Optimized Layered Integrated Video Encoding

Sangki Yun, Daehyeok Kim, Xiaofan Lu, Lili Qiu
The University of Texas at Austin

***Abstract*—Wireless video traffic has grown at an unprecedented rate and put significant burden on wireless networks. Multicast can significantly reduce traffic by sending a single video to multiple receivers simultaneously. On the other hand, wireless receivers are heterogeneous due to both channel and antenna heterogeneity, the latter of which is rapidly increasing with the emergence of 802.11n and 802.11ac. In this paper, we develop optimized layered integrated video encoding (LIVE) to guarantee reasonable performance to weaker receivers (with worse channel and/or fewer antennas) and allow stronger receivers to enjoy better quality. Our approach has three distinct features: (i) It uses a novel *layered* coding to naturally accommodate the heterogeneity of different video receivers; (ii) It uses an *optimization* framework to optimize the amount of time used for transmission and the amount of information to transmit at each layer under the current channel condition; and (iii) It uses an *integrated modulation*, where most video data are transmitted using soft modulation to enjoy efficiency and resilience while the most important video data are transmitted using a combination of soft modulation and conventional hard modulation to further enhance their reliability. To our knowledge, this is the *first* approach that handles MIMO antenna heterogeneity in wireless video multicast. We demonstrate its effectiveness through extensive Matlab simulation and USRP testbed experiments.**

I. Introduction

Motivation: Wireless video traffic grows at an unprecedented rate and puts significant stress on wireless networks. Severe wireless network congestion may arise as many users try to watch a popular video in the same area. Multicast is an effective approach to reduce congestion by sending a single video stream to all of them. However, wireless receivers are heterogeneous due to inherent heterogeneity in their channel. Receiver heterogeneity is increasing further due to antenna heterogeneity. For example, with the emergence of 802.11n and 802.11ac, the numbers of antennas on the receivers can vary from 1 to 8. Multicasting to a group of heterogeneous receivers is challenging because we should ensure not only every receiver gets video with reasonable quality but also the receivers with better channel or more antennas enjoy better performance instead of being bottlenecked by the weakest receiver. Simply multicasting at one rate to everyone has serious performance issues: either the receivers with better channel or more antennas have to suffer the same poor quality as the weaker receivers or the receiver with better channel or more resources can enjoy good performance while the performance of weaker receivers can be arbitrarily bad.

Our approach: In this paper, we study video multicast in a wireless network (*e.g.*, from an AP to clients). We propose a novel method called Layered Integrated Video Encoding (LIVE) to enable efficient video dissemination while naturally accommodating the heterogeneity of different video receivers with respect to their channel conditions and numbers of antennas. To our knowledge, this is the *first* approach that handles antenna heterogeneity in wireless video multicast.

Our method has three defining characteristics: (i) LIVE is *layered*. Video content is divided into multiple layers. Receivers are sorted in an increasing order of their bandwidth budget. The i-th receiver group receives all the layers below or equal to i, and opportunistically receives partial information of the higher layers. The layered coding provides performance guarantees to all the receivers by ensuring that each receiver gets some video information reliably and the amount of such reliable information is determined by its bandwidth budget. (ii) LIVE is *optimized*. We develop an optimization framework to determine how much information to transmit at each layer based on receivers' bandwidth budget and channel condition. (iii) LIVE *integrates* both soft and hard modulation, where soft and hard modulation differ in that an analog signal is represented as a real number in the soft modulation, and as a discrete constellation point in the hard modulation. In LIVE, most data is transmitted using soft modulation due to its high efficiency (*i.e.*, one signal can represent two real numbers) and resilience to noise (*i.e.*, noise introduces error instead of complete corruption in a group of pictures). Meanwhile, small amount of the most important video data is transmitted using a combination of soft and hard modulation to further enhance performance. Integrated modulation benefits both video multicast and unicast. Since our approach focuses on video coding, it can be applied to different networks, such as WiFi and cellular networks, by modifying their modulation.

We implement LIVE in both Matlab simulation and a USRP testbed, and compare its performance with MPEG4 and SoftCast [7], one of the latest wireless video multicast approaches. Our results show that LIVE achieves significant performance improvement in both unicast and multicast contexts. Specifically, in unicast LIVE out-performs MPEG4 by 4.1–6.1 dB and SoftCast by 1.9–3.5 dB in terms of average peak signal-to-noise ratio (PSNR) (*i.e.*, a standard video metric) due to integrated coding. In multicast, the improvement increases to 4.6–9.3 dB over MPEG4 and 2.2–4.7 dB over SoftCast, which comes from further optimizing layered coding for multicast receivers. The benefit further increases with the antenna heterogeneity at the clients. Note that 1 dB difference in PSNR is already quite visible, and 3 dB difference indicates that the video quality is doubled. These results thus clearly demonstrate the effectiveness of our approach.

Contributions: Our main contributions consist of (i) a novel *layered* coding to cope with the channel and antenna heterogeneity at different video receivers; (ii) an *optimization* framework that determines the amount of time to spend and the amount of information to send at each layer based on receivers' bandwidth budget and channel conditions; (iii) an *integrated modulation* to achieve both efficiency and resilience; and (iv) Extensive simulation and testbed evaluation to demonstrate their effectiveness.

II. Background and Related Work

Video coding: There has been considerable work on video coding for multicast. Among them, a series of works focus on layered video coding, which sends the base layer to everyone

978-1-4799-8381-0/15/$31.00 ©2015 IEEE

and enhancement layers to the receivers with better channel conditions (*e.g.*, [15], [2], [4], [11], [12]). Scalable Video Coding (SVC) is one of the most widely used layered codings. However, it has several limitations when applied to wireless video multicast: (i) It is vulnerable to channel noise since errors in a few bits can lead to corrupting an entire group of pictures (GoP) (*i.e.*, a set of successive video frames) [7]. In order to ensure reliability, significant redundancy has to be added, which significantly reduces efficiency. (ii) SVC cannot effectively support heterogeneous MIMO antennas since higher layers are transmitted using more spatial streams and cannot be decoded by the receivers with fewer antennas. In comparison, our approach allows receivers with heterogeneous antennas to all derive useful information from all the layers and the amount of information they derive increases with their numbers of antennas. (iii) SVC is not optimized. It is not clear how much resources SVC should spend in sending different layers. The optimization is challenging because different layers may be sent with different MIMO configurations and benefit different sets of receivers. Multiple Description Coding (MDC) [5] is another well-known video coding technique. Unlike layered coding, it does not require strict ordering between different descriptions and allows each description to be decoded by itself. The more descriptions a node receives, the better quality it gets. However, this comes at the cost of significant coding overhead, so it is rarely used in practice.

MPEG4 is the most popular video coding today. It codes the pixel values in a group of pictures (GoP) by applying Discrete Cosine Transform (DCT) [16]. Then it quantizes and compresses the DCT coefficients. As pointed out in [7], [1], its compression works well for a reliable channel but is fragile in noisy wireless links since loss/corruption of a few bits can lead to decoding errors in an entire GoP.

Joint channel and video coding: There are significant works on joint channel and video coding [7], [1], [10], [18], [14], [13]. The work closest to ours is SoftCast [7]. It treats pixel values in a GoP as a 3-dimensional matrix, and applies 3-dimensional DCT transform of the pixel value matrix. After DCT transform, the DCT coefficients form another 3-dimensional matrix, where most entries are zeros or close to zeros, and can be ignored without compromising the video quality. Only large DCT coefficients need to be transmitted, and they are usually clustered. SoftCast groups the DCT coefficients into chunks based on their positions in the matrix. To minimize reconstruction error, SoftCast sorts the chunks in a decreasing order of the energy of DCT chunks and transmits as many chunks as possible to fill the bandwidth.

Instead of transmitting raw entries in the chunks, SoftCast transmits linear combinations of these entries. More specifically, let X denote the DCT components in a GoP, where each row is a chunk. A SoftCast sender transmits $Y = CX$, where C is an encoding matrix. C can be Hadamard matrix as used in SoftCast or any random matrix, which gives similar performance in our evaluation. After going through the wireless channel, the signals arriving at the receiver becomes $Y = HCX$ where H is channel coefficients. Since Y, H, and C are all known, the receiver decodes X using linear least square estimator (LLSE) [9]. Our approach improves SoftCast in that it is *layered*, *optimized*, and uses integrated coding.

FlexCast [1] has the same processing as MPEG4 except that at the last stage it replaces traditional video compression with its own rateless video codec. The codec divides DCT coefficients into distortion groups based on their contribution to the reconstruction error, allocates bits to distortion groups based on their importance, and encodes them using a rateless code, such as Raptor code. Since it requires rate selection, FlexCast is not applicable to multicast when receivers have different data rates.

ParCast [10] enhances video transmission in MIMO-OFDM channels by applying SVD precoding to improve the MIMO link quality and mapping important video components to more reliable OFDM subcarriers. As FlexCast, ParCast also focuses on video unicast. Since we mainly focus on video multicast, we do not compare with FlexCast or ParCast. Moreover, none of the existing works address antenna heterogeneity, which is our focus.

III. Layered Coding

Motivation: Suppose a source broadcasts a video stream to multiple receivers. Different receivers may have different numbers of antennas and/or experience different channel conditions. One approach is to unicast a video stream to each receiver separately. This significantly reduces the video quality each receiver receives since each transmission can only benefit one client. Multicast is attractive since one transmission can potentially benefit multiple clients. But how to multicast to heterogeneous clients poses a significant challenge. Multicasting at the weakest receiver's rate significantly degrades the video quality that the stronger receivers could have received, while multicasting at the strongest receiver's rate can make the performance of the weaker receivers arbitrarily bad. Our goal is to let receivers with better channel condition and/or more antennas enjoy higher video quality and let receivers with weaker channel and/or fewer antennas still receive reasonable video quality while leveraging multicast and avoiding sending redundant information.

Soft coding: The pixel values in a GoP is a 3-dimensional matrix. As SoftCast [7], we let a video source code the pixel values in a GoP using 3-dimensional DCT. After DCT transform, the DCT coefficients form another 3-dimensional matrix. The DCT coefficients are grouped into chunks based on their positions in the matrix and the chunks are sorted and transmitted in the order of their energy. A bitmap (compressed using run-length encoding) is used to inform the receivers of which chunks are transmitted. Instead of transmitting the raw DCT coefficients, the sender transmits linear combinations of DCT coefficients. The receivers then reconstruct the DCT coefficients based on their received signals, which have linear relationships with DCT coefficients.

The data is transmitted using soft modulation as in [7], [10]. Specifically, an analog signal is a complex number, including two numbers: I-value (real) and Q-value (imaginary). These two numbers each corresponds to a linear combination result of DCT coefficients (*i.e.*, the magnitude of I or Q, denoted as Y, follows $Y = CX$, where X is DCT coefficients and C is the linear coefficients). The main benefits of soft coding over hard coding include (i) efficiency: one signal conveys two real numbers whereas conventional hard coding requires multiple signals to transmit one real number, (ii) resilience:

it gracefully degrades with transmission errors since channel noise introduces errors to Y instead of decoding failures, and (iii) supporting heterogeneous MIMO.

To understand (iii), we observe that in hard coding when a source uses spatial multiplex to transmit two streams, the receiver with two or more antennas can correctly decode the streams. However, the receiver with one antenna cannot decode anything since it receives one signal, which is a function of two unknown transmitted signals, and does not have sufficient information to decode to digital symbols. In comparison, soft coding does not require receivers to decode immediately, but instead allows it to extract useful constraints using such receptions and decode after accumulating all the related constraints. In this case, it can extract a constraint

$$h_{11}y_1 + h_{21}y_2 = R,$$

where h_{11} and h_{21} are the channel coefficients from the first and second transmitter antennas to the receiver antenna, respectively, and y's are the transmission signals on these two antennas, and R is the received signal. A channel coefficient is a complex number, whose magnitude represents the channel attenuation and angle represents a phase shift. Even if the receiver with one antenna cannot decode y_1 and y_2 immediately, it still gets one constraint involving them and can use them along with the other constraints to infer y. If the receiver does not get enough constraints (*i.e.*, fewer than the number of unknowns), it can still make inference (*e.g.*, using LLSE) but incurs inference error. The more constraints a receiver gets, the lower the inference error.

Layered coding: When there are multiple receivers in a multicast group, how should a transmitter send DCT coefficients? A simple approach, as adopted by SoftCast [7], [10], is to select a common set of coefficients to transmit to everyone and the transmissions are linear combination results of these selected coefficients. The receivers that get more constraints can more accurately infer the coefficients, and the receivers that get fewer constraints incur higher inference errors. However, when the numbers of constraints each receiver gets is rather different (which is common under heterogeneous link loss rates or heterogeneous numbers of antennas at the receivers), SoftCast cannot satisfy both strong and weak receivers at the same time.

Consider a simple example where we have two receivers: receiver 1 gets only half the constraints as receiver 2 (which happens when receiver 1 has one antenna and receiver 2 has two antennas, or receiver 1 has 50% losses while receiver 2 does not have losses). SoftCast can either transmit at the rate of the weaker receiver but this will unnecessarily slow down the stronger receiver, or transmit at the rate of the stronger receiver but this will cause trouble for the weaker receiver. To see the latter, suppose the SoftCast sender determines its compression ratio based on the stronger receiver (receiver 2), and decides to broadcast the first 1000 DCT chunks to both receivers. However, this compression ratio does not work well for receiver 1 because it only receives half of the constraints (*i.e.*, only half Y). While it can still apply LLSE to reconstruct X based on incomplete constraints, its estimation error can be arbitrarily large since there are an infinite number of solutions to satisfy the constraints and the LLSE result is one of many possible solutions. In fact, if we were to target only receiver 1, we would broadcast 500 DCT chunks so that the receiver 1 gets 500 constraints involving 500 unknowns and the linear system is full ranked and can be accurately solved.

Throughout the paper, for ease of discussion, when a sender transmits K spatial streams, we call it sends K transmissions. In the above example, a better approach to multicasting to the two receivers is to first broadcast N_1/p_1 transmissions involving the first N_1 chunks, where p_1 is the fraction of the transmissions received by receiver 1. In this way, both receivers (including the weaker receiver 1) still receives N_1 linearly independent constraints to accurately decode the top N_1 chunks. Then the sender broadcasts N_2/p_2 transmissions involving the next N_2 chunks so that receiver 2 can accurately decode these chunks while receiver 1 decodes the next N_2 chunks with some errors due to an insufficient number of constraints. We select N_1 and N_2 such that they satisfy both receivers' bandwidth budget while optimizing the overall video quality across them.

In general, we use layered video coding where each layer targets one receiver or a group of receivers with similar channel condition and antenna configuration to provide a guarantee for each receiver (group) while leveraging multicast as much as possible. To maximize the effectiveness of the layered coding, it is important to optimize the amount of resources to spend on each layer given the receivers' resource constraints and channel conditions.

This observation leads to our following layered soft coding. We sort the receivers in an increasing order of their bandwidth budgets, B_i, as determined by their channel quality and numbers of antennas. The weakest receiver receives the lowest layer accurately as well as higher layers with larger errors, while the strongest receiver receives all the layers accurately. In general, we select N_i coefficients to target the i-th receiver. The i-th receiver will receive the layers lower than or equal to i completely to accurately recover the top $\sum_{j=1..i} N_j$ DCT coefficients, and also receive parts of the layers higher than i due to the broadcast nature of wireless medium to opportunistically recover more DCT coefficients. The latter recovery is opportunistic because the receiver has fewer constraints than the number of unknowns and incur more inference error for these unknowns. While such inference is opportunistic, it can still use LLSE to estimate the additional DCT coefficients (albeit with errors) and get substantial performance benefit than simply ignoring these receptions since these receptions give some constraints, which limit the solution space.

The amount of information in the layers higher than i that is received by the i-th receiver is determined by its channel quality and number of antennas. For example, if all the receivers have the same number of antennas and the i-th receiver has a loss rate of 60% while the other stronger receivers have no losses, the i-th receiver receives 40% of the higher layers. If all receivers have no losses, the i-th receiver has one antenna, and the receivers whose indices are larger than i have two antennas, the i-th receiver receives 50% of the higher layer.

Summary: The layered coding provides performance guarantees to all the receivers: the i-th receiver can at least get the top $\sum_{j=1..i} N_j$ DCT coefficients accurately, and may use the

receptions from the layers higher than i to further enhance performance. In comparison, without layering, a weaker receiver does not have enough constraints to accurately recover any coefficients, so its performance can be arbitrarily bad.

IV. Optimized Layered Integrated Video Encoding

In this section, we first give an overview of our scheme, and then describe details of each step.

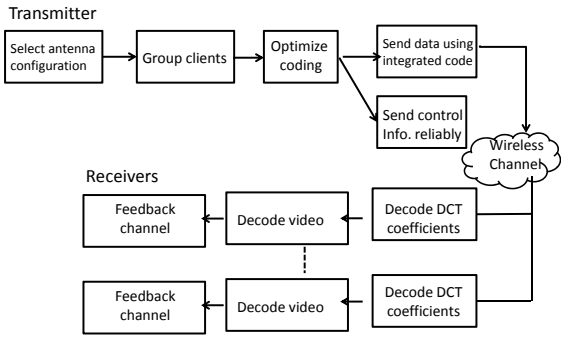


Fig. 1. Flow chart.

A. Overview

Figure 1 plots steps involved at the transmitter and receivers, where the steps are specified below.

1. Group/sort receivers: The sender sorts receivers in an increasing order of their throughput and labels them as G_1, G_2, ..., and G_n. If there are many receivers, the sender can optionally cluster multiple receivers with the same preferred antenna configuration and similar throughput to the same group and sort the groups according to their average throughput. Clustering receivers reduces the optimization problem size and speed up computation.
2. Select antenna configuration for each layer: Frames belonging to different layers in the video may use different transmission strategies. The sender and receivers always use all their antennas for transmission and reception. The main issue is to determine how many spatial streams to transmit for each layer. The number of streams to transmit in the i-th layer should be no more than the minimum number of antennas at the sender and all receivers in groups i or above. The additional antennas at the sender are used to achieve transmitter diversity and the additional antennas at any receivers are used to achieve receiver diversity.
3. Optimize layered soft coding: The sender determines how many transmissions (T_i) to make and how many DCT coefficients (N_i) to transmit at each layer.
4. Transmit video data using integrated video coding: The sender transmits according to the optimization result. That is, it broadcasts T_i constraints involving N_i DCT coefficients using G_i's antenna configuration and soft coding. The sender further enhances reliability of the most important DCT coefficients using integrated soft and hard modulation.
5. Transmit control information: The sender informs all receivers of $(N_1, N_2, ..., N_k)$ and tags the data transmission with an indicator of which layer it belongs to. Control information should be reliably delivered to all receivers.
6. Decode DCT coefficients: Upon receiving the data, the receiver groups the received packets according to the layers that they belong to. Then it uses all the received data for layer i to infer the corresponding N_i DCT coefficients.
7. Decode video: Each receiver puts the inferred DCT coefficients together into a single 3-D DCT matrix based on the index information transmitted as part of control messages, and performs inverse 3-D DCT to extract the current GoP. It repeats the same process for the next GoP.

Below we elaborate steps (3), (4), (5), and (6) since the other steps are straightforward.

B. Optimize Layered Soft Coding

Optimization approach: Our goal is to determine what information to transmit at each layer to optimize the overall video quality across all multicast group members subject to the bandwidth budgets of all receivers. More specifically, the DCT coefficients are sorted in a decreasing order. At the i-th layer, the video source makes T_i transmissions involving the top $\sum_{j=1..i-1} N_j + 1$-th to the top $\sum_{j=1..i} N_j$ DCT coefficients. For example, the first layer has the top N_1 DCT coefficients, the second layer has the next top N_2 DCT coefficients (*i.e.*, from the top N_1+1-th to the top N_1+N_2-th DCT coefficients), and so on. Our goal is to determine T_i and N_i for each layer i such that the total video quality across all receivers, denoted as $\sum_r U_r$, is optimized subject to the bandwidth budget constraints, where U_r is the r-th receiver's video quality.

We first present our optimization framework, and then describe how we approximate U_r using a simple function later in this section. Our framework is general, and can support other U_r functions and other ways of combining U_r across receivers, such as weighted sum of utility if receivers are not equally important and proportional fairness $\sum_r \log(U_r)$, which captures both fairness and total utility.

The optimization can be formally specified as follows:

$$max: \sum_r \sum_i U(T_i, p_{i,r}, \sum_{j=1..i-1} N_j + 1, \sum_{j=1..i} N_j)$$
$$s.t. \sum_{i=1..s} T_i/R_i \leq 100\% \quad (1)$$

where $p_{i,r}$ is the delivery rate of the i-th layer to the r-th receiver and $U(T_i, p_{i,r}, \sum_{j=1..i-1} N_j + 1, \sum_{j=1..i} N_j)$ is the utility of receiving $T_i \times p_{i,r}$ transmissions involving the top $\sum_{j=1..i-1} N_j$+1-th to the top $\sum_{j=1..i} N_j$-th DCT coefficients. The objective is essentially the sum of utility across all receivers r, where each receiver's utility is in turn the sum of its utility across all layers, where the utility of the i-th layer at receiver r is $U_r(T_i, p_{i,r}, \sum_{j=1..i-1} N_j + 1, \sum_{j=1..i} N_j)$. Different receivers extract different utility from the same layer due to their different delivery rates. The constraint captures resource limitation, where T_i/R_i denotes the fraction of time spent in T_i transmissions at the rate of R_i, and the complete constraint indicates the total time spent in transmitting for all layers should not exceed 100%. This optimization can be efficiently solved using $fmincon()$ in Matlab.

In this optimization, T_i's and N_i's are optimization variables and all the other variables, namely R_i and $p_{i,k}$, are given as input. R_i is determined based on the number of multiplexing

streams and delivery rates. Every receiver measures and feeds back $p_{i,r}$. The delivery rate, $p_{i,r}$, may vary across different video layers because each layer may use different antenna configurations for transmission and the delivery rate depends on the antenna configuration. For example, consider a node with good channel and one antenna, the delivery rate of the first layer transmitted using one stream is 100%, but its delivery rate of the second layer transmitted using two streams is 50%. $p_{i,r}$ can be measured by dividing the number of packets received at each layer by the total number of transmissions at this layer.

Approximating U: We assign utility based on Mean Square Error (MSE), defined as $E[(x_{est} - x_{actual})^2]$, where x_{est} and x_{actual} are the estimated and actual pixel values, respectively. Since we prefer a higher utility and a lower MSE, we use $-MSE$ as the utility. We plot MSE of receiving the top N DCT coefficients out of T total number of DCT coefficients in a GoP (*i.e.*, $U(T, 100\%, 0, N)$) using several popular videos in Figure 2 and observe they can be approximated using an exponential distribution CDF, namely $e^{-\lambda N/M}$, where $\lambda = 6$. The approximation is close in all cases.

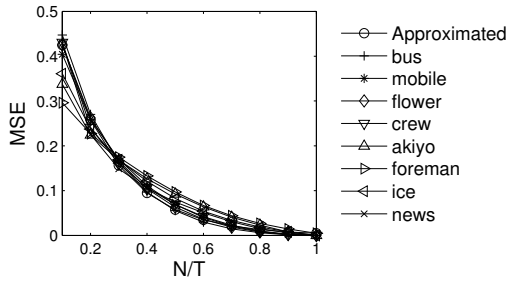


Fig. 2. Utility of real videos can be approximated using an exponential distribution.

Then the general utility $U(T, p, N_{start}, N_{end})$ can be approximated as follows:

$$\begin{aligned} &U(T, p, N_{start}, N_{end}) \\ \approx &U(T, 100\%, N_{start}, N_{end}) \times p \\ = &(U(T, 100\%, 0, N_{end}) - U(T, 100\%, 0, N_{start})) \times p \\ \approx &(e^{-\lambda N_{end}/T} - e^{-\lambda N_{start}/T}) \times p \end{aligned} \quad (2)$$

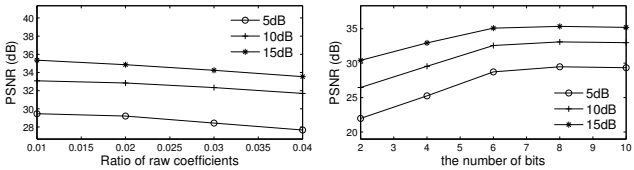
The first approximation is because when $p < 100\%$, the utility tends to decrease with p and a simple way to approximate such degradation is based on a linear function. The second equality is based on the definition of U. The third equality is simply plugging in our approximated $U(T, 100\%, 0, N)$.

C. Integrated Video Encoding

Hard modulation (*e.g.*, BPSK, QPSK, and QAM) is widely used in wireless transmission. It uses an analog signal to denote one or multiple bits. Since small errors may corrupt a few bits and lead to a loss of an entire GoP, modulation has to be chosen conservatively (*e.g.*, using effective SNR [6], which is dominated by weak subcarriers). On the other hand, soft modulation achieves a higher efficiency by using one signal to represent two real numbers. Moreover, errors in soft coding result in noise instead of corruption of a GoP. Nevertheless, soft modulation does not guarantee error-free delivery. White noise and interference can alter the received signal and cause decoding errors. Such errors may further be amplified when the desirable signal exceeds the power budget and needs to be scaled down before transmission to satisfy the power constraint. For example, if the signal to transmit is twice as large as the maximum power, the signal should be scaled down by half before transmission and scaled back up by a factor of two at the receiver. This also doubles the noise. This is especially problematic when sending the top DCT coefficients, which require a significant scale down before transmission and incur large error. Yet these top DCT coefficients are the most important, and small errors in these coefficients can cause significant degradation.

To address the issues, we use hard modulation to transmit the first few bits in the top DCT coefficients and use soft modulation to transmit the remaining bits in these coefficients as well as other coefficients. This has several benefits: (i) By using a conservative data rate, hard modulation is more reliable than soft modulation. This is most useful for sending the top DCT coefficients. (ii) Sending the remaining coefficients using soft modulation is more efficient. (iii) By removing the first few bits from the top DCT coefficients, their magnitudes become smaller. This means a smaller scaling factor (if any) can be used to fit into the power budget of the transmitter, thereby reducing noise amplification.

More specifically, in the integrated modulation, instead of sending linear combinations of DCT coefficients in all the data transmissions, we send the top $x\%$ raw DCT coefficients (out of the total number of transmissions across all layers) and the remaining transmissions are random linear combination results of DCT coefficients. For these top $x\%$ raw DCT coefficients, we use hard modulation to send their first y bits and use soft modulation to send the remaining portion of these DCT coefficients along with the remaining $1 - x$ DCT linear combination results. Hard modulation rate is selected based on effective SNR [6]. Figure 3 shows PSNR as we vary x and y. $x = 1\%$ and $y = 8$ bits give consistently high video quality under different SNR. So we use these values in our evaluation.



(a) Ratio of raw DCT coeff. (b) # bits for hard modulation

Fig. 3. Selecting parameters for integrated codes.

D. Transmit Control Information

There are two types of control information. One is a control message sent separately from the data. It includes the total number of layers, the number of coefficients for each layer (N_i), mean and variance of each DCT chunk, bitmap of the selected DCT chunks compressed using run-length coding, and random initial seed so that the sender and receivers can use to generate the same linear coding matrices without transmissions. The bitmap size is small and is further compressed to reduce the overhead [7]. The second type is control information for each data frame, including which layer the frame belongs to, sequence number, power normalization factor used to scale the transmission data to be within the power range of the transmitter (so that the receiver can scale it back), and how it is transmitted (so that the receivers know how to process it).

Both control information should be sent reliably to all the receivers. The second type of control information can be sent as part of PHY layer header, which is decodable by everyone regardless of its number of antennas. Similar to 802.11a PLCP header, it is transmitted as OFDM symbol with 64 subcarriers, BPSK modulation and 1/2 rate convolutional coding. When the field lengths of layer information, sequence number and the power normalization factor are 4 bits, 8 bits and 32 bits, respectively, two OFDM symbols are sufficient to convey this information. The first type of control information can be transmitted using standard hard modulation and use the standard ACKs/retransmissions to ensure reliability. The data rate for these control message is selected using effective SNR [6]. Moreover, in the MIMO context, the number of streams we use to transmit control information is no more than the minimum number of antennas at all receivers and sender. If the smallest number of antennas at the receivers and sender is N_{min}, the control information should be sent using at most N_{min} streams and the remaining antennas are used for diversity. In addition to the control information sent from the sender to the receiver, the receivers also periodically feed back the delivery rates of all layers using hard modulation.

E. Decode DCT Coefficients

The receiver constructs the following linear relationship for decoding. Given M transmission antennas and L receive antennas, the received signals Y on the L antennas have the following relationship with the DCT coefficients X: $Y = HCX + N$, where H is the channel matrix from transmitter antennas to receiver antennas, and C is the coding matrix generated by a random seed agreed between the sender and receiver, and N is white noise. It infers X based on Y using the standard Linear Least Square Estimator (LLSE) [9] as follows: $X_{LLSE} = \wedge_x (HC)^T ((HC) \wedge_x (HC)^T + \sum)^{-1} Y$, where $\wedge_x$ is a diagonal matrix whose diagonal entries are the variances of DCT chunks and $\sum$ is a diagonal matrix whose i-th diagonal element is the channel noise power incurred by the packet carrying i-th row of Y. The chunk variance is transmitted as part of control information.

F. Remarks

A recent trend of online video streaming is Dynamic Adaptive Streaming over HTTP (DASH). LIVE can be realized in the DASH framework by modifying video encoding, decoding, and piggybacking delivery rates of different video layers to DASH feedback with little extra overhead.

Our video encoding and decoding cost is dominated by DCT transform, and similar to SoftCast and MPEG. Let M and N denote the number of frames in a GoP and the number of pixels in a frame, respectively, the complexity of 3D-DCT is $O(MNlogN)$. The computation time of layered coding optimization takes around 60 ms for 4 client-cases where each client has different number of antennas. The time can be further reduced through code optimization and converting from Matlab to C implementation. This overhead is negligible since the optimization only needs to run when the multicast group membership or delivery rates of receivers change and optimization can take place in parallel to current video transmissions.

V. Testbed and Implementation

A. Testbed Evaluation Methodology

Testbed implementation: We implement our scheme (LIVE) and SoftCast in USRP software radio platform and Matlab. We generate I/Q samples from Matlab implementation, and feed them into USRP for OFDM processing. We modify USRP codebase to support MIMO transmitters and receivers. We run experiments in 2.4GHz channel and the channel bandwidth is 1MHz. We vary the channel condition by changing the location of USRPs and tx/rx gain parameters and perform experiments in various SNR environment. In LIVE, the sender knows the antenna configuration of the receiver in advance, and perform the layer optimization and transmit encoded signal. We also implement MPEG4 based on FFmpeg [3], whose GoP size is set to 25 (the default value in FFmpeg). It first encodes a raw video to MPEG4 part-10 format with various quantization parameter (QP) values, and then transmits encoded videos over USRP using digital modulation. We select the MAC data rate based on effective SNR [6]. The received signals are stored in traces, and processed in offline in MATLAB decoder. We also run SVC using JSVM [8] and observe it performed much worse than the above three approaches. Further optimization of SVC parameters may help improve its performance, but is expected to under-perform LIVE due to less efficient coding, lack of support for antenna heterogeneity (*i.e.*, receivers with fewer antennas extract no information from video sent with more antennas), and lack of effective optimization across layers.

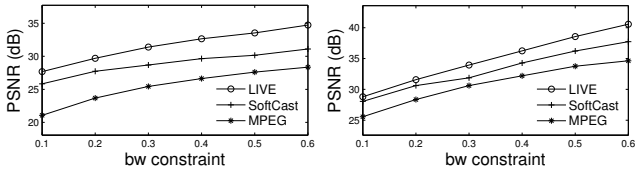
Performance metric: We use the average Peak Signal-to-Noise Ratio (PSNR) over all clients as the performance metric. PSNR is a standard video metric. It is defined as $PSNR = 20log_{10}\frac{2^L-1}{\sqrt{MSE}}$, where L is the number of bits to present pixel luminance and is usually set to 8. As mentioned in Section IV-B, our optimization can easily support other video quality functions.

Evaluation scenarios: We vary SNR, bandwidth constraint, the number of clients, and the number of antennas for each node. The bandwidth constraint we use represents the fraction of the DCT coefficients that can be transmitted given the channel bandwidth and control overhead assuming that SoftCast is used. For example, when the total number of coefficients in a GoP is 65536, the bandwidth constraints of 0.3 represents that SoftCast allows to transmit 30% (19961) coefficients. As the amount of control information in LIVE is larger than SoftCast and LIVE may use different antenna configurations for transmissions, we adjust the number of data transmissions in LIVE in order to make sure that both schemes (including both data and control traffic) take the same air time. In addition, we ensure MPEG4 to use the same air time as LIVE and SoftCast by selecting an appropriate QP value.

We use four popular video sequences: *bus*, *mobile*, *flower*, and *crew* [17]. Among them, *crew* have more static scenes and *mobile* is a more dynamic video. All videos have the frame size of 352 $\times$ 288 pixels, and each GoP consists of 4 frames. Unless otherwise specified, all the reported PSNR numbers in the testbed are the average PSNR over all four videos.

We compare both unicast and multicast performance. For unicast, the main benefit of our approach is to protect the most

important DCT coefficients. Such benefit stays the same for different numbers of antennas. Therefore, we use one antenna at both the transmitter and receiver for unicast evaluation. For multicast, the benefit comes from the ability to handle heterogeneous antenna configuration and channel conditions at different receivers, as well as integrated coding. Therefore, we use a range of antenna configurations at the receivers. LIVE uses MIMO spatial multiplexing when instructed by the optimization results, whereas MPEG4 and SoftCast have only one layer, whose number of streams is set to the maximum that can be received by all clients, and use the remaining antennas for diversity gain. We also try letting SoftCast and MPEG send more streams than this so that the stronger receivers can enjoy higher spatial multiplexing gain, but find the overall performance degrades significantly because the receivers with fewer antennas can incur arbitrarily large error even after applying our MIMO extension. This is because the weaker receivers do not get sufficient constraints to accurately decode any coefficients. Therefore, the number of streams in MPEG4 and SoftCast are bounded by the minimum number of antennas at the AP and receivers.

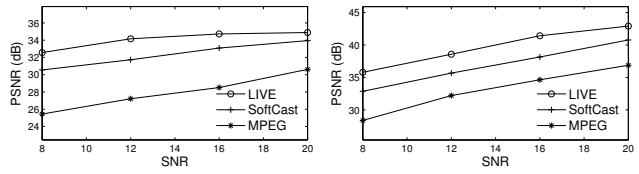


(a) SNR = 8 dB (b) SNR = 16 dB

Fig. 4. Video unicast testbed experiments with varying bandwidth constraint.

B. Testbed Results

Video unicast: Figure 4 compares unicast performance as the bandwidth constraint varies from 0.1 to 0.6 while the channel SNR is fixed to 8dB or 16dB. In both cases, LIVE > SoftCast > MPEG4. Across all videos, on average LIVE out-performs MPEG4 by 6.1 dB and SoftCast by 2.7 dB when SNR=8 dB, and out-performs MPEG4 by 4.1 dB and SoftCast by 1.9 dB when SNR=16 dB. This is significant improvement since differences of 1 dB or higher is visible, and 3 dB difference indicates that video quality is doubled. The improvement over SoftCast and MPEG4 comes from integrated video encoding to protect the most important DCT coefficients.

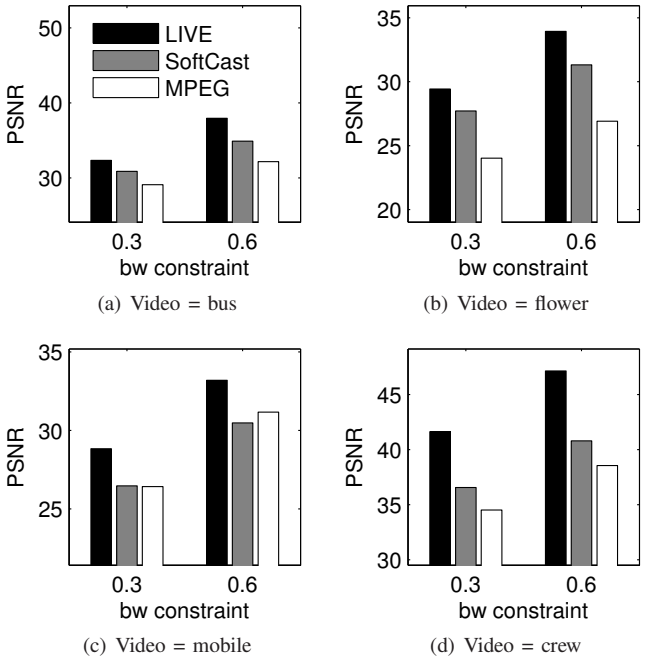


(a) bandwidth constraint = 0.3 (b) bandwidth constraint = 0.6

Fig. 5. Unicast testbed experiments with varying SNR.

Figure 5 compares unicast performance as we vary SNR while fixing the bandwidth constraint to 0.3 and 0.6. As before, LIVE consistently out-performs SoftCast and MPEG4. The average improvement is 5dB over MPEG4, and 2.4 dB over SoftCast. The gain tends to increase in lower channel quality since it is more important to protect the dominant DCT coefficients when the channel quality is poor.

Figure 6 further shows the unicast performance using different videos. We fix the channel SNR to 12dB and the bandwidth constraints to 0.3 or 0.6. As we can see, the

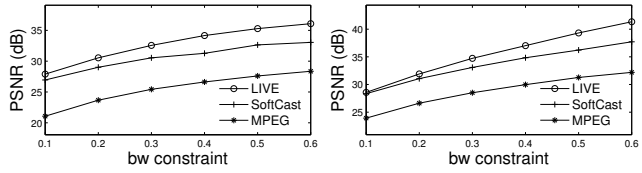


(a) Video = bus (b) Video = flower

(c) Video = mobile (d) Video = crew

Fig. 6. Unicast testbed experiments with varying testing videos (SNR=12).

performance gain varies by video: static video (*e.g.*, *crew*) achieves higher gain than dynamic video (*e.g.*, *mobile*). This is because the soft modulation is more effective when the information of the video is concentrated on the top few coefficients. Interestingly, SoftCast has slightly lower PSNR than MPEG4 when using *mobile* video. This is consistent with the result in [7], which shows SoftCast does not always out-perform MPEG4. In comparison, LIVE consistently out-performs MPEG4 and SoftCast in all cases.

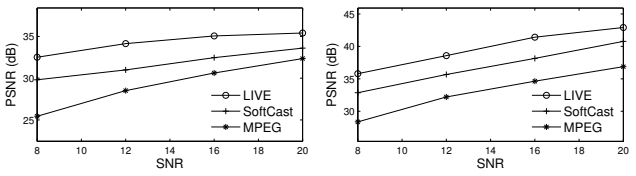


(a) SNR = 8 dB (b) SNR = 16 dB

Fig. 7. Multicast testbed experiments with varying bandwidth budget: a sender with two antennas sends to two clients with 1 and 2 antennas, respectively.

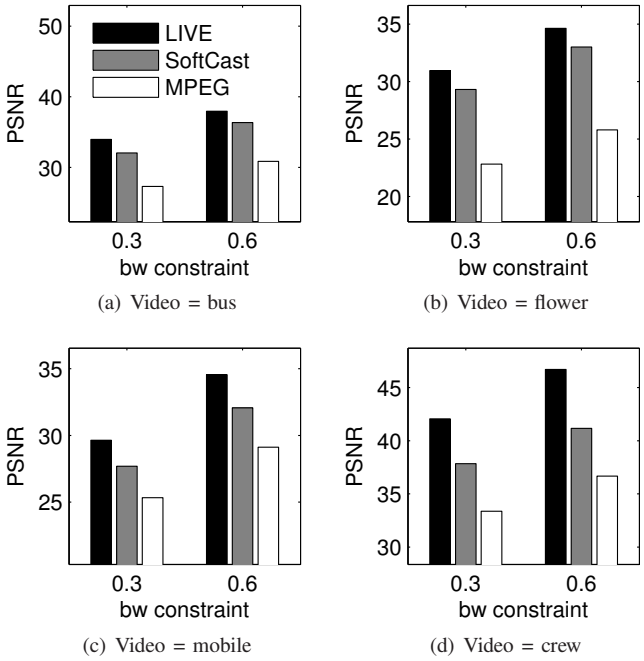
Video multicast: Next we evaluate multicast from a sender with two antennas to two receivers with one and two antennas, respectively. The sender determines the number of transmissions and coefficients to send by single-input single-output antenna (SISO) (the first layer) and by spatial multiplexing (the second layer) using the optimization described in Section IV-B. Figure 7 shows multicast performance as we vary the bandwidth constraint while fixing SNR to 8 or 16 dB. LIVE continues to out-perform both SoftCast and MPEG4. The improvement ranges between 4.6-9.3 dB over MPEG4, and between 0.2-3.6 dB over SoftCast. The performance gain in multicast is even larger than that in unicast because multicast further benefits from our optimized layered coding.

Next we vary SNR while fixing the bandwidth constraint to 0.3 and 0.6. Figure 8 shows LIVE improves over MPEG4 by 7.0 dB and over SoftCast by 2.2 dB.



(a) bandwidth constraint = 0.3 (b) bandwidth constraint = 0.6

Fig. 8. Multicast testbed experiments with varying SNR: a sender with two antennas sends to two clients with 1 and 2 antennas, respectively.



(a) Video = bus (b) Video = flower

(c) Video = mobile (d) Video = crew

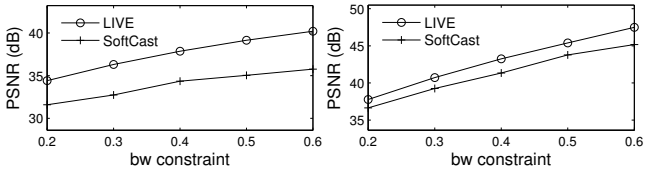
Fig. 9. Multicast testbed experiments under different videos: a sender with two antennas sends to two clients with 1 and 2 antennas, respectively.

Figure 9 summarizes the multicast performance using different videos. The results are consistent with unicast results where LIVE consistently out-performs the other two schemes and the improvement increases in static videos. Moreover, the improvement in multicast is larger than that of unicast due to effectiveness of optimized layered coding.

VI. Simulation

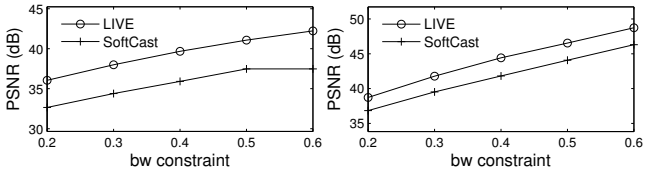
We implement our optimized layered integrated video encoding and SoftCast in Matlab, and compare their performance gain using PSNR. Simulation allows us to conduct a broader range of evaluation in a controlled environment. As in the testbed, we compare the multicast performance by varying SNR, bandwidth constraints, the number of clients, and the number of antennas for each node. Besides 4 videos used for the testbed experiment, we use 4 additional video sequences: *akiyo*, *foreman*, *ice*, and *news* [17]. We report an average of 5 random runs across the eight videos (40 runs in total), which is confirmed to be sufficient to get stable result. The evaluation setup for simulation is almost identical with USRP experimental setup, and the only difference is that Rayleigh fading channel model is applied in the received signal instead of receiving the signal over the air.

We first evaluate all the topologies used in the testbed, and find the simulation results are consistent with the testbed results. In the interest of brevity, below we present the results of topologies not in the testbed, including larger networks and mobile clients.



(a) SNR = 4 dB (b) SNR = 16 dB

Fig. 10. Video multicast simulation with varying bandwidth budget: a sender with 3 antennas sending to three clients with 1, 2, 3 antennas, respectively.



(a) SNR = 4 dB (b) SNR = 16 dB

Fig. 11. Video multicast simulation with varying bandwidth budget: a sender with 4 antennas sending to four clients with 1, 2, 3, 4 antennas, respectively.

Canonical topologies: Figure 10 and 11 show the multicast performance for 3 and 4 receivers, respectively. Here the transmitter has 3 and 4 antennas, respectively, and the numbers of antennas at the receivers range from 1 to 4. We make the following observations. First, in all cases, LIVE significantly out-performs SoftCast. The improvement ranges from 1.2-4.8 dB. Second, the improvement tends to increase with the multicast group size since optimized layer coding is more important when there are more diverse users with different budget constraints. Third, the improvement is noticeably larger in the bad channel condition than in the good channel (2.8-4.7 dB gain in 4dB channel vs. 1.2-2.6 dB gain in 16 dB channel) as integrated encoding is most useful under the bad channel.

Figure 12 further plots the performance of 4-client multicast as we vary SNR while fixing the bandwidth budget to 0.4. As we can see, LIVE out-performs SoftCast by 2.0 - 4.7 dB due to effective optimization of layered coding.

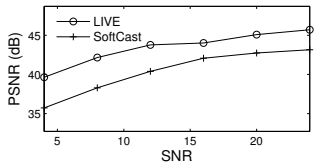
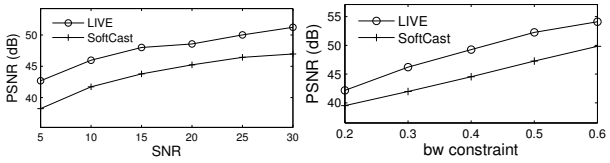


Fig. 12. Video multicast with varying SNR and bandwidth budget=0.4: a sender with 4 antennas sends to 4 clients with 1, 2, 3, 4 antennas, respectively.

Larger topologies: Next we evaluate using 10-client multicast groups where the sender has 4 antennas while each client has a random number of antennas between 1 and 4. Figure 13(a) and 13(b) plot PSNR under varying SNR and bandwidth budget, respectively. As before, we observe a significant improvement in PSNR, ranging 2.7-5.0 dB. The improvement stays high across all channel conditions.



(a) Varying SNR (b) Varying bandwidth

Fig. 13. Video multicast: a sender with 4 antennas sending to 10 clients with random numbers of antennas from 1 to 4.

In Figure 14, we vary the number of clients from 5 to 10 and assign each of them 1-4 antennas randomly. We observe 3.7-5.1 dB gain over SoftCast.

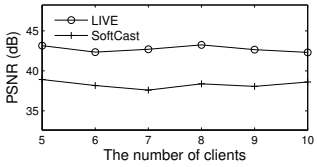
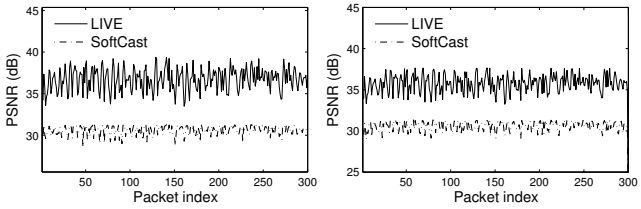


Fig. 14. Video multicast with varying numbers of clients: a sender with 4 antennas sending to 5 to 10 clients with random # antennas from 1 to 4.

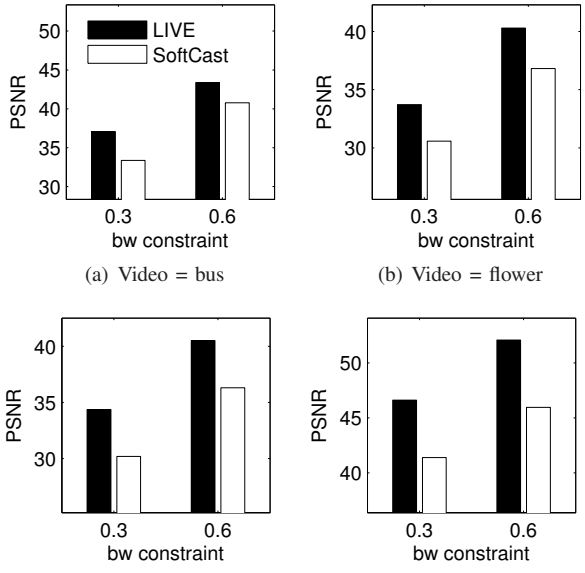
Mobile clients: We further evaluate the multicast performance of four mobile clients with random numbers of antennas ranging from 1 to 4. We collect the mobile traces using USRP at a walking speed, one for each client. The SNR ranges from 5 to 15 dB in the traces. We feed the traces to both LIVE and SoftCast. Figure 15 compares the PSNR of these schemes over time using the *crew* and *bus* videos. As we can see, LIVE consistently out-performs SoftCast. The improvement ranges between 3.9 to 7.9 dB. This demonstrates the feasibility of LIVE in mobile scenarios.



(a) Video = crew (b) Video = bus

Fig. 15. Mobile trace driven evaluation: a sender with 4 antennas sending to 4 clients with random numbers of antennas ranging from 1 to 4.

Impact of video types: Figure 16 compares PSNR in 4-client multicast groups. The improvement is significant across all the videos. It ranges from 2.6 to 6.1 dB.



(a) Video = bus (b) Video = flower

(c) Video = mobile (d) Video = crew

Fig. 16. Video multicast simulation with 4 transmitter antennas with varying testing videos (SNR=10).

VII. Conclusion

In this paper, we propose a novel optimized layered integrated coding for wireless video delivery. It can accommodate heterogeneity arising from different channel conditions and different numbers of antennas at the receivers by sending layered integrated coded transmissions to benefit all receivers and strategically optimizing the resource allocation across different layers to guarantee the performance for each receiver. Using extensive Matlab simulation and testbed experiments, we show our approach out-performs SoftCast, the state-of-the-art video delivery, by 1.9 – 3.5 dB in unicast and by 2.2 – 4.7 dB in multicast, and out-performs MPEG4 by 4.1 – 6.1 dB in unicast and by 4.6 – 9.3 dB in multicast.

Acknowledgements: This work is supported in part by NSF Grants CNS-1017549 and CNS-1343383.

References

[1] S. Aditya and S. Katti. Flexcast: Graceful wireless video streaming. In *Proc. of ACM MOBICOM*, 2011.

[2] D. Dardari, M. G. Martini, M. Mazzotti, and M. Chiani. Layered video transmission on adaptive OFDM wireless systems. *EURASIP J. Appl. Signal Process*, 2004.

[3] FFmpeg. http://www.ffmpeg.org/.

[4] M. M. Ghandi, B. Barmada, E. V. Jones, and M. Ghanbari. H.264 layered coded video over wireless networks: Channel coding and modulation constraints. *EURASIP J. Appl. Signal Process.*, 2006.

[5] V. K. Goyal. Multiple description coding: Compression meets the network. *IEEE SIGNAL Proc. Magazine*, 2001.

[6] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. In *Proc. of ACM SIGCOMM*, 2010.

[7] S. Jakubczak and D. Katabi. A cross-layer design for scalable mobile video. In *Proc. of ACM MOBICOM*, 2011.

[8] JSVM. http://ube.ege.edu.tr/~boztok/JSVM/SoftwareManual.pdf.

[9] C. Lawson and R. Hanson. Solving least squares problems. *Society for Industrial Mathematics*, 1987.

[10] X. L. Liu, W. Hu, Q. Pu, F. Wu, and Y. Zhang. Soft video delivery in MIMO WLANs. In *Proc. of MobiCom*, 2012.

[11] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the scalable video coding extension of the H.264/AVC standard. In *Proc. of ISCAS*, 2007.

[12] S. Sen, S. Gilani, S. Srinath, S. Schmitt, and S. Banerjee. Design and implementation of an approximate communication system for wireless media applications. In *Proc. of ACM SIGCOMM*, 2010.

[13] M. Skoglund, N. Phamdo, and F. Alajaji. Hybrid digital-analog source-channel coding for bandwidth compression/expansion. *IEEE Trans. on Info. Theory*, 2006.

[14] M. Stoufs, A. Munteanu, J. Barbarien, J. Cornelis, and P. Schelkens. Optimized scalable multiple-description coding and FEC-based joint source-channel coding: A performance comparison. In *Proc. of WIAMIS*, 2009.

[15] SVC reference software. http://ip.hhi.de/imagecom_G1/savce/downloads/.

[16] A. Watson. Image compression using the discrete cosine transform. *Mathematica Journal*, Jan. 1994.

[17] Xiph.org media. http://media.xiph.org/video/derf/.

[18] Q. Xu, V. Stankovic, and Z. Xiong. Distributed joint source-channel coding of video using raptor codes. *IEEE JSAC*, 2007.