# Analysis and Estimation of Playout Delay in VoIP Communications

**Fábio Sakuray†, Robinson S. V. Hoto†† and Leonardo S. Mendes†††**

†,†† State University of Londrina, Brazil
††† State University of Campinas, Brazil

**Summary**
This work presents mathematical properties to estimate the optimal buffer delay used in applications that involve the transmission of audio packets aiming to minimize jitter effects. Two important theorems are demonstrated: the first one establishes the necessary and sufficient conditions to avoid loss of data and the second one establishes the conditions for monitored losses. This work presents two algorithms to determine the optimal buffer delay, considering an adjustable loss percentage.

***Key words:***
*Playout Delay, VoIP, Computer Network, Buffer Delay.*

## 1. Introduction

From this section, input the body of your manuscript according to the constitution that you had. For detailed information for authors, please refer to [1].

VoIP applications that operate transmitting packets across the Internet must have their packets executed within the same interval they are transmitted. If a packet is not received up to the moment of its execution, it is considered a lost packet. The time interval *L* (latency), which is the time between transmission and execution of a packet (or playout time) also, affects the quality of a conversation [1][2]. The values considered acceptable for packets loss and latency are, respectively, 5% [2] and 300 ms [3].

However, traffic oscillations on the Internet induce an increase of end-to-end delay and its variation (jitter), thus directly impacting the latency and the display cadence maintenance of the audio samples at the receiver. The jitter directly affects the packet loss indexes because in VoIP applications the packets received after the execution instant are dismissed [1].

The use of a buffer to generate a waiting queue of data packets as they arrive to the receiver is one possible way to reduce the impact of information loss. The construction of a buffer can be interpreted as the insertion of a buffer delay at the execution of the packets, which implies on the reduction of data loss provoked by the jitter and also on cadence maintenance at packets execution [4].

The use of the buffer delay is not an immediate task, since it consists of an optimization problem involving a tradeoff between the loss of information and the determining the waiting time dimension. The incorrect attribution of the buffer delay can cause a disastrous loss of information or an excessive delay on the execution of the voice packets, thus making conversation impossible.

Various researches have been conduced aiming to determine the ideal buffer delay. These researches can be classified in two groups: i) those that use fix values for the buffer delay during the whole audio session and ii) those that use adaptive values for the buffer delay, adjusted during silence periods between talkspurts [5]. The works more often referred to in literature are in group (ii). Among these we can highlight the following: Ramjee [6], which presents an approach based on statistical tactics, Moon [2], which builds an algorithm based on the information history using some bounds, Ramos [8], which aims to improve the work of Moon [4], through the adjustment of the packets loss percentage and Narbutt and Murphy [8], which suggest an approach with parameters adjustment used in moments of delay high variation.

In all these works, the authors use strategies to estimate a buffer delay based on previous buffer delay estimates.

In this work we present a study about the mathematical properties related to the buffer delay and present the necessary conditions to avoid information loss. Further more, we use these results for the description and analysis of two heuristics to dynamically adjust the size of buffer delays.

Section 2 presents a brief description of the problem and section 3 presents definitions and mathematical properties about the control of buffer delay in a talkspurt. Section 4 describes the Least-Squares Buffer Delay (LSBD) Heuristic, which calculates the buffer delay using least-squares estimate and a Hybrid Algorithm, which calculates the buffer delay using some of the characteristics of the algorithm described in [7]. Section 5 organizes de numerical results obtained and section 6 presents the conclusions of our studies

## 2. The Problem

VoIP applications transmit packets in constant time intervals $\Delta t$ , with audio samples captured in activity periods or talkspurts [2]. A packet is sent at instant $t_i$ and is received at instant $a_i$ and executed at instant $p_i$, as shown in figure 1. In order to calculate the network delay

Manuscript received March 5, 2008
Manuscript revised March 20, 2008

it is necessary to determine the value of $|\ell|$, which represents the time difference existing between the clocks of transmitter and receiver. Taking the receiver clock as a reference, $t_i$ will be replaced by ($t_i$ - $\ell$) in such a way that if $\ell < 0$ ($\ell > 0$) the transmitter will be advanced (delayed) compared to the receiver and in case $\ell = 0$, transmitter and receiver will be synchronized as shown in figure 1.

The execution instants $p_i$ of the packets must respect the periodicity $\Delta t$ used by the transmitter to send them, i.e., $p_i - p_{i-1} = \Delta t$ for $i$=2,…,n, or even $p_i = p_1 + (i-1)\Delta t$.

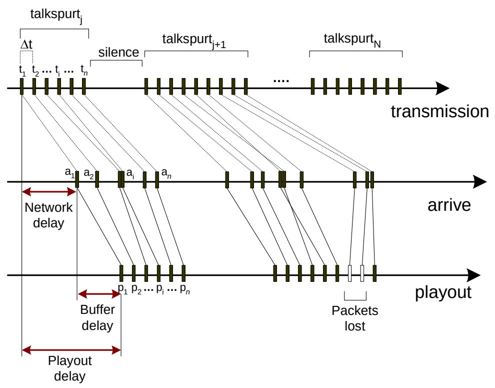


Fig. 1. Transmission of packets of a talkspurt.

Notice that the loss of the packet with index $i$ is mathematically characterized when $p_i < a_i$ or $p_i - (t_i - \ell) > L$, therefore to avoid packet loss, $p_i - a_i \geq 0$ and $p_i - (t_i - \ell) \leq L$, for every $i \in N$. The first inequation is known as playout restriction and the second one as latency restriction. This way, the playout delay (Pd) of the packet with index $i$ is given by:

$$Pd_i = p_i - (t_i - \ell) \qquad (1)$$

**Definition 1.** Buffer delay $T$ is a waiting time for the execution of the packets of a talkspurt.

The introduction of the buffer delay causes the execution instant of a packet with index i to be equal to $p_i = a_1 + T + (i-1)\Delta t$, thus allowing the Quality of Service (QoS) to be controlled. In the next section we present a series of properties related to buffer delay.

## 3. Theoretical Aspects of Buffer Delay

In the next sections, consider a talkspurt with $n$ packets, having packet indexes given by N={1,2,…,$n$}. The first result we present is a property related to the latency restriction.

**Property.** In a talkspurt, if a packet with index $i$ does not violate the playout restriction, and the difference between the reception instant and sending instant overcomes the latency $L$, then the latency restriction is violated by the packet with index $i$, no matter the buffer delay $T$ used in talkspurt.

**Proof.** We need to show that $p_i - (t_i - \ell) > L$, where $p_i = a_1 + T + (i-1)\Delta t$ and $T$ is an arbitrary buffer delay. Notice that $p_i - (t_i - \ell) = p_i - a_i + a_i - (t_i - \ell)$, besides $p_i - a_i \geq 0$ and $a_i - (t_i - \ell) > 0$, so $p_i - (t_i - \ell) > L$.

The network conditions influence the determination of the buffer delay. The previous property illustrates this fact, showing under which conditions the network can determine unavoidable losses, independently from the choice of buffer delay. In our studies we have not considered these losses. We have tried to control only conditions that allow dimensioning the buffer delay. Therefore, without loss of generality of the results, consider that $a_i - (t_i - \ell) \leq L$, for every $i \in N$.

**Theorem 1.** In a talkspurt where a buffer delay $T$ is inserted, no packet is lost, if and only if $\max_{i \in N}\{\delta_i - (i-1)\Delta t\} \leq T \leq \min_{i \in N}\{\gamma_i - (i-1)\Delta t\}$ where $\gamma_i = (t_i - \ell) - a_i + L$ for every $i \in N$ and $\delta_i = a_i - a_1$.

**Proof.** Since there is no packets loss at the talkspurt, this is equivalent to say that:
$p_i - a_i \geq 0$ and $p_i - (t_i - \ell) \leq L$, for every $i \in N$ $\Leftrightarrow$ $p_i \geq a_i$ and $p_i \leq (t_i - \ell) + L$, for every $i \in N$ $\Leftrightarrow$ $a_i \leq p_i \leq (t_i - \ell) + L$, for every $i \in N$ $\Leftrightarrow$ $a_i \leq a_1 + T + (i-1)\Delta t \leq (t_i - \ell) + L$, for every $i \in N$ $\Leftrightarrow$ $\max_{i \in N}\{\delta_i - (i-1)\Delta t\} \leq T \leq \min_{i \in N}\{\gamma_i - (i-1)\Delta t\}$, where $\gamma_i = (t_i - \ell) - a_1 + L$, for every $i \in N$.

Notice that $T_{\min} = \max_{i \in N}\{\delta_i - (i-1)\Delta t\}$ and $T_{\max} = \min_{i \in N}\{\gamma_i - (i-1)\Delta t\}$, where $T_{\min}$ and $T_{\max}$ are, respectively, the minimum buffer delay and the maximum buffer delay, to which the packets loss is not verified. Besides $0 \leq T_{\min} \leq T_{\max}$, whose proof is immediate.

The next result tells us when, in a talkspurt, at least one packet is executed at the instant of its reception and at least one packet is executed with the maximum latency.

**Lemma 1.** If the minimum buffer delay is inserted in a talkspurt, then there is a packet which is executed at the instant of its reception. If, on the other hand, the maximum buffer delay is inserted, then there is a packet that reaches the maximum latency $L$.

**Proof.** Let us see the first part of the lemma: once that $T_{\min} = \max_{i \in N}\{\delta_i - (i-1)\Delta t\} = \delta_r - (r-1)\Delta t$, then:

$p_i = a_1 + T_{mim} + (i-1)\Delta t, \forall i \in N \Leftrightarrow$
$p_i = a_1 + \delta_r - (r-1)\Delta t + (i-1)\Delta t, \forall i \in N \Leftrightarrow$
$p_i = a_r - (r-1)\Delta t, \forall i \in N$.

Considering $i = r$, we have that $p_r = a_r$, i.e., the packet with index $r$, which defines $T_{\min}$, is executed at the instant of its reception. Now the second part: considering $T_{\max} = \min_{i \in N}\{\gamma_i - (i-1)\Delta t\} = \gamma_s - (s-1)\Delta t$, then:

$p_i = a_1 + T_{\max} + (i-1)\Delta t, \forall i \in N \Leftrightarrow$
$p_i = a_1 + \gamma_s - (s-1)\Delta t + (i-1)\Delta t, \forall i \in N \Leftrightarrow$
$p_i = (t_s - \ell) + L - (s-1)\Delta t, \forall i \in N$.

Considering $i = s$, we have that $p_s = (t_s - \ell) + L$, i.e., the packet with index $s$, which defines $T_{\max}$, is executed with maximum latency.

Examining the previous proof where $p_i = a_r - (r-i)\Delta t$, we can realize that there may exist another $r' \in N - \{r\}$, such that $p_{r'} = a_{r'}$, i.e., $a_r - a_{r'} = (r - r')\Delta t$. In case this happens, besides the packet with index $r$, the packet with index $r'$ will also define $T_{\min} = \delta_i - (j-1)\Delta t, j \in \{r, r'\}$, which allows us to state that $\max_{i \in N-\{r,r'\}}\{\delta_i - (i-1)\Delta t\} \le \max_{i \in N}\{\delta_i - (i-1)\Delta t\}$. Now, suppose no other packet defines $T_{\min}$. In this case, $\max_{i \in N-\{r,r'\}}\{\delta_i - (i-1)\Delta t\} < \max_{i \in N}\{\delta_i - (i-1)\Delta t\}$, because if for some index $r'' \in N - \{r, r'\}$, $\delta_{r''} - (r''-1)\Delta t = \max_{i \in N-\{r,r'\}}\{\delta_i - (i-1)\Delta t\} = \max_{i \in N}\{\delta_i - (i-1)\Delta t\}$, then besides $r$ and $r'$, $r''$ would also defined $T_{\min}$, which contradicts the fact that no other packet defines $T_{\min}$. Similarly, suppose that $T_{\max} = \gamma_j - (j-1)\Delta t, j \in \{s, s'\}$, in this case we have that $\min_{i \in N-\{s,s'\}}\{\gamma_i - (i-1)\Delta t\} > \min_{i \in N}\{\gamma_i - (i-1)\Delta t\}$. Therefore, if we introduce a buffer delay $T$, such as $\max_{i \in N-\{r,r'\}}\{\delta_i - (i-1)\Delta t\} \le T < \min\{T_{\min}, \min_{i \in N-\{s,s'\}}\{\gamma_i - (i-1)\Delta t\}\} = T_{\min}$, then certainly only the packets with indexes $r$ and r' will be lost. Similarly, if $T_{\max} = \max\{T_{\max}, \max_{i \in N-\{r,r'\}}\{\delta_i - (i-1)\Delta t\}\} < T \le \min_{i \in N-\{s,s'\}}\{\gamma_i - (i-1)\Delta t\}$, then certainly only the packets with indexes $s$ and s' will be lost. Besides if $T_{\min} \le T \le T_{\max}$ then no packet with buffer delay $T$ will be lost.

Following this logic, it is possible to obtain buffer delays that allow a monitored loss of packets. With this purpose we will introduce the definitions of bounds due to playout and latency.

**Definition 2.** The c-th bound due to playout of a talkspurt is given by $u_c = \max_{i \in \Omega_c}\{\delta_i - (i-1)\Delta t\}$. Where $\Omega_c = N - (U_0 \cup ... \cup U_{c-2} \cup U_{c-1})$, $U_0 = N$ and $U_j = \{r_j^1, ..., r_j^{w_j}\}$ is the set of all the indexes of packets that define $u_j$, j=0,1,…,c-1.

**Definition 3.** The c-th bound due to latency of a talkspurt is given by $v_c = \min_{i \in \Gamma_c}\{\gamma_i - (i-1)\Delta t\}$. Where $\Gamma_c = N - \{V_0 \cup ... \cup V_{c-1})$, $\Gamma_0 = N$ and $V_j = \{s_j^1, ..., s_j^{z_j}\}$ is the set of all the indexes of packets that define $v_j$, j=0,1,…,c-1.

Observe that the definitions 2 and 3 are actually recursive processes where we have that $u_0 = T_{\min}$ and $v_0 = T_{\max}$ and also according to the corollary from theorem 1, $u_0 \le v_0$.

We can show that there are a finite number of bounds due to playout and bounds due to latency associated to a talkspurt. Besides, there is not a single packet that defines two bounds due to distinct playout and, the reunion of the packets that define every bound due to playout forms the set of all the packets of the talkspurt. Considering the observations in the previous paragraph, it is possible to conclude that:

**Lemma 2.** $u_j \le u_{j-1}$ for j=1,…,m and also $v_{j-1} < v_j$ for j=1,…,k.

**Proof.** First of all we will show that $u_j \le u_{j-1}$. After that we will verify that the equality makes no sense. Let us see: $u_j = \max_{i \in \Omega_j}\{\delta_i - (i-1)\Delta t\}$ and $u_{j-1} = \max_{i \in \Omega_{j-1}}\{\delta_i - (i-1)\Delta t\}$, where $\Omega_j = N - (U_0 \cup ... \cup U_{j-2} \cup U_{j-1})$ and $\Omega_{j-1} = N - (U_0 \cup ... \cup U_{j-2})$, so $\Omega_j \subset \Omega_{j-1}$, therefore $u_j \le u_{j-1}$. Now if $u_j = u_{j-1}$, then $U_{j-1} \cap U_j \ne \varnothing$, which is an absurd. The proof that $v_{j-1} < v_j$ for j=1,…,k follows the same logic and therefore will be omitted.

Along the whole text, we will refer to every interval $P_m = [0, u_m)$, $P_{m-1} = [u_m, u_{m-1}), ..., P_0 = [u_1, u_0)$ by step due to playout and by step due to latency the intervals $L_0 = (v_0, v_1], ..., L_{k-1} = (v_{k-1}, v_k], L_k = (v_k, +\infty)$. The interval $\Phi = [u_0, v_0]$ will be named hybrid step.

Consider next the concept of degree of a step, which is associated to the level of packets loss for a certain buffer delay.

**Definition 4.** At each step of the talkspurt we have associated a number named degree, given by:

$\text{degree}(P_j) = \sum_{c=0}^{j}|U_c| = \sum_{c=0}^{j} W_c$, where $u_{m+1} = 0$, j=0,…,m;

$\text{degree}(\Phi) = 0$;

$\text{degree}(L_j) = \sum_{c=0}^{j} |V_c| = \sum_{c=0}^{j} Z_c$ , where $v_{k+1} \to +\infty$ , j=0,…k;

The degree of each step is unique by definition. Besides $w_c \geq 1$ and $z_c \geq 1$ for each *c*, from what we can conclude that:

$$0 = \text{degree}(\Phi) < \text{degree}(P_0) < ... < \text{degree}(P_{m-1}) < \text{degree}(P_m)$$

$$0 = \text{degree}(\Phi) < \text{degree}(L_0) < ... < \text{degree}(L_{k-1}) < \text{degree}(L_k)$$

Notice also that if buffer delay *T* is inserted in a talkspurt, then the number of packets lost will be equal to the degree of the step to which *T* belongs, which allows us to build the graphic of figure 2. We can notice that the loss of packets will be due to the violation of the playout or latency restriction, but never due to both.

Notice that the buffer delay *T* belonging to the hybrid step ensures that no packet of its talkspurt will be lost. In this context, we have introduced a minimum waiting time in a conversation, i.e., we are interested in solving the following problem:

$$\text{mim}\{\text{T} = \text{p}_\text{i} - \text{a}_1 - (\text{i}-1)\Delta\text{t} \mid \text{a}_\text{i} \leq \text{p}_\text{i} \leq \text{L} + (\text{t}_\text{i} - \ell), i \in N\}$$

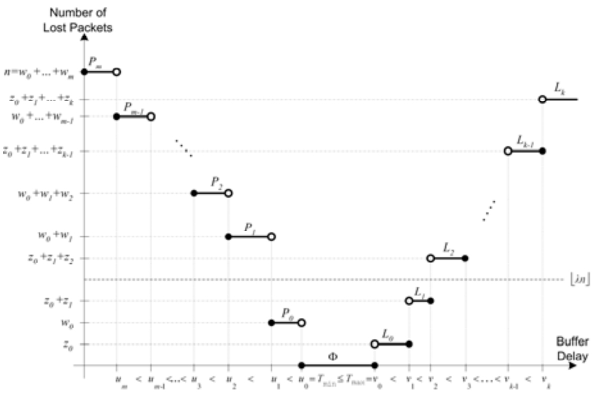


Fig. 2. Graphic of loss related to the possible buffer delays T in a talkspurt.

Theorem 1 guarantees that the previous problem can be described as: $\min\{f(T) = T \mid T \in \Phi\}$ .The solution of this last problem is equal to $T = u_0 = T_{\min}$ , i.e., the solution of the problem where no loss of packets is admitted is precisely the minimum buffer delay.

On the other hand, the good quality of voice packets communication admits a certain limit of loss. Therefore let us suppose a factor $\lambda \in (0,1)$ of packets loss in a talkspurt, i.e., at most $\lfloor \lambda n \rfloor$ packets can be lost (see figure 2) where $\lfloor n \rfloor$ is the floor function (greater integer smaller than or equal to *x*). In this case we are interested in solving a new problem: $\min\{f(T) = T \mid \Psi(T) \leq \lfloor \lambda n \rfloor, T \in [0,+\infty)\}$ .

The figure 2 allows the conclusion that in a talkspurt where the buffer delay *T* is inserted, *n'* packets are lost if and only if *T* belongs to a step whose degree is *n'*. For this reason and because $\{P_m,..., P_0, \Phi, L_0,..., L_k\}$ is a partition of $[0,+\infty) \ni T$ , a formulation equivalent to the last problem is presented in equation (2)

$$\min\{\min\{f(T) = T \mid \Psi(T) \leq \lfloor \lambda n \rfloor, T \in I\}, \quad I \in \{P_m, \ldots, P_0, \Phi, L_0, \ldots, L_k\}\} \qquad (2)$$

Notice that if *I* is a step due to latency, the associated problem can be dismissed, thus resulting in the following problem: $\min\{f(T) = T \mid \Psi(T) \leq \lfloor \lambda n \rfloor, T \in P_m \cup \ldots \cup P_0 \cup \Phi\}$ . Considering that $\{P_m,..., P_0, \Phi\}$ is a partition of $[0, v_0]$ , we have that $P_j \cup ... \cup P_0 \cup \Phi = [u_{j+1}, 0]$ for some *j*. Therefore the solution of our last problem is $T = u_{j+1} = T_{\min,\lambda}$ where $T_{\min,\lambda}$ is the minimum buffer delay with loss factor $\lambda \in (0,1)$.

In the next section we will describe computational procedures to determine buffer delay estimates along a trace.

## 4. Buffer Delay Estimation

Next, we present two algorithms to estimate the buffer delay with the following characteristics: dynamic adjustment and determination of buffer delay at the beginning of each talkspurt and adjustment of the percentage of loss to a value defined by user. Once defined the value of the buffer delay it will not be changed until the last packet of the same talkspurt is received.

### 4.1 The LSBD Heuristic

The LSBD (Least-Squares Buffer Delay) algorithm estimates the value of the buffer delay for each talkspurt using the packet loss history as input in a least-squares system, aiming a certain number of losses during trace. Consider $\rho_k$ the total of packets loss at a talkspurt with index *k*=1,2,…,K where $\varepsilon \in (0,1)$ and:

$$\sum_{i=1}^{k} \rho_i \leq \varepsilon \sum_{i=1}^{k} n_i \Leftrightarrow \rho_k + \sum_{i=1}^{k-1} \rho_i \leq \varepsilon \sum_{i=1}^{k} n_i \Leftrightarrow$$

$$\Leftrightarrow \rho_k \leq \varepsilon \sum_{i=1}^{k} n_i - \sum_{i=1}^{k-1} \rho_i \Leftrightarrow$$

$$\Leftrightarrow \frac{\rho_k}{n_k} \leq \frac{\left( \varepsilon \sum_{i=1}^{k} n_i - \sum_{i=1}^{k-1} \rho_i \right)}{n_k} = \lambda_k$$

Notice that $\lambda_k \varepsilon (0,1)$ indicates the acceptable loss of the talkspurt with index *k*, considering the loss compromise $\varepsilon \in (0,1)$ at the trace, i.e., if we take $\rho_k / n_k \leq \lambda_k$ for each

$k$=1,2,…,K at the end we will have $\sum_{i=1}^{k}\rho_i \le \varepsilon\sum_{i=1}^{k} n_i$ . Now considering that we have the values for the buffer delays $T_i$ , i=1,..$k$-1 for $k$-1<K talkspurts. We will estimate the buffer delay of the $k^{th}$ talkspurt with the aid of the following polynomial: $\tau(i) = c_0 + c_1 i + ... + c_m i^m$ .

The polynomial will be obtained by least-squares. Therefore considering function $\gamma(c_0,...,c_m) = \sum_{i=1}^{k-1}(\tau(i) - T_i)^2 = \sum_{i=1}^{k-1}\left(c_0 + c_1 i + ... + c_m i^m - T_i\right)^2$ whose gradient offers the system:

$$\partial/\partial_{C_0}\ \gamma(c_0,...,c_m) = 2\sum_{i=1}^{k-1}\left(c_0 + c_1 i + ... + c_m i^m - T_i\right) = 0$$

$$\vdots$$

$$\partial/\partial_{C_m}\ \gamma(c_0,...,c_m) = 2\sum_{i=1}^{k-1}\left(c_0 + c_1 i + ... + c_m i^m - T_i\right) = 0 .$$

Or also, $B(m)\,.c(m) = d(m)$ in matrix format where:

$$B(m) = \begin{pmatrix} \sum_{i=1}^{k-1} 1 & \cdots & \sum_{i=1}^{k-1} i^m \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{k-1} i^m & \cdots & \sum_{i=1}^{k-1} i^{2m} \end{pmatrix} ,\quad c(m) = \begin{pmatrix} c_0 & \cdots & c_m \end{pmatrix}^T \quad \text{and}$$

$$d(m) = \begin{pmatrix} \sum_{i=1}^{k-1} T_i & \cdots & \sum_{i=1}^{k-1} i^m T_i \end{pmatrix}^T .$$

Once $c(m)$ is found, the buffer delay of the talkspurt will be estimated as:

$$T_k = \tau(k) = c_0 + c_1 k + ... + c_m k^m \qquad (3)$$

In order to facilitate dimensioning tests of the buffer delay, let us assume that $\ell = 0$ , i.e., transmitter and receiver will be synchronized. It is important to stress that this conduct does not interfere on the theoretical results obtained at the first part of this work. The algorithm is presented next.

*LSBD*(ε) ;
 $\Delta t = 20$ ; degree=5; $T_1 = 200$ ;
 $k = 1$ ; {talkspurt counter}
 while not end of trace do
  $i = 1$ ; {packet counter}
  $h = 1$ ; {1st packet in latency rule}
  $\rho_k = 0$ ; {packet loss counter}
  while not end of talkspurt do
   if $a_i - t_i < L$ then
    $p_i = a_1 + T_k + (i-1)\Delta t$ ;
    if $p_i < a_i$ then $\rho_k = \rho_k + 1$ ;
     $h = h + 1$ ;
    end;
    $i = i + 1$ ;
   end;
   $n_k = h - 1$ ;
   $\lambda_k = \left(\varepsilon\sum_{i=1}^{k} n_i - \sum_{i=1}^{k-1}\rho_i\right)\Big/ n_k$ ; {lost factor}
   /* find j where */
  $\deg ree(P_{j-1,k}) < \lfloor \lambda_k n_k \rfloor < \deg ree(P_{j,k})$
   while ( $u_{j+1,k} < 0$ ) e ( $j \ge 0$ ) do
    $j = j - 1$ ;
   if $j = 0$ then $T_k = 0$ ;
    else $T_k = u_{j+1,k}$ ;
   $k = k + 1$ ;
   if $k \le 2$ then $T_k = 200$ ;
    else
    Solve
  $b(\deg ree)\,.c(\deg ree) = d(\deg ree)$ ;
   $T_k = \tau(k)$ ; {buffer delay estimated for next talkspurt}
  end;
 end;

## 4.2 The Hybrid Algorithm

In order to present the hybrid algorithm it is necessary to describe the *Move Average* [7] algorithm, which uses the Ramjee [6] algorithm to estimate the first 100 playout delays. The hybrid algorithm presents the procedures of algorithm *Move Average* to estimate the buffer delay and the results that allow the exact calculation of previous buffer delays aiming to improve the estimate of the next buffer delay.

*Hybrid*($\varepsilon$) ;
 $\Delta t = 20$ ;
 $T_1 = 200$ ; {buffer delay talkspurt}
 $k = 1$ ; {talkspurt counter}
 while not end of trace do
  $i = 1$ ; {fisrt packet of talkspurt}
  $h = 1$ ; { packet loss}
  $\rho_k = 0$ ; {packet loss in talkspurt}
  while not end of talkspurt do
   if $a_i - t_i < L$ then
    $p_i = a_1 + T_k + (i-1)\Delta t$ ;

if $p_i < a_i$ then
$\rho_k = \rho_k + 1$ ;
$h = h + 1$ ;
end;
$i = i + 1$ ; {next packet}
end;
$n_k = h - 1$ ;
$\lambda_k = \left( \varepsilon \sum_{i=1}^{k} n_i - \sum_{i=1}^{k-1} \rho_i \right) \Big/ n_k$ ; {loss factor}
/* find j where, */
$\deg ree(P_{j-1,k}) < \lfloor \lambda_k n_k \rfloor < \deg ree(P_{j,k})$ ;
while ( $u_{j+1,k} < 0$ ) e ( $j \geq 0$ ) do $j = j - 1$ ;
if $j = 0$ then $T_k = 0$ ;
else $T_k = u_{j+1,k}$ ;
$k = k + 1$ ; {next talkspurt}
/*finding next buffer delay*/
$X_k = G(T_k)$ ; { $G(x) = e^{-\alpha x}$ }
/*find $a_i$ where:*/

$$\sum_{m=0}^{M} a_{m+1} R_d(m-l) = R_d(l+1), l = 0,1,2,...,M-1 \text{ ;}$$

{with

$$R_d(r) \cong \frac{1}{k-|r|} \sum_{h=1}^{k-|r|} T_h T_{h+|r|}, r = \pm 0, \pm 1, \pm 2, ..., \pm (k-1) \text{ \}}$$

/*find the model's order M*/

$$\hat{X}_k = \sum_{i=1}^{M} a_i X_{k-i+1} \text{ ;}$$

$$\hat{T}_k = G^{-1}(X_k) \text{ ;}$$

/*find $a_i$ where:*/

$$\sum_{m=0}^{M} a_{m+1} R_d(m-l) = R_d(l+1), l = 0,1,2,...,M-1 \text{ ;}$$

/*with*/

$$R_d(r) \cong \frac{1}{k-|r|} \sum_{h=1}^{k-|r|} T_h T_{h+|r|}, r = \pm 0, \pm 1, \pm 2, ..., \pm (k-1)$$

end;

## 3. Numerical Results

In order to compare the heuristics, we have generated traces that simulate the transmission of audio packets between two interlocutors. The traffic generated follows the model presented in [9] . The packets have been transmitted in constant intervals of 20 ms. The synchronism was obtained sending and receiving packets in the same equipment situated at the State University of Londrina using a packet retransmission element at State University of Campinas. Table 1 illustrates the traces generated and figures 2, 3, 4, 6, 7 and 8 show the results obtained.

Table 1. Traces used at the simulation of algorithms.

| Trace | Time (sec) | Packets |
|---|---|---|
| #1 | 1257 | 58140 |
| #2 | 1882 | 87191 |
| #3 | 2756 | 127298 |

The average of the trace playout delays was used as performance measure, since this is the approach usually found in other works of the same area. At the LSBD heuristic the polynomial with degree five have been adjusted.
The figures 3, 5 and 7 presents the packet loss resulting of algorithms LSBD, Hybrid and Moving Average. The Bisection represents the values of packet loss required in each one. Observing the graphics of figures 4, 6 and 8, it is possible to notice that Moving Average heuristic [7] obtains the better values of playout delay. However when analyzing the graphics in 3, 5 and 7, it is possible to notice that the loss compromise of this heuristic is the worse when comparing to the other two procedures, thus compromising the quality of the conversation.

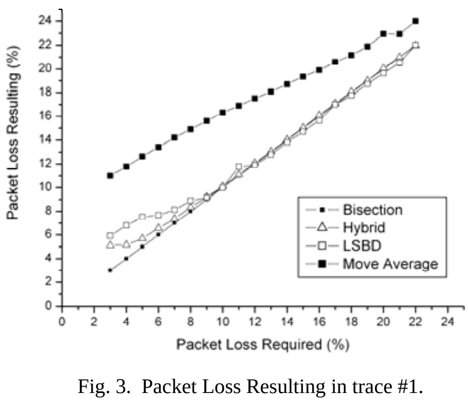


Fig. 3. Packet Loss Resulting in trace #1.

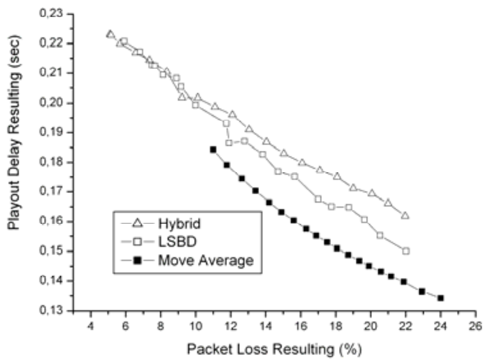


Fig. 4. Playout Delay Resulting in trace #1.

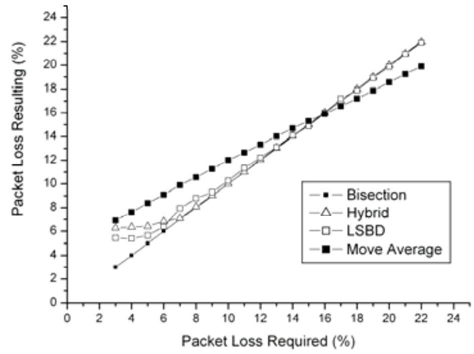


Fig. 7. Packet Loss Resulting in trace #3.

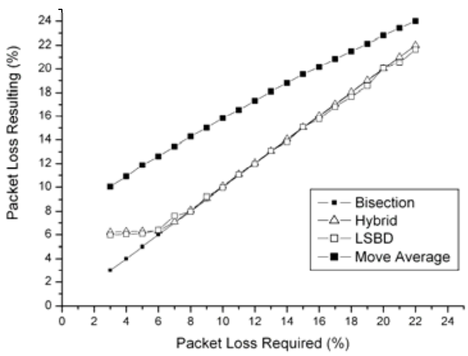


Fig. 5. Packet Loss Resulting in trace #2.

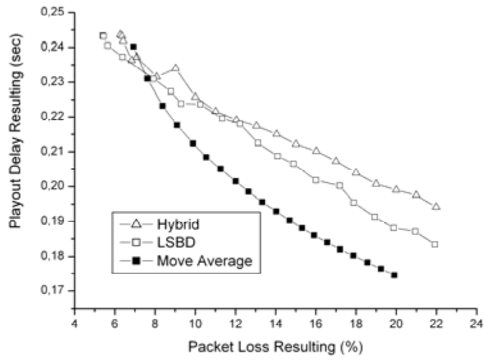


Fig. 8. Playout Delay Resulting in trace #3.

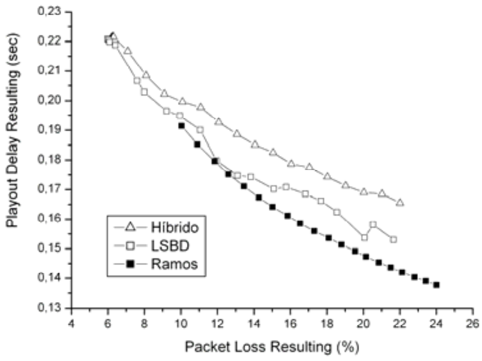


Fig. 6. Playout Delay Resulting in trace #2.

## 6. Conclusions

Determining the best value for the buffer delay reduces the delay between transmitter and receiver, which is damaging in interactive communications, and also maintains the packets loss rate (not employment due to playout delay) in acceptable levels.

The paper has presented two algorithms (LSBD and Hybrid), which dynamically determine the value of the buffer delay, considering the loss compromise. Simulations show that both have an expected behavior for the playout delay values compared to the loss percentage, i.e., as the packets loss increases, it is possible to notice a reduction of the playout delay (figures 4, 6 and 8).

Traditionally, works in this area do not observe the behavior of an algorithm on what concerns packets loss along the trace. By measuring this, we have verified that the heuristic proposed in [7] has not presented good results, according to the graphics in figures 3, 5 and 7.

On the other hand, LSBD heuristic and the Hybrid procedure proposed by us respect the packets loss tolerance along the traces.
Another point to be considered consists in testing our ideas in traces using spike [4]. For this case we are developing a predictor-corrector method based on a NARMAX model [10].
Finally, our preliminary results already indicate that applications for audio communication in real-time can still have their development improved.


### Acknowledgments

Professor Robinson Hoto would like to thank CNPq for financial support.


## References


[1] Kurose, J., Ross, K, W.: Computer Network: A top-down approach. 4th Edition. Addison-Wesley (2007).
[2] Moon, S., Kurose, J., Towsley, D.: Packet audio playout delay adjustment: performance bounds and algorithms. Multimedia Systems, volume 6, (1998) 17-28.
[3] ITU-T Recommendation G.114: One-Way Transmission Time. (2000).
[4] Perkins, C.: RTP: Audio and Video for the Internet, Addison-Wesley, Boston. (2003)
[5] Pinto, J., Christensen, K. J.: An Algorithm for Playout of Packet Voice Based on Adaptive Adjustment of Talkspurt Silence Periods. Proceedings of IEEE 24o Conference on Local Networks (1999) 224-231.
[6] Ramjee, R., Kurose, J., Towsley, D., Shulzrinne, H.: Adaptive playout mechanisms for packetized audio applications in wide-area networks. Proceedings of the IEEE Infocom, (1994) 680–688.
[7] 7. Ramos, V. M., Barak, R. C., Altman, E.: A Moving Average Predictor for Playout Delay Control in VoIP. Proceedings 11th International Workshop, Berkeley, CA, USA, (2003) 155-173.
[8] Narbutt, M., Murphy, L.: VoIP Playout Buffer Adjustment Using Adaptive Estimation of Network Delays. Proceedings 18th Int'l Teletraffic Congress (ITC-18), Elsevier, (2003) 1171–1180.
[9] ITU-T Recommendation P.59. Telephone transmission quality objective measuring apparatus: Artificial conversation speech (1993).
[10] Aguirre L. A.: Introdução à identificação de sistemas. Editora UFMG, Belo Horizonte, MG – Brazil (2004).


**Robinso Hoto** is member of the researchers' team of National Council of Scientific and Technological Development (CNPq). Him it possesses Graduation in Mathematics from São Paulo State University (1993), M.Sc. degree in Sciences of the Computation and Mathematical Computational from University of São Paulo (1996) and Ph.D. in Engineering of Systems and Computation from Federal University of Rio de Janeiro (2001). Now he works in the State University of Londrina (UEL), where it teaches since 1996. He also works and it coordinates the Course of Master's Degree in Applied Mathematics and Computacional of UEL. He is Consultant Technical of CNPq, Consultant Ad Hoc of the Foundation Araucaria of Support to the Development Inform and Technological of Paraná, and reviewer of important journals. Robinson Hoto is coordinating of the Laboratory of Simulation and Optimization of Systems (SimuLab) and he has experience in the areas of Applied Mathematics and Operational Research.

**Fabio Sakuray** received his M.Sc degree in Computer Science from Federal University of São Carlos, Brazil, in 1994 Also, he is a computer science professor since 1994 in State University of Londrina (UEL), Brazil. His research interests include VoIP communication and peer-to-peer network. He currently is leader of Laboratory of Research and Development in Computer Network (Orion) of Computer Science Department at UEL

**Leonardo S. Mendes** received his B.Sc. degree in 1985 from the Gama Filho University, Rio de Janeiro, his M.Sc. degree in 1987 from the Catholic University of Rio de Janeiro, and his Ph.D. degree in 1991 from Syracuse University, all in Electrical Engineering. In 1992 he joined the School of Electrical Engineering of the State University of Campinas, Brazil. Prof. Mendes's recent R&D focus is in the studies and development of Communications Engineering applications for metropolitan IP networks. Prof. Mendes created, at UNICAMP, the Laboratory of Communications Network (LaRCom), from which he is now the Director and also the main coordinator. At LaRCom , Prof. Mendes and his group have developed or are developing the following projects: 1) an optical system simulator to help in the analysis of optical networks; 2) an environment for the simulation of systems using event driven technique which allows the development of ATM, IP and CDMA simulators; 3) development of Internet set top boxes using J2ME for small devices; 4) communications description of Internet devices using CORBA component modules for Telecommunications; 5) development of e-Learning objects for the PGL project.