A New Adaptive FEC Loss Control Algorithm for Voice Over IP Applications

Chinmay Padhye and Kenneth J. Christensen
Computer Science and Engineering
University of South Florida
Tampa, FL 33620
{padhye, christen}@csee.usf.edu

Wilfrido Moreno
Electrical Engineering
University of South Florida
Tampa, FL 33620
moreno@eng.usf.edu

Abstract

Packet loss causes degradation in the quality of Voice Over IP (VOIP) applications. Forward Error Correction (FEC) methods which add redundant information to voice packets can be used to minimize the effects of packet loss. While these methods can reduce the effects of packet loss, they increase the amount of bandwidth used by a voice stream. This paper builds on existing work in adaptive FEC control algorithms to better control the amount of redundancy. The new adaptive FEC "USF algorithm" considers the history of packet losses in the network before changing the amount of redundancy and also does not react to burst losses. The performance of the USF algorithm is studied using a simulation model. The USF algorithm is able to maintain a loss rate one-half to one-third the loss rate maintained by the current algorithm for Internet traces and for network loss rates between 7% and 20% for synthetic traces.

1. Introduction

Considerable research effort has been directed towards transporting real-time voice over IP networks. The motivation for transporting voice over IP networks is the potential cost savings achievable by eliminating or bypassing the circuit-switched telephony infrastructure. Programs such as NeVoT [14], RAT [10], and Free Phone [8] have demonstrated the feasibility of voice transport over the Internet. Real-time voice applications have an upper bound on the end-to-end delay imposed by human factors considerations. For interactive voice applications, the maximum tolerable end-to-end delay is between 250 to 500 milliseconds [9].

In a Voice Over IP (VOIP) application, voice is digitized and packetized at the sender at regular intervals (e.g., every 20 milliseconds) using an encoding algorithm. The voice packet is then sent over the IP network to the receiver where it is decoded and played-out to the listener. VOIP applications use UDP as the transport layer protocol. The Real-time Transport Protocol (RTP) [15] is used to provide additional functionality including sequence numbers and time stamps. The Real Time Control Protocol (RTCP) is used to return receiver statistics (e.g., the number of packets detected as lost) to the sender. RTCP packets are sent every 5 seconds by a receiver to a sender and consume very little bandwidth. IP networks are inherently best effort networks with variable packet delays and loss. While voice traffic can tolerate some amount of packet loss, a packet loss rate greater than 5% is considered harmful to the voice quality [11]. The amount of packet loss rate that can be tolerated depends on the nature of the encoding algorithm and on the sampling rate of the voice stream. The length of a phoneme is typically between 80 to 100ms [17], and a loss greater than the length of a phoneme can change the meaning of a word.

Changing the IP infrastructure to support guaranteed bandwidth sessions would allow for effective transport of voice streams. Changing the Internet infrastructure is a difficult proposition, hence the interest in application-level techniques for compensating for packet delay jitter and loss. Algorithms using Forward Error Correction (FEC) methods have been developed to compensate for packet loss. However, existing methods can be shown to have shortcomings in their response to burst losses of packets and may exhibit unstable behaviors. In this paper, the adaptive FEC control algorithm developed in [5] (the "Bolot algorithm") is studied and significant improvements are suggested and evaluated.

The remainder of this paper is organized as follows. Section 2 reviews application-level, packet-loss compensation techniques. A redundancy control algorithm proposed by Bolot et al. [5] is also presented. Section 3 describes some possible shortcomings of the Bolot algorithm. Section 4 describes the new proposed algorithm, called the USF algorithm. The USF algorithm is evaluated using simulation in section 5. Section 6 presents the conclusions and future work.

0-7803-5979-8/00 $10.00 © 2000 IEEE

2. Application Level Loss Compensation

Packet loss compensation techniques are divided into receiver-based packet error correction techniques and sender-based error concealment techniques [13]. Error concealment techniques reconstruct the lost packet at the receiver and do not need any additional information from the sender. These techniques exploit the temporal relationship in a voice signal and are most effective for low packet loss rates around 2% [13]. In loss recovery techniques, the sender sends additional information and the receiver then uses this information to recover from packet losses. These techniques are further divided into retransmission-based techniques and Forward Error Correction (FEC) techniques. Retransmission-based techniques are only suited for short and very high-speed links where the round-trip delay is small enough to support a resend of lost packets as in the S-ARQ scheme in [7]. Retransmission is avoided in FEC methods.

FEC-based methods send redundant information along with the original information (e.g., [2]). Priority-based encoding methods can also be used [1]. In media independent FEC based methods, the redundant information can be sent in the form of parity packets [13]. The parity packets are generated from the original packets using a mathematical function and thus the redundant information is independent of the encoding algorithm used. In media specific FEC methods, the redundant information is generated using a different, and lower bit rate, encoding algorithm than the algorithm used to encode the original packet. Media specific schemes piggyback information about the present period with later packets [1, 2, 4, 7 and 10]. This is shown in Figure 1 as the secondary encoding in a packet (with reconstruction occurring at the receiver). Although this technique uses lower bit rate codecs, studies show that the quality of the reconstructed stream is quite good and this scheme increases intelligibility [9]. If a packet N carries a redundant encoding of packet $N - i$ and if packet $N - i$ is lost, the application waits for packet N to recover from the loss. Thus, a single lost packet can be recovered with i packets worth of delay.

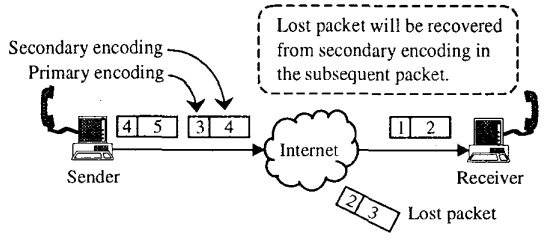

Figure 1 - An example of a media specific FEC scheme

By increasing the amount of redundancy added to the voice stream, the media specific schemes can recover from multiple losses. However, increasing the amount of redundancy when the network loss rate is low will waste bandwidth. This motivates the need to develop methods to control the amount of redundancy depending on the network loss rate. A recently proposed redundancy control algorithm is presented in Bolot et al. [5]. The "Bolot algorithm" tries to maintain the loss rate after reconstruction at the receiver between pre-specified LOW and HIGH loss rate limits. The control algorithm will add redundancy if the network loss rate is above the HIGH mark and decrease the amount of redundancy if the network loss rate is below the LOW mark. The amount of redundancy added to a packet stream in the Bolot algorithm depends on the combination number. The combination number and its associated redundancy are shown in Table 1 (taken directly from [5]). For example, an entry of (0, 1), associated with the combination number 1, means that the sender sends a redundant copy of packet $N - 1$ with packet N. The reward associated with the combination number is used to calculate the percentage of packets lost after reconstruction. The reward value is the ratio of the percentage of packets lost before reconstruction to the percentage of packets lost after reconstruction for the combination number and characterizes the quality of the audio at the receiver. The reward value for each combination was empirically determined after running 10 experiments between INRIA, France and UCL, United Kingdom as described in [5].

Table 1 - Combinations for the Bolot algorithm

Combination	Packets sent	Reward
0	0	1
1	0,1	2.5
2	0,2	6
3	0,1,2	6
4	0,1,3	10
5	0,1,2	6
6	0,1,3	10
7	0,1,2,3	18
8	0,1,2,3	18
9	0,1,2,3,4	18
10	0,1,2,4	18
11	0,1,3	10
12	0,1	2.5
13	0,2	6

Figure 2 shows the sender pseudocode for the Bolot algorithm. A received RTCP packet contains the number of packets lost before reconstruction at the receiver in the previous 5 seconds. The combination number (initialized

to 0 at the start of a voice session) is the index in Table 1 from which the reward value (and packet transmission combinations) are obtained. The sender first calculates the percentage of packets lost before reconstruction at the receiver (line 1). The percentage of packets lost after reconstruction is then computed in line 2 using the indexed reward value. The calculated value of the loss rate after reconstruction is compared with the preset tolerance limits in lines 3 and 4 and the value of combination is incremented or decremented accordingly (obviously, the combination number is not allowed to exceed the maximum value in Table 1 or be less than zero - this check is not shown in the pseudocode). The sender then sends packets with the redundancy combination associated with the current combination number.

```
For each RTCP packet received do
   1. Calculate the percentage of
      packets lost before
      reconstruction, Pb
      Pb = Number of packets lost before
      reconstruction / Number of
      packets expected
   2. Calculate the percentage of
      packets lost after
      reconstruction, Pa
      Pa = Pb / Reward associated with
      current combination number
   3. If (Pa > HIGH) then
         Increment combination
   4. If (Pb < LOW) then
         Decrement combination
```

Figure 2 - Sender pseudocode for the Bolot algorithm

3. Shortcomings of the Bolot Algorithm

The Bolot algorithm calculates the percentage of packets lost after reconstruction based on an empirically determined reward value. This value may not be representative under all network conditions and may lead to an incorrect calculation of the percentage of packets lost after reconstruction. In addition, experimental results show many instances of multiple packet losses of length 10 or greater [3 and 4]. Such instances of multiple packet losses are referred to as loss bursts in this paper. Reasonable FEC methods cannot recover from such loss bursts. Hence, increasing the redundancy as a result of a packet loss burst may be a waste of bandwidth. The Bolot algorithm does not consider the change in network loss rate before reconstruction in its decision to change the amount of redundancy. This may cause cyclical behavior. For example assume that the network loss rate is 7%. Assume that the HIGH and LOW values are set at 3%. When the algorithm receives a RTCP report it finds that the network loss rate is above the higher tolerated limit and increases the combination number. With the increase in combination number, redundancy is added to the packet stream and the receiver will experience a lower packet loss rate. When the receiver receives its next RTCP report, it calculates the percentage of packets lost at the receiver after reconstruction using the reward value. This calculated percentage is below the LOW mark. At this point the Bolot algorithm will decrease the amount of redundancy. If the network loss rate is near constant then the receiver will again experience losses above the HIGH value. The Bolot algorithm will increase the redundancy again. Such a cyclical behavior (which can occur even when LOW and HIGH have different values) will result in poor performance.

4. The New Algorithm - The USF Algorithm

The new control algorithm, the USF algorithm, builds on the Bolot algorithm. Like the Bolot algorithm, the USF algorithm uses RTCP reports to get feedback information from the receivers. The USF algorithm requires that the receiver send two additional values as an application specific extension to the RTCP packets. The first value required is the number of packets lost after reconstruction. This value will be used by the USF algorithm to calculate the percentage of packets lost after reconstruction. The second additional value is the total number of packets lost in loss bursts. This value will be used to account for packet loss bursts in the network.

The amount of redundancy added to the packet stream by the USF algorithm depends on the current value of the combination number. Table 2 shows the combination numbers and the associated redundancy used by the USF algorithm. The combination of packets is designed based on results of experiments in [12]. The USF algorithm first increases the delay and then adds more redundant packets as the combination numbers increases. The tradeoff in the design of the combination numbers is between the end-to-end delay and number of redundant packets sent with each packet.

Table 2 - Combinations for the USF algorithm

Combination	Packets sent
0	0
1	0, 1
2	0, 2
3	0, 1, 2
4	0, 1, 3
5	0, 1, 2, 3
6	0, 1, 2, 4
7	0, 1, 3, 4
8	0, 1, 2, 3, 4

The sender pseudocode for the USF algorithm is shown in Figure 3. A received RTCP packet contains the number of packets lost before reconstruction, the number of packets lost after reconstruction, and the number of packets lost in loss bursts. The value of combination (initialized to 0 at the start of a voice session) is the index in Table 2. The sender calculates the percentage of packets lost before reconstruction and the percentage of packets lost after reconstruction at the receiver (lines 1 and 2). As before, the combination number is not allowed to exceed the maximum value in Table 2 or be less than zero - this check is not shown in the pseudocode.

```
For each RTCP packet received do
   1. Calculate packet loss after
      reconstruction, Pa
      Pa = Number of packets lost after
      reconstruction / Number of
      packets expected
   2. Calculate packet loss before
      reconstruction, Pb
      Pb = Number of packets lost before
      reconstruction / Number of
      packets expected
   3. If (Pa > HIGH) then
         Subtract packets lost in loss
         bursts and recalculate Pa
   4. If (Pa > HIGH) then
         Increment combination
   5. If (Pa < LOW) then
         Loss difference = Pb(previous)
         - Pb
   6. If (Loss difference >
         MINIMUM_THRESHOLD) then
         Decrement combination
   5. Set Pb(previous) = Pb
```

Figure 3 - Sender pseudocode for the USF algorithm

The values of HIGH and LOW correspond to target high and low packet loss rates for reconstructed packets (lines 3 and 4). If the percentage of packets lost after reconstruction is greater than the HIGH mark then the number of packets not lost in bursts is calculated by subtracting the number of packets lost in loss bursts from the number of packets lost after reconstruction. The new percentage value is then checked with the value of HIGH. If this new percentage value is greater than HIGH then the combination number is increased. If the percentage of packets lost after reconstruction is lower than the LOW mark, then the percentage of packets lost in the previous RTCP interval is subtracted from the percentage of packets lost in this interval. If this difference is greater than the MINIMUM_THRESHOLD value, then the combination number is decreased. By comparing the values of the percentage of packet loss before reconstruction of the current period with that of the previous period, the USF algorithm considers the past network loss history before decreasing the amount of redundancy. This use of historical loss data is intended to prevent cyclical behavior.

5. Evaluation of the USF Algorithm

The Bolot and USF algorithms were evaluated using a simulation model.

5.1 Inputs used for the simulation experiments

The inputs to the simulation experiments were actual Internet voice packet traces collected by Yajnik et al. at the University of Massachusetts, Amherst [6] and synthetic traces. The Internet traces used in the experiments were collected by sending packet probes at regular intervals of 20ms, 40ms, 80ms and 160ms along unicast and multicast connections. This paper considers the traces with periodic intervals of 20ms and 40ms because these are the typical sampling times of VOIP applications. The receivers were at Los Angeles and Seattle for the 20ms traces and at Atlanta for the 40ms trace. Each packet has a unique sequence number that the receiver uses to keep track of packet loss [6]. Five Internet traces were used, see Table 3, with packet loss rates ranging from 1.38% to 3.76%.

To generate the synthetic traces, the approach used in [3] was employed. A finite-buffer single-server queue models the bottleneck queue in the network path of a voice session. Arrivals at this queue consist of packets generated by interactive processes such as telnet and packets generated by bulk transfer applications such as FTP. Figure 4 shows the queueing model. The simulation was implemented using the CSIM18 simulation engine [16]. Voice packets arrive at regular intervals of 20ms and the other packets have their arrival times exponentially distributed. The ratio of the number of interactive packets to the number of bulk transfer packets is taken as 4:6. The interactive packets are 32 bytes in length, the bulk transfer packets are 512 bytes, and the audio packets are 320 bytes. The bottleneck bandwidth is fixed at 512 Kbps. The rate of arrival of the interactive and bulk transfer packets at the server was increased to simulate loss rates in a range of 1.7% to 35% for the voice packets.

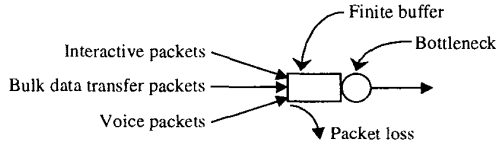

Figure 4 - Queueing model to generate synthetic traces

5.2 Simulation results for the Internet traces

For the simulation experiments, the LOW and HIGH values were set to 3% for both the Bolot and USF algorithms. For the USF algorithm, the MINIMUM_THRESHOLD value was also set to 3%. The value 3% represents a reasonable loss value (i.e., loss rates of greater than 3% typically affect voice quality). The comparison results for the Bolot and USF algorithms with the five Internet traces as input are shown in Table 3 (the first column in Table 3 and 4 is the trace number). The results (in the Ratio column) show that the USF algorithm maintains a loss rate after reconstruction of one-half to one-third that maintained by the Bolot algorithm. Table 4 shows the absolute number of these 5-second periods for which the algorithms maintained a loss percentage greater than the HIGH mark. It can been seen (column 5 of Table 4) that the USF algorithm maintains the desired loss rate over a greater number of periods (i.e., is more than the desired loss rate for fewer periods) than does the Bolot algorithm. This shows why the loss percentage is lower for the USF algorithm.

Table 3 - Simulation results for Internet traces (loss)

#	Network Loss	Loss w/ Bolot	Loss w/ USF	Ratio
1	1.69 %	1.54 %	0.61 %	2.5
2	3.76	2.64	0.91	2.9
3	1.38	1.27	0.66	1.9
4	3.37	2.47	0.86	2.8
5	2.22	1.94	1.07	1.8

Table 4 - Simulation results for Internet traces (periods)

#	Total Periods	Total Above	Above w/ Bolot	Above w/ USF
1	1775	196	167	50
2	1510	922	583	132
3	1773	118	101	40
4	1512	489	356	94
5	4320	865	689	322

Figures 5 and 6 show the number of packets lost after reconstruction for the Bolot and USF algorithms, respectively, for the first Internet trace. The difference in the number of packets lost by the two algorithms is shown in Figure 7. A positive value indicates that the USF algorithm performed better than the Bolot algorithm. It can be seen that the USF algorithm results in far fewer losses than the Bolot algorithm. The graphs for the remaining four traces are similar in appearance to that of the first trace and are not shown.

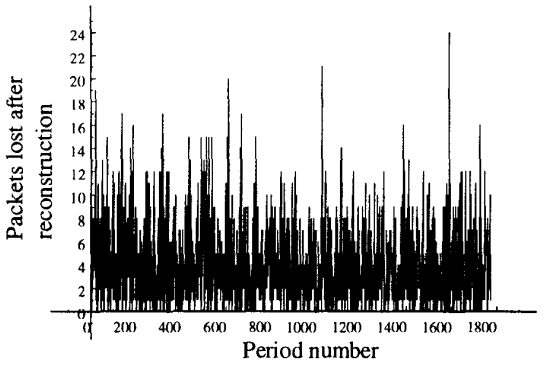

Figure 5 - Number of packets lost by Bolot algorithm

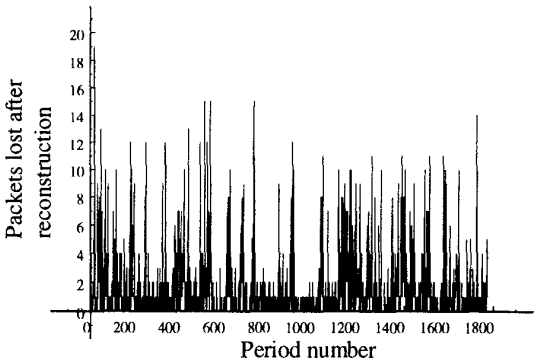

Figure 6 - Number of packets lost by USF algorithm

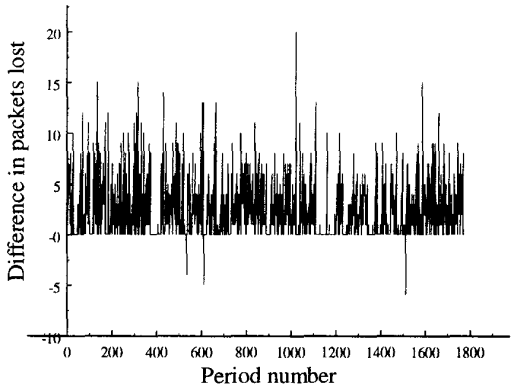

Figure 7 - Difference in the number of packets lost

5.3 Simulation results for synthetic traces

Table 5 shows the result of the simulation experiments for the synthetic trace input. The target loss rate after reconstruction was set to 3%. Figure 8 plots the variation in the loss rates for the two algorithms along

with the network loss rates. The results show that the USF algorithm maintains one-third the loss rate maintained by Bolot algorithm when the network loss rate is below 7%.

Table 5 - Simulation results for the synthetic trace

Network Loss	Loss w/ Bolot	Loss w/ USF	Ratio
1.67 %	1.66 %	0.56 %	2.9
3.67	2.41	0.28	8.6
6.53	2.90	1.02	2.8
9.61	1.67	2.13	0.8
13.11	1.76	3.36	0.5
16.57	2.60	2.18	1.2
20.04	1.80	1.92	0.9
23.42	1.36	2.42	0.5
26.62	1.86	2.96	0.6
29.63	2.39	2.93	0.8
32.44	2.78	2.51	1.1
35.09	3.18	2.80	1.1

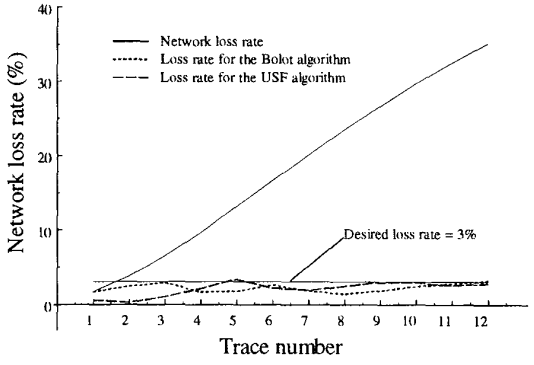

Figure 8 - Variation in loss rate for the synthetic traces

The Bolot algorithm performs better than the USF algorithm for a network loss rate between 9% to 13%. The accuracy of the Bolot reward value for the synthetic traces was studied by plotting the actual percentage loss after reconstruction against the percentage loss calculated by the Bolot algorithm for the trace with a network loss rate of 13%. If the Bolot algorithm were accurate in calculating the packet loss after reconstruction using the reward value then most of the points in the graph would lie along the diagonal. It can be seen in Figure 9 that many points lie above the diagonal for low loss rates. This indicates that the Bolot algorithm can overestimate the loss rate after reconstruction.

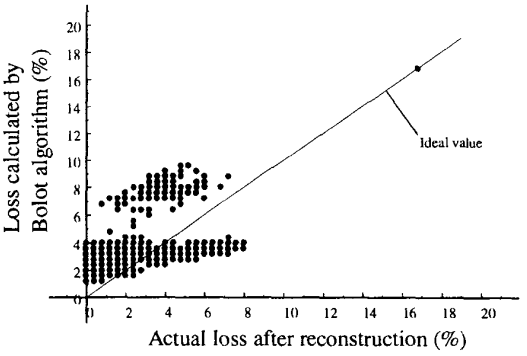

Figure 9 - Error in packet loss by Bolot algorithm

The performance of the Bolot algorithm was simulated using the loss rate after reconstruction directly. Table 8 and Figure 10 show the results. The USF algorithm still performs better than the Bolot algorithm.

Table 8 - Loss after direct reconstruction for Bolot

Network Loss	Loss w/ Bolot	Loss w/ USF	Ratio
1.67 %	1.64 %	0.56 %	2.9
3.67	2.43	0.28	8.6
6.53	3.49	1.02	3.4
9.61	5.18	2.13	2.4
13.11	6.25	3.36	1.8
16.57	5.91	2.18	2.7
20.04	3.72	1.92	1.9
23.42	3.5	2.42	1.4
26.62	4.08	2.96	1.3
29.63	4.32	2.93	1.4
32.44	3.76	2.51	1.5
35.09	3.32	2.80	1.2

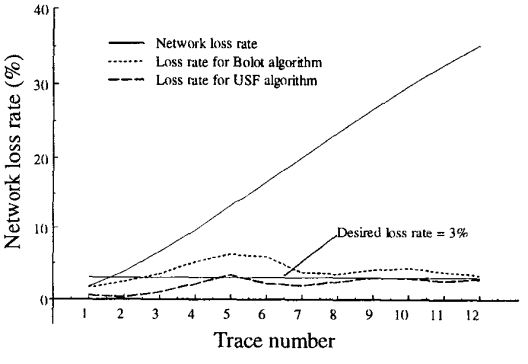

Figure 10 - Loss after direct reconstruction for Bolot

6. Conclusions and Future Work

This paper described a new redundancy control algorithm - the "USF algorithm" - for VOIP applications. A recently presented control algorithm presented in [4] by Bolot et al. is codec specific and assumes that the network loss process is a Bernoulli process. A previous control algorithm also by Bolot et al. [5], called the "Bolot algorithm", uses empirically determined reward values to select the level of redundancy to be used. The USF algorithm dynamically changes the amount of redundancy added to a packet stream at the sender as a function of the actual measured packet loss rate. The USF algorithm uses the number of packets lost after reconstruction as a measure of the effectiveness of its FEC scheme. In addition, the USF algorithm detects loss bursts and considers the network packet loss history before changing the amount of redundancy. The USF algorithm was evaluated using a simulation model with both actual packet voice traces from the Internet and synthetic traces with a loss rate ranging from 1.5% to 35%. For the Internet traces, the loss rate experienced by a voice application using the USF algorithm was one-half to one-third the loss rate experienced using the Bolot algorithm. For the synthetic traces, the USF algorithm maintains a loss rate one-half to one-third of that maintained by the Bolot algorithm for network loss rates of 7% to 20%.

Future work should quantify the bandwidth savings possible from the USF algorithm. The amount of bandwidth savings is highly dependent on the codecs used in the implementation of an algorithm (USF and Bolot). More work also needs to be done on setting MINIMUM_THRESHOLD values for the USF algorithm. Finally, additional experiments should be conducted with a wider range of Internet voice packet trace files.

Acknowledgements

The authors would like to thank Maya Yajnik and Sue B. Moon of the University of Massachusetts, Amherst for making their Internet voice packet traces available. The authors would also like to thank Jean-Chrysotome Bolot for his comments and clarifications.

References

[1] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority Encoding Transmission," *IEEE Transactions on Information Theory*, Vol. 42, No. 6, pp 1737 - 1744, November 1996.

[2] E. Biersack, "A Simulation Study of Forward Error Correction in ATM Networks," *Computer Communications Review*, Vol. 22, No. 1, pp. 36 - 47, January 1992.

[3] J-C. Bolot, "Analysis of Audio Packet Loss in the Internet," *Proceedings of NOSSDAV*, pp. 163 - 174, April 1995.

[4] J-C. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive FEC-Based Error Control for Internet Telephony," *Proceedings of IEEE INFOCOM*, March 1999.

[5] J-C. Bolot and A. Garcia, "Control Mechanisms for Packet Audio in the Internet," *Proceedings of IEEE INFOCOM*, pp. 232 - 239, March 1996.

[6] M. Borella, D. Swider, S. Uludag, and G. Brewster, "Internet Packet Loss: Measurement and Implications for End-to-End QoS," *Proceedings of the ICPP Workshop on Architectural and OS Support for Multimedia Applications Flexible Communications Systems*, pp. 3 - 12, August 1998.

[7] B. Dempsey, J. Liebeherr, and A. Weaver, "On Retransmission-Based Error Control for Continuous Media Traffic in Packet-Switching Networks," *Technical Report CS-94-09*, Computer Science Department, University of Virginia, February 1994.

[8] Free Phone, February 19, 1999. URL: *http://www-sop.inria.fr/rodeo/fphone/index.html.*

[9] V. Hardman, M. Sasse, M. Handley and A. Watson, "Reliable Audio for Use Over the Internet," *Proceedings of INET*, 1995.

[10] O. Hodson, S. Varakliotis, and V. Hardman, "A Software Platform for Multiway Audio Distribution Over the Internet," *IEE Colloquium on Audio and Music Technology: The Challenge of Creative DSP*, pp. 114 - 116, November 1998.

[11] N. Jayant and S. W. Christensen, "Effects of Packet Losses in Waveform Coded Speech and Improvements due to an Odd-Even Sample-Interpolation Procedure," *IEEE Transactions on Communications*, Vol. COM-29, No. 2, pp. 101 - 109, February 1981.

[12] I. Kouvelas, O. Hodson, V. Hardman and J. Crowcroft, "Redundancy Control in Real-Time Internet Audio Conferencing," *Proceedings of the 1997 International Workshop on Audio-Visual Services Over Packet Networks*, September 1997.

[13] 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, September-October 1998.

[14] H. Schulzrinne, "Voice Communication Across the Internet: A Network Voice Terminal," *Technical report*, Department of Computer Science, University of Massachusetts, July 1992.

[15] H. Schulzrinne, S. Casner, R. Fredrick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", *Internet Engineering Task Force*, RFC 1889, January 1996.

[16] H. Schwetman, "CSIM18 - The Simulation Engine," *Proceedings of the 1996 Winter Simulation Conference*, pp. 517 - 521, 1996.

[17] A. Watson and M. Sasse, "Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications," *Proceedings of ACM Multimedia*, pp. 55 - 60, September 1998.