Timely Detection of Lost Packets in Interactive Media

Ali C. Begen

***Abstract*— Time-constrained error recovery is an integral component of reliable low-delay video applications. Regardless of the error-control method adopted by the application, unacknowledged or missing packets must be quickly identified as lost or delayed, so that necessary actions can be taken by the server/client on time. Historically, this problem has been referred to as retransmission timeout (RTO) estimation. Earlier studies show that existing RTO estimators suffer from either long loss detection times or a large number of spurious timeouts. The goal of this study is to address these problems by developing an RTO estimation method specifically tailored for low-delay video applications. This method exploits the temporal dependence in packet delay to optimally manage the trade-off between the amount of overwaiting and redundant retransmission rate. As opposed to existing methods, our approach is completely adaptive to the source video characteristics and time-varying network conditions, and does not use any preset parameters.**

I. Introduction

THE Internet is a shared medium; any packet injected into the Internet has to wait for some time before it is serviced. It therefore experiences random delay. Because of the finite buffering capabilities of the routers and switches, a packet is assumed to be lost if it has not been received or acknowledged within some time after its transmission. In TCP jargon, this duration is referred to as the retransmission timeout (RTO). It is vital that the value of the RTO is chosen large enough so that the packets experiencing long queueing delays do not trigger spurious timeouts. However, adopting an arbitrarily large RTO is impractical for low-delay video applications. A delayed retransmission attempt eventually recovers a missing media packet. Yet, the chances are that the retransmitted packet will be late and useless for decoding at the client side. Therefore, an RTO estimation method that quickly detects lost packets is imperative for such applications. Only then can well-timed actions be taken for error control.

Naturally, retransmission-based error-control methods are unsuitable for multimedia applications where the extra delay introduced by the retransmissions is prohibitively large. However, in certain circumstances retransmission-based error-control methods can still be accommodated. As a rule of thumb, the client should not time out pre-maturely for the excessively-delayed packets, since under normal circumstances it is highly unlikely that a retransmitted packet will arrive earlier than the initial transmission.

Needless to say, the primary challenge is that the client has to decide on timeouts merely by observing the packet arrivals. It is never a clear-cut decision whether a missing packet has been lost or delayed. Naturally, a trade-off between overwaiting and spurious timeouts is present. In this study, we devise a novel RTO estimation method that involves two main steps. In the first step, an adaptive linear delay predictor produces the best estimate in terms of the mean-squared error criterion by exploiting the temporal dependence among the packet delay samples. In the second step, on the other hand, a controller optimally manages the trade-off between the amount of overwaiting and redundant retransmission rate by regulating the bias to be added to the estimate produced in the previous step. This controller has two different modes of operation: (i) *media-unaware* and (ii) *media-aware*. In the media-unaware mode, the controller ignores the unequal importance of the video packets and treats each of them equally. In the media-aware mode, however, the controller prioritizes the packets that carry a more important payload and the packets whose decoding deadlines are sooner, over the less important and non-urgent packets.

Our approach has three main contributions:

- We develop an adaptive delay predictor for high-bitrate video applications. A large number of multimedia protocols such as packet scheduling algorithms, congestion control algorithms and adaptive buffer management techniques can potentially benefit from this predictor.
- We derive an optimal media-unaware redundancy-controllable timeout estimator. This estimator allows applications to recover as many packets as possible under a given redundant rate budget.
- We formulate an optimal media-aware timeout estimator that jointly considers the interdependency relations among the video packets as well as their decoding deadlines in computing the timeout estimates while still conforming to the redundant rate constraint dictated by the application or the network[1].

It is important to note that not all of the error-control methods are necessarily retransmission-based. Precise RTO estimation is also useful in applications employing different types of error-control/protection methods. For example, based on the delay/loss predictions, the amount of redundancy in channel coding or the amount of error resiliency in video coding can be optimally adjusted to minimize the impact of packet erasures. Accurate delay prediction is also essential for an effective congestion control algorithm.

This contribution summarizes some of the findings and results that have been previously published in [1]. For detailed analysis and more results, refer to [1].

[1]The discussion related to media-aware timeout estimation is omitted in this contribution. Interested readers are referred to Section V of [1] for a detailed description.

II. Autoregressive Models for Packet Delay

A. Adaptive Linear Delay Prediction

Let us consider a stochastic process $\mathbf{s}$ and let $\mathbf{s}[n-k]$, $k \geq 1$ denote the past samples of this process. The operation of linear prediction expresses the value of $\mathbf{s}[n]$ as the linear combination of the samples $\mathbf{s}[n-k]$. The estimate based on the N most recent values is given by

$$\tilde{\mathbf{s}}_N[n] = E\left\{\mathbf{s}[n]|\mathbf{s}[n-k], 1 \leq k \leq N\right\} = \sum_{k=1}^{N} \alpha_{k,N}\mathbf{s}[n-k]. \tag{1}$$

This estimate is called the one-step forward predictor of order N. Our objective in prediction is to determine the constants $\alpha_{k,N}$ so as to minimize the mean square value of the forward prediction error $\boldsymbol{\epsilon}_N[n] = \mathbf{s}[n] - \tilde{\mathbf{s}}_N[n]$. From the orthogonality principle, we know that the prediction error, *i.e.,* $\boldsymbol{\epsilon}_N[n]$, is orthogonal to all data used to generate the prediction, *i.e.*, $\mathbf{s}[n-m]$, where $1 \leq m \leq N$. Mathematically, we have

$$E\left\{\left(\mathbf{s}[n] - \sum_{k=1}^{N} \alpha_{k,N}\mathbf{s}[n-k]\right)\mathbf{s}[n-m]\right\} = 0 \quad 1 \leq m \leq N, \tag{2}$$

which yields a set of linear equations known as the Yule-Walker equations. The coefficients $\alpha_{k,N}$ of the predictor filter $\mathbf{H}_N(z)$ can be computed from

$$R[m] - \sum_{k=1}^{N} \alpha_{k,N} R[m-k] = 0 \quad 1 \leq m \leq N, \tag{3}$$

where $R[q]$ represents the lag-q autocorrelation of $\mathbf{s}$.

The predictor filter coefficients can be easily computed by the Durbin-Levinson recursion. As the order N of prediction increases, the value of the mean prediction-error power P_N decreases or else remains the same. Since prediction-error power is always positive, we have

$$P_1 \geq P_2 \geq \ldots \geq P_N \xrightarrow[N\to\infty]{} P \geq 0. \tag{4}$$

The implication of (4) is that as we increase the order of the predictor filter $\mathbf{H}_N(z)$, we successively reduce the correlation between the adjacent samples of the input process until we ultimately reach a point at which increasing the order of prediction any further does not reduce the prediction-error power. At this point, the error is a white noise process and consists of purely uncorrelated samples.

Suppose that $P_{M-1} > P_M$ and $P_M = P_{M+1} = \ldots = P$. By definition, the process $\mathbf{s}$ is called an M^{th}-order autoregressive, denoted by AR(M), process or a wide-sense Markoff process of order M. For this process, the M^{th}-order predictor, $\tilde{\mathbf{s}}_M[n]$, equals to its Wiener predictor. Wiener predictors produce the best fit to the observed data by exploiting the existing correlation completely. However, due to their high complexity and low predictive accuracy, Wiener predictors are usually not used in practice.

B. Model Selection

Generally, it is desirable to have the values predicted by a model to be close to the actual data values. As pointed out by (4), increasing the order of prediction naturally produces better estimates and a lower prediction-error power. However, an overfitted model may not distinguish the systematic effects of the data from its random effects. For practical purposes, we seek a model that yields a high predictive accuracy with the smallest number of parameters. A popular model selection method is the Akaike's Information Corrected Criterion (AICC). The AICC score quantifies the relative goodness-of-fit of a statistical model for the given data. A lower AICC score indicates a better prediction model.

Let us illustrate the importance of model selection on three packet delay traces. To generate these traces, we simulated a moderate-sized Internet topology in *ns*-2 network simulator and used video streams that were encoded by a standard H.264 codec at 300 Kbps, 600 Kbps and 1.2 Mbps. We will refer to these delay traces with the notation of $\Delta T = 40$ ms, $\Delta T = 20$ ms and $\Delta T = 10$ ms, respectively, where ΔT denotes the average transmission interval at the server.

First, we examine the relation between the mean prediction-error power and the order of prediction. The order of Wiener prediction for the $\Delta T = 10$ ms, $\Delta T = 20$ ms and $\Delta T = 40$ ms traces is found to be 12, 32 and 60, respectively. Clearly, we require a higher order of prediction for larger ΔT. This is not surprising since the correlation between the adjacent delay samples reduces with ΔT. The mean prediction-error power gradually decreases with the order of prediction. However, the AICC scores first show a decreasing and then an increasing trend. In other words, the predictive accuracy improves with increasing N until a point and then starts to degrade. Specifically, the AICC scores for the $\Delta T = 10$ ms, $\Delta T = 20$ ms and $\Delta T = 40$ ms traces reach their global minima at $N = 3$, $N = 9$ and $N = 12$, respectively. These values are comparably smaller than the ones corresponding to the Wiener prediction, signifying that Wiener predictors are indeed overfitted and have sub-optimal predictive accuracy.

C. Practical Considerations

Generally speaking, the AICC method suggests good models that provide sufficient insight into the process being analyzed, while leaving out the random effects. Here, the main objective is to select a computationally-efficient yet intuitively plausible prediction model that adequately captures the dynamics in the packet delay process.

A *naive* approach is the AR(1) model, where the next delay estimate is solely determined by the last observation, *i.e.,* $\tilde{\mathbf{s}}_1[n] = \mathbf{s}[n-1]$. The phase diagrams plotted in Fig. 1 clearly indicate the existence of a significant lag-1 correlation among the delay samples and support the AR(1) prediction model. However, this predictor is not capable of distinguishing whether packet delays are increasing, decreasing, or remaining the same, and therefore, does not serve our goal.

A more elaborate model is the AR(2) model. AR(2) model bases its estimation on the last two observations. By definition,

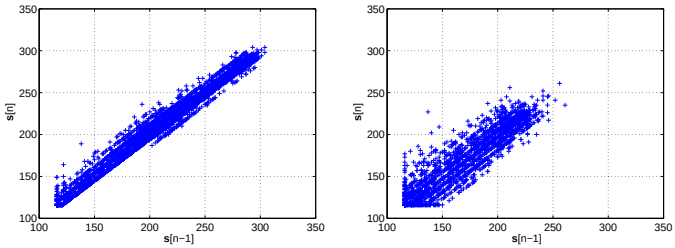

Fig. 1. Phase diagrams for $\Delta T = 10$ ms (on the left) and $\Delta T = 40$ ms (on the right).

we have

$$\tilde{\mathbf{s}}_2[n] = \alpha_{1,2}\mathbf{s}[n-1] + \alpha_{2,2}\mathbf{s}[n-2], \tag{5}$$

which can be rewritten as

$$\tilde{\mathbf{s}}_2[n] = (\alpha_{1,2}+\alpha_{2,2})\mathbf{s}[n-1] + \alpha_{2,2}\left(\Delta T - \Delta t[n-1]\right), \tag{6}$$

where $\Delta t[n]$ denotes the interarrival time for packet n, *i.e.*, the time difference between the arrivals of packets n and $n-1$. The interpretation of (6) is that the AR(2) model takes into consideration not only the last delay sample but also its deviation from the previous sample.

To understand how well an AR(2) predictor compares to its Wiener counterpart, we plot the prediction-error autocorrelation functions (ACF) for both predictors. Since Wiener predictors completely model the data, the resulting error samples are guaranteed to be uncorrelated, which is, however, not necessarily true for AR predictors of lower orders. Nevertheless, Fig. 2 shows that the correlation left out by the AR(2) predictors is rather insignificant, implying that AR(2) predictors have sufficient prediction accuracy for practical purposes.

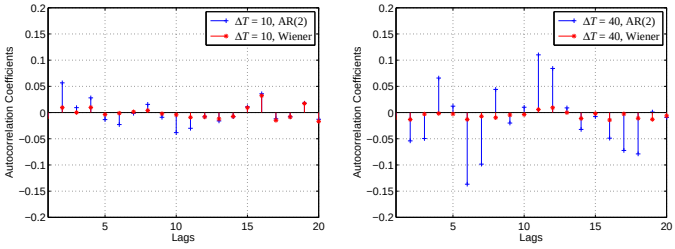

Fig. 2. ACFs of the prediction errors produced by Wiener and AR(2) predictors for $\Delta T = 10$ ms (on the left) and $\Delta T = 40$ ms (on the right).

III. Media-Unaware Timeout Estimation

When minimizing the mean square value of the prediction error, an underestimate that is marginally smaller than the actual value is as good as an overestimate that is marginally larger than the actual value. However, in the context of RTO estimation, underestimations trigger pre-mature timeouts whereas overestimations eliminate them. In this section, we formulate a computationally-efficient way to compute the minimum amount of additional waiting that is required to keep the probability of a pre-mature timeout below a desired value.

A. Methodology

In Fig. 3, we plot the prediction-error distributions for the $\Delta T = 10$ ms and $\Delta T = 40$ ms traces. We notice that each of these distributions (particularly, the tail parts) can be approximated by a Gaussian distribution whose mean and standard deviation (σ) are equal to those of the corresponding prediction-error distribution. Statistically, Gaussian-distributed samples of a white noise process are independent of each other. In the light of Fig. 2, we infer that AR(2) predictors produce error samples that are independent. This result has two important implications: First, a sequence of independent random variables is not predictable by linear or non-linear models. Thus, if packet delay sampling is sufficiently dense, the delay process can be almost completely characterized by an AR(2) model. Second, Gaussian-distributed processes are easy to work with and a rich set of mathematical tools is available.

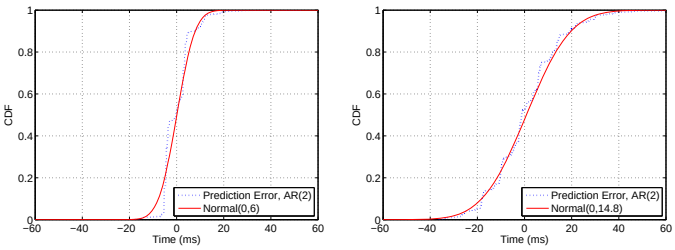

Fig. 3. Prediction-error distributions for $\Delta T = 10$ ms (on the left) and $\Delta T = 40$ ms (on the right). Plots do not include the lost packets.

Let τ denote the additional amount of waiting to be added to the initial delay predicted by (6), and let $\Phi(\tau)$ denote the underestimation probability. By definition,

$$\Phi(\tau) = P\left\{\tilde{\mathbf{s}}_2[n] + \tau < \mathbf{s}[n]\right\}, \tag{7}$$

which is a non-increasing function of τ. We seek the minimum value for τ that satisfies

$$\Phi(\tau) \leq p_f, \tag{8}$$

where p_f is the desired probability of timing out pre-maturely. By rewriting $\Phi(\tau)$ as $P\{\tau < \boldsymbol{\epsilon}_2\}$, we compute τ from

$$\tau = F^{-1}_{\boldsymbol{\epsilon}_2}(1-p_f), \tag{9}$$

where $F_{\boldsymbol{\epsilon}_2}$ is the cumulative density function of $\boldsymbol{\epsilon}_2$. A nice feature of the Gaussian distribution is that its inverse cumulative function can be directly calculated from the first and second-order moments. For example, to limit the rate of pre-mature timeouts to 5%, τ should be set to 1.65 times the standard deviation, which is $1.65 \times 14.8 = 25$ ms for the $\Delta T = 40$ ms trace. While 25 ms may seem insignificant, τ quickly increases for lower p_f values, *e.g.*, for $p_f = 0.1\%$, the required amount increases to 46 ms.

The adverse impact of large τ values is the increase in the time required to detect lost packets. To quantify the detection time of a lost packet, we use the delay of the last successfully-received packet as the hypothetical delay for the lost packet. The loss detection time is then given by the difference between the predicted and the hypothetical delays. That is,

$$w[n] = \tilde{\mathbf{s}}_2[n] + \tau - \mathbf{s}[n^*], \quad \forall n : \mathbf{s}[n] = \infty, \tag{10}$$

where n^* is the last successfully-received packet. The average loss detection time and the pre-mature timeout probability are the benchmarks that characterize the performance of an RTO estimator.

B. Simulation Results

In this section, we present several *ns*-2 simulation results and evaluate the performances of four different RTO estimators: (i) the enhanced TCP-like RTO estimator, denoted by $\mathrm{RTO}_{\mathrm{E-TCP}}$, (ii) recursive weighted median filtering, denoted by $\mathrm{RTO}_{\mathrm{RWM}(1,5)}$, (iii) a percentile-based RTO estimator that predicts the forward-trip time (FTT) of the next expected packet by computing the p^{th}-percentile of the FTT histogram (excluding the lost packets), denoted by $\mathrm{RTO}_{\mathrm{PRC}}$, and (iv) the media-unaware RTO estimator, denoted by $\mathrm{RTO}_{\mathrm{AR}(2)}$. Detailed results about each RTO estimator can be found in Section IV.B of [1].

We compare $\mathrm{RTO}_{\mathrm{E-TCP}}$, $\mathrm{RTO}_{\mathrm{PRC}}$ and $\mathrm{RTO}_{\mathrm{AR}(2)}$ on the $p_f - w$ plane. Since the loss-detection performance of $\mathrm{RTO}_{\mathrm{RWM}(1,5)}$ is the worst by a large margin, we omit it from this comparison. Here, we are interested in determining which RTO estimator detects the lost packets in the shortest amount of time without exceeding a given pre-mature timeout probability. Fig. 4 shows that $\mathrm{RTO}_{\mathrm{AR}(2)}$ substantially outperforms $\mathrm{RTO}_{\mathrm{E-TCP}}$ in all cases. For the $\Delta T = 10$ ms and $\Delta T = 20$ ms traces, $\mathrm{RTO}_{\mathrm{AR}(2)}$ also achieves a better performance than $\mathrm{RTO}_{\mathrm{PRC}}$. However, in the $\Delta T = 40$ ms trace, $\mathrm{RTO}_{\mathrm{PRC}}$ detects the lost packets 8 - 20 ms faster than $\mathrm{RTO}_{\mathrm{AR}(2)}$ at regions where $p_f > 0.6\%$. Nevertheless, at the expense of a 20 ms increase in the average loss detection time, $\mathrm{RTO}_{\mathrm{AR}(2)}$ is able to diminish the pre-mature timeout rate to 0.1%.

One important issue in RTO estimation is the rapid convergence of the timeout estimates. Based on our simulations, we found that $\mathrm{RTO}_{\mathrm{E-TCP}}$ required at least 15-20 samples to produce good estimates. Thus, when the network conditions changed rapidly, $\mathrm{RTO}_{\mathrm{E-TCP}}$ largely failed. This problem was solved to some extent by $\mathrm{RTO}_{\mathrm{RWM}(1,5)}$, which only required five samples to produce an estimate. On the other hand, $\mathrm{RTO}_{\mathrm{PRC}}$ initially required several samples to be able to work properly. In contrast, $\mathrm{RTO}_{\mathrm{AR}(2)}$ required only the last two samples for RTO estimation. This fast-convergence feature provides $\mathrm{RTO}_{\mathrm{AR}(2)}$ robustness when the packets continuously experience a large amount of jitter, or when only a small number of delay samples are available for RTO estimation.

IV. Conclusions

Our findings can be summarized as follows:

- RTO estimation is one of the most important components of any error-control/protection method and congestion control algorithm. A good RTO estimator should be able to quickly identify lost packets under rigid delay requirements. This allows the applications to react to the congestion events faster and better.
- Provided that the packets are transmitted at sufficiently short intervals, consecutive delay samples show a strong correlation. Wiener predictors can be used to fully exploit this correlation and produce uncorrelated prediction-error samples. We showed that these uncorrelated error samples could be modeled by a Gaussian distribution, implying that the error samples were indeed independent. Thus, Wiener prediction models can completely characterize the packet delay process. We also showed that AR(2) predictors could be safely used in practice instead of their Wiener counterparts.
- Adaptivity to time-varying network conditions, *e.g.*, timely reaction to congestion events, is the key in successful RTO estimation. Slow adaptation potentially leads to a significant performance degradation in terms of redundant/late retransmissions which might make the congestion worse.

References

[1] A. C. Begen and Y. Altunbasak, "An adaptive media-aware retransmission timeout estimation method for low-delay packet video," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 332–347, Feb. 2007.

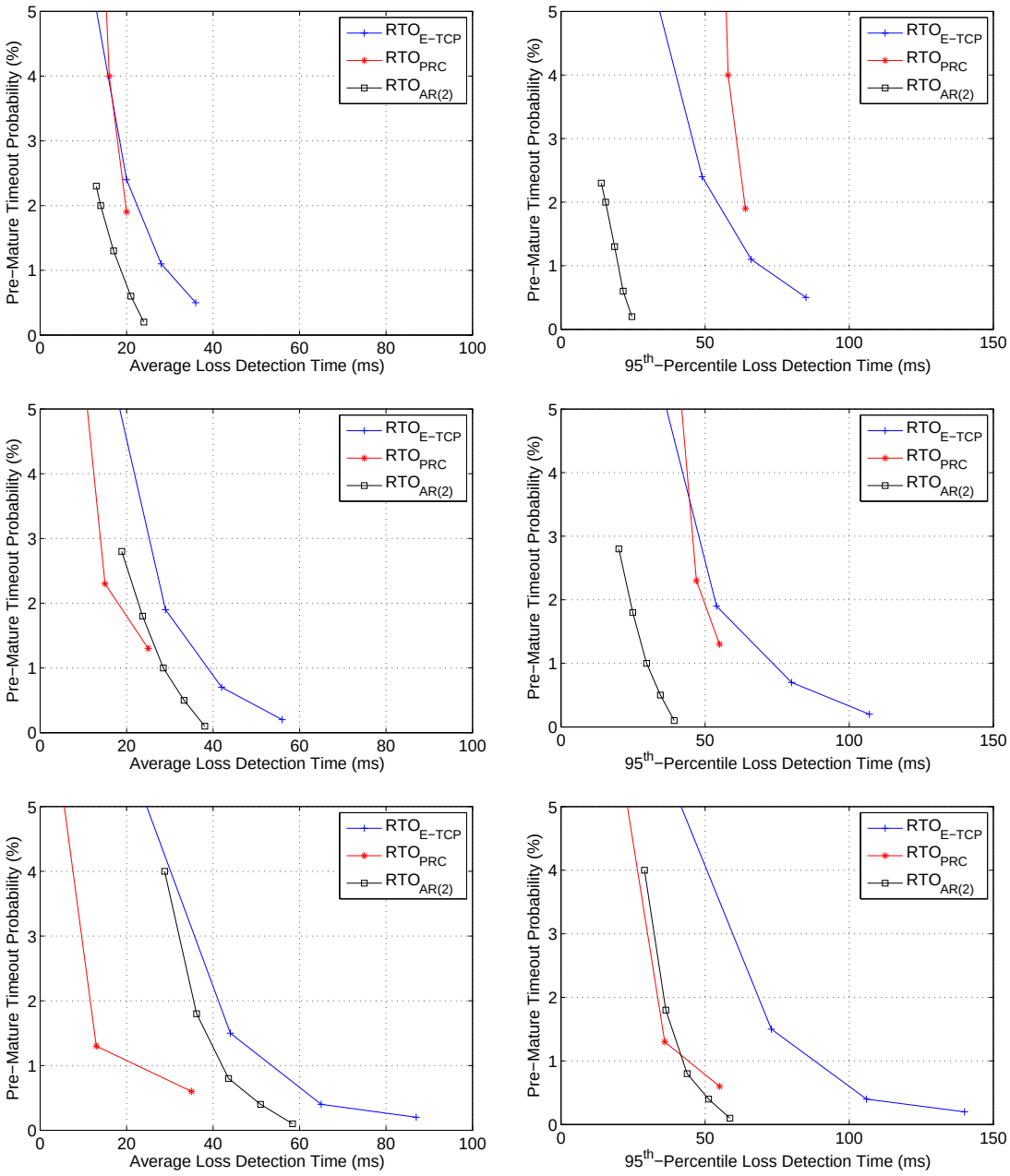

Fig. 4. Comparison of RTO estimators: $\Delta T = 10$ ms (top), $\Delta T = 20$ ms (middle), $\Delta T = 40$ ms (bottom).