Abstract

This paper discusses a playout controller called NetEQ. The focus is on explaining how this algorithm can help to improve the over all QoS in a VoIP system. Traditional alternative solutions are described and compared with the way that NetEQ operates. Performance results are provided that quantify the quality benefit of NetEQ, both in terms of delay and in terms of packet loss robustness.

Playout Controller for Real Time Speech/Audio Applications

Henrik Åström, Global IP Sound AB

1 Introduction

The demand for higher flexibility and reduced costs is driving the convergence of voice and data traffic to the same network. These networks are based on the Internet Protocol, which use a packet transmission paradigm that is focused on data communication. Naturally, a major challenge is to achieve good quality for real time voice communication in such packet networks. The transmission of voice over IP networks is generally referred to as VoIP.

Five major factors are associated with packet networks and have a significant impact on the perceived speech quality: speech coding, echo, delay, jitter, and packet loss. The last three factors stem from the nature of a packet network. There is no guarantee that a packet of speech data will arrive at the receiving end in time, or even that it will arrive at all.

Most of the existing speech processing algorithms currently deployed in VoIP requires tightly managed networks that are highly over provisioned in order to guarantee the quality. This is of course a way to get around the problems to some extent, but such a solution is both inflexible and limits the over-all cost saving.

As the cost drives the voice traffic to cheaper solutions, such as the Internet, the market needs more stable speech processing algorithms to maintain the performance. The customers are already used to a certain quality through the PSTN and will not accept much degradation when switching to VoIP.

This paper discusses NetEQ, an IP edge-point algorithm that can efficiently minimize the quality problems due to delay, jitter and packet loss.

1.1 Delay

An extremely important parameter in two-way communication is the end-to-end delay between the sender and the receiver. Delay can lead to two problems, namely echo, and talker overlap.

There are many sources of delay in a VoIP call as depicted in Figure 1. The delays can be grouped into three major categories - *algorithmic delay*, *processing delay*, and *network delay.*

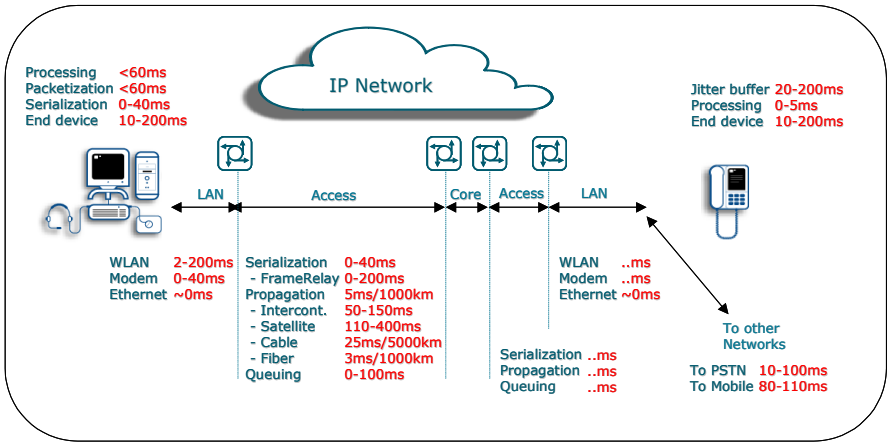

Figure 1: *Typical delay sources in a VoIP system.*

If the roundtrip delay is more than about 30 ms, any audible echo will be annoying (refer i.e. [1]). As long as the delay is not too long, echo cancellation algorithms remove most of these problems. For very long delays (above 200 ms),

even if echo cancellation is deployed, it is hard to maintain a two-way conversation without getting disturbing double talk. The double talk occurs when the long response time affects the naturalness of the conversation and people start to "step on each other's talk".

The algorithmic delay and the processing delay usually add up at least 20-60 ms delay. As Figure 1 shows, the transmission delay can very a lot depending on the technology used. Delays of more than 100 ms are fairly common, especially for intercontinental calls.

1.2 Jitter

In contrast to the constant algorithmic and processing delay, the transmission delay varies over time. The reason is that the transit time of a packet through an IP network will vary due to queuing effects. The jitter can be defined as a smoothed function of the delay differences between consecutive packets over time.

The jitter present in packet networks complicates the decoding process in the receiver device since the decoder needs to have packets of data readily available at the right time instants. If not, the decoder will not be able to produce continuous speech. Thus, in addition to adding to the delay the jitter results in a timing problem for the receiver.

A jitter buffer is required to make sure that packets are available when needed for play-out, as will be discussed in Section 2.1.

1.3 Packet loss

In general packet loss occurs either if a router in the network drops a packet or if a packet arrives too late to be handled by the decoder. By allowing for a long delay in the jitter buffer the latter type of packet loss can be almost completely removed, but at the price of increased system delay.

When packet loss occurs some mechanism for filling in the missing speech must be incorporated. Such solutions are referred to as *error concealment* algorithms or *packet loss concealment (PLC)* algorithms. For best performance these algorithms have to accurately predict the speech signal and make a smooth transition between the previous decoded speech and inserted segment.

1.4 Clock drift

Whether the communication endpoints are gateways or other devices, low-frequency clock drift between the two can cause receiver buffer overflow or underflow. Simply speaking, this effect can be described as the two devices talking to each other have different time references. For example the transmitter might send packets every 20 ms according to its perception of time while the receiver's perception is that the packets arrive every 20.5 ms. Hence, for every 40th packet the receiver has to perform a packet loss concealment in order to avoid buffer underflow. If the clock drift is not detected accurately on the two sides, delay will build up on one side and the other side will experience quality problems due to under runs. Consequently, clock drift could be a major issue that can have significant effect on the speech quality.

1.5 Summary of requirements for VoIP

- Maintaining low delay is very important in VoIP applications. Similarly, the packet arrival jitter has to be handled in order to produce continuous speech. Hence, an efficient jitter buffer is necessary.
- Packet loss concealment is necessary, should produce smooth speech, and has to be able to accurately predict several lost frames in a row.
- Clock drift must be handled appropriately

2 Typical Solution

In the previous section some of the major issues related to speech quality in a VoIP system were discussed. In order to obtain a high quality solution many demanding requirements have to be met. The designer of a VoIP end-point such as a gateway must implement efficient and effective solutions to the jitter-, packet loss-, and clock drift problems. In addition, every possibility to reduce delay has to be considered.

2.1 Jitter Buffer

A jitter buffer is required to make sure that packets are available when needed for play-out. It removes the jitter in the arrival times of the packets at the cost of increase in the overall delay. The objective of a jitter buffer algorithm is hence to keep the buffering delay as short as possible while minimizing the number of packets that arrive too late to be used. A larger jitter buffer causes increase in the delay and decreases the packet loss. A smaller jitter buffer decreases the delay but increases the packet loss.

The traditional approach is to store the incoming packets in a buffer (packet buffer) before sending them to the decoder. The most straightforward approach is to have a buffer with a fixed number of packets. This results in a constant system delay (if there is no clock drift) and requires no computations and therefore gives a minimal complexity. The drawback with this approach is that the length of the buffer has to be made sufficiently large to accommodate the maximum jitter (which in practice occur quite seldom).

In order to keep the delay as short as possible it is important that the jitter-buffer algorithm adapt rapidly to changing network conditions. Therefore, jitter buffers with dynamic size allocation, so called *adaptive jitter buffers*, are most common nowadays.

The adaptation is achieved by inserting packets in the buffer when the delay needs to be increased and removing packets when the delay can be decreased. The insertion of packets is usually done by repeating the previous packet. Unfortunately, this will almost always result in audible distortion and hence most adaptive jitter buffer algorithms are very cautious when it comes to delay adaptations in order to avoid such effects.

This traditional packet buffer approach is limited in its adaptation granularity by the packet size since it can only change the buffer length by adding or discarding one or several packets.

Another major limitation of traditional jitter buffers is that, in order to limit the audible distortion when removing packets, it can only be done during periods of silence. Hence, delay builds up during a talk spurt and it can take several seconds before a reduction in the delay can be achieved. Also, high delay at the end of a talk spurt will have a severe effect on the conversation since it increases the probability of double talk. Conversational tasks hence become much easier if the delay can be adapted even during active speech and is kept to a minimum at the end of each talk spurt.

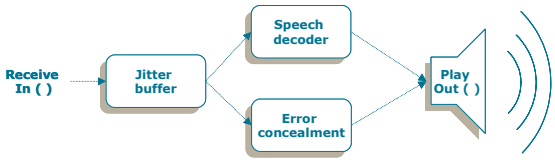

Figure 2: *Traditional system with separated Jitter buffer and packet loss concealment*

Some of the current implementations of adaptive jitter buffers have shown to experience problems when there are packet losses in the network. For example, studies for TIPHON in [2] show that the jitter buffer delay can increase a lot for cases where packet losses are present.

2.2 Packet Loss Concealment

Until recently, two simple (codec independent) approaches for PLC have prevailed. The first method, referred to as *zero stuffing* (ZS), is obtained by simply replacing a lost packet with a period of silence of the same duration as the lost packet. The second method, referred to as *packet repetition* (PR), assumes that the difference between two consecutive speech frames is quite small. Hence, the lost packet is replaced by simply repeating the previous packet. In practice, though, even a minor change in for example the pitch frequency is easily detected by the human ear. In addition, it is virtually impossible to achieve smooth transitions between the packets with this approach. However, this approach performs fairly well for very small probabilities (less than 3 %) of packet loss.

Recently, the ITU standardized a method for packet loss concealment in G.711 Appendix I (usually referred to as G.711 PLC). This is a more sophisticated method that tries to estimate the lost packet from previously decoded speech and hence cannot be implemented in the packet buffer. The performance of this algorithm is discussed in Section 3.2.

Figure 5 in section 3.2 shows MOS results from a listening test comparing the performance of these three PLC algorithms. Clearly, packet repetition outperforms zero stuffing but both methods are very sensitive to packet loss compared to the more advanced methods.

Some codecs, such as those based on CELP, have their own built-in PLC algorithm. In many cases this gives a decent concealment during the loss. Unfortunately many of these codecs suffer from their packet inter-dependencies instead. After a packet loss it usually takes several correctly received frames before the state is fully recovered. In general this leads to a smaller distortion that spreads over a much longer time period. Therefore, the quality still degrades a lot when there are packet losses.

2.3 Clock Drift handling

The traditional approach (in a TDM network) is to, at the receiver, deploy a clock synchronization mechanism to correct for clock drift by comparing the number of samples received with the local clock. In a VoIP network, on the other hand, it is hard to do reliable clock drift estimation. The reason is that the estimates only can be based on averaging packet arrivals at a rate of typically 30 - 50 per second instead of averaging on a per sample basis at a rate of 8000 per second (as done in TDM networks). In addition, because of the jitter present in VoIP it is almost impossible to obtain an accurate estimate of the clock drift and hence many algorithms designed to mitigate this effect fail.

3 The NetEQ™ Algorithm

Global IP Sound has developed NetEQ, an advanced playout controller that includes both delay adaptation and error concealment. Combining adaptive delay control and packet loss concealment into one unit makes NetEQ unique. The patented approach allows NetEQ to quickly and with high resolution adapt to changing network conditions and to ensure high speech quality with minimal buffer latency. This can be achieved because NetEQ is working on speech segments from the decoder and not on entire packets in the packet buffer.

NetEQ is a piece of embedded software that resides between the IP network interface and the sound playout mechanism, whether that would be an analog-to-digital converter or a TDM (time-division multiplexing) interface. A simple block diagram of NetEQ is depicted in Figure 3. The algorithm works well for both gateway applications using narrowband 8 kHz sampling frequency and for wideband solutions with 16 kHz sampling frequency (wideband is possible when not interfacing with PSTN).

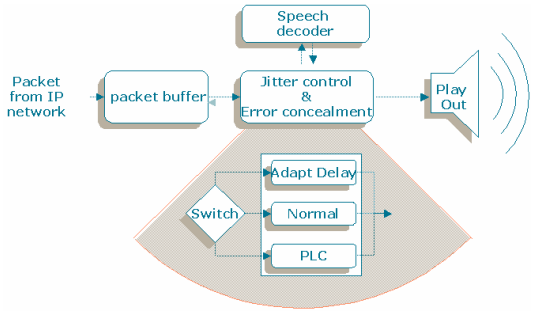

Figure 3: *Simple block diagram depicting how NetEQ interacts with the speech decoder and the packet buffer.*

NetEQ has interfaces to the two main flows on the receiver side: the process that receives packets from the network side and the process that requests more data to the playout buffer. From the RTP header information NetEQ can arrange the packets in sequential order and determine if a packet has been lost or not. When NetEQ gets a request from the playback side it checks whether it has data in the packet buffer or not. If it has available data, then it decodes it otherwise it produces synthetic speech as will be discussed below.

A packet loss is always troublesome for the receiver side since the segment that is lost will always have to be replaced in some way. This replacement is not trivial, and most of the traditional packet concealment algorithms, as discussed in Section 2.2, are not capable to do this while maintaining acceptable quality.

NetEQ performs error concealment by producing short segments of PLC data several times rather than producing a whole frame at once. Basically, a simple, yet efficient, speech synthesis model is used with parameters estimated from adjacent frames. Before producing a new short synthetic

speech segment NetEQ checks if a new packet has arrived. As soon as a new packet is available it is decoded and played out. In this way only the minimum amount of synthetic data is inserted. NetEQ also has a unique way of merging two frames of speech without audible distortion.

The jitter adaptation is done on short speech segments instead of on entire packets. Because of this NetEQ can make very small delay adaptations that are independent of frame boundaries. These adaptations are performed by signal processing algorithms that guarantee minimal resulting distortion. This enables NetEQ to make jitter adaptations in both speech and non-speech, which ensures low delay both during speech and during silence.

The superior performance is explained by the fact that NetEQ is based on frequently making small inaudible changes instead of making fewer clearly audible changes. Furthermore, NetEQ does not have to estimate the actual clock drift but is able to mitigate the effect of it automatically. It is therefore a very attractive alternative to a standard clock synchronization mechanism.

3.1 Delay performance

Studies were carried out to compare the delay of NetEQ with alternative solutions. The graph below illustrates the delay performance of different algorithms on a channel with quite a lot of jitter. NetEQ manages to keep the delay much lower than both the adaptive and the fixed jitter buffer. In general a delay improvement of 30-80 ms can be expected with the NetEQ algorithm compared to traditional approaches. The graph also shows that NetEQ adapts to the envelope of the jitter very efficiently, since the delay is reduced in less then a second after a jitter peak.

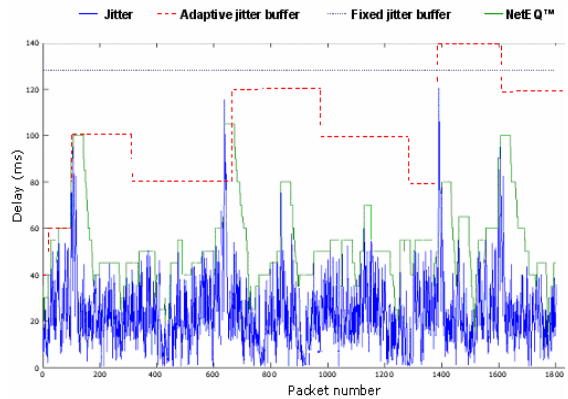

Figure 4: Delay performance for different jitter buffers on a system with 20 ms packets. Observe that the constant system delays have been removed from all curves. To include it, add another 80 ms to all the curves.

3.2 Speech Quality

Measuring speech quality is a very delicate task; Absolute Category Rating (ACR) using the Mean Opinion Score (MOS) is the most common approach. In short, a number of people are chosen to reflect the possible end-users of the system. They listen to a number of speech stimuli and rate each on a five-grade scale with the following weights:

1 = Bad, 2 = Poor, 3 = Fair,

4 = Good, 5 = Excellent

The independent test laboratory COMSAT conducted a MOS test, Figure 5, that compares NetEQ with ZS, and the ITU G.711 Appendix I PLC. It is obvious that the NetEQ algorithm outperforms both ZS and the ITU PLC. What is also shown is that NetEQ together with an internally robust codec (GIPS Enhanced G.711) can deliver almost the same quality for channels with >20% packet loss as on a loss less channel.

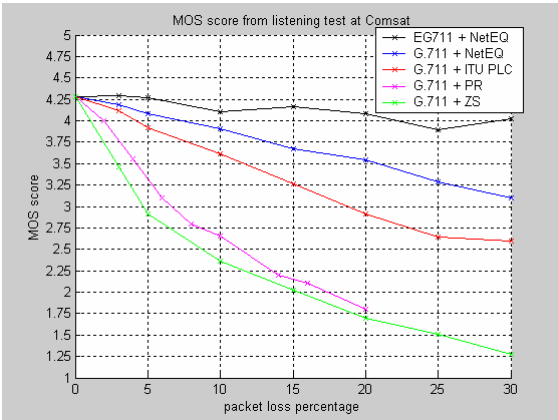

Figure 5: *MOS test showing the speech quality of different PLC algorithms at different packet loss rates. The standard jitter buffer is a fixed jitter buffer with infinite delay. The MOS test was performed at the independent test laboratory Comsat (except for the PR algorithm that was inserted from another similar test)*

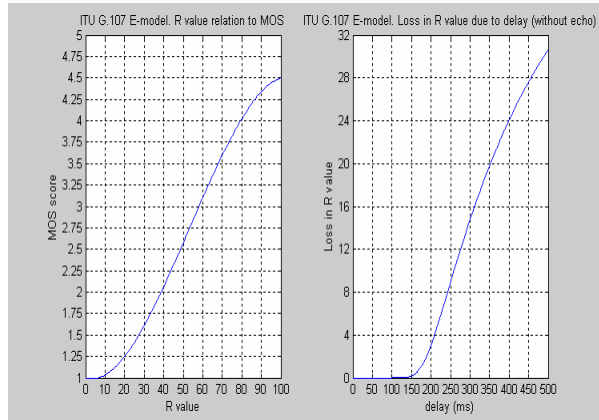

Figure 6: *Left, the relationship between the R-value and the MOS value. Right, Id as a function of delay (when echo is assumed to be zero). Both curves are according to ITU G.107 E-model*

3.3 Total Quality

The total perceived quality is actually a combination of the delay and the speech quality. In the MOS test presented in the previous section the delay advantage of NetEQ was not taken into account at all. The reason is that MOS tests are performed with pure listening and does not include any two-way conversations. It is however possible to estimate how the delay affects the quality by using the standardized ITU G.107 E-model.

The E-model provides a good tool for estimating the combined quality effects in an end-to-end system with several transmission parameters. In the model a transmission rating factor, R, is defined that has values between 0 and 100. The left curve in Figure 6 shows how the R-value relates to the MOS score.

R is calculated as:

$$R = Ro - Is - Id - Ie,eff + A$$

Where:

Ro: Principle signal-to-noise due to the effects of the noise in the room

Is: Impairments that are caused by for example quantization problems or DC noise

Id: Impairments due to delay, both in terms of echo and absolute delay.

Ie,eff: The impairment due to the speech codec and impairments due to packet loss.

A: Advantage factor due to for example that it enables increased mobility

As we stated before the MOS test does not take any consideration to the delay impairments. It does however already show the effects of the recording, quantization, speech coding algorithm and packet loss. If we assume that the advantage factor is set to 0 (which is the normal case), then we can get the simple formula:

$$R = Rmt - Id$$

Where

Rmt: The R value that corresponds to the MOS value that was received in a MOS test.

With this expression we can now include the effect of delay and calculate MOS curves that take the delay effects into account. If we assume that the echo is taken care of perfectly and that the absolute transmission is Ta, we have according to the E-model:

$$Id = 0 \text{ for } Ta<100 \text{ ms}$$

$$Id = 25((1+X^6)^{1/6} - 3(1+(X/3)^6)^{1/6} + 2) \text{ for}$$

$$Ta >= 100 \text{ ms, where}$$

$$X = \log(Ta/100)/\log(2)$$

As we saw earlier, NetEQ has an over-all delay advantage of 30-80 ms compared to the existing solutions. We can use the delay measures from Figure 4 to obtain some real values of the delays. Counting conservatively we see that NetEQ has an absolute average delay of 150 ms and that the standard adaptive jitter buffer has a average delay of 190 ms. With these estimates it's possible to calculate the Id in the E-model:

$$Id(Ta = 150 \text{ ms}) = 0.1635$$

$$Id\ (Ta = 190 \text{ ms}) = 2.1114$$

Using the two curves in Figure 6, we can see how much the MOS score is affected due to the delay. So, by using the above curves it's possible to adjust the MOS result from figure 5, so that we get a MOS score that accounts for both speech quality and delay. The new MOS curve is shown in Figure 7. This is not a MOS curves in the traditional meaning, but it gives a fair estimate of the true perceived quality.

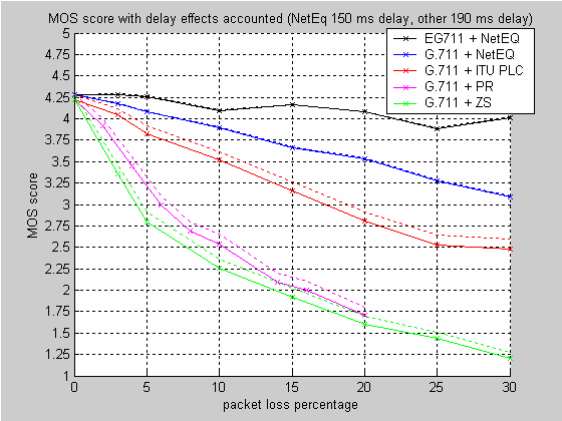

Figure 7: *MOS score with delay effects accounted for according to the E-model (dashed lines indicate MOS score without regards to the delay).*

As the graphs indicate, delays below 150 ms are not degrading the over-all quality all that much (when the echo cancellation is perfect). However, for delays larger than 150 ms it is obvious that the delay is a very important parameter that has to be taken into account. In the above example where the delay is 190 ms, the MOS has dropped roughly 0.10 compared to the MOS score with zero delay.

The NetEQ algorithm ensures that the delay is always kept at a very minimum. Thus it enables good quality for two-way conversation and limits the effect of talker overlap.

4 Summary

Voice over Packet solutions offer lots of cost savings and new features. However, because of the delay, jitter and packet loss that inevitably exists on VoIP networks it is not trivial to go from a circuit switched network to a packet switched network. Most of the current algorithms for VoIP fail to deliver good quality when the network is congested.

The NetEQ playout controller was created to ensure that the perceived quality is high even during periods of high jitter and packet loss. By using the combined knowledge about how IP networks work and how speech can be modeled by signal processing techniques, the algorithm optimizes the performance both in terms of delay and quality. As was shown in this paper, the result is that NetEQ outperforms standard solutions both in terms of delay, speech quality and over all robustness.

References:

[1] ETSI TR 101 329 v2.1.1: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); General aspects of Quality of Service (QoS)"

[2] ETSI TR 101 329-6 v2.1.1: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 3; End-to-end Quality of Service in TIPHON Systems"

[3] ITU G.107 "The E-Model, a computational model for use in transmission planning"