An application of neural networks to adaptive playout delay in VoIP

Ying Zhang ¹, Damien Fay ¹, Liam Kilmartin ²

¹ Department of Mathematics, National University of Ireland, Galway
damiен.fay@nuigalway.ie

² Department of Electronic Engineering, National University of Ireland, Galway
liam.kilmartin@nuigalway.ie

Abstract
IP networks are not designed for real-time voice data. The statistical nature of data traffic and the dynamic routing techniques employed in IP networks results in a varying network delay (jitter) experienced by IP packets. As a result voice packets generated at successive and periodic intervals at the source will typically be buffered at the receiver prior to playback in order to smooth out the jitter. However, the additional delay introduced by the payout buffer degrades the quality of service. Thus forecasting the jitter is an integral part of selecting an appropriate buffer size.

This paper compares several neural network based models for adaptive playout buffer selection. Specifically, a combined wavelet transform/neural network approach is proposed. The effectiveness of these algorithms is evaluated using recorded traces from Galway to Tokyo by comparing the buffering delay and the packet loss ratios of each technique. Simulation results indicate that the Haar-Wavelets-Packet MLP adaptive scheduling scheme is superior. Finally, the voice signal is reconstructed according to the packet loss information from estimation of each algorithm, and the quality of the voice is then given a score according to the PESQ MOS algorithm.

Keywords: VoIP, playout delay, neural networks, forecasting.

1. Introduction
In recent years Voice Over IP (VoIP) has seen a huge increase in use due to its cost effectiveness, support of multimedia technology and ease of use. However, the network delay and packet loss, which are ubiquitous nature of the Internet due to the Best-effort mechanism, are the main factors effecting the Quality of Service (QoS) of a VoIP call[1].

When audio packets are transmitted over the internet, the variable network delay (jitter), which is mainly due to the variable queuing time in routers, modifies the periodic form of the transmitted audio packets in receiver [1]. That is to say, a sequence of packets sent at consecutive and periodic intervals will arrive at the receiver at irregular intervals as it is shown in Figure 1. The playout delay is an
application to reduce the network delay variability by buffering the received packets and playing them out after a certain time. The packets arrived later than the playout delay time are regarded as ‘lost packets’, are not played out. Increasing the playout delay can reduce the packet loss, but a long playout delay has a negative impact on the real-time communication quality. Thus a trade-off exists between delay and packet loss. For interactive audio, a packet delay up to 400ms [2] and packet loss rate less than 5% are considered adequate [3].

[4] advocated a fixed playout delay as an initial solution to this problem. As it is also shown in Figure 1, with this method, there is a fixed buffer size for each arrived packets to play out. This method is easier to implement, however, it is inefficient as it does not take into account that network jitter varies with time. Alternatively, an adaptive playout delay approach, estimates the network jitter continuously and dynamically adjusts the playout delay at the beginning of each talkspurt. There have been many algorithms proposed for estimating the network jitter such as, such as an Autoregressive (AR) model [5], Moving Average (MA) model [6], other statistical models [7-10], and adaptive filter models [11,12]. In this paper, an Artificial Neural Network (ANN) and wavelet technique are proposed and compared.

\[ d_t = \nabla + K \cdot D + M \cdot \nabla d \]

where \( d_t \) is the playout time for packet \( i \) of the \( k \)th talkspurt, \( t(i) \) is the transmission time, \( K, M \) are constants known as the buffering constants, \( D \) is the mean of network delay and \( \nabla d \) is the estimated mean jitter for talkspurt \( k \), defined as:

The remainder of this paper is structured as follows. Section 2 provides a background and overview of methodology, Section 3 provides an overview of the modelling approaches taken, Section 4, presents the results and finally Section 5 gives conclusions.

2. Background and methodology.

In this paper, PJSIP [13], an opensource VoIP application written in C, was adapted to measure the network jitter between two hosts. The application used in this paper first encodes the audio stream using G.729 B [14] into 20ms packets of length 80 bytes. Real Time Transport Protocol (RTP) is then used to sequence the packets and this is then encapsulated into a UDP packet for transmission across the internet.

At the receiver, an estimate of the network jitter (defined below) is used to allocate the playout delay buffer for each talkspurt as [14]:

\[ p^k(i) = t(i) + K D + M \nabla d \]

where \( p^k(i) \) is the playout time for packet \( i \) of the \( k \)th talkspurt, \( t(i) \) is the transmission time, \( K, M \) are constants known as the buffering constants, \( D \) is the mean of network delay and \( \nabla d \) is the estimated mean jitter for talkspurt \( k \), defined as:
\[
\overline{\nabla d} = \frac{1}{N} \sum_{j=1}^{N} \nabla \hat{d}(j)
\]  

where \( \nabla \hat{d}(j) \) is the estimated jitter (defined below) between packet \( j-1 \) and packet \( j \). Typically \( K \) is set to 1 while the value of \( M \) is varied to suit particular scenarios [14]. In this paper, the value of \( M \) is set to two. The jitter between two consecutive packets, \( i \) and \( i+1 \) is defined as:

\[
\nabla d(i+1) = d(i+1) - d(i)
\]  

where

\[
d(i) = r(i) - t(i)
\]

where \( d(i) \) is the network delay, \( \nabla d(i) \) is the jitter from packet \( i-1 \) to packet \( i \), \( r(i) \) is the arrival time*. 

Packets that arrive before their playout time slot (\( p^k(i) \)) are decoded using G729 B. Packets that fail to arrive on time or that are dropped are ignored and are decoded instead using a Packet Loss Concealment (PLC) algorithm [15]. This algorithm attempts to interpolate the speech signal using previous packets in the stream.

Finally, the performance of each proposed model has been analyzed by 3 metrics: the packet loss rate (the ratio of packets received, prior to \( p^k(i) \), to those sent), the PESQ MOS (explained below) and the additional buffering delay:

\[
\text{pd}(i) = p^k(u) - r(i)
\]

Perceptual evaluation of speech quality (PESQ) is a standard to measure the voice quality published by ITU-T. It compares the degraded speech signal, which is reconstructed after the network transmission and decoding, and the original signal. A MOS (mean opinion score) value is then produced. Commonly, the MOS value ranges from 0.0 (worst) to 4.5 (best) [16]. The overall scheme is shown below in Figure 2.

![Fig. 2. Block diagram for performance analysis methodology](image)

* Prior to implementing (4) the time stamps are first adjusted so that the average delay in the received and transmitted time stamps corresponds to 20ms (thus eliminating variations between the hosts’ system clocks.)

In this paper three types of neural network are compared for forecasting the jitter in the network. The first two types are the standard multi-layer perceptron (MLP) and a recurrent-MLP ([17] provides an excellent overview of these techniques). For the last type of network, the wavelet transform is applied to the inputs prior to modelling with an MLP to produce a Wavelet Packet-MLP (WP-MLP) [18]. The traditional back-propagation algorithm using Levenberg-Marquadt with cross validation has been used to train the networks [17]. The data is split into three different data sets used for training, validation (used for cross-validation and structure determination) and testing (used to compare each model after training), as shown is Table 1 below. Each of the networks has two hidden layers. To determine a suitable structure for the network (i.e. the number of nodes in each layer), 64 different networks are trained (ranging from a 1×1 to an 8×8 network) and their Prediction Mean Squared Errors (PMSE’s) compared over the validation set. The best structure is then selected. In addition, for each structure, 10 networks are trained, and the final output is composed of a simple average (in order to improve the generalisation capabilities of the network).

<table>
<thead>
<tr>
<th>Set Size</th>
<th>Training Set</th>
<th>Validation Set</th>
<th>Testing Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>1000</td>
<td>1000</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Description of the Neural Network Data Setting

3.1 Wavelet-Packet Neural Network

In recent years, wavelet network for function approximation and more specifically time series forecasting has been proposed by [18] and [20], respectively. The wavelet transform transforms a time domain signal into a time-frequency domain signal in which the coefficients represent the signal at progressively smaller frequency bands covering larger time spans*. Specifically, given a discrete time series \( x(k) \) the wavelet transform projects this series onto a new basis known as a wavelet basis [21], as:

\[
x(k) = \sum_{i=0}^{+\infty} \sum_{j=-\infty}^{+\infty} w_{i,j}^y(k) \psi_{i,j}(k)
\]

where \( w_{i,j}^y(k) \) are called the wavelet coefficients defined as the inner product of \( x(k) \) and the basis vectors \( \psi_{i,j}(k) \):

\[
w_{i,j}^y(k) = \int_{-\infty}^{+\infty} x(t) \psi_{i,j}^*(k)dt
\]

and

\[
\psi_{i,j}(k) = a_0^{1/2} \psi(a_0(t - k\tau_0))
\]

where \( \psi(t) \) is called the mother wavelet and \( \psi_{i,j}(k) \) is defined in terms of dilations (expansion), \( a_0, \) and translations (phase shift), \( \tau_0, \) of the mother wavelet and * denotes the complex conjugate. There are various types of mother wavelet, such as Haar wavelet, Meyer Wavelet, Coiflet wavelet, Daubechies wavelet, etc. [21]. The Haar wavelet and Daubechies wavelet are used in this paper. After transforming a time series, coefficients which ‘contain less information’ may be eliminated (shrinkage). This is achieved here by using the variance of the co-efficients as a measure of information (see [21] for more details). When combined with a neural network the overall model is known as a WP-MLP as shown below in Figure 3.

* This analogy is not strictly true, see [21] for more details.
4. Results

Several readings were taken ranging from 5 to 10 hours of continuous duplex transmission from NUI, Galway to Tokyo (trace 1, shown in Figure 4 below), Sydney (trace 2) and Chengu (trace 3).

The best structure for the neural networks was estimated as per Section 3. According to the PMSE performance on the validation set, the best structure is a 7×4 network with MSE $7.56 \times 10^{-5}$. Comparatively, the structure 8×4 and 6×4 provide an PMSE performance with $7.80 \times 10^{-5}$ and $8.42 \times 10^{-5}$ respectively.

Figure 5 compares the performance of the MLP, RMLP, and WP-MLP (Haar, Daubechies4, Daubechies6, [21]) models for the packet loss rate performance. The WP-MLP Haar has the best performance. When the buffering coefficient $M$ is more than 1.25, the Packet loss rate is less than the limit for interactive audio, 5% [3]. The MLP obtains a lower loss rate compared with other three methods. When $M > 2$, the Packet loss rate performance of WP-MLP Haar, MLP, RMLP all result in a packet loss ratio less than 5%. Comparatively, the RMLP and WP-MLP DB6 are not as effective as the other three methods for reducing the packet loss. For most buffering coefficients, the Packet loss rate of RMLP and WP-MLP are above 5%.
4.1 Additional Buffering Delay

Figure 6 illustrates the performance of the five adaptive playout delay methods with a varying buffering coefficient. For interactive audio, an acceptable packet delay is up to 400ms [2]. It can be seen that the WP-MLP Haar has the worst performance among these methods. It can only achieve the requirement with buffering coefficient $M$ less than 1.75. The MLP and RMLP show a similar performance to each other. The WP-MLP DB4 has better performance than both MLP and RMLP. The WP-MLP DB8 is most effective for the additional buffering delay control. From this figure, all of the additional buffering delay value of WP-MLP DB8 method is less than 400ms.

4.2 The trade-off between additional delay and late packet loss rate

For evaluating a playout schedule, the additional delay and packet loss rate should be considered.
together (Figure 7). This figure illustrates more clearly of the trade-off between additional delay and packet loss rates for the four different methods. This shows the WP-MLP Haar performance best for the playout delay adaptation with suitable value of buffering coefficient $M$, which has been improved compared with the traditional MLP. The MLP also shows a good performance, compared with other methods. Comparatively, the RMLP is not very suitable to be used in adaptive playout delay estimation.

Fig. 7. Trade-off between additional buffering delay and packet loss rate

4.3 PESQ MOS of G.729 B code

Table 2, below, gives an indication of the PESQ MOS score for the 5 techniques. As can be seen the WP-MLP using the Haar wavelet achieved the best performance.

<table>
<thead>
<tr>
<th>Model</th>
<th>WP-MLP Haar</th>
<th>MLP</th>
<th>WP-MLP DB4</th>
<th>RMLP</th>
<th>WP-MLP DB8</th>
</tr>
</thead>
<tbody>
<tr>
<td>PESQ MOS</td>
<td>2.51466</td>
<td>2.28812</td>
<td>1.94221</td>
<td>1.73342</td>
<td>1.49653</td>
</tr>
</tbody>
</table>

Table 2. PESQ MoS for the reconstructed voice by different playout delay adaptation

5. Conclusion

In this paper, several adaptive playout algorithms based on neural network have been presented and tested. Specifically, a combined wavelet transform/neural network approach is proposed. The effectiveness of these algorithms is evaluated using recorded traces by comparing the buffering delay and the packet loss ratios of each technique. Simulation results indicate that the Haar-Wavelets-Packet MLP adaptive scheduling scheme shows a higher precision and robustness in the adaptive network delay playout algorithm. The Haar-Wavelets-Packet MLP has improved the prediction capability for the variation of VoIP network over the traditional MLP. The WP-MLP DB4 and DB8 models show their advantage for additional buffering time but with higher packet loss rate. In future, the WP-MLP model has great potential for improving its prediction performance by use of different mother wavelets and different levels of decomposition.

References


