Optimal Audio Transmission over Error-Prone Wireless Links

Ala' Khalifeh, Student Member IEEE   Homayoun Yousefi'zadeh, Sr. Member IEEE

***Abstract*—In this paper, we present an optimization framework for transmitting high quality audio sequences over error-prone wireless links. Our framework introduces apparatus and technique to optimally protect a stored audio sequence transmitted over a wireless link while considering the packetization overhead of audio frames. Utilizing rate compatible punctured RS codes and dynamic programming, it identifies the optimal assignment of parity to audio frames according to their perceptual importance such that the Segmented SNR of the received audio sequence is maximized. Our framework covers two cases. In the first case, a frame grouping technique is proposed to packetize audio frames and protect them against temporarily correlated bit errors introduced by a fading wireless channel. In this case, each packet is treated as a channel coding codeword. In the second case, a one-dimensional RS coder is applied vertically to a sequence of horizontally formed packets associated with an audio sequence in order to protect the sequence against both bit errors introduced by fading wireless channels and packet erasures introduced by network buffering. Our numerical results capture the performance advantage of our framework compared to existing techniques proposed in the literature of audio transmission. We also note that our framework can be generically applied to a variety of audio coders making it attractive in terms of implementation.**

***Index Terms*—MPEG-4 Bit Slice Arithmetic Coding (BSAC), Audio Transmission, Optimal Unequal Error Protection, Dynamic Programming, Random Bit Errors, Packet Erasures, Wireless MIMO Links**

I. Introduction

In the past few years, streaming high quality rich media content over wireless infrastructure has gained a widespread acceptance due to an ever increasing demand. In that context, delivering high quality audio over error-prone wireless networks is subject to a variety of challenges. Some of these challenges include reducing the packetization overhead of small audio frames as well as protecting audio content against both bit errors introduced by wireless fading channels and packet erasures introduced by network buffering. The use of Forward Error Correction (FEC) codes represents a popular technique of mitigating the effects of bit errors and packet erasures. In what follows, a brief review of the literature related to the transmission of audio over wired and wireless networks is provided. Considering the depth of research conducted in the field, the review cannot be exhaustive. Rather, it includes a mentioning of the work more closely related to the subject of interest to this paper. As evidenced by the work of [11] and others, audio transmission is typically performed using a frame-based approach in which a number of frames are generated from an audio source. Each frame, consisting of a number of symbols, is in turn encoded, packetized, and transmitted across a transmission medium. Packetizing audio frames is subject to a high overhead due to the large size of packet headers associated with Internet Protocol (IP), User Datagram Protocol (UDP), and Real-Time Protocol (RTP) compared to the small payload size of audio frames. Further, transmitting small packets over most of today's Medium Access Control (MAC) protocols is subject to a significant overhead, related performance degradation, and a fairness problem as described in [18] and [19]. In the literature, many different techniques have been proposed to decrease the overhead of transmission and increase packet payload sizes. In [18], the authors propose concatenating small packets into larger packets such that the overhead is shared among a group of packets instead of being applied to single packets. In [19], the authors propose a voice-audio multiplexing-multicasting scheme in which multiple packets belonging to different users are combined into large packets. Large packets are received by all users, each user extracts its own packet and drops other packets. In addition, header compression is a technique widely used to reduce packetization overhead [7]. To protect an audio sequence against bit errors introduced by the wireless transmission medium, the authors of [25] propose the use of an Unequal Error Protection (UEP) scheme relying on Reed-Solomon (RS) FEC codes. They propose two schemes, the first is an unequal frame protection scheme in which more protection is added to the header portion of a frame as opposed to its data portion. The second scheme employs UEP in which more protection is assigned to the most significant bits

The material in this paper appeared in part in the Proc. of IEEE WCNC'08 and IEEE DDC'08. The authors are with the Department of EECS, University of California, Irvine. e-mail: `[akhalife,hyousefi]@uci.edu`. This work was supported in part by a research contract from the Boeing Company and the Fulbright Scholarship Program.

of quantized data in comparison to the least significant bits. In [20], the authors suggest a Content-based UEP (C-UEP) framework for transmitting audio over wireless links. The framework protects the most perceptual audio frames against packet loss by generating a redundant secondary stream. The authors of [9] suggest a perceptually controlled error protection scheme for transmitting audio over IP networks. They present a UEP scheme in which the critical frames are transmitted twice at a high and a low bit rate in order for them to be delivered with a high probability. Other frameworks of audio transmission over the Internet are suggested in [21] and [15]. One can notice that the papers cited above consider either the existence of packet loss caused by network buffering overflow or bit errors introduced by the fading wireless links but not both packet loss and bit errors. In addition, they mostly model packet loss and bit errors by a simple Bernoulli model which does not accurately capture the temporally correlated characteristic of loss observed over wireless networks [8], [22].

Our literature review reveals a compelling need for the identification of a unified framework of audio transmission that is capable of jointly addressing the aforementioned transmission challenges in an optimal manner. In addition and unlike most of the existing work, such a framework is required to be generically applicable to a variety of audio codecs as opposed to being specific to special types of audio codecs.

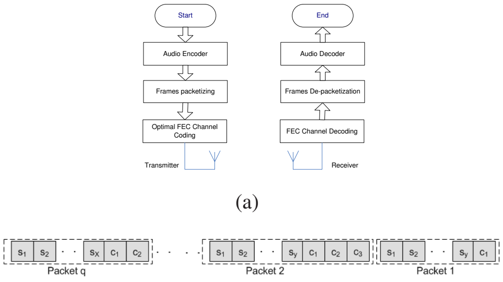
This paper proposes an optimization framework for transmitting audio sequences over wireless links based on a detailed analysis of the wireless channel. The framework leverages FEC and suggests an optimal way of assigning parity to audio frames according to the perceptual sensitivity of the frames. It protects audio frames against both bit errors and packet erasures. It also proposes an efficient way for packetizing and transmitting audio frames such that the packetization overhead is minimized. The framework does not mandate the use of a specific audio codec with certain characteristics. Rather, it can be used with a variety of publicly available audio codecs. The rest of the paper is organized as follows. In Section II, we describe our proposed framework based on a realization of the wireless channel. In Section III, we formulate a pair of optimization problems appropriate for wireless channels introducing (a) bit errors only, or (b) bit errors and packet erasures. We also offer effective solutions to those problems utilizing dynamic programming. In section IV, we describe our experimentation setup and performance evaluation results. Finally, Section V concludes the paper and proposes future work.

II. Description of the Proposed Framework and Channel Analysis

In this section, we provide a description of our proposed framework based on a realization of the underlying wireless channel. The framework considers two scenarios. While in the first scenario our framework protects an audio sequence against bit errors only, in the second scenario it protects an audio sequence against both bit errors and packet erasures.

A. Protecting an Audio Sequence against Bit Errors Only

Fig. 1(a) depicts the block diagram of our proposed framework used to protect an audio sequence against bit errors introduced by a wireless channel. An audio sequence is first partitioned into small audio clips. Each clip consists of a certain number of audio frames and each frame is encoded using the MPEG-4 Bit Slice Arithmetic Coding (BSAC). Specifically, we use MPEG-4 Natural Audio Coding Toolkit publicly available at the ISO website [1]. As explained later in this paper, the length of each clip is chosen such that it neither overflows nor underflows the receiver's buffer resulting in a smooth playout of the entire audio sequence at the receiver. Following encoding, J audio frames are grouped and packetized into a single packet as illustrated by Fig. 1(b). The audio clip consists of q packets. Each packet consists of a different number of symbols $(S_1, S_2, ...)$ depending on the frame group numbering scheme used in transmission, and each packet is assigned a different number of parity symbols $(C_1, C_2, ...)$ according to its perceptual importance. We note that a tradeoff needs to be addressed when choosing J. On one hand, increasing J reduces packetization overhead, improves transmission efficiency, and improves coding efficiency by allowing for increasing FEC block size [16]. On the other hand, increasing J increases the distortion of the received audio sequence in the case of losing a packet. Thus, frame grouping is accompanied with a FEC-based UEP technique in which the assignment of parity to each packet is performed according to the importance of packets. The metric of importance and the optimal parity assignment algorithm are described in Section III. Once parity assignment is done, packets are transmitted over the wireless channel which may be characterized by multiple antenna transmitting or receiving ends. On the receiving side, both the payload and the parity bits are buffered at the channel decoding buffer, channel decoding is applied to individual packets either fully recovering the contents of a packet or completely discarding it. In the case of discarding a packet, error concealment is used to replace the discarded packet with the last correctly received packet. Packets are then passed to the audio decoding block for playout. In this framework, the clip size is chosen as 1024 audio frames corresponding to an average duration of 30 seconds. We assume that the channel decoding buffer can accommodate the payload and associated parity of at least two



(a)

(b)

Fig. 1. A description of the transmission framework capable of protecting an audio sequence against bit errors only: (a) the block diagram of the proposed framework, and (b) an illustration of the packetization process.

audio clips. This choice of the buffer size ensures that at the time of playing the first clip, the receiver can buffer the next clip thereby allowing for a smooth and continuous play out of the entire sequence. It is also important to note that the maximum parity applicable to an audio clip is determined such that the transmission time of the clip does not exceed its end-to-end playout delay deadline. In addition, the transmitter controls its transmission rate in order to ensure that the receiver's buffer is neither in an overflow nor in an underflow condition. Once the transmitter starts transmitting the audio frames of the first audio clip, it estimates the time needed to transmit the clip and continues transmission till reaching the receiver's channel decoding buffer size limit set to the size of two audio clips. At this point, the transmitter checks the elapsed transmission time, subtracts it from the playout time of the first clip, and pauses transmission till the receiver finishes playing out the first audio clip. Once the receiver starts playing the second buffered audio clip, the transmitter resumes transmitting an additional audio clip and applies the rate control mechanism as before. The process of resuming-pausing transmission is continued till the entire audio sequence is transmitted.

B. *Protecting an Audio Sequence against both Bit Errors and Packet Erasures*

We open this section by noting that the scheme described in the previous subsection cannot be efficiently used if the transmitted packets incur both bit errors due to fading and packet erasures caused by network buffering. Hence, we revise our framework such that it is capable of jointly protecting an audio sequence against both types of errors. Similar to the first scheme, the audio sequence is first partitioned into a number of small audio clips and uses the same rate control process described before. As depicted in Fig. 2(a), each audio clip is first encoded using MPEG-4 Bit Slice Arithmetic Coding (BSAC). Next, a UEP mechanism is applied through which parity is assigned according to the importance of audio frames. The optimal parity assignment algorithm jointly protecting the audio sequence against both bit errors and packet erasures is described in Section III. Fig. 2(b) illustrates the channel coding grid alignment used in this case. $S_{x,y}$ and $C_{x,y}$ correspond to source and parity symbols of frame x, respectively. M represents the total number of frames of the audio clip. As illustrated, channel coding blocks are aligned on the columns of the grid such that each column corresponds to one coding block while packets are formed over the grid rows. Each channel coding block corresponds to one audio frame. As such, each symbol in every packet belongs to a different channel coding block and the loss of a packet results in losing one symbol per channel coding block. Not only the use of this scheme increases the payload size of packets and reduces the overhead of packetization, but it also mitigates the effects of packet erasures as the result of interleaving. Since the rectangle in Fig. 2(b) is not fully filled, variable size RS blocks may result. In order to avoid having variable size RS blocks, a technique known as shortening is used. Shortening works by having an encoder add padding zero symbols to short RS blocks until their sizes become equal to that of a standard block size known to both encoder and decoder. It is important to note that in shortening, the padded zeros are not transmitted. Rather, at the decoder padding zeros are locally added to those short blocks. Once packet payloads are formed, the header of each packet is added and compressed. In order to protect the header of each packet against bit errors, an additional 80 parity bits are assigned to packet header bits. Without this protection, losing a single header bit will render the entire packet useless. Packets are transmitted over a wireless fading channel which may be utilizing multiple antenna transmitting or receiving ends. We assume that at the beginning of the transmission, the transmitter forms a meta-data packet which includes important information needed by the receiver to re-establish the grid and attempt at reconstructing the blocks of the grid from the received packets. Such meta-data includes the total number of blocks, individual block sizes, and the number of packets. We assume that this packet is delivered error free by applying proper protection. Once all packets associated with a grid are received, the information grid associated with a transmitted clip can be formed and channel coding can be applied to the columns of the grid in order to recover the blocks of the grid. As the result of applying channel coding, every individual block is either fully recovered or completely discarded. In the case of discarding a block, error concealment is used to replace the discarded block with the content of the received immediately before the current block. The blocks are then passed to the audio decoder for playout.

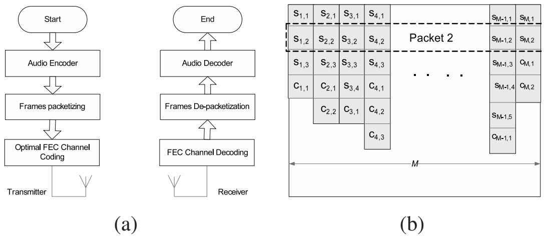

Start
Audio Encoder
Frames packetizing
Optimal FEC Channel Coding
Transmitter
End
Audio Decoder
Frames De-packetization
FEC Channel Decoding
Receiver
(a)
s1,1 s2,1 s3,1 s4,1 sM-1,1 sM,1
s1,2 s2,2 s3,2 s4,2 Packet 2 sM-1,2 sM,2
s1,3 s2,3 s3,3 s4,3 sM-1,3 cM,1
c1,1 c2,1 s3,4 c4,1 sM-1,4 cM,2
c2,2 c3,1 c4,2 sM-1,5
c4,3 cM-1,1
M
(b)

Fig. 2. A description of the transmission framework capable of protecting an audio sequence against both bit errors and packet erasures: (a) the block diagram of the proposed framework, and (b) the channel coding grid alignment.

C. Wireless Channel Analysis

In what follows, we briefly describe the wireless channel model, the calculation of the Symbol Error Rates (SERs), and the FEC channel coding scheme used to protect packets against both bit errors and packet erasures. As described by the work of [24] and [23], a Multiple Input Multiple Output (MIMO) wireless fading channel is characterized by a temporally correlated pattern of bit loss. In order to capture this loss behavior, we use the two-state Gilbert-Elliott (GE) model. In the GE model, the corruption pattern of an audio bitstream is described by a two-state Markov chain introducing a good (G) state and a bad (B) state. State G represents a bit error rate of ε_G while state B represents a bit error rate of ε_B where $\varepsilon_B >> \varepsilon_G$. Let $P(t,q,G)$ and $P(t,q,B)$ denote the probability of receiving q bits from t transmitted bits and ending up in state G and B of the GE model, respectively. Then the overall probability of receiving q bits from t transmitted bits under the GE model is calculated [24] as

$$P(t,q) \;=\; P(t,q,G) + P(t,q,B), \tag{1}$$

where the recursive probabilities $P(t,q,G)$ and $P(t,q,B)$ are given by

$$\begin{aligned} P(t,q,G) \;&= \varepsilon_G \left[\gamma\, P(t-1,q,G)\right. \\ &+ \left.(1-\beta) P(t-1,q,B)\right] \;+\; (1-\varepsilon_G) \\ &\times [\gamma\, P(t-1,q-1,G) \;+\; (1-\beta) P(t-1,q-1,B)], \end{aligned} \tag{2}$$

and

$$\begin{aligned} P(t,q,B) \;&= \varepsilon_B \left[(1-\gamma)\, P(t-1,q,G)\right. \\ &+ \left.\beta P(t-1,q,B)\right] \;+\; (1-\varepsilon_B) \\ &\times [(1-\gamma)\, P(t-1,q-1,G) \;+\; \beta P(t-1,q-1,B)]. \end{aligned} \tag{3}$$

In the equations above, $t \geq q > 0$ and the initial conditions are described as:

$$\begin{aligned} P(0,0,G) &= \; g_{ss} \;=\; \tfrac{1-\beta}{2-\gamma-\beta}, \\ P(0,0,B) &= \; b_{ss} \;=\; \tfrac{1-\gamma}{2-\gamma-\beta}, \\ P(1,0,G) &= \; \varepsilon_G \left[\gamma\, g_{ss} \;+\; (1-\beta)\, b_{ss}\right], \\ P(1,0,B) &= \; \varepsilon_B \left[(1-\gamma)\, g_{ss} \;+\; \beta\, b_{ss}\right], \end{aligned} \tag{4}$$

where γ and β are the probabilities of self transitioning for state G and state B, respectively. Further, per state bit error rates ε_G and ε_B can be calculated in terms of the number of transmit/receive antennas and the average received signal-to-noise ratios. In [24], closed-form expressions describing these per state error rates are identified assuming a flat fading Rayleigh channel. Based on that discussion, the generic Z-PSK modulation symbol error rate of a link associated with a single-transmit W-receive antenna link using Maximum Ratio Combining (MRC) is identified as

$$\begin{aligned} \varepsilon_G \;=\; & \tfrac{Z-1}{Z} - \tfrac{1}{\pi}\sqrt{\tfrac{\vartheta_G}{1+\vartheta_G}} \\ & \times\big\{ (\tfrac{\pi}{2} + \tan^{-1}\chi_G) \textstyle\sum_{j=0}^{W-1} \binom{2j}{j} \tfrac{1}{[4(1+\vartheta_G)]^j} \\ & + \sin(\tan^{-1}\chi_G) \textstyle\sum_{j=1}^{W-1} \sum_{i=1}^{j} \tfrac{\sigma_{ij}}{(1+\vartheta_G)^j} \\ & \times [\cos(\tan^{-1}\chi_G)]^{2(j-i)+1} \big\}, \end{aligned} \tag{5}$$

where $\vartheta_G = SNR_G \sin^2(\frac{\pi}{Z})$, $\chi_G = \sqrt{\frac{\vartheta_G}{1+\vartheta_G}} \cot\frac{\pi}{Z}$, and $\sigma_{ij} = \frac{\binom{2j}{j}}{\binom{2(j-i)}{j-i} 4^i [2(j-i)+1]}$. According to the same discussion, Equation (5) can also be used to calculate the modulation symbol error rate of a link utilizing Space-Time Block Codes (STBCs) of [17] with the insertion of a proper SNR scaling factor. In the case of BPSK modulation, i.e., ($Z = 2$), Equation (5) maps modulation symbol error rates to bit error rates. In order to differentiate between per state bit error rates ε_G and ε_B, two different measures SNR_G and SNR_B are considered for state G and state B where $SNR_G >> SNR_B$.

For simulating packet erasures, we apply the Gilbert (G) model * to the symbols of a packet and use the probability of symbol loss (P_{sym}) as the measuring metric for packet loss. In the absence of FEC codes, a packet is declared lost if one or more of its symbols are lost. We note that the transitioning parameters of the G model are functions of network buffering dynamics but not fading channel dynamics. As such, those parameters are calculated independent of the GE model transitioning parameters.

III. Optimization Formulation and Solutions

In this section, we describe our optimization approach for two types of channels introducing (1) bit errors only, and (2) both bit errors and packet erasures.

A. Protecting against Bit Errors Only

In this section, we propose the use of an RS FEC code at the link layer to mitigate the effects of random bit errors. Each packet is coded as a stand-alone channel coding block consisting of a number of channel coding symbols. A channel coding symbol is differentiated from

*The G model is a special case of the GE model in which $\varepsilon_G = 0$ and $\varepsilon_B = 1$.

a modulation symbol and may itself consist of a number of modulation symbols. The maximum block size of a code is determined by the channel coding symbol size s. An RS code operating on an s-bit symbol size can have up to $n = 2^s - 1$ symbols per block [16]. An encoded block contains k data symbols and $C = n - k$ parity symbols. An RS channel coder $RS(n, k)$ can correct as many as $t_C = \lfloor \frac{C}{2} \rfloor$ symbol errors in a block. Suppose the RS coder generates a set of channel coding symbols where each symbol consists of s bits. A channel coding symbol is received error free if all of its s bits are received free of errors. Thus, the probability of receiving a channel coding symbol free of errors under the GE model is described by Equation (1) with $t = q = s$ as $P(s, s)$. Referring to the discussion of [23], we rely on a hybrid loss model to describe the probability of channel coding block loss. In our hybrid model, channel coding inter-symbol correlation is assumed not to be significant in comparison with channel coding intra-symbol correlation captured by the expression $P(s, s)$. As such, the probability of channel coding block loss is described as

$$\Psi(L_i, t_c, \varphi) \;=\; \textstyle\sum_{j=0}^{L_i - t_c - 1} P(t, j) = \\ \textstyle\sum_{j=0}^{L_i - t_c - 1} \binom{L_i}{j} (1 - P(s, s))^{L_i - j} (P(s, s))^j, \tag{6}$$

where L_i is the size of packet i and $P(s, s)$ is calculated recursively using Equation (1). We note that the use of a hybrid loss model does not affect the generality of discussion as the model can be replaced with one in which both inter-symbol and intra-symbol correlations are captured utilizing a pair of nested GE models.

Having described the channel coding approach, the main objective of the optimization problem of this section is to find the optimal parity assignment for each packet maximizing the quality of received audio clips. Each packet is formed as a collection of channel coding data and parity symbols. Each packet obtains a different number of parity symbols according to the perceptual importance of its data content.

We use Segmented Signal to Noise Ratio ($SSNR$) [4], commonly considered as one of the best time domain objective metrics of quality, to evaluate the quality of an audio sequence. The $SSNR$ is defined as

$$SSNR = \tfrac{10}{M} \textstyle\sum_{m=0}^{M-1} \log \\ \left\{ 1 + \frac{\sum_{n=1}^{N} x^2(mN+n)}{\sum_{n=1}^{N} [y(mN+n) - x(mN+n)]^2 + \delta} \right\}, \tag{7}$$

where $x(.)$ is the set of normalized pre-encoding samples of the transmitted audio sequence and $y(.)$ is the set of normalized post-decoding samples of the received audio sequence. N is the frame length in samples, M is the number of frames of the audio clip, and δ is a small number used to prevent division by zero. We note that a higher measure of $SSNR$ metric indicates a better quality. Relying on (7), we define $PSSNR$, the Segmented SNR of a packet, as

$$PSSNR = \textstyle\sum_{j=0}^{J-1} \log \\ \left\{ 1 + \frac{\sum_{n=1}^{N} x^2(jN+n)}{\sum_{n=1}^{N} [y(jN+n) - x(jN+n)]^2 + \delta} \right\}, \tag{8}$$

where J is the number of frames in a packet to which we refer as the Frame Group Number (FGN). We note that the summation in the denominator of (8) represents distortion D measured in terms of Mean Square Error (MSE). Thus, the $SSNR$ of an audio sequence can also be represented in terms of $PSSNR$ as

$$SSNR = \tfrac{10}{M} \textstyle\sum_{i=1}^{I} \sum_{j=0}^{J-1} \log \\ \left\{ 1 + \frac{\sum_{n=1}^{N} x^2(jN+n)}{\sum_{n=1}^{N} [y(jN+n) - x(jN+n)]^2 + \delta} \right\}, \tag{9}$$

where I is the number of packets of the audio sequence, and $M = JI$. If packet i is received successfully, $PSSNR(i)$ is expressed as

$$PSSNR(i) = \textstyle\sum_{j=0}^{J-1} \log \left\{ 1 + \frac{\sum_{n=1}^{N} x_i^2(jN+n)}{D(s) + \delta} \right\}. \tag{10}$$

Notice that in this case, the denominator of Equation (8) is equal to $D(s)$, where D(s) is the source coding distortion expressed as $D(s) = \sum_{n=1}^{N} [\overline{x}(jN+n) - x(jN+n)]^2$, and $\overline{x}(jN+n)$ is the set of normalized samples of the error free decoded signal. In the event of a packet loss, we use the Insertion-Base Repair (IBR) algorithm of [13] to represent a lost packet. In IBR, a lost packet is replaced by the last accurately received packet. If there is no previously received packet as is the case for the first transmitted packet, IBR replaces a lost packet with the next accurately received packet. In addition, the receiver inserts a comfort noise signal to substitute any sequence of lost packets that may include 8 consecutive frames. Consequently, $PSSNR$ is calculated using Equation (9) after calculating the distortion term. The distortion term is computed using the MSE between the original reference sample values of the packet frames $x(n)$ and the sample values of the packet frames used in the error concealment process. Hence, the value of $E[PSSNR]$ for packet i is expressed as

$$\mathcal{E}[PSSNR(i)] \;=\; (1 - \Psi_i) \textstyle\sum_{j=0}^{J-1} \log \\ \left\{ 1 + \frac{\sum_{n=1}^{N} x_i^2(jN+n)}{D(s) + \delta} \right\} + \Psi_i \textstyle\sum_{j=0}^{J-1} \log \\ \left\{ 1 + \frac{\sum_{n=1}^{N} x_i^2(jN+n)}{\sum_{n=1}^{N} [y_i(jN+n) - x_i(jN+n)]^2 + \delta} \right\}, \tag{11}$$

Therefore, $\mathcal{E}[SSNR]$ for the audio sequence is equal to

$$\mathcal{E}[SSNR] \;=\; \tfrac{10}{M} \textstyle\sum_{i=1}^{I} \mathcal{E}[PSSNR(i)], \tag{12}$$

where Ψ_i is the probability of losing packet i derived from Equation (6). Consequently, the optimization problem is given by

$$\max_{(C_1,\cdots,C_I)} \quad \mathcal{E}[SSNR] \tag{13}$$

$$\text{Subject To :} \quad \textstyle\sum_{i=1}^{I} (C_i + R_i) \le B_T \tag{14}$$

$$0 \le C_i + R_i < 2^{s_i} - 1, \ \forall i, \tag{15}$$

where B_T is the transmission budget allocated to the audio clip and expressed in channel coding symbols. We note that B_T includes the overall data budget $B_R = \sum_{i=1}^{I} R_i$ and the overall parity budget $B_C = \sum_{i=1}^{I} C_i$ allocated to the audio clip. We note that the data budget R_i for packet i includes both audio bitstream payload and packet header bits. As mentioned in Section II, we also note that the number of parity bits allocated to transmit the audio sequence is upper bounded by the maximum parity that the channel decoding buffer at the receiver can accommodate. Further, s_i is the symbol size of packet i chosen such that the packet size $L_i = R_i + C_i$ does not exceed the maximum RS block size of $(2^{s_i} - 1)$ symbols.

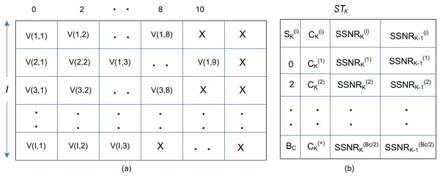


Fig. 3. (a) The structure of $SSNR$ matrix in which symbol X represents points at which the number of parity symbols for a given packet/frame exceeds the maximum possible block size, and (b) the state table of stage K used in dynamic programming.

We use dynamic programming [5] to solve the optimization problem. We divide the original problem into sub-problems and solve the sub-problems optimally in order to construct the optimal solution of the original problem. The values of $\mathcal{E}[PSSNR]$ for each packet, corresponding to all possible parity symbol assignments that each packet can have, are calculated and inserted into a so-called $SSNR$ matrix. Denote the values of this matrix as $V(r, m)$ where r is the row index and m is the column index. Fig. 3(a) demonstrates how this matrix is formed. Consider an audio sequence of I packets where the number of rows of the $SSNR$ matrix is equal to I and each row corresponds to one packet. To find the elements of the row associated with packet i, we simulate a loss event for packet i and conceal the packet using the IBR concealment algorithm. Then, we calculate the distortion measured in terms of MSE between the original frames and the frames used in IBR concealment algorithm. Next, we use the distortion measure to calculate $\mathcal{E}[PSSNR(i)]$ which evaluates the perceptual sensitivity of packet i. To calculate $\mathcal{E}[PSSNR(i)]$, we calculate the probability of losing packet i for all possible parity assignments. More specifically, $\mathcal{E}[PSSNR(i)]$ is calculated using Equation (11) for each parity assignment in the set $\{0, 2, \cdots, B_C\}$. The first column element of row i of $SSNR$ matrix is set as the value of $\mathcal{E}[PSSNR(i)]$ with a parity assignment of zero. Then the number of parity symbols is incremented by two symbols, the packet size associated with this assignment is calculated, and compared against Constraint (15). If the constraint is satisfied, Ψ_i and $\mathcal{E}[PSSNR(i)]$ corresponding to this assignment are calculated and inserted into the second column element of row i of $SSNR$ matrix. The process of incrementing the parity by two symbols, calculating the packet size, checking the packet size constraint, calculating Ψ_i, and $\mathcal{E}[PSSNR](i)$ is repeated till the maximum allowable packet size determined by Constraint (15) is reached. The reason for incrementing the parity by two symbols in each assignment relates to the fact that Ψ is a function of the maximum correctable symbols $t_C = \lfloor \frac{C}{2} \rfloor$ in Equation (6). The rows associated with each packet i where $i \in \{1, \cdots, I\}$ are filled the same way as described above. We now find the optimal parity assignments for each packet maximizing the $\mathcal{E}[SSNR]$ of the overall sequence for a given budget B_T. The problem is divided into I stages. Stage 1 finds the optimal solution for packet 1, stage 2 finds the optimal solution for both packets 1 and 2, stage 3 finds the optimal solution for packets 1, 2, and 3, and so on till reaching stage I in which the optimal solution for the entire set of packets is identified. As illustrated by Fig. 3(b), each stage has a state table ST_K containing the optimal solution for stage K and the information needed to move from stage $K-1$ to stage K. The first column of the state table contains the parity budget in symbols of increment 2. The second column contains the optimal parity assignments for stage K, the third column contains the maximum value of $\mathcal{E}[SSNR]$ for the first K stages, and the fourth column contains the maximum value of $\mathcal{E}[SSNR]$ for the first $K-1$ stages. For $K = 1$, the values of the third and fourth columns of the state table are computed directly from the $SSNR$ matrix. Let ST_1 be the state table of the first stage. For each state value $S_1^{(j)} \in \{0, 2, 4, \cdots, B_C\}$, the packet size is calculated and checked against Constraint (15). If the packet size limit is not reached, the optimal parity assignment is equal to $S_1^{(j)}$ and the optimal value of $\mathcal{E}[SSNR]$ associated with this assignment is equal to element $V(1, h)$ of the $SSNR$ matrix where index h is calculated as $h = \frac{S_1^{(j)}}{2} + 1$. If the maximum allowable packet size is reached and corresponds to $S_1^{(j)} = S_1^{(+)}$, the optimal parity assignment for that state value and state values larger than $S_1^{(+)}$ is equal to $S_1^{(+)}$. Further, the optimal value of $\mathcal{E}[SSNR]$ associated with this assignment is equal to $V(1, h)$ where $h = \frac{S_1^{(+)}}{2} + 1$. For stage K with $1 < K \le I$, the information of stage $K-1$

is used to determine the optimal assignments. Let ST_K be the state table for stage K with $1 < K \leq I$. For each state value $S_K^{(j)} \in \{0, 2, 4, \cdots, B_C\}$, the packet size is calculated and checked against Constraint (15). If the packet size limit is not reached, the valid parity assignments for state value $S_K^{(j)}$ are $\{0, 2, \cdots, S_K^{(j)}\}$. If the maximum allowable packet size is reached and corresponds to $S_K^{(j)} = S_K^{(+)}$, the valid parity assignments for state value $S_K^{(+)}$ and state values larger than $S_K^{(+)}$ are $\{0, 2, \cdots, S_K^{(+)}\}$. For each assignment, the values of $\mathcal{E}[SSNR]$ for stage K and stage $K-1$ are calculated. The maximum value of $\mathcal{E}[SSNR]$ in the third column identifies the index of the row associated with the optimal assignment in stage K. The process is repeated till reaching the last stage, I. At the end of this stage, state table ST_I identifies the solution to the optimization problem. The row containing the largest value of $\mathcal{E}[SSNR]$ identifies the maximum overall value of $\mathcal{E}[SSNR]$ and the optimal parity assignment for packet I in its third and second column positions, respectively. To identify the optimal parity assignment for packet $I-1$, we find the optimal value of $\mathcal{E}[SSNR]$ in the previous stage from the fourth column position of the optimal row of state table ST_I. Then, we use that value to search the values of the third column of state table $I-1$. The corresponding parity assignment of the second column of state table $I-1$ is the optimal parity assignment for stage $I-1$. We continue the process till we find the optimal parity assignments of all packets. At the end of this section, it is important to note that the time complexity of our proposed dynamic programming algorithm is in the order of $\mathcal{O}(I.B_C)$. The worst case time complexity of the algorithm is $\mathcal{O}(I_{max}.B_C)$ where I_{max} represents the number of packets associated with a minimum packet length necessary to keep the shortest single frame. While the worst case time complexity remains much better than $\mathcal{O}(B_C^2)$, even a quadratic complexity is much better than an exponential complexity associated with an exhaustive search algorithm. In addition, the proposed algorithms of [9], [10], and [14] introduce a linear time complexity in the order of $\mathcal{O}(B_C)$. However, the performance of those algorithms is sub par to our proposed algorithm as shown in Section IV.

B. Protecting against both Bit Errors and Packet Erasures

Similar to the previous subsection, we propose the use of an RS FEC code at the link layer to mitigate the loss effects for this subsection. However unlike the previous subsection, we design the RS FEC code such that it can jointly mitigate the effects of random bit errors and packet erasures. In order to do so, we form the packets horizontally while forming the coding blocks vertically. An RS code with C parity symbols can correct up to N_{err} symbol errors and N_{ers} symbol erasures for as long as $2N_{err} + N_{ers} \leq C$ [8]. According to the discussion of [8], the probability of channel coding block loss $\Psi(L_m, C_m)$ is described as:

$$\begin{aligned}\Psi(L_m, C_m) = 1 - \textstyle\sum_{q=0}^{L_m} p\left(N_{err} \leq \lfloor \frac{C_m - q}{2} \rfloor | N_{ers} = q\right) \\ \times P_{ers}(L_m, q) u(C_m - q),\end{aligned} \tag{16}$$

where N_{err} is the number of non-overlapping symbol errors, L_m is the size of block m in symbols, $P_{ers}(L_m, q)$ is the probability of q symbol erasures out of L_m transmitted symbols, and $u(C_m - q)$ is the unit step function equals to 1 if $q \leq C_m$ and 0 if $q > C_m$. In this case, notice that a packet is erased if one or more of its symbols are erased. Further, losing a packet is translated to losing a symbol per each coding block due to the formation of the grid of blocks. Thus, $P_{ers}(L_m, q)$ is calculated as

$$P_{ers}(L_m, q) = \tbinom{L_m}{q} (P_{sym})^q (1 - P_{sym})^{L_m - q}. \tag{17}$$

It follows that

$$\begin{aligned} p\left(N_{err} \leq \lfloor \frac{C_m - q}{2} \rfloor | N_{ers} = q\right) = \\ \textstyle\sum_{j=0}^{\lfloor \frac{C_m - q}{2} \rfloor} p(N_{err} = j | N_{ers} = q). \end{aligned} \tag{18}$$

If packets are sufficiently large, symbol errors can be considered independent and as a result

$$\begin{aligned} p(N_{err} = j | N_{ers} = q) = \tbinom{L_m - q}{j} (1 - P(s,s))^j \\ \times (P(s,s))^{L_m - q - j}. \end{aligned} \tag{19}$$

Having described the channel coding approach, the main objective of the optimization problem of this section is to find the optimal parity assignment of each vertical channel code maximizing the quality of received audio clips in the presence of random bit errors and packet erasures. Once more, we use $SSNR$ as defined in Equation (7) for performance evaluation. Defining Frame Segmented SNR ($FSSNR$) as the Segmented SNR for one frame, we have

$$\begin{aligned} FSSNR(m) = \log \\ \left\{1 + \frac{\sum_{n=1}^{N} x^2(mN+n)}{\sum_{n=1}^{N} [y(mN+n) - x(mN+n)]^2 + \delta}\right\}. \end{aligned} \tag{20}$$

We note that the summation in the denominator of (20) represents the distortion D between the received and the transmitted frames measured in terms of MSE. Thus, the $SSNR$ of an audio sequence can also be represented in terms of $FSSNR$ as

$$SSNR = \tfrac{10}{M} \textstyle\sum_{m=0}^{M-1} FSSNR(m). \tag{21}$$

If frame m is received successfully, $FSSNR(m)$ is expressed as

$$FSSNR(m) = \log \left\{1 + \frac{\sum_{n=1}^{N} x^2(mN+n)}{D(s) + \delta}\right\}. \tag{22}$$

Notice that in this case, the denominator of Equation (20) represents the source coding distortion $D(s)$. In the event of a frame loss, we use the IBR algorithm to represent a lost frame. More specifically, $FSSNR$ is calculated using Equation (20) after calculating distortion by computing the MSE between the original reference sample values of the frames $x(n)$ and the sample values of the frames used in the error concealment process. Hence, the value of $E[FSSNR(m)]$ for a frame m is expressed as

$$\begin{aligned}\mathcal{E}[FSSNR(m)] \;=\; & (1-\Psi_m)\log \\ & \left\{1+\tfrac{\sum_{n=1}^{N} x^2(mN+n)}{D(s)+\delta}\right\}+\Psi_m \log \\ & \left\{1+\tfrac{\sum_{n=1}^{N} x^2(mN+n)}{\sum_{n=1}^{N}[y(mN+n)-x(mN+n)]^2+\delta}\right\}.\end{aligned} \tag{23}$$

Equally, we can express

$$\mathcal{E}[SSNR] \;=\; \tfrac{10}{M}\ \textstyle\sum_{m=0}^{M-1}\mathcal{E}[FSSNR(m)]. \tag{24}$$

Consequently, the optimization problem is given by

$$\max_{(C_0,\cdots,C_{M-1})} \qquad \mathcal{E}[SSNR] \tag{25}$$

$$\text{Subject To}: \qquad \textstyle\sum_{m=0}^{M-1}\ C_m \leq B_C \tag{26}$$

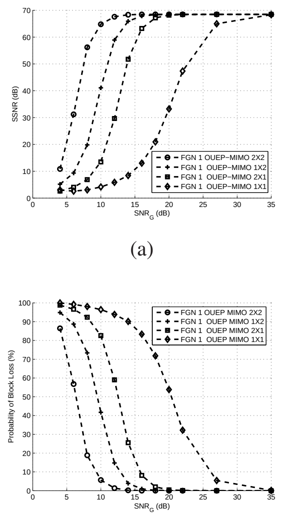
$$0 \leq C_m + R_m < 2^s - 1,\ \ \forall m \tag{27}$$

We note that s the symbol size of block m is chosen such that the block size L_m, consisting of frame payload symbols R_m and parity symbols C_m assigned to that frame, does not exceed the maximum RS block size of 2^s-1. Further, B_C the parity budget allocated to transmit the audio clip equals to $B_C = B_T - B_R - B_H$ where B_T is the total budget allocated to transmit the audio clip, B_R is the payload budget (the size of audio frames), and B_H is the packetization overhead. We calculate $B_H = (2^s-1) * H$ where 2^s-1 is the maximum number of packets of the audio clip corresponding to the maximum block size of a frame. H is the sum of the UDP/RTP/IP compressed header size and the header parity symbols added to protect the header against bit errors. As before, the amount of parity allocated to transmit the audio sequence is upper bounded by the maximum parity that the channel decoding buffer at the receiver can accommodate. To solve the optimization problem, we follow the same procedure described earlier. However, the way we construct the $SSNR$ matrix is a little bit different. The values of $\mathcal{E}[FSSNR]$ for each frame corresponding to all possible parity symbol assignments are calculated and inserted into the $SSNR$ matrix. As illustrated in Fig. 3(a), denote the values of this matrix as $V(r,w)$ where r is the row index and w is the column index. Consider an audio sequence in which the number of rows of the $SSNR$ matrix is equal to M and each row corresponds to one frame. In order to find the row elements of the $SSNR$ matrix associated with frame m, we simulate a loss event for frame m and conceal it using the IBR algorithm. Then, we calculate the distortion measured in terms of MSE between the original frames and the frames used in the IBR concealment algorithm. Next, we use the distortion measure to calculate $\mathcal{E}[FSSNR(m)]$ which evaluates the perceptual sensitivity of frame m. To calculate $\mathcal{E}[FSSNR(m)]$, we calculate the probability of losing frame m for all possible parity assignments. More specifically, $\mathcal{E}[FSSNR(m)]$ is calculated using Equation (23) for each parity assignment in the set $\{0,1,\cdots,B_C\}$. The first column element of row m of the $SSNR$ matrix is set as the value of $\mathcal{E}[FSSNR(m)]$ with a parity assignment of zero. Notice that in this problem, the number of parity symbols is incremented by one symbol not by two as described in the first optimization problem. Then the block size associated with this assignment is calculated and compared against Constraint (27). If the constraint is satisfied, Ψ_m and $\mathcal{E}[FSSNR(m)]$ corresponding to this assignment are calculated and inserted into the second column element of row m of $SSNR$ matrix. The process of incrementing the parity by one symbol, calculating the block size, checking the block size constraint, calculating Ψ_m, and $\mathcal{E}[FSSNR](m)$ is repeated till the maximum allowable block size determined by Constraint (27) is reached. The rows associated with each block m where $m \in \{0,\cdots,M-1\}$ are filled the same way as described above. Next using dynamic programming, we find the optimal parity assignments of each block maximizing the $\mathcal{E}[SSNR]$ of the overall sequence for a given parity budget B_C.

IV. Performance Evaluation

In this section, we present our performance evaluation results based on our proposed framework and considering the two different cases of wireless channels. We consider the transmission of MPEG-4 encoded sequences and conduct our experiments on a number of music audio sequences that vary in characteristics and lengths. The first group of short sequences with which we experiment have an average duration of 30 seconds. Sample audio sequences belonging to this category include trpt21-2, horn23-2, vioo10-2, and sopr44-1 [12]. The second group of sequences with which we experiment have average durations of 3 minutes or more. Sample audio sequences belonging to this category include Hips don't Lie and La Tortura by Shakira. For all sequences, the sampling rate is $48K$ samples/second, the sample size is 16 bits, and each frame is assumed to have 1024 samples [12]. All the sequences are encoded and decoded at 64 Kb/s.

Our protocol stack utilizes IP, UDP, and RTP resulting in a header size of 40 bytes. However, in order to reduce the transmission overhead, we use compressed packet headers reducing the size of the header to 5 bytes [7]. We generically emulate the effects of PHY and MAC layers loss relying on the two-state GE model and the effects of the NETWORK layer loss relying on the two-state G model. We protect packets using RS codes utilizing a symbol size of s bits. For the case of Section III-A, the symbol size s is chosen such that the combined size of payload, header, and parity in symbols does not exceed the maximum allowable packet size determined by Equation (15). We choose $s \in \{8, 9, 10\}$ bits for FGNs $J \in \{1, 2, 4\}$, respectively. We use a symbol size of $s = 8$ bits allowing a maximum block size of up to 255 bytes for the case of Section III-B. The transition probabilities of the GE model are set as $\gamma = 0.99875$ and $\beta = 0.875$ representing average burst lengths of $1/(1-\gamma) = 800$ and $1/(1-\alpha) = 8$ bits for state G and B states, respectively. Further, we consider $SNR_G = 10SNR_B$ to differentiate between the qualities of the transmission link in state G and B. We consider four different MIMO configurations representing improved SER characteristics of the transmission link in an ascending order. They are namely (1) single-transmit single-receive (1×1) utilizing MRC, (2) double-transmit single-receive (2×1) utilizing STBC, (3) single-transmit double-receive (1×2) utilizing MRC, and (4) double-transmit double-receive (2×2) utilizing STBC. For the G model, we choose the value of $\gamma = 0.99875$ and calculate $\beta = 2 - \gamma - ((1-\gamma)/P_{sym})$ for different values of probability of symbol loss P_{sym} [22]. At the receiving side, a packet is passed from the MAC layer to upper layers for decoding if not corrupted. In the case of having a corrupted frame, the last received frame is used to conceal the corrupted frame. Our performance evaluation experiments generate three main sets of curves. The first set of curves capture $\mathcal{E}[SSNR]$ measured in dB for the entire received audio sequence as a function of SNR_G also measured in dB. The second set of curves capture the Perceptual Evaluation Audio Quality ($PEAQ$) for the entire received audio sequence as a function of SNR_G. The third set of curves measure the average block loss in percent as a function of SNR_G in dB. Every point on each curve indicates an average value taken over 80 experiments. The following remarks are in order with respect to the use of $PEAQ$ in our second set of curves. The $PEAQ$ is a measured widely used in perceptual evaluation of audio performance [2]. The $PEAQ$ measure compares a test audio signal with a reference audio signal and generates a score called the Objective Difference Grade (ODG) that ranges from 0 (indicating the best quality) to -4 (indicating the worst quality). However as reported by [6] and others, the $PEAQ$ measure is known to be only accurate for not highly distorted audio signals. To that end, we performed subjective measures to identify accurate thresholds of measurement in the $PEAQ$ implementation of [3]. Accordingly, we assign an ODG of -4 to any audio signal affected by an average transmission block loss of at least 30%. We assume that the parity budget of each experiment is fixed as it represents the difference between the total available budget and the combined size of data and meta-data. As such, our task is to distribute the parity budget among audio frames, packetize them, and then transmit them over the wireless channel. In our experiments, we utilize a playout buffer and a rate control mechanism capable of guaranteeing a smooth and continuous playout at the receiving end as described on Section II. We conduct our experiments by varying the MIMO configuration, the budget, and the signal-to-noise ratios while utilizing BPSK modulation. We investigate the effectiveness of our framework under different settings as compared to other literature frameworks most closely related to our work. We provide a performance comparison of our framework to which we refer as Optimal Unequal Error Protection (OUEP) with three other UEP algorithms. The first one is an ARQ-based UEP algorithm to which we refer as AUEP. The AUEP algorithm utilized in our work is an improved version of the algorithm of [9], [10], and [14] in which all frames are ordered based on perceptual importance and transmitted upon the availability of budget. Once the full set of frames are transmitted, AUEP continues with retransmitting packets that are reported lost in transmission by the receiving end and in the order of the perceptual importance. The process may get repeated in multiple rounds and comes to an end after all of the remaining budget has been used or frames have missed their playout deadline. The other two algorithms are called Perceptual-Controlled Unequal Error Protection I and II. We refer to these two algorithms as PC-UEPI and PC-UEPII. In PC-UEPI, the parity symbols of a block i are assigned proportional to payload size R_i of the block, i.e., the parity budget B_i of block i is specified as $B_i = \frac{R_i}{\sum_{i=1}^{I} R_i} B_C$ where B_C and I represent the total parity budget and the total number of blocks to be transmitted, respectively. This parity assignment assures that a block with a higher number of data symbols receives a higher number of parity symbols compared to a block with a smaller number of data symbols. PC-UEPII is another version of the algorithm proposed by [9] and [14] in which a block with a higher distortion is considered more important from a perceptual stand point and is protected more compared to a lower distortion block. Accordingly, PC-UEPII assigns parity symbols of a block proportional to the distortion associated with the loss of that block. Thus, the parity budget of block i is specified as $B_i = \frac{D_i}{\sum_{i=1}^{I} D_i} B_C$ where D_i is the MSE distortion of the denominator of Equation (8) applied to block i.



(a)

(b)

Fig. 4. A performance comparison of different MIMO configurations utilized in conjunction with our OUEP algorithm to depict (a) $SSNR$, and (b) the average block loss as functions of SNR_G. The horn23-2 audio sequence, an FGN of 1, and a total budget of $B_T = 92.2KB$ are used.

A. Performance Evaluation Results over Bit Error Only Channels

In this subsection, we discuss the performance evaluation results obtained from transmitting the audio sequences mentioned above over a fading wireless channel subject to bit errors only.

First, we investigate the performance effects of applying OUEP in conjunction with using different MIMO configurations. **Then, we illustrate how performance is improved utilizing our proposed OUEP scheme for different permutations of transmission budgets and MIMO configurations.** Fig. 4 compares the performance of the four MIMO scenarios indicated above for horn23-2 utilizing an FGN of 1. All curves show that the quality of the reconstructed audio improves following an S-shaped pattern as the quality of the channel improves. However, the transitioning segment of an $SSNR$ curve shifts to the left as a MIMO configuration with a better SER characteristic is used. The results are consistent with experiments performed using other FGNs as well as other audio clips.

Next, we investigate the performance effects of the value of FGN. As illustrated by Fig. 5(a) and Fig. 5(b), increasing the value of FGN improves performance. Interestingly, the reason behind this performance improvement is captured by Fig. 5(c). As observed, increasing the value of FGN increases packet size in turn decreasing the total number of packets required to transmit the audio sequence and consequently the packetization overhead. The reduction in packetization overhead is the main reason behind the improvement of performance.

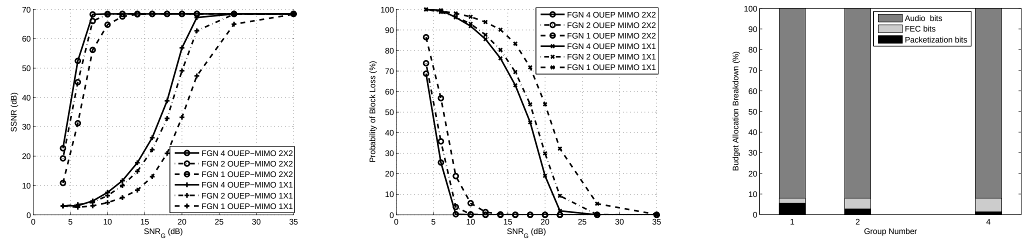
We now provide a performance comparison of our OUEP algorithm with AUEP, PC-UEPI, and PC-UEPII. In all of our experiments, the overall transmission budget is fixed. Fig. 6 compares the performance results of the four schemes for sopr44-1 sequence utilizing the $SSNR$ and $PEAQ$ metrics. While not shown, we have observed similar patterns in the results associated with other audio sequences. The most important observation based on the trend of our results is that not only $SSNR$ but also $PEAQ$ performance of our OUEP scheme is better than the other three schemes independent of the choice of audio clip and for comparable choices of FGN, MIMO configuration, and FEC budget. We have also noticed that the performance of OUEP increases as the overall available budget increases, since increasing the overall budget increases the channel coding budget thereby reducing the probability of block loss.

B. Performance Evaluation Results over Tandem Channels

In this section, we discuss the performance evaluation results of transmitting audio sequences over a tandem channel affected by both bit errors and packet erasures. Recall that in this case our optimization technique jointly protects an audio sequence against both bit errors and packet erasures while maximizing the quality of the received audio. It does so by allocating the available parity budget to audio blocks vertically according to the formation of a grid of symbols. First, we study the performance of OUEP in the presence of different MIMO configurations. Similar to what was observed in the previous section, we anticipate that utilizing a higher quality MIMO link will achieve a lower average block loss and a better $SSNR$ value. Fig. 7 compares the performance results of the Hips don't Lie audio sequence using OUEP, AUEP, PC-UEPI, and PC-UEPII schemes. Once more, we observe that not only $SSNR$ but $PEAQ$ performance of our OUEP scheme is better than the other schemes independent of the choice of audio sequence with different MIMO configurations, different symbol loss rates, and different budgets. The results are, in fact, consistent in all of the conducted experiments using a variety of audio sequences.

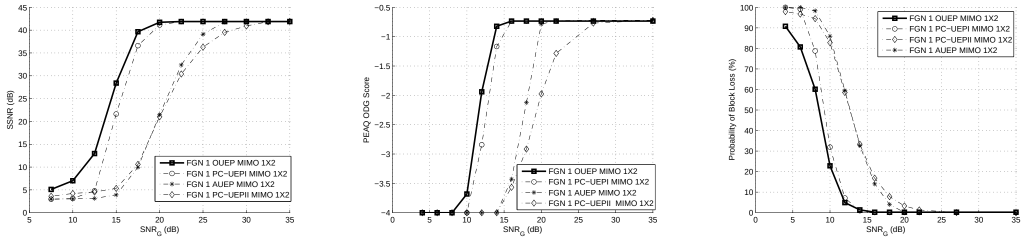
V. Conclusion

In this paper, we presented an optimization framework for transmitting high quality audio sequences that could be generically applied to a number of publicly available audio codecs including BSAC codec implementing the MPEG-4 standard. Our framework introduced apparatus and technique to optimally protect a stored audio sequence transmitted over a wireless channel while considering the packetization overhead of audio



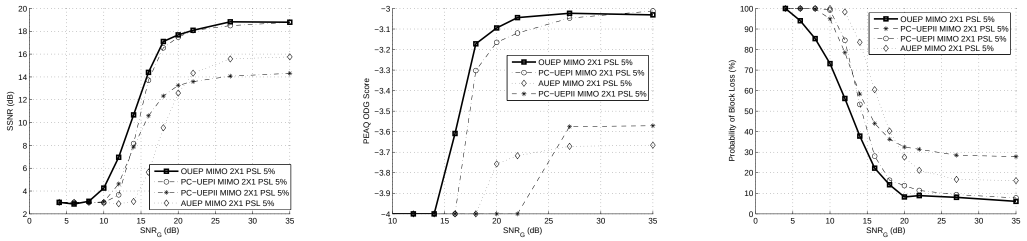
(a) (b) (c)

Fig. 5. A performance comparison of different MIMO configurations and FGNs measured in terms of (a) $SSNR$, (b) the average block loss both as functions of SNR_G. The horn23-2 audio sequence and a total budget of $B_T = 92.2KB$ are used. (c) The budget allocation of horn23-2 audio sequence for FGN=1, 2, 4. For FGN= 1, the audio source bits, the FEC budget, and the packetization overhead constitute 92%, 2.4%, and 5.6% of the total budget of $B_T = 92.2KB$, respectively. For FGN=2, the allocation of the same budget between FEC and packetization overhead is improved to 4.8% and 3.2% while preserving the source bit allocation. For FGN=4, the allocation of FEC and packetization overhead is further improved to 6.6% and 1.4% while preserving the source bit allocation.



(a) (b) (c)

Fig. 6. A performance comparison of OUEP, PC-UEPI, PC-UEPII, and AUEP for sopr44-1 audio sequence illustrating (a) $SSNR$, (b) $PEAQ$, and (c) the average block loss as functions of SNR_G. The experiment is performed for FGN=1, $B_T = 120KB$, and a 1×2 MIMO configuration.



(a) (b) (c)

Fig. 7. A performance analysis of the Hips don't Lie audio sequence illustrating (a) $SSNR$, (b) $PEAQ$, and (c) the average block loss as functions of SNR_G. An average symbol loss probability of 5%, a 2×1 MIMO configuration, and a total budget of $B_T = 1.403MB$ are utilized.

frames. Our framework considered two cases. In the first case, a frame grouping technique was proposed to packetize audio frames and protect them against temporarily correlated bit errors introduced by a fading wireless channel. In this case, each packet was treated as a channel coding codeword. In the second case, a one-dimensional RS coder was applied vertically to a sequence of horizontally formed packets associated with an audio sequence in order to protect the sequence against both bit errors introduced by fading wireless channels and packet erasures introduced by network buffering. Utilizing rate compatible punctured RS codes and dynamic programming, our framework identified the optimal assignment of parity to audio frames according to their perceptual importance such that the Segmented SNR of the received audio sequence was maximized. Numerical results captured the performance advantage of our framework compared to existing techniques proposed in the literature of audio transmission. As a part of our ongoing work, we are investigating the possibility of utilizing network coding schemes as alternatives of error protection and the associated impacts on our optimization scheme.

References

[1] ISO Standard Website, URL http://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_IEC_14496-5_2001_Amd_6_Reference_Software/.

[2] Method for objective measurements of perceived audio quality, in Recommendation ITU-R BS.1387-1, Geneva, Switzerland, 1998-2001.

[3] P. Kabal, Matlab implementation of Perceptual Evaluation of Audio Quality (PEAQ). Available: http://www-mmsp.ece.mcgill.ca/Documents/Software/.

[4] A. Bayya and M. Vis. ”Objective Measures for Speech Quality Assessment in Wireless Communications”. *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1996.

[5] T.H. Cormen, C. E. Leiserson, R.L. Rivest, and C. Stein. *”Introduction to Algorithms”*. MIT press, second edition, 2003.

[6] C. D. Creusere. ”Understanding Perceptual Distortion in Mpeg Scalable Audio Coding”. *IEEE Transactions on Acoustics Speech and Signal Processing*, VOL. 13, NO. 3, pages: 422-431, May 2005.

[7] M. Degermark, M., B. Nordgren, and S. Pink. ”Low-Loss TCP/IP Header Compression for Wireless Networks”. *In Proc. ACM Mobile Computing and Networking (MOBICOM)*, 1996.

[8] F. Etemadi and H. Jafarkhani. ”An Efficient Progressive Bitstream Transmission System for Hybrid Channels with Memory”. *IEEE Trans. on Multimedia*, VOL. 8, NO. 6, pages: 1291-1298, Dec 2006.

[9] E. Hellerud, J.E. Voldhaug, and U.P. Svensson. ”Perceptually Controlled Error Protection for Audio Streaming over IP Networks”. *In Proc. IEEE International Conference on Digital Telecommunications (ICDT)*, 2006.

[10] Z. Li, L. Sun, Z. Qiao, and E. Ifeachor. ”Perceived Speech Quality Driven Retransmission Mechanism for Wireless VoIP”. *In Proc. IEEE International Conference on 3G Mobile Communication Technologies*, 2003.

[11] S.K. Marks and R. Gonzalez. ”Object-Based Audio Streaming Over Error-Prone Channels”. *In Proc. IEEE International Conference on Multimedia and Expo (ICME)*, 2005.

[12] M. Olausson, A. Ehliar, J. Eilert, and D. Liu. ”Reduced Floating Point for MPEG1/2 Layer III Decoding”. *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2004.

[13] C. Perkins, O. Hodson, and V. Hardman. ”A Survey of Packet Loss Recovery Techniques for Streaming Audio”. *IEEE Network*, VOL. 12, NO. 5, pages: 40-48, Sep 1998.

[14] M. Petracca, A. Servetti, and J.C. De Martin. ”Voice Transmission over 802.11 Wireless Networks Using Analysis-By-Synthesis Classification”. *In Proc. IEEE International Symposium on Control, Communications and Signal Processing (ISCCS)*, 2004.

[15] L. Roychoudhuri and E.S. Al-Shaer. ”Adaptive Rate Control for Real-Time Packet Audio Based on Loss Prediction”. *In Proc. IEEE Global Communications Conference (GLOBECOM)*, 2004.

[16] B. Sklar. *”Digital Communications: Fundamentals and Applications”*. Prentice Hall, second edition, 2001.

[17] V. Tarokh, H. Jafarkhani, and A.R. Calderbank. ”Space-Time Block Coding from Orthogonal Designs”. *IEEE Trans. Information Theory*, VOL. 45, NO. 5, pages: 1456-1467, July 1999.

[18] J. Tourrilhes. ”Packet Frame Grouping: Improving IP Multimedia Performance over CSMA/CA”. *In Proc. IEEE International Conference on UniversalPersonal Communications (ICUPC)*, 1998.

[19] W. Wang, S.C Liew, and V.O.K. Li. ”Solutions to Performance Problems in VoIP over 802.11 Wireless LAN”. *IEEE Trans. Vehicular Technology*, VOL. 54, NO. 1, pages: 366-384, Jan 2005.

[20] Y. Wang, A. Ahmaniemi, D. Isherwood, W. Cheng, and D. Huang. ”Content-based UEP: A new Scheme for Packet Loss Recovery in Music Streaming”. *In Proc. ACM International Conference on Multimedia*, 2003.

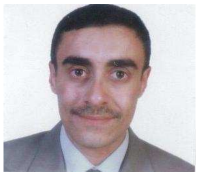
[21] A. Xu, W. Woszczyk, Z. Settel, B. Pennycook, R.Rowe, P.Galanter, J.Bary, G.Martin, J. Corey, and J. Cooperstock. ”Real-Time Streaming of Multichannel Audio Data over Internet”. *Journal of the Audio Engineering Society*, VOL. 48, NO. 7, pages: 627-639, July-Augest 2000.

[22] J. Yee and E. Weldon. ”Evaluation of the Performance of Error-Correcting Codes on a Gilbert Channel”. *IEEE Trans. on Communications*, VOL. 43, NO. 8, pages: 2316-2323, Aug 1995.

[23] H. Yousefi’zadeh, H. Jafarkhani, and F. Etemadi. ”Transmission of Progressive Images Over Noisy Channels: An End-to-End Statistical Optimization Framework”. *IEEE Journal of Selected Topics in Signal Processing*, VOL. 2, NO. 2, pages: 220-231, April 2008.

[24] H. Yousefi’zadeh, H. Jafarkhani, and M. Moshfeghi. ”Power Optimization of Wireless Media Systems with Space-Time Block Codes”. *IEEE Trans. Image Processing*, VOL. 13, NO. 7, pages: 873-884, July 2004.

[25] C.W. Yung, H.F. Fu, C.Y. Tsui, R.S. Cheng, and D. George. ”Unequal Error Protection for Wireless Transmission of MPEG Audio”. *In Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, 1999.



Ala’ Khalifeh (M’98) received the B.S., M.S. degrees in Electrical Engineering from the University of Jordan in 2001, 2004, respectively. He joined the University of California, Irvine in 2005 and currently he is a Ph.D. candidate in the Electrical and Computer Engineering department. Mr. Khalifeh is the recipient of the Fulbright scholarship (2005-2007) sponsored by the Bureau of Educational and Cultural Affairs of the United States Department of State. In April 2008, he was selected as a Pedagogical Fellow (PF) by the University of California, Irvine and received the Alpha Phi Beta Kappa award for his excellent academic and leadership merits. Mr. Khalifeh’s research is in communications technology and networking with particular emphasis on optimal resource allocations for audio and video transmission over wired and wireless networks.

Homayoun Yousefi’zadeh (SM’06) received the B.S. degree from Sharif University of Technology, Tehran, Iran, the M.S. degree from Amirkabir University of Technology, Iran, and the Ph.D. degree from University of Southern California, Los Angeles, all in electrical engineering, in 1989, 1993, and 1997, respectively. He is currently with the faculty of the Department of Electrical Engineering and Computer Science at the University of California, Irvine. He also holds a consulting scientist position at the Boeing company. Most recently, he was the CTO of TierFleet, Inc., working on distributed database systems, a Senior Technical and Business Manager at Procom Technology, focusing on storage networking, and a Technical Consultant at NEC Electronics, designing and implementing distributed client-server systems. Dr. Yousefizadeh has served as the chairperson of the systems management workgroup of the Storage Networking Industry Association (SNIA), and a member of the scientific advisory board of the Integrated Media Services Center (IMSC) at the University of Southern of California. He is an associate editor for the IEEE Communications Letters, an editor of the IEEE Wireless Communications, and has been with the TPC of various IEEE conferences.