Proc. Int. Conf. Wireless Commun., Signal Processing (Nanjing, China), 5 pp., Nov. 2009

A New Optimum Jitter Protection for Conversational VoIP

Qipeng Gong, Peter Kabal

Electrical & Computer Engineering, McGill University
Montreal,Quebec,Canada H3A 2A7
qi.gong@mail.mcgill.ca
peter.kabal@mcgill.ca

***Abstract*—In Voice-over-IP, jitter buffers are introduced at both sides of the sender and the receiver to compensate for delay jitters. A longer buffer reduces the possibility of packet loss and packet disorder at the expense of increasing conversational delays. In this paper, we propose a novel criterion for the calling quality of conversational VoIP, including the effect of delay on interactivity of a conversation. Using this criterion, we propose a quality-based playout scheduling algorithm with improved voice quality and reduced conversational delays. The Simulation results show that the proposed algorithm can achieve the best calling quality compared with other algorithms.**

I. Introduction

Voice over IP (VoIP) is a technology to exchange voice packets over the public Internet. The major challenge facing VoIP is how to match the voice quality with that provided by the traditional telephone network, i.e. PSTN. Therefore, for real-time voice communication over IP, the requirements on delay and packet loss are stringent to maintain proper QoS. However, the IP service model is “best effort”, which makes no guarantee on quality. To get higher voice quality in VoIP, efforts have been made in the literature to reduce the delay, smooth the delay variation (also known as jitter) and conceal packet losses [1].

Since the transmission delays for the delivered packets over IP are always varying, the jitter-free playout scheme would lead to the increasing rate of packet loss and the misordered recovery voice, which impact the voice quality dramatically. In practice, a jitter buffer is introduced at the receiver side to compensate for jitter effect. The size of this buffer can be fixed or adaptive. The trade-off of the size of this buffer is that the longer buffer increases the probability that a packet arrives before it is scheduled for playout, at the expense of increasing the conversational delay to break the interactivity of the conversation. The interactivity of the conversation can be transparent if the end-to-end delay is less than 150ms [2]. ITU recommends that the upper limit of end-to-end delay is 400ms [2]. Therefore, for applications which experience the long network delays, it is desirable to keep the size of jitter buffers as low as possible. Many solutions have been proposed to design a jitter buffer in the past 25 years, and [3] gave a survey and an analysis of most classic approaches in this area.

In current quality-based buffer design/optimization for VoIP, voice quality is used as the key metric since it links directly to end-user perceived quality. Most methods (see [3]) use the E-model [4] to predict voice quality. Some algorithms, see for example [5], use packet loss rate (PLR) as the cost index for designing a buffer size. However, a low PLR does not guarantee high quality, especially in the case that a PLR is reduced by choosing a large buffer to protect more late packets. For conversational VoIP, a conversational delay also plays a very important role for calling quality. A large conversational delay would cause double talk, echo or even the termination of the conversation. However, current E-model does not take a conversational delay into account. Therefore, we propose a new criterion for the optimization of calling quality: maximize voice quality and reduce conversational delays as much as possible.

Accordingly, our quality-based playout scheme is designed by two parts. To maximize voice quality, We use the R factor in E-Model [4] as the cost index. As other quality-based algorithms, the estimate of delay distribution is required in our scheme. Assuming the shape of the distribution tail is known, many works use a priori selected distributions to estimate the CDF of the delay distribution, for examples, Exponential in [6], Pareto in [7], Weibull in [8]. However, it was noticed that the playout delay may be very sensitive to the type of distribution used [9]. Therefore, we choose the statistical model based on the histogram which is more general and makes no assumption on the delay distribution. To reduce conversational delay, we adopt the idea in [5] which takes advantage of using “hangover” frames, which trigger the compression of the decoded voice to decrease a jitter buffer depth at the end of the “talk-spurt”. The simulation results show that our jitter buffer gives better calling quality with shorter conversational delays.

The contributions of this paper are three-fold:

1) A new quality assessment for conversational VoIP which take into account both voice quality and conversational delays
2) A new optimum jitter buffering based on this new criterion
3) A practical way to calculate conversational delays which is more reasonable for human’s perception.

The paper is organized as follows: the VoIP system used

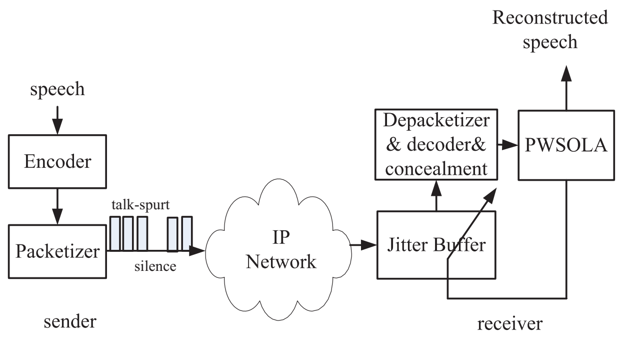


Fig. 1: VoIP System.

in this paper is introduced in Section II; in Section III, we propose a new quality assessment for conversational VoIP which includes voice quality and conversational interactivity; an optimum playout schedule algorithm is proposed in Section IV. Finally, simulations and conclusion are presented in Section V and Section VI.

II. VoIP System

The VoIP system used in this paper is shown in Fig. 1. At the sender side, the speech is first encoded using G711 encoder, and then encapsulated into IP/UDP/RTP packets, whose format follows [10]. In this paper, the payload of a packet is 20ms.

At the receiver side, the packet is first put into the buffer to smooth out the delay jitter caused by various delays over the network, and the packets inside the buffer are sorted in order. The jitter buffer is designed based on optimization of calling quality which is described in Section IV. Each packet in the buffer is de-packetized into bit stream before it be decoded by the G.711 decoder. A packet loss concealment (PLC) technique is used to handle the lost packets. Finally, the time-warping speech signal using PWSOLA is played out and the depth of jitter buffer is varied by $(\alpha - 1) \times T_F$, where α is the scaling factor and T_F is the payload length of a packet. In Section IV, we will describe how to design α in detail.

For conversational applications, both sides are sender and receiver. A jitter buffer is used at both sides for the received packets.

III. Novel Quality Assessment for Conversational VoIP

For VoIP applications, the calling quality is of the most concern. With conversational interactivity in mind, we propose a new quality assessment for conversational VoIP, which takes into account both voice quality and conversational delays. The Mean Opinion Score (MOS), a scale from 1 to 5, is typically used for assessing voice quality. However, such a subjective assessment is costly to get and is unsuitable for a real-time application. Fortunately, ITU provides a computational assessment based on network components (delay and loss): the E-model, which can be used to estimate MOS using the formula in [4] (Equation B-4). Besides, a conversational delay also plays an important role on perceived quality for conversational VoIP. A long conversational delay would break up the interactivity of a conversation. In this paper, we also give a practical way to calculate conversational delay.

A. E-Model

E-Model is a computatioanl model provided by ITU-T which assesses the combined effects of variations in several parameters. The output of the model, R factor, can be used to estimate customer opinion. According to [4], the R factor can be written as

$$R = 93.2 - I_e - I_d, \tag{1}$$

where I_d is the delay impairment factor, and I_e is the equipment impairment factor. I_d can be derived by a simplified fitting process from [8],

$$I_d = 0.024d + 0.11(d - 177.3)\,H(d - 177.3), \tag{2}$$

where

$$H(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0. \end{cases}$$

The equipment impairment factor is codec dependent. For G.711 with PLC, it can be approximated as [9]

$$I_e = I_{ec} + I_\rho = 0 + 7\ln(1 + 50\rho), \tag{3}$$

where I_{ec} is the impairment caused by encoder, which is 0 for G.711, and ρ is the packet loss including network loss and the loss caused by jitter buffer.

Defining I_m as the impairment caused by loss and delay, it can be expressed as

$$I_m = I_e + I_d, \tag{4}$$

Then, (1) can be expressed as

$$R = 93.2 - I_m. \tag{5}$$

B. Conversational Delay

In [11], conversational delay is defined by: the time interval between when User 1 stops speaking and when User 1 hears the User 2's response. Mathematically, we formulate it as:

$$D_{cov} = t_{play}(i_{user2}) - t_{send}(j_{user1}) + D_{codec} \tag{6}$$

where D_{cov} is the conversational delay, D_{coded} is the delay caused by the codec. $t_{play}(\cdot)$ is the time scheduled to play, $t_{play}(\cdot)$ is the sending time. i_{user2} is User 2's first packet of the first talk-spurt and j_{user1} is User 1's last packet of the last talk-spurt.

Obviously, to calculate D_{cov}, the key is how to detect the start and the end of a talk-spurt. The first packet of a talk-spurt can be recognized by M field of RTP header, which is 1 for the first voiced packet after silence period [10]. The problem is how to define the end of the talk-spurt. From our observation on listening tests, we noticed that people tend to start replying when the "hangover" packets are received. According to the definition of "hangover" in [12], the "hangover" packet is actually the unvoiced packet which is sent as a voiced packet to avoid speech clipping. So it is reasonable to define the end of talk-spurt when the first "hangover" packet is perceived. In our system described in Section II, the latest version of VAD/DTX from the G.729 [12] is used on a receiver's side to detect the "hangover" packet to calculate D_{cov}.

Overall, our new quality assessment can be expressed as

$$Q_{conversation} = R + g(D_{cov}). \tag{7}$$

It is still an open question to define $g(\cdot)$, and it would be our future focus. Fortunately, the relation between $Q_{conversation}$ and D_{cov} is known: $Q_{conversation}$ goes down when D_{cov} goes up, and vice versa. Hence, $Q_{conversation}$ can be optimized by maximizing R factor and minimizing D_{cov}.

IV. Design of Playout Scheduling Algorithm

Human Speech consists of silence and one or more talk-spurts. Packet losses during talk-spurts decreases the perceived quality dramatically, while losses during silence period cause almost no effect on the perceived quality. Therefore, most playout scheduling algorithms tune a jitter buffer at the beginning of each talk-spurt. Compared with continuously updating approaches, a per-talkspurt approach takes the advantage of producing a smoother playout voice. In this paper, we aim to design a playout scheme based on the optimization of calling quality. According to Section III, the optimization of calling quality for conversational VoIP is equal to optimize voice quality, i.e. R factor, and keep the conversational delay as low as possible.

Recently, many quality-based scheduling algorithms have been developed by maximizing the perceived voice quality, for examples [7] and [8]. Although these methods can reduce a conversational delay in some sense, we believe that it can be done further. In [5], a playout algorithm was proposed to reduce conversational delays with zero initial buffering while providing jitter protection to most voiced packets. Fig. 2 describes the jitter management in details. According to [5], a conversational delay was reduced by two steps: first, playout the first packet of a talk-spurt as soon as it arrives; second, compress the voiced packets in a jitter buffer whenever the "hangover" packet is detected. The proposed algorithm adopts these two steps to reduce the conversational delay.

In [5], the maximum depth of the jitter buffer d_{max} is predetermined and fixed for all talk-spurts, the value is recommended to be more than 60ms to achieve an acceptable level of packet loss. The problem is that the propagation delay distribution is unknown, which makes it hard to choose a proper d_{max}. Although a large value can be chosen for d_{max} to reduce the probability of packet erasures, this increases the mouth-to-ear delay and degrades voice quality (see Fig. 3). Moreover, a large d_{max} also increases the conversational delay and accordingly increases the risk of disrupting the conversation interactivity. In Section V, the experimental result would be presented to prove it.

In this paper, we propose an optimum jitter buffering algorithm based on the optimization of calling quality defined in (7), that is, the optimization of voice quality and keeping

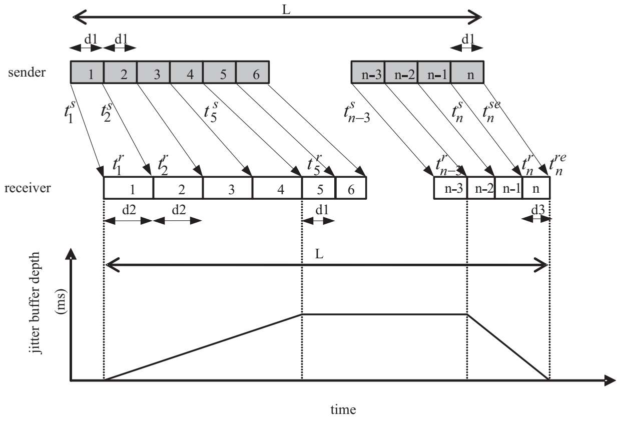


Fig. 2: Delay-Free Jitter Buffer.

the conversational delay as low as possible. The following operations are performed:

- During a silence period, comfort noise is played out every 20ms, no matter whether the SID packet arrives or not. The jitter buffer size is zero. Information about occurred packet losses and transmission delay are stored.
- When the first voiced packet of the first talk-spurt arrives, PWSOLA is applied to stretch the decoded speech before it is played out. The jitter buffer size increases by $(\alpha - 1) \times T_F$. The d_{max} is estimated based on previously stored information (window size is 1000 packets)
- When the estimated d_{max} is achieved, the decoded speech is not stretched any further. The depth of jitter buffer keeps the maximum value d_{max} and $\alpha = 1$.
- At the end of a conversation turn, when the hangover is detected, PWSOLA is applied to compress the decoded speech before it is played out. The jitter buffer size decreases by $(1-\alpha)\times T_F$. Compression stops when jitter depth is decreased to zero. It is possible for "hangover" to happen in the middle of the talk-spurt (see Fig. 6), for example, the silence gap within a word. In this case, we stretched the subsequent voiced packet as if it were the beginning of the talk-spurt. A noticeable silence gap can be avoided [5].

In our proposed algorithm, two variables, α and d_{max}, need to be designed. α is chosen as follows: $\alpha \geq 1+T_p/T_F$ (T_p is one pitch period) during the stretching process; $\alpha \leq 1-T_p/T_F$ during the compression process; $\alpha = 1$ during a silence period or when the estimated d_{max} is reached. For each talk-spurt, d_{max} is estimated based on maximizing the expected voice quality. According to (5), the maximization of the R factor is equal to minimizing I_m, which is the function of playout delay given the network loss.

According to (2) and (3), (4) can be expressed as:

$$\begin{aligned} I_m &= I_d + I_\rho \\ &= 0.024d + 0.11(d - 177.3)\,H(d - 177.3) \\ &\quad + 7\ln\big(1 + 50(\rho_n + \rho_d)\big), \end{aligned} \tag{8}$$

with ρ_n is the network loss and ρ_d is the loss caused by buffer, which depends on the playout delay. ρ_d can be calculated as

$$\begin{aligned} \rho_d &= (1 - \rho_n)P(X > d) = (1 - \rho_n)(1 - P(X \leq d)) \\ &= (1 - \rho_n)(1 - F(d)), \end{aligned} \tag{9}$$

in which $F(d)$ is the CDF of delay. In this paper, $F(d)$ is calculated as a function of playout delay using the histogram of the most recent w packet delays. In our simulation, $w = 1000$ packets. Fig. 3 shows I_m vs. d for our trace. With the optimum playout delay d that minimizes I_m, it is easy to get an optimum d_{max} for each talk-spurt. For a conversational application, this design is used on the both sides of sender and receiver.

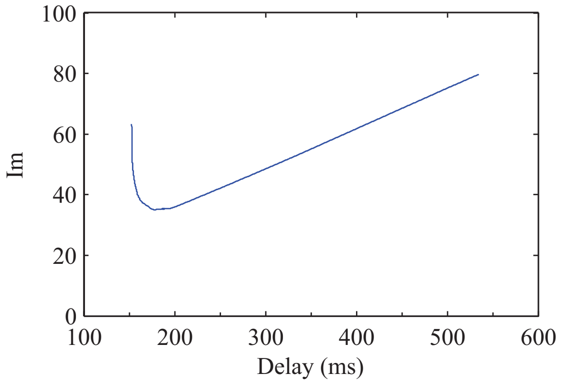


Fig. 3: I_m vs. d.

V. Results

To simulate the transmission over the internet, we use a trace file from our database to obtain the network delay for each packet, and Fig. 4 shows the distribution of the network delay. It is a typical one which presents most features of the traces in our database for long distance VoIP applications. Fig. 5 shows the PDF of the network delay. The network delay is half of the round-trip delay in the trace file, which was collected in January, 2009. We used *hrping* because of its useful features (timing the round trip delay in microseconds and sending out packet every x milliseconds). The data was collected from the Internet connection between McGill University (Canada) and Shanghai Jiao Tong University (China), which is inside the China Education Network, sending packet every 20ms for half an hour. The network loss is 1%, and min/avg/max of round-trip delays is $305ms$/$308ms$/$442ms$, and $153ms$/$154ms$/$221ms$ for one-way delays accordingly. The conversation used in our simulation is from the recording of a real dialog, which consists of only two conversation turns, in a "ask-response" pattern.

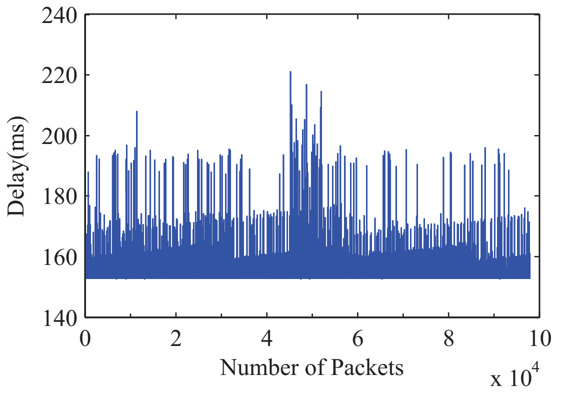


Fig. 4: Network Delay ($153ms$/$154ms$/$221ms$).

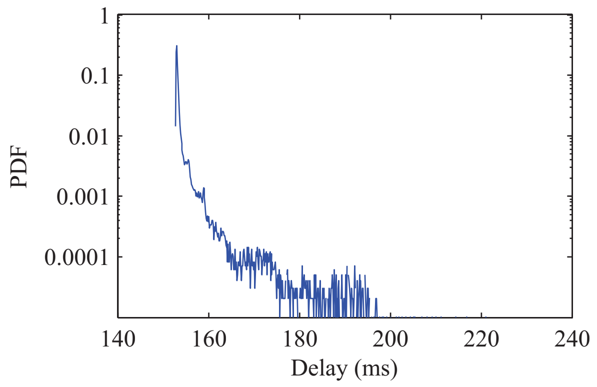
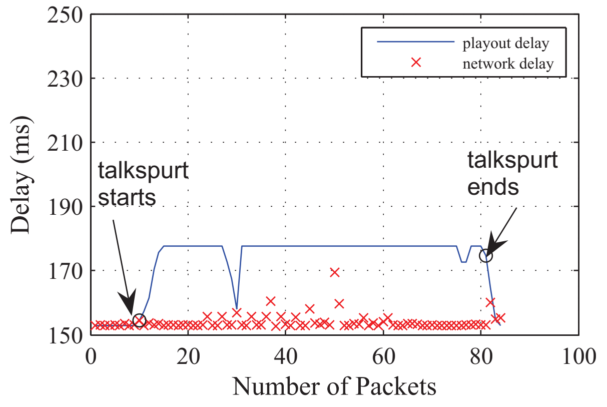


Fig. 5: PDF of the Network Delay.

Fig. 6 shows the variable jitter buffering for one conversation turn (the "asking" turn), which contains only one talk-spurt. The first packet of talk-spurt is stretched and played out immediately when it is received. The speech compression begins when "hangover" is detected. In our experiment, the 81-st packet is the first "hangover" packet. For the proposed algorithm, compression starts at the 80-th packet, and for the case of $d_{max} = 60ms$, compression starts at the 78-th packet, because the latter case has larger buffer and the 81-st packet is stored in buffer before the 78-th packet is played out. Although more packets are contracted, more buffer delay remains for the buffer with $d_{max} = 60ms$, which increases the conversational delay accordingly.

Table I shows the performance comparison of different jitter buffering algorithms: "Optimum" is the proposed algorithm; "VJM_$60ms$" is the algorithm proposed in [5] with $d_{max} = 60ms$; "VJM_$80ms$" is the algorithm proposed in [5] with $d_{max} = 80ms$; "Sun_opt" is the algorithm proposed in [8]. Even though the PLR is the highest, "Optimum" obtained the higher R factor (computed at the end of a conversation turn) than "VJM_$60ms$" and "VJM_$80ms$", because R factor is affected by not only the packet loss but also the mouth-to-ear delay. With the increase of d_{max}, the packet loss decreases because more packets with large network delay can be played out, and thus decrease the negative effect on the



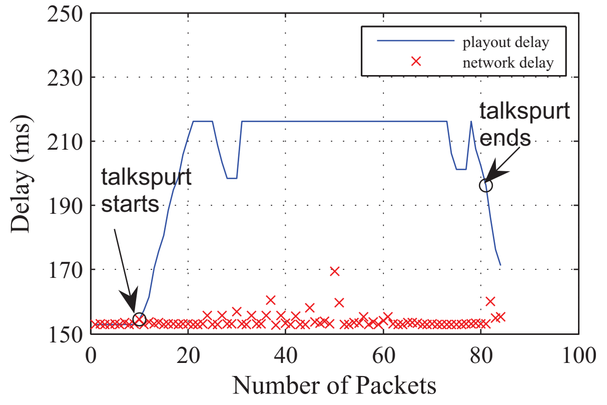


Fig. 6: Variable Jitter Protection: upper is the proposed buffering algorithm; bottom is the buffering algorithm in [5] with fixed depth of 60ms

R factor. However, the increment of d_{max} also increases the mouth-to-ear delay, and the negative effect on the R factor increases accordingly. In "VJM_60ms" and "VJM_80ms", the impairment caused by mouth-to-ear delay is stronger than that caused by packet loss, whereas "Optimum" is designed by the algorithm proposed in Section IV which concerns both impairments on the R factor. The results also proves that lower PLR does not guarantee a higher voice quality. Moreover, it can be seen that "Optimum" achieves the lowest conversational delay among all these algorithms. According to the criteria of calling quality for conversational VoIP in Section III, the proposed algorithm achieves the best calling quality.

TABLE I: Performance Comparison of Jitter Buffering Algorithms

Buffering algorithms	Condelay (ms)	R Factor	Loss (%)
Optimum	327.7	58.15	6.6
VJM_60ms	349.5	55.00	3.4
VJM_80ms	369.5	52.70	3.0
Sun_opt	358.5	58.15	6.5

VI. Conclusion

In conversational VoIP, customers care about the perceived calling quality the most. For some applications, low conversational delay is desired. In this paper, we introduce a new quality assessment including R factor and conversational delay. And our optimum playout scheme is developed based on optimizing this calling quality, that is, maximizing voice quality estimated by E-model and keeping a conversational delay as low as possible. Simulation results show that the proposed algorithm can obtain the best calling quality compared with other algorithms.

References

[1] Y. J. Liang, N. Farber, and B. Girod, "Adaptive Playout Scheduling using Time-Scale Modification in Packet Voice Communications," in *IEEE ICASSP '01*, Salt Lake City, USA, Jan. 2001, pp. 1445–1448.
[2] ITU, *Recomendation G.114: One-way Transmission Time*, ITU Std., May 2003.
[3] L.Atzori and M. L. Lobina, "Playout Buffering in IP Telephony: A Survey Discussing Problems and Approaches," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 3, pp. 36–46, 2006.
[4] ITU, *The E-Model, a computational model for use in transmission planning*, ITU Std., Mar. 2003.
[5] M. Lee, J. McGowan, and M. C. Recchione, "Enabling Wireless VoIP," *Bell Labs Technical Journal*, vol. 11, pp. 201–215, Nov. 2007.
[6] M. Narbutt and L. Murphy, "VoIP Playout Buffer Adjustment Using Adaptive Estimation of Network Delays," in *Proc. 18th International Teletraffic Congress (ITC-18)*, Berlin, Germany, 2003, pp. 1171–1180.
[7] K. Fujimoto, S. Ata, and Murata, "Adaptive Playout Buffer Algorithm for Enhancing Perceived Quality of Streaming Applications," *Telecommunication Systems*, vol. 25, no. 3-4, pp. 259–271, 2004.
[8] L. Sun and E. Ifeachor, "New Models for Perceived Voice Quality Prediction and their Applications in Playout Buffer Optimization for VoIP Networks," in *Proc. IEEE ICC*, Paris, France, Jun. 2004, pp. 1478–1483.
[9] M. Ghanassi and P. Kabal, "Optimizing Voice-over-IP Speech Quality Using Path Diversity," in *Proc. IEEE Workshop on Multimedia Signal Processing*, Victoria,BC,Canada, Oct. 2006, pp. 155–160.
[10] H. Schulzrinne, S. Casner, and V. Jacobson. (2003) RTP: A Transport Protocol for Real-Time Applications. [Online]. Available: http://www.networksorcery.com/enp/protocol/rfc/rfc3550.htm
[11] M. Yavuz, S. Diaz, R. Kapoor, M. Grob, P. Black, Y. Tokgoz, C. Lott, S. Guha, N. Daswani, and R.Jain, "VoIP over CDMA2000 1xEV-DO Revision A," *IEEE Communications Magazine*, vol. 44, pp. 88–95, Feb. 2006.
[12] ITU, *Recommendation G.729 Annex B: A Silence Compression Scheme for G.729 Optimized for Terminals Conforming to Recommendation V.70*, ITU Std., Jan. 2007.