Design and Implementation of an "Approximate" Communication System for Wireless Media Applications

Sayandeep Sen, Tan Zhang, Syed Gilani, Shreesha Srinath, Suman Banerjee, *Member, IEEE*, and Sateesh Addepalli

***Abstract*—All practical wireless communication systems are prone to errors. At the symbol level, such wireless errors have a well-defined structure: When a receiver decodes a symbol erroneously, it is more likely that the decoded symbol is a good "approximation" of the transmitted symbol than a randomly chosen symbol among all possible transmitted symbols. Based on this property, we define *approximate communication*, a method that exploits this error structure to natively provide unequal error protection to data bits. Unlike traditional [forward error correction (FEC)-based] mechanisms of unequal error protection that consume additional network and spectrum resources to encode redundant data, the approximate communication technique achieves this property at the PHY layer without consuming any additional network or spectrum resources (apart from a minimal signaling overhead). Approximate communication is particularly useful to media delivery applications that can benefit significantly from unequal error protection of data bits. We show the usefulness of this method to such applications by designing and implementing an end-to-end media delivery system, called Apex. Our Software Defined Radio (SDR)-based experiments reveal that Apex can improve video quality by 5–20 dB [peak signal-to-noise ratio (PSNR)] across a diverse set of wireless conditions when compared to traditional approaches. We believe that mechanisms such as Apex can be a cornerstone in designing future wireless media delivery systems under any error-prone channel condition.**

***Index Terms*—Algorithms, design, measurement, modulation coding, performance evaluation.**

I. Introduction

ANY COMMUNICATION system, whether wired or wireless, is impacted by errors. Typically, a transmitter encodes application data in a frame as a set of symbols, where each symbol encodes one or more data bits. When errors happen, one or more of these symbols are incorrectly decoded at the receiver, leading to loss in performance.

Conventional designs of networked systems have assumed that communication channels are error-prone and handle them in various ways. Often, a checksum field is added to each data unit. A receiver verifies the received checksum against the expected checksum for the data bits to validate correct reception. Higher layers deal with these errors in different ways.

At the MAC layer, some protocols (such as 802.11) typically discard packets received with checksum errors. A newer class of MAC-PHY mechanisms attempt to recover correct bits from packets with partial errors. Examples include partial packet recovery (PPR) [18], SOFT [36], ZipTx [21], and Maranello [13]. Similarly, at the transport layer, TCP would retransmit erroneous or lost segments. UDP simply discards them silently. UDP-lite delivers erroneous packets to applications and allows applications to recover correct portions of these packets using partial checksums.

In all the above techniques, it is the job of the receiver to determine and discard erroneous bits within a packet, and the transmitter has no *a priori* knowledge of which bits (or bit positions) are likely to be in error. In this paper, we demonstrate an alternate possibility: *At the PHY layer, there is a systematic structure to wireless errors, and a wireless transmitter can leverage this structure to ensure that certain bit positions are less likely to be in error than other bit positions*. If the transmitter is aware of relative importance of data bits, it can place more important data bits in more protected positions and the less important data bits in less important positions. In other words, a transmitter can ensure an unequal error protection (UEP) of its data by simply placing data bits in specific bit positions. Thus, unlike prior approaches of UEP that explicitly need to add redundant bits to higher-priority data [15], [24], [35], this proposed approach would not require any redundant data bits to be transmitted to achieve UEP and is available *natively* from the wireless channel (apart from minimal signaling overhead).

Furthermore, adaptation of the relative degree of UEP available to different bit positions can be done based on application needs and ambient channel condition.

The relevant error structure across the wireless channel is a consequence of the following phenomenon: When a wireless symbol is decoded in error, this erroneous symbol is still a good "approximation" of the original transmitted symbol. (We will make this notion more precise in Section II.) Based on this approximation property, we design a wireless communication method to provide UEP that we call *approximate communication*. Approximate communication can provide significant performance gains to all applications in which different data bits have different levels of priority, e.g., the I-, P-, and B-frame structure in MPEG-4-AVC [3] encoded video. In particular, we design and implement a specific approximate communication system called Apex (an Approximate communication system

Manuscript received September 17, 2011; revised June 29, 2012; accepted August 22, 2012; approved by IEEE/ACM Transactions on Networking Editor K. Papagiannaki. Date of publication December 13, 2012; date of current version August 14, 2013.

S. Sen, T. Zhang, S. Gilani, S. Srinath, and S. Banerjee are with the University of Wisconsin–Madison, Madison, WI 53706 USA (e-mail: sdsen@cs.wisc.edu).

S. Addepalli is with Cisco Systems, San Jose, CA 95134 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2012.2226470

for media exchange) that combines knowledge of relative bit priorities of the application with the structure of wireless errors and client feedback on channel conditions to achieve improved performance for media applications.

Apex requires a few, albeit marginal, modifications to the radio transmit–receive paths. These modifications include a few additional buffers, multiplexers, and a little amount of additional addressing logic. The latencies due to such additional data processing is negligible. Through detailed experiments on the WARP SDR platform [6] under various interference and multipath scenarios, we demonstrate how Apex can improve the quality of video delivered over a wireless link, using a mechanism that is complementary to existing approaches.

Key Contributions: We summarize the above discussion by identifying the main contributions of our work.

- *Apex, an approximate communication system to improve applications:* We show that wireless errors have a well-defined structure at the PHY layer. This structure allows the wireless PHY layer to natively provide UEP to data bits, which form the foundation of approximate communication.

  Furthermore, we show that by suitably adapting system parameters, based on current channel conditions and application requirements, Apex significantly improves performance of applications that prioritize different data bits differently.
- *Design of modifications to transmit–receive paths of an 802.11 radio to support approximate communication in Apex:* Apex requires minor modifications to the transmit–receive paths of a wireless radio. We describe how the transmit–receive chains of present day 802.11 hardware should be modified to implement approximate communication. We demonstrate such modifications can coexist with other complementary mechanisms implemented in the lower layers, such as a scrambler system, convolution coding, data interleaving, data modulation, and PHY transmission rate selection.
- *Implementation of the Apex-based video delivery system:* We demonstrate that Apex can be practically implemented by building a prototype system using the WARP Software Defined Radio (SDR) hardware and running H.264 video streams through them over the air.
- *Experimentation over range of scenarios:* We show that Apex can, indeed, provide significant performance gains for media applications over a range of wireless communication scenarios. In different experiments, we varied the transmit power levels, the position of nodes, and the degree of external interference to demonstrate the robustness of our schemes. Over various experiment scenarios, it provided a video quality improvement of 5–20 dB [measured in terms of peak signal-to-noise ratio (PSNR)].

The fact that wireless symbol errors of popular modulation schemes have a specific error structure has both been analytically studied by researchers [8], [2], [23], [38] and experimentally demonstrated [30]. However, many popular wireless communication standards, e.g., IEEE 802.11, consider such a structure to be a nuisance. In fact, standards such as IEEE 802.11 typically employ a set of randomization and data protection mechanisms (convolution coding, data scrambling, and data interleaving) to ensure that all application bits are equally likely to be in error. These data randomization mechanisms have a number of other advantages. Hence, their wide adoption and specific placement in the protocol stack ensured that opportunities of natively providing UEP on the wireless channel was lost.

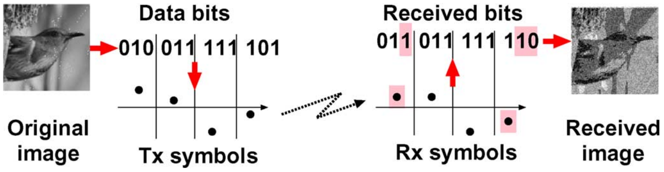


Fig. 1. High-level overview of wireless communication. (Some MAC layer operations, e.g., coding, scrambling, and interleaving are not shown.)

In this paper, we explain how the wireless symbol error structure can be leveraged to implement UEP in approximate communication (Sections II), how it can be practically implemented in Apex (Section IV) to coexist with the popular data randomization and protection techniques (convolution coding, data scrambling, and data interleaving), and how relevant parameters like modulation scheme and constellation mappings can be dynamically adapted (Section IV) to improve the quality of media delivered.

We believe that approximate communication can offer a useful design alternative for wireless media delivery systems and facilitate greater interaction between characteristics observable at the PHY layer and expectations of the applications.

II. Intuition and Approach for Approximate Communication

Fig. 1 shows a high-level (and somewhat simplified) schematic of data transfer across a wireless link as it happens for common technologies today. Let us assume that the content is encoded into a data bit sequence using a popular format, e.g., MPEG-4 [3]. After packetization, the transmitter of the wireless link maps these bits into symbols for transmission across the wireless channel. In our example, each symbol represents a set of three data bits. The receiver attempts to infer the transmitted symbols, but sometimes makes decoding errors. Hence, when the received symbols are mapped back into a bit sequence, bit errors might result (erroneous bits and symbols are shaded). An important thing to note is that when a symbol error occurs, not all its constituent bits are actually in error. Symbol errors in common wireless technologies, e.g., those based on the popular I/Q modulation schemes, have a very well-defined structure. When a receiver decodes a symbol erroneously, it is more likely that the decoded symbol is a good "approximation" of the transmitted symbol than a randomly chosen symbol among all possible symbols. In this section, we explain this phenomenon through an example based on quadrature amplitude modulation (QAM).[1]

In QAM, data elements are encoded into amplitude values of two sinusoidal waves that are 90° out of phase with each other. A QAM modulation scheme is usually represented by a I/Q constellation diagram, as shown in Fig. 2. Each constellation point (or symbol) is mapped to the amplitude of the in-phased and

[1] Some of the different mechanisms manipulated in the paper might appear similar in meaning. Hence, for the sake of clarity, we define the use of our terms and their meaning in Table I.

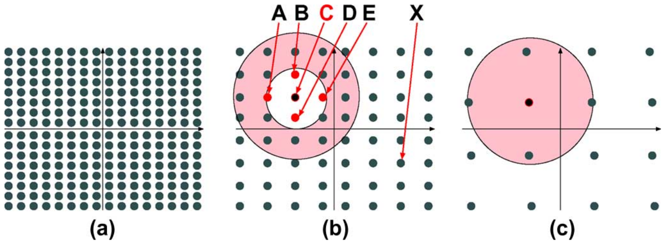


Fig. 2. QAM constellations. (a) 256 QAM. (b) 64 QAM. (c) 16 QAM.

TABLE I
SOME TERMINOLOGY USED FOR DIFFERENT MECHANISMS USED IN THE PAPER

Term	**Meaning / Examples**
Modulation scheme	16-QAM, 64-QAM, etc.
Constellation map	Mapping from bit-sequence to symbols (Gray, Block, etc.)
Data bit placement	Mapping of data bits to bit positions in symbols

the quadrature-phased signals and corresponds to a certain bit sequence to be transmitted. In a 256-QAM scheme [shown in Fig. 2(a)], there are 256 distinct symbols. Each such symbol encodes a 8-bit sequence of data to be transmitted. In contrast, in a 64-QAM scheme, there are 64 distinct symbols, and each symbol encodes a 6-bit data sequence [Fig. 2(b)]. When a transmitter wants to communicate a specific bit sequence, it emits the corresponding symbol. The goal of the receiver is to identify which symbol was sent.

Let us consider the case of a transmitter-receiver pair using 64-QAM and the transmitter emits a symbol, say C [Fig. 2(b)]. Depending on the conditions of the channel, there is a reasonable chance that the receiver will correctly decode this symbol. However, there is also some possibility for the receiver to make an error in decoding this symbol. It turns out that if an error does occur, such errors are most likely confined to the near neighborhood of the transmitted symbol C, i.e., the receiver is more likely to decode this symbol to be one of A, B, D, or E, which are the nearest neighbors of C [indicated by the inner circle of Fig. 2(b)], as compared to a faraway symbol, X. In other words, within the two-dimensional I/Q space, an erroneously decoded symbol tends to be a reasonable approximation of the transmitted symbol.

As noise in the environment increases, the likelihood of making incorrect decoding decisions to "faraway" symbols can increase to some extent [as indicated by the larger circle in Fig. 2(b)]. However, likelihood of making errors within close proximity also increases and continues to significantly dominate such faraway errors. We illustrate this behavior in Fig. 3, in which 10 million 64-QAM symbols were transmitted between a transmitter–receiver pair under two different conditions, ambient noise, and explicitly generated interference in the background. The y-axis represents the probability that the transmitted symbol is decoded erroneously to be another symbol at a distance given in the x-axis. The logarithmic scale of the y-axis demonstrates the dramatic decay in this probability with increasing distance between the transmitted and decoded symbols.

Furthermore, if the noise conditions continue to increase in this manner, a reasonable communication system will adjust to

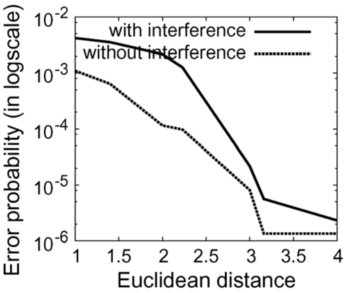


Fig. 3. Fast decay in probability of erroneous decoding as the distance between the correct symbol and the erroneously decoded symbol increases. Data based on 10 million known 64-QAM symbols transmitted. In the ambient noise scenario, the overall BER was 3×10^{-4}. In the explicit interference scenario, the overall BER was 6×10^{-3}.

a lower rate modulation scheme, such as 16-QAM [Fig. 2(b)], which separates constellation points farther apart. This allows the system to revert back to the original scenario where erroneous symbols are mostly the nearest neighbors of the transmitted symbol.

A. UEP and Approximate Communication

The wireless error structure demonstrated in Fig. 3 leads to a construction of UEP for data bits. This is achieved by appropriately choosing a constellation map, i.e., the map between bit sequences and symbols in a constellation diagram.

For the sake of simplicity, let us assume that when symbol errors occur, the erroneous symbol is just one unit away from the actual transmitted symbol. In the 64-QAM example in Fig. 2(b), this assumption implies that when C is transmitted, a decoding error would result in the received symbol to be one among A, B, D, and E.

Now imagine if a constellation map was to map bit sequences randomly to symbol positions. Then, when an error occurs, the likelihood of a correct value in any given bit position is purely by chance. The probability of such an event is 0.5, i.e., the probability that a neighboring symbol has the same most significant bit (MSB) value as the MSB of the actually transmitted symbol is 0.5, and the same is true for the least significant bit (LSB) and all other bit positions in between. Such a constellation map does not help in achieving our desired goal in approximate communication.

Fortunately, various common constellation maps, e.g., the Gray code (used in 802.11) map bit sequences to symbols in a way that increases the resilience of certain bits, even when the symbols are in error. As an example, the Gray code corresponding to a 16-QAM constellation is shown in Fig. 4. If we examine the MSB of the different symbols to the left of the Q-axis, we can observe that all of them have a value of 0. Similarly, the MSB of different symbols to the right of Q-axis all have a value of 1. In such a case, if a decoding error occurs for any symbol within the shaded region, there are no errors in the MSB (under our assumption that symbol errors do not exceed one unit). For the remaining symbols, we expect an error probability of either 1/4 or 1/3 for the MSB, depending on its position in the constellation. If the likelihood of transmitting each symbol is identical, then the probability of error in the MSB, given a symbol error has occurred, is 1/6. For the LSB, on the other hand, the probability of making an error, given a symbol error has occurred,

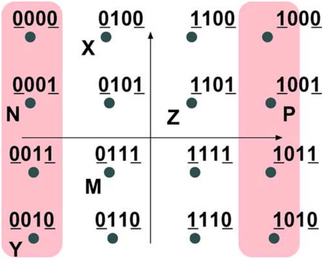


Fig. 4. 16-QAM with Gray code constellation map. MSBs and LSBs are underlined.

is 1/3. This gives rise to an intriguing possibility. If an application identifies different priority levels for its various data bits, then instead of performing data bit placement (mapping these data bits to bit positions in symbols) in an agnostic manner, we can achieve the desired impact of differential data protection by placing the higher-priority data bits to the MSB positions of symbols and the lower-priority data bits to the LSB (and other) positions.

Note that such differential protection of data values is possible by simply placing the data bits to appropriate bit positions, and without adding any form of redundancy into the system. In a way, such data protection is available natively through the wireless channel, and can be better exploited by applications, without imposing any overheads. This is in contrast to the traditional forward error correction (FEC)-based methods for providing UEP that require communication of additional bits that redundantly encode more important data.

Also, while we explained the phenomenon of differential error resilience of bit positions with the example of QAM modulation, the same would hold for other modulation schemes such as phase shift keying (PSK), pulse position modulation (PPM), frequency shift keying (FSK), etc., as well.

Based on the above intuition, we build an approximate communication system that can provide differential protection to different data bits of an application. We use streaming of MPEG-4 encoded video as an example of such an application. Such a video stream consists of frames with different levels of importance, e.g., I-frames are more important than P-frames, which in turn are more important than B-frames.

Akin to regular communication systems, the application creates and continuously sends data bits corresponding to different frames (I, P, and B) down the network protocol stack. Bits from each video frame will be packetized in smaller chunks at the network and MAC layers, and then will be handed into the digital component of the PHY layer. This is shown in Fig. 5, with four video frames—two I-frames with data 0100 and 0010, a P-frame with data 1101, and a B-frame with data 1001. In traditional communication systems, the 4-bit sequences of each frame gets placed together in a single wireless frame, e.g., the first I-frame is mapped to the first wireless frame (consisting of a single symbol X). The key difference in approximate communication is that data bits of different priority levels will be placed together and combined into a single symbol, such that the most protected bit positions are occupied by the higher-priority data bits (say, I bits), and the least protected bit positions are occupied by the lower-priority data bits (say, P and B bits). For example, the first two bits of the first I-frame, i.e., 01, are mapped to MSBs of the first symbol of a wireless frame, while the second two bits of the same I-frame, i.e., 00, are mapped to the MSBs of the symbol of the next frame.[2] The bits of P- and B-frames are placed into the LSB positions of different symbols. Overall, this means the I-frame bits will be better protected than the P- and B-frame bits.

When symbols are decoded (possibly in error), the receiver simply passes them along to the higher layers, even if in error. When the different application bits are extracted out of these erroneous symbols, the highest-priority bits are most likely to be correct and are of great value to the application. Lower-priority ones maybe in error and will be suitably handled. The key observation is that the higher-priority bits are thus successfully extracted out of the symbols, even if in error, through this mechanism, without requiring retransmissions.

This differential protection is in contrast to the traditional communication method used in 802.11a/b/g, where I-frame bits occupy both MSB and LSB positions of a symbol, the same as P- and B-frame bits, and there is no difference in the level of protection for bits of different video frames.

III. Validation

To validate that our approach of approximate communication is indeed possible, we need to identify one or more constellation maps that provide unequal error protection among different bit positions, *across a range of wireless conditions—different transmit power levels, different degrees of interference at the receiver, and with and without PHY-layer convolutional codes*. Such confirmation of unequal protection will provide further evidence that our approximation property holds true for a range of common scenarios, where approaches such as Apex can be quite worthwhile.

Given a constellation diagram of k points, each symbol will represent a bit sequence of length $\log_2(k)$ bits. Overall, there are no more than $k!$ different constellation maps possible, although some of them can be identical due to rotation and reflection-based symmetries. Each constellation map is likely to protect individual bit positions differently. The above property can be leveraged by altering the constellation map during an ongoing transmission, based on the channel conditions and the relative priority of the application data bits.

In this section, we present results for two constellation maps, each of which offer varying amount of error protection for different bit positions.

Both are, therefore, suited for approximate communication. We note that our contribution is not in designing these constellation maps, but in identifying suitable ones and in exploiting them to implement approximate communication for media applications. We first briefly describe the schemes, and then evaluate their error protection properties in the rest of this section.

A. Example Constellation Maps

We evaluate the following two constellation maps for use in approximate communication.

[2] In our actual implementation, the I-, P-, and B-frame bits would first go through the usual 802.11-style digital PHY processes, including the scrambler, convolution coder, and the interleaver for added resilience, before being mapped to the symbols.

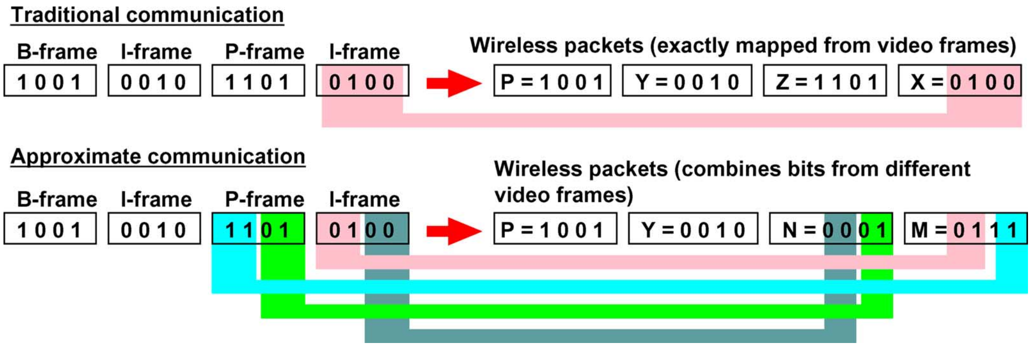


Fig. 5. Wireless frames in approximate communication combines bits from video frames of different priority levels. Example uses 16-QAM and assumes that the two MSBs can be better protected than the two LSBs. The data bit placement strategy maps I-bits to MSBs and P- and B-bits to LSBs. Symbols M, N, P, X, Y, and Z are as shown in Fig. 4 that uses a Gray code constellation map. (Only subset of data bit placements shown.)

Gray Code: In Gray code [12], symbols that are immediate neighbors either along the I-axis or the Q-axis of the I/Q space differ in exactly one bit position; the rest of the bits are identical between the two neighbors. An example of a Gray code is shown in Fig. 4 for a 16-QAM system. Gray codes can be constructed for any QAM scheme in a systematic manner based on the above observation. Gray code is widely used in many popular communication systems, including in 802.11a/g/n and 802.16-based systems.

Block Code: In block code [23], the constellation points on the same side of the I- (Q-) axis have the same value for the first (second) bit position. Hence, all points in a given I-Q quadrant have the same value for the first two bit positions. For each of these quadrants, the symbols are partitioned into four subquadrants, and the same process is repeated to assign bit values for the next two bit positions. The process is repeated iteratively for the remaining positions.

Both of these schemes can be implemented on different QAM-based modulation schemes. While we have implemented and experimented with different QAM schemes in our SDR radio platform, for sake of uniform comparison, in this section we focus on the 64-QAM scheme. In 64-QAM, each symbol encodes 6-bit sequences, and in diverse experiments we observe how the different schemes differentially protect the two MSBs, the two middle bits (MIDs), and the two LSBs.

B. Experiment Configurations

We use the following experiment parameters.

Hardware and Software: We implemented our system on the WARP SDR radio platform [6]. In our implementation, we used the WARP boards as RF front end to transmit the packets over air and carried out the digital-layer PHY signal processing activities in the connected laptop. The laptops also acted as the traffic source and sink. The setup for our experiments is depicted in Fig. 6. We summarize various aspects of our implementation in Table II.

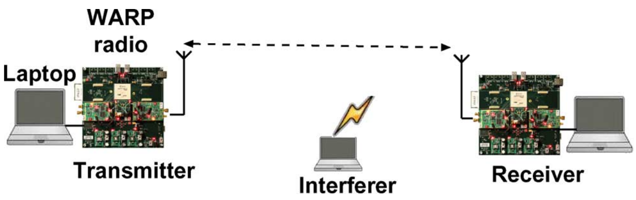


Fig. 6. Setup for communication between the two WARP SDR radios. The relative position, transmit power levels, and degree of external interference varied during different experiments.

TABLE II
PROPERTIES OF OUR IMPLEMENTED SYSTEM

Modulation schemes	16 and 64 QAM
Encoding schemes	Gray, Block
Frequency	2.4 GHz
Convolutional codes (when used)	Generator polynomials: $g0(133)$ $g1(171)$, Rate = $1/2, 2/3, 3/4$
Base data rate	485 Kbps (at rate $1/2$)

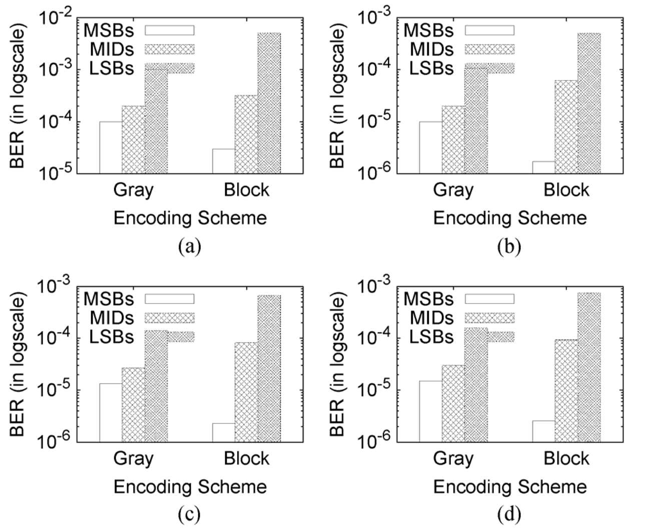
Frequency and phase synchronization between the sending and receiving side was achieved using an 11-symbol barker sequence sent as a preamble along the packet. We experimented by varying the transmit power levels of the transmitter, adding external source of interference, and changing the location of the nodes. The receiver sensitivity of WARP hardware in our implementation is -70 dBm for 64-QAM, and at lower RSSI values (< -70 dBm)[3], it becomes difficult to derive any meaningful results. Hence, for all of transmit power variations, we report on RSSI values that varied between -30 and -70 dBm.

Metrics: In this section, we are interested in the relative bit error rates (BERs) experienced by different bit positions of a symbol. Hence, we compare the two constellation maps by examining the BERs of the MSBs, MIDs, and LSBs across different conditions. In certain plots, we report *BER gain*, which is the ratio of the BER of Block code to that of Gray code, for specific bit positions. BER gain can thus be a number greater than 1.

Experiment Parameters: In each experiment, we sent 10 million randomly generated symbols. For each symbol received, we decode its value using both maps and calculate the error rates for different bit positions.

A typical 802.11 PHY layer (optionally) applies convolution code (a PHY-layer FEC) to further protect the data bits. In order to evaluate the effect of such PHY-layer convolution codes, we have also experimented with (and without) different convolution codes (rates include 1/2, 2/3, and 3/4) as well. Our results show that unequal error protection property holds both with and

[3] We calculate the RSSI values by reading the MAX2829 ADC output register value and mapping it to corresponding RSSI value from the datasheet [7].



(a) (b) (c) (d)

Fig. 7. BER of various bit positions when using Gray and Block data coding schemes. The overall uncoded BER of during the experiment was order of 10^{-3} for both schemes. Ten million QAM-64 symbols were transmitted. (a) No PHY convolution codes. (b) With PHY convolution code, Rate $= 1/2$. (c) With PHY convolution code, Rate $= 2/3$. (d) With PHY convolution code, Rate $= 3/4$.

without convolution codes. In this paper, we show the error resilience of different bit positions in presence of PHY-layer convolution coding only for a convolution code of rate 1/2.

C. Experimental Results

We describe our experiments and associated observations.

With and Without PHY Cnvolution Codes: We plot the error rate at the MSBs, MIDs, and LSBs for the two constellation maps at an intermediate transmit power level (RSSI -55 dBm) in Fig. 7, both with and without PHY convolution codes (rates 1/2, 2/3, and 3/4). Each constellation map provides different levels of protection to these bit positions. In both cases, Block provides the greatest error protection to the MSBs at the cost of lower protection to the LSBs. The use of convolution codes clearly reduce the BERs suitably in proportion to the degree of redundancy introduced by each code, but the effect of unequal error protection exists both with and without use of convolution codes. Hence, in the rest of this paper, we focus on results with PHY convolution codes applied.

Ambient Noise, Varied Transmit Power Levels, With PHY Convolution Codes: We present the performance of the Gray and Block constellation maps for an entire range of received power (varied by changing the transmit power levels as well as the separation between nodes and their locations) when using convolution codes in Fig. 8. Instead of the BER, we plot the *BER gain*. With decreasing RSSI, symbol errors increase. As a consequence, the BER of Block shrinks in comparison to BER of Gray for MSBs, i.e., Block protects MSBs even better compared to Gray as RSSI decreases. For the range of operating parameters, the relatively greater protection of MSBs using Block changes from a factor of 2 (at -30 dBm) to a factor 15 (at -70 dBm). The increased protection of MSBs comes at a cost of decreased protection for MIDs and LSBs in Block.

Varied External Interference: We next examine the behavior of different encoding schemes in presence of an explicit source of external interference (Fig. 9). We had used a laptop sending WiFi traffic as the source of interference. As a nearby WiFi source would back off on sensing transmissions from our system, we had to place our interfering source with some care to turn the WARP radio and the interfering laptop into hidden nodes for each other. For this, we reduced the signal strength on the laptop and placed it at a far by location such that the achieved throughput for both systems at their highest data rates (manually fixed) was the same regardless of whether the other node was working on not. We then increased the power on the laptop to a level such that it started interfering with the Warp node (confirmed by increased errors when the laptop was transmitting).

We show the BER gain in the plot. The external interferer injects regular 802.11 traffic into the wireless medium, which causes significant symbol errors in the data transmitted between WARP nodes. Across all levels of interference, Block provides significantly higher protection of MSBs than Gray.

IV. Design and Implementation of Apex

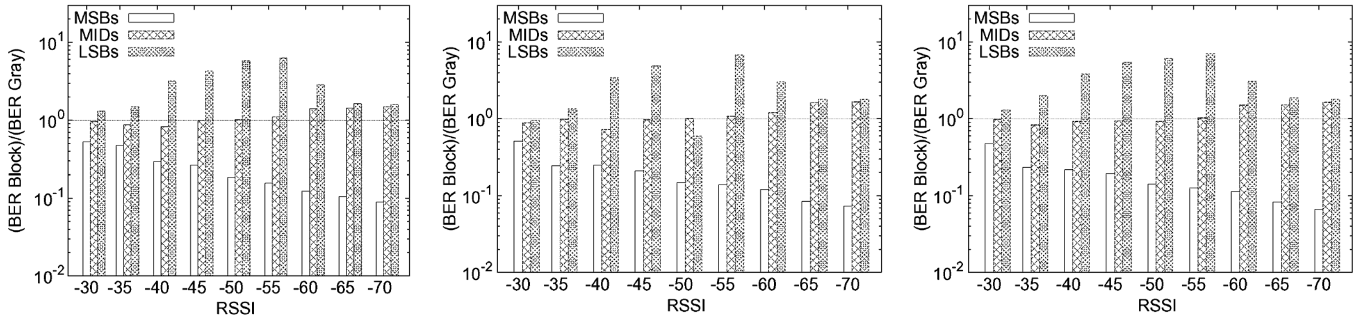
Section III validates that unequal error protection for different bit positions is consistently possible across very diverse wireless channel conditions. In this section, we describe our specific approximate communication scheme that can leverage these specific constellation maps to improve the performance of media applications in a natural way.

In particular, we focus on video streams in which content is partitioned into multiple classes, and each class has a different level of importance in correct decoding of the stream. This is commonly observed in all MPEG-based video streams in which a “Group of Pictures” (GOP) consists of a sequence of I-, P-, and B-frames that have been coded together.

The application (or an application helper) provides a simple estimate of the relative priority value of these different data bits and passes it down the stack. There are many effective ways to compute priority of data bits to an application [22], [32], and in this paper we use a relatively simple one based on decoding dependencies. More specifically, in Apex, the *value* of any data unit i is proportional to the number of data bytes that require i for successful decoding at the receiver. In our case, we use frames within a GOP as a data unit. Hence, all bytes of a given frame type in a GOP has the same value, making it easy to perform the computation once for each GOP. By design, I-frame data is necessary to decode the highest number of dependent frames, and hence they have the highest priority. B-frames have no other dependent except themselves, so they have the least priority. P-frame has a priority that is intermediate of the two other frame types. The relative priorities of the three frame types can change between different GOPs. As mentioned before, better ways of defining the value of different bytes are indeed possible; we choose to use our simple approach, as it is sufficient to illustrate the performance of approximate communication.

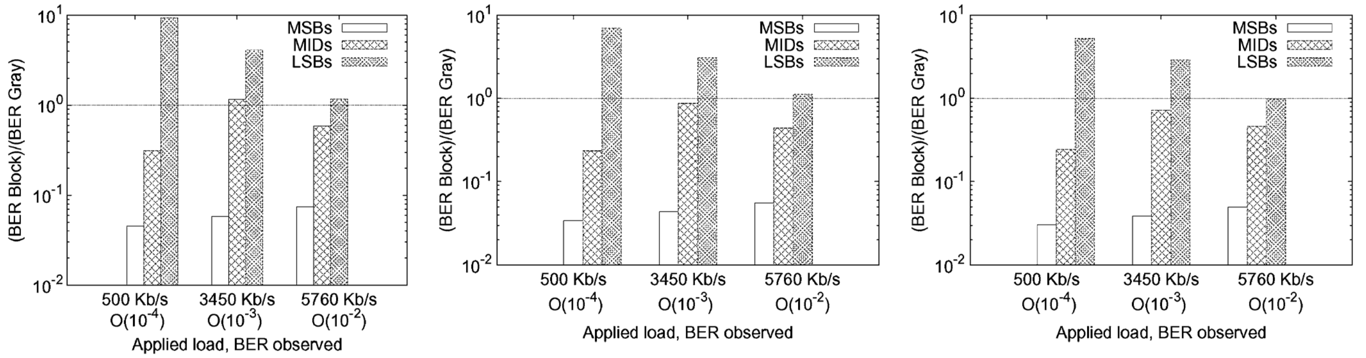
We also note, that our approach generalizes to any other media encoding, where the content is structured in layers, and there are different levels of priority (value) for each layer.

The MAC-PHY layer of Apex is provided with a few different constellation maps. In our case, we use two alternatives—Block and Gray. Based on the application-specified value of different data units, the MAC and PHY layers of Apex makes two simple decisions: 1) given a constellation map, how to place application data bits to bit positions for the desired levels of protection; and



(a) (b) (c)

Fig. 8. Variation in BER gain at various bit positions of Block relative to Gray in presence of an external interferer. Convolution coding at rate 1/2, 2/3, and 3/4. The horizontal line corresponds to the case where both schemes offer equal error protection for a given bit position. (a) With PHY convolution code, Rate $= 1/2$. (b) With PHY convolution code, Rate $= 2/3$. (c) With PHY convolution code, Rate $= 3/4$.



(a) (b) (c)

Fig. 9. Variation in BER gains at different bit positions with Block relative to Gray with varying RSSI, and with PHY convolution coding at rate 1/2, 2/3 and 3/4. The horizontal line corresponds to the case where both schemes offer equal error protection for a given bit position. (a) With PHY convolution code, Rate $= 1/2$. (b) With PHY convolution code, Rate $= 2/3$. (c) With PHY convolution code, Rate $= 3/4$.

2) which specific constellation map to use. We discuss how each of these components is implemented by making small changes to an 802.11-style PHY-MAC subsystem, while retaining all its features.

Placing Application Data Bits to Bit Positions: Each constellation map provides varying degrees of protection for different bit positions of a symbol. However, once a constellation map has been chosen, there is only one optimal way to place application data bits to various protection levels to maximize value of protected data—*greedily*. Given x data bits across all priority levels, and m bits per symbol, the total number of symbols available is x/m. We start by placing the highest-priority bits to the most protected bit positions of each of these symbols. We move over to the next protected bit positions, if we exhaust the most protected bit positions. Once we complete assigning all highest-priority bits, we move to bits at the next priority level. We continue until we exhaust all bits. Unlike a traditional communication system, where each wireless frame carries data bits from a single video frame, *in approximate communication, each typical wireless frame carries data bits from an I-frame, a P-frame, and a B-frame*. The I-frame bits within the wireless frame will be most protected, while the B-frame bits will be least protected. Note that the above greedy approach does not require that the number of protection levels available in different bit positions be equal to the number of desired priority levels in application data, and it just provides a relative ordering in the level of protection for these bits.

Also, the above approach ensures that we do not need a fixed ratio of I-, P-, and B- frame data, as the excess data bits belonging to one priority level can be sent by using all bit positions in a symbol.

Choosing a Constellation Map: As Section III shows, constellation maps differ in the level of protection available at different bit positions. Hence, the choice of a constellation map depends on the relative *utility* of protecting different application bits differently.

In order to evaluate the utility of a constellation map, we need to estimate the BERs at different bit positions for different constellation maps. For this, we add a small number of well-known pilot symbols into each wireless frame (1 pilot byte for every 100 B). The receiver will decode these symbols with the different constellation maps in consideration, e.g., Gray and Block. Since the pilot symbols are known, the receiver can estimate the BERs for different positions and provide this as a feedback in the 802.11-style ACK frame. Note that such BER computation occurs at transmission speeds, as such computation already occurs for wireless frame decoding and ACK generation. We use a single byte to carry each BER value of the MSBs, MIDs, and LSBs of 64-QAM, for a total overhead of 6 additional bytes for both schemes in the ACK frame. (16-QAM and 256-QAM have two and four protection levels, respectively, and the overheads in the ACK frame are 4 and 8 B, respectively.)

With this information, the transmitter can calculate the utility $u(\mathcal{E})$ of a constellation map $\mathcal{E}$ as follows. Let v_i be the *value* associated with application data units at the ith priority level. Let p_j indicate the protection level of the jth significant bit position group, and let x_{ij} be the number of application data units with priority level i allocated to bit positions with protection level j (using the greedy approach). Then, $u(\mathcal{E}) = \sum_i v_i \sum_j p_j x_{ij}$, where v_i is presented as an input from the application and is computed once for each video stream GOP. x_{ij} is obtained from our bit mapping strategy and can also be computed per constellation map, once for each video stream GOP, and p_j is received as feedback in the ACK frame. We pick the constellation map that maximizes this value. We illustrate the benefits of switching constellation maps as part of our evaluation in Section V. This computation itself is quite fast. Hence, in principle, the constellation map can be changed for every frame. However, in our implementation, the decoding process for the different constellation maps happens, not in the field programmable gate array (FPGA) of the WARP radios, but in the associated laptops. This adds latencies to the process such that our constellation map usually changes at a rate of once every 10 packets or more. Our evaluation illustrates the performance advantages of this dynamic choice in spite of this latency. The actual gains of this component in a real system are therefore likely to be even better.

Interaction With Choice of Modulation Scheme and PHY Rate of Transmission: Approximate communication has a direct interaction with the choice of data modulation scheme and the PHY-layer transmission rate of data units. In general, any good rate and modulation selection scheme will ensure that the approximation property of symbol errors holds. In our work, we implement a rate selection scheme that is based on the SoftRate algorithm [34], which is known to be fairly agile and accurate. Each change in the rate triggers an immediate reevaluation of the specific constellation map, although the constellation map might sometimes change at an even faster rate if necessary due to change in channel conditions. We anticipate that the performance of the system can be improved even further if the rate selection decision is combined with the constellation map selection. In this paper, we do not explore this joint problem further, and relegate it as part of our future work.

Apart from this data bit mapping process, the rest of the MAC layer in Apex works as per IEEE 802.11 protocol. In particular, it makes the system's external properties completely compatible and compliant to the IEEE 802.11 standard.

Modifications to PHY PLCP Header and Payload: A few modifications need to be made to the PHY-layer PLCP header. First, we added a two-bit constellation map selector within the PLCP header to inform the receiver which encoding scheme is used in the wireless frame. This limits the number of constellation map alternatives to four, which we believe should be sufficient for most applications. We also added information on how the data of different priority levels are placed into various bit positions using our greedy approach. This can be simply expressed by indicating the number of bytes (expressed in multiples of eight bytes) in each level (the greedy algorithm for placing data bits to bit positions can be used to partition the transmitted data). We limit ourselves to four priority levels for application data currently. We add a 1-B field in the PLCP header for each level, for a total of 4 additional bytes. This limits the maximum payload size to 2048 B.

Note that only the two-bit constellation map selector is placed in the early part of the PLCP header to be transmitted at the base data rate, e.g., 802.11a uses BPSK with 1/2 PHY convolution codes, and using a predefined constellation map (Gray). The four priority-level size fields in the latter part of the PLCP header can be transmitted at higher data rates like the rest of the frame. Hence, although the PLCP header increases from 40 to 72 bits, only 2 of these additional bits need to be transmitted at the base rate. Finally, we add the pilot symbols through the data field for BER estimation of different bit positions with different constellation maps. This is presented in Fig. 10.

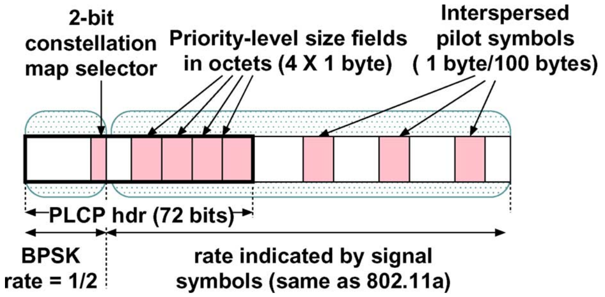

2-bit constellation map selector
Priority-level size fields in octets (4 X 1 byte)
Interspersed pilot symbols (1 byte/100 bytes)
PLCP hdr (72 bits)
BPSK rate = 1/2
rate indicated by signal symbols (same as 802.11a)

Fig. 10. Modifications to PHY PLCP header and payload for Apex.

Modifications to the Digital Component of the PHY-Layer Pipeline: We describe modifications needed to a standard 802.11 PHY-layer pipeline to implement all features in Apex and present it pictorially in Fig. 11. All components in the figure are from the 802.11a reference pipeline, and the shaded parts of the figure indicate locations where some changes are needed.

In the transmitter, data is available to the PHY pipeline in multiple queues, one for each priority level present. Our system allows for four priority levels, and in the figure we show only three queues populated with data (corresponding to I-, P-, and B-frame data). In a regular 802.11a pipeline, all data coming into the controller will be a single queue, and for a regular-sized Ethernet-style frame, may have a maximum payload of 1500 B. In our case, we expect the total bytes corresponding to a single wireless frame still to be 1500 B, but this data may consist of a different number of bytes from I, P, and B queues. Each of these I, P, and B fragments will pass through the scrambler and the convolution encoder *independently* like in a regular 802.11 pipeline and arrive at the interleaver. Note that, at the interleaver, the scrambled and the convolutionally encoded data are still separated according their priority levels and are placed in separate queues.

In the interleaver, the bits will be placed into bit positions based on the greedy algorithm. Each bit sequence, thus generated, will correspond to a symbol and will likely consist of bits collected from the three different queues. Like a regular 802.11 interleaver, we will allow interleaving of bit values across different OFDM subcarriers (if multiple OFDM subcarriers are used). However, unlike a regular interleaver, we will only limit interleaving of bits within a single subcarrier to within the bit positions allocated to data from the same priority class.

Finally, based on the application data value and the BER feedback from the receiver, one of the data constellation maps will be selected. The bit sequences will be mapped to the corresponding symbol values using a simple table lookup process as in a regular pipeline and handed over the RF front end.

A similar, but reverse process would occur in the receive chain.

As part of our implementation, we have modified *the WARP OFDM Reference Design v14.1*, which is an FPGA-based

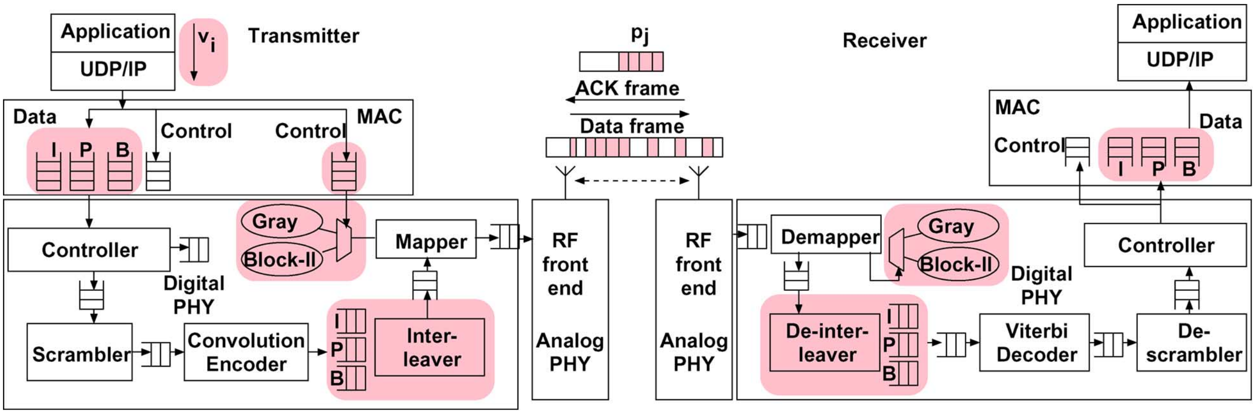


Fig. 11. Changes to basic 802.11 layers for approximate communication in Apex. I, P, and B labels are for illustration purpose only, indicating three arbitrary priority layers. Shaded parts are modifications made.

implementation of WiFi like wireless PHY layer. The implemented PHY runs on a Xilinx Virtex-II Pro FPGA that has a PowerPC processor, a 36-Mb on-chip SRAM, and programmable slices. We implemented the application, UDP, and IP layer of the video application on the desktop while we built the MAC layer on the PowerPC processor and the PHY layer in programmable slices on the FPGA.

In the PHY layer, a read controller is introduced to generate read signals for the packet buffer. The reading is performed in a way to ensure that bytes from different types of frames are loaded in an *alternate* fashion. Each loaded byte is passed through the unmodified scrambler, FEC encoder, and interleaver. After the interleaver, a 3-B shift register is introduced for each type of frames to store the interleaved bytes of the corresponding type. After the shift registers, we added a logic component to merge bits from different shift registers, e.g., 2 bits from each register for 64-QAM and 1 or 2 bits for 16-QAM. The merge logic retrieves a different number of bits from each type of frames and performs merging based on the placement field in the header. The merge logic passes the merged bit sequences to the QAM mapper. Finally, the QAM mapper maps bit sequences to QAM symbols based on the constellation indicated by the header. Finally, the generated QAM symbols are sent to the analog front end for transmission.

As some miscellaneous details, the frame register discards the bits read by the merge logic. It informs the read controller to load more bytes whenever bits stored in the register are about to be exhausted. The read controller coordinates the timing of reads to ensure that at most one read operation per clock cycle on the packet buffer. This is because the SRAM on our platform does not support multiple simultaneous reads. We have implemented the entire system mentioned above, except for the dynamic constellation switching, which is left as future work.

Overhead: We characterize the overhead of the modified MAC and PHY in terms of *latency* and *hardware cost*. The extra latency incurred by our implementation is 9 clock cycles, which is negligible (~ 0.225 μs) compared to the round-trip time (RTT) delay (698 μs for 16-QAM modulation). The latency comes from the necessity to preload bytes for each shift register before the merging process starts. Note that the merge logic is implemented as a combinational logic with no extra delay. The overhead in terms of added hardware components is also minimal because different types of frames *share* the same main datapath at the PHY layer. Moreover, only a few functional blocks are added to the original design, the majority of which are small combinational logics.

Coexistence With Legacy Devices: To coexist with legacy devices, we envision the use of one reserved bit in the service field of the PLCP header to distinguish between Apex communication and traditional communication. The Apex system uses the same retransmission mechanism and channel contention protocol specified in IEEE 802.11 standard. This guarantees fairness to legacy devices in channel utilization. In fact, the spectrum savings from Apex will benefit all the devices sharing the medium.

V. Evaluation

We now evaluate the performance benefits offered by Apex in detail. In addition, we also show that the advantages of approximate communication are complementary to some alternate mechanisms of adding unequal error protection, e.g., application-layer FECs, and data retransmissions.

A. Experiment Configurations

We describe the new aspects of our experimental configuration vis-a-vis Section III for evaluating performance of video streaming in Apex. We use the same WARP SDR platform, with the signal processing functions implemented in laptops connected to these radios.

Video Content: We use Claire and Foreman video clips, encoded to MPEG-4 format using *ffmpeg* tool. The rate of encoded video is 384 kb/s, frame rate is 30 frames per second, and each of its GOPs consists of 30 frames. We induce a buffering of 1 s to collect all the frames in a GOP. The video is looped multiple times to get a playback length of 500 s. Each experiment is repeated 20 times unless otherwise stated. We use Evalvid tool [4] to stream the video. The tool has been modified to provide information about frame type and priority while streaming.

Metrics: We have compared the quality of the received video both visually and using a widely used metric—PSNR. The PSNR of a video is well correlated with the perceived quality of video experienced by the user. The relationship between user perception expressed in mean opinion score (MOS) and the

TABLE III
MOS-BASED USER PERCEPTION OF VIDEO QUALITY MAPPED TO THE PSNR RANGE [25]

MOS Rating of video quality	PSNR range
Excellent	> 37
Good	31-37
Fair	25-31
Poor	20-25
Bad	< 20

PSNR range were detailed in [17] and [25] and are summarized in Table III.

Schemes Compared: We compare the performance of an approximate communication (Apex) and a traditional communication (Trad.) system (which is unaware of the relative protection levels of the different bit positions) when both are allowed to transmit the same amount of application data. The traditional scheme as well as Apex use the partial packet recovery method, as proposed in [18].

We compare the relative volume of retransmissions needed for the Apex and traditional systems to achieve similar PSNR. We also measure the additive gains of Apex in presence of application-layer FEC.

Apex is complementary to other data protection mechanisms that can be implemented at the different layers, e.g., MAC-layer mechanisms (ZipTx and MRD) and PHY-MAC mechanisms (PPR and SOFT). It was challenging for us to implement all of these schemes within the experimental WARP SDR platform. In Section VI, we present an intuitive explanation of why other schemes—PPR, SOFT, MRD, and ZipTx—are also expected to provide gains that are complementary to Apex.

Trace-Based Evaluation for Dynamic Rate Adaptation Scenarios: In our SDR platform (WARP radios), a high latency is associated with processing RF samples from a received wireless frame and sending feedback to the transmitter for efficient rate adaptation. To avoid this latency-related inaccuracy in evaluation, we adopt the practice of trace-driven analysis as is commonly applied in evaluating rate adaptation algorithms with PHY-layer processing needs [34].

Symbols carrying application data are sent at different rates in succession. We iteratively send small packets of 200 B at all the different rates for building the trace. Also, we ensure that the BER is monotonically increasing with increasing data rate for a iteration (discarding samples for which this criteria does not hold) signifying a coherent channel [34]. At the receiver, we calculate the achieved throughput by different data rates, and only those symbols that correspond to the rate selected by the rate adaptation algorithm are considered to be part of the media flow. As the selected rate varies over time, the symbols selected to be part of the flow at different time instants are picked accordingly. The actual performance of the media delivery process can then be evaluated by identifying loss behaviors due to errors in an offline manner.

B. Experimental Results

In this section, we show how Apex helps improve the media quality under different interference scenarios, show how the gains of Apex are significant when compared to FEC-based or retransmission-based data recovery schemes, and describe some parameter selections made in the system.

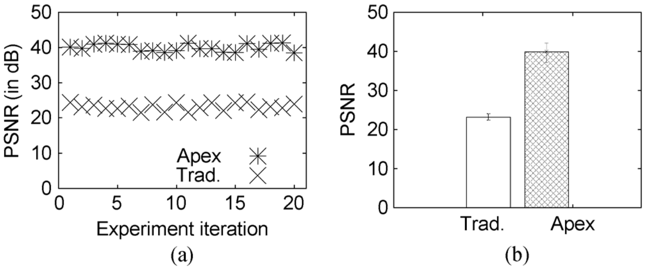


Fig. 12. PSNR differences between traditional and approximate communication with rate adaptation at a representative wireless environment (Claire video). (a) PSNR of each iteration. (b) Average and standard deviation.

TABLE IV
PSNR FOR THE TWO VIDEO CLIPS IN THE REPRESENTATIVE WIRELESS SCENARIO WITH STATICALLY CHOSEN CONSTELLATION MAP

Video	**Trad.**	**Apex (Gray)**	**Apex (Block)**
Claire	21.8	29.5	**37.8**
Foreman	21.0	**34.5**	31.0

Apex Versus Traditional—A Representative Scenario: We present the relative performance for traditional media delivery and the Apex scheme with dynamic rate and constellation map adaptation enabled for the same channel conditions as above in Fig. 12. The average uncoded BER for this scenario was 4.1×10^{-3} (corresponds to a coded BER of $O(10^{-5})$, which is normal operating condition for 802.11-based radios). The average PSNR improvement due to Apex across a set of 20 runs was about 16 dB. The better performance of Apex stems from the fact that a throughput optimal data delivery can be further improved upon by the importance-aware mapping of data bits to bit positions within symbols.

We next illustrate how the dynamic adaptation of constellation map is useful to improving the performance of Apex.

Impact of Constellation Map Selection: The best choice of constellation map depends both on the channel conditions (p_j values) and the differential value of application bit (v_i values) channel conditions. We demonstrate this in turn.

—*Dependence on application data:* For the same representative scenario as in Fig. 12, we show the relative performance of traditional and Apex for two different video clips, Claire and Foreman, in Table IV, but with statically chosen constellation maps for the entire duration of the clip. Apex outperforms the traditional system in all cases. However, the relative amount of improvement depends on the specific constellation map selection. Due to the distribution of bytes between I-, P-, and B-frames in these clips, the relative values of different data bits are different in these two schemes. As a consequence, Gray code leads to a better average PSNR performance for the Foreman video than Block code, while the reverse is true for the Claire video. By dynamically selecting the most appropriate constellation map, Apex would be able to deliver the best performance among various alternatives.

—*Dependence on Channel Conditions:* We next illustrate how the dynamic adaptation of the constellation map occurs in practice, also due to changing channel conditions. Fig. 13 shows how the dynamic choice of the data encoding scheme ensures that the best encoding scheme is picked as the quality of the channel changes. In particular,

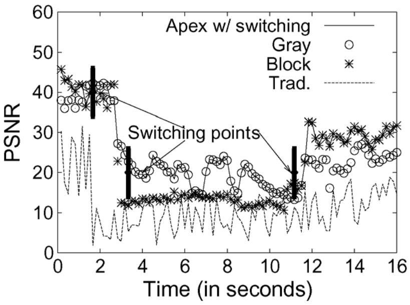


Fig. 13. Dynamic switching of data encoding scheme computed based on the utility of each encoding alternative (block and gray).

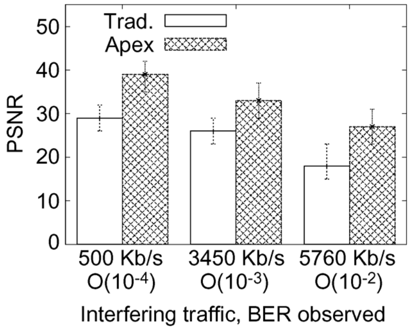


Fig. 14. PSNR for traditional and approximate communication under varying external interference (Claire video). x-axis marks both the imposed (802.11) interference load on the channel and the corresponding average uncoded BER.

our algorithm decides to switch constellation maps around time 1.8, 3.3, and 10.9 s, soon after the PSNR quality using the constellation map falls below the other alternative. We also show the performance of a traditional communication system in the figure for the sake of completeness. We find that Apex performs better than traditional communication at all times.

Currently, our feedback latencies are significantly higher since the decoding operation with different encoding schemes is performed in the associated laptop, incurring high latencies, although they are adequate to provide performance gains. A future version of the system will implement the feedback process in an FPGA, leading to a more agile response and potential performance gain.

With External Interference: We next experimented by changing the level of external interference generated by a third (laptop-based) node in the vicinity, as described in Section III. As expected, both schemes suffer with increasing interference. However as the channel condition worsens, due to increasing interference, the relative gains of Apex over traditional communication increases (Fig. 14). At the higher interference levels, the PSNR in Apex is about 1.8 times that of traditional communication. The higher gains of Apex can be attributed to informed placement of application bits to bit positions of symbols.

Complementarity to Application-Level FEC: We show that gains of Apex communication are complementary to application-level FECs through evaluation of traditional and approximate communication, both with and without such FEC. We used a Reed–Solomon RS(255,233) code for application-layer FEC, where the amount of coding was determined based on ambient channel conditions as suggested in related efforts [9], [21]. In Fig. 15, we show the performance comparison for a scenario where traditional communication achieved the highest possible PSNR. The plot shows that Apex achieves better performance than traditional communication both in the presence and absence of application-layer FECs, and the gap in PSNR between the two approaches is somewhat similar in both cases.

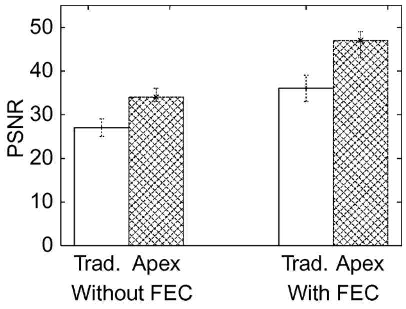


Fig. 15. PSNR of video with and without application-level FEC (Claire video).

Comparison to a MAC-Layer Scheme: Consider a MAC-layer variant of Apex as follows. Given an optimal PHY rate discovered by a rate adaptation algorithm, in this variant we assume that I-frames are sent at a rate that is one lower (to protect it better from errors), and the P- and B-frames are sent at some higher rate (reflecting their lower priority). A recently proposed work for video streaming [29] uses a similar MAC-layer rate adaptation strategy, albeit in context of multicast communication. To show that the PHY-layer implementation of Apex is the most efficient version of this idea, we compared Apex to this variant of MAC-layer rate adaptation. In our experiments, we found that Apex outperformed this MAC-layer variant by at least 4 dB or more in different experiments, primarily owing to the fact that the former was able to achieve differential data prioritization at a finer granularity.

We next augmented the MAC-layer scheme even further by allowing it to send an increasing number of retransmissions until it was able to match the PSNR of Apex (without any retransmissions). We found that the number of symbols required by the MAC-layer scheme is 1.85× that of Apex to achieve the same PSNR performance (when the uncoded BER was 4.1×10^{-3}). As the channel condition became more error-prone, the gains of Apex increased even further. This result prompted us to study how Apex compares to various retransmission strategies next.

Apex Versus Retransmission Strategies: One other way to compute the gains of Apex is to compute how much retransmission traffic it can save. To characterize this, we let the traditional system to retransmit each lost symbol multiple times. We bound the retransmission limit of the traditional system to up to seven retransmission attempts (as is common in 802.11 system). In Fig. 16, we show the PSNR of the two schemes (Apex with no retransmissions and traditional with one, two, or three retransmissions). At the limit of three retransmissions, the traditional scheme achieved the same PSNR as Apex. It is more instructive to see the relative bandwidth consumed by Apex with no retransmissions when compared to traditional with a three-retransmission limit for this wireless scenario. Fig. 17 shows the increased volume of traffic (including retransmissions) for traditional communication compared to the number of transmissions in Apex. Note that both schemes now achieve the same

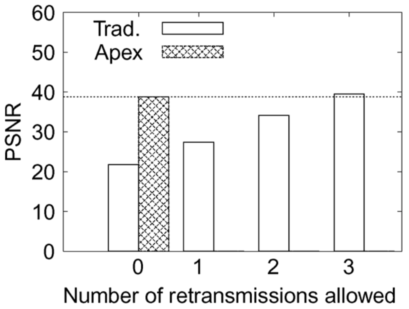


Fig. 16. Variation in PSNR for traditional communication (with different number of retransmissions) and Apex (without retransmissions).

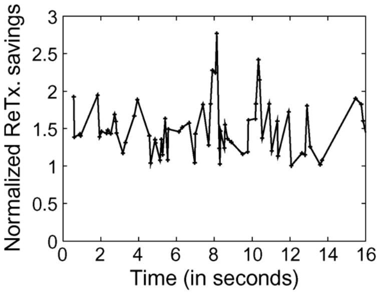


Fig. 17. Saved retransmissions in Apex. y-axis shows the normalized number of bytes transmitted by traditional communication (three retransmissions) with respect to Apex.

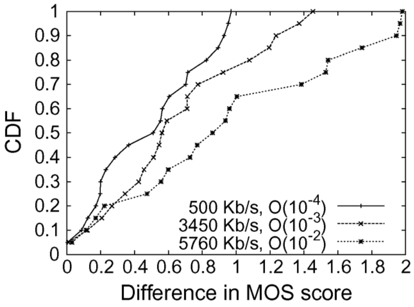


Fig. 18. Difference in MOS score calculated using PESQ, between traditional and approximate communication in presence of interference.

TABLE V

ERROR OF BER ESTIMATION FOR DIFFERENT BIT POSITIONS IN 64-QAM, BLOCK CODING, WITH VARYING NUMBER OF PILOT SYMBOLS. WE USE ONE PILOT SYMBOL FOR EACH 100 SYMBOLS IN APEX. EXPERIMENTS USED 1500-B PACKETS. MEAN (STANDARD DEVIATION) OF PREDICTION ERROR IS PRESENTED

Pilot symbols	Error in estimating BER in MSBs	in MIDs	in LSBs
1 in 25	0.05 (0.03)	0.05 (0.03)	0.04 (0.02)
1 in 50	0.06 (0.04)	0.05 (0.04)	0.05 (0.03)
1 in 100	**0.08 (0.06)**	**0.07 (0.05)**	**0.06 (0.05)**
1 in 150	0.12 (0.08)	0.11 (0.13)	0.09 (0.07)
1 in 200	0.14 (0.11)	0.16 (0.07)	0.10 (0.08)

PSNR for the video, but as the channel deteriorates from time to time, the bandwidth requirements of traditional communication sharply increases.

Audio on Apex: We have characterized the performance improvements offered by Apex for audio as well. For this, we generated a compressed encoding of an audio clip using WavPack [5]. The WavPack tool generates two output files, one corresponding to the audio frequencies that are important for human perception (the frequencies in the middle of the auditory range) and another with information about frequencies not perceived properly (frequencies at the fringe of auditory range) by human ears. We map the bits corresponding to the first set on more error-resilient bit positions, while the bits corresponding to the second set to more error-prone bit positions. In Fig. 18, we present the cumulative distribution function (CDF) of improvement in MOS score that Apex offers over traditional communication as the amount of interference is varied. We find that Apex can improve the mean opinion score calculated using perceptual evaluation of speech quality (PESQ) of the received audio by 0.5–1 unit (on a scale of 0–5) in presence of interference. Research has shown that an MOS reduction of only 0.1 is noticeable by the human ear [26]. In particular, we find that with increasing interference, the gap in performance between Apex and traditional communication increases.

Parameter Selection—Number of Pilot Symbols Needed: One of the parameters for the Apex system is the number of pilot symbols used per frame (1 in 100). We made this choice through a sequence of experiments, where we varied the number of pilot symbols in a 1500-B packets and observed the accuracy of predicting the BERs in different bit positions. The greater the number of pilot symbols, the greater is the overhead of the scheme, but the higher is the likelihood of accurate estimation. Table V shows this variation, where error is defined as $(\text{Actual BER} - \text{Estimated BER})/\text{Actual BER}$. To balance the tradeoff between overheads and accuracy, we decided to use one pilot symbol for every 100 symbols of data, i.e., an overhead of 1% due to Apex BER estimation process.

VI. RELATED WORK

We partition prior related efforts into the following categories.

PHY-MAC Approaches: In recent years, a growing number of efforts have designed and studied wireless communication techniques that utilize mechanisms at the PHY and MAC layers to recover more bits out of erroneous packets, e.g., PPR [18], ZipTx [21], SOFT [36], and Maranello [13]. While these schemes apply generally to all data, the transmitters in such schemes do not exploit the unique approximation properties of wireless errors. In contrast, transmitters in Apex learn about and utilize the structure in wireless symbol errors to improve the performance of media delivery applications.

SoftCast [17] is a recently proposed, related scheme that improves wireless media delivery performance in multicast settings by using properties of wireless errors in a manner similar to Apex. SoftCast represents data values in a "raw" numerical (analog) format and maps them directly to wireless symbols with specific transmit power levels. However, there are three main differences between the two schemes.

1) SoftCast requires that media content be represented and transmitted in the raw numerical format, different from what is used in popular standards, e.g., H.264. Media represented in these popular formats have to be converted into the SoftCast format (through a computationally intensive process) to achieve its performance gains. In contrast, Apex does not attempt to define any new media representation format and can utilize any media format, including the one suggested by SoftCast.

2) SoftCast uses a static data representation format that is agnostic of wireless channel conditions. In Apex, our mappings between bit sequences to symbols, and between data bits to bit positions, are dynamically altered based on channel conditions.
3) SoftCast is designed specifically for multicast traffic in which no receiver feedback is assumed. In contrast, Apex focuses on unicast traffic, where immediate MAC-layer acknowledgments are used to dynamically fine-tune the mappings of bit sequences to symbols, and data bits to bit positions for improved performance. Such dynamic adjustments are precluded in SoftCast. Hence, SoftCast and Apex are applicable to video delivery in vastly different scenarios.

Network-Layer Approaches: Some recent research efforts, e.g., MIXIT, have shown knowledge of correct or incorrect symbol decoding from the PHY layer can be leveraged in multihop wireless settings to better recover data on opportunistic end-to-end paths [19]. Approaches such as MIXIT operate on a network-wide scale by leveraging diversity across multiple nodes. However, such a scheme also discards symbols received in error, unlike our approach. Furthermore, our scheme is designed for a single wireless hop.

Transport-Layer Approaches: Transport layer protocols such as UDPlite [20] have been designed to allow applications to accept corrupted data values to recover some content from them. Several notable efforts [31] have shown that under certain network conditions (e.g., high latency), this leads to a better media streaming performance. Again, such schemes complement our approach.

Application-Layer Approaches: There have been several research efforts to improve the quality of streaming content, mainly at the application layer through effective data prioritization. The primary technique in this regard is addition of FEC to different data components to make them more resilient than others. A good survey of such schemes is presented in [35]. Other researchers have also studied hybrids of ARQ-based and FEC-based mechanisms for loss recovery [11], [28]. Authors in [14] and [37] propose specific architectures for such cross-layered systems. These techniques do not take advantage of the natural properties of the wireless channel, which we effectively exploit in Apex. As demonstrated in Fig. 15, such an approach is complementary to Apex since the latter leverages the inherent error approximation property of wireless channel, while the former does not.

Prior theoretical studies [2], [10], [16], [23], [27], [33] have explored the existence of differential error properties at bit positions for individual constellation maps. Our work augments such theoretical findings with design of constellation map switching, data bit placement strategies, and radio RF pipeline modifications and evaluates various nuances through practical implementation on real wireless hardware and experimentation and for real media applications.

Finally, other recent work [8], [38] highlights benefits of exploiting differential error resilience of bit positions for video delivery in unicast and broadcast settings based on numerical simulations. In particular, authors in [38] propose some modifications to DVB-H [1] standard to leverage differential error resilience for terrestrial broadcast of video. In contrast, Apex explores the notion of approximate communication through a full system implementation and extensive experimentation.

VII. Conclusion and Future Work

Our efforts in this paper have focused on demonstrating that approximate communication is indeed a promising idea. Our results on the WARP SDR platform show that our approximate communication system provides significant improvement in video quality (ranging from 5 to 20 dB in different scenarios). However, this is a small first step in realizing the full capability of this system. We believe this work leaves open a few optimization problems that should lead to further performance gains. For example, a joint construction of data modulation schemes and constellation map selection might lead to further performance gains. In particular, since our goal is unequal protection of bit positions, modulation schemes in which constellation points are unequally spaced might be useful to explore.

References

[1] DVB, Geneva, Switzerland, "Hierarchical modulation in DVB-T," 2000 [Online]. Available: http://www.dvb.org/technology/fact_sheets/WP01_DVB-T%20Hierarchical.pdf

[2] "Hierarchical QAM modulation," 2011 [Online]. Available: http://en.wikipedia.org/wiki/Hierarchical_modulation

[3] "MPEG-4, Moving Picture Experts Group-4 standard," 2002 [Online]. Available: http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm

[4] TU Berlin, Berlin, Germany, "Video quality evaluation tool-set," 2012 [Online]. Available: http://www.tkn.tu-berlin.de/research/evalvid/

[5] "WavPack audio compression," [Online]. Available: http://www.wavpack.com/

[6] Rice University, Houston, TX, "Wireless open-access research platform (WARP)," [Online]. Available: http://warp.rice.edu/

[7] Rice University, Houston, TX, "Wireless open-access research platform (WARP) public forum," [Online]. Available: http://warp.rice.edu/forums

[8] A. Abdurrhman and M. Woodward, "Unequal error protection of H.264/Avc video using hiererchical QAM," in *Proc. Telekomun. Forum TELFOR*, 2008, pp. 1–6.

[9] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," in *Proc. ACM SIGCOMM*, 1998, pp. 56–67.

[10] A. R. Calderbank and N. Seshadri, "Multilevel codes for unequal error protection," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1234–1248, Jul. 1993.

[11] N. Feamster and H. Balakrishnan, "Packet loss recovery for streaming video," in *Proc. Packet Video Workshop*, 2002.

[12] F. Gray, "Pulse code communication," U.S. Patent 2,632,058, 1947.

[13] B. Han, A. Schulman, F. Gringoli, N. Spring, B. Bhattacharjee, L. Nava, L. Ji, S. Lee, and R. Miller, "Maranello: Practical partial packet recovery for 802.11," in *Proc. NSDI*, 2010, p. 14.

[14] W. Heinzelman, "Application specific protocol architectures for wireless networks," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2000.

[15] U. Horn, K. Stuhmuller, M. Link, and B. Girod, "Robust Internet video transmission based on scalable coding and unequal error protection," *Signal Process., Image Commun.*, vol. 15, pp. 77–94, 1999.

[16] M. Isaka, M. P. C. Fossorier, R. H. Morelos-Zaragoza, S. Lin, and H. Imai, "Multilevel coded modulation for unequal error protection and multistage decoding. II: Asymmetric constellations," *IEEE Trans. Commun.*, vol. 48, no. 5, pp. 774–786, May 1999.

[17] S. Jakubczak, H. Rabul, and D. Katabi, "SoftCast: One video to serve all wireless receivers," MIT-CSAIL-TR-2009-005, 2009.

[18] K. Jamieson and H. Balakrishnan, "PPR: Partial packet recovery for wireless networks," in *Proc. SIGCOMM*, 2007, pp. 409–420.

[19] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," in *Proc. ACM SIGCOMM*, 2008, pp. 401–412.

[20] L. Larzon and M. Degermark, "The lightweight user datagram protocol (UDP-Lite)," 2004 [Online]. Available: http://www.ietf.org/rfc/rfc3828.txt

[21] K. C.-J. Lin, N. Kushman, and D. Katabi, "ZipTx: Exploiting the gap between bit errors and packet loss," in *Proc. ACM MobiCom*, 2008, pp. 351–362.

[22] A. Majumdar, D. Sachs, I. Kozintsev, K. Ramchandran, and M. Yeung, "Multicast and unicast real-time video streaming over wireless LANs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 524–534, Jun. 2002.

[23] R. H. Morelos-Zaragoza, M. P. C. Fossorier, S. Lin, and H. Imai, "Multilevel coded modulation for unequal error protection and multistage decoding. I: Symmetric constellations," *IEEE Trans. Commun.*, vol. 48, no. 2, pp. 204–213, Feb. 1999.

[24] T. Nguyen and A. Zakhor, "Distributed video streaming with forward error correction," in *Proc. Packet Video Workshop*, 2002.

[25] Z. Orlov, "Network-driven adaptive video streaming in wireless environments," in *Proc. IEEE PIMRC*, 2008, pp. 1–6.

[26] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ)—A new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2001, vol. 2, pp. 749–752.

[27] M. Sajadieh, F. Kschischang, and A. Leon-Garcia, "Modulation-assisted unequal error protection over the fading channel," *IEEE Trans. Veh. Technol.*, vol. 47, no. 3, pp. 900–908, Aug. 1998.

[28] H. Seferoglu, Y. Altunbasak, O. Gurbuz, and O. Ercetin, "Rate-distortion optimized joint ARQ-FEC scheme for real-time wireless multimedia," in *Proc. IEEE ICC*, 2005, vol. 2, pp. 1190–1194.

[29] S. Sen, N. K. Madabushi, and S. Banerjee, "Scalable WiFi media delivery through adaptive broadcasts," in *Proc. NSDI*, 2010, p. 13.

[30] S. Sen, S. Schmitt, M. Donahue, and S. Banerjee, "Exploiting 'approximate communication' for mobile media applications," in *Proc. HotMobile*, 2009, Article no. 11.

[31] A. Singh, A. Konrad, and A. D. Joseph, "Performance evaluation of UDP lite for cellular video," in *Proc. NOSSDAV*, 2001, pp. 117–124.

[32] D. Tian, X. Li, G. Al-regib, Y. Altunbasak, and J. R. Jackson, "Optimal packet scheduling for wireless video streaming with error-prone feedback," in *Proc. IEEE WCNC*, 2004, vol. 2, pp. 1287–1292.

[33] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 1, pp. 55–67, Jan. 1982.

[34] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," in *Proc. ACM SIGCOMM*, Barcelona, Spain, Aug. 2009, pp. 3–14.

[35] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," *Proc. IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.

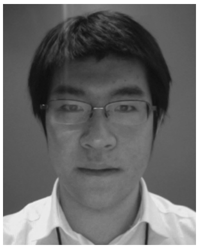
[36] G. R. Woo, P. Kheradpour, D. Shen, and D. Katabi, "Beyond the bits: Cooperative packet recovery using physical layer information," in *Proc. MobiCom*, 2007, pp. 147–158.

[37] S. Worrall, S. Fabri, A. Sadka, and A. Kondoz, "Prioritisation of data partitioned MPEG-4 video over mobile networks," *Eur. Trans. Telecommun.*, vol. 12, pp. 169–174, 2001.

[38] Z. Wu, J. Boyce, and A. Stein, "An unequal error protection framework for DVB-H and its application to video streaming," in *Proc. IEEE GLOBECOM*, 2008, pp. 1–6.

Sayandeep Sen received the B.Tech. degree in computer science and engineering from the University of Kalyani, Kalyani, India, in 2004, and the M.Tech. degree in computer science and engineering from the Indian Institute of Technology, Kanpur, India, in 2006, and is currently pursuing the Ph.D. degree in computer science at the University of Wisconsin–Madison.

His research interests include designing protocol and systems for wireless networking.



Tan Zhang received the B.S. degree in electrical engineering from the Beijing Institute of Technology, Beijing, China, in 2009, and is currently pursuing the Ph.D. degree in computer sciences at the University of Wisconsin–Madison.

His research interests include designing systems and protocols for dynamic spectrum access in wireless networks.

Syed Gilani received the B.Sc. degree in electrical engineering from the University of Engineering and Technology (UET), Lahore, Pakistan, in 2006, and the M.S. degree in computer engineering from the University of Wisconsin–Madison in 2009, and is currently pursuing the Ph.D. degree in computer engineering at the University of Wisconsin–Madison.

His research interests include low-power and energy-efficient processor architectures, embedded systems, and computer arithmetic.

Mr. Gilani is a Fulbright Fellow at the University of Wisconsin–Madison.

Shreesha Srinath received the M.S. degree in electrical and computer engineering from the University of Wisconsin–Madison in 2010.

Suman Banerjee (M'05) received the B.Tech. degree in computer science and engineering from the Indian Institute of Technoogy (IIT), Kanpur, India, in 1996, and the M.S. and Ph.D. degrees in computer science from the University of Maryland, College Park, in 1999 and 2003, respectively.

He is an Associate Professor in computer sciences with the University of Wisconsin–Madison, where he is the Founding Director of the WiNGS Laboratory, which broadly focuses on research in wireless and mobile networking systems.

Dr. Banerjee is a recipient of the NSF Career Award. He is also the recipient of the ACM MobiCom 2009 Best Paper Award. Furthermore, technology he developed has won various accolades, including the grand prize at the Wisconsin Governor's Business Plan Competition in 2011 and the second prize in the Interdigital Innovation Challenge in 2012.

Sateesh Addepalli is Executive Director of Research, Innovations, New Ventures and Business Initiatives with Cisco Systems, Inc., San Jose, CA. He is a world-class business and technology leader with over 20 years of experience. Currently, he is driving various programs in the area of IoE/IoT/M2M, smart cities and communities, big data, edge/fog/cloud computing, wireless and mobile, among others.