# Exploiting "Approximate Communication" for Mobile Media Applications

Sayandeep Sen, Stephen Schmitt, Mason Donahue, Suman Banerjee
University of Wisconsin, Madison, WI 53706, USA

## ABSTRACT

Errors are integral part of every communication system, whether wired or wireless. There are two broad approaches to deal with errors: (i) detection and discard of data elements in error and (ii) optional recovery from such errors either through proactive or reactive (re)-transmissions. Both these approaches assume that errors are binary in nature, i.e., an error in transmission implies a need to discard or recover the erroneous bits. In this paper, we consider an intriguing alternative, one in which data elements in error are accepted as "approximately correct" values. We call this approximate communication. More specifically, we introduce the notion that data elements being received are not just correct or incorrect. Instead, there exists a degree of correctness in the received data elements that can be effectively exploited by certain classes of popular applications operating across mobile communication systems.

## 1. INTRODUCTION

Traditionally any data transferred over a communication channel, whether an end-to-end Internet path, or for a mobile device across a single-hop wireless link, has been expressed as a sequence of application-layer bits. The application-layer data bits get successively encapsulated in data units of different layers, including the MAC layer. Eventually these bits gets modulated onto a physical carrier and transmitted across a physical channel. On receiving this physical signal, the receiver demodulates and decodes back the MAC layer bits and makes them available to higher layers for further processing.

To preserve the layered abstraction, lower layers typically do not interpret the bits presented to them by the upper layers. In a sense, the goal of the physical channel has been to transport these bits in a perfect manner. For example, if the transmitter sends a bit sequence 111100 and the receiver decodes the physical signal to obtain a bit sequence of 111111, then we consider this sequence to have two bit errors.


*HotMobile 2009*, February 23-24, 2009, Santa Cruz, CA, USA.


Wireless receivers have mechanisms to detect, and sometimes, correct such errors. In general, when bit errors are detected, the higher layers are alerted. In some cases, such higher layers may choose to re-transmit these bits in error. This allows the receiver to acquire the exact sequence of transmitted bits for its higher layer protocol. Alternatively, the receiver may choose to silently discard erroneous bits (or frames) and make the application deal with loss of such information.

In this paper, we explore a third alternative in which we may choose to treat the received bit sequence as a "good approximation" of the transmitted bit sequence, thereby extracting useful value out of the received sequence. We explain this through the following example.

Imagine the intent of the transmitter is to communicate an integer value of 60, represented by the bit sequence 111100. Due to noisy channel conditions, the receiver decodes this sequence as 111111, i.e., as the integer value of 63. By accepting this value to be correct, the receiver's application will make an error of 3/60, i.e., 5%. In comparison to discarding the entire integer value due to perceived bit errors, accepting this slightly distorted value may be more useful to certain applications, especially media streaming applications. We refer to this form of communication, where certain amount of errors are acceptable, *approximate communication.* This is in contrast to the traditional design of exact communication systems, where only correctly received bits are delivered to the upper layers.

While approximate communication appears to be intriguing, it is useful, only if the received bit sequence provides a good approximation of the intended data being transmitted. Therefore, to make approximate communication useful, we need to answer the following question: *Given a received bit sequence, how do we know whether the error in transmitted value is small or large?* In a six-bit sequence, there are 15 different ways in which two bit errors can occur. The bit errors shown above is the one which distorts the original integral value of 60 the least (by 5%). Any other two bit error will lead to higher distortion in the received value. For instance, if the signal is decoded to be the sequence 110000, i.e., 48, then the error in the received value is 12/60, i.e., 20%. Similarly, if the signal is decoded to be 001100, i.e., 12, then the error in the received value is 48/60, i.e., 80%. At a first glance, it might appear that in order to quantify the potential impact of these errors, we need to know the exact bit positions that are in error. Obtaining such precise knowledge of erroneous bit positions in a general setting can be prohibitively expensive. In this work, we explain inherent

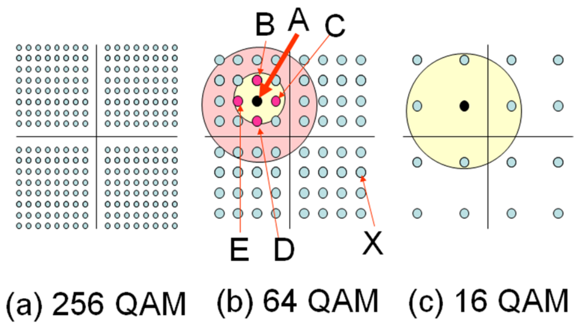



**Figure 1: Quadrature Amplitude Modulation (QAM) constellations.**

properties of the wireless channel that make it possible to ensure that the *overall impact of channel errors on transmitted data values* is small. Hence, in certain applications it may be possible to obtain better overall performance by letting through small errors across the communication channel.

**Key contributions:** Our main contributions include: proposing the notion of approximate communication, explaining how such a mechanism naturally fits the wireless channel, and demonstrating the potential of such a mechanism for popular mobile media applications.

Our notion of approximate communication is made feasible through interactions between the PHY layer and upper layers of the protocol stack. In the recent past, a number of new techniques have been proposed that also exploit understanding of the PHY layer behavior to improve performance on a wireless channel. Examples include PPR [3] and SOFT [7]. As discussed in Section 6, all such mechanisms, however, continue to adhere to the exact communication model. In contrast, in this paper we design techniques that will accept bits, even when received in error, and yet allow certain applications to improve their performance.

## 2. APPROACH OVERVIEW

The key to success of approximate communication is to ensure that the impact of errors in data elements are "small." Fortunately, data communication across a wireless medium can be designed to meet this goal. We explain this through an example based on the commonly applied Quadrature Amplitude Modulation (QAM) modulation system.

In QAMs, data elements are encoded into amplitude values of two sinusoidal waves that are 90 degrees out-of-phase with each other. A QAM modulation scheme is usually represented by a I/Q constellation diagram, as shown in Figure 1. Each constellation point (or symbol) is mapped to the amplitude of the in-phased and the quadrature-phased signals, and corresponds to a certain bit sequence to be transmitted. In a 64-QAM scheme (shown in Figure 1(b)), there are 64 distinct symbols. Hence, each such symbol encodes a 6-bit sequence to be transmitted. In contrast, in a 256-QAM scheme, there are 256 distinct symbols, and each symbol encodes a 8-bit sequence. During communication, a transmitter emits a particular symbol, corresponding to a specific bit sequence, and the goal of the receiver is to identify which symbol was sent.

In the 64-QAM example, when symbol $A$ is sent (corresponding to say, bit sequence 111100, i.e., a integer value of 60) and the channel condition is good, then the likelihood of the receiver decoding the symbol as $A$ as well, is high. There is certainly a probability of making errors in symbol decoding. However, if an error occurs, such errors are most likely going to be confined to the neighborhood of symbol $A$, i.e., the receiver is more likely to decode this symbol to be one of $B, C, D$, and $E$, as shown by the small circle in Figure 1(b). The probability of decoding the transmitted symbol as a further away symbol, e.g., as symbol $X$, is significantly lower. In Figure 2, we illustrate the probability of decoding a transmitted symbol $A$, as another symbol $A'$ as a function of distance between the two symbols in the I/Q space. The distance between immediate neighbors, (such as $A - B$ and $A - C$ in Figure 1), is assumed to be 1. The data points for this graph were collected by sending 30,000 16-QAM modulated symbols across two USRP radio boards, operating in 2.4 GHz frequency range. The receive signal strength (RSS) was -70 dBm Therefore, if we map the value 60 to symbol $A$, and then map the neighboring symbols ($B, C, D$, and $E$) to nearby values, e.g., to values 61, 62, 59, and 58, then the likelihood of decoding the transmitted value to within a small error bound, is high. Under such circumstances, approximate communication can be quite effective.

If the interference level in the channel is higher, say due to host mobility or multi-path, then the distortions in decoding the received signals will be greater as well. It is possible that common symbol errors (when transmitting $A$) spread from being limited to the inner circle of Figure 1(b), to being limited to the outer circle of the figure, potentially leading to larger errors.

Current wireless devices adapt to such changing channel conditions by performing rate adaptation. For example IEEE 802.11-based systems adapt the link to operate using a smaller constellation space, i.e., using a 16-QAM modulation scheme, in which neighboring symbols are spread wider apart, as shown in Figure 1(c). Approximate communication would also use similar rate adaptation schemes to ensure that the errors are withing bounds.

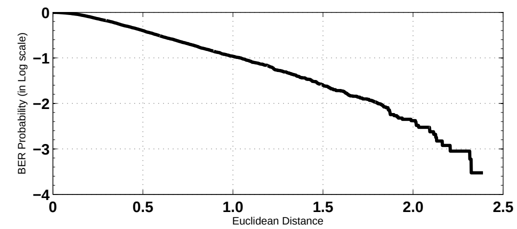



**Figure 2: Experimental results showing variation in the error probabilities with distance between transmitted and received symbols with Tx power of -70 dBm. (Y-axis in log scale.)**

## 3. APPLYING APPROXIMATE COMMUNICATION TO MOBILE APPLICATIONS

There has been a great surge in small form factor, mobile WiFi devices in recent years. Examples include Voiceover-WiFi phones (iPhone), WiFi-enabled music players (iTouch, Zune), and other such gadgets. Streaming media forms an important class of applications for these devices, often en-

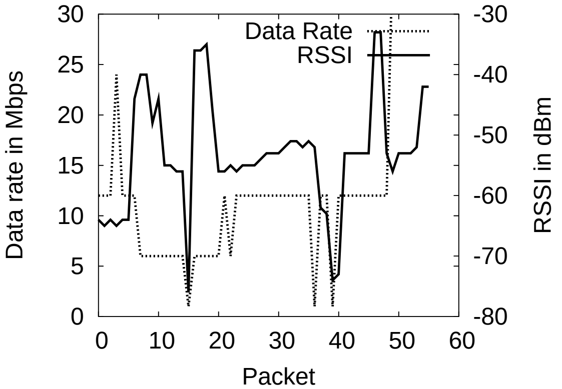



**Figure 3: Traditional coarse grained rate adaptation in wireless devices**

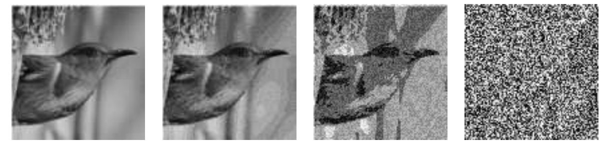
joyed by users while on the move. It is exactly this class of applications that can leverage approximation communication to provide better utilization of the medium under mobility conditions.

We explain this through a simple mobility experiment in which a static host is transmitting a stream of packets to a mobile host approaching the former at walking speeds. Figure 3 shows the signal strength (RSSI) and the physical transmission rate of received packets at the approaching mobile host. The signal strength has a increasing trend over the set of packets, but there are continuous variations. A popular rate adaptation algorithm, SampleRate, cannot adapt to these fast changes in the channel. For example, packets 1 to 6 started at a transmission rate of 12 Mbps. However, user mobility led to packet losses in these early packets, leading to a lowering of the data transmission rate for the subsequent 15 packets (from 7 to 21). Although the signal strength of these 15 packets are significantly higher than the first 6, it is only with the 22nd packet, does the data rate return to 12 Mbps. The experiment highlights a difficulty faced by any rate adaptation algorithm in reacting to fast changes in channel conditions under mobile scenarios.

In contrast, using approximate communication, the transmitter can intentionally select a relatively higher transmission rate. If the channel condition for individual packets stay good, then such packets will be received without errors. If, however, the channel temporarily transitions to a poorer state, then the natural property of approximate communication ensures that the lower significant bits of the data are impacted, distorting the real values slightly. In a way this allows the link to gracefully degrade with arbitrary changes in channel conditions without requiring a rate adaptation algorithm that is both accurate and agile to such changes.

We illustrate this, further, using an example depicting the impact of approximate communication for a media (image) file. For simplicity, we consider a simple digital image format, BMP, that is being transmitted across the channel. A BMP file (like any other image file) starts with a certain amount of meta-data about the image (color palette and image description). The remaining data bits in the file represent the Red, Green, and Blue (RGB) color intensities of different pixels.

For correct rendering of the BMP image at the receiver, it is critical that the meta-data have no errors. However, the receiver can potentially accept a limited amount of errors in the RGB color intensities. Hence, we assume that the higher layer can indicate to the lower layer which data bits require exact communication (the meta-data part) and which data bits can survive well under approximate communication (RGB values).



(a) Actual Image (b) Errors in 4 MSBs (c) Errors in 6 MSBs (d) Errors in all bits

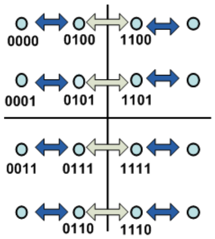
**Figure 4: Effect of random errors at different bit-pairs of a representative BMP image**

In our example, we assumed that the wireless link is using a 256-QAM for communication, where each symbol encodes 8 bits, and apply approximate communication selectively to those bits and symbols carrying RGB values of the BMP file. In Figure 4 we show a sequence of figures that illustrate the impact of increasing degree of error in the communicated data values. Figure 4(a) shows the original image. The image, with random errors in the two least significant bits (LSBs) of the RGB values has non-discernable distortion from the original (and hence, is not shown). Figures 4(b), (c), and (d) show the impact of approximate communication when it introduces random errors in the four LSBs, the six LSBs, and all the eight bits of the RGB values. The graceful degradation in performance is apparent.

Going beyond this simple illustration, any data element that represents a physical phenomenon, e.g., the color intensity of a pixel, the intensity of audio frequency in a audio signal, temperature reading from a wireless sensor, etc., tends to exhibit such a graceful degradation of quality with accuracy. In fact, most lossy image and audio compression techniques leverage this behavior to achieve their redundancy elimination goals.

Besides approximately preserving transmitted data values, approximate communication can also be used to implement unequal error protection of different bits. Unequal error protection is desirable in many media applications. For example, in MPEG-coded, the data is partitioned in into I-, P-, and B-frames, with I-frames requiring the greatest protection, and B-frames requiring the least. Similarly, various audio streaming systems, e.g., one-dimensional SPIHT [4], also partition its data into different layers with different levels of priority.

Unequal error protection of data bits have commonly been provided through two alternative approaches: (i) greater redundancy in encoding higher priority bits at the application layer [2], and (ii) lower PHY data rates when carrying bits of higher priority. Approximate communication provides a new mechanism to achieve unequal error protection. This can be made possible by mapping bits of different priorities within each symbol of a constellation, and ensuring that the different values for the higher priority bits are further separated in the constellation space, than the lower priority bits. We elaborate on this further in Section 4. In particular, unequal error protection through approximate communication, therefore, allows applications to achieve the improved per-




| Bit Position | Neighbor Pairs (with diff. values) |
|---|---|
| 1 | 4 |
| 2 | 8 |
| 3 | 4 |
| 4 | 8 |

**Figure 5: Bit map using Gray coding for a 16-QAM.**

formance, without requiring more redundant bits or reducing the data rate. This makes the approximate communication approach particularly suited to carry video and audio streams over wireless links.

# 4. TECHNICAL DETAILS

The key to implementing approximate communication is to ensure that when symbol errors occur, the most significant bits of the data value have a lower probability of error, when compared to the less significant bits of data. We will show that this is achievable through careful mapping of bit sequences to symbols in a constellation diagram. In addition, the choice of the encoding decoding scheme also play an important role in providing differentiated protection to different bit positions transmitted across a wireless link. We explore these two issues in turn.

**Mapping data to bit positions:** A coding scheme typically specifies a mapping between bit sequences to symbols for the constellation diagram. Figure 5 shows this mapping for the commonly used Gray coding scheme. We consider the probability of erroneous values at different bit positions when symbol errors occur. Based on data presented in Figure 2, we assume that a symbol in error is most likely to be decoded as one of the immediate neighbors either in the horizontal or in the vertical directions. Therefore, the likelihood of a bit in error depends on the number of occurrences when immediate neighbors differ in that bit position.

For example, for the Gray coding scheme, the number of occurrences of two immediate neighbors differing in bit position 1 (the most significant bit), is four, and is marked by the grey bi-directional arrows). Similarly, the number of occurrences of two neighbors differing in bit position 2, is 8, which is marked by the black bi-directional arrows). Hence, if a symbol error occurs, the likelihood of an error in bit position 2, is higher than the likelihood of an error in bit position 1. Figure 5 also indicates the number of occurrences of neighboring symbols being different in the four different bit positions across the entire 16-QAM.

This suggests that the data bits should be mapped to bit positions according to their priority to achieve the desired protection and approximation. We illustrate this further with an example of encoding a video sequence consisting of four frames: I-B-P-B in that order. For illustration purposes, we assume that each frame consists of 4 bits only.

Note that in Gray coding, bits 1 and 3 are best protected, as they have fewer immediate neighbors with different values. Hence, if we were to map all I-frame bits to bit positions 1 and 3 only, they would achieve their desired highest level of protection. We can fill the remaining bit positions with

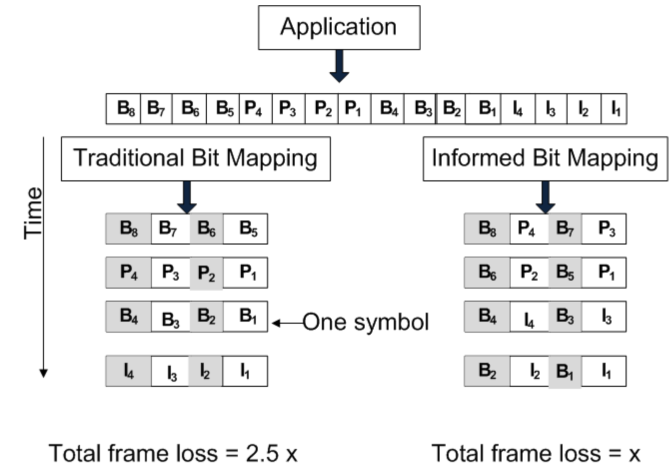



**Figure 6: Informed bit mapping for bit error differentiation.**

the other B- and P-frame bits.

In Figure 6 we compare such informed mapping between data bits to symbols, to a standard approach where all bit positions are assumed to be equivalent. Through trace analysis of sample videos, we found that the loss of B frame leads to the loss of that frame alone, a loss of a P-frame usually affects a total of 2 frames, and a loss of an I-frame usually affects a total of 6 frames on average. Under these assumptions, and through simple probabilistic calculations, the more intelligent mapping of data bits to symbols *decrease frame losses by a factor of 2.5,* when compared to the standard approach. This can have significant impact on the perception o the video.

**Choice of encoding-decoding scheme:** The Gray coding scheme, presented in the previous example, was not designed with error differentiation between bits in mind. While it offers some differentiation of errors between different bit positions, by no means is it the most effective one. We illustrate this by designing a more effective bit error differentiation scheme that utilizes by using a combination of two existing coding schemes, known in the literature as block coding [8] and Ungerboeck coding [6].

*Block coding:* In this scheme, all bit sequences with different values in bit position 1 are on opposite side of the Y-axis, while all bit sequences with different values in bit position 2 are on opposite side of the X-axis. Effectively, all sequences in each quadrant of a QAM has the same value for the first two bits, and the values of different quadrants are different. Within each quadrant, the same process recurses with the next set of bits. This approach ensures a higher error resilience for the more significant bits than the less significant ones.

*Ungerboeck coding:* In this scheme, the differential error resilience of bit positions is met by mapping bits to symbols using a set partitioning method. In particular, it starts with the MSB and divides the symbols of the constellation into two sets such that the members of a set have a minimum separation of $\sqrt{2}$ (assuming unit distance between immediate neighbor pairs, all set members are at diagonals). The two sets would have 0 and 1 at their MSB. In the next level, symbols belonging to a set are again partitioned into two

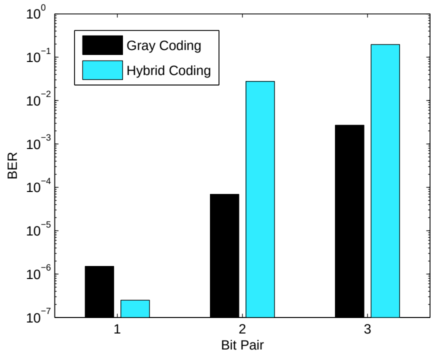



**Figure 7: BERs at different bit positions for a 64 QAM, using Gray coding (dark bars) and hybrid coding (light bars) for a 12 dB SNR channel. Y-axis in logscale.**

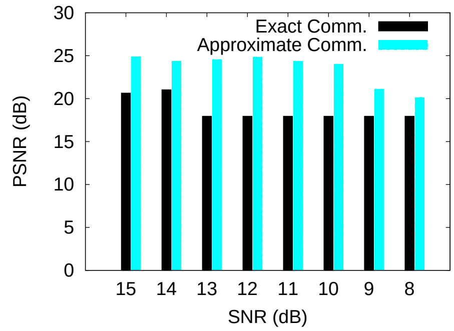



**Figure 8: PSNR (in dB) of exact communication (using Gray coding, uninformed bit mapping) and approximate communication (using hybrid coding, informed bit mapping).**

sets, such that, the the members of level two sets have an minimum euclidean distance of 2 and so on, till the last level when each set has only one element. Ungerboeck coding scheme uses a sequential decoder, i.e., the decoding of higher bits has information of lower bits as input.

We constructed a *hybrid* coding scheme that utilizes a combination of block coding and Ungerboeck coding for differential error protection of various bits of a symbol. In particular, in the hybrid scheme, we encode the first $2k$ bits using a block coding approach, and the remaining bits using Ungerboeck coding, with $k$ being a configurable parameter. This allows us to create different coding schemes with different levels of protection for different bits.

In Figure 7 we compare the performance of this hybrid scheme (with $k = 1$) to a standard Gray coding scheme (discussed before). The plot shows the error probability of different bit positions of a 64-QAM, using the two schemes. The interesting point to note is that the hybrid scheme allows us to protect the first two bit positions better, i.e., with lower errors, at the cost of higher errors at the remaining four bit positions. Such tradeoffs can be effectively used by applications to optimize its own performance.

**Handling convolution coding:** The traditional wireless communication hardware convolutionally encodes the input bits before mapping them to constellation points. This makes it difficult to provide differential error protection to input bits, as relative priority is not mapped across convolution coding. We intend to overcome this problem by applying convolution coding separately to bits belonging to different priority classes. This would ensure that the relative priority to the bits is carried across the convolution process.

## 5. EVALUATION

We evaluated the performance of approximate communication by combining the effect of intelligent bit mapping to symbols, and our hybrid scheme for bit error differentiation by simulating its performance on the well known Foreman video sequence, when transmitted over an AWGN channel with different SNR values. In this sequence each frame is $178 \times 144$ pixels, and the video is encoded at 30 frames per second (QCIF). We used the H.264 reference encoder software for encoding the video frames. The encoder generates a set of I, P and B frames. We compare our approximate communication system, to a traditional exact communication system (that only uses Gray coding).

In Figure 8 we present the Peak Signal-to-Noise Ratio (PSNR) values of the decoded video for the two cases for a range of noisy channel conditions. (Higher PSNR implies better performance.) Even though, the designed hybrid coding is simplistic, it can easily lead to better performance over the traditional, exact communication approach in this range of noise conditions.

## 6. RELATED WORK

In recent years, a growing number of efforts have designed and studied wireless communication techniques that effectively combine mechanisms at the physical layer with those at the MAC layer and higher. For example, Jamieson and Balakrishnan [3] suggested partial recovery of wireless frames received in error. In their scheme, named PPR, they introduce additional checksums into a wireless frame. If a frame has errors, then such checksums can be used to identify its fragments that do not have any errors. The correctly received bits from such error-free fragments are, extracted out of the corrupted frame, while fragments, with numerous bit errors, are discarded and potentially recovered through subsequent re-transmissions. In SOFT, Woo et. al. [7] use physical layer confidence values of symbols decoded by different receivers to recover the transmissions more accurately than possible with a single receiver.

There have been several research efforts to improve the quality of streaming content. These works involve either masking the errors present in the video frames by intelligently reconstructing the missing portions of a video frame, by using information from the neighboring frames [5]. Other research have also studied techniques to combine ARQ-based and FEC-based data recovery mechanisms once packet losses have occurred [1, 5].

However, in all the above schemes, errors have still been considered to be binary. While they increase the channel efficiency by increasing the number of correctly received data elements from each frame, any data element that is considered to be in error is discarded. In contrast, our proposal is to expose a certain degree of error in data element decoding to the higher layers (as permitted by their semantics).

## 7. CONCLUSION AND FUTURE WORK

In this paper, we have presented the notion of approximate communication, whereby erroneous values are exposed to applications that gracefully deal with such errors in wireless settings.

The initial evaluations results are quite intriguing, and we believe this approach can open a whole new toolbox of optimizations for mobile media applications. Towards this end, significant further studies need to be done. They include:

- Adaptation to dynamics in interference and noise: Our initial evaluation have been simulation-based and trace-based. A real system needs to consider *dynamic* effects of interference and noise on the channel, including the effects of mobility. In particular, our bit-mapping and error differentiation schemes for approximate communication need to effectively interact with rate adaptation techniques adopted by the wireless link.
- Exact vs Approximate: It is clear that not all data types are well suited for approximate communication. More specifically, any content that has strict reliability semantics need to be communicated through exact means. However, when a pair of mobile devices intend to interact using the two different forms of communication for different types of content, co-existence between the two components, will be an interesting challenge.
- Interactions between applications and MAC/PHY: An important requirement in this design is for the application express appropriate hints to the MAC and PHY layers about the relative importance of data. Proper mechanisms and semantics need to be designed to achieve these goals.

## Acknowledgment

All authors were supported in part by the US National Science Foundation under grants CNS-0639434, CNS-0627589, CNS-0627102, CNS-0520152, and CNS-0747177.

## 8. REFERENCES

[1] N. Feamster and H. Balakrishnan. Packet loss recovery for streaming video. In *Packet Video Workshop*, 2002.

[2] W. R. Heinzelman, M. Budagavi, and R. Talluri. Unequal error protection of mpeg-4 compressed video. ICIP, 1999.

[3] K. Jamieson and H. Balakrishnan. Ppr: partial packet recovery for wireless networks. *SIGCOMM*, 2007.

[4] Z. Lu and W. A. Pearlman. An efficient, low-complexity audio coder delivering multiple levels of quality for interactive applications. IEEE Workshop on Multimedia Signal Processing, 1998.

[5] H. Seferoglu, Y. Altunbasak, O. Gurbuz, and O. Ercetin. Rate-distortion optimized joint arq-fec scheme for real-time wireless multimedia. ICC, 2005.

[6] G. Ungerboeck. Channel coding with multilevel/phase signals. In *IEEE Trans. on Information Theory*, 2006.

[7] G. R. Woo, P. Kheradpour, D. Shen, and D. Katabi. Beyond the bits: cooperative packet recovery using physical layer information. In *MOBICOM*, 2007.

[8] R. Zaragoza, M. P. C. Fossorier, S. Lin, and H. Imai. Multilevel coded modulation for unequal error protection and multistage decoding-part 1: Symmetric constellations. In *IEEE Transactions on Communications*, 1999.