Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2008-063 October 18, 2008

One Vide Stream to Serve Diverse Receivers

Szymon Chachulski, Grace Woo, and Dina Katabi

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

CSAIL

One Video Stream to Serve Diverse Receivers

Szymon Chachulski
szym@mit.edu

Grace Woo
gracewoo@mit.edu

Dina Katabi
dk@mit.edu

Abstract

The fundamental problem of wireless video multicast is to scalably serve multiple receivers which may have very different channel characteristics. Ideally, one would like to broadcast a single stream that allows each receiver to benefit from all correctly received bits to improve its video quality.

We introduce Digital Rain, a new approach to wireless video multicast that adapts to channel characteristics without any need for receiver feedback or variable codec rates. Users that capture more packets or have fewer bit errors naturally see higher video quality. Digital Rain departs from current approaches in two ways: 1) It allows a receiver to exploit video packets that may contain bit errors; 2) It builds on the theory of compressed sensing to develop robust video encoding and decoding algorithms that degrade smoothly with bit errors and packet loss. Implementation results from an indoor wireless testbed show that Digital Rain significantly improves the received video quality and the number of supported receivers.

1 INTRODUCTION

Wireless video multicast is increasingly popular. Emerging applications are driven by users' desires to watch live TV and sporting events on laptops and handheld devices [17, 44]. There is also interest in broadcasting promotional clips, security videos, and entertainment clips, in malls, airports, and train stations [23]; universities want to broadcast their lectures live everywhere on campus; and, rich multimedia homes create a market for multi-room audio and video systems that stream the same movie to multiple screens around the house [1, 50].

Wireless video multicast, however, differs from its wired counterpart in that it needs to deal with the challenging characteristics of the wireless medium. Specifically:

1. *Wireless is prone to errors.* Current video encoding, based on MPEG, is highly vulnerable; a few lost packets or flipped bits can dramatically reduce the quality of a video stream and render the frames unrecognizable [20]. The wireless medium however suffers relatively high error rate. Typically, for unicast, this problem is solved by having the source pick a proper bit-rate (i.e., modulation and FEC) to ensure that the encountered bit-errors are corrected. But video multicast has multiple receivers that differ in link quality, and hence one cannot pick a single bit rate that fits all receivers.

2. *Wireless channel quality is highly time-varying.* The quality of the wireless channel varies quickly with time due to both fading and congestion. Existing video codecs based on MPEG pick a preset video rate and cannot adapt the video quality to quickly changing channel conditions.

3. *Wireless channel quality widely varies across receivers.* The video stream should be delivered to multiple receivers, each of which has a different channel quality. Transmitting a separate stream to each receiver is wasteful. However, broadcasting a single stream to all receivers while ensuring that they can all decode requires the source to transmit at a low bit rate that satisfies the receiver with the worst channel quality, hence reducing everyone to the performance of the worst receiver in the multicast group.

Current approaches to wireless video multicast are unable to deal with all of the above characteristics of the wireless medium. They may use MPEG to encode the video into a preset fixed quality stream. The stream is protected with strong error-correcting codes (FEC) and is then broadcast on the wireless medium to multiple receivers [44, 23, 20]. However, wireless receivers can span a wide range of throughputs, packet loss and bit-error rates [30, 2, 45]. Adjusting the error correcting code to support all receivers reduces everyone to the quality of the worst receiver in the group. Other schemes may transmit layered video, where the base layer is encoded at a low resolution and protected with heavy error correcting codes, making it decodable by all or most receivers [47, 22]. The enhancement layer improves the quality of the base layer video, and is encoded with less redundancy and hence decodable only by receivers with good link quality. While layered video is effective for wired multicast, where a congested receiver joins only the base layer [19], it is wasteful in a wireless network, where transmissions are broadcast on a shared medium. Wireless receivers cannot pick which layer they receive, and hence receive equally from all layers; all bytes they hear from a layer that they cannot completely decode are wasted.

This paper introduces Digital Rain, a new approach for streaming video to heterogeneous wireless receivers. Digital Rain is adaptive and simple; the source does not encode the video into a preset fixed quality stream or apply FEC to protect transmissions. Digital Rain develops a novel rateless video codec, where the source simply broadcasts coded packets belonging to a video stream. Receivers exploit all received video packets, even those with bit errors. Receivers that collect more packets, or experience fewer bit errors in each packet, see higher video quality, without any retransmissions or changes to the codec rate. Digital Rain addresses the challenges presented by the wireless medium:

1. *It adapts to varying link quality:* Digital Rain leverages recent advances in the theory of compressed sensing to compress the video in a robust and rateless manner [15, 8, 7]. Specifically, video frames are natural images, which do not change much from one pixel to the next. As a result, the frequency representation of a video frame is a sparse signal. The theory of compressed sensing states that sparse signals can be efficiently compressed using random linear projections [15, 8, 7], and that the quality of the recovery scales with the number of received linear combinations [18]. Digital Rain encodes a frame by taking random linear combinations of the values of its pixels. A receiver will get a certain number of linear combinations depending on the channel quality at that instant, and therefore will be able to decode a video stream with proportional quality. When the channel is good, it will decode a higher quality stream and a lower quality one when the channel is bad. The source itself does not need to change its encoding strategy to adapt to the channel quality.

2. *It deals with diverse receivers:* Digital Rain naturally adapts to receivers with diverse channel quality. The source broadcasts random linear combinations of the pixels in a video stream. Receivers with better links will get more linear combinations than ones with lower quality links. Since decoded video stream quality is proportional to the number of linear combinations received, receivers decode a video stream whose quality is commensurate with the quality of their links; they are not limited to the same quality as the receiver with the worst link.

3. *It stays efficient in the presence of bit errors.* Current wireless networks retransmit a packet because of a few bit errors, ignoring that most of the bits have been correctly received, and hence wasting wireless bandwidth [26, 24, 49]. In contrast, Digital Rain equips compressed sensing with a novel decoding algorithm that identifies erroneous bits in a corrupt packet, excludes them and uses the correctly received bits in video decoding. Thus, the waste incurred in throwing away entire packets due to a few bit errors is eliminated. Prior approaches that identify corrupt bits using on soft information [26, 49, 24] exist, but they require hardware modifications. Digital Rain's approach is entirely software based and works on existing hardware.

Finally, Digital Rain's encoding is fundamentally similar to finite field based network coding. However, Digital Rain takes linear combinations over real numbers, while traditional network coding [9, 31, 27, 4] takes linear combinations over finite fields. Finite field operations make traditional network coding rigid; it has an all or nothing behavior: when a batch of n packets is coded together, the receiver needs to get n coded packets before it can decode [9]. But the simple yet fundamental shift from finite fields to real fields, allows Digital Rain to build new network codes that do not have the all or nothing behavior while retaining the desirable properties of traditional network coding. Specifically, a Digital Rain receiver that does not receive enough coded packets, rather than giving up on decoding the coded batch, can seek an approximate decoding that produce a low resolution frame. Using these ideas Digital Rain develops new network coding protocols for media streaming that do not suffer from the all-or-nothing behavior, and hence are suitable for video and audio applications.

We have built a prototype of Digital Rain and evaluated it in a 18-node wireless testbed. We measure video quality using the Peak Signal-to-Noise Ratio (PSNR), a standard metric for video applications, where improvements in PSNR of magnitude larger than 0.5 dB are visually noticeable [40, 36]. Our results reveal the following findings:

- Digital Rain maintains good video quality and scales to a large number of receivers. Specifically, it improves the average video quality across receivers by 3 dB over single-layer multicast with FEC, 7 dB over layered video, and 22 dB over transmitting separately to each receiver.
- While Digital Rain builds on the idea of compressed sensing, its main strength comes from its novel decoder which improves video quality by as much as 15 dB in the presence of bit errors.
- Combined with network coding, Digital Rain produces a video quality gain of 8 dB over tree-based multicast routing, and 5 dB over opportunistic routing without network coding.

2 RELATED WORK

This paper builds on a rich literature that spans diverse areas in signal processing, networking, and video coding.

(a) Compressed Sensing. Compressed sensing is an emerging area of signal processing. It started in 2004 with the pioneering work of Donoho [15] and Candes [8, 7], who showed that sparse signals can be accurately reconstructed from a small number of random projections. Specifically, in order to capture a sparse signal, $\mathbf{x}$, of very high dimension n, it often suffices to compute a measurement vector $A\mathbf{x}$, where the matrix A is a "random" linear mapping into a low m-dimensional space. This insight has motivated many papers to explore the conditions under which compressed sensing is effective, develop tighter bounds, and explore potential applications. Applications emerged in signal acquisition and modulation [16], geophysics [32], biosensing [34], Radar [6], sensor networks [5], and image processing and camera design [21].

Digital Rain employs compressed sensing for video encoding, but differs from prior work in that area in three main ways: 1) Digital Rain has a new decoding algorithm that identifies erroneous video measurements (elements of $A\mathbf{x}$ with bit errors) and prevents them from degrading the reconstructed signal; 2) Digital Rain combines compressed sensing with the idea of exploiting partially correct packets, and weaves them together in a system architecture for wireless video multicast; 3) Finally, Digital Rain is implemented and evaluated in a actual wireless testbed.

(b) Video Multicast & Video Encoding Early proposals for video multicast over the Internet have used distributions trees [37, 29]. The success of Bittorrent, however, has motivated a peer-to-peer approach for video streaming, which has led to implementations like Joost and P2PTV [51, 12, 42]. Recent papers have extended this peer-to-peer design to the wireless environment [2]. Digital Rain differs from this work because it focuses on serving diverse receivers with a single video broadcast, employs novel video encoding and decoding schemes, and integrates its design with network coding.

A rich literature addresses the problem of video encoding [41, 23, 44]. Two approaches are particularly targeted at networked applications. The first approach is scalable video coding [41, 22], where a video is encoded into a base layer providing basic video quality at low bit rate and one or more enhancement layers. In contrast to Digital Rain which adopts a "one size fits all" approach, scalable video coding follows a "take what you need and ignore the rest" strategy. This strategy works well for streaming video to multiple receivers over a wired network, where each receiver can subscribe to the layers that match its bottleneck capacity, but is ineffective in wireless environment, where a receiver cannot pick which packets it correctly receives [43]. The second approach is multi description coding [22], where a sender transmits different descriptions of an image to different users. Each user uses her description to see a low resolution version of the image. Alternatively, the users can combine their descriptions to obtain a better quality. This form of MDC, however, is inefficient over wireless because it ignores the broadcast nature of the medium, and that a receiver cannot control which description it receives.

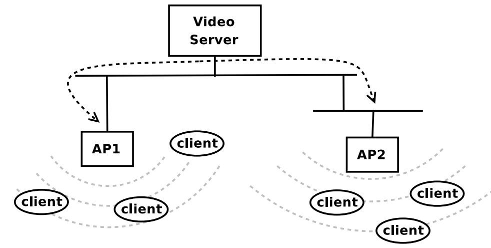

Figure 1: **Digital Rain's System Architecture.** The system has three components: the video server, the access points, and the video clients. The server multicasts the video to the APs over the local Ethernet. Each AP broadcasts the stream on its wireless interface. A client receives the stream from the AP with which it associates.

(c) Wireless Network Coding. This paper is closely related to prior work on intra-flow wireless network coding [25, 39, 48, 33] and particularly MORE [9]. Intra-flow network coding however is "all-or-nothing"; a whole batch of packets is either decoded correctly or completely lost. Video and audio streaming applications however do not require full reliability, and would rather trade off high video quality to limit delay and ensure proper streaming. Digital Rain differs from prior work on network coding because it integrates network coding with compressed sensing, applies it to wireless video multicast, and introduces a new decoder that smoothly scales the video quality with errors and losses in received packets.

(d) Partial Packet Recovery. Recent work has observed that current wireless receivers drop an entire packet with a few bit errors, though most of the bits are correctly received. They proposed using soft values from the physical layer to recover partially correct packets [24, 49]. Digital Rain builds on this prior work, but differs from it significantly. In Digital Rain, there is no need to recover incorrect bytes since the coding makes all bytes equally useful, and allows them to replace each other. Thus, Digital Rain extracts from a corrupt packet as many of its correct bytes as possible and ignores the rest. Additionally, Digital Rain does not need soft information or access to the physical layer.

3 DIGITAL RAIN

Digital Rain is a new protocol for video multicast over a wireless network. It broadcasts a single video stream but each receiver observes a video quality that scales with the quality of its channel. Digital Rain is designed for environments with moderate to large number of receivers with diverse channel quality, e.g., broadcasting live lectures on a university campus.

3.1 Digital Rain Architecture

Figure 1 shows Digital Rain's architecture, which includes three components: the video server, the access points, the video clients.

(a) Video Server: The server encodes the video as described in §3.2, and multicasts the encoded stream to the local access points using an Ethernet multicast address. The server design is highly scalable. It need not know about individual clients, track packet losses, or adapt the video codec to channel quality. The code ensures that more received packets naturally translate to higher video quality.

(b) Access Points: The APs are configured to broadcast the received video multicast stream on the wireless interface. The APs remain unchanged for unicast traffic and applications other than video multicast.

(c) Wireless Video Clients: The wireless card at the client appears to the kernel as two virtual interfaces. The first interface operates as a regular 802.11 station associated with an access point. The second interface operates in the monitor mode and hence can receive any packets including those with bit errors. This way the client can receive packets with bit errors for media applications, while supporting other applications in the traditional way. The card virtualization is already supported by popular drivers, e.g., Madwifi [35].

Digital Rain inserts a shim layer on top of the monitor interface to allow interested applications to receive packets that contain bit errors, i.e., packets that do not satisfy the 802.11 checksum. The shim layer opens a raw socket to the monitor device to receive all packets on that interface, and exports standard Unix sockets to interested applications. The video application opens a socket to the shim layer and receives the streamed packets and decodes them according to the algorithm in §3.3.

In addition to the single hop mode shown in Figure 1, where the receivers obtain their video stream directly from a nearby AP, Digital Rain also operates in the multi hop mode. As described in §3.4, Digital Rain exploits opportunistic reception in multi hop networks, by having nearby receivers forward linear combinations of their received video packets to distant receivers.

3.2 The Encoder

Digital Rain uses a new approach to video coding that is customized for lossy and error prone wireless environments. The main strength of this code is its ability to smoothly scale the video quality with the wireless channel quality, in order to accommodate differences in link quality across space (i.e., different receivers) or time (congestion periods).

A video is a stream of frames, where each frame can be represented as a matrix of pixels. A raw video file is extremely large and needs to be compressed for most practical purposes. Most video compression techniques are based on two basic concepts. There is a lot of intra-frame and inter-frame redundancy present in a video stream. We do not discuss inter-frame redundancy and motion compensation here, since Digital Rain uses standard techniques [20] to handle them.

To remove intra-frame redundancy, MPEG for example, first divides a frame into small blocks of say 8×8 pixels and then takes a Discrete Cosine Transform (DCT) of the luminance in each block. DCT is commonly used in image compression because of its energy compacting property [38]. 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. Second, MPEG throws away "less important" information by quantizing these DCT components to achieve further compression. These quantized values are then encoded using Huffman encoding which is a variable length code that exploits the non-uniform distribution of the DCT components to expend few bits on the common values and more bits on rare values.

While these techniques produce good compression ratios, they also produce undesirable effects for wireless multicast. The quantization already decides the video fidelity for all receivers, and hence it forces all multicast receivers to the same quality. Huffman encoding, on the other hand, 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 rendering the whole frame irrecoverable [47].

As in MPEG, Digital Rain divides each frame into small blocks on which it applies a DCT. However, in order to tackle the undesirable effects for wireless multicast, Digital Rain does away with quantization and Huffman encoding. Instead, Digital Rain exploits the sparsity of the DCT representation to compress the video frame. In particular, the theory of compressed sensing shows that one can accurately reconstruct a sparse signal from a small number of random projections [15, 8, 7]. Let $\mathbf{x}$ be an n-dimensional vector that refers to the DCT coefficients from all blocks in a frame. Digital Rain compresses the frame by taking linear random projections of the DCT vector, $\mathbf{y} = A\mathbf{x}$, where A is a $m \times n$ matrix with $m < n$, and y is an m-dimensional vector that we call *the measurement vector*. Each element of $\mathbf{y}$ is encoded in 8 bits using fixed-point representation. As explained in §3.3, the larger m is, the higher the quality of the reconstructed frame [18].

It is typical to choose the matrix A by picking m random rows from the Fast Fourier Transform (FFT) after randomly permuting its columns [13]. This approach allows for fast encoding and decoding because the FFT of a vector of size n can be done in $O(n \log n)$ steps, whereas standard matrix multiplication is $O(n^2)$.

Each transmitted packet, p_i, contains a subset of the measurement vector $\mathbf{y}$ as follows:

$$\mathbf{y} = \begin{pmatrix} p_1 \\ \vdots \\ p_d \end{pmatrix} = \begin{bmatrix} A_1 \\ \vdots \\ A_d \end{bmatrix} \mathbf{x} = A\mathbf{x}, \qquad (1)$$

where A_i refers to the rows of A that fit in packet p_i. Packet p_i also contains, s, the seed used to permute the FFT matrix, the video frame id f_{ID}, and the packet index i. This meta data is protected from bit errors using a strong error correcting code. The overhead of such code is negligible given that the indices and the seed are only 2 bytes each.

The number of packets that the server transmits per frame, d, should be bounded by the maximum bandwidth the video is allowed to consume on the wireless channel. For example, if the maximum video throughput should be bounded by 6 Mb/s and the video server generates 30 frames per second, d should be $6000/(30*1500*8) \approx 16$ 1500B packets.

Two points are worth noting about our video codes:

- Digital Rain provides in network compression of video data in order to allow the video to match the quality of a receiver's channel. Specifically, consider a receiver with a relatively poor channel. Say that instead of receiving all d packets of a particular frame, the receiver gets $p_1 \dots p_k$ where $k < d$. We can express the information that the receiver obtains as:

$$\mathbf{y}' = \begin{pmatrix} p_1 \\ \vdots \\ p_k \end{pmatrix} = \begin{bmatrix} A_1 \\ \vdots \\ A_k \end{bmatrix} \mathbf{x} = A'\mathbf{x}, \quad \text{for k} \leq \text{d}. \quad (2)$$

  Hence, the receiver obtains a fewer linear projections $\mathbf{y}' = A'\mathbf{x}$, where the matrix A' has fewer rows than A, and thus provides a lower quality representation of the original image $\mathbf{x}$ [18] (see Section §3.3). Said differently, Digital Rain does not need to predict the quality of the channel or adapt the codec rate to it, the adaptation occurs naturally when the channel drops packets.
- The Huffman code used in MPEG is venerable to errors and packet loss. To protect the transmission one would need to encapsulate the packets with strong error correcting codes. Such encapsulation, however, prevents real time tradeoffs between the level of compression - i.e., the quality of the image- and the redundancy in the error correction code - i.e., the quality of the channel. In contrast, Digital Rain uses one code. It allows a receiver with a bad channel to use the extra information to correct byte errors, and a receiver with good channel to use the same information to improve its video quality.

3.3 The Decoder

The basic operation of the decoder is to solve a few linear equations. Again, say that the receiver got the packets $p_1 \dots p_k$ of a particular frame, and that these packets can be expressed according to Equation (2), the receiver then needs to solve the linear system:

$$\text{find } \mathbf{x} \text{ such that } \mathbf{y}' = A'\mathbf{x} \quad (3)$$

Note that the receiver can regenerate the matrix A' by first generating the matrix A, then picking the rows that correspond to its received packets. Given the seed s, the matrix A is easy to generate by permuting the rows and columns of the FFT matrix.

The system of equations in (3) has two important properties.

- First, it is underdetermined, i.e., it has more unknowns (the elements of $\mathbf{x}$) than equations. This is because the matrix A' has more columns than rows, which is the result of packet loss, as well as the original compression in Equation (1).
- Second, it has corrupt measurements. Some elements in $\mathbf{y}'$ include bit errors and hence may dramatically deviate from their original values.

(a) Solving an underdetermined system: Let us first assume there are no bit-errors in $\mathbf{y}'$ and focus on solving Equations (3). Since the system has more unknowns than equations, it allows a whole space of potential solutions. We can, however, identify the best solution by exploiting the sparsity of the signal $\mathbf{x}$. As described in §3.2, this signal is a block-by-block DCT of a natural image, and hence most of its components are of negligible magnitude or 0. We can use this property to make the system (3) better determined. In fact, one of the core results of compressed sensing[8] is that if the n-dimensional vector, $\mathbf{x}$, is *sparse*, i.e., the number of non-zero elements is $\ll n$, then it is possible to recover $\mathbf{x}$ exactly by solving the following convex optimization

$$min||\mathbf{x}||_1 \text{ subject to } \mathbf{y}' = A'\mathbf{x} \quad (4)$$

where $||\mathbf{x}||_1 = \sum_i |x_i|$. Thus, we can recover the full n-pixel image from much less than n values, which effectively provides compression, but also protection from packet loss.

Many algorithms were proposed to solve the optimization in (4) [13, 14, 10]. Digital Rain adapts a variant of the greedy StOMP algorithm [13], which has a low computational complexity. The intuition underlying StOMP is fairly simple. Since $\mathbf{x}$ is sparse, it has few non-zero elements. The indices of these non-zero elements are referred to as the support set I. If one can guess the support set, the system of equations simplifies to solving for $|I|$ unknowns, which are way fewer than all elements in $\mathbf{x}$. Such a system is likely to become overdetermined and hence solvable using least square fitting. Alg. 1 presents the details of StOMP. In order to guess the support set, StOMP uses a matched filter (A'^T). The purpose of the matched filter is to emphasize the non-zero indices not included in the current guess. The algorithm includes these indices in its guess for the next iteration. StOMP stops when the fit is sufficiently good (judged by the norm of the residual) or when it cannot add any new indices to the support set.

(b) Dealing with bit errors: Current compressed sensing does not deal with bit flips in the measurement vector $\mathbf{y}'$, which is the type of errors introduced by the wireless channel. The channel can flip any bit or sequence of bits causing a dramatic change in the corresponding element in $\mathbf{y}'$. These wrong measurements confuse StOMP and result in a wrong solution for the image signal $\mathbf{x}$ (see Figure 7).

1 Stagewise Orthogonal Matching Pursuit

$\mathbf{x} = \mathbf{0}, I = \phi$
repeat
(1) $\mathbf{r} = \mathbf{y}' - A'\mathbf{x}$
(2) $\mathbf{c} = A'^T\mathbf{r}$
(3) $J =$ largest few components of $\mathbf{c}$
(4) $I = I \cup J$
(5) $\mathbf{x} = \min_{\mathbf{x}} ||\mathbf{y}' - A'\mathbf{x}||_2$ s.t. $\forall_{i \notin I}\ x_i = 0$
until should stop

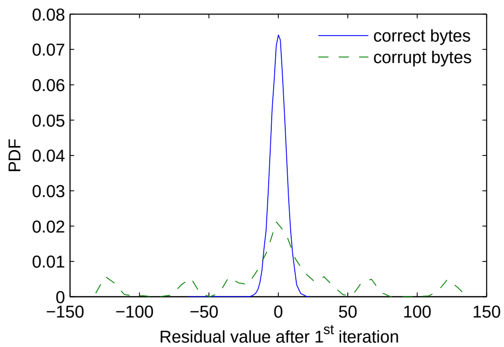

Figure 2: **Identifying Corrupt Measurements.** The residual distance of corrupt measurements is much more spread than that of correct measurements.

Digital Rain uses a heuristic to identify the erroneous elements in the measurement vector $\mathbf{y}'$ and remove the corresponding equations from the linear system in (3) in order to prevent them from biasing the solution. The heuristic works as follows. Each element in the received measurement vector, y'_i, is either correct or corrupt by bit flips. The correct elements are all linear projections of a sparse vector $\mathbf{x}$. Thus, they all live on a high-dimensional hyper-plane. The corrupt elements do not belong to that hyper-plane. At the beginning we do not know this hyper-plane. However, after each iteration of StOMP, we have a low dimensional estimate of $\mathbf{x}$ which lives in the space spanned by the current support set I. Thus, we can compute a projection of the measurement hyper-plane on a lower dimension space, i.e., the space spanned by the current support, I. This projection is $A'\mathbf{x}'$, where $\mathbf{x}'$ is the current estimate of $\mathbf{x}$ based on the current support set I. We guess that correct measurements should be close to this hyper-plane, whereas corrupt measurements are farther from this hyper-plane. Said differently, for a measurement value y_i, the residual distance $r_i = y_i A' - \mathbf{x}'$ is likely to be large if the measurement is corrupted.

Measurements from actual video data support our intuition. In particular, Figure 2 shows the residual distance after the first iteration of StOMP for both corrupt measurements and correct measurements. Clearly, the corrupt measurements are much more spread. Given this intuition we can design a simple algorithm to identify corrupt measurements and ignore them. After every iteration of StOMP, compute the residual distance $r_i = y_i A' - \mathbf{x}'$ for every measurement i. Compute the standard deviation in the residual distance and eliminate all measurements whose residual distance is more than a few standard deviations.

Note that the above approach can cause us to lose some correct measurements that happen to have large residual distance. This is however fine since no measurement is special and the loss will only cause a smooth degradation in the quality of the video. Also, we might admit measurements with errors. This is also not disastrous, since the erroneous measurements that we may admit are close to be fitted, and likely these are measurements with errors in the least significant bit and. Hence, they do not cause the solution of $\mathbf{x}$ to deviate by much.

(c) Dealing with high packet loss rates: Since an underdetermined system cannot be solved uniquely, the success of step (5) depends on having sufficiently more elements in $\mathbf{y}'$ than in I ($|I|$ is the number of free variables). Hence, if a receiver experiences massive packet losses, the number of received measurements is severely limited, and one should not attempt to recover more elements of $\mathbf{x}$ than algebra allows. In such a case, one would, ideally, retrieve only the most significant elements in $\mathbf{x}$ in order to minimize the distortion caused to the image.

By design, StOMP retrieves the components of $\mathbf{x}$ roughly in the order of decreasing magnitude. Thus, the remaining issue is to decide when to stop decoding. Recall, that for the sake of compression, JPEG/MPEG remove the least significant components of $\mathbf{x}$ by quantizing before Huffman coding. In Digital Rain, we can apply a similar technique *a posteriori*, on the decoded signal. After every iteration, we apply a hard non-uniform[1] threshold to $\mathbf{x}$ and clear the components of insufficient magnitude. We also treat them as false positives and remove them from I. StOMP naturally stops iterating when it can no longer find sufficient number of new components that would pass the threshold.

This non-uniform threshold is scaled depending on the expected quality of the image, which in turn is scaled to the number of received measurements. Hence we achieve natural rate-distortion at the decoder.

3.4 Integration with Network Coding

Prior work has shown that network coding can exploit opportunistic receptions to improve wireless throughput in multi-hop wireless networks [9, 26]. Network coding, however, is all-or-nothing; If a batch of n packets are combined using network coding, the receiver has to receive at least n packets in order to decode; otherwise the batch is lost [9]. This is in direct conflict with video streaming applications, where ensuring full reliability is unnecessary, and can be problematic if it leads to increased delay.

Digital Rain harnesses the benefits of network coding while avoiding its all-or-nothing behavior. To do so, Digi-

[1] The threshold depends on the DCT frequency encoded in the component in a way matching the non-uniform quantization tables of JPEG/MPEG.

tal Rain exploits the following two features:

- Video and audio signals are described best in the Real domain, and thus their network coding should be in the Real field rather than a finite field as it is typically the case [31, 27, 4, 9]. Specifically, Digital Rain pushes network coding to the application layer where it is performed on linear projections of a frame's DCT components, which are Real numbers. Operating over the Real field allows Digital Rain to seek approximate solutions and hence avoid all-or-nothing decoding. Say that the luminance of a particular pixel is 2.41. Approximating this real number with a close real number, like 2.50, is a meaningful operation, with minor impact on the image itself. In contrast, flipping a bit in codewords over a finite field, e.g., transforming "10101000" to "10101001" may lead to completely irrelevant values.
- In contrast to MPEG and JPEG, which use non-linear codes (e.g., Huffman coding), Digital Rain uses linear codes (see Equation eq1). As a result, Digital Rain is amenable to integration with network coding, which also relies on linear codes [31, 27, 4]. Say the source code is $\mathbf{y} = A\mathbf{x}$; the network between the sender and a receiver in the multicast group, creates linear combinations of the source coded signal, i.e., the network transforms $\mathbf{y}$ to $\mathbf{z} = B\mathbf{y}$, where B is a matrix that refers to the linear combinations applied using network coding. (In the above example, the matrix B refers to the random linear coefficients, the b_i's, used by Alice to produce coded packets.) The receiver needs to decode $\mathbf{z} = (BA)\mathbf{x}$, which itself is a linear code, and hence can be decoded using the same decoder in §3.3 after replacing the matrix A with the matrix BA.

A Digital Rain exploits network coding to multicast video over multi-hop networks. In particular, when some of the receivers are too distant such that they cannot be satisfied directly by the source, Digital Rain can use a multi-hop network, where intermediate nodes forward the video traffic to more distant receivers. In this case, the source encodes the video stream using the encoder algorithm described earlier and broadcasts the resulting packets. The intermediate nodes in the network create linear combinations of the packets they receive and broadcast them. Both nearby and distant nodes use any independent coded packets they receive to increase their video quality.

Coding alone is not sufficient. We need to control how much coded traffic the intermediate nodes forward, otherwise they might consume the whole capacity of the medium, and dramatically reduce the throughput of the wireless source. Digital Rain limits the forwarding traffic as follows. First, Digital Rain organizes the nodes in a shortest path tree rooted at the source and computed using ETX as a distance metric [11]. Each parent in the tree is responsible for satisfying the video requirements of its children. When a child's video requirements are satisfied, it informs its parent. The parent stops forwarding packets from a particular frame once all of its children have received enough packets from that frame to satisfy their minimum quality requirements.

3.5 Complexity

Digital Rain has the same complexity as a typical MPEG or JPEG codec. Similar to these standards, Digital Rain takes a Discrete Cosine Transform (DCT) over the pixels in the image [20, 46]. Digital Rain, however, replaces the Huffman encoder/decoder used in these standards with novel encoder/decoder algorithms based on greedy compressed sensing (the StOMP [13] algorithm) and network coding. These algorithms have a low complexity of $O(n \log n)$, where n is the number of pixels in a frame. This does not increase the complexity in comparison with MPEG or JPEG since these standards already use a DCT transform, which is $O(n \log n)$.

4 EXPERIMENTAL RESULTS

This section uses results from a 18-node wireless testbed to study the performance of our approach. Our testbed is shown in Figure 3. Each node in the testbed is a PC equipped with a NETGEAR WAG311 wireless card attached to an omni-directional antenna. They operate in the 802.11b/g monitor mode, with RTS/CTS disabled. Nodes in the testbed run Linux, the Click toolkit [28] and the Roofnet software package [3].

4.1 Compared Schemes

We consider a video source that generates a frames once every 33 ms (i.e., 30 frames/s). Thus, the source has 33 ms of air-time for every frame which, as described below, can be used in a number of ways to server the receivers. The objective of the source is to maximize the average video quality at all the receivers.

- *Multiple Unicasts:* The source uses the air-time allocated to each frame to unicast it to the maximum number of receivers. Specifically, the source first picks the receiver with the best channel and transmits the frame at the rate supported by the receiver's channel. Then the source picks the receivers with the next best channel and unicasts the frame. The source continues this process for 33 ms, at which time it moves on to the next frame and repeats the same process.
- *Single Layer Multicast:* The source broadcasts a single MPEG video stream on the wireless medium. However, in order to ensure that most of the receivers can decode the frames correctly, we use Reed-Solomon codes to protect every MPEG frame from bit errors and packet loss. The redundancy of the Reed-Solomon code is picked to ensure that a video frame after Reed-Solomon encoding fills up the 33 ms air time allocated to it.
- *Two-Layer Multicast:* This approach is a hybrid of the above two approaches. Conceptually, the receivers are divided into two categories: receivers with low channel quality and receivers with high channel quality. In or-

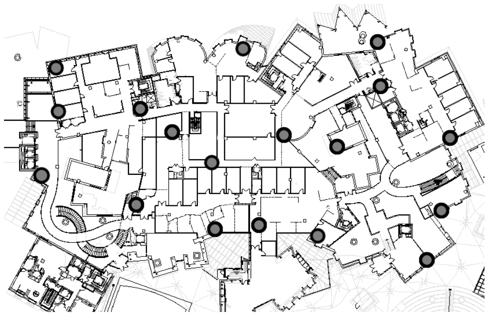

Figure 3: **Testbed Topology.** The dots refer to the wireless nodes.

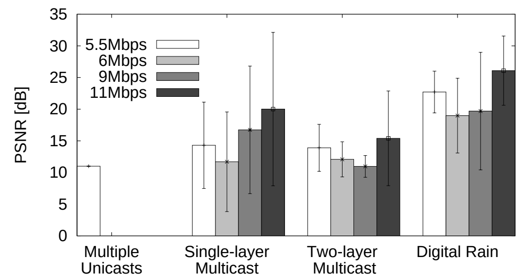

Figure 4: **Comparison of the average PSNR across all receivers.**

der to provide different video quality to these two categories, the source divides the video stream into two layers: a base layer and an enhancement layer. The base layer MPEG-encodes a video sequence at a low resolution. The residue between the original video and the reconstructed base layer is then encoded as the enhancement layer. The base layer can be decoded in the same way as typical MPEG, but to decode the combined base and enhancement layers, both layers must be received. The base layer is protected with twice as much Reed Solomon code as the enhancement layer. The amount of redundancy is picked to ensure that the total frame size (i.e., both layers with their corresponding error correcting codes) fills up the 33 ms air time.

- *Digital Rain:* The source uses the encoding algorithm described in §3.2, on every frame and transmits packets for a period of 33 ms for every frame.

4.2 Performance Metric

The Peak Signal-to-Noise Ratio (PSNR) is a standard measure of video/image quality [36]. It is a function of the mean squared error (MSE) between the decoded video frame and the original frame:

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

where L represents the number of bits used to encode the luminance signal, typically 8 bits, and the MSE between an $n \times m$ decoded frame I and its original version K is:

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

The PSNR is expressed in decibels (dB). Typical PSNR values range between 20 and 40. Improvements in PSNR of magnitude larger than 0.5 dB would be visible [36].

4.3 Setup

We show results for the known Salesman video sequence, where each frame is 178 × 144 pixels, and the video is encoded at 30 frames/s (QCIF format). Our prototype implementation produces only I-frames.[2] The video clip is looped to obtain 5-min multicast sessions. Video quality is recorded at the different receivers, measured in terms of average peak signal-to-noise ratio (PSNR).

Each experiment involves picking a random source from the nodes in the testbed to multicast the video to all other nodes. Receivers log all packets including those that do not satisfy the CRC test. We distinguish between the first compared scheme, which uses multiple unicasts, and the others, which use 802.11 broadcast. For the multiple unicast scheme, we pick the best rate for each receiver and allow the sender to retransmit corrupted packets up to four times. For the schemes that use 802.11 broadcast, we try all broadcast rates and pick the one that works the best, i.e., it maximizes the average PSNR at all receivers.

Furthermore, we ensure that all compared schemes experience exactly the same packet loss and bit error patterns, and hence differences in performance are due to the characteristics of the compared schemes, rather than changes in interference or cross traffic. Specifically, in each experiment, we multicast a long video file and collect the traces of the received video packets at all receivers. We compare the transmitted stream with the received trace at every node. Since we know the exact packets that were transmitted, we can easily identify which packets were lost and which bits were incorrectly received, at each node. Thus, we map each experiment into a set of per-receiver bit and packet masks. When applied to a file, these masks produce the same patterns of packet loss and bit errors that were observed at all receivers in the corresponding experiment. Thus, instead of repeating the transmission for each compared scheme, incurring different interference and error patterns, in each experiment we multicast one file at various 802.11 rates, and apply the corresponding masks to all compared schemes.

4.4 Comparing the Performance of Various Schemes

We evaluate Digital Rain and compare it to three alternatives: multiple unicasts, single layer multicast, and two-layer multicast. For the multiple unicasts scheme, we allow the source to unicast the video to each receiver at the highest rate supported by the receiver as described in §4.1. For the other schemes we try the following four bit rates:

[2] Adding P frames and motion compensation is straightforward and we are in the process of doing it.

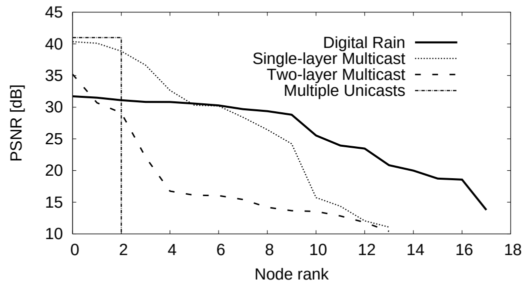

Figure 5: **Digital Rain provides a good PSNR to more receivers.**

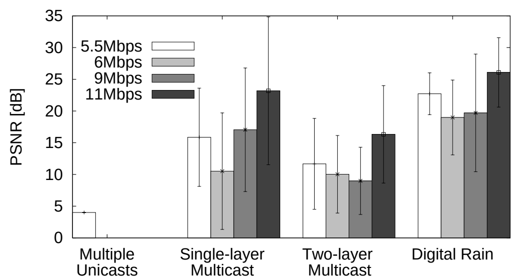

Figure 6: **Comparison of the average PSNR across all receivers, allowing all four schemes to exploit packets with bit errors.**

5.5 Mb/s, 9Mb/s, 11 Mb/s, and 12 Mb/s.

Figure 4 plots the average PSNR across all receivers.[3] The figure shows that the multiple unicasts scheme performs badly because it can support only a very limited number of receivers and delivers no video to the others. Interestingly the single layer multicast performs better than the two-layer multicast. As we argued in §1, layered video, while works for wired networks, is unsuitable for the wireless environments because a receiver cannot pick which packets to receive, and hence receives equally from both layers. However, receivers with bad channel quality do not obtain enough packets to decode the enhancement layer and waste any packet they received from that layer. As for Digital Rain, it outperforms the other schemes at every bit rate. It improves the average PSNR by 22 dB over multiple unicasts, 9 dB over two-layer multicast, and 3 dB over single layer multicast.

Figure 5 shows the PSNR of the individual receivers in the testbed ordered according to their decreased PSNR. The results are for a bit rate of 11Mb/s, except for the multiple unicasts scheme, which uses different rates to different receivers. The figure shows that the multiple unicasts scheme has a very low performance because it supports only a couple of receivers. The single layer multicast and two-layer multicast support more receivers but do not degrade smoothly. Finally Digital Rain presents a good PSNR to most receivers and degrades smoothly as the channel quality worsens.

[3]The minimum PSNR is 10 since a fully grey picture produces around 10 dB PSNR.

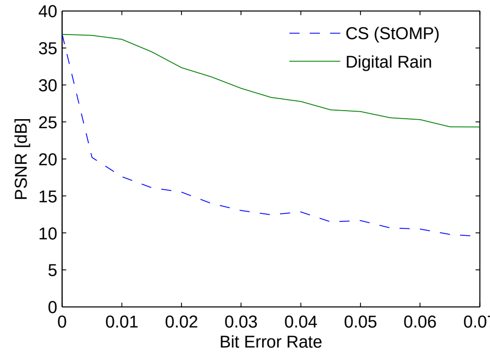

Figure 7: **Efficacy of the Decoder in Dealing with Bit Errors** Digital Rain's decoder can recover from bit errors while traditional compressed sensing (CS) cannot.

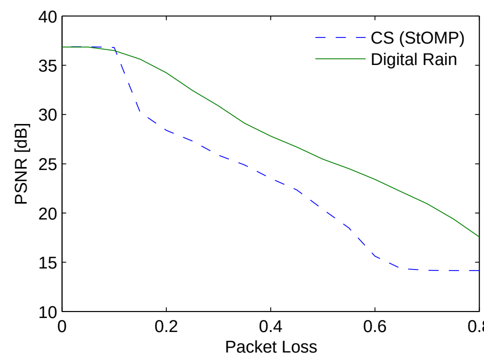

Figure 8: **Dealing with High Packet Loss Rates.** Digital Rain's decision to scale the desirable reconstruction quality with the fraction of lost packets allows for a slower degradation of PSNR as a function of lost packets, and hence improves video quality. Note that the data used here has only packet losses and no bit errors.

Next, we would like to check whether Digital Rain maintains its performance lead if we allow the other schemes to exploit packets that do not pass the 802.11 checksum test. Figure 6 repeats the experiment in Figure 4 but while allowing all four schemes to exploit packets with bit errors. The figure shows that the multiple unicasts scheme degrades drastically because MPEG encoding is highly vulnerable to bit errors and this scheme does not use error correcting codes. The figure also shows that the two-layer multicast and the single-layer multicast both show improved PSNR. These schemes can benefit from the correct bits in a corrupt packet because of their use of Reed-Solomon codes, which can correct both bit errors and packet erasures. However, Digital Rain still performs significantly better than these schemes. Hence, Digital Rain's good performance stems from both its ability to exploit packets with bit errors and its encoding and decoding algorithms.

4.5 Decoder Benchmarks

Digital Rain is motivated by recent advances in compressed sensing [15, 8, 7] but adopts its own decoder that rejects erroneous video measurements and adapts the reconstruction quality to the number of received packets. We examine the impact of these new features and whether Digital Rain could have performed as good by simply adopting compressed sensing. Our Digital Rain implementation is based on StOMP, a known algorithm for compressed sensing. Thus, we compare the performance of Digital Rain with the unmodified StOMP.

Figure 7 shows that Digital Rain's approach to rejecting erroneous measurements is necessary. The figure plots the video quality as a function of the bit error rate (BER), for StOMP and Digital Rain. Both algorithms are given the same number of video packets that is sufficient to fully recover the signal in the absence of bit errors. The figure shows that traditional compressed sensing, i.e., StOMP, cannot deal with bit errors. Flipping bits in the video can lead to values significantly different from the original measurements and hence confuses the StOMP decoder. In contrast, Digital Rain's decoder identifies most of the erroneous bytes and ignores them, and hence is more resilient to bit errors. Indeed, the difference in performance is as high as 15 dB, which means dramatic improvement in video quality.

Also, as described in §3.3(c), Digital Rain's decoder differs from traditional compressed sensing in that it decides the desirable reconstruction quality based on the percentage of lost packets. A receiver that receives only a few packets from a particular frame should not try to shoot for a high quality frame, but should rather focus on retrieving the largest DCT components in the vector $\mathbf{x}$, which usually capture most of the structure in a frame. Here, we would like to examine the value of Digital Rain's decision to scale the desirable reconstruction quality with the number of received video packets.

Since we want to focus just on packet losses, we turn on the CRC check which drops all packets with bit errors in them. Fig. 8 plots the PSNR as a function of the percentage of lost packets. The figure shows that when the percentage of lost packets is small ($< 10\%$), scaling the desired reconstruction quality has little effect. However, for receivers that lose a significant fraction of the transmitted packets, Digital Rain's decision to scale the desirable reconstruction quality results in about 4-7 dB increase in PSNR, which is a significant improvement in video quality.

4.6 Benefits of Network Coding

Last, we compare three options for routing Digital Rain's traffic in a multi-hop wireless network. The first option is typical multicast routing, which delivers traffic along a shortest-path tree that minimizes the ETX metric between the source and individual receivers [11]. The second is opportunistic multicast, which uses the same distribution tree as before, however it allows the nodes to exploit any off-tree opportunistic receptions they may hear from nodes other than their parent. The third is network coding as described in §3.4, where intermediate nodes codes the packets before forwarding them to distant nodes, and each intermediate node has a set of children to satisfy. Note that in all three schemes, a parent node does not unicast packets to each child separately; a parent broadcasts packets to all of its children. We run our experiments over the topology in Fig. 9, where we randomly pick 7 nodes from the testbed and organize them in a tree according to the ETX metric. We repeat the experiment 10 times, and compute the average PSNR at each of the 6 receivers.

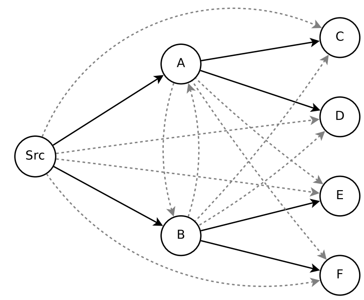

Figure 9: **Topology used to evaluate Digital Rain in a multi-hop scenario.**

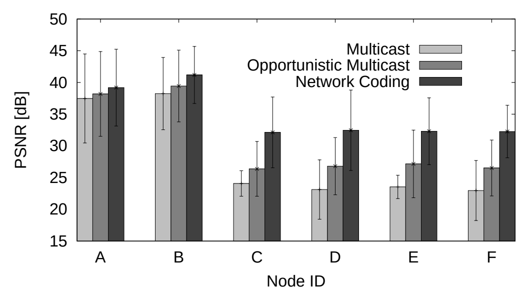

Figure 10: **Digital Rain's performance in a multi-hop network.** Digital Rain leverages the benefits of network coding to achieve high PSNR, but it does not require that all receivers obtain all packets in a batch/frame.

Fig. 10 plots the average PSNR at all receivers in Fig. 9, for the three compared schemes. As expected, exploiting opportunistic receptions improves the overall quality compared to traditional multicast routing. Further, network coding improves the quality over pure opportunistic routing. This is because uncoded packets create duplicate receptions at many nodes who have already heard that packet. In contrast, a randomly coded packet is less likely to be a duplicate and hence benefits more receivers. Receivers close to the source, however, experience good video quality regardless of the routing scheme. In contrast, the benefits of network coding are particularly high at distant receivers where there is a significant opportunity for improvement. For those receivers, network coding brings an average improvement of 5 dB over opportunistic routing and 8 dB over traditional multicast routing.

Note that traditional network coding does not apply to this scenario because the distant receivers which do not receive all packets in a frame, would not be able to decode any of the coded packets, and hence there is no point using network coding.

5 CONCLUSION

Traditional video encoding and decoding algorithms are oblivious of the quality of the communications channel. Additional error correcting codes are independently applied over the encoded video to account for channel errors and packet loss. This separation limits one's ability to dynamically tradeoff the video quality for the channel quality.

In this paper, we question whether this separation is always desirable. In particular, we present Digital Rain a new approach to wireless video multicast, where the sender broadcasts one video stream but each receiver perceives the video at a quality that matches the quality of her wireless channel. Digital Rain accomplishes this by building on the theory of compressed sensing to develop robust video encoding and decoding algorithms that degrade smoothly with bit errors and packet loss. In addition, Digital Rain naturally works with other technologies like network coding, which can further improve the video quality in a large network.

References

[1] Sonos digital music system. http://www.sonos.com.

[2] P2P Video Multicast for Wireless Mobile Clients, 2006.

[3] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11b mesh network. In *SIGCOMM*, 2004.

[4] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung. Network Information Flow. In *IEEE Trans. on Information Theory*, Jul 2000.

[5] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak. Compressive wireless sensing. In *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*, New York, NY, USA, 2006. ACM.

[6] R. Baranuik and P. Steeghs. Compressed radar. *IEEE Radar Conference*, 2007.

[7] E. Candes and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies?, Apr 2006.

[8] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509, 2006.

[9] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In *Proc. of ACM SIGCOMM 2007, Kyoto, Japan*.

[10] S. Chen and D. Donoho. Basis pursuit. *Signals, Systems and Computers, 1994. 1994 Conference Record of the Twenty-Eighth Asilomar Conference on*, 1:41–44 vol.1, 31 Oct-2 Nov 1994.

[11] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *MOBICOM*, 2003.

[12] T. Do, K. Hua, and M. Tantaoui. P2vod: providing fault tolerant video-on-demand streaming in peer-to-peer environment. *Communications, 2004 IEEE International Conference on*, 3:1467–1472 Vol.3, 20-24 June 2004.

[13] D. Donoho, I. Drori, Y. Tsaig, and J. Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. *Preprint*, 2007.

[14] D. Donoho, Y. Tsaig, I. Drori, and J.-C. Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit, Mar 2006.

[15] D. L. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.

[16] M. Duarte, S. Sarvotham, D. Baron, M. Wakin, and R. Baraniuk. Distributed compressed sensing of jointly sparse signals. *Signals, Systems and Computers, 2005. Conference Record of the Thirty-Ninth Asilomar Conference on*, pages 1537–1541, October 28 - November 1, 2005.

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

[18] A. Fletcher, S. Rangan, and V. Goyal. On the rate-distortion performance of compressed sensing. *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, 3:III–885–III–888, 15-20 April 2007.

[19] 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.

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

[21] L. Gan. Block compressed sensing of natural images. *Digital Signal Processing, 2007 15th International Conference on*, pages 403–406, 1-4 July 2007.

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

[23] L. Hanzo, P. Cherrimana, and J. Streit. *Wireless*

Video Communications: Second to Third Generation and Beyond. John Wiley, 2001.

[24] K. Jamieson and H. Balakrishnan. Ppr: Partial packet recovery for wireless networks. In *Proc. of ACM SIGCOMM 2007, Kyoto, Japan*.

[25] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein. Growth Codes: Maximizing Sensor Network Data Persistence. In *SIGCOMM*, 2006.

[26] S. Katti and D. Katabi. MIXIT: The Network Meets the Wireless Channel. In *Proc. of ACM Hotnets 2007, Atlanta*.

[27] R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM Trans. on Networking*, 2003.

[28] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The click modular router. *ACM Trans. on Computer Systems*, Aug 2000.

[29] V. Kompella, J. Pasquale, and G. Polyzos. Multicast routing for multimedia communication. *Networking, IEEE/ACM Transactions on*, 1(3):286–292, Jun 1993.

[30] R. Kumar, A. Paul, and U. Ramachandran. Fountain broadcast for wireless networks.

[31] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. *IEEE Trans. on Information Theory*, Feb 2003.

[32] T. T. Lin and F. J. Herrmann. Compressed wavefield extrapolation. *Geophysics*, 2008.

[33] D. S. Lun, M. Médard, and R. Koetter. Efficient operation of wireless packet networks using network coding. In *IWCT*, 2005.

[34] O. M. R. B. MA Sheikh, S Sarvotham. Dna array decoding from nonlinear measurements by belief propagation. *Statistical Signal Processing*, 2007.

[35] MADWiFi: Multiband Atheros Driver for WiFi. http://madwifi.org.

[36] 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 lossscenarios 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.

[37] N. Maxemchuk. Video distribution on multicast networks. *Selected Areas in Communications, IEEE Journal on*, 15(3):357–372, Apr 1997.

[38] A. Oppenheim. *Digital Signal Processing*. Prentice-Hall, Englewood, NJ, USA, 1975.

[39] J. S. Park, M. Gerla, D. S. Lun, Y. Yi, and M. Médard. Codecast: A network-coding based ad hoc multicast protocol. *IEEE Wireless Comm. Magazine*, 2006.

[40] P. W. Rabbani, M.; Jones. *Digital Image Compression Techniques*. Bellingham (WA):SPIE Optical Engineering Press, 1991.

[41] 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.

[42] E. Setton, J. Noh, and B. Girod. Rate-distortion optimized video peer-to-peer multicast streaming. In *P2PMMS'05: Proceedings of the ACM workshop on Advances in peer-to-peer multimedia streaming*, pages 39–48, New York, NY, USA, 2005. ACM.

[43] T. Stockhammer. Is fine-granular scalable video coding beneficial for wireless video applications? 2000.

[44] M.-T. Sun and A. R. Reibman. *Compressed Video Over Networks*. Marcel Dekker, 2000.

[45] M. Tagliasacchi, A. Majumdar, K. Ramchandran, and S. Tubaro. Robust wireless video multicast based on a distributed source coding approach. *Signal Process.*, 86(11):3196–3211, 2006.

[46] G. K. Wallace. The jpeg still picture compression standard. *Commun. ACM*, 34(4):30–44, 1991.

[47] S. B. Wicker. *Reed-Solomon Codes and Their Applications*. IEEE Press, Piscataway, NJ, USA, 1994.

[48] J. Widmer and J.-Y. L. Boudec. Network Coding for Efficient Communication in Extreme Networks. In *SIGCOMM WDTN*, 2005.

[49] G. Woo, P. Kheradpour, and D. Katabi. Beyond the bits: Cooperative packet recovery using phy information. In *Proc. of ACM MobiCom 2007, Montreal, Canada*.

[50] 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.

[51] D. Xu, M. Hefeeda, S. Hambrusch, and B. Bhargava. On peer-to-peer media streaming. *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on*, pages 363–371, 2002.