PLAYOUT SCHEDULING TECHNIQUE BASED ON NORMALIZED LEAST MEAN SQUARE (NLMS) ALGORITHM

BY

SALAH HUSSAIN AL-KHALIFA

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

May 2006

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

Dhahran 31261, Saudi Arabia

DEANSHIP OF GRADUATE STUDIES

This thesis, written by Salah Hussain Alkhalifa under the direction of his thesis advisor and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE IN ELECTRICAL ENGINEERING.

Thesis Committee

Dr. Jamil M. Bakhashwain
Department Chairman

Professor Asrar U.H. Sheikh
Thesis Advisor

Dr. Mohammad A. Al-Ohali
Dean of Graduate Studies

Dr. Adel Balgonaim
Thesis Advisor

20 May 2006
٢٢/٤/١٤٢٧هـ

Date

Dr. Adnan Al-Andalusi
Thesis Advisor

ACKNOWLEDGMENT

Acknowledgment is due to the King Fahd University of Petroleum & Minerals for supporting this research.

I wish to express my appreciation to Professor Asrar U. H. Sheikh who served as my major advisor. I also wish to thank the other members of my thesis committee Dr. Adel Balgonaim and Dr. Adnan Al-Andalusi.

TABLE OF CONTENT

LIST OF FIGURES

THESIS ABSTRACT

NAME: SALAH HUSSAIN AL-KHALIFA.

TITLE OF STUDY: PLAYOUT SCHEDULING TECHNIQUE BASED ON NORMALIZED LEAST MEAN SQUARE (NLMS) ALGORITHM.

MAJOR FIELD: ELECTRICAL ENGINEERING.

DATE: 20 May, 2006

In the packet-switched networks, real-time quality of multimedia communications can be adversely affected by transmission delays and their variations. Depending on the network conditions (traffic, load, bandwidth, ect.) packets are transmitted through different route paths that provide the optimum flow for the packets and reach the destination with variable delays. In the presence of jitter, the received packets are first queued into a smoothing buffer before being played out. We seek an optimum receiver that is able to receive and schedule the playout of the video packets at the correct time such that the tradeoff between playout delays and packet loss are improved.

In this work, we describe the packet flow in packet switched networks and identify the sources of delay that affect the real-time quality of multimedia packet communications. The Normalized Least Mean Square (NLMS) technique to schedule the playout delay for packet voice communications is reviewed and compared with Least-Mean Square (LMS) technique. We introduce packet video systems where we analyze the MPEG variable bit rate (VBR) encoder. Finally, we propose the NLMS technique in videoconferencing application to estimate video playout scheduling to obtain improved real-time video with lower packet losses.

ملخص الرسالة

الإســـــــــم: **صلاح حسين الخليفة**

عنوان الرسالة: **تعيين التشغيل الخارج بإسلوب خوارزمية تسوية أقل مربع الوسط.**

التخصـــــص: **الهندسة الكهربائية**

تاريخ التخرج: **20 مايو 2006**

في شبكات مقسمات–الحزم, يعتبر التأخير و التفاوت في وقت وصول الحزم من أهم العوامل التي تؤثر سلباً على جودة تشغيل الحزم متعددة الوسائط بصورة حية و متجانسة. بحسب حالة الشبكة, من حيث كمية الأحمال و سرعة نقل الحزم و غيرها من العوامل, ترسل الحزم من خلال الطرق و الإتجاهات التي توفر أفضل الظروف الممكنة لتوصيل الحزم. إن التأخير الغير–محدد يتسبب في الإختلافات المتباينة التي تطرأ علي الحزم الصوتية خلال عملية التوصيل و النقل. عند حدوث إختلافات في وقت الوصول, فإن الحزم الصوتية تستقبل في ذاكرة مرحلية لتخفيف التفاوت قبل تشغيلها الى الخارج., تسعى هذه الدراسة إلى إيجاد طريقة مثلى لإستقبال الحزم متعددة الوسائط و تعيين تشغيلها للخارج في الوقت الصحيح بحيث أن تتحسن العلاقة بين التأخير في تشغيل الحزم و كمية فقدانها.

في هذا البحث، سيتم وصف عملية انتقال الحزم في شبكات مقسمات–الحزم حيث تتم الإشارة إلى مصادر التأخير و التفاوت في وقت وصول الحزم متعددة الوسائط بصورة حية و متجانسة. أيضاً، يتم وصف طريقة عمل خوارزمية تسوية أقل مربع الوسط في تحديد الوقت الصحيح لتشغيل الحزم الصوتية. كذلك نستعرض حزمة الفيديو الرقمية التي تستخدم نظام التشفير MPEG بمعدل غير ثابت لمعالجة الصورة. أخيرا، يقدم البحث مقترحاً باستخدام خوارزمية تسوية أقل مربع الوسط في تطبيقات الاجتماعات المرئية (videoconferencing) لتحديد أفضل زمن للتشغيل الخارج و بالنتيجة تحسين عرض الفيديو الرقمي و تقليل كمية الحزم المفقودة.

CHAPTER 1

INTRODUCTION

1.1 PACKET MULTIMEDIA COMMUNICATION NETWORKS

Packet multimedia communication technologies use the data network infrastructure and resources to provide audio and video communication across dispersed geographical locations. It enables long distance calls and remote video conferencing to be conducted in any part in the world at very low costs. Similar to data packets, multimedia packets travel through the Internet with considerable reduction of operating costs. Also, the integration of multimedia and data traffic provides efficient utilization of network resources. Additionally, as the resources to the internet become available, exchanging multimedia does not require extra costs and the operation of adding, removing or configuring features is less complicated and can be conducted easily.

Conventional circuit-switched networks, which are designed for digital multimedia communications offer better quality because the communication channel between two stations has its own dedicated path and bandwidth. With circuit switching, multimedia packets are sent directly from the transmitter to the receiver where they are received in order, one after another in a single path. In packet-switching, routers determine a path for each packet on dynamic basis where they are directed over many paths to reach to the destination.

In IP telephony, for example, analog voice signals generated for transmission are first converted into a bit stream. The digitized bits are then packetized and sent over the network. The packetization process collects compressed voice frames and converts them into an IP packet. At the receiving end, the process is reversed and the voice frame is decompressed.

In packet-switched networks, delay in multimedia packets becomes an important challenge as the perceived quality of voice or video is sensitive to delay. Variation in the delay or *jitter* is another problem for interactive multimedia applications. Hence, the receiver must exploit an effective playout mechanism to account for delay variations experienced by each packet and provide acceptable and synchronous communication between sending and receiving ends.

1.2 PRINCIPLES OF PACKET SYNCHRONIZATION

Packet switched network represents an effective technology to integrate multimedia and data packets over transmission path. The transmission over packet-switched networks, however, requires reconstruction of the continuous stream of data from the set of packets sent through the network. Because the packets travel through several network routers before reaching the destination, they encounter different amount of delay due to the queuing process in the routing path.

Multimedia packets are produced at the *packet sender* (PS) and sent at a regular time interval through the network. Each packet passes through the network encounters varying amount of delays due to router queuing. The arrival of each packet shall occur before the read time or *playout time* of the *packet receiver* (PR) so that packets can be reconstructed into a continuous stream of multimedia samples such as video images or voice samples. Hence, the network shall provide a mechanism to maintain a proper amount of time delay to avoid packet loss. In general, the packet synchronization is more significant in low speed networks assuming constant traffic rate of packets on the network.

To optimize the delay and minimize packet loss, each packet has to be received at a fixed time interval D_m, which represents the maximum allowable delay time before the packet is effectively become too late to be useable. Advanced studies of packet delay show that D_m is basically controlled by two delay factors namely, fixed delay D_f and variable delay D_v. The fixed delay arises from the packet propagation, fixed buffering delays, in PS and PR. The variable delay results mainly from the queuing process in the network. The variable delay D_v introduces the tradeoff between the delay and packet loss. If D_v is increased, packet delay will increase as a result and hence the task of packet synchronization and consequently building of the original message stream will become more difficult. The relation between packet delay and packet loss rate is a challenging task to formulate. Lower playout delay will result in a lower percentage of received packets because there will be a lower time margin to receive useable packets and hence late packets will be dropped. Thus, packet variable delay D_v should be chosen to achieve a certain service quality target [4]. Equation (1-1) shows the relation between D_v, D_f and D_m.

$$D_m = D_f + D_v \qquad (1.1)$$

To properly choose the target delay D_m, an effective mechanism is needed to determine the playout time for each incoming packet. Thus, the PR must determine the amount of delay experienced by each packet. In fact, delay can be estimated for one packet and the relative production time of subsequent packets can be encoded in the information sent in each packet such as sequence number. In this way, delay measurements could be made once per packet, once per multimedia session or any in between interval. This scheme relies on the clock of the PS to be as close in frequency as possible to the clock of at the PR so that clock skew does not distort the timing.

The clock of the PR should operate synchronously with that in the PS. If the PR clock is slower than the PS clock (higher D_m), packets will begin to accumulate in the buffer to the extent that the buffer is unable to accept any more packets. On the other hand, if the PR clock runs faster than PS clock packet loss rate will ultimately increase. To optimally tolerate these situations, the delay target of the first packet shall be corrected by the amount of time drift between the two clocks in the PR and PS.

One of the techniques that can be used to provide effective clock synchronization between PR and PS is *Absolute Timing* which can be used in higher speed networks and lower speed. In this technique, PR and PS are synchronized through a master clock that controls packets transmission and reception through networks is used. Specifically, each packet transmitted from the PS contains an indication (time-stamp) about its production time, which is interpreted by the PR to determine the playback time and packet sequence. It must be noted that the distribution of master clock timing to local clocks over communication channels may be affected by propagation delay. Hence, to provide optimum timing reference to telecommunication devices, the propagation delay must be exactly known.

1.3 THESIS OVERVIEW

In asynchronous packet-switched networks, real-time quality of multimedia communications can be adversely affected by transmission delays and their variations. Packet-switched networks provide multiple routes by which packets are transported from source to destination. Depending on the network conditions, such as congestion and bandwidth, packets are transmitted through different route paths that provide the optimum flow for the packets and reach the destination with variable delays. In the presence of jitter, the received packets are first queued into a smoothing buffer before playout. We seek an optimum receiver that is able to receive and schedule the playout of the video packets at the correct time such that the tradeoff between playout delays and packet loss are improved.

In this work, we describe the packet flow in packet switched networks and identify the sources of delay that affect the real-time quality of multimedia packet communications. The Normalized Least-Mean Square (NLMS) technique to schedule the playout delay for packet voice communications is reviewed. We introduce the delay in packet voice communications where we apply NLMS, Least-Mean Square (LMS) and Auto-regression (AR) algorithms to estimate the playout scheduling and compare their performances in terms of total end-to-end delay and packet loss rate. Similarly, we introduce the delay in MPEG packet video communications (*Motion Picture Expert Group:* which is a video coding standard designed for digital storage of quality video for later playout) and identify the associated delay components. We apply NLMS, Least-Mean Square (LMS) and Auto-regression (AR) algorithms to estimate the playout scheduling and compare their performances in terms of total end-to-end delay and packet loss rate.

Our contribution in this context is to develop a NLMS-based receiver for videoconferencing applications so that the total end-to-end delay is reduced and packet loss rate is minimized.

CHAPTER 2

BACKGROUND

2.1 PACKET DELAYS

The delay packets experience as they move from source to destination results from different factors related to data processing and communication network. The packet delay contains constant and variable components. The constant delays typically result from packet propagation time whereas the variable delays mainly results from network queuing. The packet encoding process can contribute to constant delays if the packets are generated at constant time intervals as in packet voice. If the packets are generated at variable time intervals as in packet video, then the encoding process will contribute to variable delays. The *encountered* delay is the sum of all delay components a packet experiences from the transmitter to the receiver just before playout as shown in figure (2-1).

The variable encountered delays at the receiver may result in out of order and overlapping packets. In real-time applications, this can impair the perceived voice or video quality as it leads to inconsistence and incoherent playout rate of packets. Hence, the variable delays must be smoothed before playing packets out.

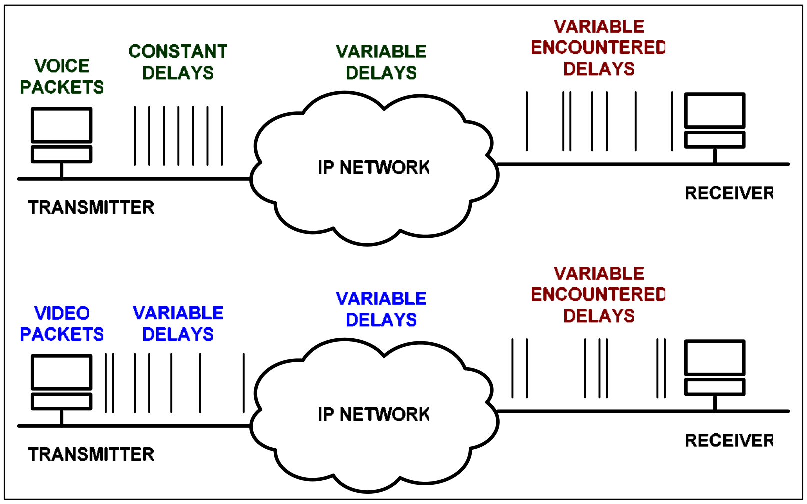

Figure (2-1): The packet encountered delays.

2.2 PLAYOUT SCHEDULING MECHANISM

To obtain consistent playout of real-time packets, the variable delays must be smoothed by exploiting a playout buffer at the receiver. The role of the playout buffer is to hold and schedule the delay of the received packets for a short amount of time just before playout rate as shown in figure (2-2). The delay the playout buffer exerts on each packet is added to the packet encountered delay. The sum of the playout buffer delay and the packet encountered delay is referred to as the *end-to-end* delay.

In real-time applications, the buffering delay has to be scheduled such that it provides lower end-to-end delay at reduced packet loss rate. For example, a scheduling mechanism that delays all received packets for a fixed amount of time greater or equal the largest packet encountered delay will be theoretically able to receive all packets with out losses on expense of large end-to-end delay as shown in figure (2-3). On the other extreme, if the scheduling mechanism was selected to delay all received packets for a fixed amount of time equal to the shortest packet encountered delay, then the receiver will be able to reduce the end-to-end delay on the expense of higher packet loss rate as shown in figure (2-4).

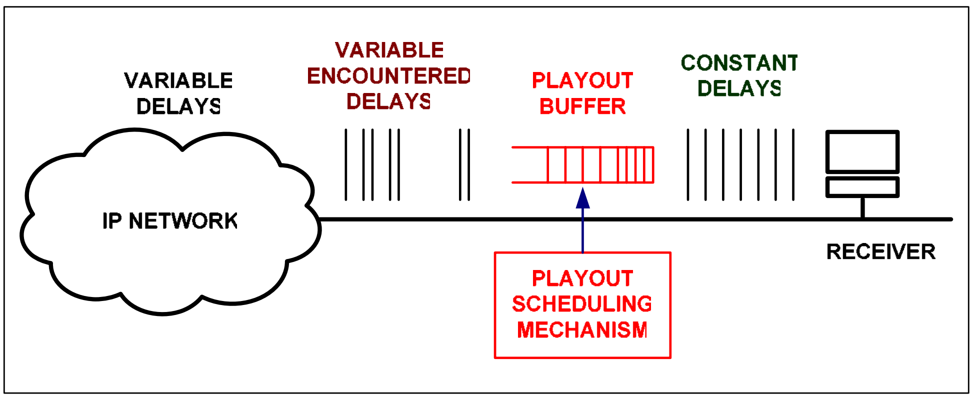

Figure (2-2): Playout scheduling mechanism.

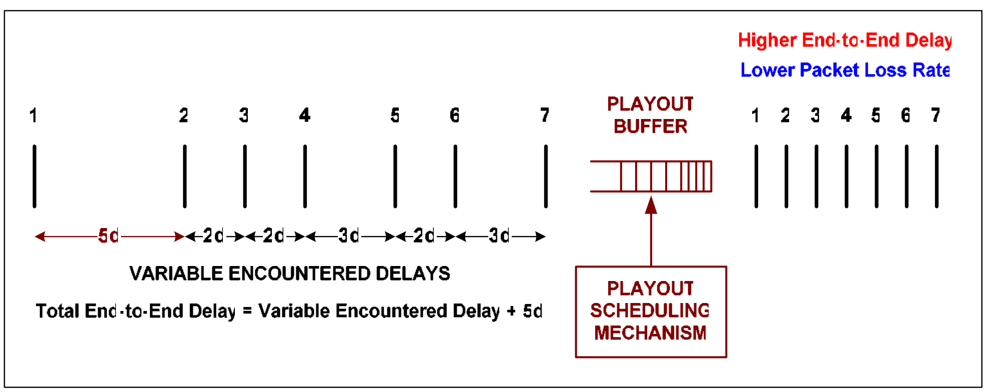

Figure (2-3): High playout delay receiver.

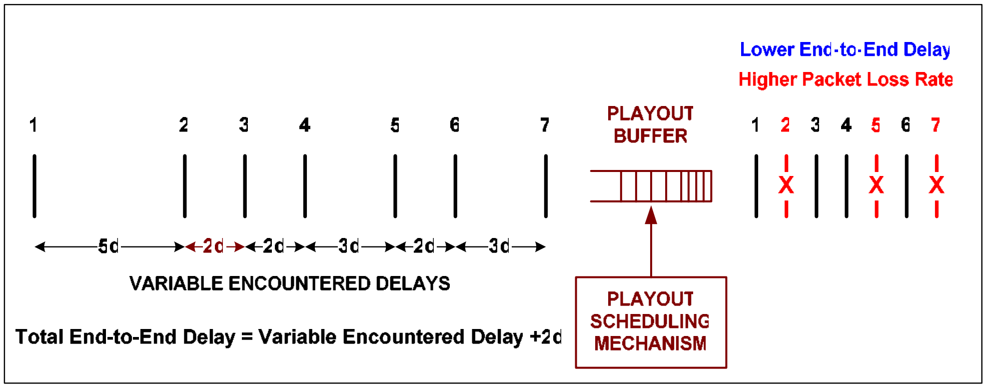

Figure (2-4): Low playout delay receiver.

In real-time application such as voice and video multimedia communications, the playout scheduling mechanism should be designed to improve the tradeoff between the total end-to-end delay and packet loss rate by estimating the packet delay and adjusting the playout buffer delay on dynamic basis. This can be achieved at the receiver by applying certain algorithm that estimates the packet delay and adjusts the playout delay time just before playout as shown in figure (2-6).

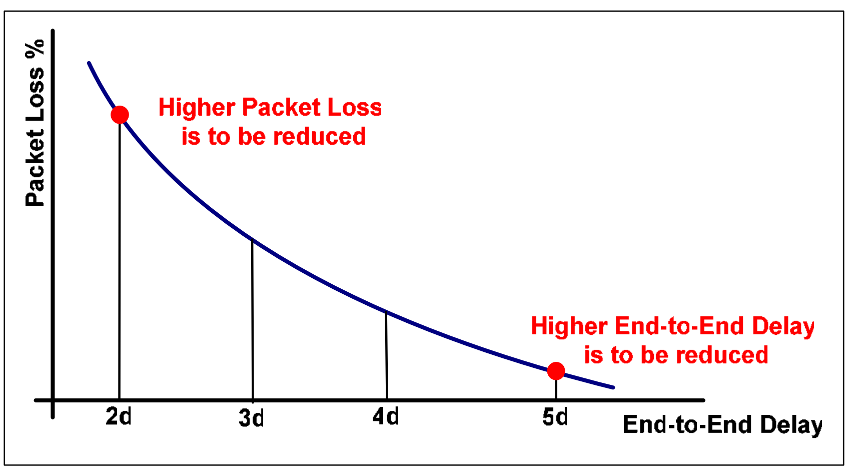

Figure (2-5): Relation between end-to-end delay and packet loss rate.

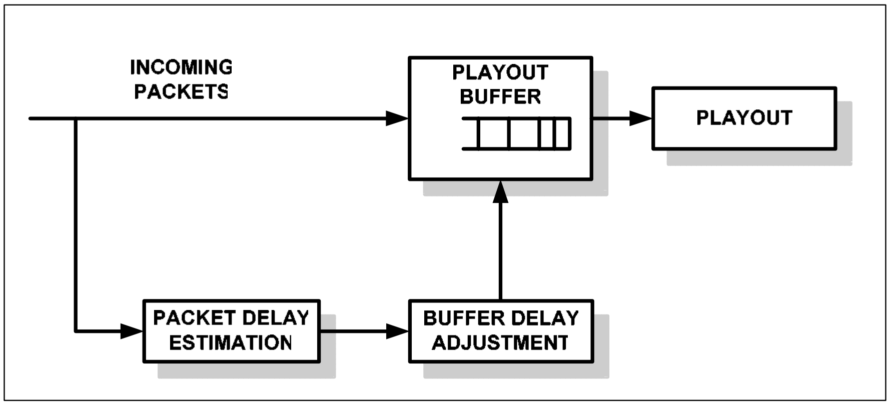

Figure (2-6): Packet delay estimation and playout buffer adjustment.

2.3 RELATED WORK OF DELAY ESTIMATION AND ADJUSTMENT

Several related studies have been proposed to improve the estimation of packet multimedia delay and minimize the packet loss rate. In the following subsection, the works related to packet delay estimation is reviewed.

2.3.1 ADJUSTMENT OF PACKET DELAY BETWEEN TALKSPURT

The delay adjustment between talkspurts approach was proposed in [4] where playout delay adjustments were made during silence periods as they are less likely to be perceptible by users. In this way, the playout delay is adjusted on a per talkspurt basis during the silence periods between talkspurts as shown in figure (2-7). The basic algorithm for this approach [17] sets the playout time for the first packet voice in talkspurt k to:

$$p_1^k = t_1^k + D^k \qquad (2\text{-}1)$$

where:

- t_1^k is the sender time-stamp of the first talkspurt.
- p_1^k is the playout time of the first talkspurt.

- D^k is the total end-to-end delay for the packets in talkspurt k.

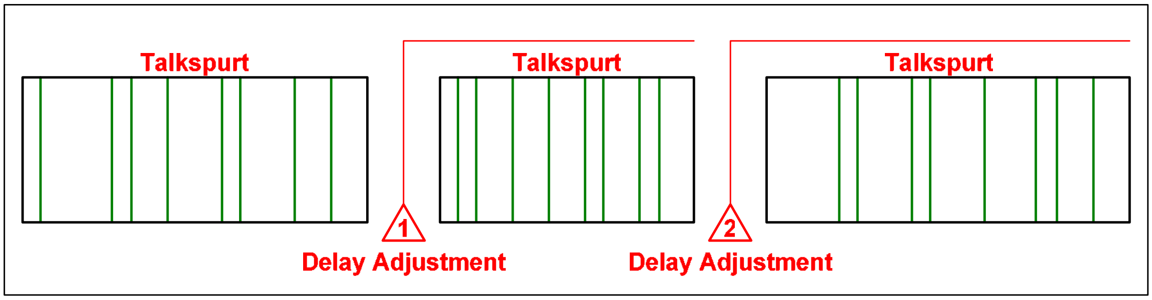

Figure (2-7): The delay adjustment between talkspurts approach.

Subsequent packets in a talkspurt have the same end-to-end delay and hence, the playout time for packet i in talkspurt k can be calculated as an offset from the first packet in the talkspurt

$$p_i^k = t_i^k + D^k$$

or equivalently (2-2)

$$p_i^k = p_1^k + (t_i^k + t_1^k)$$

In this algorithm, the playout of each packet in the talkspurt is directly controlled by the choice of the scheduled end-to-end delay D^k in each talkspurt k as formulated in equation (2-2). Because the encountered delay of each packet in the talkspurt has different amount of latency compared with original packet production delay as shown in figure (2-8), the scheduled end-to-end delay D^k can be too small to reduce the packet loss rate or too large to reduce the packet

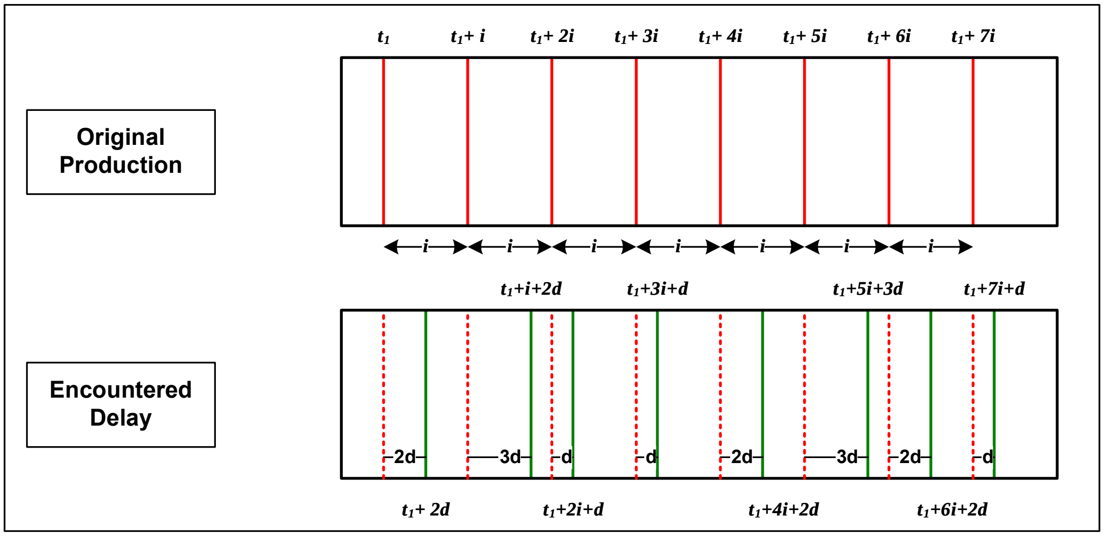

Figure (2-8): Original and encountered packet delay.

end-to-end delay. For example, if the scheduled delay is adjusted such that $D^k = d$, only those packets with encountered delay of less than or equal to d will be accurately received and all other packets encountering delays greater than d will be effectively lost as shown in figure (2-9).

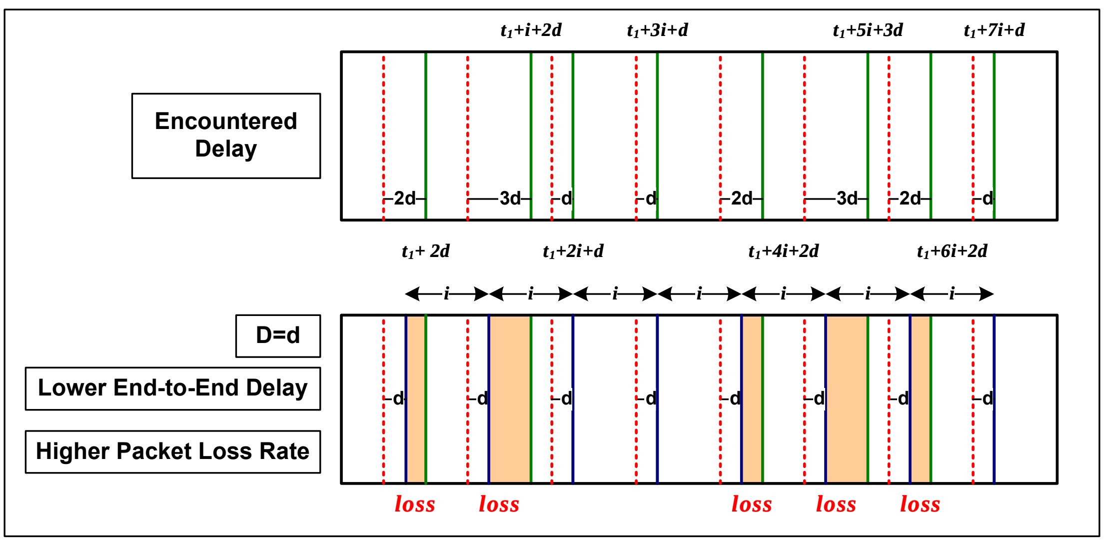

Figure (2-9): Adjustment between talkspurts with scheduled delay $D^k = d$.

Increasing the scheduled end-to-end delay, such that the adjustment is made at $D^k = 2d$, can improve the packet loss rate as shown in figure (2-9). In this case, the packets with encountered delay of less than or equal to $2d$ will be accurately received whereas and all packets encountering delays larger than $2d$ will be effectively lost.

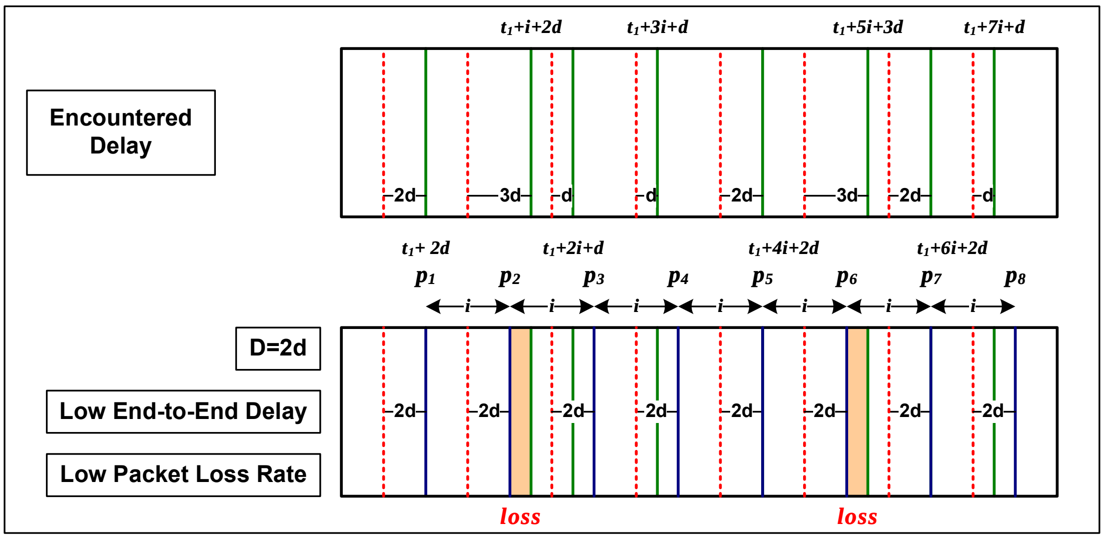

Figure (2-10): Adjustment between talkspurts with scheduled delay $D^k = 2d$.

Further increasing to the scheduled end-to-end delay will reduce the packet loss rate. As shown in figure (2-11), adjusting the delay to $D^k = 3d$ will guarantee accurate reception of all packets but at higher end-to-end delay.

Because the end-to-end adjustment is made once per talkspurt and applied to all packets in it, this algorithm may not be very effective when the variance of the packets encountered delay is large. Therefore, the need to estimate the delay on per-packet basis becomes another attractive approach to improve the estimation of the packet delay and improve the trade off between the end-to-end delay and packet loss rate.

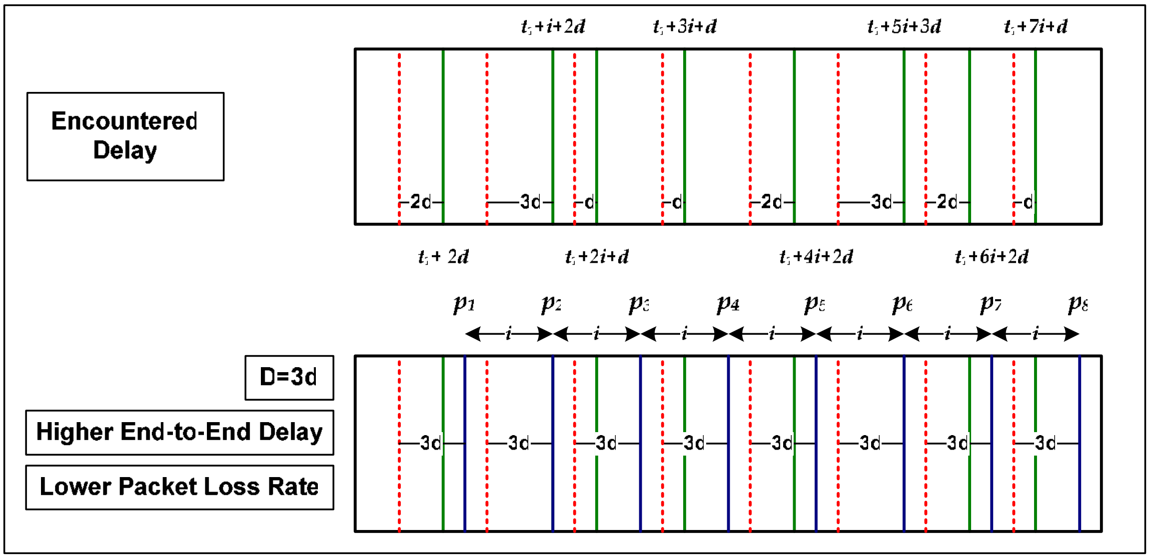

Figure (2-11): Adjustment between talkspurts with scheduled delay $D^k = 3d$.

2.3.2 AUTOREGRESSIVE ALGORITHM

Unlike the adjustment between talkspurts algorithm, the end-to-end delay in the autoregressive approach is estimated autoregressively for each packet. In [17], the autoregressive (AR) estimate was proposed to determine and compute the packet voice playout on per-packet basis. In this algorithm, the end-to-end delay is adjusted by estimating the average packet delay and variable packet delay in equations (2-3) and (2-4) respectively.

$$\hat{r}(i) = \alpha.\hat{r}(i-1) + (1-\alpha).n(i) \qquad (2\text{-}3)$$

$$\hat{v}(i) = \alpha.\hat{v}(i-1) + (1-\alpha).|\hat{r}(i) - n(i)| \qquad (2\text{-}4)$$

where:

- $\hat{r}(i)$ is the autoregressive estimate for the average packet delays.
- $n(i)$ is the actual delay by the i-th packet.
- α is a factor to control the convergence of the algorithm.
- $\hat{v}(i)$ is the variation in the packet delay.

The term $|\hat{r}(i) - n(i)|$ in equation (2-4) represents the magnitude of the error and the variance between the estimated average packet delay and the actual packet delay where only positive magnitudes are considered.

The end-to-end delay, $D(i)$ is computed as

$$D(i) = \hat{r}(i) + \beta.\hat{v}(i) \qquad (2\text{-}5)$$

where:

- β is a factor to control the tradeoff between the packet loss rate and end-to-end delay.

The value of α determines how fast the AR delay estimation adapts to fluctuations in network delay. β is a control factor that accommodate changes in network conditions. A higher value of β results in lower packet loss rate, however it also increases the total end-to-end delay.

Also in [17], a modification to the autoregressive algorithm was proposed, which accounts for increasing and decreasing network delays. Specifically, the modification focuses on the value of α in the previous algorithms where it is changed based on the following cases:

- $n(i) > d(i)$ then: $\alpha = 0.75$
- $n(i) < d(i)$ then: $\alpha = 0.998002$

With the value of α properly chosen, the same set of equations (2-3), (2-4) and (2-5) are used to calculate the average network delay, network variation delay and the end-to-end delay.

2.3.3 ADAPTIVE FILTERING ALGORITHMS

In the AR approach, the adjustment to the end-to-end delay is adjusted by autoregressively computing the average packet delay and its variations. In [18], the Normalized Least-Mean Square (NLMS) technique was proposed to adaptively predict the average packet voice delay. An accurate prediction of the packet delays can rapidly track network changes and hence adjust the playout delay more effectively. The adaptive predictor diagram is shown in figure (2-12).

The adaptive algorithm suggested in [18] provides better means of determining the average packet delay compared with that of the AR as given in equation (2-3). Thus, in adaptive filtering technique, NLMS will be used to predict the packet average delay where it will be used in equations (2-4) and (2-5) to estimate the varying packet delay and the end-to-end delay respectively.

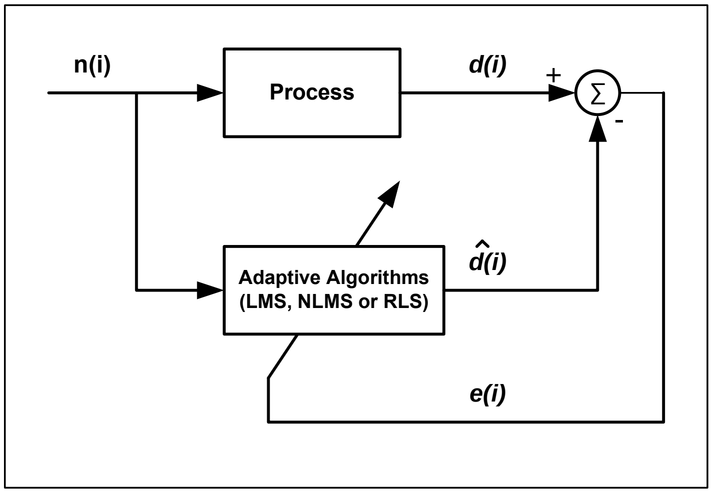

Figure (2-12): Adaptive predictor diagram.

Adaptive algorithms, such as Least-Mean Squares (LMS), Normalized Least-Mean Squares (NLMS) and Recursive Least Squares (RLS), tend to minimize the expected mean-square error between the actual delays and the estimate ones. Previously received delays are passed through a finite impulse response (FIR) adaptive filter with specific taps length to determine and compute the current estimate of the packet delay. The mean-square error between the actual and estimated delay is then used to adjust the tap weights of the adaptive filter via adaptive weight control mechanism.

A. LMS Algorithm (LMS)

Given a certain desired signal $d(i)$, the LMS algorithm processes an $N \times 1$ input signal vector $\vec{n}(i)$ through a controlled $N \times 1$ FIR filter $\vec{w}(i)$ to produce an estimated value $\hat{d}(i) = [\vec{w}(i)]^T * \vec{n}(i-1)$. The error between the estimated and desired response $e(i) = d(i) - \hat{d}(i)$ is used to update the FIR filter coefficients such that the next error becomes smaller. The next filter update is $\vec{w}(i+1)=\vec{w}(i)+\mu\cdot e(i)\cdot\vec{n}(i)$ where μ is a step size constant controlling the convergence of the algorithm.

B. Normalized Least Mean Square Algorithm (NLMS)

The NLMS algorithm procedure is similar to LMS except for the mechanism in which the FIR filter is updated. In NLMS, the step size $\tilde{\mu}$ is adjusted in every iteration step according to the squared norm of the input signal $\vec{n}(i)$. The next filter update using NLMS is $\vec{w}(i+1)=\vec{w}(i)+\tilde{\mu}\cdot\frac{e(i)}{\|\vec{n}(i)\|^2}\cdot\vec{n}(i)$.

C. Recursive Least Squares Algorithm (RLS)

The RLS algorithm procedure is also similar to NLMS except for the mechanism in which the FIR filter is updated. In RLS, the step size is adjusted in every iteration step according to more accurate variance estimation of the input signal $\vec{n}(i)$. The next filter update in RLS is $\vec{w}(i+1)=\vec{w}(i)+P_i\cdot e(i)\cdot\vec{n}(i)$ where

$$P_i=\lambda^{-1}\left[P_{i-1}-\frac{\lambda^{-1}P_{i-1}u_i^*u_iP_{i-1}}{1+\lambda^{-1}u_iP_{i-1}u_i^*}\right],$$

$$P_{i-1}=\varepsilon^{-1}I\ ,$$

$$0<<\lambda\leq 1$$

and I is the identity matrix.

In the next chapter, the performance of Normalized Least Mean Square (NLMS) algorithm will be analyzed and the mechanism of estimating the delay in packet multimedia communications will be discussed [25, 26].

2.3.4 STATISTICALLY-BASED ALGORITHMS

Statistical based approach that uses statistics of past delays was proposed to determine and compute the current playout delay [12]. Network delays for the past i packets are stored and playout delay is selected such that only a tolerable percentage of packets will be lost.

The *Adaptive Gap-Based* Statistical algorithm [21], for example, stores the network delay values for a talkspurt and computes the optimum playout delay for the talkspurt. The minimum playout buffering delay resulting in a specific loss rate for a talkspurt defines the optimum playout delay for each talkspurt. The amount of delay is calculated once the talkspurt is finished. The playout delay for the next packet is increased or decreased based on the minimum playout delay calculated for the previous talkspurt.

Another statistical algorithm uses *Histogram Approach* where the delay of each packet is logged and used to update a histogram of packet delays after the arrival of each packet [21]. The histogram exploits previous packet delays create delay distribution function. The end-to-end delay is computed for each packet by finding the average delay for a given percentage of delay points in the distribution. This approach suffers from poor efficiency at two extremes specifically, when the number of histogram points is either too small or too large. Small number of delays in the histogram will not provide strong view of the past characteristics of the delays. Large number of delays in the histograms will degrade the algorithm performance because of weakness of the algorithm in tracking and reacting to the changes in the network. As suggested in [21], the proposed delay points needed to build the histogram is 10,000 packets.

In our context, we will not be using the statistically-based algorithms due to its complexity compared with AR and NLMS algorithms and therefore more details in this scope will be left for further study.

CHAPTER 3

ADAPTIVE PACKET PLAYOUT USING NLMS ALGORITHM

3.1 REAL-TIME PACKET PLAYOUT TECHNIQUES

Because the internet does not guarantee the delivery of packets, the packets are subject to loss during network transmission. The delay each packet encounters varies depending on the path it takes in the network and the level of congestion at network queues. Methods to reduce the delay variations can be classified into three main approaches:

1- Source-based Approach: According to network congestion and load conditions, the transmitter adjusts the codec mode to increase or decrease the bit production rate. The network conditions are determined by measuring total end-to-end delay, delay variation and packet loss rate.

2- Network-based Approach: In this case, resources are reserved in each network nodes to guarantee certain level of quality of service (QoS). In this way, real-time packets can be identified and given higher priority as they move through different networks.

3- Receiver-based Approach: In this case, the variation of delay is smoothed in the receiver buffer for short period of time before playout. Packets which arrive after their schedule playout time are considered ‘lost’. Although increasing the buffer delay can reduce the loss rate, the total end-to-end delay of the packet will increase. Hence, there is a tradeoff between average packet end-to-end delay and packet loss.

Unlike source-based approaches, which require bit rate adjustments at the encoder / decoder, and network-based approaches, that require modification on the existing network infrastructure, Receiver-based approaches have gained large interest as they only require playout buffer and delay control mechanism. Several techniques and algorithms have been developed to estimate the playout delay using the receiver-based approaches. Basic algorithms adjust the packet playout time during silence periods such that all packets in a talkspurt are equally delayed. Another approach is to scale individual voice packets using time-scale modification. Moreover, adaptive filtering technique can be also used to predict network delays.

3.2 MATHEMATICAL ANALYSIS OF NLMS

The NLMS is an adaptive filtering algorithm which is widely used due to its computational simplicity and ease of implementation. NLMS is known to be robust against finite word length effects and has faster convergence compared with the conventional LMS algorithm. Also, NLMS has a stable operation and eliminates the gradient noise amplification [24, 25, 26].

The NLMS algorithm can be used as an adaptive predictor to estimate the output of stochastic systems such as variable delay of packet multimedia communication systems. Specifically, NLMS tends to identify the behavior of a stochastic system by processing the system inputs $\vec{n}(i)$, and providing an estimate of the next output value, $\hat{d}_i$. The error between the actual and estimated outputs e_i, is used to adjust the tap weights of the adaptive filter via adaptive weight control mechanism.

As shown in figure (3-1), the stochastic system produces new outputs, the adaptive system processes the $N \times 1$inputs $\vec{n}(i)$, equation (3-1), and the $N \times 1$FIR filter $\vec{w}(i)$, equation (3-2), computes the estimation of the new output $\hat{d}(i)$ as in equation (3-3).

$$\vec{n}(i) = [\ n(i-1),\ n(i-2), \cdots, n(i-N)\ ] \tag{3-1}$$

$$\vec{w}(i) = [\ w(i),\ w(i-1), \cdots, w(i-N+1)\ ] \tag{3-2}$$

$$\hat{d}(i) = [\vec{w}(i)]^T * \vec{n}(i) \tag{3-3}$$

The desired response $d(i)$is continuously compared with $\hat{d}(i)$ and the error function $e(i)$, equation (3-4), is used to adjust the FIR filter tap-weights new response $\vec{w}(i+1)$.

$$e(i) = d(i) - \hat{d}(i) \tag{3-4}$$

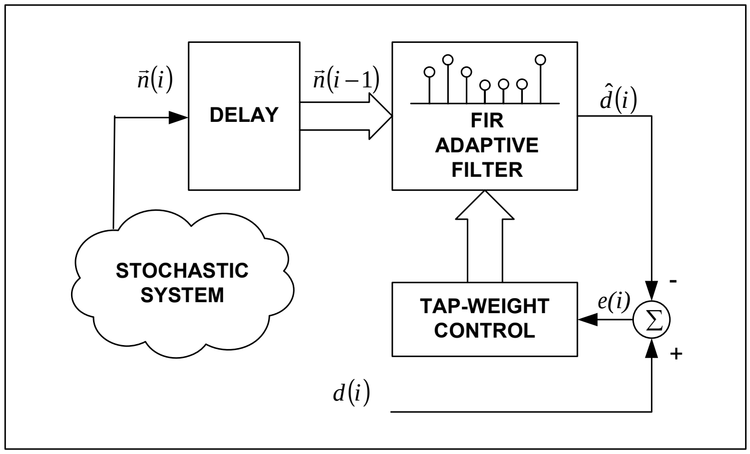

Figure (3-1): Adaptive on-step predictor.

Figure (3-2) shows an example for NLMS algorithm with N=7.

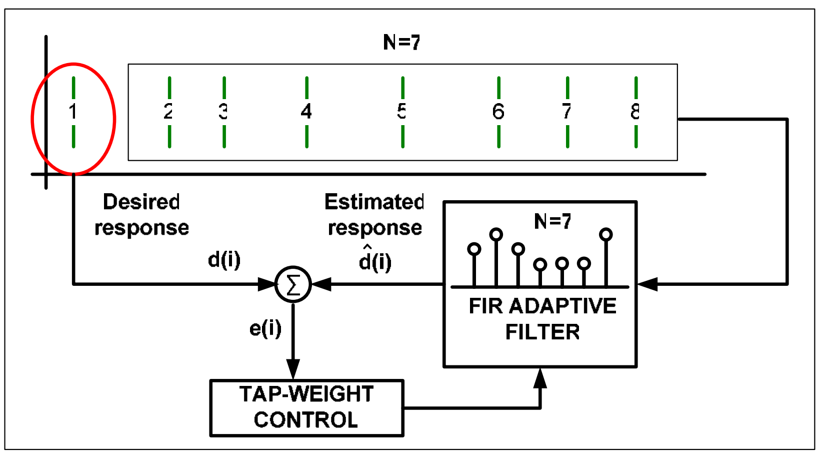

Figure (3-2): Detailed adaptive on-step predictor.

As $i \to \infty$, the FIR filter $\vec{w}(i)$ will steadily come closer to the system response $\vec{W}_o$, or equivalently $\vec{w}(i) = \vec{w}_o$. Vector interpretation of successive NLMS estimates is shown in figure (3-3). As can be seen from the figure, the NLMS tends to update $\vec{w}(i)$ during each step such that it becomes closer to $\vec{W}_o$ and hence more accurate identification of stochastic system behavior is obtained [24].

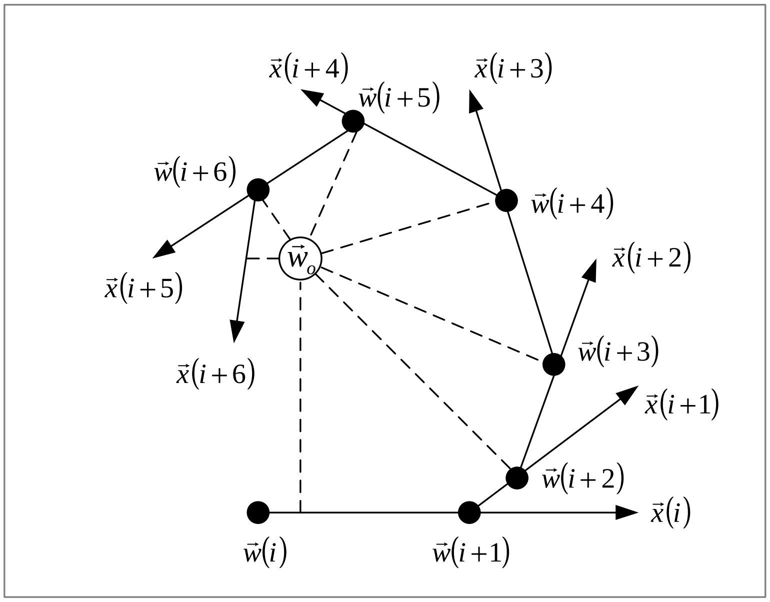

Figure (3-3): Vector Interpretation of Successive Estimates Produced by NLMS.

NLMS follows the principle of *Minimum Disturbance*. That is, given $\vec{n}(i)$ and $d(i)$, find $\vec{w}(i+1)$ such that the squared Euclidean of $\delta \cdot \vec{w}(i+1)$, or $\| \delta \cdot \vec{w}(i+1) \|^2$ is minimum, where δ is a small value multiplication factor [23, 24, 25]. Hence,

$$\delta \cdot \vec{w}(i+1) = \vec{w}(i+1) - \vec{w}(i) \tag{3-5}$$

The next estimate is given by:

$$\vec{w}(i+1)^T * \vec{n}(i) = d(i) \tag{3-6}$$

According to the method of Lagrange Multipliers, the cost function is given by:

$$J(i)=\| \ \delta \cdot \vec{w}(i+1) \ \|^2 + RE\left[\lambda^*\left(d(n)-\vec{w}(i+1)\cdot \vec{n}(i)\right)\right]$$

or equivalently, (3-7)

$$J(i)=\left(\vec{w}(i+1)-\vec{w}(i)\right)*\left(\vec{w}(i+1)-\vec{w}(i)\right)+RE\left[\lambda^*\left(d(n)-\vec{w}(i+1)\cdot \vec{n}(i)\right)\right]$$

where λ is the *Lagrange Multiplier*.

Note that the change in cost function shall be kept close to zero to assure proper convergence of the algorithm. Hence, assuming real values, we differentiate both sides of equation (3-7) with respect to $\vec{w}(i+1)$ and equating to zero to find $\vec{w}(i+1)$:

$$0 = 2\left(\vec{w}(i+1)-\vec{w}(i)\right)+\lambda \cdot \vec{n}(i)$$

or equivalently: (3-8)

$$\vec{w}(i+1)=\vec{w}(i)+\frac{1}{2}\lambda \cdot \vec{n}(i)$$

By substituting the result of equations (3-8) in (3-6) and solving for λ:

$$d(i)=\left(\vec{w}(i)+\frac{1}{2}\lambda \cdot \vec{n}(i)\right)^T * \vec{n}(i)$$

or equivalently: (3-9)

$$\lambda = \frac{2\cdot e(i)}{\| \ \vec{n}(i) \ \|^2}$$

By using the result of equations (3-9) and (3-8) and substituting in equation (4-5) we have:

$$\delta \cdot \vec{w}(i+1) = \frac{e(i)}{\| n(i) \|^2} \cdot \vec{n}(i) \tag{3-10}$$

To have control over the change of the tap-weight from iteration to the other while preserving the direction of the vector $\vec{n}(i)$, we introduce a positive real scaling factor $\tilde{\mu}$ to equation (3-10) to become:

$$\delta \cdot \vec{w}(i+1) = \tilde{\mu} \cdot \frac{e(i)}{\| n(i) \|^2} \cdot \vec{n}(i) \tag{3-11}$$

using the result of equation (4-11) and substituting in equation (4-5) we obtain:

$$\tilde{\mu} \cdot \frac{e(i)}{\| \vec{n}(i) \|^2} \cdot \vec{n}(i) = \vec{w}(i+1) - \vec{w}(i)$$

or equivalently, (3-12)

$$\vec{w}(i+1) = \vec{w}(i) + \tilde{\mu} \cdot \frac{e(i)}{\| \vec{n}(i) \|^2} \cdot \vec{n}(i)$$

The adjustment applied to tap-weights $\vec{w}(i)$ is directly proportional to $\vec{n}(i)$. If $\vec{n}(i)$ is large in magnitude, the system will suffer from gradient noise amplification. Therefore, NLMS overcomes this difficulty by applying normalized adjustments with respect to $\vec{n}(i)$ [23, 24, 25].

To avoid division by zero (when $\| \vec{n}(i) \|^2 = 0$), we introduce a factor a, (where $a > 0$) of very small value to equation (3-12). Therefore, we have:

$$\vec{w}(i+1) = \vec{w}(i) + \tilde{\mu} \cdot \frac{e(i)}{\| \vec{n}(i) \|^2 + a} \cdot \vec{n}(i) \tag{3-13}$$

3.3 PLAYOUT BASED ON NLMS ADAPTIVE PREDICTOR

The NLMS is an adaptive algorithm that seeks to minimize the expected error (cost function) between the actual data and the estimate by adjusting the coefficients of FIR filter. In our context, we use the predicted estimate of network delay to adjust the buffer delay because proper adjustment to buffer delay can lead to either (or both) lower packet loss percentage (at certain total end-to-end delay) or lower end-to-end delay (at certain loss percentage). The main objective here is to adjust the buffer delay more effectively and to track rapid changes packet delays more accurately [18, 23].

Predicted delay $\hat{d}(i)$ is computed from the vector containing the past N delays $\vec{n}(i) = [\ n(i-1),\ n(i-2), \cdots, n(i-N)\ ]$ and the Nx1 FIR filter $\vec{w}(i)$, equation (3-14):

$$\hat{d}(i) = [\vec{w}(i)]^T \cdot \vec{n}(i) \tag{3-14}$$

The error between actual and estimated delays is given by:

$$e(i) = d\,(i) - \hat{d}(i) \tag{3-15}$$

The filter tap-weights are then updated after each packet using the NLMS as in equation (3-13).

The reactive algorithm, equations (4-4) will be used to compute the variation auto-regressively. Then the total end-to-end delay $D(i)$ will is computed using $\hat{n}(i)$ instead of $\hat{r}(i)$. Hence, the total end-to-end delay $D(i)$ (equation 4.5) will be modified by using the packet delay estimate:

$$D(i) = \hat{r}_{NLMS}(i) + \beta.\left[\alpha.\hat{v}(i-1) + (1-\alpha).|\hat{r}_{NLMS}(i) - n\ (i)|\right] \qquad (3\text{-}16)$$

CHAPTER 4

APPLYING LMS AND NLMS ALGORITHMS TO PACKET VOICE

4.1 PACKET VOICE COMMUNICATIONS DELAY

Real-time quality in audio applications over the internet can be achieved through lowering the total end-to-end delay and by exploiting effective playout mechanism in the receiver. The objective in this context is to improve the voice applications over internet by lowering the packet loss and reducing the total end-to-end delay.

Varying total end-to-end delays can negatively impact the perceived quality of packet voice streams. Beside network queuing, varying delays can result from encoding and compression algorithms when the encoder produces different frames size. Although compression is required to produce efficient packets size on the network, it is usually not essential for voice applications especially when the voice samples production is constant and frame size is small. Hence, most of the total end-to-end delay experienced by packet voice results by the varying delays in the network due to packet queuing.

Packet loss results from short buffering delay in the receiver. Retransmission of lost packet video can not achieve real-time video quality and *Forward Error Correction* (FEC) techniques proved to be ineffective when bursty losses are encountered [56].

In this chapter, 64kbps bit rate *Pulse Code Modulation* (PCM) will be considered in for audio encoding and NLMS technique will be presented to adjust the playout delay of packet voice. Finally, the performance of NLMS will be compared with LMS technique and hence NLMS will be proposed for packet video playout in the chapter 5.

4.2 THE PACKET VOICE MODEL

To properly examine the delay sources in packet voice communications, the need to investigate the stages which a packet encounters from source to destination becomes very essential. At the source, a continuous voice signal is sampled at a typical rate of 8000Hz. Each sample is encoded with 8 bits. Compression is can also made in this stage if higher encoding bits are used to produce high fidelity voice with reasonable network frame size. Assuming constant encoding rate (64kbps) and constant size of produced voice frames and no compression, the encoding process introduces constant processing delay. The encoded voice frames are packetized and sent to the receiver through the network as shown in figure (4-1). Due to network queuing and congestion conditions, the delay a packet experiences as it being sent across the network becomes more variable. Before voice packets are played out by the receiver, the varying delays must be

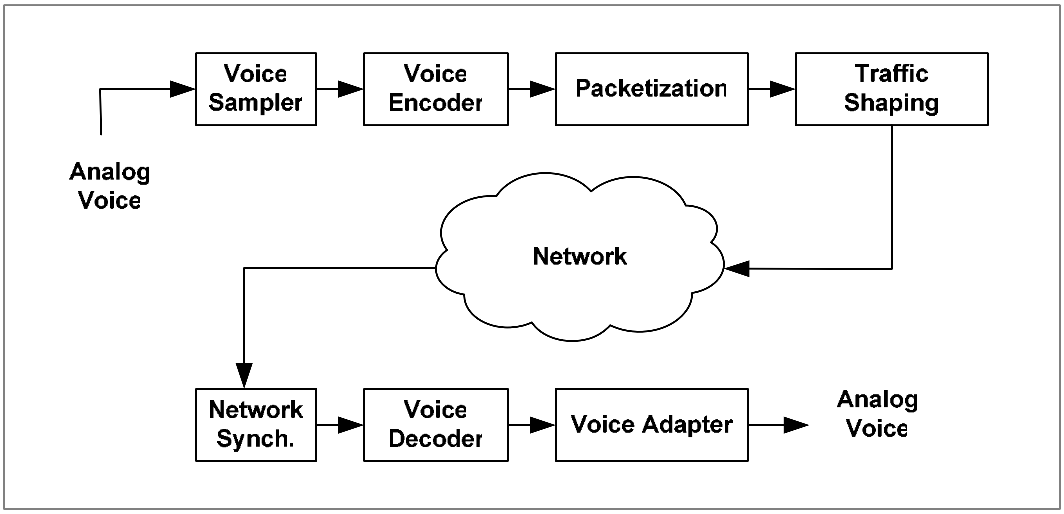

Figure (4-1): Packet voice model.

adjusted to obtain continuous playout of voice packets. The sum of delay components which contribute to most of the delays that a voice packet encounters is:

$$\Delta = \Delta p + \Delta pkt + \Delta n + \Delta p_r \qquad (4\text{-}1)$$

where:

- Δp is the encoder and decoder processing delay.
- Δpkt is the packetization and depacketization delays.
- Δn is the network delay
- Δp_r is the propagation delay.

In general, a receiver that can estimate the proper amount of packet delay will be capable of playing out more real-time audio

4.3 ASSUMPTIONS FOR PACKET VOICE STREAMING

In packet voice communications, packet voice encounters variable delays mainly due to network queuing Δn. Also, due to the small size and uniform length of the PCM voice frames, the delays resulted from processing Δp, packetization Δpkt and propagation Δp_r will be considered constant. In the simulation, the following will be assumed:

1) The continuous voice signal is sampled at rate of 8000Hz.
2) 8 bits per sample encoding with no compression.
3) The encoding rate is 64kbps.
4) The encoder delay Δp is constant with 0.125 ms.
5) Constant delay of 12 ms for each of packetization and depacketization.
6) Varying network delay is geometrically distributed with maximum and minimum delays of 120 ms and 20 ms per packet respectively.
7) Constant propagation and traffic shaping delays of 25 ms.

Using the assumptions above, both LMS and NLMS algorithms will be used to estimate the proper amount of delay $\Delta_{Adaptive}$ required to adjust the total end-to-end delay of each packet. Using equations (4-2) & (4-3), the adaptively predicted buffering delay $\Delta_{Adaptive}$ is used to synchronize the decoded voice frame just before playout as shown in Figure (4-2).

$$\hat{v}(i) = \alpha.\hat{v}(i-1) + (1-\alpha).|\Delta_{Adaptive} - n(i)| \tag{4-2}$$

$$D(i) = \Delta_{Adaptive} + \beta.\hat{v}(i) \tag{4-3}$$

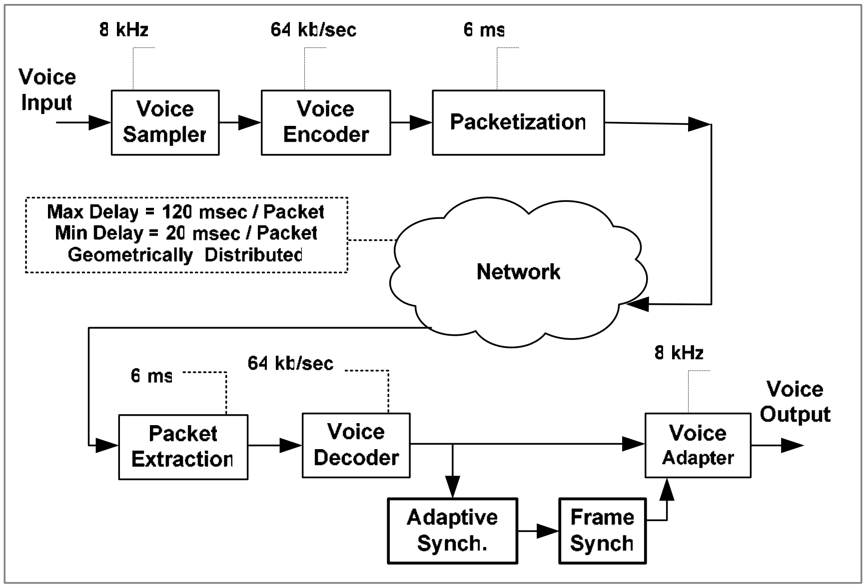

Figure (4-2): Proposed adaptive scheduler for packet voice.

4.4 SIMULATION AND RESULTS FOR VOICE PACKET USING GEOMETRIC DISTRIBUTION

The PCM voice encoder processes each voice sample and produces uniform frame lengths of 8 bits. With constant encoder bit rate of 64kbps and compression ratio of 1 (no compression), the produced voice frame will experience 0.125 ms delay. The packetization process of voice adds constant processing delay of 6.0 ms per packet as shown in figure (4-3). The total delay a packet encounters just before transmission is 6.125 ms. The inter-packet delay of the network has a mean of 50 ms. We assume that generated inter-packet delays contain both propagation and network variable delay components. Any clock drifts between the sender and receiver will be neglected. The geometric

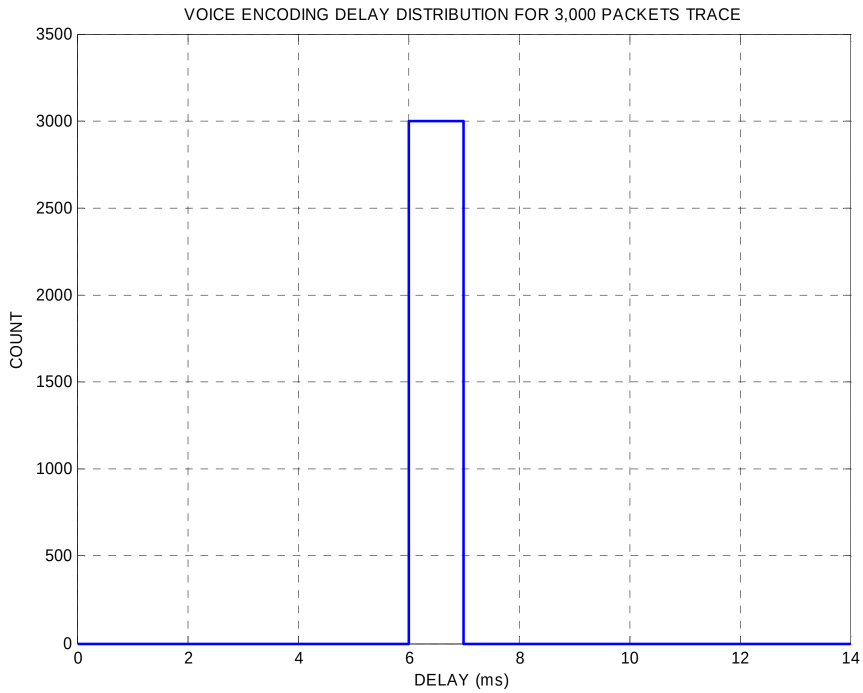

Figure (4-3): Voice codec and packetization delay distribution.

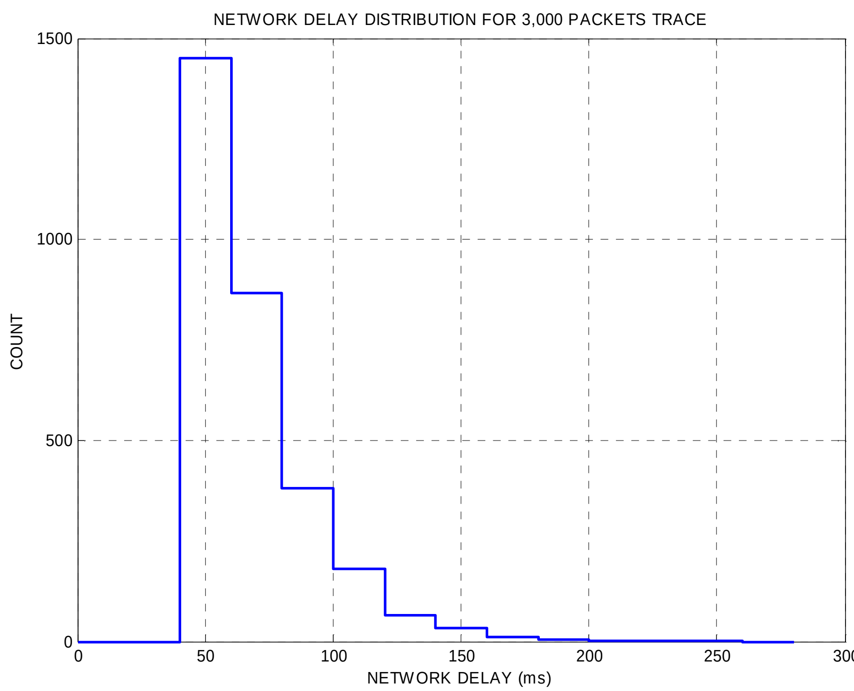

Figure (4-4): Network delay distribution.

distribution pattern was generated via MATLAB georand function. The distribution of network inter-packet delay is depicted in figure (4-4).

The total encountered delay distribution per packet is shown in figure (4-5). The delay distribution has a mean of 70 ms and includes the total sum sender and receiver coding delays, packetization and packet extraction delays, propagation delays and network queuing delays as in equation (4-2). The playout buffering delay at the receiver should be adjusted according to the amount of delay each packet experienced.

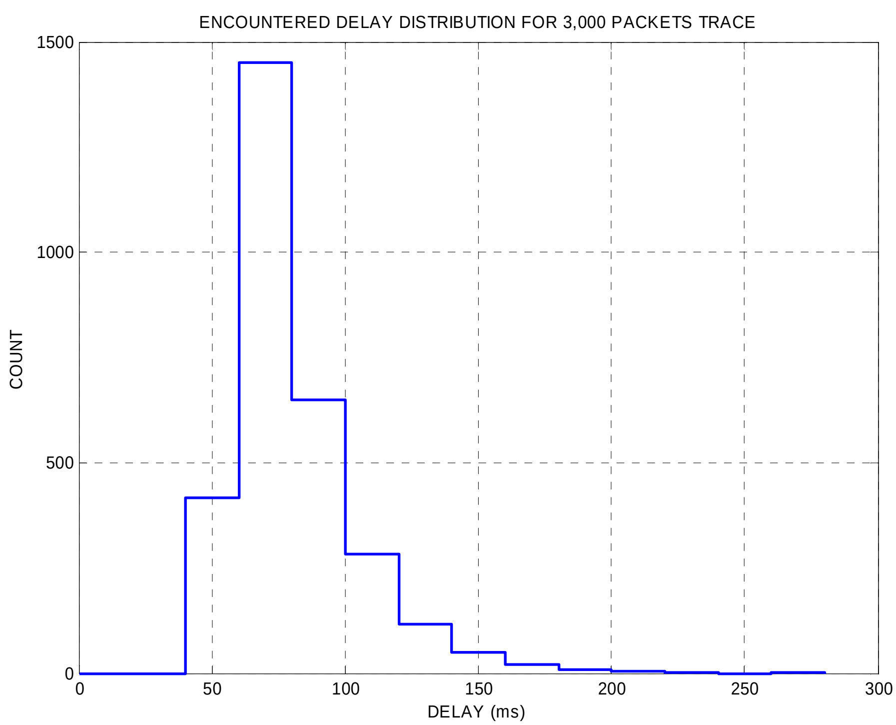

Figure (4-5): Total encountered delay distribution.

The total encountered delays are applied to the LMS and NLSM algorithms to predict the next delay. Both algorithm filters were selected to have 11 weights tap with initial weight of $\vec{w}(i) = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$. The choice of the FIR filter tap was selected based on the NLMS convergence results for 9, 11, and 15 taps selections as shown in figure (4-6). The delays matrix has 3000 inter-packet delays to be processed by each algorithm with step size μ =0.95. Also, the choice of step size was selected based on the NLMS convergence results for different step size selections as shown in figure (4-7) The values $\alpha = 0.998002$ and $\beta = 0.5$ will be used to estimated delay in equations (4-2) and (4-3). The averaged mean-squared error between the actual and estimated delays for LMS and NLMS algorithms is shown in figures (4-8). The figure shows that NLMS performs better than LMS as it converge faster to the lowest level of mean-squared error.

The total tradeoff between end-to-end delay and packet loss is shown in figure (4-9) for autoregressive, LMS and NLMS approaches. The average end-to-end delay and the corresponding packet loss were obtained for different values of β ranging from 0.3 to 1.0. It was found that NLMS approach outperforms autoregressive and LMS algorithms by reducing both the average end-to-end delay and packet loss rate.

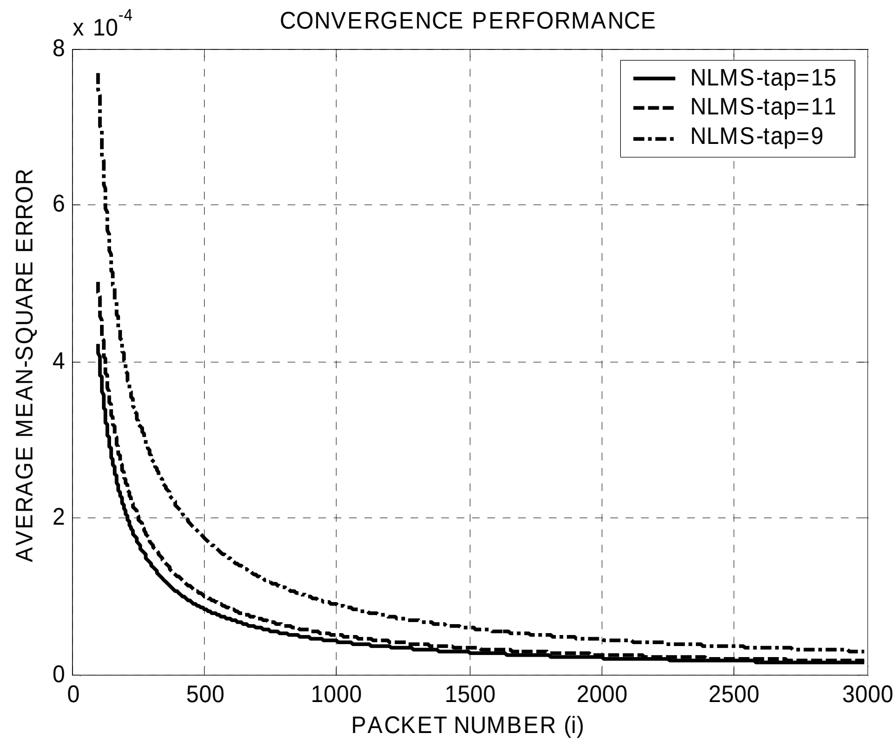

Figure (4-6): MSE for different filter taps in NLMS.

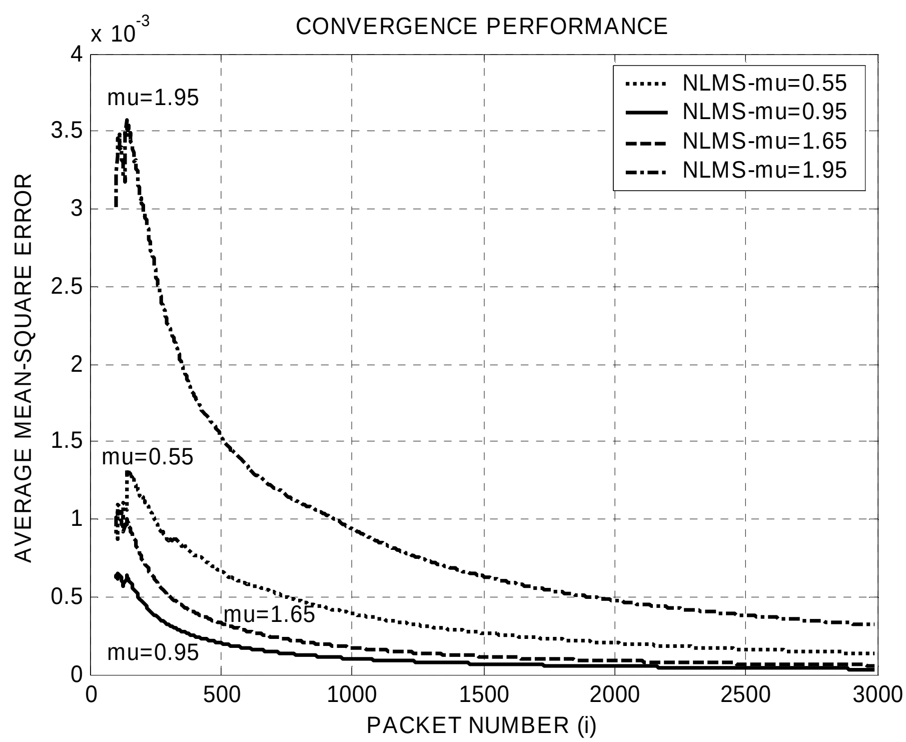

Figure (4-7): MSE for different step-size in NLMS.

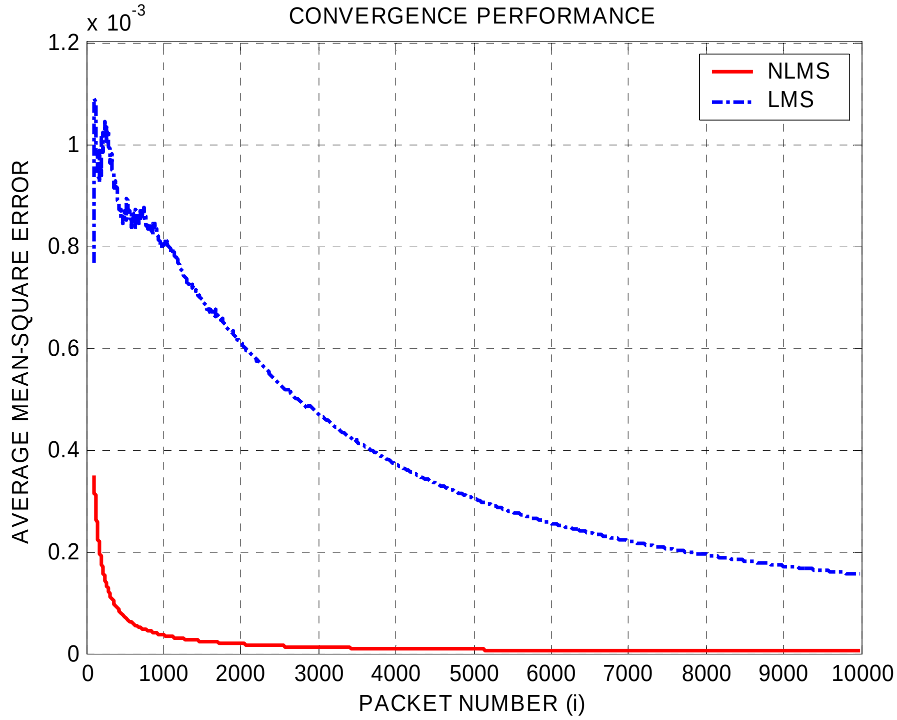

Figure (4-8): MSE between the actual and the NLMS estimated delays.

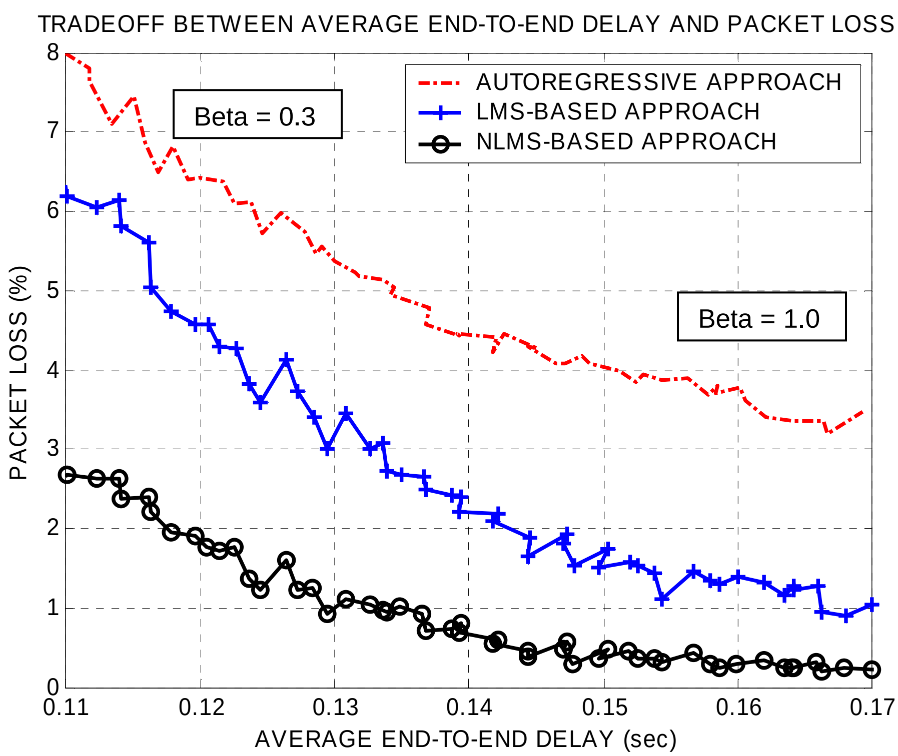

Figure (4-9): The total tradeoff between end-to-end delay and packet loss.

4.5 SIMULATION AND RESULTS FOR VOICE PACKET USING PARETO DISTRIBUTION

In this section, we apply *Pareto distribution* on the network delay to examine and compare the performance of autoregressive, LMS-based and NLMS-based algorithms on the end-to-end packet voice delay and packet loss rate. Similar to section 4.4, the PCM voice encoder processes each voice sample and produces uniform frame lengths of 8 bits. With constant encoder bit rate of 64kbps and compression ratio of 1 (no compression), the produced voice frame will experience 0.125 ms delay. The packetization process of voice adds constant processing delay of 6.0 ms per packet as shown in figure (4-10). The total delay a packet encounters just before transmission is 6.125 ms. The inter-packet delay of the network has a mean of 50 ms. We assume that generated inter-packet delays contain both propagation and network variable delay components. Any clock drifts between the sender and receiver will be neglected. The Pareto distribution pattern was generated via MATLAB *simpareto* function. The distribution of network inter-packet delay is depicted in figure (4-11).

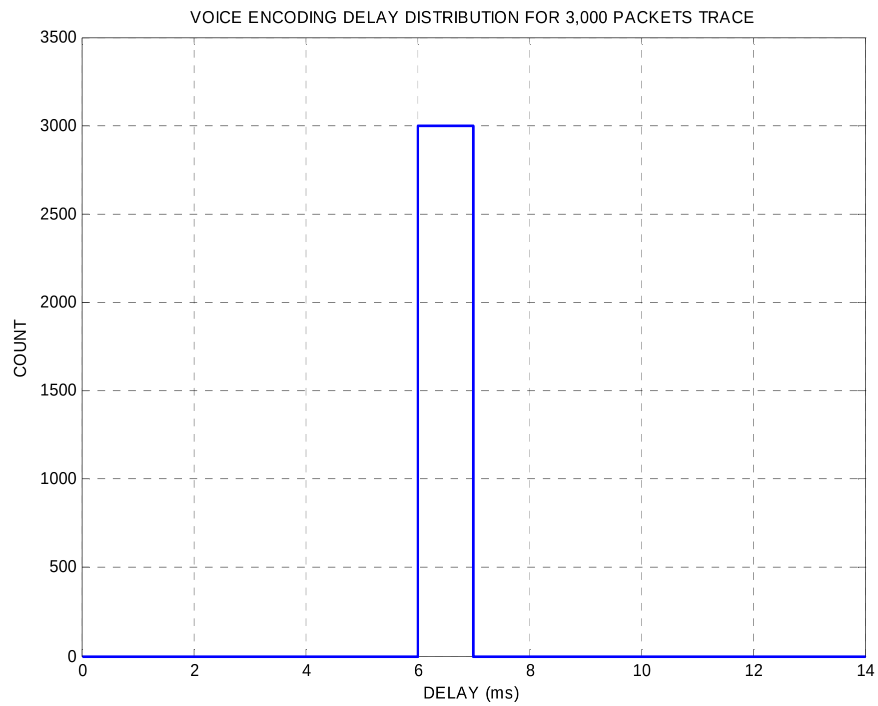

Figure (4-10): Voice codec and packetization delay distribution.

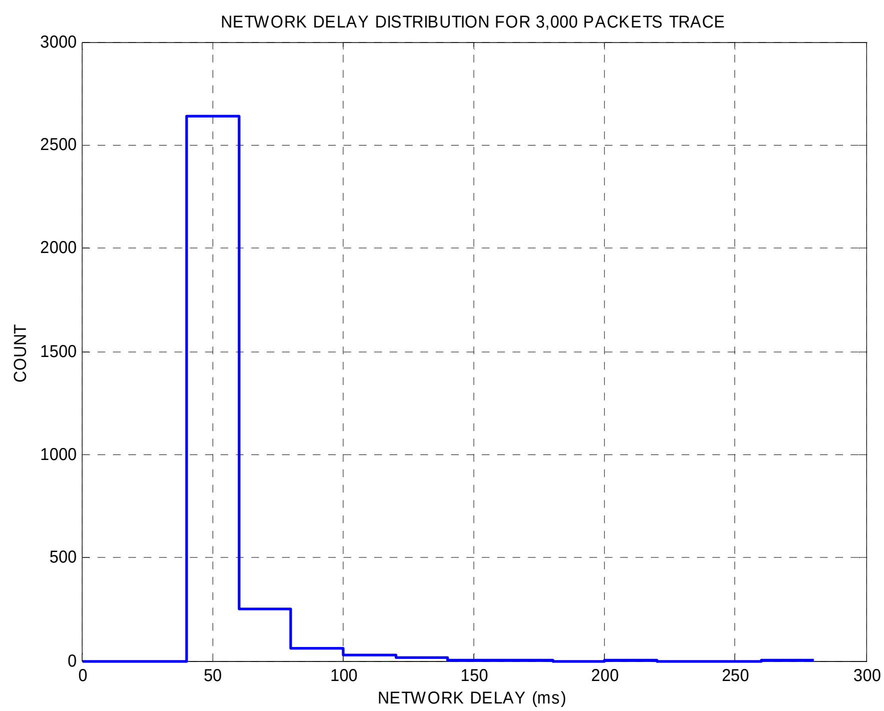

Figure (4-11): Network delay distribution - Pareto.

The total encountered delay distribution per packet is shown in figure (4-12). The delay distribution has a mean of 70 ms and includes the total sum sender and receiver coding delays, packetization and packet extraction delays, propagation delays and network queuing delays as in equation (4-2). The playout buffering delay at the receiver should be adjusted according to the amount of delay each packet experienced.

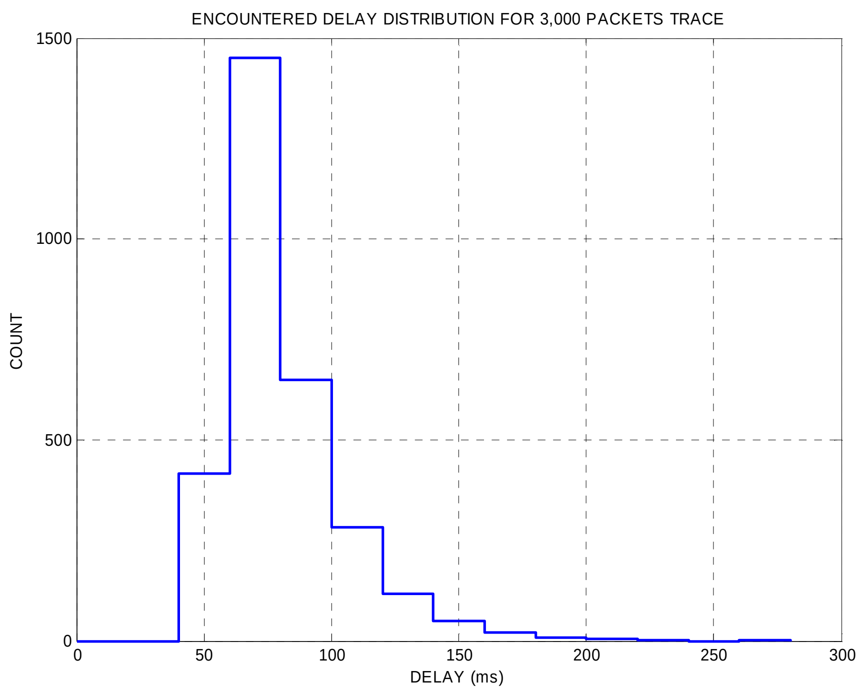

Figure (4-12): Total encountered delay distribution.

The total encountered delays are applied to the LMS and NLSM algorithms to predict the next delay. As described in section 4.4, adaptive filter were selected to have the following parameters:

- 11 weights tap with initial weight of $\vec{w}(i) = [1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0]$.
- The delays matrix has 3000 inter-packet delays.
- Step size μ =0.95.
- The values $\alpha = 0.998002$ and $\beta = 0.5$ will be used to estimated delay in equations (4-2) and (4-3).

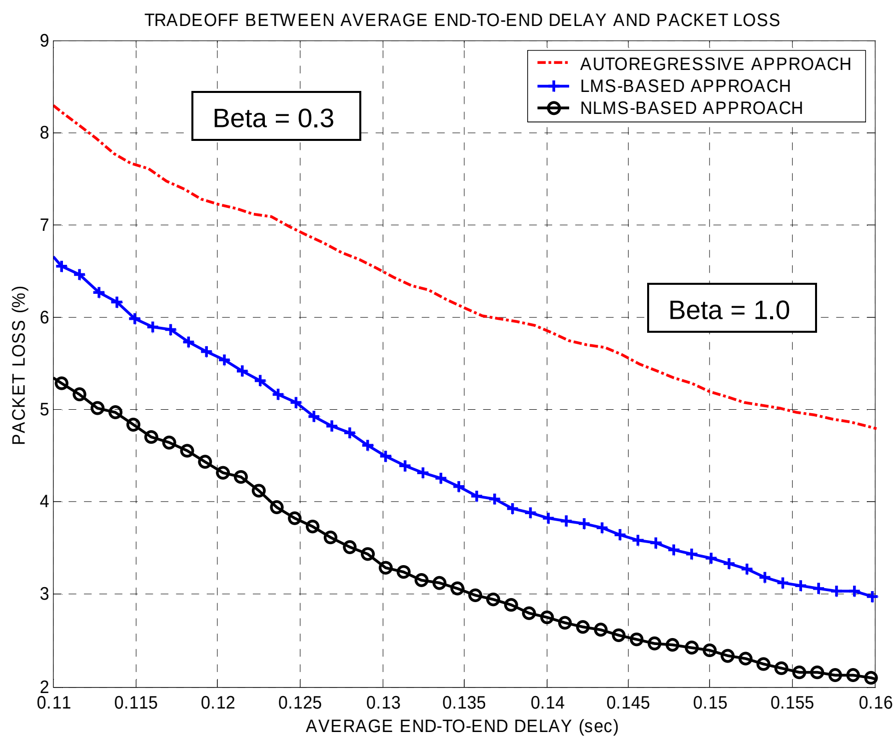

Figure (4-13): The total tradeoff between end-to-end delay and packet loss.

The total tradeoff between end-to-end delay and packet loss is shown in figure (4-13) for autoregressive, LMS and NLMS approaches. The average end-to-end delay and the corresponding packet loss were obtained for different values of

β ranging from 0.3 to 1.0. As shown in figure (4-13), lower values of β produce lower average end-to-end delays whereas higher values of β produce higher average end-to-end delays. It was found that NLMS approach outperforms autoregressive and LMS algorithms by reducing both the average end-to-end delay and packet loss rate.

CHAPTER 5

APPLYING NLMS ALGORITHM TO PACKET VIDEO

5.1 THE PACKET VIDEO COMMUNICATIONS DELAYS

Achieving low latency is of utmost importance to enhance the real time quality in both audio and video applications over the internet. Internet video communication, in particular, is bandwidth intensive and subject to packet delay, and packet loss.

Several studies have been introduced to improve the applications of internet video communications such as prerecorded video materials, live events and video conferencing. Because the amount of motion or changes in the picture in each situation is different, video encoders produce varying sizes of video frames. For example, broadcasting live sport event produces larger frames compared with a videoconferencing. This is because the amount of motions in a sport event is much higher than those in a conference room where a still picture dominates the small movements.

Large production of video frames, in general, results in higher video rates which can lead to network congestion especially when exceeding the network bandwidth capacity. This may adversely affect other *transmission control protocol* (TCP) traffic which reduce transmission rate as a reaction to the network congestion. Thus, network-friendly packet video streams are to be obtained by encoders to fit the network traffic conditions.

Varying delay can negatively impact the perceived quality of packet video streams. Good quality video frames require bits compression and extensive encoding and decoding processes at the transmitter and the receiver. The codec processing delay varies according to the size of each video frame. As a result, the total end-to-end delay experienced by packet video is affected by the varying delays in the codec as well as varying network delays.

Another challenge in internet video communications is packet loss which may results in video error. Lack of error adjustment introduces error propagation and hence gives time-varying quality of real-time video. Also, retransmission of lost packet video can't achieve real-time video quality and *Forward Error Correction* (FEC) techniques proved to be ineffective when bursty losses are encountered.

In this chapter, variable bit rate MPEG coding will be considered in videoconferencing systems and NLMS technique will be proposed to adjust the playout delay of packet video by responding to changing network delays.

5.2 THE PACKET VIDEO MODEL

To properly examine the delay sources in packetized video communications, the need to investigate the stages which a packet video encounters from source to destination becomes of great importance.

At the source, continuous video is sampled into images. The sampling process is performed by an image sampler at a typical rate of 5 to 30 images per second. In general, higher sampling rates allows for shorter image production delays. The resulted samples of images are encoded and compressed so that quality video with reasonable network frame size is produced. Assuming constant encoding rate, the encoding and compression processes introduce variable processing delay due to varying size of produced video frames. The encoded video frames

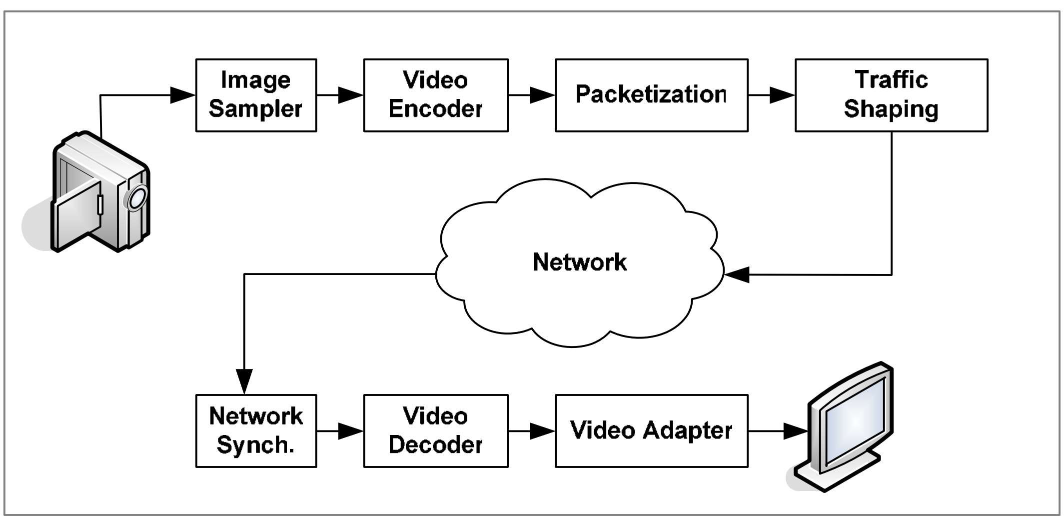

Figure (5-1): Packet video model.

are packetized and sent to the receiver through the network. Due to network queuing and congestion conditions, the delay a packet experiences as it being sent across the network becomes more variable. Before video images are displayed at the receiver, the varying delays must be adjusted to obtain continuous playout of video packets. The sum of delay components which contribute to most of the delays that a video packet encounters must be synchronized at the receiver are:

$$\Delta = \Delta p + \Delta pkt + \Delta n + \Delta p_r \qquad (5\text{-}1)$$

where:

- Δp is the encoder and decoder processing delay.
- Δpkt is the packetization & packet extraction delay.
- Δn is the network delay
- Δp_r is the propagation delay.

In general, a receiver that can estimate the proper amount of packet with lower end-to-end delay will be capable of displaying more real-time video over packet switched networks such as the internet.

5.3 MPEG OVERVIEW

MPEG is a video coding standard designed for digital storage of quality video for later playout. The MPEG system encodes a sequence of digitized images to produce two different types of coding format, namely Intra-frame and Predictive-frame coding.

During the Intra-frame coding, the images are processed to eliminate the spatial redundancy. The image is further divided into 8x8 pixel blocks and processed by *discrete cosine transform* (DCT). The resulted 8x8 matrixes of coefficients are quantized, Huffman encoded and processed for further compression. The resulted encoded image is referred to as I-frame.

During the Predictive-frame coding, an image is processed to eliminate the temporal redundancy relative to the previous one. The image is further divided into macro-blocks each of 16x16 pixel matrixes of luminous information and two 8x8 pixel blocks of the two chrominance components. Motion estimation is performed for each macro-block such that previous image is searched for similar macro-block. If similar macro-block is found, the difference is calculated between the actual and the reference macro-blocks. The difference is then coded by performing DCT, quantization and Huffman encoded. The resulted encoded image is referred to as P-frame. If no similarity is found, each block is treated like a block in an I-frame.

The produced P-frame is typically 2 to 4 times smaller than I-frame as shown in figure (5-2). Due to the size differences between I-frame and P-frame, the encoder will produce variable bit rate streams especially when fast motions are detected. However, the predictive coding provides more compression during slow motion scenes. Thus, using MPEG coding in video conferencing applications, where still cameras are mounted at fixed locations with relatively small encountered movements, it is feasible to obtain lower sizes for packet video communications over the internet.

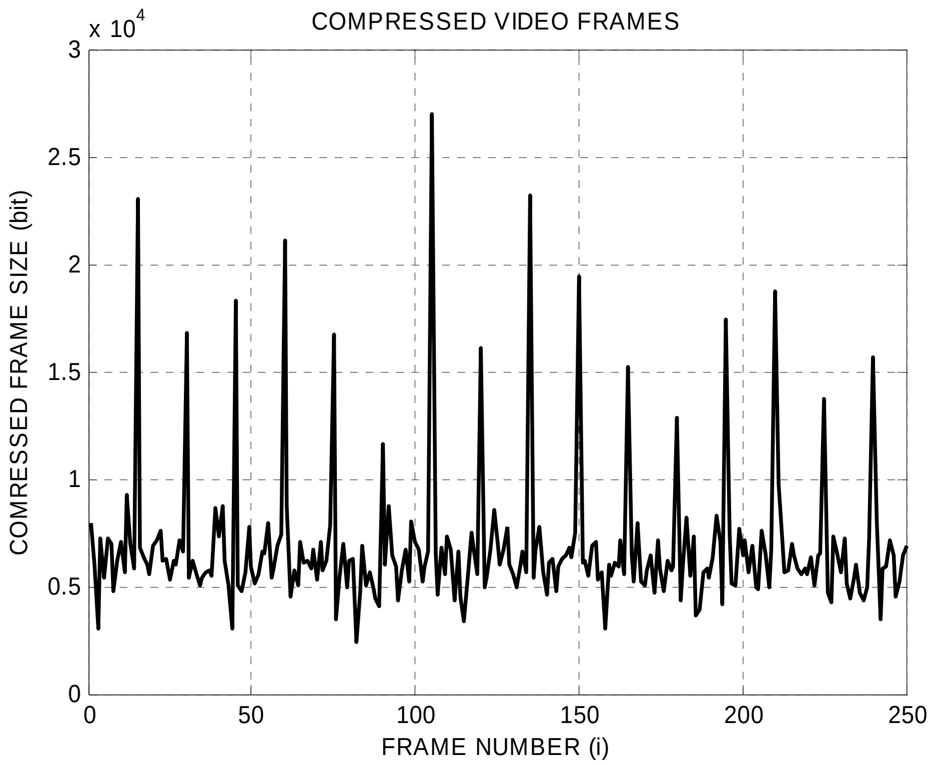

Figure (5-2): P-frames have larger size compared with I-frames.

5.4 CBR AND VBR VIDEO STREAMS

With constant codec rates, the amount of bits needed to encode each image is variable and hence the production rate of encoded video frames is also variable. To minimize the variable production rate, a smoothing buffer is used so that video bits are sent to the network at a *constant bit rate* (CBR). To prevent the buffer from overflow and underflow, a control mechanism is used to monitor the utilized size of the buffer and adjust the encoder production rate accordingly. This buffering process adds sensible delay that must be compensated at the receiver and hence adversely affect the total end-to-end delay. CBR buffer flow control is shown in figure (5-3).

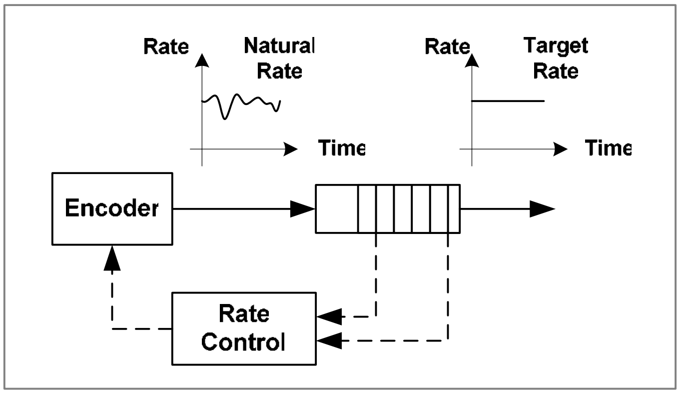

Figure (5-3): CBR buffer flow control.

In *variable bit rate* (VBR) encoders, the packet is sent through the network once there are enough bits to assemble a packet. While the buffering delays are minimized, the packets will still experience the network variable queuing delays. In this case, the total encountered delay can be give by

$$\Delta = \Delta p_{VBR} + \Delta pkt + \Delta n + \Delta p_r \qquad (5\text{-}2)$$

where:

- Δp_{VBR} is the VBR codec/compression processing delay.

Δp_{VBR} can be reduced by using small packets which are decoded as soon as they arrive to the decoder. In VBR video streams, the total end-to-end delay is dominated by Δn [reference].

Source burstiness, can lead to network congestion and packet losses and hence decreases the real-time video quality. To reduce the effects of bursty sources, traffic shaping is exploited at the network boundary to assure certain bandwidth (B) to the video source such that the burstiness is kept below a predefined value (A) as shown in figure (5-4).

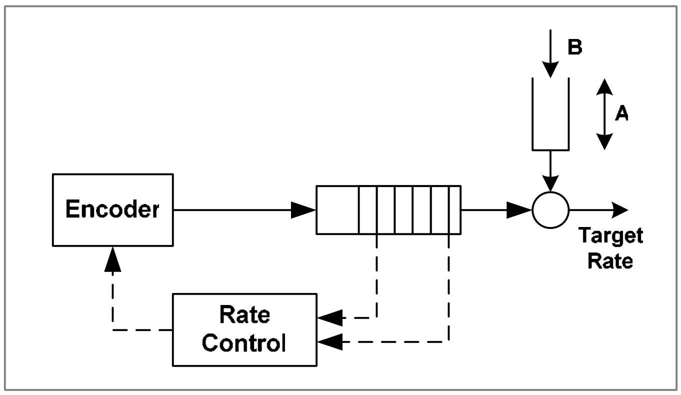

Figure (5-4): Traffic shaping.

Since the traffic pattern generated by VBR encoder is not known in advance, the traffic shaper may not be compatible with packet produced by the encoder such that the produced video packet is larger than the maximum burst size of the traffic shaper. As a result, the packet will be discarded by traffic shaper.

To serve the purpose of this work, the analysis of traffic shaping will be left for further study and we will assume that the VBR encoder does not produce packets larger than the maximum burst value (A). We will only consider the traffic shaping contribution on the total end-to-end delay. Therefore, equation (5-2) can be modified to include the traffic shaping delay Δ_{TS} as

$$\Delta = \Delta p_{VBR} + \Delta pkt + \Delta_{TS} + \Delta n + \Delta p_r \qquad (5\text{-}3)$$

In this way, the NLMS algorithm shall estimate the packet encountered delay $\Delta = \Delta_{NLMS}$ and adjust the playout buffering delay according to the estimated value of Δ_{NLMS} .

5.5 NLMS ALGORITHM FOR PACKET VIDEO STREAMING

In packet video communications, packet video encounters variable delays as a result from video codec processing Δp_{VBR} and network queuing Δn. In the simulation, the following will be assumed:

1. Video images are captured from video conferencing system with fixed camera and small amount of movements in the location.
2. The video sampler produces images with rate of 15 images per second.
3. The video encoder produces tolerable sequences of I-frames and P-frames with predefined maximum frame size. This assumption is feasible in video conferencing systems due to the small pace of camera movement. The encoding rate is 500 kb per second with compression ration of 150. The encoder delay Δp_{VBR} varies according to the frame size. VBR video is also assumed and hence the buffering delay of the encoded frames is negligible.
4. Constant delay of 30 ms for each of packetization, traffic shaping and packet extraction processes.
5. Varying network delay is geometrically distributed with maximum and minimum delays of 120 ms and 20 ms per packet respectively.
6. Constant propagation delay of 24 ms.

The NLMS algorithm intends to estimate the proper amount of delay Δ_{NLMS} required to adjust the total end-to-end delay of each packet video such that the packet loss is minimized using equations (5-4) and (5-5) as described in chapter 3. The predicted delay is used to synchronize the decoded frame just before video playout as shown in Figure (5-5).

$$\hat{v}(i) = \alpha.\hat{v}(i-1) + (1-\alpha).|\Delta_{NLMS} - n(i)| \tag{5-4}$$

$$D(i) = \Delta_{NLMS} + \beta.\hat{v}(i) \tag{5-5}$$

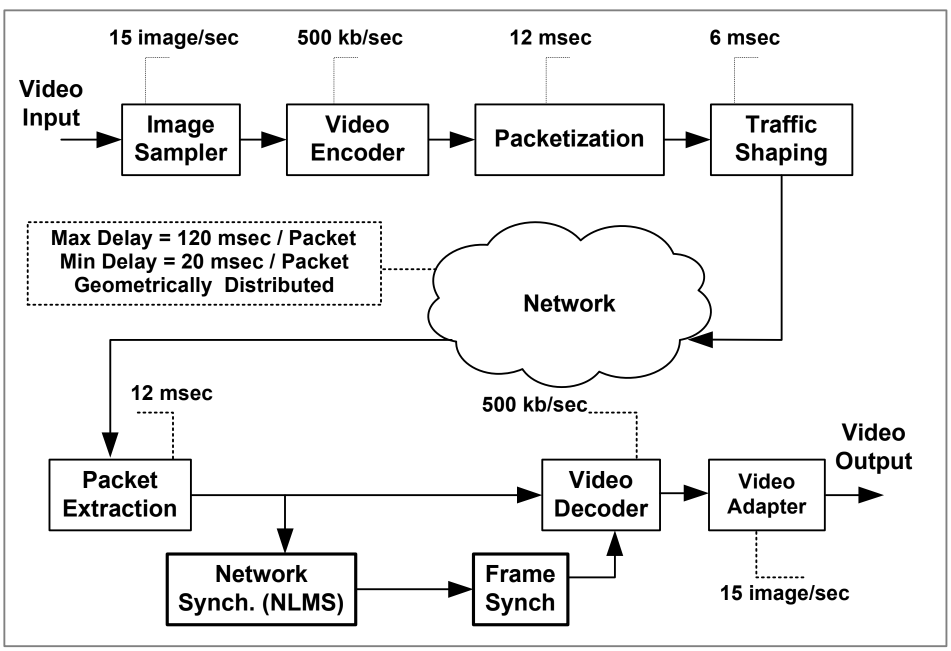

Figure (5-5): Proposed NLMS scheduler for Packet Video.

5.6 SIMULATION AND RESULTS FOR PACKET VIDEO USING GEOMETRIC DISTRIBUTION

The video encoder processes each image sample and produces I-frames and P-frames. As described in section 5.3, I-frames is 2 to 4 times higher in size than P-frames. Assuming constant encoder bit rate of 500kbps and compression ration of 150 with 15 images per sample, the produced frames varies in size based on the image sample. In turn, this will result in variable video frame delay with mean of 25 ms shown in figure (5-6) where the high counts represent the delays of I-frames. The delay distribution also assumes small movements in the generated image samples (small P-frames) which is typically achievable in videoconferencing situations.

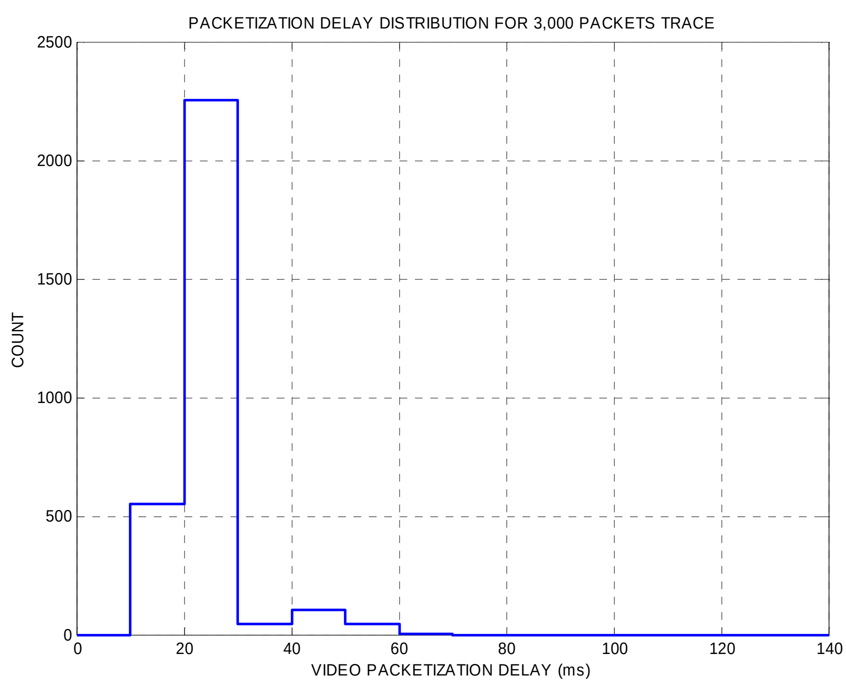

Figure (5-6): Video codec/compression frame delay.

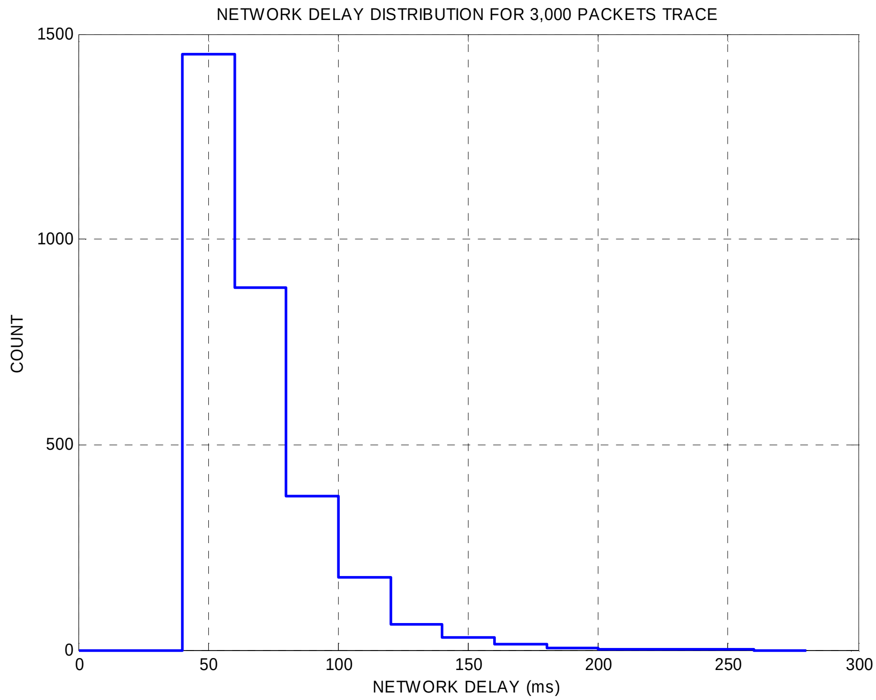

Figure (5-7): Variable distribution of Network delay.

The inter-packet delay of the network has a mean of 50 ms. We assume that generated inter-packet delays contain propagation time and network variable delay components. Any clock drifts between the sender and receiver will be neglected. The geometric distribution pattern was generated via MATLAB geornd function. The distribution of network inter-packet delay is shown in figure (5-7).

The total encountered delay distribution per packet is shown in figure (5-8). The delay pattern includes the total sum sender and receiver codec delays, packetization and depacketization delays, traffic shaping delays, propagation delays and network queuing delays as in equation (5-3). The playout buffer at the receiver should be according to the delay each packet experienced.

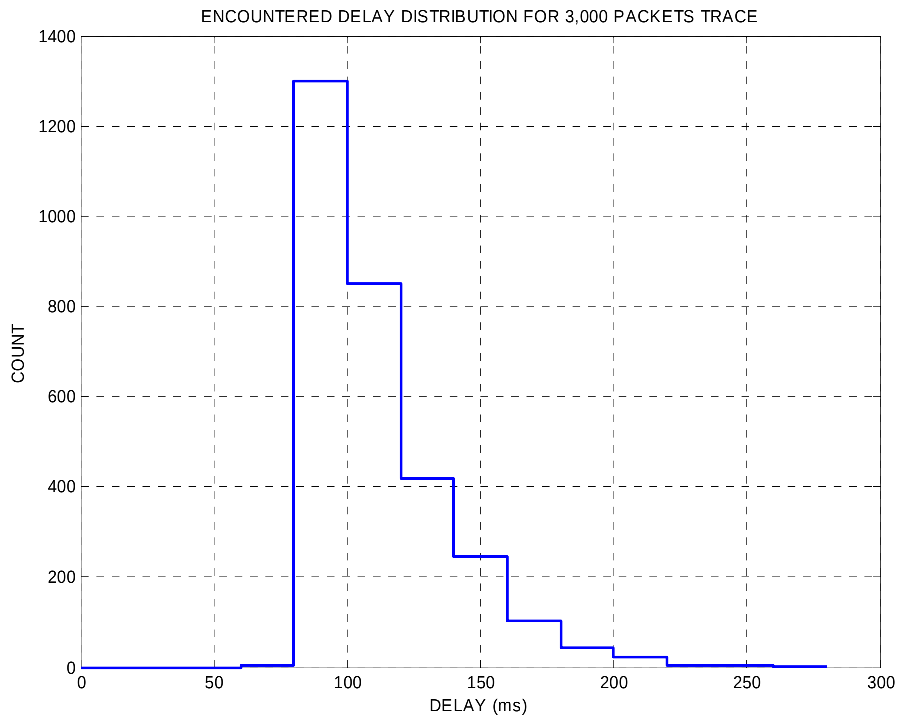

Figure (5-8): Total encountered delay distribution.

The total encountered delays are applied to the NLSM filter to predict the next delay as described in chapter 3. The NLMS FIR filter was selected to have 11 weights tap with initial weight of $\vec{w}(i) = [1\,0\,0\;0\,0\,0\,0\;0\,0\,0\,0]$. The delays matrix has 5000 inter-packet delays to be processed by NLMS algorithm with scale factor $\tilde{\mu}$ = 0.95. The values $\alpha = 0.998002$ and $\beta = 0.5$ will be used to estimated delay in equations (5-4) and (5-5). The mean-squared error between the actual and estimated delays is depicted in figures (5-9).

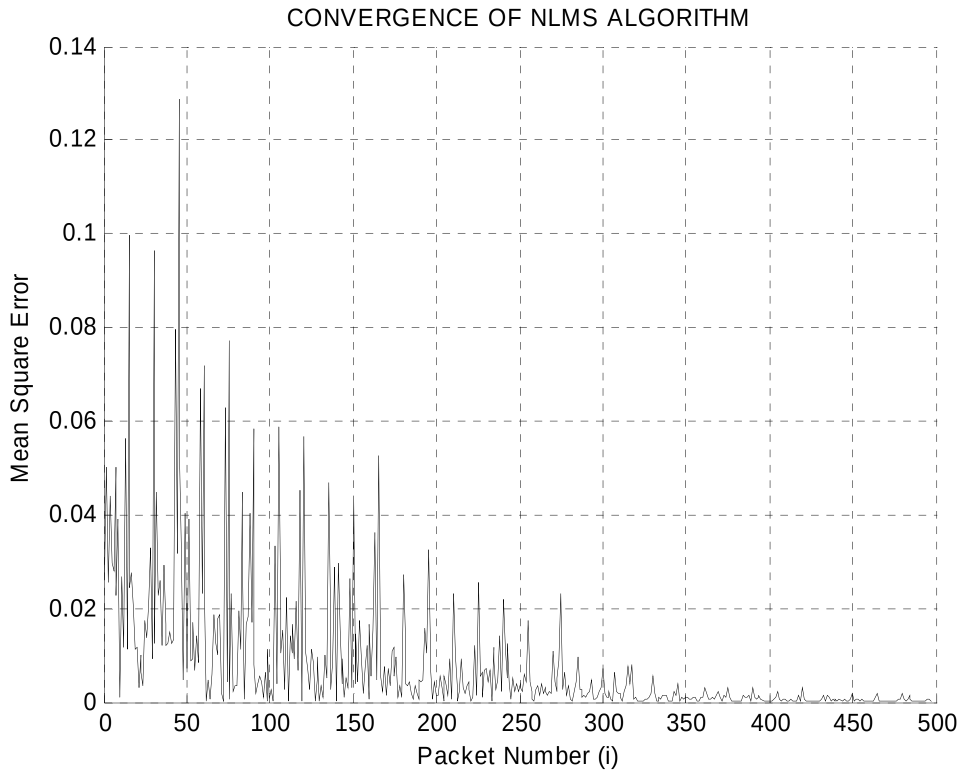

Figure (5-9): MSE between the actual and the NLMS estimated delays.

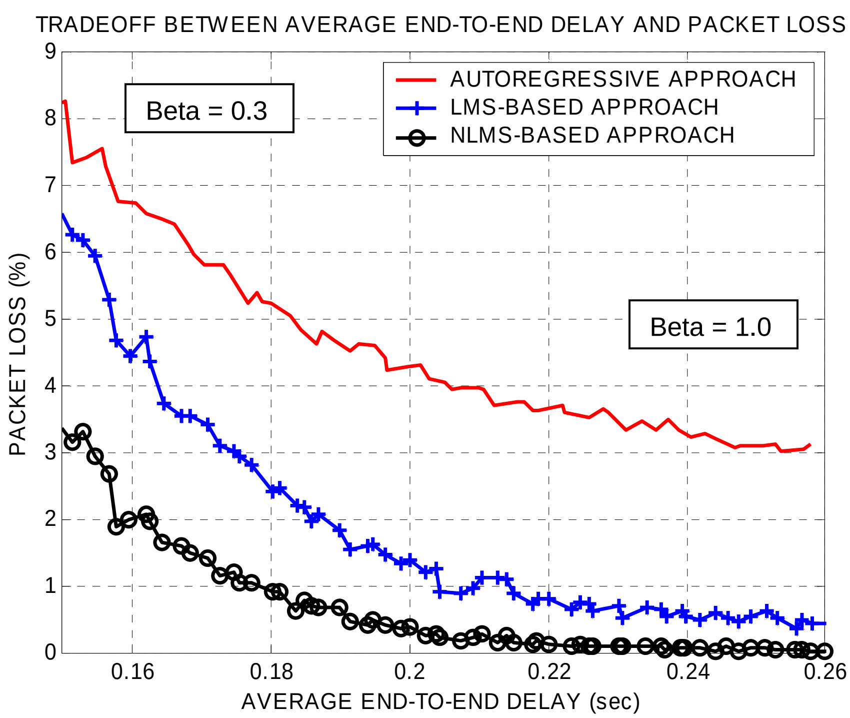

Figure (5-10): The total tradeoff between end-to-end delay and packet loss.

The total tradeoff between end-to-end delay and packet loss is shown in figure (5-10) for autoregressive, LMS and NLMS approaches. The average end-to-end delay and the corresponding packet loss were obtained for different values of β ranging from 0.3 to 1.0. It was found that NLMS approach outperforms both AR and LMS algorithms by reducing both the average end-to-end delay and packet loss rate.

5.7 SIMULATION AND RESULTS FOR PACKET VIDEO USING PARETO DISTRIBUTION

In this section, we apply *Pareto distribution* on the network delay to examine and compare the performance of autoregressive, LMS-based and NLMS-based algorithms on the end-to-end packet video delay and packet loss rate. Similar to section 5.6, the video encoder processes each image sample and produces I-frames and P-frames. I-frames are 2 to 4 times higher in size than P-frames. Assuming constant encoder bit rate of 500kbps and compression ration of 150 with 15 images per sample, the produced frames varies in size based on the image sample. In turn, this will result in variable video frame delay with mean of 25 ms shown in figure (5-11) where the high counts represent the delays of I-frames. The delay distribution also assumes small movements in the generated image samples (small P-frames) which is typically achievable in videoconferencing situations.

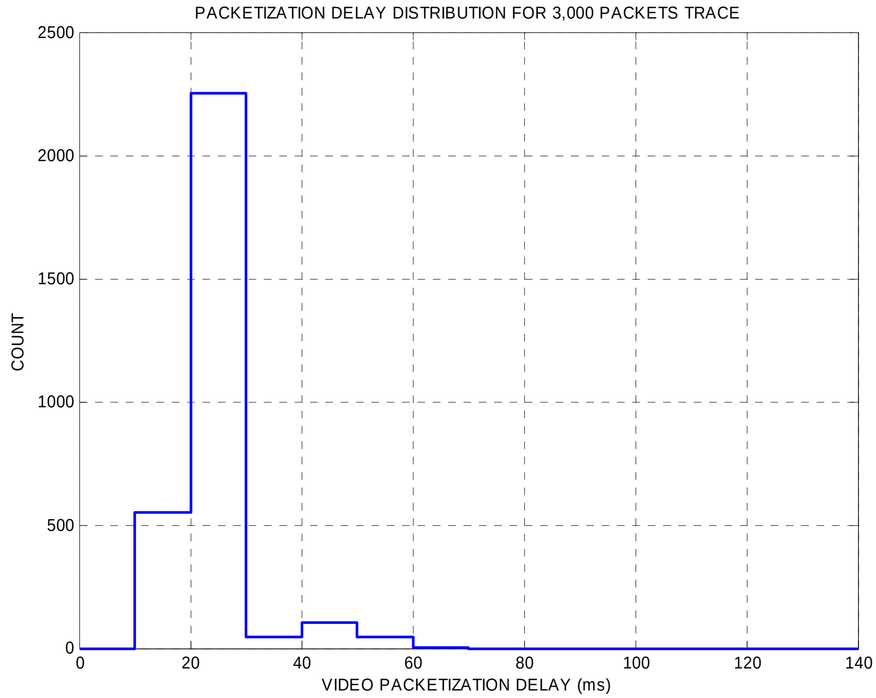

Figure (5-11): Video codec/compression frame delay.

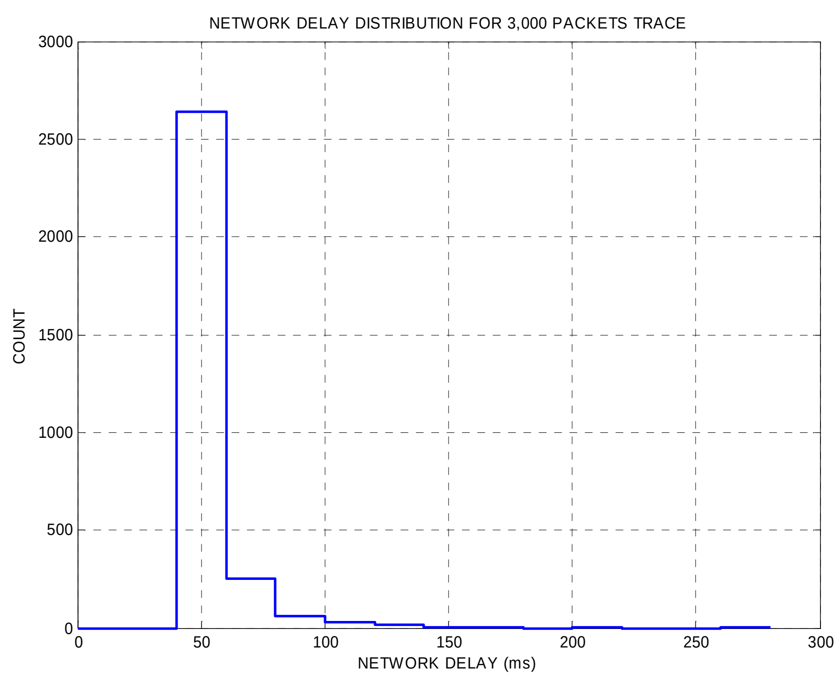

Figure (5-12): Variable distribution of Network delay – Pareto.

The inter-packet delay of the network has a mean of 50 ms. We assumed that the generated inter-packet delays contain propagation time and network variable delay components. Any clock drifts between the sender and receiver will be neglected. The Pareto distribution pattern was generated via MATLAB *simpareto* function. The distribution of network inter-packet delay is shown in figure (5-12).

The total encountered delay distribution per packet is shown in figure (5-13). The delay pattern includes the total sum sender and receiver codec delays, packetization and depacketization delays, traffic shaping delays, propagation delays and network queuing delays as in equation (5-3). The playout buffer at the receiver should be according to the delay each packet experienced.

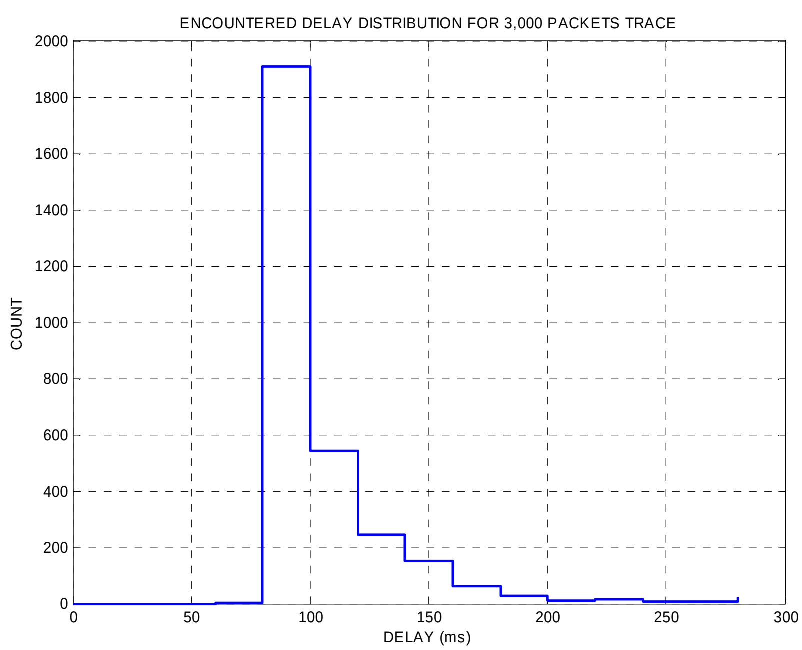

Figure (5-13): Total encountered delay distribution.

The total encountered delays are applied to the NLSM filter to predict the next delay as described in chapter 3. The NLMS FIR filter was selected to have:

- 11 weights tap with initial weight of $\vec{w}(i) = [1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0]$.
- The delays matrix has 5000 inter-packet delays.
- scale factor $\tilde{\mu}$ = 0.95.
- The values $\alpha = 0.998002$ and $\beta = 0.5$ will be used to estimated delay in equations (5-4) and (5-5).

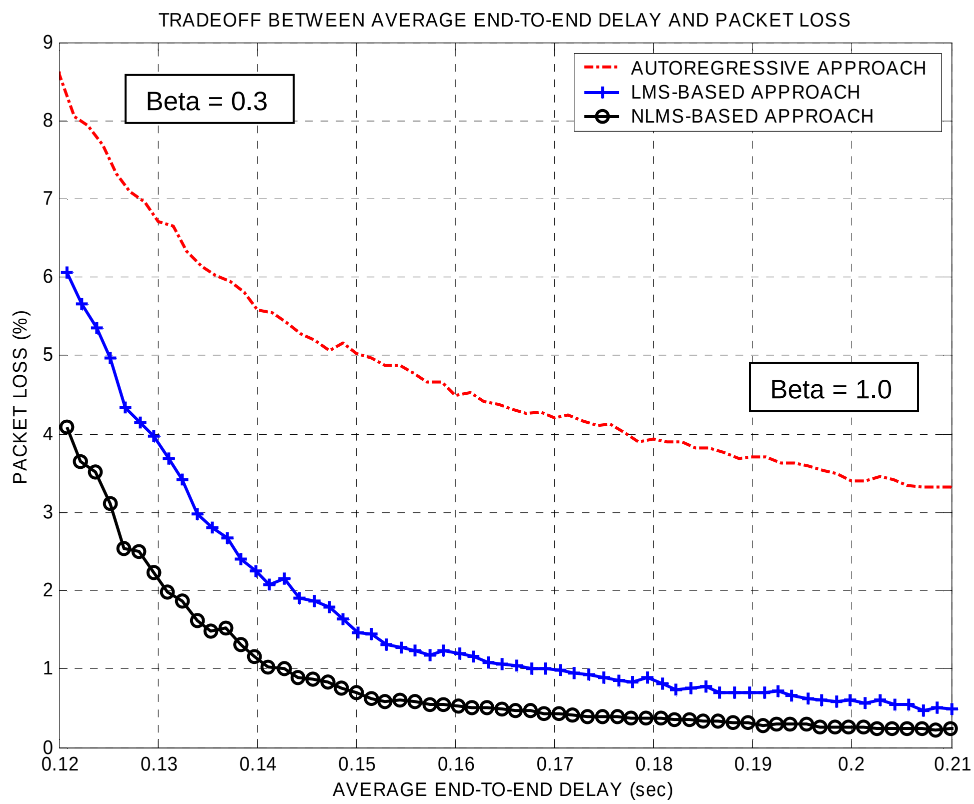

Figure (5-14): The total tradeoff between end-to-end delay and packet loss.

The total tradeoff between end-to-end delay and packet loss is shown in figure (5-14) for autoregressive, LMS and NLMS approaches. The average end-to-end delay and the corresponding packet loss were obtained for different values of β ranging from 0.3 to 1.0. As shown in figure (5-14), lower values of β produce lower average end-to-end delays whereas higher values of β produce higher average end-to-end delays. It was found that NLMS approach outperforms both AR and LMS algorithms by reducing both the average end-to-end delay and packet loss rate.

CHAPTER 6

SUMMERY, RESULTS AND FURTHER STUDY

In this work, the effects of packet delay and loss in packet multimedia networks were investigated. In chapter one, we started by introducing basic packet synchronization where we pointed out the relation between the components of the total delay and identified the fixed and variable delays.

In chapter two, we provided a background for packet voice and video delays where we described the constant and variable delay components. Also, we discussed the effects of variable delays in packet multimedia communications and indicated the delay smoothing procedure that eliminates the varying delays of received packets before playout. In addition, we described the packet playout adjustment and provided two examples of delay scheduling schemes where we found that small delay adjustment contributes less to the end-to-end delay on the expense of large packet loss rate whereas large delay adjustment contributes in less packet loss rate on expense of large end-to-end delay. We indicate that for packet multimedia communications, the receiver must be able to reduce the end-to-end delay and decrease the packet loss rate to achieve real-time packet multimedia communications. Furthermore, to tolerate between the end-to-end delay and packet loss rate, we reviewed related work on packet voice playout adjustment techniques and algorithms such as *adjustment between talkspurt*, *autoregressive delay estimation* and *adaptive delay estimation*.

In chapter three, NLMS adaptive algorithm was introduced as an adaptive playout mechanism to estimate the delay for packet multimedia communications. Mathematical analysis of NLMS algorithm and its vector interpretation were explored and the NLMS filter model was obtained. We showed that NLMS algorithm tends to minimize the error function between estimated and actual inter-packet delays. The NLMS estimates the next packet average delay by processing previously received delays through FIR filter. The error between the NLMS estimate and the actual delays is used to update the weight taps of a FIR filter such that the next error is minimized.

In chapter four, the packet voice delay estimation was considered. In packet voice model, we identified the variable and fixed delay components and formulized them to include codec/compression delays, packetization delays, propagation delays and network delays. NLMS algorithm was studied for packet voice communication assuming 8 kHz voice sampler, 64 kbps encoder rate, 12 ms fixed delay for packetization, fixed propagation delay of 24 ms, and geometrically distributed variable network of maximum and minimum values of 120 ms and 20 ms respectively. Finally, the NLMS simulation result was obtained and compared with the AR and LMS based algorithms where we showed that NLMS provides more enhanced delay adjustment mechanism with lower packet loss rate for small movement video application.

In chapter six, the packet video delay estimation in videoconferencing applications was considered. In packet video model, we identified the variable and fixed delay components and formulized them to include codec/compression delays, packetization delays, traffic shaping delays, propagation delays and network delays. We also provided an overview about MPEG encoding and discussed the production of I-frame and P-frames in the encoder where we identified the packet video delay in small movement situations such as videoconferencing. The CBR and VBR video frame production was introduced from the delay point of view where we selected VBR production as it introduces less end-to-end compared to CBR. In addition, the source burstiness in packet video was considered such assuming that it contributes in fixed amount of delay per packet. NLMS algorithm was studied for packet video communication assuming 15 images per second at the image sampler, 500 kbps encoder rate with compression ration of 150, 29 ms fixed delay for packetization and traffic shaping, fixed propagation delay of 24 ms, and geometrically distributed variable network of maximum and minimum values of 120 ms and 20 ms respectively. Finally, the NLMS simulation result was obtained and compared with the AR and LMS algorithms where we showed that NLMS provides more enhanced delay adjustment mechanism with lower packet loss rate for small movement video application.

In this work, we found that NLMS approach outperforms both AR and LMS approaches for both packet voice and video multimedia application.

Further work on this topic can be extended to study actual packet video performance on IEEE 802.11g wireless LAN environment which have recently gained popular utilization as a relatively inexpensive mean of deploying wireless packet switched network. A suggested scenario in this regard can involve the study of live broadcast pack video streaming from wireless multimedia server to wireless client station (video receiver) with several wireless stations generating certain data traffics on the network. The study can apply NLMS algorithm to examine real-time quality of video playout at different network load conditions.

References

[1] Stefano Bregni. “Synchronization of Digital Telecommunications Networks”. Wiley, 2002.

[2] James L. Massey. “Optimum Frame Synchronization”. IEEE Transaction on Communications, Com-20, No. 2, April 1972.

[3] Heinrich Meyer and Gerd Ascheid. “Synchronization in Digital Communications “. Volume 1. Wiley, 1990.

[4] W. A. Montgomery. “Techniques for Packet Voice Synchronization”. Selected Areas in Communications, IEEE Journal on. Volume 1, Issue 6, Dec 1983 Page(s):1022 - 1028

[5] Stefano Bregni. “A Historical Perspective On Telecommunications Network Synchronization”. IEEE Communications Magazine, June 1998.

[6] Mark Minasi and Todd Lammle, “TCP/IP for NT Server 4 Study Guide”. Sybex press, Calif. 1997.

[7] John Bellamy, “Digital Telephony”. Wiley Interscience, NY: 2000.

[8] CISCO Documentation. “Designing a Long-Distance VoIP Network”. www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/voipsol/longd.htm

[9] Alberto Leon-Garcia and Indra Widjaja, "Communication Networks", McGraw-Hill, 2 edition, 2003.

[10] Juniper Networks Documentations. “Voice over IP Solutions”. Sean Christensen.
www.juniper.net/solutions/ literature/white_papers/200011.pdf

[11] Jonathan Davidson and James Peters, “Voice over IP Fundamentals”. CISCO Press, IN: 2000.

[12] CISCO Documentation. “Understanding Delay in Packet Voice Networks”. http://www.cisco.com/warp/public/788/voip/delay-details.html

[13] K. S. Trivedi, “Probability Statistics with Reliability Queuing and Computer

Science Applications". Wiley Interscience, NY: 2002.

[14] Alan Percy. "Understanding Latency in IP Telephony". www.telephonyworld.com/training/ brooktrout/iptel_latency_wp.html

[15] A. P. Markopoulou ,F. A. Tobagi, and M. J . Karam. "Assessing the Quality of Voice Communications over Internet Backbones". IEEE/ACM Transactions on Networking, Volume 11, Issue 5, Oct. 2003 Page(s):747 – 760.

[16] Yi J. Liang, N. Farber and B. Girod. "Adaptive Playout Scheduling Using Time-Scale Modification in Packet Voice Communications". Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on.Volume 3, 7-11 May 2001 Page(s):1445 - 1448 vol.3

[17] R. Ramjee, J. Kurose, D. Towsley and H. Schulzrinne. "Adaptive Playout Mechanisms for Audio Applications in Wide-Area Networks". Proceedings of IEEE Conference on Computer Communications (INFOCOM), pp.680–688, June1994.

[18] P. DeLeon and C. J. Sreenan. "An Adaptive Predictor for Media Playout Buffering". In Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pages 3097-3100, March 1999.

[19] C. J. Sreenan, Jyh-Cheng Chen, P. Agrawal and B. Narendran. "Delay Reduction Techniques for Playout Buffering". IEEE Transaction on Multimedia. IEEE Transactions on Multimedia, 2(2):88-100, June 2000.

[20] T. Kostas, M. S. Borella, Ikhlaq Sidhu, G. M. Schuster, J. Grabiec and J. Mahler. "Real-Time Voice Over Packet-Switched Networks". Network, IEEE. Volume 12, Issue 1, Jan.-Feb. 1998 Page(s):18 - 27

[21] S. B. Moon, J. Kurose and D. Towsley. "Packet Audio Playout Delay Adjustment: Performance Bounds and Algorithm". ACM/Springer Multimedia Systems, vol. 5, no. 1, pp. 17--28, Jan. 1998.

[22] F. Liu, J. Kim, and C.-C. J. Kuo, "Adaptive delay concealment for internet

voice applications with packet-based time-scale modification," in Proc. IEEE ICASSP, May 2001.

[23] A. Shallwani and P. Kabal. “An Adaptive Playout Algorithm With Delay Spike Detection for Real-Time VoIP”. Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on. Volume 2, 4-7 May 2003 Page(s):997 - 1000 vol.2

[24] S. Kalluri and G. R. Arce, “A general class of nonlinear normalized LMS—type adaptive algorithms,” in Proc. Of the 1998 CISS, (Princeton, NJ), Mar. 1998.

[25] S. Haykin, “Adaptive Filter Theory”. Prentice Hall, NJ: 2002.

[26] Ali H. Sayed, “Fundamentals of Adaptive Filtering”. IEEE Press, Wiley Interscience, NJ: 2003.

[27] V. K. Ingle and J. G. Proakis, “Digital Signal Processing Using MATLAB”. Brooks/Cole, CA: 2000.

[28] Steven M. Kay, “Fundamentals of Statistical Signal Processing: Vol. 1 Estimation Theory”. Prentice Hall, NJ: 1993.

[29] Steven M. Kay, “Fundamentals of Statistical Signal Processing: Vol. 2 Detection Theory”. Prentice Hall, NJ: 1993.

[30] Aman Kansal and Abhay Karandikar. “Adaptive Delay Estimation for Low Jitter Audio over Internet”. IEEE GLOBECOM ’01, 2001.

[31] Y. Lau, Z. Hussain and R. Harris, “A Time-Varying Convergence Parameter for the LMS Algorithm in the Presence of White Gaussian Noise”. Australian Telecommunications, Networks and Applications Conference (ATNAC), Melbourne: 2003.

[32] S. Agnihotri, K. Aravindhan, H. S. Jamadagni and B. I. Pawate. “A New Technique for Improving Quality of Speech in Voice over IP Using Time-Scale Modification”. IEEE International Conference (ICASSP 2002).

[33] S. G. Sankaran and A. A. (Louis) Beex. “Normalized LMS Algorithms with

Orthogonal Correction Factors”. Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers, 1997.

[34] “Real-time Voice Communication over the Internet Using Packet Path Diversity”. Yi J. Liang, Eckehard G. Steinbach, and Bernd Girod. Proceedings of the ninth ACM international conference on Multimedia, 2001..

[35] Marco Roccetti, Marco Bernardo and Roberto Gorrierri. “Packetized Audio for Industrial Applications: A Simulation Study”. Proc. 10th European Simulation Symposium (ESS'98).

[36] Saeed Gazor. “Prediction in LMS-Type Adaptive Algorithms for Smoothly Time Varying Environments”. IEEE Transactions on Signal Processing, JUNE 1999.

[37] L. Guo, L. Ljung, G. J. Wang. “Necessary and Sufficient Conditions for Stabilityof LMS”. IEEE Trans. Autom. Control, 1997.

[38] W. Kellerer, E. Steinbach, P. Eisert and B. Girod. “A Real-Time Internet Streaming Media Testbed”. ICME 2002.

[39] M. Kalman, E. Steinbach and and B. Girod. “Adaptive playout for real-Time media streaming”. ISCAS 2002.

[40] E. Steinbach, Y. J. Liang and and B. Girod. “A Simulation Study of Packet Path Diversity for TCP File Transfer and Media Transport on the Internet”. IWDC 2002.

[41] Aman Kansal. “Jitter-free Audio Playout over Best Effort Packet Network”. ATM Forum International Symposium, New Delhi, India, 2001.

[42] E. Steinbach, Y. J. Liang and and B. Girod. “Multi-Stream Voice Over IP Using Packet Path Diversity”. In Proceedings of IEEE 4th Workshop on Multimedia Signal Processing, Oct. 2001

[43] E. Masala, C. F. Chiasserini, M. Meo and J. C. De Martin. "Real-Time Transmission of H.264 Video Over 802.11-Based Wireless Ad Hoc Networks". Processing and Reliable Delivery of Multimedia Signals,May 2003.

[44] C. Raghavan, K. Ravish and P. Mazumdar. "Adaptive Bit Rate Schemes for

Video Streaming Over Wireless Networks". www.comp.nus.edu.sg/~cs4274/ termpapers/0405-I/paper-2.pdf

[45] Y. Koucheryavy, D. Moltchanov and J. Harju. "Performance Evaluation of Live video Streaming Service in 802.11b WLAN Environment Under Different Load Conditions". In Proc. of MIPS'2003, Napoli, Italy, November 2003.

[46] T. Ahmed, A. Mehaoua, R. Boutaba, and Y. Iraqi. "Adaptive Packet Video Streaming Over IP Networks: A Cross-Layer Approach". IEEE Journal on Selected Areas in Communications February 2005, Vol.23 No.2.

[47] H. Gharavi and K. Ban. "Dynamic Adjustment Packet Control for Video Communications over Ad-hoc Networks". IEEE ICC 2004.

[48] B. Girod, Mark Kalman, Yi J. Liang, Rui Zhang. "Advances in Channel-Adaptive Video Streaming". IEEE International Conference on Image Processing (ICIP 2002).

[49] Tianbo Kuang and Carey Williamson. "Real Media Streaming Performance on an IEEE 802.11b Wireless LAN". http://www.cs.wpi.edu/~rek/Adv_Nets/Spring2005/Syllabus05.pdf

[50] Abhik Majumdar, Daniel Grobe Sachs, Igor V. Kozintsev, Kannan Ramchandran, and Minerva M. Yeung. "Multicast and Unicast Real-Time Video Streaming Over Wireless LANs". Ieee Transactions on Circuits and Systems for Video Technology, June 2002.

[51] Nick Feamster and Hari Balakrishnan. "Packet Loss Recovery for Streaming Video". 11th International Packet Video Workshop. April, 2002.

[52] Miroslaw Narbutt and Liam Murphy. "Adaptive Playout Buffering for Audio/Video Transmission Over The Internet". Proceedings of the 17th IEE UK Teletraffic Symposium, Dublin, Ireland, May , 2001.

[53] K. Mayer-Patel, B. C. Smith, and L. A. Rowe. “The Berkeley Software MPEG-1 Video Decoder”. ACM Transaction on Multimedia Computing,

Communications, and Applications, 2005.

[54] C. Schlatter. “Basic Architecture of H.323”. The Swiss Education and Research Network. www.switch.ch/vconf/ws2003/h323_basics_handout.pdf

[55] “MPEG-4 AVC/H.264 Video Coding”. Wipro Technologies. www.wipro.com/insights/mpeg4videocoding.htm

[56] M. Baldi, and Y. Ofek. “End-to-End Delay Analysis of Videoconferencing over Packet-Switched Networks”. IEEE/ACM Transactions on Networking, Vol. 8, No. 4, August 2000.

[57] G. Thom and A. R. Deutermann. “A Comparison of Video Compression Algorithms”. Delta Information Systems, Inc. 2001. www.delta-info.com/DDV/reference.htm

[58] J. Boyce and R. Gaglianello. “Packet Loss Effects on MPEG Video Sent over the Public Internet”. ACM Multimedia ’98 Bristol UK.

[59] Yu-Liang Kuo, Chi-Hung Lu, Eric Hsiao-Kuang Wu and Gen-Huey Chen. “An admission Control Strategy for Differentiated Services in IEEE 802.11”. www.cs.nccu.edu.tw/~ttsai/ mobilecomm_ttsai/papers2004/0707kuo.pdf

[60] “Voice Over IP – Per Call Bandwidth Consumption”. www.cisco.com/warp/public/788/pkt-voice-eneral/bwidth_consume.html

VITA

Name:	Salah Hussain Saleh Al-Khalifa	
Address:	PO Box 9946, Dammam-31423, Saudi Arabia	
Contact Info.	Tel: 966-3-8413807 Fax: 966-3-8415988	Mobile: 966-505-813914 e-mail: salah_alkhalifa@yahoo.com
Education:	B.S. Electrical Engineering, 1997. KFUPM, Dhahran, S.A. M.S. Electrical Engineering, 2006. KFUPM, Dhahran, S.A.	
Employment:	1998-present: Unit Head, Customer Call Center. Operation Div. DAM. E.D. Saudi Electricity Company, Eastern Operating Area.	
Certification & Membership:	Microsoft Certified System Engineer. (1999) IEEE, Member. ACM, Member.	
Technical Interests:	Packet-switched communication networks. Wireless data communications. Real-time systems and synchronization algorithms.	