A Hybrid Media Transmission Scheme for Wireless VoIP

Ala' Khalifeh Homayoun Yousefi'zadeh
Department of EECS
University of California, Irvine
[akhalife,hyousefi]@uci.edu

Abstract

In this paper, we propose an optimization framework for real-time voice transmission over wireless tandem channels prone to both bit errors and packet erasures. Utilizing a hybrid media dependent and media independent error correction scheme, our proposed framework is capable of protecting voice packets against both types of errors. For each group of frames associated with one speech spurt, the framework finds the optimal parity assignment of each voice frame according to its perceptual importance such that the quality of the received group of frames is maximized. Our performance evaluation results show that the proposed scheme outperforms a number of alternative schemes and has a low computational complexity.

I. Introduction

With the advent of wireless technologies such as WiFi, WiMax, and LTE, the use of Voice over IP (VoIP) has extended to wireless networks. However, the majority of VoIP tools such as Robust Audio Tool (RAT) [1] and FreePhone [2] are designed to work over wired networks. Consequently, they only provide mechanisms to protect voice packets against packet erasures caused by network congestion. Transmission over wireless networks is prone to two types of errors: a) packet erasures caused by network congestion, and b) bit errors caused by wireless media such as fading and interference. As such, using traditional VoIP tools may not efficiently protect voice packets against tandem loss. In addition, the algorithms used for protecting voice packets have to be run in real-time with a minimal delay and treat those packets according to their perceptual importance. In what follows, we list some of the literature work most closely related to our work. In [15], the authors present a framework of voice transmission over congested networks. The authors of [11] present rate adaptation algorithms for the Adaptive Multi-Rate (AMR) voice codec. The authors in [8] propose a media dependent unequal error protection scheme in which the most important voice packets are duplicated. In our previous works of [9] and [10], we propose optimization frameworks of transmitting stored audio sequences over tandem channels. In this paper, we propose a hybrid protection scheme for the transmission of voice packets over tandem channels that jointly utilizes media dependent and media independent protection schemes. The main difference between this paper and our previous works is that the algorithms proposed in our previous works were designed for transmitting stored audio but not live VoIP content. Such sophisticated yet more complex dynamic programming algorithms proposed before are not appropriate for real-time voice transmission. Thus, this work focuses on real-time voice transmission schemes with low processing overhead and minimal end-to-end transmission delay. The rest of this paper is organized as follows. Section II describes the background information. In Section III, we describe our voice transmission framework. Section IV formulates and solves the optimization problem associated with the framework of Section III. In Section V, we provide our performance evaluation results. Finally, Section VI concludes the paper.

This work was supported by a research contract from the Boeing Company.

II. BACKGROUND

In this section, we provide a brief description of the wireless channel model used in our analysis, followed by an overview of different error correction mechanisms used for protecting voice frames against transmission errors.

A. Wireless Channel Model

Transmitting data over a wireless channel is susceptible to two types of errors, bit errors caused by PHYSICAL and DATA LINK layers of a wireless channel and packet erasures caused by buffer overflows of the NETWORK layer. In our analysis, we model bit errors introduced by the wireless channel utilizing the two-state Gilbert-Elliott (GE) model. In this model, random bit errors are described by a two-state Markov chain. The good state referred to as state G has a self transitioning probability γ, while the bad state referred to as state B has a probability of self transitioning β. 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 equal to

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

We refer the reader to [16] for the details of calculating the recursive probabilities $P(t, q, G)$ and $P(t, q, B)$. For modeling packet erasures, we use the Gilbert (G) model which is considered to be a special case of the GE model with $\varepsilon_G = 0$ and $\varepsilon_B = 1$. We apply the GE model at the bit level for the sequence of bits that form consecutive symbols and in turn packets. A symbol is lost if one or more bits in it are lost. A packet can be recovered if the RS code can correct its erroneous symbols. Further, we apply the G model to the capture network layer packet loss. We denote the average probability of packet loss by P_{ers} and conduct our experiments with different values of P_{ers} resulting from changing β where $\beta = 2 - \gamma - (1 - \gamma)/P_{ers}$.

B. VoIP Error Correction Schemes

A transmitted voice bitstream may typically be protected using two different schemes to which we refer as media dependent and media independent error correction schemes. In what follows, we briefly describe these schemes.

Media Dependent Error Correction: As shown in Fig. 1(a), a Media Dependent Error Correction (MD-EC) scheme [14] alleviates the effects of packet loss by piggy-backing a lower quality copy of the contents of packet i in the subsequent packet $i + 1$. At the receiving side, the receiver waits till either the original packet or its lower quality copy arrive. If both of them arrive successfully, the receiver uses the high quality copy at the decoding stage. Otherwise, the receiver uses the low quality copy as a replacement of the high quality copy if it solely receives the low quality copy. The main advantage of this scheme is its ease of implementation for real-time voice as neither the transmitter nor the receiver have to buffer a large number of packets. However, this error protection scheme is only efficient when the network is prone to packet loss, while it performs poorly when bit errors are present.

Media Independent Error Correction: Media Independent Error Correction (MI-EC) schemes such as Reed-Solomon codes are widely used to protect and correct multimedia bitstreams against bit errors. As shown in Fig. 1(b), an MI-EC scheme utilizing an RS code forms a block of n symbols with each symbol consisting of s_m bits and $n = 2^{s_m} - 1$. An encoded $RS(n, k)$ block contains k data symbols and $C = n - k$ parity symbols. An $RS(n, k)$ block can correct as many as $t_C = \lfloor \frac{C}{2} \rfloor$

symbol errors. The main advantage of an MI-EC scheme is its bandwidth efficiency and being able to protect voice packets against bit errors.

Hybrid Media Dependent and Media Independent Error Correction: As illustrated in Fig. 1(c), we propose to use a hybrid media dependent and media independent error correction scheme to which we refer as Hybrid Media Error Correction (HM-EC). Our proposed HM-EC scheme combines the advantages of both MI-EC and MD-EC schemes as it utilizes MI-EC for protecting the symbols of each packet against bit errors and MD-EC for protecting against packet erasures. However, implementing HM-EC scheme raises the following questions: What is the optimal encoding rate of each voice frame such that the quality of the received stream is maximized? What is the optimal size of each packet such that more protection is applied to more important frames? We attempt at addressing the above questions and related issues in our proposed framework. We conclude this section by noting that in all of the above schemes, each high quality voice frame along with the low quality frame attached to it form a single packet.

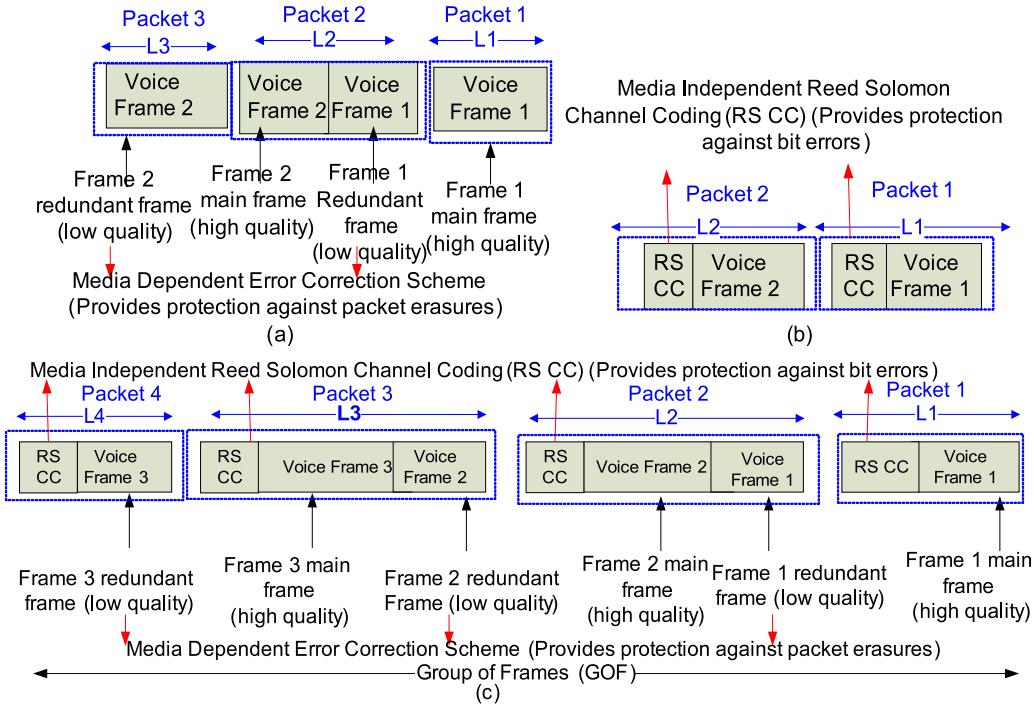


Fig. 1. An illustration of (a) media dependent, (b) media independent, and (c) hybrid media error correction schemes.

III. A Description of Framework Design

Our proposed framework utilizes HM-EC making it robust against both bit errors and packet erasures introduced by wireless channels. It is also important to note that the proposed framework requires a small amount of buffer space and introduces a very low processing delay. Fig. 2 shows a block diagram of our proposed framework. The voice signal is first passed through a Voice Activity Detection (VAD) module which suppresses silence periods, and produces a series of speech frames. We use the VAD algorithm proposed by [13]. To reduce the buffering and processing time, speech frames are divided into Groups of Frames (GOFs). Consisting of a certain number of frames, each GOF is analyzed, encoded, and transmitted over wireless channel. Selecting the size of a GOF represents a design parameter and is discussed in Section V. Each frame in a GOF is analyzed and

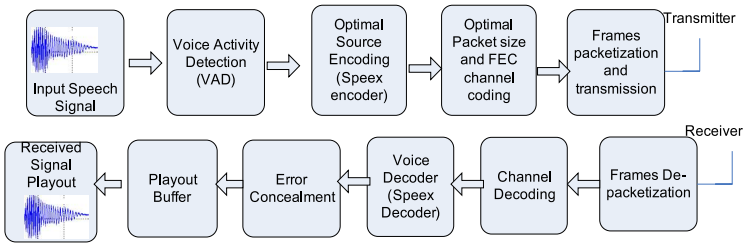


Fig. 2. A block diagram of the proposed framework.

encoded using the Speex encoder. We note that the Speex encoder is used in the Average Bit Rate (ABR) mode in order to determine the optimal encoding rate of each frame. In the ABR mode, Speex encodes voice frame i using an encoding rate r_i selected such that the average encoding rate of the encoded sequence is equal to a specific target bit rate r_{avg}. We set up the target bit rate to a high value of r_{hi} for main frames, and to a low value of r_{low} for redundant copies. We observe that higher encoding rates are used to encode frames of higher perceptual importance. We refer the reader to [15] for the details of Speex encoding algorithm and its selection of optimal encoding rate. After encoding all of standard and redundant frames of a GOF, each frame is packetized in a different packet, while the redundant copy of each frame is piggy-backed to the next packet. As shown in Fig. 1(c), the number of packets needed to transmit a group of voice frames of size M is equal to $M + 1$. Further, packet sizes are not the same for all frames but depend on source and channel coding symbols of each packet. As such, the optimization problem described in Section IV attempts at finding the optimal channel coding symbols of each packet such that the quality of the received GOF is maximized. Once the number of channel coding symbols for each packet of GOF are identified, they are transmitted over the wireless channel. Once the voice frames belonging to a GOF are received at the receiving side, the receiver attempts at correcting symbols errors in the received packets. As a result, packets are either dropped or successfully corrected. Recovered packets are then decoded and used to conceal the lost packets. The error concealment and decoding process is performed as follows. If all packets belonging to the same GOF are correctly received, the decoder only decodes high quality frames and discards the redundant frames. If a packet is lost, the receiver checks to see whether it received the low quality copy of that packet. If received, the low quality copy replaces the lost packet. If both high and low quality copies of the frame are lost, then the receiver uses the the Insertion Based Error Concealment (IBEC) [12] method which simply replaces the lost frame within the packet by the last correctly received frame. Hence, each frame is concealed by either its low quality copy or by the frame prior to it in the sequence. Distortion is calculated in terms of Log Spectral Amplitude Distortion (LSAD) [6] between the original voice signal and the one used in error concealment, which will be used later in the optimization problem described in Section IV as a measure of the voice perceptual importance.

IV. Optimization Formulation and Solution

The main objective of the optimization problem is to find the optimal parity assignment of each packet maximizing the quality of the received GOF under random bit errors and packet erasures. We use LSAD defined below as our metric of performance evaluation.

$$LSAD[X_l(k), \hat{X}_l(k)] \;=\; [\log A_l(k) - \log \hat{A}_l(k)]^2. \tag{2}$$

In the equation above, $A_l(k)$ and $\hat{A}_l(k)$ are the magnitudes of the reference signal $X_l(k)$ and the distorted signal $\hat{X}_l(k)$ both measured in the frequency domain. The main objective is to minimize

the expected distortion of the received voice frames in turn maximizing the quality of the received signal. Defining Frame Log Spectral Amplitude Distortion ($FLSAD$) as the distortion caused by losing one frame, we have

$$FLSAD(m) \ = \ \sum_{n=1}^{N}[\log A_m(mN+n) - \log \hat{A}_m(mN+n)]^2 + f_m(r_i), \tag{3}$$

where N is the number of samples per frame, m is the frame number with $m \in \{1, 2, .., M\}$, and M is the GOF size. Further, $A_m(.)$ and $\hat{A}_m(.)$ represent the spectral envelop of the transmitted and received signal after applying the error concealment algorithm, respectively. Finally, $f_m(r_i)$ is the distortion of the source encoder expressed as:

$$f_m(r_i) \ = \ f_o(m)2^{2k(r_{max}-r_i)}, \tag{4}$$

In Equation (4), $f_o(m)$ is calculated by measuring the LSAD of that frame with respect to a silence frame presented by a zero output signal. In addition, r_i is the encoding rate used by the Speex encoder and r_{max} is the maximum encoding rate. Interestingly, the authors of [7] found out that Equation (4) constitutes a good and tight upper bound on the real rate-distortion curve of any voice encoder by experimentally identifying the value of parameter k for that encoder. One can notice that the number of packets required to transmit a GOF of size M is equal to $M+1$ as shown by Fig. 1(c). However, the value of $\mathcal{E}[FLSAD(m)]$ is calculated for the first M packets only since each packet of the first M packets contains a new voice frame while the last packet contains a redundant copy of the last frame in the GOF. In order to calculate $\mathcal{E}[FLSAD(m)]$ of a packet m, we consider two cases. Case 1 which calculates the $\mathcal{E}[FLSAD(m)]$ in the event of successfully receiving the packet is denoted as $\mathcal{E}[FLSAD(m)_{no-loss}]$, while Case 2 which calculates $\mathcal{E}[FLSAD(m)]$ in the event of packet loss is denoted as $\mathcal{E}[FLSAD(m)_{loss}]$.

Case 1: If frame m is received successfully, then $\mathcal{E}[FLSAD(m)_{no-loss}] \ = \ (1 - \Psi(m))f_m(r_i)$ where $f_m(r_i)$ is the source distortion due to encoding of the voice frame using the encoding rate r_i as determined by the Speex encoder. Further, $\Psi(m)$ the probability of losing packet m is calculated as

$$\Psi(m) \ = \ \min(1, \Psi_{err}(m) + \Psi_{ers}(m)), \tag{5}$$

In the equation above, $\Psi_{err}(m)$ and $\Psi_{ers}(m)$ are probabilities of losing packet m due to bit errors and packet erasure, respectively. In what follows, we describe how each of these two terms are calculated. First, we note that $\Psi_{err}(m)$ is calculated according to the discussion of [9] as:

$$\Psi_{err}(L_m, t_C) \ = \ \sum_{j=0}^{L_m - t_c - 1} P(t, j) \ = \ \sum_{j=0}^{L_m - t_C - 1} \binom{L_m}{j} (1 - P(s,s))^{L_m - j} (P(s,s))^j, \tag{6}$$

In Equation (6), L_m is the size of packet m in symbols, t_C is the RS code error correction capability identified as $t_C = \lfloor \frac{C_m}{2} \rfloor$, C_m is the number of parity symbols assigned to packet m, and $P(s,s)$ is calculated recursively using Equation (1). Next, we note that $\Psi_{ers}(m) = P(GOF+1, GOF)$ where $P(GOF+1, GOF)$ is calculated recursively using Equation (1) by setting $\varepsilon_G = 0$ and $\varepsilon_B = 1$. Notice that $\Psi(m)$ in Equation (5) is the sum of the two probabilities since the two types of errors are mutually exclusive.

Case 2: In the event of losing packet m with $m \in \{1, 2, .., M\}$, $\mathcal{E}[FLSAD(m)]$ is calculated as

$$\begin{aligned} \mathcal{E}[FLSAD(m)_{loss}] \ &= \ \Psi(m)(1 - \Psi(m+1))f_m(r_{min}) + \\ &\Psi(m)\Psi(m+1)(f_m(r_i) + \textstyle\sum_{n=1}^{N}[\log A_m(mN+n) - \log \hat{A}_m(mN+n)]^2), \end{aligned} \tag{7}$$

where $f_m(r_{min})$ is the distortion caused by using the low quality copy of frame m and calculated from Equation (4). After calculating $\mathcal{E}[FLSAD(m)_{no-loss}]$ and $\mathcal{E}[FLSAD(m)_{loss}]$, we

have $\mathcal{E}[FLSAD(m)_{total}] = \mathcal{E}[FLSAD(m)_{loss}] + \mathcal{E}[FLSAD(m)_{no-loss}]$. Consequently, the optimization problem minimizes $\mathcal{E}[LSAD_{total}]$ of the GOF denoted by $\mathcal{E}[LSAD_{GOF}]$ as

$$\begin{array}{ll} \min\limits_{(C_1,\cdots,C_{M+1})} & \mathcal{E}[LSAD_{GOF}] \;=\; \frac{1}{10\cdot M}\sum_{m=1}^{M}\mathcal{E}[FLSAD(m)_{total}] \\ \text{Subject To :} & \sum_{m=1}^{M+1} C_m \le B_C \\ & 0 \le L_m = C_m + R_m + H \le 2^{s_m} - 1.\ \ \forall m \end{array} \tag{8}$$

We note that s_m, the symbol size of packet 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 do not exceed the maximum RS block size of $2^{s_m} - 1$. The parity budget B_C is calculated as $B_C = B_T - \sum_{m=1}^{M+1} R_m + (M+1)H$ where B_T is the budget allocated to transmit the GOF and H is the sum of the UDP/RTP/IP compressed header size. Once more, the target of the optimization problem is to find the parity assignment C_m of each packet such that the quality of the received voice frames of a GOF is maximized. To solve the optimization problem described in Equation (8), we construct an Optimal Search Tree (OST) that limits the search space to the set of feasible points satisfying the optimization constraints. As depicted by Fig. 3, we define the OST as a non-binary

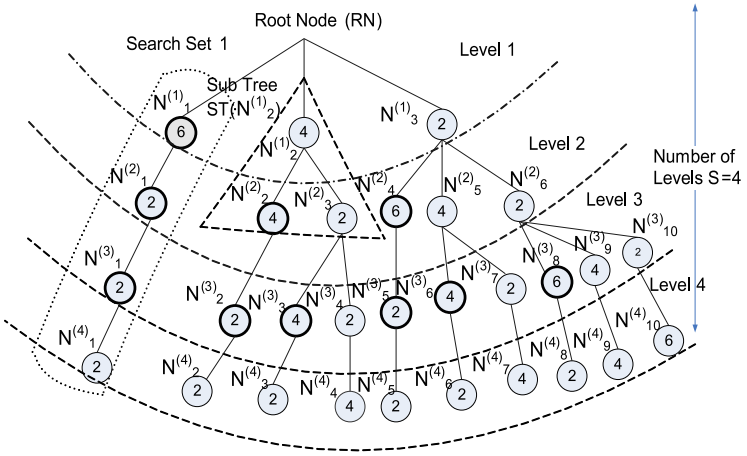


Fig. 3. An illustration of the formation of the optimal search tree for sample values of $B_C = 12$ symbols, a GOF size of $M = 3$, and a tree depth of $S = 4$ packets.

search tree that has the following properties:

1) The tree depth or the number of levels S is equal to $M+1$ where M is the number of packets in the GOF. For example, Fig. 3 shows an OST with $S = 4$ that corresponds to a $M = 3$. Notice that each level s with $s \in \{1, 2, ..., S\}$ contains a set of nodes ($N_j^{(s)}$) with $j \in \{1, 2, ..., N_t^{(s)}\}$ where $N_t^{(s)}$ is the total number of nodes at level s. As described later, $N_t^{(s)}$ is determined while building the tree.
2) A SubTree (ST), named using its parent node, is a tree that contains a parent node with all of its first level children. For example, in Fig. 3, $ST(N_2^{(1)})$ is composed using the parent node $N_2^{(1)}$ with its children $N_2^{(2)}$, $N_3^{(2)}$.
3) A pilot node $\mathcal{P}$ is the first left child node of an ST. For example, Node $N_1^{(1)}$ is a pilot node of the root node $ST(RN)$. For further clarification, all pilot nodes are marked in Fig. 3 using circles with bold borders.
4) The Search Set (SS) is defined as the set of nodes chosen such that each node belongs to a separate level and the set of selected nodes span the tree from the leaf node to the Root Node

(RN). For example, in Fig. 3 SS(1) is composed of $N_1^{(1)}, N_1^{(2)}, N_1^{(3)}$, and $N_1^{(4)}$. Notice that the total number of Search Sets in Fig. 3 is equal to 10.

5) The values of the parity assignment of each node shown inside the circles are chosen such that: a) The minimum parity value C_{min} is equal to 2 symbols corresponding to the lowest possible value of $t_C = 1$; b) the parity assignment values are multiples of 2 symbols considering the effect of the floor function used in t_C, and c) the sum of parity symbols of each node within the same SS is equal to B_C, e.g, the sum of the parity assignment of the nodes of SS(1) is equal to ($6 + 2 + 2 + 2 = B_C = 12$) in Fig. 3.
6) A valid SS has parity assignments $(C_1, C_2, \cdots, C_{M+1})$ that result in packet sizes $(L_1, L_2, \cdots, L_{M+1})$ satisfying Constraint 8 of the optimization problem.
7) Each node's data structure has the following parameters: (*Key*) which is equal to the parity assignment of that node, (*Parent*) which points to the node's parent on the OST, (Nc) the number of node's children, (Bcr) the remaining parity which is equal to the total parity B_C minus the sum of the parity symbols assigned to the predecessor nodes within the SS. Further, each parent node stores the value of the parity assigned to the pilot node of its subtree to which we refer as Pilot-Parity (*PP*).

Building the OST: A top-down approach is used to build the OST. In such approach, the parent nodes of level s with $s \in \{1, 2, \cdots, S\}$ and their associated subtrees are constructed using Algorithm (1) and Algorithm (2). Then, the children nodes of each parent node are constructed using Algorithm (3). It is also important to note that while building the OST, a two dimensional doubly linked list referred to as *LevelNodeList* is used to save the pointers of nodes' locations on the tree of each level, which reduces the complexity of building and traversing the tree.

Searching the OST: Once the OST is constructed, it is traversed vertically using Algorithm (4). In this algorithm, the tree is traversed using a bottom-up approach. One starts from each leaf node and goes up to the parent nodes of the first level $s = 1$. Notice that the number of possible values of SS is equal to the number of leaf nodes. While traversing the tree, the algorithm determines the feasible SS values which result in a packet size satisfying Constraint 8 of the optimization problem. As such, the output of this algorithm specifies all valid parity assignments for each packet resulting in a valid packet size. For example, consider SS(1) in Fig. 3 which consists of parity assignments (6,2,2,2) symbols for packets (1,2,3,4) with packet sizes L_1, L_2, L_3, L_4, respectively. The search algorithm will check if these parity assignments result in valid packet sizes $L_{T1}, L_{T2}, L_{T3}, L_{T4}$ where $L_{T1} = L_1 + 6, L_{T2} = L_1 + 2, L_{T3} = L_3 + 2, L_{T4} = L_4 + 2$. If all packet sizes satisfy Constraint 8, the algorithm marks this SS as a valid SS and stores the resulting packet sizes to be used in determining the optimal parity assignment. After traversing the tree, the values of $\mathcal{E}[LSAD_{GOF}]$ for all valid SS values are calculated using the definition of Equation (8). As a result, the optimal packet sizes of a GOF which determine the optimal parity assignment of each packet form the SS with a minimum value of $\mathcal{E}[LSAD_{GOF}]$.

Considering the real-time nature of voice transmission, we form the OST off-line for different parity budgets B_C and different values of S. We store the resulting search sets into lookup tables. Further and to further reduce search and processing time, the quantities of Ψ can be calculated off-line and stored into look up tables for different values of L and channel conditions since the calculation of $\Psi(m)$ from Equation (5) does not depend on the input signal X.

Algorithm 1 Pseudocode: BuildOST(B_C, S, $Cmin$)

```
s ← 1{a counter variable for the number of tree levels}
while s ≤ S do
    N_s^total ← 0 {the total number of children on level s}
    CreateSubTrees( s, S, N_s^total, B_C, C_min)
    s ← s + 1 {increment s to move to the next level}
end while
```

Algorithm 2 Pseudocode: CreateSubTrees(s, S, N_s^{total}, B_C, C_{min})

if $s = 1$ **then**
  Create Node $Root$ {build the node data structure and return a pointer to "Root". This node is labeled RN on Fig. 3}
  {assign the values of a node's parameters}
  $Key[Root] \leftarrow 0$ {the parity of this node is equal to zero since the root node does not refer to any packet}
  $Parent[Root] \leftarrow NIL$ {the root node does not have a parent}
  $B_{cr}[Root] \leftarrow B_C - Key[Root]$ {the remaining parity is B_C}
  $PP[Root] \leftarrow B_{cr}[Root] - C_{min}[S - s]$ {calculate the parity assignment of the pilot node of the subtree $ST(RN)$}
  $Nc[Root] \leftarrow \frac{PP[Root]}{C_{min}}$ {the number of children nodes of the root node RN}
  $N_s^{total} \leftarrow$ CreateChildrenNodes($Root$, s, S, N_s^{total}, C_{min}){create the children nodes of the root node and return the number of children nodes on the first level $s = 1$}
else
  for each node $N_q^{(s-1)}$ in $LevelNodeList[s-1][q]$, $q \in \{1, 2, \cdots, N_{(s-1)}^{total}\}$ **do**
    $N_s^{total} \leftarrow$ CreateChildrenNodes($N_q^{(s-1)}$, s, S, N_s^{total},C_{min})
  end for
end if

Algorithm 3 Pseudocode: CreateChildrenNodes(Parent, s, S, N_s^{total},C_{min})

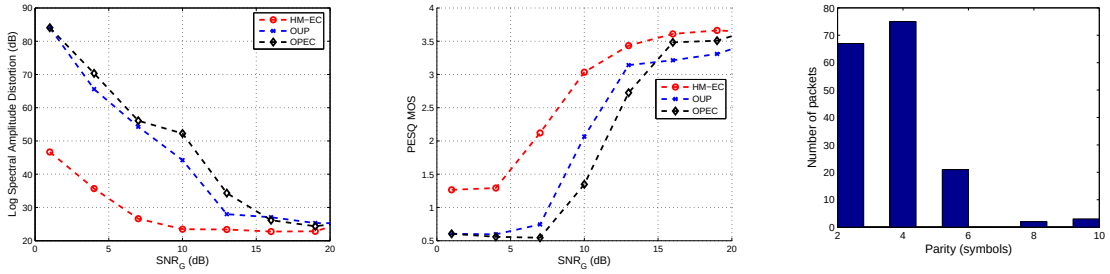
if $s \neq S$ **then**
  for $v = 1$ to $Nc[Parent]$ **do**
    $j \leftarrow v + N_s^{total}$ {this variable is used to name the node $N_j^{(s)}$ as shown on Fig. 3 where j is the order of this node in level s}
    Create Node $N_j^{(s)}$ {build a node's data structure, $N_j^{(s)}$ is the node's pointer}
    $LevelNodeList[s][j] \leftarrow N_j^{(s)}$ {save the location of each node of each level in $LevelNodeList$ linked list}
    $Key[N_j^{(s)}] \leftarrow PP[Parent] - (v-1)C_{min}$ {the parity assignment of each node}
    $Parent[N_j^{(s)}] \leftarrow [Parent]$ {each node has a pointer that points to its parent node}
    $B_{cr}[N_j^{(s)}] \leftarrow B_{cr}[Parent] - Key[N_j^{(s)}]$ {save the remaining parity budget}
    $PP[N_j^{(s)}] \leftarrow B_{cr}[N_j^{(s)}] - C_{min}[S-(s+1)]$ {find the parity assignment of the pilot node (the first child of node $N_j^{(s)}$ of the node's subtree $ST(N_j^{(s)})$}
    if $s = S - 1$ **then**
      $Nc[N_j^{(s)}] \leftarrow 1$
    else
      $Nc[N_j^{(s)}] \leftarrow v$ {the number of children that this node has}
    end if
  end for
  $N_s^{total} \leftarrow N_s^{total} + Nc[Parent]$
else
  for $v = 1$ to $Nc[Parent]$ **do**
    $j \leftarrow v + N_s^{total}$
    Create Node $N_j^{(s)}$
    $LevelNodeList[s][j] \leftarrow N_j^{(s)}$
    $Key[N_j^{(s)}] \leftarrow B_{cr}[Parent]$ {the parity of the leaf node is equal to the remaining parity}
    $Parent[N_j^{(s)}] \leftarrow [Parent]$
    $B_{cr}[N_j^{(s)}] \leftarrow 0$ {the remaining parity is equal to zero since the leaf nodes have no children}
    $PP[N_j^{(s)}] \leftarrow 0$ {the leaf node does not have children or subtrees and it does not have a pilot node either}
    $Nc[N_j^{(s)}] \leftarrow 0$ {the leaf node has no children}
  end for
  $N_s^{total} \leftarrow N_s^{total} + Nc[Parent]$
end if

V. Simulation Results

In this section, we discuss the performance evaluation results of transmitting speech signals over a channel prone to bit errors and packet erasures. In our experiments, we perform a lookup on a table that is formed off-line to identify the optimal choice of parity assignments. We perform experiments with several speech signals with different acoustic characteristics. These signals belong to people with different ages, genders, and languages. More specifically, we use the ITU P.862 conformance speech test files [3]. In our experiments, we use a compressed header size of 5 bytes which significantly reduces the transmission overhead. We simulate bit errors and packet erasures using two independent GE and G chains, respectively. For the GE model, we set $\gamma = 0.99875$ and $\beta = 0.875$ which corresponds to average burst lengths of $1/(1-\gamma) = 800$ and $1/(1-\alpha) = 8$ bits for state G and B, respectively. For the G model, the value of γ is set to 0.99875 and β is calculated as $\beta = 2 - \gamma - ((1-\gamma)/P_{sym})$. Further and in order to differentiate between the

Algorithm 4 Pseudocode: TraverseTree(LevelNodeList, S, L)

{L is an array that has the GOF packet sizes without parity, i.e., $L[1] = R_1 + H$ where R_1 is the size of the source bits of packet 1 and H is the header size}
$i \leftarrow 1$ { a counter for the number of nodes at each level}
$ss \leftarrow 0$ {a counter for valid search sets}
{start the bottom-up search process from the leaf nodes to the top nodes}
while $LevelNodeList[S][i] \neq NIL$ **do**
  $Np \leftarrow LevelNodeList[S][i]$ {store the pointer of the leaf node in variable Np}
  $ss \leftarrow ss + 1$
  for $j = S$ to 1 **do**
    $L_T \leftarrow L[j] + Key[Np]$ {L_T is equal to the total packet size with parity, i.e., $L_T = L[j] + B_j$}
    if $L_T[j] > 2^{sm} - 1$ **then**
      $ss \leftarrow ss - 1$ {ignore this search set since it has at least one node that does not satisfy Constraint 2 of the optimization problem}
      EXIT {exit the for loop, move to the second leaf node, and test the next SS}
    end if
    $SS[ss][j] \leftarrow L_T$ {$SS[ss][j]$ stores the packet sizes of valid search sets}
    $Np \leftarrow Parent[Np]$ {move to the next node (the parent of the current node) in the search set}
  end for
  $i \leftarrow i + 1$ {move to the next leaf node}
end while



(a) (b) (c)

Fig. 4. The illustrations of (a) LSAD as a function of SNR_G, (b) PESQ-MOS as a function of SNR_G, and (c) the histogram of optimal parity assignments for $SNR_G = 7$ dB. The dg145 speech clip with $B_T = 8.056KB$ over a 2×2 MIMO link with a 10% average packet erasure rate is used.

qualities of the transmission link in state G and B, we set $SNR_G = 10SNR_B$. For the RS code, we choose a symbol size s_m of 8 bits and a GOF size of 3 frames. We run our experiments using the Speex codec version 1.0.5 [4] used in the ABR mode. We set the average bit rate for the high and low quality voice frames to be $r_{hi} = 15$ Kbps and $r_{low} = 3$ Kbps, respectively. To identify the value of k in Equation (4), we conducted several experiments to calculate the Rate-Distortion (R-D) curves of several audio sequences of different acoustic characteristics. We found that a value of $k = 5.25e(-6)$ yields an R-D curve that constitutes a tight upper bound on the R-D curves of a variety of voice sequences. Further, the value of r_{max} in Equation (4) is equal to 24.6Kbps which is the maximum encoding rate supported by Speex. Our performance evaluation experiments generate two main sets of curves. The first set shows the value of $LSAD$ for the entire received voice frames as a function of SNR_G, both measured in dB. The second set measures the voice quality using the ITU-T P.862 metric known as the Perceptual Evaluation of Speech Quality - Mean Opinion Score (PESQ-MOS) metric generating a score that ranges from 0 to indicate the worst quality, to 4.5 to indicate the best quality. Every point on each curve is the average value of 100 experiments. We run our experiments using different budgets, signal-to-noise ratios, packet loss ratios, different MIMO configurations, and without loss of generality BPSK modulation scheme. Referencing Equation (1) and the work of [16], we note that our optimization approach can capture a variety of MIMO configurations. That said, we only report our results for a double-transmit double-receive link since the trend of the results are the same for other antenna configurations. We compare our scheme with two other schemes to which we refer as Optimal Unequal Protection (OUP), and Optimal Piggy-backing Error Correction (OPEC). The OUP scheme is a media-independent error correction scheme that utilizes the adaptive unequal packet level RS coding policy proposed by

[5]. In this scheme, the sender calculates the expected distortion of all VoIP frames of the GOF and protects the most important packets with the highest expected distortion. OPEC is a media dependent error correction scheme that utilizes the adaptive piggy-backing error protection policy widely used in different VoIP applications such as Robust Audio Tool (RAT) [1]. In this scheme, the sender calculates the expected distortion of all GOF frames and protects the most important frames by sending a copy of those frames piggy-backed onto the subsequent packet. Notice that in both schemes a fixed encoding rate of 15 Kbps is used, in turn treating all voice frames equally and not considering the perceptual importance of each frame. The illustrations of Fig. 4(a) and 4(b) show how our proposed HM-EC outperforms other schemes under both LSAD and PESQ-MOS metrics. In fact, our experiments show similar performance trends for a variety of voice frames with different characteristics, under different channel conditions as well as packet erasure rates, using different budgets, and different MIMO configurations. Finally, Fig. 4(c) shows an example of unequal assignment of parity symbols to different packets when solving the optimization problem.

VI. Conclusion

Jointly utilizing media dependent and media independent protection schemes, this paper presented a new hybrid live voice transmission scheme. We formulated a constrained optimization problem to optimally protect voice packets against bit errors and packet erasures introduced by wireless channels. We also proposed an efficient solution to the optimization problem by constructing an optimal search tree significantly reducing the search time. We explained how a lookup table of optimal parity assignments as functions of transmission parameters could be formed off-line before attempting at the transmission of the live content, thereby, allowing for merely performing a simple table lookup at the time of transmission. Our experimental results showed that our proposed scheme outperforms other alternatives while offering a relatively low complexity.

References

[1] Robust Audio Tool (RAT) Website, http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/index.html.

[2] The FreePhone webpage, http://www-sop.inria.fr/rodeo/fphone/.

[3] ITU p.862 Perceptaul Equaluation of Speech Quality (PESQ) conformance tests files, P series.

[4] The Speex codec website http://www.speex.org.

[5] M. Chen and M. Murthi. "Optimized Unequal Error Protection for Voice over IP". *In Proc. IEEE ICASP*, 2004.

[6] I. Cohen. "Relaxed Statistical Model for Speech Enhancement and a Priori SNR Estimation". *IEEE Trans. Speech and Audio Processing*, Sep. 2005.

[7] Z. Han, A. Kwasinski, and K. Liu. "A Near-Optimal Multiuser Joint Speech Source-Channel Resource-Allocation Scheme Over Downlink CDMA Networks". *IEEE Trans. Communications*, Sep. 2006.

[8] C. Hoene, I. Carreras, and A. Wolisz. "Voice Over IP: Improving the Quality Over Wireless LAN by Adopting a Booster Mechanism - An Experimental Approach". *In Proc. SPIE*, 2001.

[9] A. Khalifeh and H. Yousefi'zadeh. "An Optimal UEP Scheme of Audio Transmission over MIMO Wireless Links". *In Proc. IEEE WCNC*, 2008.

[10] A. Khalifeh and H. Yousefi'zadeh. "Optimal Audio Transmission over Wireless Tandem Channels". *In Proc. IEEE DCC*, 2008.

[11] J. Matta, C. Pepin, K. Lashkari, and R. Jain. "A Source and Channel Rate Adaptation Algorithm for AMR in VoIP Using the Emodel". *In Proc. ACM NOSSDAV*, 2003.

[12] C. Perkins, O. Hodson, and V. Hardman. "A Survey of Packet Loss Recovery Techniques for Streaming Audio". *IEEE Network*, Vol. 12, No. 5, pp. 40-48, Sep 1998.

[13] L.R. Rabiner and M.R. Sambur. "An Algorithm for Determining the End Points of Isolated Utterances". *Bell Syst. Tech. Journal*, Feb. 1975.

[14] G. Rubino and M. Varela. "Evaluating the Utility of Mediadependent FEC in VoIP Flows". *In Proc. Workshop on Quality of future Internet Services (QofIS)*, 2004.

[15] F. Sabrina and J. Valin. "Priority Based Dynamic Rate Control for VoIP". *In Proc. IEEE GLOBECOM*, 2009.

[16] H. Yousefi'zadeh, H. Jafarkhani, and F. Etemadi. "Transmission of Progressive Images Over Noisy Channels: An End-to-End Statistical Optimization Framework". *IEEE JSTSP*, April 2008.