

Real-Time Analysis of Delay Variation for Packet Loss Prediction

Lopamudra Roychoudhuri and Ehab S. Al-Shaer

School of Computer Science, Telecommunications and Information Systems,
DePaul University, 243 S. Wabash Ave., Chicago, IL-60604, USA

Abstract. The effect of packet loss on the quality of real-time audio is significant. Nevertheless, Internet measurement experiments continue to show a considerable variation of packet loss, which makes audio error recovery and concealment challenging. Our objective is to predict packet loss in real-time audio streams based on the available bandwidth and delay variation and trend, enabling proactive error recovery for real-time audio over the Internet. Our preliminary simulation and experimentation results with various sites on the Internet show the effectiveness and the accuracy of the Loss Predictor technique.

1 Introduction

Quality of an audio communication is highly sensitive to packet loss [4],[17]. Majority of packet loss in the Internet occurs as the result of congestion in the links. Packet loss for audio is normally rectified by adding redundancy using Forward Error Correction (FEC) [11]. However, unnecessarily high degree of FEC can actually be detrimental rather than beneficial to the ongoing communication because of the excessive traffic. Here the challenge is to ensure a bandwidth-friendly transmission with an effective degree of loss recovery by dynamically changing the degree of FEC. In this paper, we present the investigations to develop a mechanism to predict packet loss in real-time audio streams based on delay variation and trend, which will enable proactive error recovery and rate control for real-time Internet audio.

The basic idea of our proposed on-line loss prediction method is to successfully track the increase and decrease trends in one way delay, and accordingly predict the likelihood of packet loss due to congestion leading to lack of available bandwidth. However, this task becomes difficult due to the unpredictable and dynamic nature of cross traffic in the Internet. In this paper we attempt to formalize a framework to express the likelihood of loss in the next packet train in terms of (1) changes in the available bandwidth, manifested as end-to-end delay variations, and (2) near-past history of congestion in terms of short-term and long-term trends in delay variation. The value returned by the Predictor indicates the current degree and severity of congestion and lack of available bandwidth, hence the likelihood of packet loss, created by cross traffic bursts. The predictor value is fed back from the receiver to the sender in order for the sender to take proactive FEC actions and rate control.

In our approach, we designate the minimum delay of a path as the baseline delay, signifying the delay under no congestion. We also identify the delay at the capacity saturation point of a path as the loss threshold delay, after which packet loss is more likely. We track the increase patterns or trends of the delay as an indication of congestion causing packet loss. We have seen in our experiments that each site shows a consistent minimum baseline delay. We also observe a range of loss threshold delay values after which loss is observed more often. The loss threshold delay shows a variety of ranges and behaviors due the unpredictable nature of the cross traffic in the network at that time. To measure these path delay characteristics, we propose certain measurement metrics classified in three categories – Delay Distance, Short Term Trend and Long Term Trend. Delay Distance gives an absolute ratio of the delay value in relation to the baseline and loss thresholds. The Short-term Trend and the Long-term Trend metrics give indications of sharpness and consistency of upward and downward delay trends in short and long term window of past packets. In the Loss Predictor approach we determine these metrics from the ongoing traffic and combine them with different weights based on their importance in order to estimate of the packet loss likelihood.

Subsequent sections are organized as follows. Section 2 contains references to the related work. Section 3 presents the Predictor approach, with subsections describing the delay-loss correlation model and the Predictor formalization. We present the evaluation results in Section 4, and conclusion and future work in Section 5.

2 Related Work

Paxson examined the delay-loss correlation issue in his PhD thesis [16]. But he concluded that the linkage between delay variation and loss was weak, though not negligible. In contrast with Paxson’s observations, we predict packet loss based on the observed patterns of delay variation, rather than depending on the overall amount of delay variation as indicator of congestion.

Moon’s Technical Report and PhD thesis [13] explored this issue to a certain degree. They reported a quantitative study of delay and loss correlation patterns from offline analysis of measurement data from the Internet. But they did not attempt to take a further step of real-time prediction of packet loss from the delay variation data of an ongoing communication.

The researchers measuring performance of Mbone, a virtual Internet Backbone for multicast IP, observed that there is a correlation between the bandwidth used and the amount of loss experienced [5],[6].

Pathload [10] uses the delay variation principle to measure available bandwidth. They send streams of increasing rate till the stream saturates the available bandwidth and starts showing distinct increase in delay. The same principle is used in TCP Vegas [3], which exploits RTT variation to measure the difference between the actual and the expected sending rate to provide better congestion detection and control. Packet loss is highly probable when the available bandwidth is low and is consumed by the ongoing cross traffic. Our research methods and experiments are based on this premise.

Sender based Error and Rate Control mechanisms for audio, such as by Bolot & Garcia [2], Bolot & Fosse-Parisis [1], Padhye, Christensen, & Moreno [15], Mohamed et al. [12] adapt to packet loss using RTCP feedback from the receiver – thus these mechanisms react to packet loss. In contrast, we predict packet loss and take FEC actions based on the nature of the prediction; hence our approach handles packet loss proactively based on the current loss likelihood.

3 Loss Predictor Approach

3.1 Analysis of Delay-Loss Correlation Model

Here we formalize a number of propositions which constitute our framework based on baseline, loss threshold delay and trends.

Let the path from the sender to the receiver consist of H links, $i = 1, \dots, H$. Let the capacity of the link be C_i . If the available bandwidth at link i be A_i , the utilization of link i is $u_i = (C_i - A_i)/C_i$. We assume that the links follow FCFS queuing discipline and Droptail packet dropping mechanism. Let the sender send a periodic stream of K packets, each of size L bytes, at rate R_0 , to the receiver. Hence a packet is sent in $L/R_0 = T$ time interval. Also there are $(C_i - A_i)T$ (in other words, $u_i C_i T$) bytes of cross-traffic, in addition to the source stream, arriving at link i in interval T . It has been shown in [10] that the One-Way Delay (OWD) D^k of the k -th packet is:

$$D^k = \sum_{i=1}^H \left(\frac{L}{C_i} + \frac{q_i^k}{C_i} \right) = \sum_{i=1}^H \left(\frac{L}{C_i} + d_i^k \right) \quad (1)$$

and the OWD difference between two successive packet k and $k + 1$ is

$$\Delta D^k = D^{k+1} - D^k = \sum_{i=1}^H \frac{\Delta q_i^k}{C_i} = \sum_{i=1}^H \Delta d_i^k \quad (2)$$

where q_i^k is the queue size at link i upon arrival of packet k (not including the k th packet), and d_i^k is the queuing delay of packet k at link i , and $\Delta q_i^k = q_i^{k+1} - q_i^k$, $\Delta d_i^k = d_i^{k+1} - d_i^k$. Clearly, the minimum OWD a packet can have is when there is no queuing delay, denoted as the baseline delay. From (1):

$$D_{baseline} = \sum_{i=1}^H \frac{L}{C_i} \quad (3)$$

Proposition 1. Δd^k increases as the available bandwidths at the tight links decrease, and vice versa.

Let a set of ‘tight’ links be the set of links in the path that contribute to the majority of the queuing delay. Over the interval T , the link i receives $(R_{i-1}T +$

$u_i C_i T$) bytes (stream and cross traffic) and services $C_i T$ bytes, where R_{i-1} is the exit-rate from link $i - 1$. As shown in [10], we can express R_i as follows:

$$R_i = \begin{cases} R_{i-1} \frac{C_i}{C_i + (R_{i-1} - A_i)} & \text{if } R_{i-1} > A_i \\ R_{i-1} & \text{otherwise} \end{cases} \tag{4}$$

The change in queue size Δq_i^k can be expressed as the difference of the arrival rate (packet and cross-traffic) at i -th link and service rate at i -th link: $(R_{i-1}T + u_i C_i T) - C_i T = (R_{i-1} - A_i)T > 0$ if $R_{i-1} > A_i$. Thus,

$$\Delta d_i^k = \begin{cases} \frac{(R_{i-1} - A_i)}{C_i} T > 0 & \text{if } R_{i-1} > A_i \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Suppose there are k ‘tight’ links, i.e. for links $m_1, m_2, \dots, m_k, R_{i-1} > A_i$. Then from (2) and (5),

$$\Delta D^k = \sum_{i=1}^H \Delta d_i^k = \frac{R_{m_1-1} - A_{m_1}}{C_{m_1}} T + \dots + \frac{R_{m_k-1} - A_{m_k}}{C_{m_k}} T \tag{6}$$

Hence ΔD^k will increase if the available bandwidths at the tight links $A_{m_1}, A_{m_2}, \dots, A_{m_k}$, decrease, and vice versa. Thus D_k will show steeper rate of increase if there is lesser available bandwidth at the tight links, i.e. if there is higher rate of cross traffic through the tight links.

Suppose in particular, m -th link is the considerably narrow link, s. t. $C_m \ll C_i, i = 1, \dots, H, i \neq m$. For the sake of simplicity, let us also consider that (i) m -th link is the most tight link (bottleneck link), i.e. the available bandwidth at this link $A_m = (1 - u_m)C_m$ is the minimum available bandwidth of the path, and (ii) $R_{i-1} > A_i$ only for $i = m$, since m -th link is much narrower than others. Then from (4), $R_{m-1} = R_{m-2} \dots = R_0$. Then (6) can be simplified as:

$$\Delta D^k = \frac{R_{m-1} - A_m}{C_m} T = \frac{R_0 - A_m}{C_m} T \tag{7}$$

Hence ΔD^k increases if the minimum available bandwidth at the bottleneck link decreases, and vice versa. \square

Thus under simplified conditions, D^k will show steeper rate of increase if there is lesser available bandwidth at the tightmost link, i.e. if there is higher rate of cross traffic through the tightmost link.

Proposition 2. *There exists a range of OWDs in which packet loss is likely.*

In this proposition we show the existence of a range of delays in which loss is likely, that is, we establish a range of loss threshold delays. Since $C_m < C_i$, hence $1/C_m > 1/C_i, i = 1, \dots, H, i \neq m$. Then $M/C_m \geq q_i/C_i, i = 1, \dots, H$, where M is the queue capacity at m -th link, assuming that the queue capacities at each link are same. When the queue at m -th link is filled up, the OWD is given by:

$$D_{ObservedLoss} = \sum_{i=1, i \neq m}^H \left(\frac{L}{C_i} + \frac{q_i^k}{C_i} \right) + \frac{L + M}{C_m} \tag{8}$$

$D_{ObservedLoss}$ is the observed loss threshold delay (OTD). Delay values higher than this value indicate that loss of the next packet is likely. From (3) and (8),

$$D_{ObservedLoss} > D_{baseline} \quad (9)$$

In particular,

$$\sum_{i=1}^H \frac{L}{C_i} + \frac{HM}{C_m} \geq D_{ObservedLoss} \geq \sum_{i=1}^H \frac{L}{C_i} + \frac{M}{C_m}$$

Thus OTD is bounded by the minimum loss threshold delay (lower bound), $D_{minOTD} = \sum_{i=1}^H \frac{L}{C_i} + \frac{M}{C_m}$, when only one link is the bottleneck, and the maximum loss threshold delay (upper bound) $D_{maxOTD} = \sum_{i=1}^H \frac{L}{C_i} + \frac{HM}{C_m}$ when all H links are bottlenecks. In particular, if $\frac{M}{C_m} \geq \sum_{i=1}^H \frac{q_i^k}{C_i}$, i.e. the occupied queue size at the tight link is larger than the sum of queue sizes at the rest of the links, which is possible when there is a single bottleneck considerably narrower than the rest of the links,

$$D_{ObservedLoss} = \sum_{i=1}^H \frac{L}{C_i} + \frac{M}{C_m} \geq \sum_{i=1}^H \left(\frac{L}{C_i} + \frac{q_i^k}{C_i} \right) \quad (10)$$

In that case the observed loss threshold delay is equal to the maximum observed delay. \square

Thus the likelihood of loss of the packet k at m -th link can be expressed as a combination of the following delay factors: (1) the current total queuing delay relative to the worst queuing delay (from (1), (3), (8) and (9)),

$$= \frac{D^k - D_{baseline}}{D_{ObservedLoss} - D_{baseline}} = \frac{\sum_{i=1}^H d_i^k}{\sum_{i=1, i \neq m}^H d_i^k + \frac{M}{C_m}} \quad (11)$$

which is a value between 0 and 1, and (2) rate of delay increase, (from (7))

$$= \frac{\delta D^k}{\delta T} = \frac{R_0 - A_m}{C_m} \quad (12)$$

3.2 Loss Predictor Formalization

The challenges of implementing the above model and deriving the likelihood of packet loss from network delay measurements are as follows: (1) to establish a range of delays that can be considered as the baseline delay for a particular path, (2) to determine a range of delays for a loss threshold, (3) to measure the delay increase ratio and trends, and (4) to find a likelihood of loss from the combining the above factors.

These challenges lead us to develop the Loss Predictor metrics - (1) Delay Distance, (2) Short-term Trend and (3) Long-term Trend. Delay Distance is

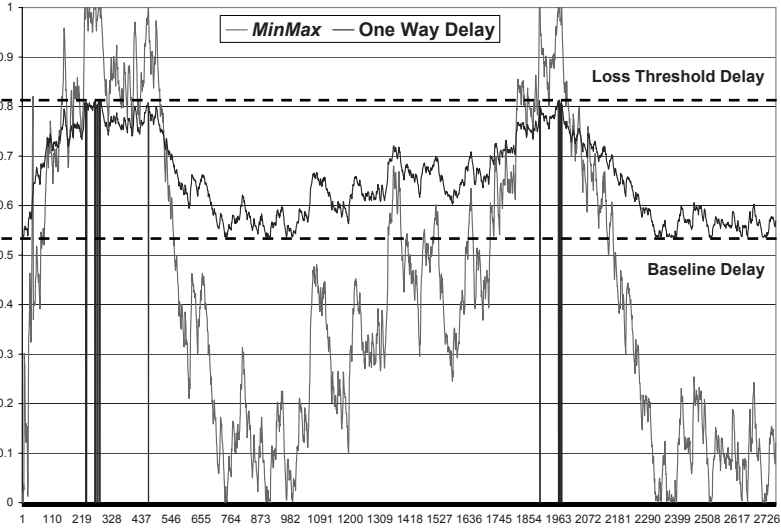


Fig. 1. Behavior of *MinMax*

derived from equation (11). It gives an absolute ratio of the delay value in relation to the loss threshold. The Short term Trend and the Long term Trend metrics are derivations of equation (12). They indicate the sharpness and consistency of upward and downward delay trends over a short and long term window of past packets. Short term trend metric tracks sharp changes in the delay, due to sudden burst of high degree of cross-traffic, which is more critical for loss if the delay is close to the loss threshold. In contrast, the long-term increasing trend tracks gradual rising trends due to persistent decrease in bandwidth, signifying gradual congestion build-up leading to packet loss. By considering the rate of increase, the short-term and the long-term metrics prevent Delay Distance metric from over-reacting to the absolute value of the delay. Thus these metrics work complementary to one another. We formalize the Loss Predictor as a weighted function of these three metrics. The Loss Predictor can be expressed as the following:

$$\begin{aligned}
 0 &\leq f(\text{DelayDist}, \text{ShortTermTrend}, \text{LongTermTrend}) \\
 &= w_1 * \text{DelayDist} + w_2 * \text{ShortTermTrend} + w_3 * \text{LongTermTrend} \leq 1 \\
 &\text{and } w_1 + w_2 + w_3 = 1
 \end{aligned}$$

The Predictor uses dynamic weights that depend on the current delay situation and congestion level. This is described later.

Delay Distance (*MinMax*). This metric gives an absolute ratio of the current delay value above the baseline in relation to the loss threshold. This can be expressed as the following:

$$\text{MinMax} = \min \left(1, \frac{D^k - \text{base}}{\text{thr} - \text{base}} \right) \tag{13}$$

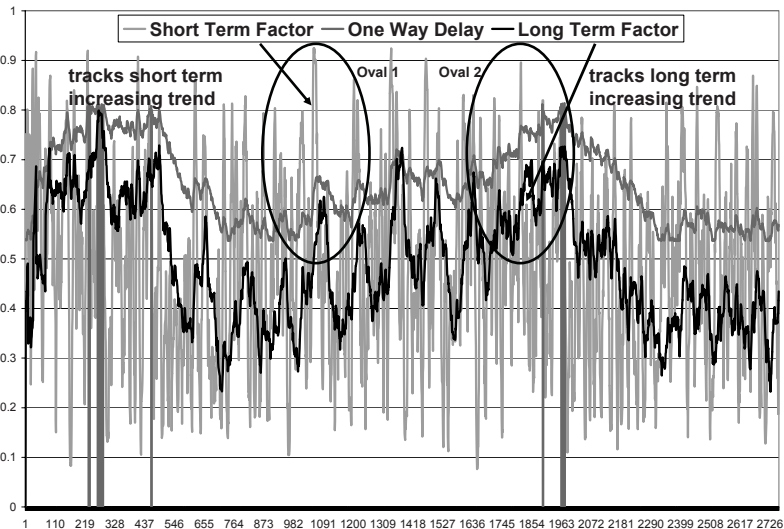


Fig. 2. Behavior of Long Term and Short Term Trends

where, $base$ = the most observed delay so far, considered to be the baseline delay, D^k = the delay value of the k -th packet, thr = the threshold delay at which a loss is observed.

The value of this metric lies between 0 and 1. The minimum of the linear increase ratio and 1 is taken, since we have observed that the maximum delay in a session can increase much higher than the threshold delay, and thus D_k can be larger than thr .

This metric is computationally simple. Fig. 1 shows the behavior of the metric. This value of the metric lies between 0 and 0.2 when the delay is close to 0 and the baseline, but rises up consistently to high values (0.8 to 1.0) when the delay reaches the loss threshold. Thus it is a good indicator of the absolute delay increase or decrease between the baseline and the loss threshold, and hence is an important component of loss prediction.

Long Term Trend Metrics. These metrics are required to measure long-term trends in the delay increase, indicating if an over all increasing trend is evident for a large number of preceding packets. The length of this packet window is adjusted dynamically based on the delay mean deviation observed so far, and is typically 20 to 50 packets. The following metrics indicate consistency in increasing trends, i.e. how consistently the delay is increasing for every measured packet pair, and are useful to measure this trend.

Long Term Consistency Measures ($Spct/Spdt$). We use variations of PCT (Pairwise Comparison Test) and PDT (Pairwise Difference Test) presented in [10]. Both of them are indicators of consistent increasing trends in a packet train. $Spct$ is smoother in showing the long-term trends, whereas $Spdt$ reacts with larger degree of increase or decrease over a packet train length of Γ . The range

of $Spct$ is $[0,1]$ and of $Spdt$ is $[-1,1]$, scaled to $[0,1]$. We use Exponential Weighted Average [9] with gain 0.9 to smooth out undesired oscillations.

$$Spct = \frac{\sum_{k=2}^{\Gamma} I(D^k > D^{k-1})}{\Gamma - 1}, \quad I(X) = 1 \text{ if } X \text{ holds, } 0 \text{ otherwise}$$

$$Spdt = \frac{D^{\Gamma} - D^1}{\sum_{k=2}^{\Gamma} |D^k - D^{k-1}|}$$

We also use the average of *MinMax* for a long term packet window, which provides a long-term trend of the relative increase in delay magnitude. We consider an average of above three measures to calculate the long-term factor. In Fig. 2 oval 2 this metric successfully tracks consistent increasing and decreasing trends of one way delay over 20 to 50 packets.

Short Term Trend Metrics. These metrics are required to measure short-term trends in the delay increase, signifying how fast the delay is increasing over last small window of packets. The length of this packet window is adjusted dynamically based on the mean delay deviation observed so far, and is typically 5 to 10 packets. As the delay gets close to the loss threshold, more weight should be given to these metrics, because a sharp change in the delay will become more critical for loss. We present *SI* as an indicator of sharpness of increase (the ‘slope’ of the increase), and $SpctST/SpdtST$ as indicators of the consistency of increase.

Sharpness Indicator (SI) This metric determines how fast the delay is approaching the loss threshold by measuring the slope of the line joining the delay values of the current packet and the previous packet.

$$SI = \max(-1, \min(1, (D^k - D^{k-1})/(t^k - t^{k-1})))$$

Under a sudden burst of high congestion, the slope is observed to be steeper. Thus higher degree of slope indicates higher congestion, and hence higher likelihood of packet loss. The range of this metric is truncated from $[-\infty, +\infty]$ to $[-1,1]$ and scaled to $[0,1]$.

We also use $SpctST$ and $SpdtST$, which are short term versions of $Spct$ and $Spdt$. As $SpdtST$ is more sensitive to sudden changes to delay, we choose the short-term trend to be the average of SI and $SpdtST$. This metric fluctuates considerably from packet to packet, but successfully tracks the short term increasing and decreasing trends (Fig. 2 oval 1).

3.3 Predictor Weight Factor Formalization

Here we describe the intuitions behind the selection of the metric weights w_1 , w_2 and w_3 described in 3.1. Since *MinMax* is the best indicator of the delay increase in terms of the baseline and loss threshold, w_1 is chosen higher than the short term and the long term trend weights w_2 and w_3 . We divide the range of possible values $[0,1]$ of *MinMax* into three zones ‘Yellow’, ‘Orange’ and ‘Red’, signifying the levels of criticality in regard to packet loss likelihood.

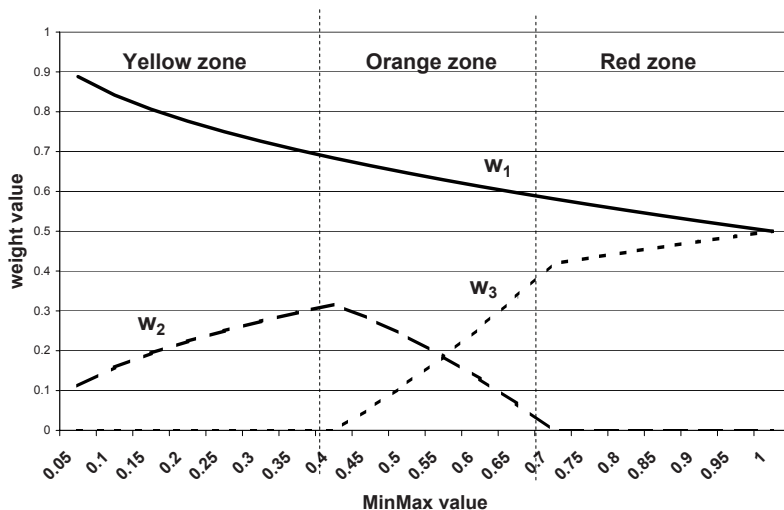


Fig. 3. Weight Function Set

Yellow Zone ($0 \leq MinMax \leq 40\%$): The low value of *MinMax* is not important in this zone, neither are short-term fluctuations. The long-term metric indicates if there is an over all increasing trend, and thus is a significant factor to consider. Thus we choose $w_2 = 0$, and an increasing w_3 .

Orange Zone ($40\% < MinMax \leq 70\%$): This is the critical alert period where we see the signs of delay increase which is of concern, and any sign of sharp increase, showed by the short-term trend, should be considered with increasing weight. Thus we choose an increasing w_2 , and a decreasing w_3 .

Red Zone ($70\% < MinMax \leq 100\%$): In this zone the delay is high enough to be close to the loss threshold. Here every possible sign of short term increasing trend should be taken into account along with the high *MinMax* value. The long term is ignored here. Thus we choose an increasing w_2 , and $w_3 = 0$.

In the following prototype function we use functions w_1 , w_2 and w_3 as weights for *MinMax*, *STermTrend*, *LTermTrend* and a fourth function w_4 to control the behaviors of *STermTrend* and *LTermTrend* following the weight control ‘rules’ in different zones.

$$predictor = w_1 * MinMax + w_2 * STermTrend + w_3 * LTermTrend$$

$$w_1 = 1 - \sqrt{MinMax}/2, w_2 = \sqrt{MinMax}/2 * w_4, w_3 = \sqrt{MinMax}/2 * (1 - w_4)$$

$$w_4 = \begin{cases} 1 & \text{in Yellow zone} \\ 1 - (MinMax - 0.4)/0.3 & \text{in Orange zone} \\ 0 & \text{in Red zone} \end{cases}$$

In Fig. 3 w_1 , w_2 and w_3 follow the weight ‘rules’, with proper increasing and decreasing behaviors in different zones. We give more appreciation to *LTermTrend* by selecting an increasing w_2 in the Yellow zone. But as *Min-*

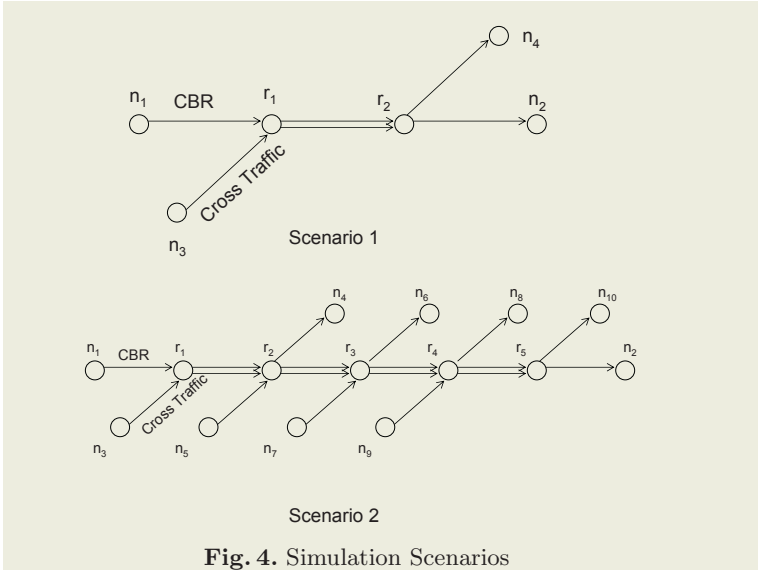


Fig. 4. Simulation Scenarios

Max approaches Orange and Red zones, *STermTrend* is given more priority by choosing an increasing w_3 . This enhances the importance of the trend factors and incorporates the zones in the predictor calculation.

4 Predictor Evaluation: Simulation and Experiment Results

In this section we present the simulation and experiment results that evaluate the Predictor algorithm in terms of *accuracy* and *efficiency*. Ideally the predictor should behave accurately, that is, the predictor should report high values for the majority of packet loss occurrences. The predictor should also be efficient by not over-estimating when there is no loss.

We used the network simulator ns-2 [14] to evaluate the Predictor algorithm simulating congestion under a range of cross-traffic and intermediate hop scenarios. We created two scenarios described in Fig 4. We assumed FCFS queuing and Droptail packet dropping for the links. In scenario 1, a CBR stream of 64kbps flowed from n_1 and n_2 through two intermediate hops, r_1 - r_2 being the bottleneck link (capacity 700kbps) and four Pareto cross traffics with shape parameter=1.9 (meaning infinite variance) flowed from n_3 to n_4 via r_1 - r_2 . In scenario 2, a CBR stream of 64kbps flowed from n_1 and n_2 through five intermediate hops (r_1 , r_2 , r_3 , r_4 and r_5), and four Pareto cross traffics flowed through the intermediate hops as shown in the figure. In both cases we gathered a large number of data by introducing a varied number of transient cross traffics of different packet sizes (100 bytes to 1000 bytes) and rates (200kbps to 900kbps) at dispersed points in time, causing different degrees of congestion resulting in stream packet loss at the intermediate links.

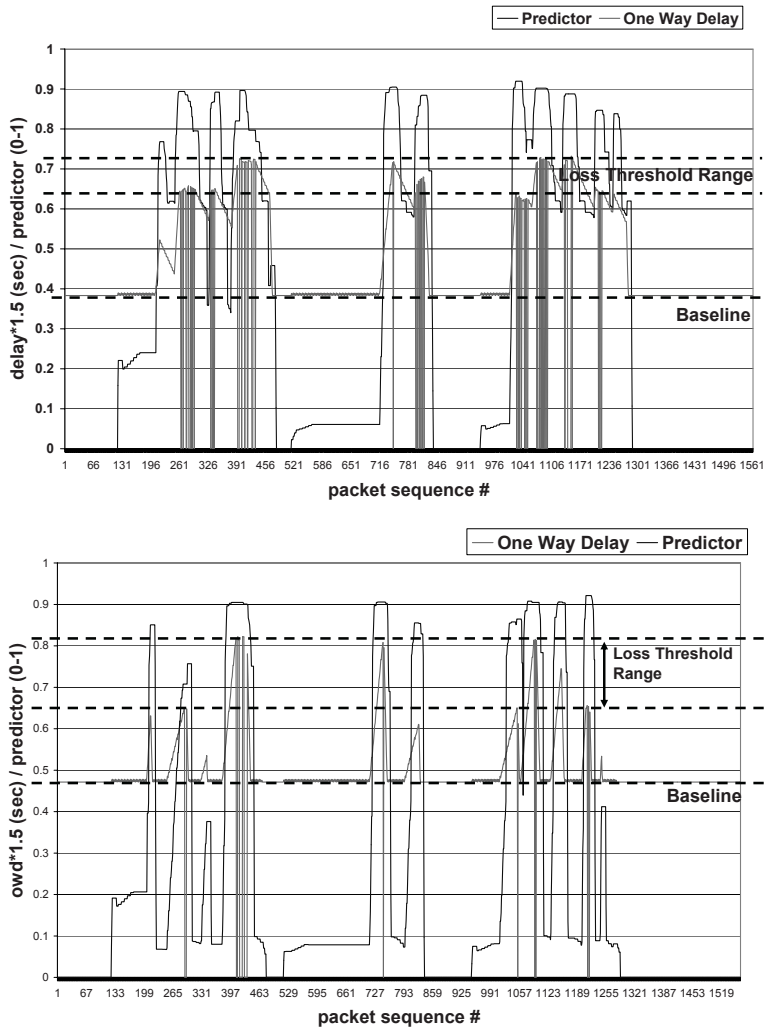


Fig. 5. (a) & (b): OWD, loss threshold and Predictor (Simulation Scenarios 1 and 2)

For the Internet experiments we conducted two sets of experiments on RON/NetBed [7], and PlanetLab [8], two wide-area network testbeds consisting of sites located around the Internet. A "Master" program from a US site sent a 1 minute speech segment to various sites across the world and received it back recording statistical information. The packet size varied between three sizes (256 bytes, 512 bytes, 1K) and the send rate varied among two values (1ms gap, 10ms gap), generating streams of high rates ranging from 204Kbps to 8Mbps in order to create temporary congestion in the path and observe the effects manifested as packet loss. We ran this experiment every 4 hours for a period of seven days.

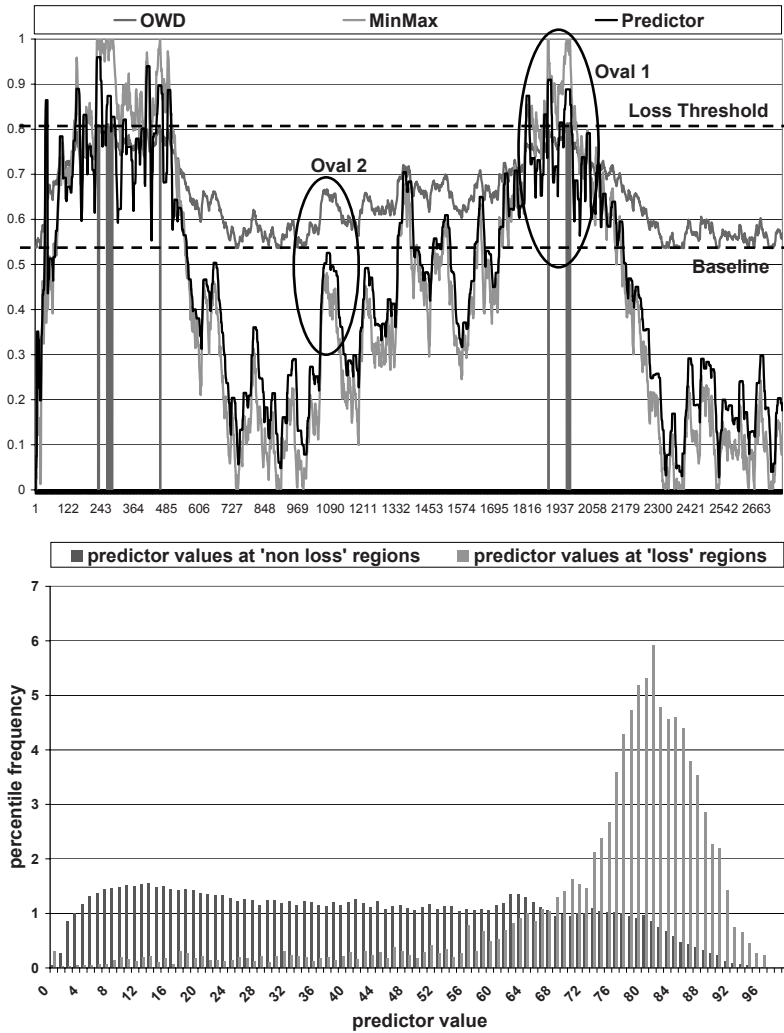


Fig. 6. Predictor behavior in Internet Experiments: (a) Loss Threshold and Predictor: USA-Korea (b) Predictor Accuracy and Efficiency

Predictor Behavior in Simulation and Internet Experiments. In Figs. 5(a) and (b) we present examples of simulation results from scenarios 1 and 2. Both these figures show an identifiable baseline delay. The loss thresholds on the other hand, vary from one another considerably. Scenario 2 has increasingly wider loss threshold range and delay variation compared to scenario 1. This is due to variable and random degrees of cross traffic flowing through multiple links causing unpredictable congestion at different parts of the path. In contrast, congestion at one bottleneck link, though created by various degrees of cross traffic, produces more predictable results in scenario 1. In both scenarios,

Table 1. Predictor Accuracy in Simulation Scenarios

| Predictor value in ‘around loss’ region | Predictor Value Percentage Ratio | |
|---|----------------------------------|------------|
| | Scenario 1 | Scenario 2 |
| > 0.6 | 78% | 59% |
| 0.4 to 0.6 | 20% | 22% |
| ≤ 0.4 | 2% | 19% |

predictor value lies consistently between 0.75 and 0.92 around the loss regions, denoting high likelihood of packet loss, but decreases to low values in the range of 0 to 0.2 when the delay decreases close to the baseline. Thus the predictor successfully reacts to the baseline, loss threshold and increasing and decreasing trends of the delay in a wide variety of congestion situations.

Figure 6(a) shows the Predictor for a selected example from an Internet experiment. We see that around the loss regions the predictor value lies consistently between 0.8 and 0.9, denoting high likelihood of packet loss, but decreases to low values in the range of 0.1 to 0.3 when the delay decreases close to the baseline. Thus the predictor successfully reacts to the baseline, loss threshold and increasing and decreasing trends of the delay. Fig 6(a) also shows the effect of the Short-term and Long-term Trend on the Predictor. In Oval 1, marking a Red zone (*MinMax* value larger than 0.70), the short-term trend accentuates the Predictor based on immediate sharp increase of delay. In Oval 2, the Long-term trend pulls up the Predictor value based on the long-term increasing trend in the Yellow and Orange zones, even when *MinMax* value is not very high.

Predictor Accuracy. We ran the Predictor algorithm against all the data we collected. The data is presented in Figure 6(b) as a summary of the Predictor efficiency and accuracy for Internet experiments. We divided the data into ‘around loss’ and ‘no loss’ regions - an ‘around loss’ region being -20 to +20 packets around a loss, and packets outside ‘around loss’ regions being in ‘no loss’ region. We took predictor values for the packets in the ‘around loss’ regions to determine the predictor accuracy. Ideally, the predictor should predict accurately, that is there should not be any loss at low values of the predictor. We see that there is about 10% loss at low values (values smaller than 0.6).

The accuracy for the simulation is shown in Table 1. Around loss regions, the predictor value is greater than 0.6 78% of the times in scenario 1 vs. 59% in scenario 2. Only 2% of the times the predictor value has been under 0.4 in loss regions, compared to 19% for scenario 2. In scenario 2 we simulate more variable and random degrees of congestion at different parts of the path compared to scenario 1, resulting in less accuracy for the predictor. This motivates us into refining the predictor for better accuracy under such variable conditions.

Predictor Efficiency. Here we study the other side of the same coin, that is, how efficient the predictor is by not over-estimating when there is no loss. Both the simulation scenarios show very small percentages (5% for Scenario 1, 2% for scenario 2) of predictor value larger than 0.7 in the ‘non loss’ region. Figure 6(b)

shows a small percentage (about 16%) of high predictor value (value larger than 0.7) in the non loss region for Internet experiment sites.

The efficiency is a critical input to the Predictor. As part of our future work, we shall use the efficiency factor as a self-feedback to the Predictor mechanism in order for it to evaluate and refine its prediction on an ongoing basis.

5 Conclusion and Future Work

This paper presents a framework of a Packet Loss Predictor: a novel mechanism to predict packet loss in real-time audio streams by observing the delay variation and trends. Loss Predictor approach is based on (i) determining certain metrics for measuring the delay variation characteristics from the ongoing traffic, (ii) combining them with weights based on their importance, and (iii) deriving a predictor value as the measure of packet loss likelihood. The Predictor value, fed back to the sender, indicates current degree and severity of congestion and likelihood of packet loss, and can be a vital component in sender-based error and rate control mechanisms for multimedia. The proactive predictor feedback makes the framework superior to mechanisms with static and reactive feedbacks, and a viable technique for majority of network conditions.

We present simulation and experiment results showing the accuracy and efficiency of the algorithm achieved so far. The results of the Predictor under simulation scenarios and experiments show 60%-90% accuracy and 85%-98% efficiency. As future work, we need to refine the metrics and the weight factors to improve the accuracy and efficiency of the Predictor. This will enable us to use the Predictor more reliably in proactive error and rate control mechanisms. Various packet window sizes introduced in this paper also need to be tuned. Formalization is also necessary for sender initiated proactive FEC and rate control actions, which depend on the Predictor feedback values. Also, we would like to extend the concepts of the Loss Predictor to paths with Random Early Detection (RED) packet dropping mechanisms and apply the Loss Predictor to DiffServ, Overlay and Multicast frameworks.

References

1. Bolot, J., Fosse-Parisis, S., Towsley, D.: Adaptive FEC-Based Error Control for Internet Telephony. *IEEE Infocom* (1999)
2. Bolot, J., Vega-Garcia, A.: Control Mechanisms for Packet Audio in the Internet. *Proceedings IEEE Infocom, San Francisco* (1996) 232-239
3. Brakmo, L.S., O'Malley, S. W., Peterson, L.: TCP Vegas: New Techniques for Congestion Detection and Avoidance. *ACM SIGCOMM Conference* (1994) 24-35
4. Cole, R.G., Rosenbluth, J.H.: Voice over IP Performance Monitoring. *ACM SIGCOMM* (2001)
5. Handley, M.: An examination of Mbone performance. USC/ISI Research Report: ISI/RR-97-450
6. Hermanns, O., Schuba, M.: Performance investigations of the IP multicast architecture. *Proc. IEEE Global Internet Conference* (1996)

7. <https://www.emulab.net/index.php3>
8. <http://www.planet-lab.org/>
9. Jacobson, V.: Congestion avoidance and control. Proc. ACM SIGCOMM Conference, Stanford (1988) 314-329.
10. Jain, M., Dovrolis, C.: End-to-end Available Bandwidth: Measurement Methodology, Dynamics and relation with TCP Throughput. SIGCOMM (2002)
11. Lin, S., Costello, D.J.: Error Correcting Coding, Fundamentals and Applications, Prentice Hall, Englewood Cliffs, NJ (1983)
12. Mohamed, S., Cervantes-Perez, F., Affi, H.: Integrating Network Measurements and Speech Quality Subjective Scores for Control Purposes. IEEE Infocom (2001)
13. Moon, S.B.: Measurement And Analysis Of End-To-End Delay And Loss In The Internet. PhD Thesis, Department of Computer Science, University of Massachusetts, Amherst (2000)
14. The Network Simulator: ns-2: <http://www.isi.edu/nsnam/ns/>
15. Padhye, C., Christensen, K., Moreno, W.: A New Adaptive FEC Loss Control Algorithm for Voice Over IP Applications. Proceedings of the 19th IEEE International Performance, Computing, and Communication Conference (2000) 307-313
16. Paxson, V.: Measurements and Analysis of End-to-End Internet Dynamics. PhD Thesis, Computer Science Division, University of California, Berkeley (1997)
17. Roychoudhuri, L., Al-Shaer, E., Hamed, H., Brewster, G.B.: Audio Transmission over the Internet: Experiments and Observations. IEEE International Conference on Communications (ICC), Anchorage, Alaska (2003)
18. Roychoudhuri, L., Al-Shaer, E., Hamed, H., Brewster, G.B.: On Studying the Impact of the Internet Delays on Audio Transmission. IEEE IP Operations and Management (IPOM), Dallas, Texas (2002)