Computer Networks 54 (2010) 3108–3122

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A Garch-based adaptive playout delay algorithm for VoIP

Ying Zhang [a,*], Damien Fay [b], Liam Kilmartin [a], Andrew W. Moore [b]

[a] Electrical & Electronic Engineering, College of Engineering and Informatics, National University of Ireland, Galway, Ireland
[b] Computer Laboratory, University of Cambridge, UK

ARTICLE INFO

Article history:
Received 10 August 2009
Received in revised form 9 April 2010
Accepted 3 June 2010
Available online 10 June 2010
Responsible Editor: N. Agoulmine

Keywords:
ARMA
Garch
Playout delay
Time series forecasting
VoIP

ABSTRACT

Network delay, packet loss and network delay variability (jitter) are important factors that impact on perceived voice quality in VoIP networks. An adaptive playout buffer is used in a VoIP terminal to overcome jitter. Such a buffer-control must operate a trade-off between the buffer-induced delay and any additional packet loss rate. In this paper, a Garch-based adaptive playout algorithm is proposed which is capable of operating in both inter-talkspurt and intra-talkspurt modes. The proposed new model is based on a Garch model approach; an ARMA model is used to model changes in the mean and the variance. In addition, a parameter estimation procedure is proposed, termed *Direct Garch* whose cost function is designed to implement a desired packet loss rate whilst minimising the probability of consecutive packet losses occurring. Simulations were carried out to evaluate the performance of the proposed algorithm using recorded VoIP traces. The main result is as follows; given a target Packet Loss Rate (PLR) the Direct Garch algorithm produces parameter estimates which result in a PLR closer than other algorithms. In addition, the proposed Direct Garch algorithm offers the best trade-off between additional buffering delay and Packet Loss Rate (PLR) compared with other traditional algorithms.

1. Introduction

Voice over IP (VoIP) technology [1] has become widely used amongst both business and consumer users due to its cost effectiveness, its support of multimedia technology and its ease of use. However, during packet transmission over the Internet, queuing and contention result in a varying network delay (jitter) experienced by the individual IP packets which form a VoIP flow. As a result, voice packets generated repeatedly at periodic time intervals at a source (typically only during actual speech talkspurts) will arrive at the receiver with different time intervals between the packets. Thus, if not compensated for, this effect would likely result in gaps in the audio waveform that would be played out to a listener. Typically, a smoothing buffer is used at a receiver to compensate for these variable delays. The received packets are first buffered (for a certain duration of time) prior to their playback in order to counteract the impact of the jitter. The influences of the delay variations within the network can be minimised by this additional buffering delay which is referred as to the playout delay. All the packets which arrive later than their playout delay time are regarded as lost packets and hence typically are not played out. Increasing the playout delay can reduce this packet loss rate but a longer playout delay has a negative impact on the quality and nature of the real-time communication. Thus, a trade-off exists between the effects of excessive playout delays and the packet loss rate due to inadequate playout delays. For interactive audio, a one-way delay of less than 400 ms [2] and packet loss rate less than 5% [2] are generally accepted as being required for conversational VoIP. However, a *one-way* delay of 150 ms [3] is considered as a more acceptable target figure for most VoIP applications.

In early VoIP systems, a fixed playout delay was commonly utilised as the solution to this problem. While this method offered an easily implemented solution, it was

* Corresponding author. Tel.: +353 91 524411.
E-mail address: y.zhang3@nuigalway.ie (Y. Zhang).

1389-1286/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2010.06.006

not an optimum solution since it did not take into account the fact that network jitter varies with time, as illustrated in Fig. 1.

Modern VoIP systems utilise adaptive playout delay approaches which estimate the network jitter continuously and dynamically adjust the playout delay either at the beginning of each talkspurt (known as *inter-talkspurt* playout delay adaptation) or continuously within each talkspurt (known as *intra-talkspurt* delay adaptation). Inter-talkspurt playout delay adaptation techniques adjust the playout buffer delay duration during the silence periods between speech talkspurts, and hence update the playout delay value at the beginning of each talkspurt. That playout delay is then utilised for all the packets within that talkspurt. Intra-talkspurt playout delay adaptation techniques are used in combination with speech waveform modification techniques to allow the playout delay to be adjusted within individual speech talkspurts. This is a potentially more advantageous approach in terms of responding to changes that may be occurring to the underlying network delay. However, such approaches do tend to be computationally expensive. The waveform modification techniques which are applied with an intra-talkspurt delay adaptation approach are needed in the process of expanding or compressing the speech waveform duration when a playout delay adjustment occurs. An example of such a technique was reported in Liang et al. [4] where a time scale modification technique (namely the Waveform Similarity Overlap-Add (WSOLA) algorithm [5]) was applied in combination with intra-talkspurt playout delay adaptation.

1.1. Inter-talkspurt delay adaptation algorithms

For the inter-talkspurt delay adaptation paradigm, the playout delay is adjusted during the silence period between each speech talkspurt (i.e. while no new speech packets are being received). The playout time for the first packet of the next talkspurt is obtained by delaying the playout of this packet after its arrival at the receiver by an amount of time equal to the playout delay, as indicated in (1). Once this decision is made, the playout time for all subsequent packets in that talkspurt have been effectively fixed, as given by (1):

$$\hat{p}_1^k = r_1^k + \hat{d}^k, \tag{1}$$

$$\hat{p}_i^k = \hat{p}_1^k + (i-1)\times\tau \quad \text{for } i \neq 1, \tag{2}$$

where $\hat{p}_1^k$ is the playout time for first packet in the kth talkspurt, which means the first packet in k talkspurt will be buffered $\hat{d}^k$ ms (predicted playout delay) after its arrival at time r_1^k; the other ith packet in kth talkspurt will be continuously played out at time $\hat{p}_i^k$, which can be directly predicted by the playout time of the first packet $\hat{p}_1^k$ and packet length τ. In this study, τ = 20 ms was used exclusively.

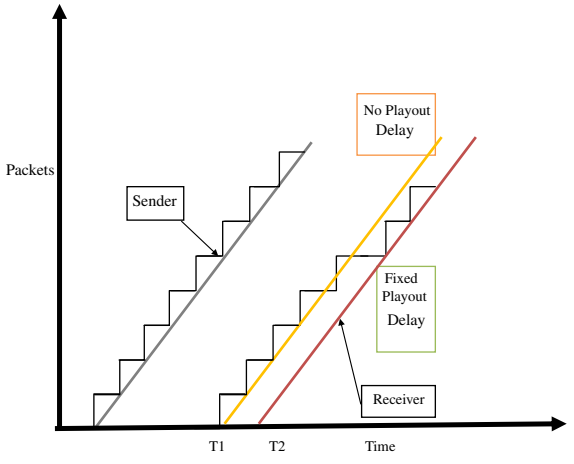

Fig. 1. VoIP packets over network (M = 2000).

1.2. Intra-talkspurt delay adaptation algorithms

The use of intra-talkspurt delay adaptation introduces a much more complex approach but with the potential benefit of superior performance. With such algorithms, the playout delay is regularly updated during each talkspurt (and not just once at the start of a talkspurt as is the case with inter-talkspurt delay adaptation algorithms). Hence, Eq. (1) above can be generalised for the intra-talkspurt case into the form of:

$$\hat{p}_i^k = r_i^k + \hat{d}_i^k, \tag{3}$$

where $\hat{d}_i^k$ is the playout delay (also called the additional buffering delay) at the arrival time of the ith packet of the kth talkspurt. The playout delay can be adapted as each packet arrives or this adaptation can be implemented in a batch mode after a number of packets have arrived or after some fixed time interval.

1.3. Packet loss concealment methodologies

All playout delay algorithms (including those described above) result in lost packets from time to time. A *Packet Loss Concealment* (PLC) stage is thus advantageous (after the playout delay stage) to improve the QoS. This attempts to maintain an adequate level of perceptual voice quality despite any residual packet loss. Packet Loss Concealment is most typically realised by some form of waveform modification involving the generation of replacement speech segments which are used to replace the speech waveform being conveyed within 'lost' (or 'late-arriving') packets. Typical waveform modification techniques include both insertion-based schemes and interpolation-based schemes [6]. With insertion-based schemes, the missing speech segment is replaced by inserting either silence\background noise or by repeating the last previously received packet (perhaps with some minor modifications). In interpolation-based schemes, a replacement waveform is generated by one of a number of different algorithms which capture the recent *characteristics* (e.g. frequency spectrum) of the speech signal, e.g. waveform substitution, pitch waveform replication and time scale modification. An interpolation-based scheme will achieve superior performance with respect to the perceptual quality of the resultant speech waveform but such algorithms are more complex and computationally costly to implement compared to the simpler insertion-based schemes.

1.4. Paper structure

This paper presents a new technique based on statistical modelling to implement a playout delay prediction algorithm for VoIP. Related work which provides a background to the problem and previous research in the literature which proposed solutions to the problem are presented in Section 2. The proposed ARMA/Garch model is introduced and described in detail in Section 3. In addition, a number of possible Packet Loss Concealment algorithms which can be used in partnership with the proposed adaptive playout delay algorithm are also introduced in Section 3. Section 4 of the paper presents the results with respect to three real-time traces. Finally, Section 5 of this paper presents the conclusions which have been drawn from this study.

2. Related work

In general, adaptive playout delay algorithms have traditionally been categorised in the literature into three classes:

(i) *Reactive algorithms* which continuously estimate the network delay and the delay variance in order to calculate playout deadlines for each packet without any consideration of Packet Loss Rate (PLR) control. One of the seminal works utilising reactive algorithms was by Ramjee et al. [7] which suggested the use of fixed weighting factors for network delay and network delay variance. An extension of this algorithm was proposed by Narbutt and Murphy [8] which examined the use of a methodology for dynamically selecting the weighting network delay coefficient. An additional approach was outlined in [9] which suggested adaptively calculating the network delay variance coefficient.

(ii) *Distribution-based algorithms* determine a playout delay by utilising both an estimation of the distribution of measured packet delays and a desired PLR. For example, the work outlined in [10] focused on estimating the tail of the network delay Probability Distribution Function (PDF); only packets which arrive after their playout time are of interest and these lie on the right tail of the PDF. Fujimoto et al. [10] assume a Pareto distribution for the tail. This approach was shown to deliver better results for playout delay determination compared with algorithms which used the complete network delay probability distribution. An alternative to the use of an a priori known distribution is the Concord algorithm as outlined in [11]. This algorithm uses a determined Packet Delay Distribution (PDD) histogram as the basis for estimating the required playout delay in order to achieve a certain desired PLR [11].

(iii) *Quality-based algorithms* estimate the playout delay by minimising a cost function which is based on some form of quality metric. The work presented in [12] is one of the early works suggesting this approach and it outlines a technique to estimate the playout delay by making use of the E-Model, which is a standard speech quality evaluation methodology outlined in the ITU-T standard G.107 [13]. Fujimoto et al. [14] also proposed a quality-based playout algorithm which selects the appropriate playout delay in order to maximise a perceptual quality metric [15]. More recently, the *Play-late* jitter buffer algorithm outlined in [6] was reported as providing impressive results by enhancing user-perceived speech quality by effectively removing any packet loss rate (resulting from the playout delay process only) by the insertion of periods of replacement packet portions, but at the cost of introducing potentially much longer playout delays.

The Linear Recursive Filter model (LRF) [7] is a very commonly used traditional inter-talkspurt approach. The end-to-end network delay is determined by consideration of both the end-to-end network delay estimation in recent past history and the current observation of the network delay. α is a fixed weighting factor, which determines the rate of convergence of the algorithm with $\alpha = 0.998$ being suggested with smooth network jitter while $\alpha = 0.75$ being chosen for burst network jitter. However, this algorithm does not offer the flexibility to be adaptive for different network conditions as it is based on a fixed weighting factor. However, it does offer a suitable algorithm to act as a comparison for our proposed technique. One of the first neural network-based inter-talkspurt approaches for the playout of voice frames in ATM networks was proposed by Tien and Yuang [16]. A Multi-layer Perceptron (MLP) was designed to predict the mean and variance of the network delay of the next packet at the beginning of every talkspurt and this algorithm is also used in this study for comparison purposes.

Arguably, the most commonly implemented intra-talkspurt approach is the Concord algorithm [11] which is based on a gradual ageing procedure to estimate the *Packet Delay Distribution* (PDD) curve. The end-to-end network delay, which includes one-way network delay and the playout buffer delay, is estimated according to a determined PDD curve and the desired PLR. By use of the inbuilt ageing algorithm, delay information from *older* packets has less impact than more recent measurements. As a result older information from the network delay histogram is gradually discarded or retired. The histogram method is used for the estimation of the *Packet Delay Distribution* (PDD) curve which is computed from the histogram bin.

3. Garch-based adaptive playout delay algorithm

The *General Autoregressive Conditional Heteroskedasticity* (Garch) model was first introduced by Engle originally as a model for financial time series forecasting [17]. Garch models explicitly target the *heteroskedacity* of time series (also known as volatility clustering) via a hierarchical model [18]. The model typically consists of an *AutoRegressive Moving Average* (ARMA) model for the mean of the time series and a separate ARMA model for the variance. In this paper, we propose a Garch model for playout delay

adaptation in VoIP network. As is shown in Fig. 2, jitter exhibits characteristics of self-similarity and burstiness. In a statistical sense, if a time series exhibits a *bursty* characteristic, this means that its variance is varying with time. Therefore, the Garch model, which is a classic solution for dealing with heteroskedasticity in a time series, can be considered as a suitable approach for modelling the network delay jitter.

3.1. ARMA/Garch model

The core objective in adaptive playout delay prediction is to determine an appropriate model for a jitter time series from a real VoIP network trace. In this paper, an ARMA(r,s)/Garch(p,q) model is proposed for playout delay prediction. The proposed algorithm can be summarised into the following steps:

(i) Select a suitable ARMA/Garch model structure.
(ii) Determine the ARMA/Garch model parameters using a cost function minimisation algorithm.
(iii) Estimate a playout delay setting by consideration of a suitable Probability Distribution Function (PDF) for the time series whose mean and variance have been estimated by step 2.

The initial stage in the process of implementing this modelling technique is to pre-process the jitter time series. According to Daniel et al. [19], a jitter time series is multi-structure in nature and consists of a short term non-stationary component and a long-term stationary component. The author suggested that a Laplacian probability distribution could be used to model the small time scale, non-stationary, process and that the large time scale stationary process could be modelled by additive Gaussian white noise.

In this paper, network jitter time series over short-time periods are considered to be non-stationary processes. Hence, an initial differencing operator is required by the modelling process to ensure that the resultant time series appears to be the result of a stationary process. It is often found that the first order ($d = 1$) differencing of non-seasonal data is adequate as in:

$$y_k = J_k - J_{k-1}, \tag{4}$$

where J_k is the jitter measurement of the kth packet in a VoIP packet flow. An ARMA(r,s) process can be used for describing the conditional mean of the jitter time series as:

$$y_k = a_0 + \sum_{i=1}^{r} a_i y_{k-1} + \sum_{j=1}^{s} b_j \varepsilon_{k-j} + \varepsilon_k, \tag{5}$$

where y_k is the conditional mean of the jitter difference at time k, ε_k are the error terms, which are generally assumed to be random variables, r is the order of the autoregressive part, s is the order of the moving average part, a_0 is the conditional mean constant, a_i are the conditional mean autoregressive coefficients, b_j are the conditional mean moving-average coefficients.

From (5), an ARMA(r,s) model can be used for the process of estimation of the conditional mean of the jitter time series, as implemented by:

$$\hat{y}_k = a_0 + \sum_{i=1}^{r} a_i y_{k-1} + \sum_{j=1}^{s} b_j \varepsilon_{k-j}, \tag{6}$$

where $\hat{y}_k$ is the forecasted mean of the jitter justifying difference:

$$\widehat{J}_k = J_{k-1} + \hat{y}_k, \tag{7}$$

where $\widehat{J}_k$ is the estimated mean of jitter from AR modelling.

A 1-step-ahead prediction-based AR model was shown to adequately model the conditional mean of a jitter time series in [7] and hence an ARMA(1,0) model will be subsequently used as the 'standard' model in this proposed algorithm. As a result of no consideration of 'moving average' part in this algorithm, Eq. (6) reduces to:

$$\hat{y}_k = a_0 + a_1 y_{k-1}. \tag{8}$$

The playout delay can then be expressed as:

$$d_k = \widehat{J}_k + e_k = J_{k-1} + \hat{y}_k + e_k. \tag{9}$$

Note the form of the expression above; the delay for the kth packet is composed of the jitter from the previous packet, J_{k-1}, plus the expected difference in the jitter $\hat{y}_k$ plus an additional term, e_k, which is the additional delay encountered by the packet. The aim is to set the buffering delay, ζ_w, to be greater than e_k except PLR% of the time. The additional delay is here assumed to be governed by a Laplacian distribution with zero mean. A Garch model is used to predict the conditional variance of the Laplacian distribution (which is then used to estimate the additional delay, i.e. ζ_w that results in the desired PLR value).

A general Garch(p,q) model for the estimation of the variance is given by:

$$\hat{\sigma}^2_{k+1} = \alpha_0 + \sum_{j=1}^{q} \alpha_j \varepsilon^2_{k+1-j} + \sum_{i=1}^{p} \beta_i \sigma^2_{k+1-i}, \tag{10}$$

where $\hat{\sigma}^2_k$ is the conditional variance forecast, σ^2_k is the conditional variance, p is the autoregressive lag, q is the moving average lag, α_0 is the conditional variance constant, α_j are the coefficients related to lagged residuals, β_i are the

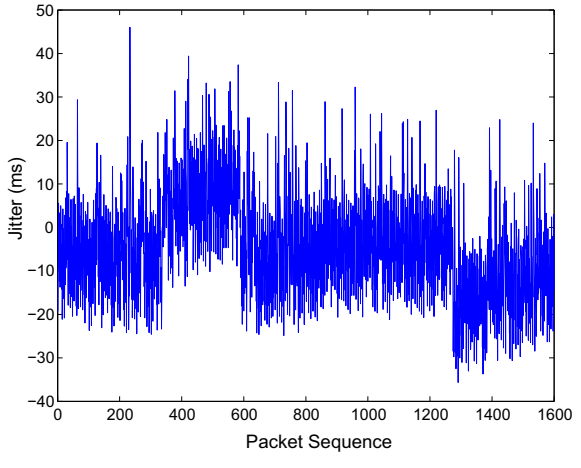

Fig. 2. Sample jitter plot (NUI Galway to University of Tokyo).

coefficients related to lagged conditional variances. The identification of a suitable Garch model order (i.e. p and q) is not an easy task either in theory or in practice. However, a common approach in many applications has been to use a Garch(1,1) model [20,21]. This approach has also been adopted in this work and as a result (10) reduces in complexity to

$$\hat{\sigma}^2_{k+1} = \alpha_0 + \alpha_1 \varepsilon^2_k + \beta_1 \sigma^2_k. \tag{11}$$

It is now necessary to utilise these two modelling techniques as a means of producing an overall model which is capable of providing a method for estimating a suitable playout delay value. Typically, *Maximum Likelihood Estimation* (MLE) is used to estimate the set of parameters [22]; $\{a_0, a_1, \alpha_0, \alpha_1, \beta_1\}$. With MLE, parameters are estimated such that they maximise the likelihood of having observed the data, as a function of the parameters. In this paper, the ARMA/Garch model with parameter estimation using MLE estimation will be termed the *Standard Garch method*.

In our work, the value of ζ_w is determined by utilising the desired PLR value, denoted χ, applied to the determined Laplacian PDF with zero mean and a separate forecast conditional variance estimated from the Garch model. In (12), we define w as the upper probability limit which equals a value of $1 - \chi$. This value w is determined by integrating the Cumulative Distribution Function (CDF) $P(e_k)$ as in (13). The upper limit of the integration ζ_w which satisfying the condition $P(e_k) = w$, can be calculated by using the inverse CDF as illustrated in Fig. 3. The value ζ_w is the exact forecast residual for controlling the playout delay with a specific desired χ. In this paper, χ was chosen to be in the range of 1–5% and hence the upper probability limit w would vary from 0.99 to 0.95, respectively,

$$w = 1 - \chi, \tag{12}$$

$$P[e_k \leqslant \zeta_w] = w, \quad 0.95 \leqslant w \leqslant 0.99. \tag{13}$$

A number of playout algorithms [9,11] have been proposed with are based on controlling the PLR due to 'late' packet arrival. We consider here a more direct approach

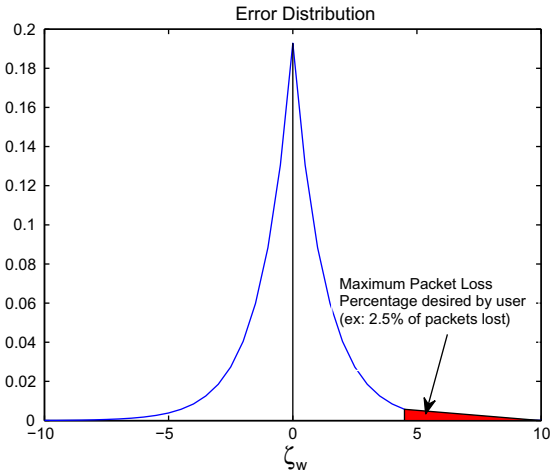

Fig. 3. Error in Laplace distribution.

for tuning the Garch model. As χ is the key quantity of interest, a cost function based on this is constructed in favour of the usual MLE criterion. This cost function has the added advantage that consecutive lost packets are also penalised. For a fixed packet loss rate, the impact on perceived speech quality of a certain number of randomly occurring lost packets is substantially less than the impact of losing a consecutive sequence of the same number of packets. Thus, the proposed Direct Garch model can also be designed to minimise the number of consecutive lost packets (*drops*). The associated cost function for such a Direct Garch model is given by

$$F = \left(\frac{1}{\chi} - \sum_k k\rho(k) \right)^2, \tag{14}$$

where $(1/\chi)$ is the desired number of packets between dropped packets resulting from a specific desired χ. The summation term in (14) represents the mean number of packets until the next packet drop due to a late-arriving packets for a given delay value. This can be estimated by applying a histogram analysis to the packet inter-arrival time of the training data. For a given playout delay value, the location of each resultant dropped packet in the training set can be determined and hence a histogram estimation of the frequency distribution $\rho(k)$ that there will be k packets until the next drop can be derived for a range of k values. The aim of using the cost function in (14) is to force this mean value of the frequency distribution to be equal to the reciprocal of the desired packet loss rate $(1/\chi)$. Since it is likely that the desired PLR will be less than 0.05 in practice, this cost function minimisation will also reduce the likelihood of very destructive (in terms of perceived speech quality) consecutive (i.e. $k = 1$) packet losses occurring. The AR parameters and unconditional variance do not impact on the packet drop inter-arrival distribution and hence they are not be updated by this process.

3.2. Algorithm operation for playout delay adaptation

The Standard Garch and Direct Garch have been applied in both inter-talkspurt and intra-talkspurt playout delay adaptation scenarios as detailed in the following sections.

3.2.1. Inter-talkspurt playout delay adaptation

The general equations that define the inter-talkspurt playout delay process were given by (1) and (2) in Section 1.1. The proposed ARMA/Garch models can generate a running 1-step-ahead prediction of the playout buffer delay for each packet. The playout delay, $\hat{d}^k$ for the kth talkspurt is set by consideration of the mean and standard deviation of the predicted playout delay of the last N packets as in:

$$\hat{d}^k = K \times \hat{\mu}_k + M \times \hat{\sigma}^2_k, \tag{15}$$

$$\hat{\mu}_k = \frac{1}{N} \sum_{j=T-N+1}^{T} \hat{d}_j, \tag{16}$$

$$\hat{\sigma}^2_k = \frac{1}{N} \sum_{j=T-N+1}^{T} (\hat{d}_j - \hat{\mu}_k)^2, \tag{17}$$

where j is the index of the last N packets, and T is the index of the last received packet. From [23], the mean duration of a talkspurt is assumed to be 352 ms and the mean duration of a silence period is assumed to be 650 ms in steady state. In this paper, an average of $N = 18$ packets is selected which corresponds to a duration of approximately 352 ms worth of speech for the typical case where each packet represents a duration of 20 ms of a speech waveform. The value of K in Eq. (15) is typically set to 1 whilst the value for M is varied to suit particular scenarios. The inter-talkspurt approach does not support the concept of accurately controlling the packet loss rate as it only adjusts the jitter buffer delay at the beginning of each talkspurt. The efficiency of the playout delay adaptation algorithm is mainly quantified in terms of the trade-off between the average additional buffering delay and the resultant *PLR*.

3.2.2. Intra-talkspurt playout delay adaptation

Intra-talkspurt techniques are advantageous in terms of their ability to adapt to substantial variations in the network delay which can occur during longer talkspurts. The proposed ARMA/Garch models continuously predict the playout buffer delay for each packet. However, at any instance where the playout delay is altered (whether increased or decreased) it is necessary to modify the speech waveform. In the case where the playout delay is increased, there is a time period (equal to the increase in the playout delay) where some surrogate waveform must be *inserted*. In the case where the playout delay is reduced, there is a need to *remove* some duration of the speech waveform (equal in length to the reduction in the playout delay). In order to play out the packets consecutively according to the different predicted buffer delays for each packet, a number of such time scale modification techniques [4,6] have been proposed. In this paper, a simple insertion-based waveform technique which has been termed *Repeat and Truncate* has been adopted. The basic idea of this algorithm is to play out the packets at the predicted playout time by repeating (either wholly or partially) the waveform of the current packet. With this algorithm, an increase in the playout delay results in a repetition of a *part* of the *last packet* which was received, whereas a reduction in the playout delay results in a *truncation* of the speech contained in the *next* packet. It should be emphasised that the focus of this research is on the adaptation algorithm rather than the waveform modification technique and hence it was felt that the use of such a basic algorithm offered an adequate compromise between perceptual performance and algorithm complexity.

3.3. Play-late algorithm for packet loss concealment

The issue of concealing the impact of any residual packet loss (due to *late* packet arrival) needs to be addressed using some form of Packet Loss Concealment algorithm. In this paper, an adaptation of the packet concealment algorithm (the Play-late algorithm) has been used in conjunction with the proposed Garch-based playout delay prediction models. In traditional playout delay processes, any packets which arrive later than their playout time are regarded as *lost* packet and hence are not played out. The operation of the proposed *Play-late* algorithm is to play out the segment of the late-arriving packet that is still on time. For example, if a packet in length of 20 ms arrives 5 ms late, the last 15 ms of the packet will be played out, and the first 5 ms of the packet being discarded.

4. Results

In this section, a performance analysis of the proposed algorithms and a comparison with the performance of some of the standard techniques in the field is presented. There is no single metric which provides a definitive guide as to which is the optimally performing playout delay algorithm in a study. The performance of playout delay algorithms are not evaluated in a manner to determine a single optimum performance metric rather their performance is evaluated from the perspective of the behaviour of the algorithm in terms of the trade-off between the packet loss rate resulting from the algorithm and the additional playout delay introduced by the algorithm. Alternatively from a perceptual perspective, a PESQ\MOS-based analysis of the speech waveforms produced at the output of the various playout delay algorithms offers an alternatively metric to packet loss rate with which to examine this trade-off with the additional playout delay. A final methodology for evaluating the performance of a playout delay algorithm is to examine the actual distribution of packet losses due to the buffering algorithm with the desired PLR in order to examine the ability of the algorithm to minimise the impact of consecutive packet losses. The results presented in this section provide a comparison of the performance of the evaluated algorithms using a variety of these evaluation methodologies.

4.1. Evaluation methodology

The proposed ARMA/Garch models are applied to both inter-talkspurt playout delay adaptation and intra-talkspurt playout delay adaptation using a simulated network environment whose delay characteristics were based on real VoIP traces. The application used in this paper first encodes the audio stream using G.729B [24] codec into 20 ms packets of length 80 bytes. The Realtime Transport Protocol (RTP) is then used to sequence the packets and these packets are then encapsulated into a UDP packet for transmission across the internet. Since it was not feasible to take traces using terminals whose clocks were accurately synchronised, only information concerning inter-packet arrival times was available for these traces. These VoIP traces were gathered using an adapted version of PJSIP [25], an open source VoIP application, and the duration of the traces ranged from between 6 and 22 h. These traces consisted of continuous full duplex transmission of 20 ms speech packets between NUI, Galway, Ireland and University of Tokyo (a sample of which is shown in Fig. 2), University of New South Wales, Sydney, Australia and Chengdu, People's Republic of China as shown in Table 1. For each of these traces, the jitter information was recorded for the full duration of the connection and this jitter informa-

Table 1
Trace details.

Trace no.	Internet path	Trace time (GMT)	Length (h:min)
1	NUIG ⇄ UT, Japan	21/05/2007 17:39	06:49
2	NUIG ⇄ UNSW, Australia	23/05/2007 07:32	10:15
3	NUIG ⇄ Chengdu, China	23/05/2007 12:32	21:36

tion was then subsequently used in a simulated VoIP network model. Trace 1 from NUI, Galway to University of Tokyo, Japan displayed jitter values in the region of 30 ms. Trace 2 from NUI, Galway to University of New South Wales, Sydney, Australia typically showed a jitter of less than 30 ms. Trace 3 from NUI, Galway to Chengdu, China contained a typical jitter value of 25 ms. Jitter traces from all of these recording displayed self-similarity and burstiness within the record duration. The ARMA/Garch playout delay prediction model has been applied to all of these traces for experiments during the algorithm development. In this paper, all of the collected traces have been used to quantify the performance of the proposed and traditional models.

The evaluation of algorithm performance was quantified using some traditional metrics such as additional playout delay introduced and *late arrival* packet loss rate. However, a more informative performance evaluation has been achieved through the use of a perceptual speech quality-based metric, Perceptual Evaluation of Speech Quality (PESQ) [15] using an evaluation methodology similar to that utilised in [26].

4.2. Prediction of jitter time series

Initial experiments focused on establishing and quantifying the ability of the various algorithms to predict the jitter time series. The *Prediction Sum Squared Error* (PSSE) was used to evaluate the prediction accuracy and Fig. 4 shows the relative performance of each of the algorithms when predicting the jitter in a recursive manner. As would be expected the PSSE increases as the predicted jitter values are fed back for longer prediction runs but the Garch models are shown to exhibit superior performance. This highlights the suitability of both the Garch and Concord algorithms in particular for operation in an intra-talkspurt adaptation mode.

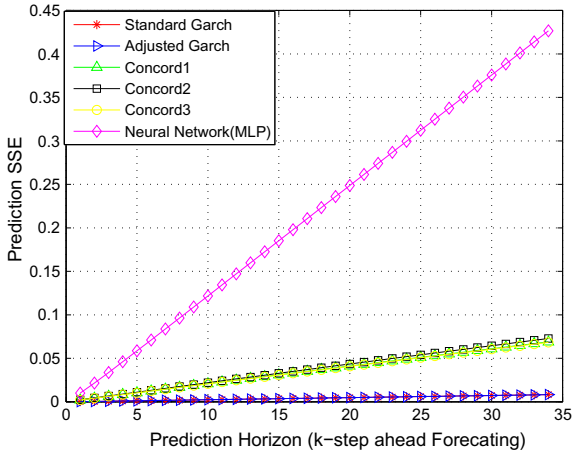

Fig. 4. Prediction accuracy with forecast horizon.

Fig. 5 illustrates the operation and relative performance of the Direct Garch and Concord 3 algorithms during a very long (artificial) talkspurt for the three traces, respectively. From this graph it can be seen the Direct Garch model is capable of capturing the traffic burstiness whilst the Concord algorithm offers a comparatively smooth prediction as a result of being based on a long history of network delay information. These results are promising in terms of proving the basic abilities of the proposed Garch-based algorithms but a more comprehensive set of evaluation tests for both inter-talkspurt and intra-talkspurt adaptation modes need to be completed.

4.3. Inter-talkspurt delay adaptation algorithm performance

The performance of an inter-talkspurt delay adaptation algorithm is typically quantified in terms of a trade-off. This trade-off is between the average additional delay (introduced by the buffering process) and the 'late arrival' packet loss rates (resulting from the algorithm).

4.4. Additional buffering delay and packet loss rate

In Fig. 6, the performance of the proposed Standard and Direct ARMA/Garch models have been compared to that of a traditional Linear Recursive Filter model [7] and the MLP-based neural network model [16].

The general goal of most play delay algorithms is to get an optimal trade-off between packet loss rate and the playout delay time. That is, minimising the packet loss rate and playout delay simultaneously is the expectation.The results show that both Garch-based models and the MLP-based model achieve very similar performance in terms of the trade-off between packet loss rate and additional buffering delay. As it is shown in this figure, with the same playout delay time, there will be more lost packets by the traditional LRF algorithm. The diagram illustrates the clear superiority of these non-linear algorithms to that offered by the more traditional LRF algorithm.

4.5. PESQ MOS-based algorithm evaluation

Mean Opinion Score (MOS) determined by the PESQ algorithm, which is referred to as PESQ MOS, is calculated by the simulated wav files according to the packet loss results of the four algorithms as in Fig. 7. As the locations of lost packets in the simulated wav files are random in this experiment, 10 simulated wav files were generated for each PLR and the average of the MOS scores are calculated to reduce the impact of packet loss location on the perceived speech quality. In Fig. 7, the PESQ MOS, is plotted against the additional buffering delay that would be introduced by each of the four algorithms under investigation. For the same playout delay, the traditional LRF shows a comparatively poor performance in terms of the perceived speech quality. For the purpose of applying the PESQ algo-

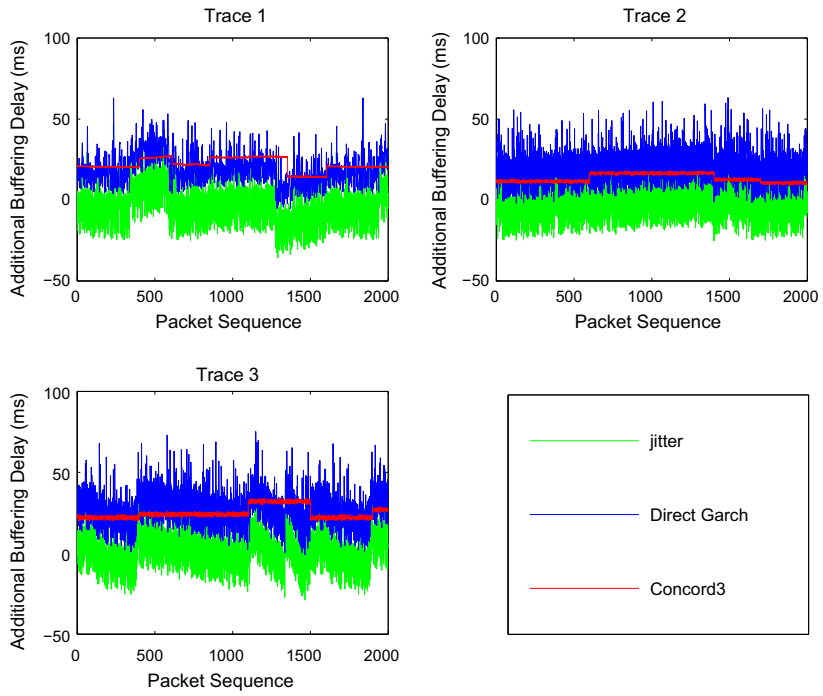

Fig. 5. Playout delay adaptation within a talkspurt.

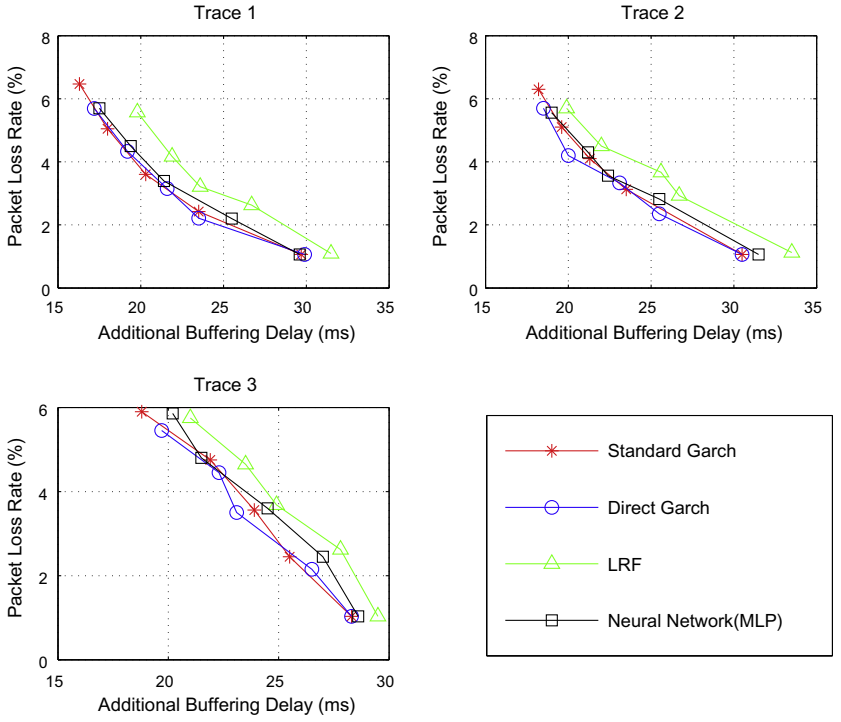

Fig. 6. Packet loss rate versus playout delay for inter-talkspurt adaptation.

rithm, any *late arrival* lost packets were replaced using a standard packet repetition noise substitution concealment methodology. The results show that, as expected, the three non-linear algorithms deliver very similar performances, which is noticeably better than that of the LRF algorithm for any specific additional buffering delay value.

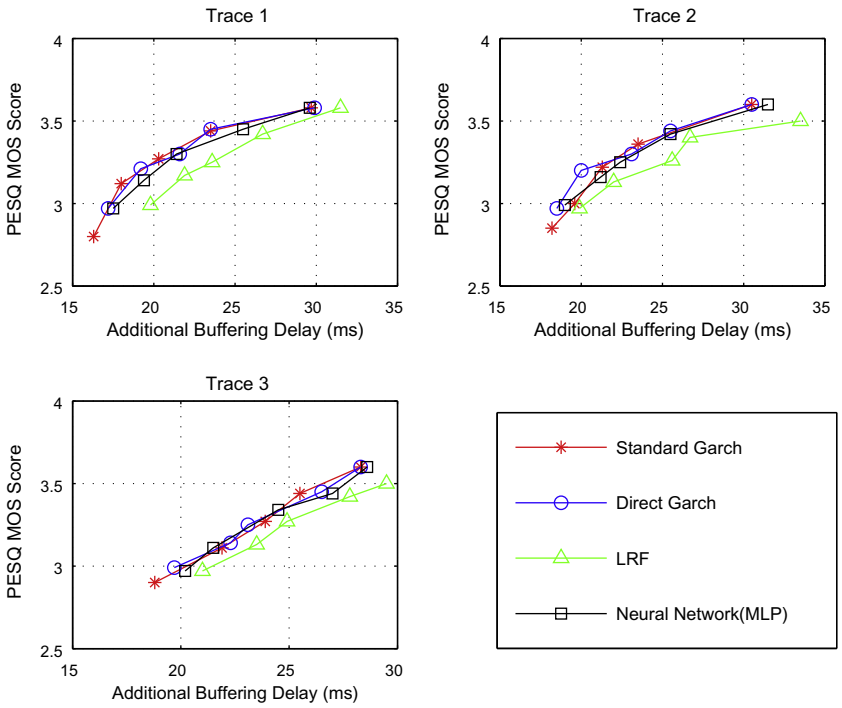

Fig. 7. PESQ MOS versus additional buffering for inter-talkspurt delay adaptation.

The impact of the inclusion of the '*Play-late*' algorithm for packet loss concealment was then evaluated utilising the PESQ MOS metric. Fig. 8 shows the impact on perceptual quality which incorporation of the *Play-late* packet loss concealment algorithm has on the performance of the various algorithms. The *x*-axis represents the original

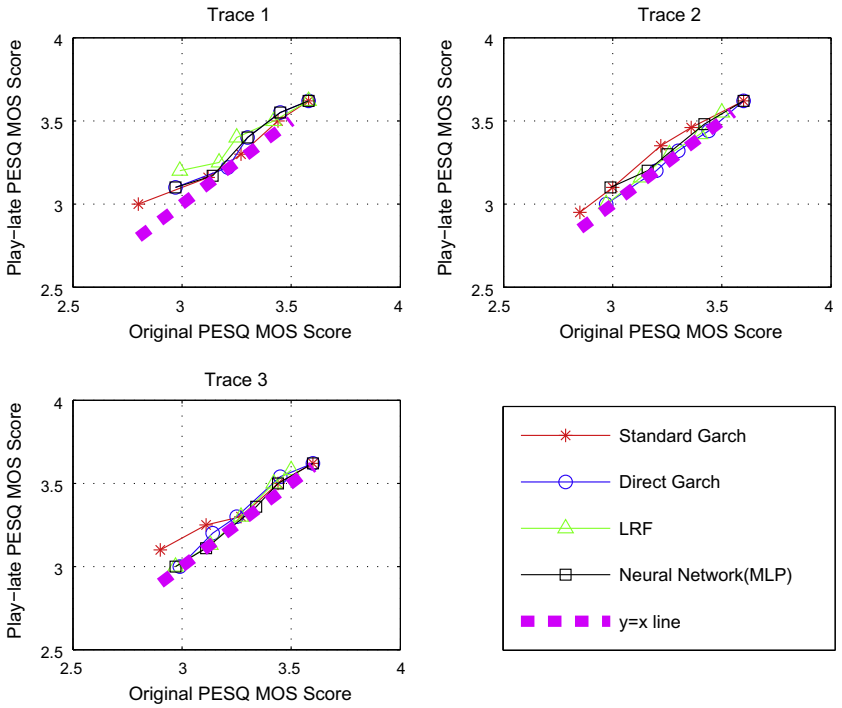

Fig. 8. PESQ MOS performance in inter-talkspurt mode after inclusion of Play-late algorithm.

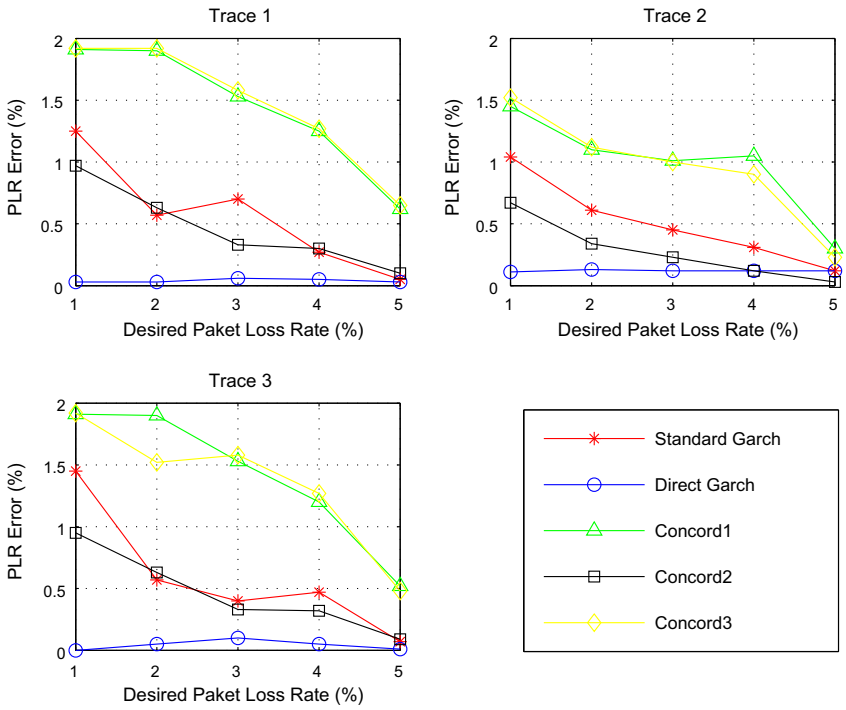

Fig. 9. Analysis of algorithms' ability to achieve a target PLR.

PESQ MOS score as in Fig. 7, and the *y*-axis represents the *Play-late* PESQ MOS score. The pink[1] dash is the line $y = x$. As it is shown in the Fig. 8, there is a slight improvement for any given packet loss rate with each playout delay algorithms.

4.6. Intra-talkspurt delay adaptation algorithm performance

When operating in an intra-talkspurt adaptation mode, the performance of Garch models were compared with those of the Concord algorithms in terms of their ability to implement a specified 'desired' packet loss rate, their ability to minimise the occurrences of consecutive packet losses, the offered trade-off between additional buffering delay and packet loss rate and a PESQ MOS-based evaluation. The performance of the proposed Standard and Direct ARMA/Garch models are compared with the standard Concord algorithms [11].

Fig. 9 presents a comparison of the performance of the Garch models versus that offered by three Concord-based variants in terms of their ability to implement a 'target' packet loss rate.

This figure shows the variation of absolute error between the 'target' and 'actual' measured packet loss rates for each algorithm for a variety of different 'target' loss rates. It is clear from this graph that the Direct Garch model offers a very stable PLR Error which is close to zero and that it outperforms the other four algorithms in this criterion. *This is the main advantage of the proposed algorithm.*

A comparison of the algorithms' performance is also possible using the consecutive packet loss rate as a metric. This reflects the probability of the algorithm resulting in two or more consecutive packets being lost due to their 'late arrival'. Whilst the percentage of packets for which this occurs is quite small the impact of such events on perceived speech quality can be severe. Figs. 10 and 11 illustrate the performance of the algorithms when the consecutive packet loss rate is evaluated at different 'target' packet loss rates and at various additional playout buffer delay values, respectively. It is clear from both of these graphs that the Garch models offer a much reduced probability of such scenarios occurring compared to the Concord algorithm variants. In particular, the Direct Garch model always achieves the best performance of all the evaluated techniques which would be expected due to the inclusion of these criteria in the cost function on which this algorithm variant is based.

Fig. 12 offers a summary comparison of the relative performance of the five algorithms under investigation in terms of the trade-off between measured packet loss rate and additional playout delay introduced.

The performance of the Garch models is shown to be superior to that of the Concord algorithms in this comparison with the Direct Garch model marginally offering the best performance.

Finally, the performance of the algorithms was also compared using the perceptually motivated PESQ MOS evaluation criteria. The result of this comparison is shown in Fig. 13 and again this illustrates the superior performance offered by the Direct Garch model.

In Fig. 14, the *x*-axis represents the original PESQ MOS score as in Fig. 13, and the *y*-axis represents the 'Play-late'

[1] For interpretation of color in Fig. 8, the reader is referred to the web version of this article.

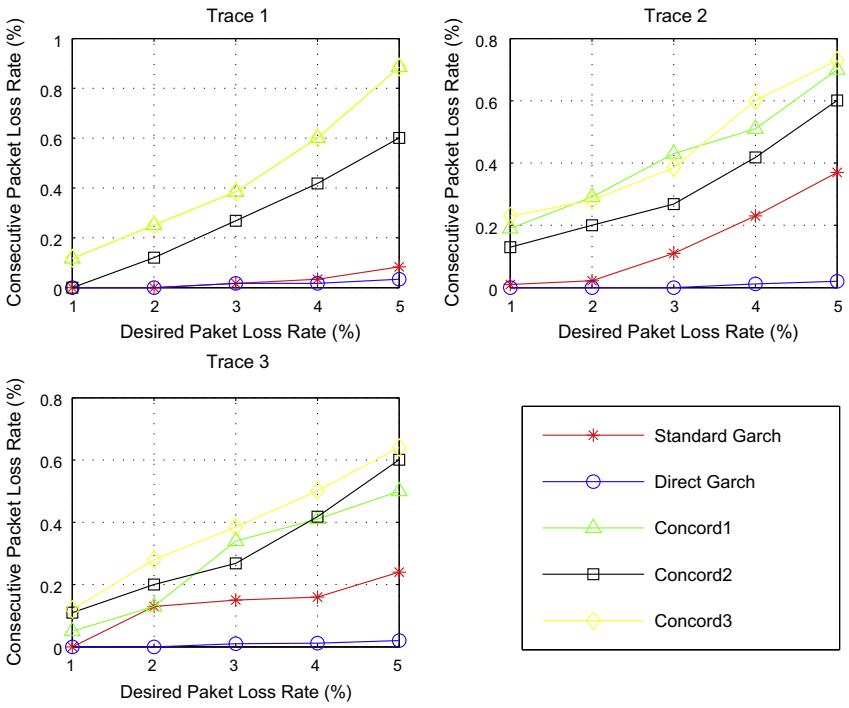

Fig. 10. The consecutive PLR with desired PLR.

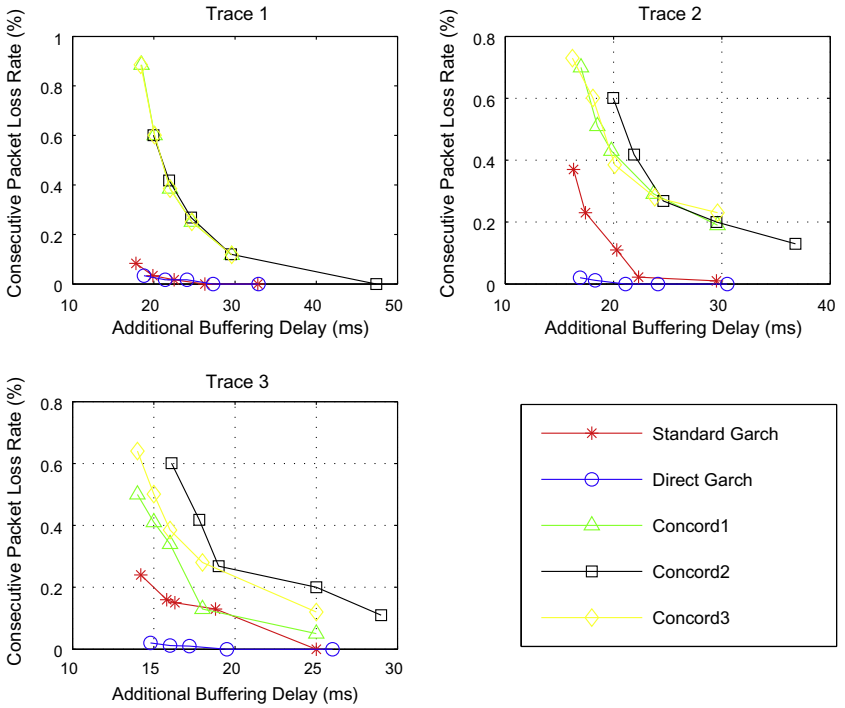

Fig. 11. The consecutive PLR with additional buffering delay in intra-talkspurt adaptation.

PESQ MOS score. The pink dash is the line $y = x$. Fig. 14 shows the impact on perceptual quality evaluation of using the same set of algorithms but this time with the 'Play-late' packet loss concealment algorithm being incorporated. Compared with the original PESQ MOS scores, there is an obvious improvement for each algorithm but this improvement is close to 0.2 MOS for each algorithm. This improvement compared to the results in Fig. 14 is far more

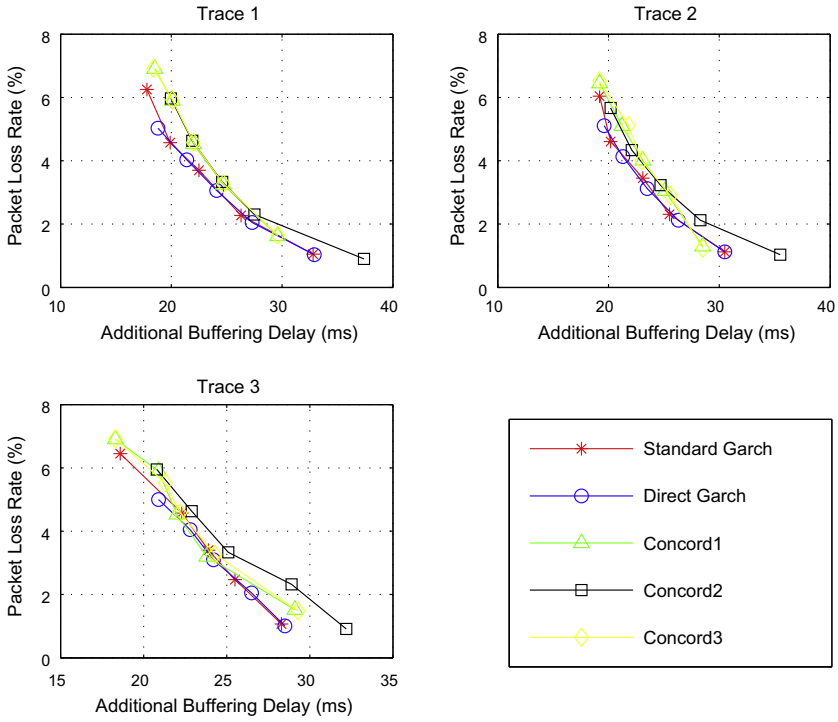

Fig. 12. Packet loss rate versus playout delay for intra-talkspurt delay adaptation.

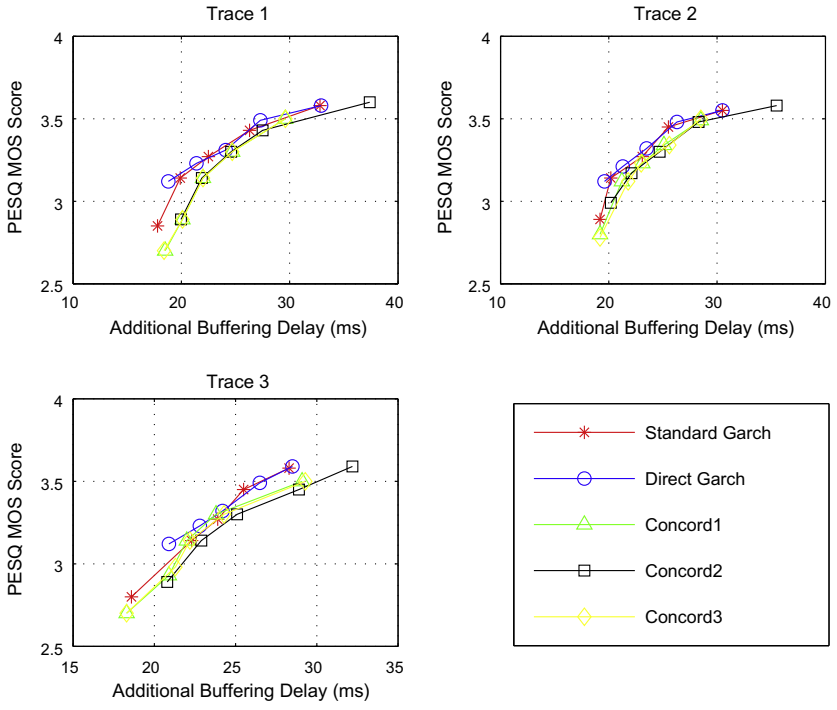

Fig. 13. PESQ MOS versus additional buffering delay for intra-talkspurt delay adaptation.

significant compared to the improvement highlighted for the inter-talkspurt case (i.e. Fig. 8). This is likely due to the fact that when operating in an intra-talkspurt adaptation mode, the 'Play-late' algorithm is able to implement

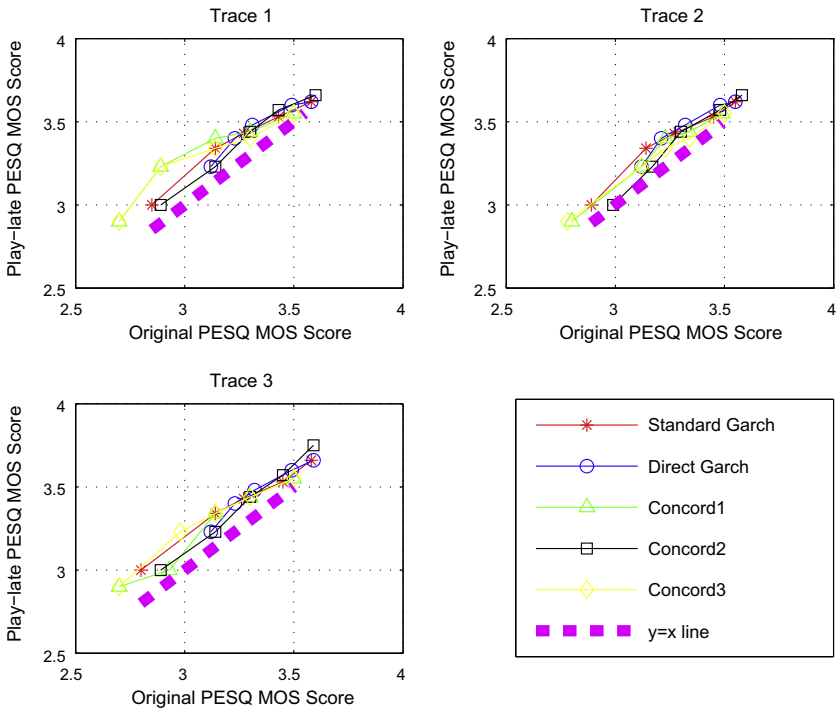

Fig. 14. PESQ MOS performance in intra-talkspurt mode after inclusion of Play-late algorithm.

a complete 'late' packet playout, whereas when operating in an inter-talkspurt mode, it is more likely that only a partial 'late' packet playout will occur thus causing more perceived distortion levels.

5. Conclusions

This paper proposed a new adaptive ARMA/Garch-based algorithm to address the issue of implementing a jitter buffer delay estimator in a receiving VoIP terminal. The proposed techniques are capable of operating in either inter-talkspurt or intra-talkspurt delay adaptation modes. Experiments were carried out to evaluate the performance of the proposed algorithms in both modes of operation and to offer a comparison of their performance with other commonly used techniques.

The proposed Standard and Direct ARMA/Garch models have been compared with the traditional LRF model and the neural network-based MLP model when operating in the inter-talkspurt delay adaptation mode. The results of the experiments presented, which were based on real VoIP delay traces, showed that the Standard and Direct ARMA/Garch models and the MLP model all achieve very similar performance with respect to the trade-off between packet loss rate and additional playout delay. However, any MLP-based algorithm will require the structure (i.e. number of nodes, weight values, number of layers, etc.) to be updated at regular intervals (using inter-packet delay training data from recently received packets). The training process for MLPs (even when using fast back propagation algorithms) is computationally very demanding as is the process of cross validation which is required to validate the generalisation capabilities of a trained network. Hence, comparatively the proposed ARMA/Garch models are much simpler to implement while offering very good performance.

When operating in the intra-talkspurt playout delay adaptation mode, both Concord and ARMA/Garch models are computationally simple to implement. The implementation of the Concord algorithm requires the maintenance of a histogram estimation of the probability distribution of inter-packet arrival times. This histogram model requires updating and the application of a relatively complex data ageing process (on the previously determined data) when updating the model with new inter-packet arrival time data points. The Direct ARMA/Garch algorithm achieves the best performance in terms of matching a desired packet loss rate, consecutive packet loss control and also achieving the best trade-off between packet loss rate and additional playout delay compared with the Concord algorithm. The standard Garch algorithm also achieves an improved performance in the trade-off between packet loss rate and additional buffering delay but it offers no superiority in terms of matching the desired packet loss rate or reducing the probability of consecutive packet losses occurring. In addition, the proposed Packet Loss Concealment scheme, Play-late, can partially or totally recover lost packets waveform information at the receiver. The results of additional experiments show that the additional inclusion of this algorithms results in an improvement in the perceptual quality of the received speech of between 0.1 and 0.2 MOS.

Acknowledgments

This research was supported under NUI, Galways Millennium Research Fund and under its College of Engineering and Informatics Postgraduate Fellowship programme. This work was started while Ying Zhang was a visitor with the University of Cambridge Computer Laboratory.

The authors would like to express their thanks to Mr. Shane Butler of NUI, Maynooth, Dr. Tim Moors from the University of New South Wales, Sydney, Australia, Mr. Yingfei Xiong of University of Tokyo, Japan and Mr. Yaoxing Wang, Huawei Technologies Co. Ltd., China for their help in obtaining the network delay traces which were used in the study. The authors would also like to express their appreciation to the NetOS group within the Computer Laboratory at the University of Cambridge for their support of this research.

References

[1] G. Thomsen, Y. Jani, Internet telephony: going like crazy, IEEE Spectrum (2000) 52–58.
[2] P. Zhu, C. Wilson, Effects of packet loss on waveform coded speech, in: Proceedings of the Fifth International Conference on Computer Communications, Atlanta, GA, 1980, pp. 275–280.
[3] ITU-T, ITU ITU-T Recommendation G.114, 2003.
[4] Y.J. Liang, N. Färber, B. Girod, Adaptive playout scheduling and loss concealment for voice communication over IP networks, IEEE Transactions on Multimedia 5 (2002) 532–543.
[5] W. Verhelst, M. Roelands, An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech, in: IEEE International Conference on Acoustics, Speech, Signal Processing, Minneapolis, MN, 1993, pp. 554–557.
[6] C. Perkins, O. Hodson, V. Hardman, A survey of packet loss recovery techniques for streaming audio, IEEE Network 12 (5) (1998) 40–48.
[7] R. Ramjee, J. Kurose, D. Towsley, H. Schulzrinne, Adaptive playout mechanisms for packetized audio applications in wide-area networks, in: INFOCOM '94, Networking for Global Communications, 13th Proceedings IEEE, 1994, pp. 680–688.
[8] M. Narbutt, L. Murphy, A new VoIP adaptive playout algorithm, in: Telecommunications Quality of Services: The Business of Success, QoS 2004, IEE, 2004, pp. 99–103.
[9] Y. Jung, W.J. Atwood, Beta-adaptive playout scheme for voice over IP applications (internet), IEICE Transactions on Communications 88 (5) (2005) 2189–2192.
[10] K. Fujimoto, S. Ata, M. Murata, Playout control for streaming applications by statistical delay analysis, in: Proceedings of IEEE International Conference on Communications (ICC), 2001, pp. 2337–2342.
[11] C. Sreenan, J.-C. Chen, P. Agrawal, B. Narendran, Delay reduction techniques for playout buffering, IEEE Transactions on Multimedia 2 (2000) 88–100.
[12] L. Sun, E. Ifeachor, Prediction of perceived conversational speech quality and effects of playout buffer algorithms, in: IEEE International Conference on Communications (ICC), vol. 1, 2003, pp. 1–6.
[13] ITU-T, The E-Model, A computational model for use in transmission planning, 1998.
[14] K. Fujimoto, S. Ata, M. Murata, Adaptive playout buffer algorithm for enhancing perceived quality of streaming applications, in: Global Telecommunications Conference, GLOBECOM '02, IEEE, vol. 3, 2002, pp. 2451–2457.
[15] ITU-T, Perceptual Evaluation of Speech Quality (PESQ), An objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs, 2001.
[16] P. Tien, M. Yuang, Intelligent voice smoother for silence-suppressed voice over internet, IEEE Journal on Selected Areas in Communications 17 (1) (1999) 29–41.
[17] R.F. Engle, Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation, Econometrica 50 (4) (1982) 987–1007.
[18] T. Mikosch, Modeling financial time series, in: New Directions in Time Series Analysis, Centre International de Rencontres Mathematiques, Luminy, France, 2001.
[19] E. Daniel, C. White, K. Teague, An inter-arrival delay jitter model using multi-structure network delay characteristics for packet networks, in: Proceedings of the 37th Asilomar Conference on Signals, Systems, and Computers, Asilomar, CA, 2003, pp. 1738–1742.
[20] R. Garcia, J. Contreras, M. van Akkeren, J. Garcia, A garch forecasting model to predict day-ahead electricity prices, IEEE Transactions on Power Systems 20 (2) (2005) 867–874.
[21] L. Gazola, C. Fernandes, A. Pizzinga, R. Riera, The log-periodic-ar(1)-garch(1,1) model for financial crashes, European Physical Journal B 61 (3) (2008) 355–362.
[22] S. Ling, M. McAleer, The log-periodic-ar(1)-garch(1,1) model for financial crashes, Econometric Theory 19 (2) (2003) 280–310.
[23] K. Sriram, W. Whitt, Characterizing superposition arrival processes in packet multiplexers for voice and data, IEEE Journal on Selected Areas in Communications 4 (6) (1986) 833–846.
[24] ITU-T, A silence compression scheme for g.729 optimized for terminals conforming to recommendation v.70.
[25] PJSIP homepage. <http://www.pjsip.org>.
[26] M. Ranganathan, L. Kilmartin, Neural and fuzzy computation techniques for playout delay adaptation in VoIP networks, IEEE Transactions on Neural Networks 16 (5) (2005) 1174–1194.

Ying Zhang finished her B.E. in Chongqing University of Posts and Telecommunications, China (2005), and M.E. in London South Bank University, UK (2006). Currently, she is currently a Ph.D. student in National University of Ireland, Galway. Her research interests include time series analysis of network structures.

Damien Fay obtained a B.E. from University College Dublin (1995), an M.E. (1997) and Ph.D. (2003) from Dublin City University and worked as a mathematics lecturer at the National University of Ireland (2003–2007) before joining the NetOS group, Computer Laboratory, Cambridge in 2007 as a research associate. He is currently a research associate at Cambridge. His research interests include applied graph theory, time series analysis and social network analysis.

Liam Kilmartin received the B.E. and M.E. degrees in electronic engineering from University College Galway, Galway, Ireland, in 1990 and 1994, respectively. He has been a lecturer in the Department of Electronic Engineering, National University of Ireland, Galway, since 1994. His current research interests include advanced communication networks, mobile networking technologies and the application of speech processing and neural network techniques in communication networks.

Andrew W. Moore is a lecturer at the University of Cambridge, Computer Laboratory. His interests lie in addressing the scalability, usability, and reliability of the internet. He completed his Ph.D. with the Cambridge University Computer Laboratory in 2001 and prior to that took a Masters degree and an honours degree from Monash University in Melbourne. Australia. He is a chartered engineer with the IET and a member of the IEEE, ACM and USENIX.