ISSUES IN TRANSNET PACKETIZED VOICE COMMUNICATION

Danny Cohen
Computer Science Department
Caltech, Pasadena, California 91125
and
USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90291

ABSTRACT

In recent years, very important progress was made in both real-time digital computer communication and real-time digital voice communication, due to the growing importance of geographically distributed computation, the growing need for resource sharing, and the need for secure voice communication. The progress in digital voice communication--mainly aimed at low data rate and high fidelity speech communication--has led to various vocoding techniques such as LPC and CVSD. The progress in real-time computer communication--which is mainly aimed at low delays, high bandwidth and high cost-effectiveness--led to technologies such as packet switching, packet-radio, and satellite communication, as well as intercomputer communication protocols. It is the natural result of the progress in both areas that the use of packet switching technology be considered for voice communication. In this paper the issues involved in the application of packet-switching networks to real-time voice communication are identified and explained.

INTRODUCTION

This paper is intended to define the problems associated with real time voice communication over a packet switched network rather than solve them.

Even though our experience was gathered around the ARPANET, the following discussion of the issues holds for any packet switched network.

The model assumed here is a speaker talking into the source vocoder (SV) which converts the speech waveform into a bit stream, which is packetized by the transmitting process (XMTR) and is given to the network. At the destination the packets are given by the network to the receiving process (RCVR) which feeds bits into the destination vocoder (DV), which synthesizes a waveform, which (hopefully) sounds like the original one.

The reproduction quality of the total end-to-end communication depends on (1) the acoustic quality of the speech reproduction ("fidelity"), (2) the total delay, T , between the generation of the original waveform and its reproduction at the destination, and (3) the continuity ("smoothness") of the reproduction, C .

The quality function is not well defined. It is clear that (1) the higher the fidelity, (2) the smaller the total delay, and (3) the more continuous the communication, the better the total communication.

ON VOCODING

Digital speech representation should be compressed for efficiency of communication. The source vocoder, SV, performs the task of converting the speech waveform into a bit stream in an efficient way. This means using as low a data rate (bits/second) as possible, while maintaining a certain fidelity, or (dually) achieving the highest quality

for a given data rate. There are two main types of vocoding methods: (1) methods which result in synthesis of waveforms which LOOK like the original ones, and (2) methods which result in synthesis of waveforms which SOUND like the original ones.

PCM, DPCM, ADPCM, and CVSD are just a few methods of the first type. LPC is a method of the second type. Information theory techniques are used generally for compression of the first type. Acoustic techniques, used in the second type, extract only the dominant acoustic features, without the insignificant redundancies, and use some efficient method to transmit only these essential features. The acoustic methods usually yield better compression than the methods of the first type. However, such schemes are more sensitive to deviation of the waveform from the acoustical model on which it is based (here, human voice) and therefore is less robust than the methods of the first type.

There are three major tradeoffs: (1) compression vs. computation, (2) compression vs. fidelity, and (3) compression vs. robustness. The choice of the ideal balance is still a research topic.

Once the data rate is decided, it is not yet obvious how to use it in the most efficient manner for a given vocoding scheme. Most vocoders describe each discrete vocoding period, V , by a parcel of B bits, which is not necessarily always of the same size. The data rate utilized is $R=B/V$. One can change the data rate by changing either B or V . For example, it is possible to use 8 bits 10 KHz sampling or 10 bits 8KHz sampling. Both yield the same data rate, but they have different acoustic qualities. A similar tradeoff exists also for LPC and most of the other vocoding schemes.

In a network environment, as the total network load, $L(t)$, decreases one might wish to use a higher data rate, R , to improve the fidelity. Similarly, when $L(t)$ increases the data rate, R should be decreased to minimize delays at the cost of fidelity.

These dynamic changes of of the data rate, R , can be accomplished by changing either the parcel size, B , or the vocoding period, V . Due to hardware inflexibilities (e.g., analog filters) it is usually more desirable to change B , rather than V . However, in systems with a large vocoding period relative to the sampling period (e.g., LPC) it might be as easy to change V as it is to change B .

In variable rate systems, both B and V may change. B might change according to the complexity of the acoustic features (or their differences) in the waveform (e.g., long vowels require less description than most other sounds). V might vary according to the variability of these features.

For example, one might transmit a parcel whenever a change in the features-space exceeds a certain threshold, T . Obviously, the smaller T is, the larger R is, and vice versa. Hence by dynamically modifying the value of T , according to $L(t)$, the resulting R may be controlled.

Similarly, in PCM one may fix the sampling frequency (by keeping V constant) and vary B such that when $L(t)$ increases, the quality becomes more "grainy", and when $L(t)$ decreases, B is increased to improve fidelity.

ON TRANSNET COMMUNICATION

Figure 1 shows the major components of the data flow. The source vocoder, SV, output is shown at a constant rate, even though this is not necessarily the case. This data stream is represented by $A(1)$ through $A(10)$.

The packetized data output of the transmitting process, XMTR, the is represented by $B(1)$ through $B(10)$.

The data given by the network to the receiving process, RCVR, at the destination is represented by $C(1)$ through $C(10)$.

The data stream given to the destination vocoder, DV, is represented by D(1) through D(10). This is also shown as a constant rate flow, even though this is not necessarily the case.

Figure 1

Note that the total communication delay, T , at the time t is:

$$T(t) = V + P + N(t) + W$$

where V is the delay due to the size of the vocoding quantization (the Vocoding Period), P is the delay due to the Packetizing Period, $N(t)$ is the network delay at time t , and W is the delay due to the time that the RCVR chooses to wait from the instance of receiving the first data until handing it to the vocoder. Define $XY(i)$ as "the time from the event $X(i)$ until the event $Y(i)$ ". For example, $AB(i) = B(i) - A(i)$, $BC(i) = C(i) - B(i)$, and $CD(i) = D(i) - C(i)$. Since $D(1)$ happens W units of time after $C(1)$ we get $W = CD(1)$. Similarly, the network delay is $N(i) = BC(i)$, and the packetizing period is $P = AB(i)$.

Note that if W is chosen large enough, i.e.,

$$W \geq \max_{1 \leq i \leq M} \{N(i) - N(1)\}$$

then the output to the DV is guaranteed to be continuous (i.e., perfectly smooth)--at least up to packet M . Note also that determining the right (i.e., minimal and large enough) value for W requires prediction of the $\{N(t)\}$ into the future.

Figure 2 shows an example of the relation between the continuity, C , and the waiting period, W .

Figure 2

The exact nature of the function $C = C(W)$ is not known, but it is clear that it approaches 100% (i.e., perfect smoothness) as W approaches infinity, and that its first derivative is non-negative. Its exact shape depends obviously on the distribution of $\{N(t)\}$.

The network performance, as reflected in the $\{N(t)\}$ distribution, depends on the momentary total network load, $L(t)$, on the packet size, S , which is proportional to R and P , and probably also on the interval between the transmission of the packets. Note that if the vocoding rate, R , is fixed, then the time between successive packets is inversely proportional to S , the packet size.

V , the vocoding period, is usually very small, and insignificant for this discussion. Typically it is between 10 and 20 milliseconds. However, this parameter is most significant for the acoustic fidelity of the voice reproduction.

Therefore, the total communication quality, Q , may be viewed as a function of the data rate, R , the vocoding period, V , the packet size S , the packetizing period, P , the waiting period, W , and the variable network delay, $N(t)$. Since $N(t)$ depends on S , P and the variable network load $L(t)$, and since $S(t) = P(t) \cdot R(t)$, we obtain:

$$Q(t) = Q(R(t), S(t), W, L(t))$$

where R , S , P and W are variables which we can control (R , S and P by the XMTR and W by the RCVR) but $L(t)$ is beyond our control.

The control of packetizing period, P , in a variable rate environment is some relation between packet size, S , and the transmission time T , in the form of $F(S, T) = 0$, such that transmission occurs when this relation holds. For example, the relation $F(S, T) = T - T_1$ means that every T_1 units of time, a packet is sent, regardless of its size. The relation $F(S, T) = S - S_1$ means that whenever the packet reaches the size S_1 , it is sent, regardless of the frequency. The relation $F(S, T) = \max(S/S_1, T/T_1) - 1$ means that the packet is sent whenever it reaches the size S_1 or whenever T_1 units of time have elapsed since the transmission of the previous one. In general, any fixed transmission algorithm may be expressed as $F(S, T) = 0$. On the other hand, algorithms which are adaptive to the network load are of the form $F(S, T, L) = 0$. Figure 3 demonstrates several possible fixed transmission algorithms.

Figure 3

Representation of 4 transmission Algorithms:

- A. ——— $F(S, T) = T/T_1 - 1 = 0$
- B. - - - - $F(S, T) = S/S_1 - 1 = 0$
- C. $F(S, T) = S/S_2 + T/T_2 - 1 = 0$
- D. - . - . $F(S, T) = \max(S/S_3, T/T_3) - 1 = 0$

— — — represent the maximum and the minimum data rate of the variable rate vocoder output.

The above discussion is based on the assumption that messages are sent by the XMTR process whenever it "wants", and therefore the transmission algorithm is only a function of S and T . However, this is not necessarily the case.

Whenever the XMTR process decides to send a message, all it can do is to notify the network about it. Only when the network is ready to accept the message the real transmission commences. Ideally the network is ready to accept immediately after receiving the signal from XMTR. In reality some flow control mechanism may delay the acceptance for a number of reasons (e.g., too many unacknowledged messages). If this is the case, then the transmission algorithm should adapt itself to cope optimally with these delays. For simplicity in this discussion, let us ignore this problem of network acceptance delays.

Since at time t the packet size, $S(t)$, depends on both the actual data rate, R , and the packetizing period, P , we have $S(t) = R(t) * P(t)$. Therefore the quality $Q = Q(R, S, P, W, L)$ can be expressed as $Q = Q(R, P, W, L)$, where all these arguments vary with t .

The choice of the waiting period, W , by the RCVR has to take place at the beginning of every sequence of non-silence data flow. At the end of each sequence, the RCVR knows which value of W is optimal for the PREVIOUS (unfortunately!) sequence. By using some prediction technique one should be able to adapt the choice of W to $L(t)$.

There are several playback strategies and all produce perfectly smooth playback if W is chosen to be sufficient, i.e., $W > \text{MAX}\{N(t) - N(1)\}$. However, they differ in the event that too small a W is chosen, due to unpredicted degradation in the network performance. The two main flavors which we use are (1) strategy A, which tries to maintain the rhythm at the cost of losing information, and (2) strategy B, which tries to minimize information loss at the cost of the overall rhythm. There is a continuum between the two approaches and one can design many other playback strategies which maintain other relations between maintaining rhythm and information loss.

Figure 4 shows a choice of W which is optimal, in the sense that it is the minimal delay which guarantees that the continuity is perfect and no data is lost. Both Figures 5 and 6 show a value of W which is too small for the same network performance with the same timing. Figure 5 shows strategy A, which maintains the overall rhythm at the expense of discontinuities ("glitches") and loss of data both at packet number 6 and packet number 8. Figure 5 shows strategy B, which has only one "glitch" at packet number 5 and does not lose data at the expense of the total rhythm, as seen from $AD(10) > AD(1)$.

Figure 4

Here $W = N(5) - N(1)$, which is the best choice of W .

Note that $AD(10) = AD(1)$, meaning a fixed delay (i.e., rhythm preserved) along the entire sequence.

The above discussion was aimed at the question of how to best utilize the network under its current load. It might happen that the network load is so high that even the best available performance is below any acceptable level. In this case some other measures must be taken, such as priority assignment and removing the less essential load from the network temporarily.

A packet-switching network might have a flow control mechanism that limits the number of pending messages (i.e., unacknowledged messages or messages in transit). This mechanism might have some side effects which might cause adaptive sending algorithms to oscillate. It is possible that the way in which the flow control is implemented would bias the transmission algorithm in favor of shorter messages or longer ones. On the other hand, non-adaptive transmission algorithms might cause a total disaster under such conditions. For example, due to increased load, $L(t)$, on the network the $\{N(i)\}$ increase, the total round-trip time increases, and the rate in which the network accepts messages decreases. Once it becomes less than the packetizing period, P , the communication breaks because of what looks like "the network cannot keep up with the required bandwidth". However, the situation could be improved by increasing the message size, S , and hence the packetizing period, P . Adaptive transmission algorithms are expected to do just this.

Another important issue is the definition of total network optimization. Obviously, local optimization by each user does not lead necessarily to global optimization. This issue has to be addressed together with the notion of communication cost, since the cost should be defined in such a way as to encourage users to optimize their use of the communication medium in a manner that is compatible with the global optimization. For example, a cost function which has a fixed cost associated with each message plus a variable cost related (monotonically) to its length tends to encourage users to use longer messages. This may be suboptimal for the user (because of the additional delay associated with the message length), but it is better for the entire network (since overhead is reduced, store and forward buffers are better utilized, etc.), hence helping the user indirectly.

Figure 5

Strategy A:

Same $\{P(i)\}$ and $\{N(i)\}$ as in Figure 4.

W is too short since $W < N(5) - N(1)$.

Information loss, and discontinuities at packets number 5 and 8, since $N(5) > N(1)$ and $N(8) > N(1)$.

Rhythm preserved as seen from $AD(10) = AD(1)$.

Figure 6

Strategy B:

Same $\{P(i)\}$ and $\{N(i)\}$ as in Figure 4. Also same W as in Figure 5, which is too short since $W < N(5) - N(1)$.

Discontinuity only at packet number 5 but not at number 8, even though $N(8) > N(1)$.

Total rhythm modified as seen from $AD(10) > AD(1)$.

No information loss.

SUMMARY

In order to optimize the overall quality of voice communication over a packet-switched network, several parameters have to be dynamically adjusted:

- (1) Both the packetization algorithm at the time, t , and the vocoding data rate, R , should vary as a function of the network load, $L(t)$. This should be done by the XMTR process at the source.
- (2) The waiting period, W , at the destination (by RCVR) as a function of the network performance, which depends also on $L(t)$ and is reflected in the distribution $\{N(t)\}$.
- (3) In addition, as a function of the network performance, the vocoding parameters may be dynamically varied to affect the fidelity/data-rate tradeoff, e.g., improving the fidelity by using higher data rates when the network is lightly loaded. This is also performed, obviously, by the XMTR.
- (4) Since all the above parameters depend heavily on the network performance, a systematic way of predicting that performance must be developed. This will probably be done best by measuring and smoothing (a la Kalman filtering techniques).

The following paper discusses other aspects of transnet packetized voice, and is recommended for reading:

"System Implications of Packetized Voice", by Gino J. Coviello et al. Printed in the *Proceedings of the 14th Annual Technical Symposium*, sponsored by the ACM, June 1975.