A Source and Channel Rate Adaptation Algorithm for AMR in VoIP Using the Emodel

Johnny Matta Christine Pépin Khosrow Lashkari Ravi Jain
DoCoMo Communications Laboratories, Inc.
181 Metro Drive, Suite 300
San Jose, CA 95110
{matta, pepin, lashkari, jain}@docomolabs-usa.com[i]

ABSTRACT
We present a dynamic joint source and channel coding adaptation algorithm for the AMR speech codec based on the ITU-T Emodel. This model takes both delay and packet loss into consideration. We address the problem of finding the optimal choice of source and channel bit rates given QoS information about the wired and wireless IP network and subject to constraints on maximum packet loss, maximum delay and maximum allowed transmission rate. Our results show that an adaptation is necessary to preserve acceptable levels of quality while making optimal use of the allowed bandwidth. Our technique requires a small number of computations that allows real time operation in parallel to voice streams.

Categories and Subject Descriptors
C.2.1 [**Computer - Communication Networks**]: Network Architecture and Design – *network communications, wireless communication*; C.4 [**Performance of Systems**]: *design studies, performance attributes, measurement techniques*; H.4.3 [**Information Systems Applications**]: Communications Applications – *computer conferencing, teleconferencing, and videoconferencing*.

General Terms
Algorithms, Management, Performance, Design, Reliability.

Keywords
AMR, Forward Error Correction, Emodel, QoS, VoIP, MOS.

1. INTRODUCTION
Existing speech coders were not designed for use over IP packet switched networks. A packet switched network is a shared medium designed for asynchronous transmission on a best effort basis. In IP networks, congestion, delay and packet loss vary over time. However, time-critical applications such as voice and video have traditionally assumed guaranteed bandwidth, delay and synchronous transmission. In addition, most of the speech coders operate under preset schemes for data and channel code rates making them vulnerable to the varying conditions on wired and wireless IP-based hops. Some kind of adaptation is therefore needed to dynamically adapt the codec bit rate to changing network conditions so as to preserve acceptable levels of reliability and quality.

In this paper, we present an analysis of the best tradeoff between source and channel bit rates given constraints on maximum acceptable packet loss, maximum end-to-end delay and maximum transmission rate (including source data, channel data codec and necessary network overhead). The approach uses the AMR speech codec along with Reed-Solomon codes to maximize a measure of subjective quality, namely the *R*-factor given by the Emodel [2]. It takes into account the above QoS constraints on voice transmission (as provided by a QoS manager entity) as well as actual network QoS performance. The rest of the paper is organized as follows. In the next section, previous work related to rate adaptation is described. An overview of the AMR, Reed-Solomon codes and Emodel is given in Section 3. Section 4 presents the analytical results and Section 5 presents the adaptation algorithm. Finally, Section 6 concludes the paper.

NOSSDAV'03, June 1-3, 2003, Monterey, California, USA.

2. RELATED WORK
Several types of degradations occur in IP networks, among them: 1) packet loss due to network congestion, 2) packet loss due to network jitter (e.g. a real-time packet does reach the destination but is dropped by the receiver due to late arrival), 3) delay due to packetization and transmission, 4) delay due to congestion, and 5) packet loss due to random or bursty communication noise. The first four degradations are prominent in the wired IP networks while the last one occurs primarily in wireless networks due to residual bit errors at the link layer. The retransmission mechanism used in the TCP protocol for error control cannot be used due to its inherent delay that might be unacceptable for real-time, interactive voice applications.

Codec rate adaptation is an effective method to mitigate the effects of packet loss due to network congestion. The ETSI/3GPP adaptive multi-rate (AMR) speech codec is suitable for this purpose [8]. In [7], the authors reduce the packet loss by dividing the network conditions into eight states and assigning each state to one of the eight bit rates of the AMR codec. Network conditions are monitored using the difference in timestamps between successive speech frames at the receiving side (i.e., the authors monitor the network jitter). Their results show a drastic reduction in packet loss rate when compared to fixed-rate codecs such as G.711 and G.723.1.

In some situations, reducing the source bit rate alone does not help. Such situations may be short-term transient congestion, congestion caused by others' traffic or residual bit errors caused by a noisy wireless link. Channel coding – or forward error correction (FEC) – can then be used in conjunction with

congestion control to optimally allocate the amount of redundant bits and information bits in response to varying channel conditions.

In [18], Kaindl and Görtz propose a flexible scheme for voice transmission over the mobile Internet in which the AMR codec is combined with a systematic convolutional code of rate *1/n*. The packetization scheme is done according to the optimal puncturing patterns of the code, that is, all bits stemming from the same generator polynomial are put into the same packet. Hence, *n-1* FEC packets follow one media packet. Packet loss is seen as puncturing of the convolutional code for which decoding methods are known. The code reduces both random packet loss in the wired IP network and random bit errors on the wireless link (similar to a physical layer FEC code). Although the authors mention that appropriate feedback can adaptively control the amount of source and channel bits, they do not implement any rate adaptation scheme.

In [9], Bolot *et al*. refine their approach and state the problem as a constrained optimization one: given the maximum allowed transmission bandwidth, what is the combination of main and redundant information which provides the best perceived audio quality? They use redundant audio coding [10] but try to optimize a hypothetical measure of quality taking into account the rate constraint. Congestion control and loss rate are obtained through a TCP-friendly module and RTCP reports.

The need to distinguish network congestion from bit errors on radio links serves as a basis for a new QoS control architecture proposed by Yoshimura *et al*. [21] for mobile multimedia streaming. In this paper, they introduce a new type of proxy called "RTP monitoring agent" located at the edge of the wired network and wireless link. The RTP monitoring agent sends feedback reports about the wired network conditions such as jitter, loss, etc. to the media server. The latter also receives RTCP reports from the media receiver containing statistics about both the wired and the wireless networks. The media server is then able to apply the appropriate strategy depending on whether packet losses are due to network congestion (reduce the encoder bit rate) or radio link errors (increase robustness by adding more FEC). The adaptation algorithm consists of pre-defined combinations targeted to make the total packet loss rate less than 1%.

Our approach is different from the above work in the sense that we derive a dynamic adaptation algorithm of source and channel bit rates to maximize a metric of subjective quality, the Emodel. The algorithm does not consist of pre-defined combinations but rather determines the optimal solution given specific constraints on the maximum delay, packet loss, and allowed bandwidth. In that regard, our method is close to the adaptive FEC-based error control scheme developed by Bolot *et al*. in [9]. However, several differences exist. We exploit the multi-rate capability of the AMR codec in conjunction with media-independent FEC. Also, the Emodel provides a comprehensive evaluation of transmission quality that lacks in Bolot's approach (this includes taking delay into consideration).

Recently, there have been some attempts to evaluate the tradeoff between delay and packet loss recovery, i.e. among the delay FEC introduces and the improvement in speech quality FEC brings. These are the works of Boutremans and Le Boudec [23] and of Jiang and Schulzrinne [24]. In [23], Boutremans and Le Boudec develop an adaptive error control scheme for real-time audio over the Internet, which selects the FEC scheme according to its impact on the end-to-end delay. Their paper is an extension of the work reported in [9] with the addition of the end-to-end delay in the utility function chosen to assess the quality of perceived audio. However, this utility function does not include the impact of packet loss; the packet loss rate after FEC is subject to a constraint in the optimization problem instead. The Emodel - which is presented in the next section - does take packet losses into account. It is also the quality metric used in [24] by Jiang and Schulzrinne. In that paper, the authors investigate the quality tradeoff between Reed-Solomon codes and iLBC's robustness [25] under packet loss with similar bandwidth requirements. While their evaluation tests reveal conclusions that are close to the ones presented in this paper, our method is different in at least three respects. First, we derive the end-to-end delay thoroughly (taking FEC into consideration) and plug this value into the Emodel. Second, a single speech codec, the AMR, is used, whereas [24] requires the use of up to three different codecs. And third, we propose an optimal dynamic algorithm - rather than a heuristic approach - to derive the best combination of source and channel bits with respect to packet loss within specific constraints.

3. VOIP SYSTEM OVERVIEW

3.1 AMR Speech Codec

The AMR speech codec has been developed by ETSI and adopted by the 3rd Generation Partnership Project (3GPP). The coding scheme is the algebraic code excited linear prediction (ACELP). Voice activity detection, comfort noise generation, source controlled rate operation and error concealment of lost frames are also provided in the specifications. The multi-rate codec has eight encoding modes corresponding to eight source bit rates ranging from 4.75 kb/s to 12.2 kb/s (see Table 1). The codec is adaptive in the sense that it can switch its bit rate every 20 ms of speech frame depending upon channel and network conditions. At the output of the encoder, bits are ordered according to their subjective importance and further divided into three classes with decreasing perceptual importance: Class A, Class B, and Class C. In our proposal the three classes are subject to the same level of error protection. We assume the real-time transport protocol (RTP) is used over UDP and IP [12].

Table 1. AMR speech codec bit rates and class division

Mode Index	Bit rates (kb/s)	Class			Total # bits
		A	B	C	
1	4.75	42	53	0	95
2	5.15	49	54	0	103
3	5.90	55	63	0	118
4	6.70	58	76	0	134
5	7.40	61	87	0	148
6	7.95	75	84	0	159
7	10.2	65	99	40	204
8	12.2	81	103	60	244

3.2 Reed-Solomon Codes

Reed-Solomon (RS) codes are a special class of non-binary linear block codes called "erasure codes" [11]. The popularity of RS codes relies on two facts: first, they offer very good erasure protection and second, efficient yet simple decoding algorithms make it possible to implement relatively long codes in many practical applications. An *(n, k')* RS code defined over the

Galois Field $GF(2^q)$ is described by the following parameters: the block length n is equal to $2^q - 1$, the number of information symbols encoded into a block of n symbols is $k' = 1, 2, \ldots, n-1$, and the code rate is k'/n. The code is guaranteed to correct up to $n-k'$ erasures. In this paper, we consider $q = 8$ so that eight information bits (one byte) are mapped into one of the 256 symbols. A systematic (n, k') block code consists of k' information symbols followed by $(n-k')$ redundant symbols. A $(255, k')$ RS code can be shortened to a $(255-m, k'-m)$ RS code to accommodate a given application.

Since we are concerned with the loss of entire packets, redundancy is spread across packets as shown in figure 1. Each media packet can be seen as consisting of several symbols, each symbol being an element of $GF(2^8)$. A group of k symbols from k different media packets is used to generate $(n-k)$ redundant symbols, creating a length n codeword. These redundant symbols are then placed in separate packets. In this way, a systematic block code consisting of n packets is generated from k media packets.

This FEC scheme is very similar to the RTP payload format proposed in [20]. The difference is that the former uses RS codes whereas the latter focuses on simple exclusive OR (XOR) parities to generate FEC packets. Both codes (RS and parity codes) are media-independent FEC techniques, as opposed to media-specific FEC such as redundant audio coding. Media-independent FEC's main advantage is the fact that the repair is an exact replacement for a lost packet. Its disadvantages are the increased bandwidth and additional decoding delay in case of packet loss.

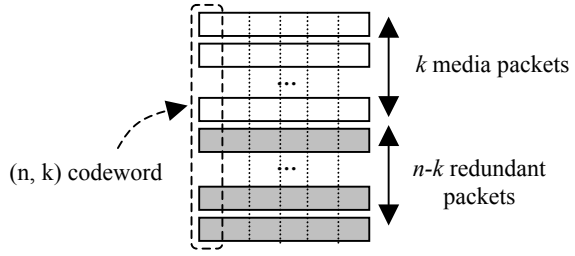

Figure 1. *(n, k)* RS encoding across packets

3.3 Assessing Speech Quality: The Emodel

To assess the tradeoff between source and channel bit rates (R_s and R_c respectively), one must relate these and other QoS factors, such as packet loss and allowed bandwidth, to a criterion that can be used as a basis of comparison. One such criterion is the Emodel, a computational model standardized by the ITU-T [2]. The output of the model is a scalar quantity rating value R, on a scale from 0 to 100, where R=0 represents an extremely bad quality and R=100 represents a very high quality. The difference between *MOS* and R scores is that the former is based on real subjective ratings of a number of voice calls by a set of people, whereas the latter is an analytical estimate of the subjective score.

Objective tests, such as the perceptual speech quality measure (*PSQM*) [15] and the perceptual evaluation of speech quality (*PESQ*) [16], offer fast alternatives to subjective testing. Although correlations between *PESQ* scores and ITU-T database subjective *MOS* results are around 93%, the objective metric does not provide a comprehensive evaluation of transmission quality. It only measures the effects of one-way speech distortion and noise on speech quality. The effects of loudness loss, delay, echo, and other impairments related to two-way interaction are not reflected in the *PESQ* scores. On the other hand, these impairments are taken into consideration in the Emodel. We therefore choose the R-factor as a measure of speech quality.

The R -factor in the Emodel is defined as follows:

$$R = R_0 - I_s - I_d - I_e + A \quad \textbf{(1)}$$

where R_0 incorporates the effect of noise, I_S accounts for quantization, I_d represents the effect of delay, and I_e captures the effect of signal distortion due to low bit rate codecs – the "equipment impairment factor" - as well as packet losses of random distribution. The advantage factor A captures the fact that some users might be willing to accept a reduction in quality in return for service convenience, such as cellular or satellite phones. Basically, measured delay can be mapped into I_d and packet loss into I_e. R_0 and I_S do not depend on network performance and are inherent to the voice signal itself. The R-factor can be translated into *MOS* [2]:

$$\begin{aligned} &\textit{For } R < 0: && MOS = 1 \\ &\textit{For } R > 100: && MOS = 4.5 \\ &\textit{For } 0 < R < 100: && MOS = 1 + 0.035R + 7.10^{-6}R(R-60)(100-R) \end{aligned} \quad \textbf{(2)}$$

The rating of voice quality and the corresponding R and *MOS* scales are shown in Table 2. *MOS* versus R values are plotted in figure 2.

Table 2. *R*-factor and *MOS* scales with corresponding voice rating quality

***R*-factor**	**Quality of Voice Rating**	***MOS***
90 < R < 100	Best	4.34 - 4.50
80 < R < 90	High	4.03 - 4.34
70 < R < 80	Medium	3.60 - 4.03
60 < R < 70	Low	3.10 - 3.60
50 < R < 60	Poor	2.58 - 3.10

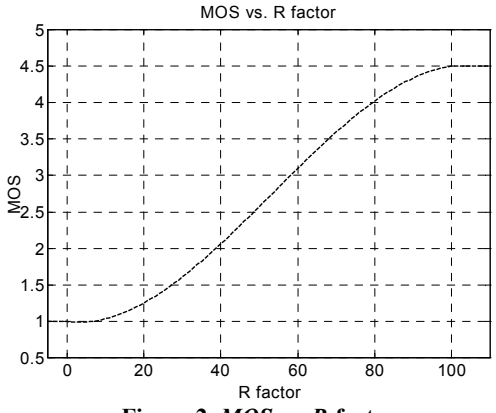

Figure 2. *MOS* vs. *R*-factor

For voice traffic, the R-factor is given by [2]:

$$R = 93.2 - (I_d + I_e) \quad \textbf{(3)}$$

The mapping from end-to-end delay to I_d is available for various types of voice conversations referred to as tasks and defined in [3] (e.g. task 1 and task 6 are a business and casual conversation

respectively) and Echo Loss (*EL*) factors. *EL* factors give a measure of echo loss in dB at points of reflection, so they depend on the echo cancellation scheme [3][4]. The I_d factor versus the end-to-end delay mapping is shown in figure 3.

The mapping from packet loss to I_e is not readily available for the AMR speech codec, except for one work where the AMR codec is tested over the WCDMA physical layer [14]. In this work, the performance is expressed in terms of differential *MOS* versus the speech frame erasure rate (*FER*). The output bits of the AMR are convolutionally encoded using a rate *1/3* code for Class A bits and a rate ½ code for Class B and C bits. In addition, a 12-bit CRC code is applied to Class A bits. *FER* represents the average rate of speech frames for which the CRC check fails in Class A bits. The differential *MOS* is the difference between the *MOS* of the codec for a given mode and given *FER*, and a fixed reference value that is the *MOS* of the codec at 12.2 kb/s under error-free condition. The differential *MOS* thus represents the quality degradation with respect to the best quality achievable with the AMR We derive the I_e factor by first extracting the *MOS* value, for a given mode and given *FER*, from the differential *MOS* result. We then invert equation (2) to obtain the *R*-factor and, assuming that $I_d = 0$, we get $I_e = 93.2 - R$. The I_e factor versus *FER* mapping is shown in figure 4.

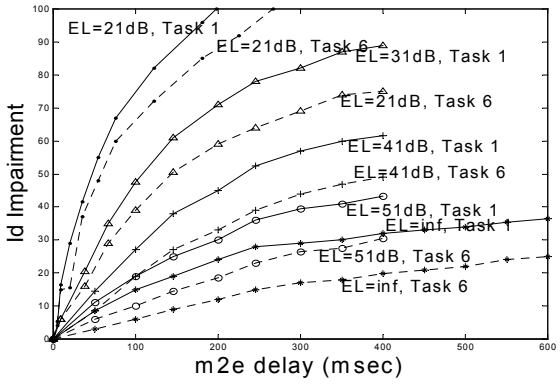

Figure 3. I_d vs. mouth to ear (or end-to-end) delay [4]

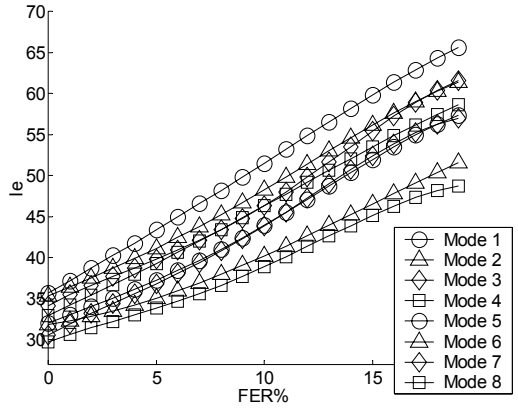

Figure 4. I_e vs. frame error rate (*FER*) for the eight modes of the narrowband AMR [14]

4. DELAY AND PACKET LOSS FOR JOINT AMR-RS CODING

As described above, the Emodel shows that increasing delay leads to reducing the *MOS* value while decreasing packet loss leads to increasing the *MOS*. Therefore, we would like to find the optimum source-channel coding rates resulting in the best *MOS* score. As a next step, we will derive the end-to-end delay as well as the packet loss rate after FEC.

4.1 End-to-End Delay

The end-to-end delay has various components. First, the encoding delay which consists of the sum of the frame size T (20 ms for AMR) and the look-ahead delay l_a (5 ms for mode 8 and zero for the other modes). Second, the packetization delay which accounts for the time it takes to group f frames together into one packet, i.e. $D_{pack} = (f-1)T$. Since we have to generate k speech packets for each FEC block, the sum of the encoding delay and the packetization delay amounts to a formation time $T_f = kTf + l_a$. We assume the processing delay for both the ACELP coder and the RS coder to be negligible. The first k speech packets can be transmitted as soon as they are formed provided copies are kept in a buffer for the computation of the *(n-k)* FEC packets. The transmission of each of the first *(k-1)* packets occurs in parallel to the formation time of the next packet so it does not contribute to the total transmission time T_t. The transmission time for the k^{th} packet does however contribute to T_t. In addition, *(n-k)* FEC packets are formed immediately after all of the k speech packets are generated. They are transmitted in sequence, thus contributing to T_t (note that all n packets are of the same size under the FEC scheme we consider). We assume each packet transmitted on the network incurs a transmission delay T_h, propagation delay P_h and a queuing delay Q_h at each hop h in the path from source to destination. Under simplifying assumptions that are conservative (i.e. they slightly overestimate the value of the end-to-end delay), we derive the total transmission delay as:

$$T_t = (n-k+1)(TfR_s + \Omega) \sum_{hops} 1/B_h$$

where R_s represents the AMR bit rate before channel coding, B_h denotes the bandwidth at hop h, and Ω is the overhead in bits introduced by IP, UDP and RTP headers (typically 20, 8 and 12 bytes, respectively).

The propagation delay is negligible if within a local area; for intra-continental calls, the propagation delay is on the order of 30 ms and for inter-continental calls the delay can be as large as 100 ms [13]. Upon reception, packets are usually delayed in a jitter buffer and the fixed playback delay must be equal to at least one speech frame of data. The end-to-end delay becomes:

$$D = kTf + l_a + (n-k+1)(TfR_s + \Omega)\sum_{hops}\frac{1}{B_h} + \sum_{hops}(Q_h + P_h) + J_{buff} \quad \textbf{(4)}$$

It is to be noted that the B_h and Q_h can be obtained from QoS estimation measurements such as in [1] and [26] (more details follow in section 5). The I_d factor can then be deduced using figure 3.

Equation (4) actually represents the worst-case scenario where the receiver has to wait for the *(n-k)* FEC packets in order to start decoding the RS encoded packets. Indeed, as has been pointed out by a number of authors, media-independent FEC does not introduce any delay unless there is packet loss. However from a practical point of view, the delay introduced by

FEC should be taken into account to ensure smooth playback of the reconstructed speech frames when packet loss occurs. In that respect, joint adaptive FEC and playout buffer algorithms such as the one presented in [6] constitute promising integrated approaches.
Finally, depending on the particular system implementation, the designer can use a more precise estimate for the end-to-end delay to tune the performance further.

4.2 Packet Loss and Frame Loss After FEC

The adaptation algorithm we consider assumes that estimates for the packet loss rate p_s on the end-to-end path is available at the time an adaptation decision is being made.
The packet loss measurement is made after FEC schemes try to recover errors. Assuming a random loss model, one can relate the measured packet loss p_s to the parameters of the FEC scheme, namely n and k, and the "raw" packet loss rate on the end-to-end path p_r, i.e. what the packet loss would be in the absence of any correction scheme. So we can write [6]:

$$p_S = p_r\left(1-\sum_{i=k}^{n-1}\binom{n-1}{i}(1-p_r)^i\, p_r^{n-1-i}\right) \quad \textbf{(5)}$$

Equation (5) shows that given the measurement p_s and an (n, k) RS code, it is possible to deduce the current p_r. This value of p_r is computed once per adaptation period and is used in the adaptation algorithm as shown in the next section. Figure 4 shows I_e versus FER, therefore we need to relate p_s to the frame erasure rate after FEC. In percentage, we have:

$$FER = p_s.100 \quad \textbf{(6)}$$

We can then obtain I_e by referring to figure 4.
The data for I_e in figure 4 assumes random errors. One could argue that the statistical nature of packet loss after FEC is bursty not random; this is because a failure to decode the block leads to a loss of k packets (kf speech frames). A more accurate approach to use data such as in figure 4 for I_e is to have multiple curves for each mode corresponding to different numbers of speech frames per packet as well as different burst lengths. This information is not available to us at the time of writing this paper. We acknowledge the limitation in using the data as is but our aim in section 5 will be to justify and illustrate the need for the adaptation process.

5. ADAPTATION ALGORITHM

The algorithm we describe below addresses the issue of choosing appropriate source and channel bit rates given information on the allowed bandwidth, maximum allowed delay, maximum permitted packet loss and minimum goodput. It also requires knowledge of current network conditions in terms of bandwidth, congestion and packet loss.

5.1 Location of the Adaptation Algorithm

The adaptation algorithm resides at the transmission end of the path. It requires the presence of a QoS estimation module, which is expected to provide real-time estimates for packet loss, bandwidth and congestion (see figure 5). Such a QoS estimation module can be the technique presented in [26]. It assumes that routers can be modified or software downloaded to routers that measures the relevant instantaneous parameters such as queue lengths and sends raw or summarized information back to a QoS estimation module. This method gives accurate estimates at the expense of infrastructure modifications. It may be realistic (or even desirable) if a service provider has control over the core IP network and wishes to differentiate itself from competitors by providing better service. However, it may not always be feasible. An alternative technique [27] sends several pairs of packets from the edge of the network to routers along the path from the source to the destination and uses router timestamps to obtain estimates of packet loss and bottleneck link bandwidth. This method requires no infrastructure modifications but the results may not be as comprehensive or accurate as when routers can be modified. Several other research groups are also pursuing edge-based network measurement techniques, e.g. [29][30].
In addition to the QoS estimates, constraints on packet loss, delay and allowed bandwidth are provided by a QoS manager as shown in figure 5. Limits on delay and packet loss can in most cases be considered static and defined in such a way as to satisfy minimum requirements resulting in acceptable voice quality. Allowed bandwidth reflects the constraint set by the congestion control policy on the maximum allowed transmission speed. In one possible form, congestion control is achieved independently for each flow such as by using DCCP [28]. The congestion control algorithm will converge to the available bandwidth of the path and in this case allowed bandwidth will be equal to available bandwidth (i.e. the maximum transmission speed such that congestion does not build up). In a more elaborate form, congestion control may attempt to optimize overall call quality across several simultaneous voice conversations; allowed bandwidth may then be less than available bandwidth.
The inputs to the source and channel rate adaptation algorithm are then 1) QoS information: Path packet loss p_s, path bandwidth B_h and congestion Q_h, and 2) QoS constraints: Maximum end-to-end delay D_{max}, maximum allowed bandwidth $AlBw$, minimum goodput G_{min}, and maximum permitted packet loss p_{max}. The output will be a decision on the choice of R_s and R_c, the source and channel bit rates respectively.

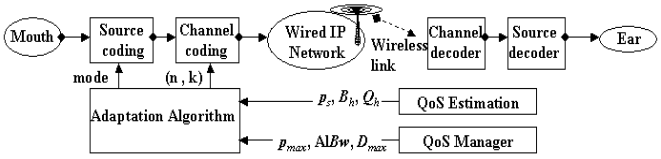

Figure 5. Framework for VoIP and adaptation

5.2 Proposed Algorithm

We start by setting the following constraint:

$$AlBw \geq \frac{n}{k}\left(R_s + \frac{\Omega}{Tf}\right) \quad \textbf{(7)}$$

where the right-hand side of equation (7) represents the total transmission rate (source, channel, and overhead bits included). This constraint ensures that no congestion is generated due to adaptive FEC.
Now R_s can only take on values from among eight possible rates: 4.75, 5.15, 5.90, 6.70, 7.40, 7.95, 10.2 and 12.2 kb/s[1]. The algorithm starts by picking the lowest value of 4.75 kb/s for R_s. We then vary k and n over the ranges k=1,..., k_{max} and n=k,..., n_{max} (n=k is the case without FEC). Parameter k_{max} is such that the delay D given by (4) is bounded by a maximum value D_{max}

[1] Nine different rates are possible with the wideband AMR.

beyond which voice quality becomes unacceptable (around 150-200 ms). Using (7) in (4), we obtain:

$$k_{\max} = \frac{D_{\max} - l_a - \sum_h (Q_h + P_h) - Q_{\max} - (TfR_s + \Omega)\sum_h \frac{1}{B_h}}{Tf + (TfAlBw - TfRs - \Omega)\sum_h \frac{1}{B_h}} \quad (8)$$

Parameter n_{max} is defined as:

$$n_{\max} = \min\left\{ \frac{k.AlBw}{\left(R_s + \Omega/Tf\right)}, n^*, n' \right\} \quad (9)$$

where n^* and n' are imposed by the constraints on packet loss and goodput respectively. Hence n^* is the length of the RS code for which packet loss p_s after FEC goes below p_{max}; and n' is defined by:

$$\frac{k}{n} \geq G_{\min} \Rightarrow n \leq \frac{k}{G_{\min}} \Rightarrow n' = \frac{k}{G_{\min}} \quad (10)$$

where G_{min} is the minimum goodput required by the application. The values for n^* and n' are recomputed for every k.

For each (n, k) pair, we obtain a value for D using (4) and this is mapped into a value for I_d using the data from figure 3.

The p_r value is derived from the measured p_s value using (5) and the current settings for n and k. This step actually requires computing a table off-line to relate these two values. The new selection of n and k combined with the calculated value of p_r allows computation of the new p_s using (5) again. Finally the new p_s is used to obtain FER, which in turn is mapped into a value for I_e using the data from figure 4.

At this stage, we have values for both I_d and I_e corresponding to the chosen pair (n, k). We can then obtain R using (3), and MOS using (2). The process described above is repeated for all R_s values. We finally choose the combination of R_s and R_c that gives the maximum MOS score.

Further refinements of the algorithm take into account the various possible outcomes of the process. In one instance, a subset of (R_s, n, k) combinations may be optimum in that the MOS score corresponding to each combination may be very close to the highest MOS score. By "very close" we mean that the absolute difference between each of these optima and the highest MOS is less than or equal to 0.1. In terms of user perception, such a difference in MOS is not noticeable. In this case, instead of automatically choosing the (R_s, n, k) combination with the highest MOS, we pick the triplet from this subset that leads to the lowest total bandwidth utilization.

In another instance, a solution may not exist. In that case, we do not use channel coding and pick the highest AMR mode that does not violate the allocated bandwidth constraint.

We summarize the steps of our adaptation algorithm below:

1) **A. For all** $R_s \in$ {AMR mode 1, …, AMR mode 8}:
- Find k_{max} using (8)

B. For all k=1, …, k_{max}:
- Find n_{max} given by (9)

C. For all n=k, …, n_{max}:
- Compute D using (4) ⇒ Find I_d (figure 3)
- Compute p_s using (5)
- Compute FER using (6) ⇒ Find I_e (figure 4)
- Find R given I_d and I_e using (3)
- Find MOS using (2)
- Let S={$(R_{s,i}, n_i, k_i)$, i=1,…, u} denote the set of solutions.

2) **If** S≠∅:

A. Set $(R_{s,opt}, n_{opt}, k_{opt}) \in$ S such that
$MOS(R_{s,opt}, n_{opt}, k_{opt}) \geq MOS(R_{si}, n_i, k_i)$, i=1,…, u

B. If ∃ T={$(R_{s,j}, n_j, k_j)$, j=1,…, v} such that
$|MOS(R_{s,j}, n_j, k_j) - MOS(R_{s,opt}, n_{opt}, k_{opt})| < 0.1$:
- Choose $(R_s, n, k) \in$ T such that $\frac{n}{k}\left(R_s + \frac{\Omega}{Tf}\right)$ is minimum.

C. Else:
- Choose $(R_{s,opt}, n_{opt}, k_{opt})$

3) **Else if** S=∅:
- Choose $R_s \in$ M={4.75, 5.15, 5.9, 6.7, 7.4, 7.95, 10.2, 12.2} such that $R_s \leq AlBw$ (no FEC).

The algorithm described above consists of an exhaustive search over a relatively small set of values. The number of modes for the AMR codec is typically 8 (9 for wideband AMR) and n and k are typically less than 20 (25 at most). This means the algorithm requires relatively few computational operations, making it suitable for real-time voice over IP communication.

5.3 Results and Discussion

We make the following assumptions about the IP network setup and voice environment:

- One speech frame per packet, i.e. f = 1.
- The path consists of 15 hops, 13 of which are fast core network links at 622 Mb/s, and two are wireless edge links at 384 kb/s (WCDMA) and 11 Mb/s (IEEE802.11b), respectively.
- P_h = 0.06 ms per hop.
- Q_h is random between 0 and 1 ms.
- J_{buff} = 20 ms. The size of the jitter buffer is in fact quite implementation dependent. We will just set it to be a nominal value of 20 ms.
- D_{max} = 200 ms.
- Ω = 40 bytes (IP/UDP/RTP).
- We assume Task 6 (free conversation) with $EL = \infty$.
- The goodput constraint G_{min} is ignored.
- $p_{max} = 10^{-4}$.

The result in figure 6 shows that it is possible to obtain several solutions that are close in terms of achieved MOS value but have a large difference in terms of required total bandwidth. This situation illustrates step 2 of the algorithm (e.g. points E and F in figure 7 where a 0.1 degradation in MOS corresponds to sparing 8.5 kb/s of bandwidth).

In figure 7, we show the best MOS score for different allowed bandwidth constraints and a fixed raw packet loss rate (p_r=15%). Figure 8 shows the best MOS score for different raw packet loss rates and a fixed allowed bandwidth ($AlBw$=54 kb/s). In Table 3, we indicate n, k, the selected mode, and values for I_d and I_e for each point in figures 7 and 8. From Table 3, we can see that mode 7, k=*1* and n=*2* is the best solution up to p_r=11% (points G-I). This stems from the fact that mode 7 is inherently more resilient to losses than lower modes (mode 8 being the most

robust); moreover, using only one media packet to generate one FEC packet minimizes the delay. For points J and K, corresponding to p_r=13% and p_r=17% respectively, switching to a lower mode reduces the delay impairment in order to compensate for the increase in loss rate. For a very noisy channel (p_r>17%) and the same allowed bandwidth, switching to a lower mode and adding more FEC is the best option (point L, p_r=22%). The delay impairment is increased due to more FEC but this is accepted in order to improve the correction capability of the RS code in very noisy environments.

As for the loss model, burst packet loss over the Internet occurs more frequently than random packet loss. Under a bursty packet (Gilbert) loss model, our *MOS* results would most likely not be as good as under a random loss model (see [17]). However, we can expect the general behavior to be somewhat similar in both models, in which case the optimal points remain the same.

As future work, we intend to verify this assumption by testing the AMR codec over a two-state Gilbert model using the same adaptation algorithm. The case where a packet contains multiple frames should also be taken into account as the number of frames per packet affects the perceived audio quality. This is part of a subsequent paper.

Table 3. Details of figures 7 and 8

Points	*n*	*k*	Mode	I_d	I_e	*AlBw* (kb/s)	p_s (%)
A	5	4	6	7.54	37.61	30	7.17
B	4	3	6	5.97	36.22	35	5.79
C	5	3	6	6.07	33.07	40	1.64
D	3	2	8	4.89	33.56	45	4.16
E	2	1	6	3.08	33.44	50	2.25
F	2	1	8	3.46	32.06	60	2.25
G	2	1	7	3.10	30.97	54	0.01
H	2	1	7	3.10	31.65	54	0.49
I	2	1	7	3.10	32.61	54	1.21
J	2	1	6	3.08	33.10	54	1.69
K	2	1	6	3.08	33.87	54	2.89
L	4	2	6	4.58	33.76	54	2.73

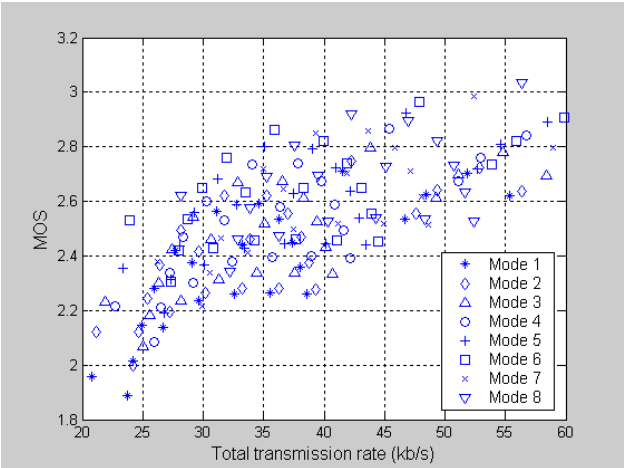

Figure 6. *MOS* for all modes vs. total transmission rate (p_r=10%, *f*=1, *AlBw*=60 kb/s)

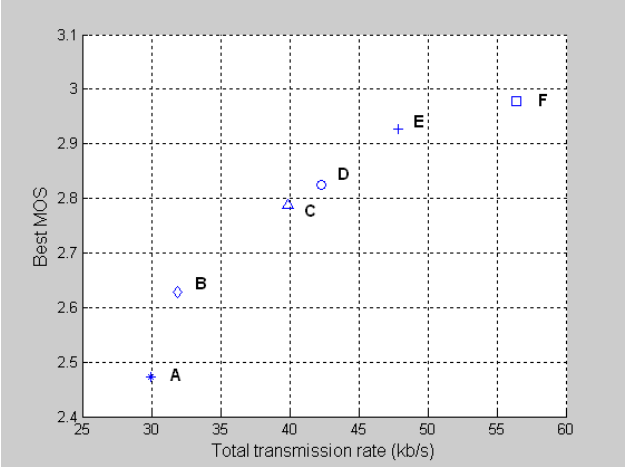

Figure 7. Best possible *MOS* among all modes vs. total transmission rate (p_r=15%, *f*=1)

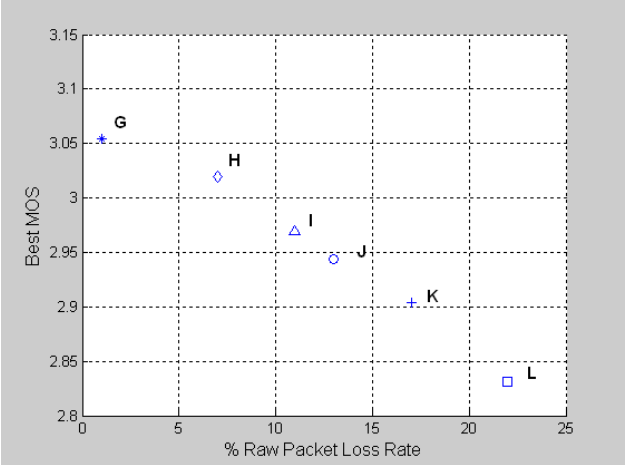

Figure 8. Best possible *MOS* among all modes vs. p_r (*f*=1, *AlBw*=54 kb/s)

6. CONCLUSION

This paper proposes an adaptation algorithm of source and channel coding rates to QoS conditions in wired and wireless IP networks using the AMR speech codec and Reed-Solomon codes. The metric used for optimization is *MOS* as determined by the ITU-T Emodel for voice traffic. We show that for noisy links and given QoS constraints it is necessary to sacrifice source bits for increased robustness to packet loss. Moreover, we acknowledge the fact that while FEC mitigates the effects of packet loss, it also increases the end-to-end delay. These two trends work opposite to each other, the first causing an increase and the second a decrease in voice quality. Our algorithm finds the optimum compromise between packet loss recovery and end-to-end delay to maximize perceived voice quality.

7. ACKNOWLEDGEMENTS

Special acknowledgement goes to Genista Corporation for providing us with raw data and useful information about their experiments on the AMR coder. We are also grateful to Nobuhiko Naka for his valuable comments, as well as Toshiro

Kawahara, Minoru Etoh, and James Kempf for their feedback. Finally we thank the reviewers for their comments.

8. REFERENCES

[1] Jain, M., and Dovrolis, C. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. In *Proceedings of ACM Sigcomm 2002* (Pittsburgh, PA, August 2002).

[2] *ITU-T* Recommendation G.107. The Emodel, a computational model for use in transmission planning. July 2002.

[3] Kitawaki, N., and Itoh, K. Pure delay effects on speech quality in telecommunications. *IEEE Journal on Selected Areas in Communications*, *9*, 4 (May 1991).

[4] Markopoulou, A., Tobagi, F., and Karam, M. Assessment of VoIP quality over Internet Backbones. In *Proceedings of IEEE Infocom 2002* (New York, NY, June 2002).

[5] *ITU-T* Recommendation G.108. Application of the Emodel: A planning guide. September 1999.

[6] Rosenberg, J., Qiu, L., and Schulzrinne, H. Integrating packet FEC into adaptive voice playout buffer algorithms on the Internet. In *Proceedings of IEEE Infocom 2000* (Tel-Aviv, Israel, March 2000).

[7] Seo, J.W., Woo, S.J., and Bae, K.S. A study on the application of an AMR speech codec to VoIP. In *Proceedings of IEEE ICASSP 2001* (Salt Lake City, UT, May 2001).

[8] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Mandatory Speech Codec speech processing functions AMR speech codec; Transcoding functions (3G TS 26.090 version 3.1.0).

[9] Bolot, J.C., Fosse-Parisis, S., and Towsley, D. Adaptive FEC-based error control for Internet telephony. In *Proceedings of IEEE Infocom 1999* (New York, NY, March 1999).

[10] Perkins, C., *et al*. RTP payload for redundant audio data. RFC 2198, IETF, September 1997.

[11] Lin, S., and Costello D., Error Control Coding: Fundamentals and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[12] Sjoberg, J., *et al.* Real-time transport protocol (RTP) payload format and file storage format for the adaptive multi-rate (AMR) and adaptive multi-rate wideband (AMR-WB) audio codecs. RFC 3267, IETF, June 2002.

[13] Karam, M., and Tobagi, F. Analysis of the delay and jitter of voice traffic over the Internet. In *Proceedings of IEEE Infocom 2001* (Anchorage, AL, April 2001).

[14] Genista Corporation. 3G Voice Service Quality, Objective Characterization of WCDMA Voice Quality. 2001.

[15] *ITU-T* P.861. Objective quality measurement of telephone-band (300-3400 Hz) speech codecs. February 1998.

[16] *ITU-T* P.862. Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs. February 2001.

[17] Jiang, W., and Schulzrinne, H. Comparison and optimization of packet loss repair methods on VoIP perceived quality under bursty loss. In *Proceedings of NOSSDAV 2002* (Miami Beach, FL, May 2002).

[18] Kaindl, M., and Görtz, N. AMR voice transmission over mobile Internet. In *Proceedings of IEEE ICASSP 2002* (Orlando, FL, May 2002).

[20] Rosenberg, J., and Schulzrinne, H. An RTP payload format for generic forward error correction. RFC 2733, IETF, December 1999.

[21] Yoshimura, T., Ohya, T., Kawahara, T., and Etoh, M. Rate and robustness control with RTP monitoring agent for mobile multimedia streaming. In *Proceedings of IEEE ICC 2002* (New York, NY, April-May 2002).

[23] Boutremans, C., and Le Boudec, J.Y. Adaptive delay aware error control for Internet telephony. In *Proceedings of the 2nd IP-Telephony Workshop* (New York, NY, April 2001).

[24] Jiang, W., and Schulzrinne, H. Comparisons of FEC and codec robustness on VoIP quality and bandwidth efficiency. In *Proceedings of ICN 2002* (Atlanta, GA, August 2002).

[25] Andersen, S., *et al*. Internet low bit rate codec. Internet Draft (work in progress), IETF. February 2002.

[26] Matta, J., and Takeshita, A.. End-to-end voice over IP quality of service estimation through router queuing delay monitoring. In *Proceedings of IEEE Globecom 2002* (Taipei, Taiwan, November 2002).

[27] Matta, J., and Jain, R. Extended CAT Probe, US Patent Application, filed March 2003.

[28] Kohler, E. *et al*. Datagram congestion control protocol. Internet Draft (work in progress), IETF. March 2003.

[29] *Proceedings of the Second Internet Measurement Workshop (ACM IMW 2002),* (Marseille, France, November 2002).

[30] Cooperative Association for Internet Data Analysis: http://www.caida.org/

[i] The first author's current email address is jmatta@stanfordalumni.org