15(6):983–1001, Aug 1997.

[31] A. K. Miu, H. Balakrishnan, and C. E. Koksal. Improving loss resilience with multi-radio diversity in

wireless networks. In *MOBICOM*, 2005.
[32] A. Oppenheim. *Digital Signal Processing.* Prentice-Hall, Englewood, NJ, USA, 1975.
[33] P. W. Rabbani, M.; Jones. *Digital Image Compression Techniques.* Bellingham (WA):SPIE Optical Engineering Press, 1991.
[34] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based models of delivery and interference. In *SIGCOMM*, 2006.
[35] Rice University. Rice University Wireless Open-Access Research Platform. WARP.
[36] L. Rizzo and L. Vicisano. Rmdp: an fec-based reliable multicast protocol for wireless environments. In *ACM Mobile Computer and Communication Review*, 1998.
[37] A. Said and W. Pearlman. An image multiresolution representation for lossless and lossy compression. *Image Processing, IEEE Transactions on*, 5(9):1303–1310, Sep 1996.
[38] D. Salomon. *A Guide to Data Compression Methods.* Springer, 2002.
[39] T. Stockhammer. Is fine-granular scalable video coding beneficial for wireless video applications? In *IEEE International Conference on Multimedia & Expo (ICME)*, 2000.
[40] M.-T. Sun and A. R. Reibman. *Compressed Video Over Networks.* Marcel Dekker, 2000.
[41] R. Talluri. Error-resilient video coding in the ISO MPEG-4 standard. *Communications Magazine, IEEE*, 36(6):112–119, Jun 1998.
[42] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication.* Cambridge University Press, 2005.
[43] VLC. Streaming HowTo with VideoLAN Manager. VLC Multiple Streams.
[44] W. Wei. *Multipath Unicast and Multicast Video Communication over Wireless Ad Hoc Networks.* PhD thesis, EECS Department, University of California, Berkeley, Dec 2006.
[45] S. B. Wicker. *Reed-Solomon Codes and Their Applications.* IEEE Press, Piscataway, NJ, USA, 1994.
[46] Wikipedia. Cliff Effect. http://en.wikipedia.org/wiki/Cliff_effect.
[47] D. Wu, Y. Hou, and Y.-Q. Zhang. Scalable video coding and transport over broadband wireless networks. *Proceedings of the IEEE*, 89(1):6–20, 2001.
[48] O. Zigmund. Network-driven adaptive video streaming in wireless environments. In *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2008.

APPENDIX

Proof of Lemma 6.1

We want to determine the set of optimal gains (linear multipliers) for each band that minimizes the expected reconstruction error (computed as mean square error), while within the total power constraint, as inspired by [25]. Note that, since the DCT transform is orthogonal, the mean square error of $\vec{x}$ is directly proportional to the mean square error in the video frame.

Let us model the channel as one with additive white noise (e.g. Gaussian). Thus, for each value in band i, x_i, we transmit $y_i = g_i x_i$, and the receiver receives $\hat{y}_i = y_i + n$, where g_i is the linear gain for this band, and n is a random variable with zero-mean and specific variance, σ (the same for all bands). Subsequently, the receiver decodes

$$\hat{x}_i = \frac{\hat{y}_i}{g_i} = x_i + \frac{n}{g_i}.$$

The expected mean square error is:

$$err = E\left[\sum_i (\hat{x}_i - x_i)^2\right] = \sum_i \frac{E[n^2]}{g_i^2} = \sum_i \frac{\sigma^2}{g_i^2}$$

Clearly, the best gain would be infinite, if not for the power constraint. Let $\lambda_i = E[x_i^2]$ be the power of band x_i, $\mu_i = E[y_i^2]$ be its power after applying the gain, and P the total power budget. Formally, the problem is as follows.

$$\begin{aligned} \min err &= \sigma^2 \sum_i \frac{\lambda_i}{\mu_i} \qquad (9) \\ \text{subject to:} & \\ \sum_i \mu_i &\leq P \\ \mu_i &\geq 0 \end{aligned}$$

We can solve this optimization using the technique of Lagrange multipliers. The Lagrangian is

$$L = \sigma^2 \sum_i \frac{\lambda_i}{\mu_i} + \gamma\left(\sum_i \mu_i - P\right)$$

Differentiating separately by μ_i and γ and setting to 0, yields:

$$\begin{aligned} \sqrt{\gamma} &= \sum_j \sqrt{\lambda_j \sigma^2}/P \\ \mu_i &= \sqrt{\frac{\lambda_i \sigma^2}{\gamma}} = P\frac{\sqrt{\lambda_i}}{\sum_j \sqrt{\lambda_j}} \\ g_i &= \sqrt{\frac{\mu_i}{\lambda_i}} = \sqrt{\frac{P}{\sqrt{\lambda_i}\sum_j \sqrt{\lambda_j}}} \end{aligned}$$

The optimal gain for each band is therefore such that the resulting power of the row is proportional to the *square root* of its original power. Some readers might find it more intuitive that the optimal solution should completely equalize the resulting power, i.e. $\mu_i = P/k$; but substituting this in (9) shows otherwise.