Internet Roundtrip Delay Prediction Using the Maximum Entropy Principle

Peter Xiaoping Liu, Max Q-H Meng, and Jason Gu

Abstract: **Internet roundtrip delay/time (RTT) prediction plays an important role in detecting packet losses in reliable transport protocols for traditional web applications and determining proper transmission rates in many rate-based TCP-friendly protocols for Internet-based real-time applications. The widely adopted autoregressive and moving average (ARMA) model with fixed-parameters is shown to be insufficient for all scenarios due to its intrinsic limitation that it filters out all high-frequency components of RTT dynamics. In this paper, we introduce a novel parameter-varying RTT model for Internet roundtrip time prediction based on the information theory and the maximum entropy principle (MEP). Since the coefficients of the proposed RTT model are updated dynamically, the model is adaptive and it tracks RTT dynamics rapidly. The results of our experiments show that the MEP algorithm works better than the ARMA method in both RTT prediction and RTO estimation.**

Index Terms: **Internet, delay, prediction, maximum entropy principle.**

I. INTRODUCTION

For today's Internet, a best-effort packet-switching network, data packets may get lost or corrupted somewhere on the route from the source to the destination due to innumerous reasons, such as network congestion and component failure. Dynamically estimating the timer, i.e., the roundtrip timeout (RTO), is a key function in many reliable transport protocols to detect packet loss and determine whether the network is congested. For example, the TCP protocol guarantees reliable delivery by employing a retransmission mechanism: When a packet is unacknowledged within the RTO, it is assumed that the packet is lost and a retransmission is triggered [1], [2]. Thus, proper RTO is important to achieve high network throughput: If RTO is too large, network clients suffer from needless long waits before retransmitting lost packets; on the other hand, if the RTO is too small, it will reduce the effective network throughput due to unnecessary retransmission [3] Since the RTO (timer) for the next packet is computed from the estimate of the RTT, inaccurate RTT prediction probably leads to incorrect timers. Consequently, to attain a proper timer, it is crucial that we shouldto first have an accurate predictor for RTT.

Manuscript received February 2, 2002; approved for publication by Saewoong Bahk, Division III Editor, November 30, 2002.

P. X. Liu is with the Department of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada K1S 5B6, e-mail: xpliu@sce.carleton.ca.

M. Q-H Meng is with the Department of Electronic Engineering, Chinese University of Hong Kong, Shatin, Hong Kong, e-mail: max@ee.cuhk.edu.hk.

J. Gu is with the Department of Electrical and Computer Engineering, Dalhousie University, Halifax, Nova Scotia, Canada B3J 2X4, e-mail: jason.gu@dal.ca.

As underlying Internet technologies continue to advance rapidly, more and more realtime applications, such as network conferencing, multimedia and teleoperation, are appearing on the Internet. For these applications, timely delivery is critical and the conventional TCP protocol is unsuitable. Currently, significant efforts are being devoted to developing new TCP-friendly transport protocols which are suitable for real-time tasks, such as the TFRC (TCP-Friendly Rate Control) protocol [4], the RAP (Rate Adaptation Protocol) [5] and the trinomial protocol [6]. For these protocols that are usually rate-based by nature, accurately estimating RTT is particularly important since the values of RTT estimates determine the transmission rate of the protocol directly. For the RAP protocol, for instance, the sending rate is controlled by adjusting the value of the *inter-packet-gap* (IPG) at the source, which relies on an accurate prediction of RTT to determine the IPG value, the interval between two succeeding packets. For example, if the RTT estimate is 1 second and the transmission rate is 5 packets per RTT, then the value of IPG is 0.2 second. Consequently, smaller IPGs mean higher transmission rate while larger IPGs indicate lower transmission rate. In other words, too large RTT estimates lead to a conservative protocol for which network bandwidth is used ineffectively. On the contrary, too small RTT estimates give rise to an aggressive protocol, which introduces unfairness among competing flows and is harmful to network stability [7]. Of course, for these real-time protocols, RTT prediction is also necessary to estimate the timer in determining whether the network is congested just like in TCP.

In addition, for applications such as Internet-based teleoperation [8], various sensory sources (visual, range and haptic, for instance) about the remote scene may not arrive at the human operator's end simultaneously. A proper estimate of RTT is desirable for the synchronization and fusion of different data sources when the information is presented to the human operator. It is also quite helpful for designing teleoperation compensation control algorithms and analyzing system stability if RTT is known in advance.

Due to its key importance as discussed above, RTT prediction has been studied for quite a long time [2], [3], [9]–[16]. Before we proceed further, let us first review the widely adopted RTT estimation algorithm – an autoregressive and moving average (ARMA) model with *fixed* parameters. The parameter-fixed ARMA model has been well acknowledged because of its low computing cost and more of Jacob's implementation of TCP: Jacob successfully solved the problem of frequent network congestion and collapse owing to unnecessary retransmission by employing the ARMA model for RTT estimation in 1980s [2]. For convenience, throughout the rest of the paper, we refer to R_n as the RTT measurement of the n^{th} data packet, $\hat{R}_n$ as the RTT prediction of the n^{th} data packet, and $\hat{\sigma}_n$ as the prediction of the RTT variation of the n^{th} data packet. In Jacob's TCP implementation, the estimate of RTO for the next packet, RTO_{n+1}, is given by

$$RTO_{n+1} = \hat{R}_{n+1} + \beta \times \hat{\sigma}_{n+1}, \quad (1)$$

where β is a constant variance factor that is 4, and $\hat{R}_{n+1}$is computed from

$$\hat{R}_{n+1} = (1-\alpha)\hat{R}_n + \alpha \times R_n, \quad (2)$$

which is a least mean-square error predictor based on an ARMA process where α is a constant smoothing factor that is between 0.1 and 0.2 [1]. Jacob calls $\hat{R}_{n+1}$ the smoothed RTT (SRTT) in [2].

We visit the topic of RTT estimation mainly for three reasons. First, we have developed a new TCP-friendly transport protocol called the trinomial protocol for Internet-based teleoperation in which the transmission rate is a function of RTT estimation [6]. Though the protocol is theoretically TCP compatible, we found from the simulation that it was hard for the protocol to be TCP compatible by employing the ARMA model with fixed parameters. This is the primary motivation of our efforts to look for new algorithms for RTT estimation. Secondly, since there has been an explosive growth in the size, types and complexity of the Internet since 1990s, the models of RTT dynamics and the network in the literature may not be applicable to today's Internet. Thirdly, the ARMA model with *fixed* parameters has inherent limitations and is not sufficient for all scenarios [3], [5], [6], [9], and [10]. In Jacob's TCP implementation, for example, to avoid unnecessary retransmission, the inaccuracy of the RTT estimate from the ARMA model is absorbed by choosing a quite large variance factor ($-\beta = 4$) in the process of estimating RTO. However, in many rate-based protocols for real-time applications, the estimated RTT determines transmission rate directly as mentioned earlier. The weakness of the parameter-fixed ARMA method is shown by the implementations of both the RAP protocol [5] and the trinomial protocol [6].

The shortcomings of the conventional ARMA model lie in the fact that it filters out all the high-frequency components of RTT dynamics and thus is incapable of tracking rapid RTT variation. By doing some mathematical manipulations, the ARMA model shown by (2) can be converted into the following form [10]:

$$\hat{R}_{n+1} = M + e_{n+1} a \sum_{k=1}^{\infty} e_{n-k+1}, \quad (3)$$

where M is a constant representing the mean of RTT; e_i, $i = -\infty, \cdots, n, n+1$, is a sequence of independent, identically distributed (iid) random variables with zero mean and a certain variance. From (3), we see that: As $\alpha \rightarrow 1$, $\hat{R}_{n+1}$ is modeled by a Brownian motion, which is a nonstationary process; as $\alpha \rightarrow 0$, $\hat{R}_{n+1}$ is characterized by a base plus a white noise process, which is a stationary process; as $0 < \alpha < 1$, $\hat{R}_{n+1}$ is the sum of a white noise process and a weighted Brownian motion process. First, with such a small and fixed value of $\alpha (\alpha = 0.1 \backsim 0.2)$, this model presumes implicitly that RTT dynamics is dominated by a slow stationary noise process and independent of concurrent network states, such as load, congestion, routing and path etc. However, we know that when the network is heavily loaded or congested, RTT grows rapidly and significantly [3], [12]. Accurate RTT prediction is particularly essential for rate-based protocols to be TCP-friendly when the network is congested. Secondly, RTT potentially varies largely and quickly [3], [13], and [17]. For example, since IP is a datagram protocol with dynamic routing, packets may be routed through different paths from source to sink dynamically with quite different delays. Thirdly, due to the rapid change of connection types and significant increase in the number of users, the RTT dynamics of today's Internet is quite complicated [9], [10]: A measured RTT sequence may be governed by a noise process in some intervals, but may show strong structures in others. These characteristics are not well fitted by models with *fixed* parameters. In addition, the conventional ARMA method does not work with short-lived connections, in which the senders and receivers only exchange a small number of packets [3], [10].

In this paper, rather than sticking to parameter-fixed models such as the conventional ARMA model, we introduce a model-adaptive algorithm for RTT prediction based on the information theory and the maximum entropy principle (MEP) inspired by Klan and Darbellay's work [18]. The rest of the paper is organized as follows. In Section II, we present a *parameter-varying autoregressive* (AR) RTT model in which the estimated RTT is regarded as a linear combination of past observations. The prediction problem thus is figuring out a way to get the coefficients of the AR model. Section III describes the key idea of the proposed MEP algorithm and shows how to update the coefficients of the RTT model adaptively and dynamically. In Section IV, the performance of the MEP algorithm is examined and compared with that of the traditional ARMA method. Section V summarizes the paper with conclusions.

II. PROPOSED PARAMETER-VARYING RTT MODEL

The RTT prediction issue can be simply expressed as: Given the knowledge of current RTT and $k-1$ past RTT observations, $\{R_1 i = n, n-1, \cdots, n-k+1\}$, what is the most likely value of the next RTT, $\hat{R}_{n+1}$. This is a univariate short-term (one-step ahead) forecasting problem for which the predicted RTT, $\hat{R}_{n+1}$, can be regarded as a function of k currently available observations, i.e.,) $\hat{R}_{n+1} = f(R_n, R_{n-1}, \cdots, R_{n-k+1})$, where f is unknown and k is the order, or 'memory,' of the model. The forecasting problem is actually to choose an optimal f based on some criteria so that $\hat{R}_{n+1}$ is as close as possible to R_{n+1} that has not occurred.

First let us consider a tentative linear AR model for RTT dynamics, where $\hat{R}_{n+1}$ is a convex combination of k observations in the form:

$$\hat{R}_{n+1} = a_{n1}R_n + a_{n2}R_{n-1} + \cdots + a_{nk}R_{n-k+1}, \quad (4)$$

where $a_{n1}, a_{n2}, \cdots, a_{nk}$ are the coefficients of the model. The lower index n expresses the *time-dependence* of the coefficients. The coefficients satisfy the following condition:

$$\sum_{i=1}^{k} a_{ni} = 1, \ \ a_{ni} \geq 0. \quad (5)$$

The constraint shown in (5) makes the RTT model shown in (4) different from ordinary AR models. However, the inappropriate of the RTT model is evident. From (5), we know that $\hat{R}_{n+1}$ can lie only in the interval $\lfloor a_n, b_n \rfloor$, where $a_n = \min\{R_n, R_{n-1}, \cdots, R_{n-k+1}\}$ and $b_n = \min\{R_n, R_{n-1}, \cdots, R_{n-k+1}\}$. Hence, the interval, $\lfloor a_n, b_n \rfloor$, should be extended to be practicable. For this purpose, two more terms, R_n^L and R_n^H , such that $R_n^L \leq a_n$ and $R_n^H \geq b_n$, are added into (4). The RTT model then becomes

$$\begin{aligned}\hat{R}_{n+1} &= a_{n1}R_n + a_{n2}R_{n-1} + \cdots \\ &\quad + a_{nk}R_{n-k+1} + a_{n(k+1)}R_n^L + a_{n(k+2)}R_n^H, \quad (6)\end{aligned}$$

and the constraint on the coefficients now becomes

$$\sum_{i=1}^{k+2} a_{ni} = 1, \quad a_{ni} \geq 0. \qquad (7)$$

Now, we assume that the coefficients of the model shown in (6) still possess validity when they are used to calculate R_n with the real time series, $\{R_i, \quad i = n-1, n-2, \cdots, n-k\}$ The following reflex condition is used for this purpose:

$$\begin{aligned}R_n &= a_{n1}R_{n-1} + a_{n2}R_{n-2} + a_{nk}R_{n-k} \\ &\quad + a_{n(k+1)}R_n^L + a_{n(k+2)}R_n^H. \qquad (8)\end{aligned}$$

This assumption is well justified in [18]–[20]. From (7), we know that all the coefficients, $a_{ni}, \quad i = 1, 2, \cdots, k+2$, are positive numbers and the sum of them is equal to 1. Hence, the coefficients can be regarded as a set of probability distributions over $R = \{R_{n-1}, R_{n-2}, \cdots, R_{n-k}, R_n^L, R_n^H\}$ and the reflex condition, (8), can be understood as the constraint on the set of probability distributions (This is the key idea of the proposed method). If this is the case, we can use the maximum entropy principle to update the coefficients at each step. Because the coefficients of the RTT model are updated dynamically, the model is parameter-varying and adaptive. In the next section, we show how to calculate the coefficients by using the maximum entropy principle.

III. MAXIMUM ENTROPY PRINCIPLE (MEP) ALGORITHM

The reflex condition specified by (8) is the only relevant information available on R_{n+1} at the instant of packet n and there are infinite distributions satisfying (8). According to the information theory, the maximum entropy principle is the most unbiased prescription to choose the coefficients, $a_{ni}, i = 1, 2, \cdots, k+2$, for which the Shannon entropy, i.e., the expression:

$$H(a_n) = -\sum_{i=1}^{k+2} a_{ni} In a_{ni} \qquad (9)$$

is maximal under the reflex condition, (8). This is a typical optimization problem. Using the Lagrange multipliers, we get the solution as

$$a_{ni} = \frac{e^{-\beta_n R_{n-1}}}{\sum_{j=1}^{k+2} e^{-\beta_n R_{n-1}}}, \quad i = 1, 2, \cdots, k+2, \qquad (10)$$

where β_n is the solution of the following equation,

$$\sum_{j=1}^{k+2} (R_{n-j} - R_n) e^{-\beta(R_{n-1} - R_n)} = 0. \qquad (11)$$

In (10) and (11), $R_{n-(n+1)} = R_n^L$ and $R_{n-(n+2)} = R_n^H$.

The maximum entropy probability distribution represents the least biased judgment possible for $a_{ni}, \ i = 1, 2, \cdots, k+2$. This claim is justified in [21], [22] and thereafter. For example, the MEP contains the classic principle of insufficient reason (PIR) as a special case. Indeed, in the absence of any reasons, i.e. in the case there no or only trivial constraints are imposed on the probability distribution, $H(a_n)$ is maximal when all probabilities are equal, which corresponds to a uniform distribution.

Since RTT is usually updated at every ACK received, the computing overhead plays a crucial role in designing RTT prediction algorithms. Because (11) is an exponential equation, the computation would be too intensive to be practical for real-time on-line implementation if it were solved numerically. To get around this problem, we use an approximate analytical approach to solving $a_{ni}, \ i = 1, 2, \cdots, k+2$. From (7), we can write

$$a_{ni} = \frac{1}{k+2} + \varsigma_{ni},$$

where $\varsigma_{ni}, \ i + 1, 2, \cdots, k+2$, is a real number and $\sum \varsigma_{ni} = 0$. The entropy of the associated distribution at instant n then becomes:

$$H(a_n) = -\sum_{i=1}^{k+2} (g + \varsigma_{ni}) \ln(g + \varsigma_{ni}),$$

where $g = 1/(k+2)$. Below, to simplify the notations, we drop the index n. The logarithm term in the above equation can be written as

$$\begin{aligned}\ln(g + \varsigma_i) &= \ln((g)(1 + \frac{\varsigma_1}{g})) = \ln(g) + \ln(1 + \frac{\varsigma_1}{g}) \\ &\approx \ln(g) + \frac{\varsigma_1}{g} - \frac{\varsigma_1^2}{2g^2}.\end{aligned}$$

The entropy then becomes

$$\begin{aligned}H(a) &\approx -\sum_{i=1}^{k+2} (g + \varsigma) \left[\ln(g) + \frac{\varsigma_i}{g} - \frac{\varsigma_i^2}{2g^2} \right] \\ &\approx -\sum_{i=1}^{k+2} g \ln(g) - \sum_{i=1}^{k+2} \varsigma_i + \sum_{i=1}^{k+2} \frac{\varsigma_i^2}{2g} - \sum_{i=1}^{k+2} \varsigma_i \ln(g) \\ &\quad - \sum_{i=1}^{k+2} \frac{\varsigma_i^2}{g}.\end{aligned}$$

Since $\sum_{i=1}^{k+2} \varsigma_i = 0$ and g is a constant, we get

$$H(a) = -\sum_{i=1}^{k+2} g \ln(g) - \sum_{i=1}^{k+2} \frac{\varsigma_i^2}{2g}.$$

Substituting g in the above equation with $2/(1+k)$, the optimization problem becomes:

$$\max_a \{H(a)\} = \max_a \left\{ -\sum_{i=1}^{k+2} (\frac{1}{k+2} \ln(\frac{1}{K+2})) - \frac{k+2}{2} \sum_{i=1}^{k+2} \varsigma_i^2 \right\}.$$

Table 1. Data collection specifications.

Collection Date	Sept. 24, 2001 (Mon)	Sept. 25, 2001 (Tue)	Sept. 26, 2001 (Wed)
Collection Time	10:00am	14:00pm	18:00pm
Pingee Host Location	Northern America	Asian	Europe
Host Type	Academy	Business	Government

Since the first sum at the right-hand side of the above equation is a constant and $\varsigma_{ni} = a_i - 1/(k+2)$, this is equivalent to

$$\min_{a} \left\{ \sum_{i=1}^{k+2} (a - \frac{1}{k+2_i})^2 \right\}.$$

Taking the constraints, (7) and (8), into account, using the Lagrange function and reintroducing the index n for a_{ni} , we get the final solution as

$$a_{ni} = C - (R_{n-1} + AC)(R_n - AC)/(B - A^2C) , \quad (12)$$

where $A = \sum_i R_{n-i}$, $B = \sum_i R_{n-1}^2$ and $C = 1/(k+2)$.

Since (12) is only a simple polynomial equation and k is usually less then 5 as in our experiments, the computing cost of updating the coefficients should not be a big burden on real-time implementation especially for today's computers which are quite several orders faster than the ones 15 years ago when the ARMA method was first adopted although the network is orders faster than before as well. In fact, the proposed MEP algorithm was successfully implemented on-line in the trinomial protocol [6]. In addition, for real-time applications, the computing cost devoted to RTT estimation using the MEP method is quite trivial compared to the costs for other computations. In Internet conferencing, for instance, one live image is usually transferred every 30ms~50ms, during which the computing overhead of RTT estimation is nearly nothing compared to image-related computation that usually includes image capturing, digitalization, compression, encoding, and buffering.

IV. EXPERIMENTS

To verify the validity of the MEP algorithm, it is examined by using the data collected from real-world Internet connections. For numerical evaluation, the results are compared with those of the current ARMA method in both RTT prediction and RTO estimation.

A. Data Collection

The ICMP protocol based *Pinger* is chosen as the tool to collect data for two reasons. First, the implementation mechanism of the ICMP protocol is similar to rate-based realtime protocols. Second, ICMP based *Pinger* tools are widely available. The Pinger tool we used is *TJPingPro©1.2.1* , an RTT measurement tool developed by Top Jimmy Software [23]. In the experiments, nonfragmented ICMP packets are sent out at regular intervals from the *Pinger* host, *meng.ee.ualberta.ca*, at the University of Alberta, Canada, to the selected *Pingee* hosts around the world. Due to the huge diversity of the Internet, it is impossible to examine the MEP algorithm under all conditions for all connections. For a fair representative, one *Pingee* host is chosen randomly from each combination of locations and sectors in Table 1. As a result, we have 9 links for the experiments.

Table 2. Performance comparison summary.

	OH (%)		ERR (%)	
	ARMA	MEP	ARMA	MEP
Monday	19.06	16.38	3.75	1.81
Tuesday	25.51	16.31	1.98	1.31
Wednesday	20.26	19.71	3.73	1.69

The time scheduled for data collection in Table 1 is based on Mountain Time (US & Canada). Data is collected for 3 successive working days and we have a total of 81 samples of RTT time series. In the experiments, the size of probing packets is 69 bytes and the timeout is large enough (10 seconds) to make sure no RTT samples are missed out. In each experiment, 50 successive data packets are sent out from the *Pinger* to the *Pingee*, and the *Pinger* receives packets of the same size returned from the *Pingee* if the packet is not lost. If a packet is received by the *Pinger*, the corresponding RTT is sampled. Otherwise, the corresponding packet position is null and removed from the RTT time series.

B. RTT Variation Prediction $\hat{\sigma}_i$

To estimate RTO, we should also have the prediction of RTT variation. In the implementation of the current ARMA algorithm, the following equation is used for this purpose:

$$\hat{\sigma}_{n+1} = (1-g)\hat{\sigma}_n + g \times |R_n - \hat{R}_n| , \quad (13)$$

where g is a constant factor which is usually 0.25. RTO_{n+1} is then calculated by (1). For an unbiased comparison with the ARMA method, we adopt (13) for RTT variation prediction in the MEP algorithm as well.

C. Experiment Results and Performance Comparison

For the MEP algorithm, R_n^L and R_n^H are determined as follows respectively

$$R_n^L = (1-\gamma) \times \min\{R_n, R_{n-1}, \cdots, R_{n-k+1}\}, \quad (14)$$

$$R_n^H = (1+\gamma) \times \min\{R_n, R_{n-1}, \cdots, R_{n-k+1}\}, \quad (15)$$

where $\gamma = 0.2$ and $k = 4$. We use two criteria, percentage overhead *(OH)* and percentage error *(ERR)* for the comparison of RTO estimation. Their definitions are

$$OH = \frac{\sum_i (RTO_i - R_i)}{\sum_i R_i} , \quad (16)$$

Table 3. Experimental results I (Date: Monday, Sept. 24, 2001).

Pingee Location	Pingee Host	10:00am				14:00pm				18:00am			
		OH (%)		ERR (%)		OH (%)		ERR (%)		OH (%)		ERR (%)	
		ARMA	MEP	ARMA	MEP	ARMA	MEP	ARMA	MEP	ARMA	MEP	ARMA	MEP
Nothern America	164.109.71.245	19.94	22.84	0.00	0.00	20.80	31.34	3.57	3.57	18.08	25.90	5.00	2.50
	18.181.0.31	21.09	16.11	0.00	2.00	16.80	15.14	0.00	0.00	13.05	13.48	2.00	2.00
	128.231.56.110	26.20	35.13	2.00	0.00	36.51	37/00	6.00	0.00	14.38	18.35	10.00	0.00
Asia	202.106.68.19	66.14	23.10	4.17	4.17	13.72	14.00	2.38	0.00	41.47	40.19	2.63	0.00
	202.112.144.70	17.64	12.28	2.56	0.00	6.06	8.62	5.40	2.70	6.46	9.70	8.57	2.85
	202.108.249.206	8.60	9.85	0.00	0.00	16.15	14.08	2.77	2.77	35.10	10.75	0.00	2.70
Europe	212.160.117.137	23.65	32.65	5.40	2.7	9.52	14.95	2.86	2.86	11.96	7.33	0.00	0.00
	148.88.2.37	3.79	5.63	12.00	4.00	3.70	5.52	12.00	4.00	4.77	7.68	10.00	6.00
	130.192.239.1	41.96	28.03	2.04	0.00	8.33	9.70	0.00	2.00	8.74	9.99	0.00	2.00
Average		25.45	20.62	3.13	1.43	14.62	12.59	3.89	1.99	17.11	15.93	4.24	2.01

Table 4. Experimental results II (Date: Tuesday, Sept. 25, 2001).

Pingee Location	Pingee Host	10:00am				14:00pm				18:00am			
		OH (%)		ERR (%)		OH (%)		ERR (%)		OH (%)		ERR (%)	
		ARMA	MEP	ARMA	MEP	ARMA	MEP	ARMA	MEP	ARMA	MEP	ARMA	MEP
Nothern America	164.109.71.245	51.89	20.79	0.00	2.08	29.88	34.82	8.70	4.34	22.80	26.16	4.26	2.13
	18.181.0.31	21.20	27.07	0.00	2.00	20.56	20.49	0.00	0.00	19.24	18.30	0.00	0.00
	128.231.56.110	35.01	28.14	4.17	0.00	26.56	28.17	6.00	2.00	18.73	20.73	0.00	2.00
Asia	202.106.68.19	109.19	19.53	2.04	2.04	70.30	36.20	2.04	0.00	6.97	4.58	2.08	2.08
	202.112.144.70	12.20	12.27	3.70	0.00	6.18	5.93	0.00	2.38	7.61	8.67	0.00	0.00
	202.108.249.206	4.92	4.27	0.00	0.00	9.05	8.41	2.00	2.00	8.85	6.44	0.00	2.17
Europe	212.160.117.137	50.73	13.61	2.56	0.00	84.36	19.04	0.00	0.00	14.69	14.15	2.08	2.08
	148.88.2.37	7.97	12.01	0.00	0.00	5.37	5.58	8.00	4.00	8.72	7.84	2.00	0.00
	130.192.239.1	12.28	14.34	2.00	4.00	10.51	12.49	2.04	2.04	13.08	10.53	0.00	0.00
Average		33.93	16.89	1.60	1.12	29.20	19.01	3.20	1.86	13.41	13.04	1.16	0.94

Table 5. Experimental results III (Date: Wednesday, Sept. 26, 2001).

Pingee Location	Pingee Host	10:00am				14:00pm				18:00am			
		OH (%)		ERR (%)		OH (%)		ERR (%)		OH (%)		ERR (%)	
		ARMA	MEP	ARMA	MEP	ARMA	MEP	ARMA	MEP	ARMA	MEP	ARMA	MEP
Nothern America	164.109.71.245	12.39	16.74	6.38	4.26	19.79	29.16	2.78	0.00	25.57	20.11	2.27	0.00
	18.181.0.31	14.80	20.41	2.00	0.00	15.60	21.32	0.00	0.00	15.77	14.84	0.00	2.00
	128.231.56.110	39.51	49.67	2.00	2.00	37.42	40.60	8.00	2.00	24.41	24.99	6.00	0.00
Asia	202.106.68.19	7.78	5.53	2.00	0.00	31.60	16.15	7.14	2.38	20.28	24.15	0.00	2.08
	202.112.144.70	10.30	11.73	3.57	0.00	12.62	12.64	0.00	2.70	48.46	34.30	3.48	3.48
	202.108.249.206	12.24	8.7	0.00	2.00	4.29	3.37	4.65	2.32	9.36	8.38	2.22	0.00
Europe	212.160.117.137	106.37	72.72	6.67	3.33	6.72	11.32	5.71	0.00	30.39	29.53	10.53	5.26
	148.88.2.37	4.69	7.94	8.00	4.00	4.61	4.50	0.00	0.00	8.90	10.44	8.00	6.00
	130.192.239.1	7.71	11.43	4.00	2.00	7.43	7.74	2.00	0.00	8.07	13.78	6.00	0.00
Average		23.98	22.76	3.85	1.94	15.56	16.31	3.06	1.04	21.25	20.06	4.28	2.09

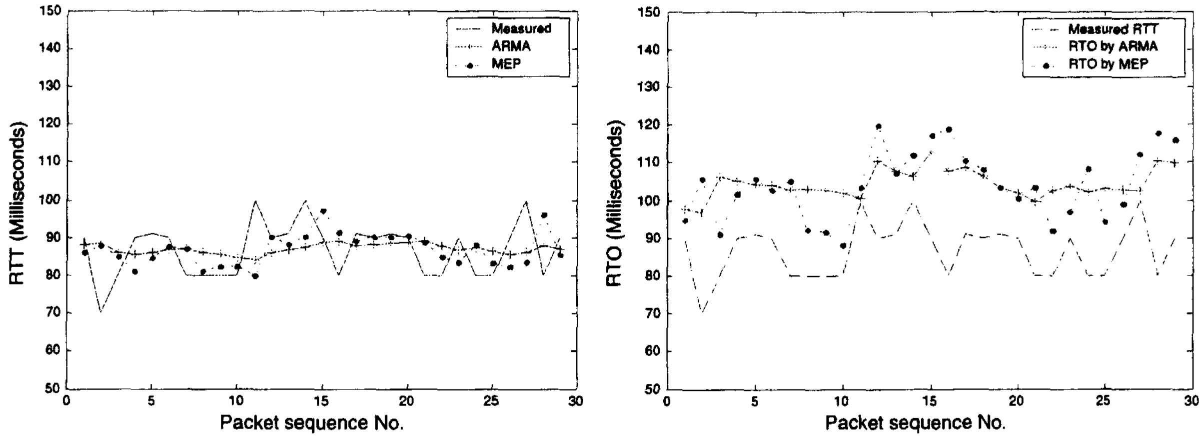
$$ERR = \frac{m}{N}, \tag{17}$$

where m is the number of failed prediction for which RTO_i (predicted roundtrip timeout) is smaller than the real $RTT(R_i)$ and N is the total number of returned data packets the *Pinger* receives. From the above definitions, *OH* can be viewed as the indicator of the overhead for waiting packets in detecting packet loss. On the other hand, *ERR* is the ratio of the number of packets which would be rejected improperly to the total number of returning packets received by the *Pinger* should the estimated RTO be used. If the RTO estimate is too large, as a result, we have a large *OH*. On the contrary, if RTO estimate is too small, *ERR* will be high. A summary of the experimental results is given by Table 2. The detailed results are shown in Tables 3, 4, and 5. From these results, we can see that the MEP algorithm has smaller *OH* and *ERR* than the ARMA method statistically.

In this part, we show the comparisons between the MEP and ARMA algorithms graphically. Figs. 1(a), 2(a), 3(a), and 4(a) are the typical curves of the measured RTT, the predicted one using the ARMA method and the predicted one using the MEP algorithm. Fig. 1(a) shows a typical scenario where there is no congestion on the network. In Fig. 2(a), the RTT increases abruptly, grows gradually for a while and then suddenly jumps down, which may be the result of network burst. It can be inferred that the network is congested from the RTT curves shown in Fig. 3(a) and Fig. 4(a).

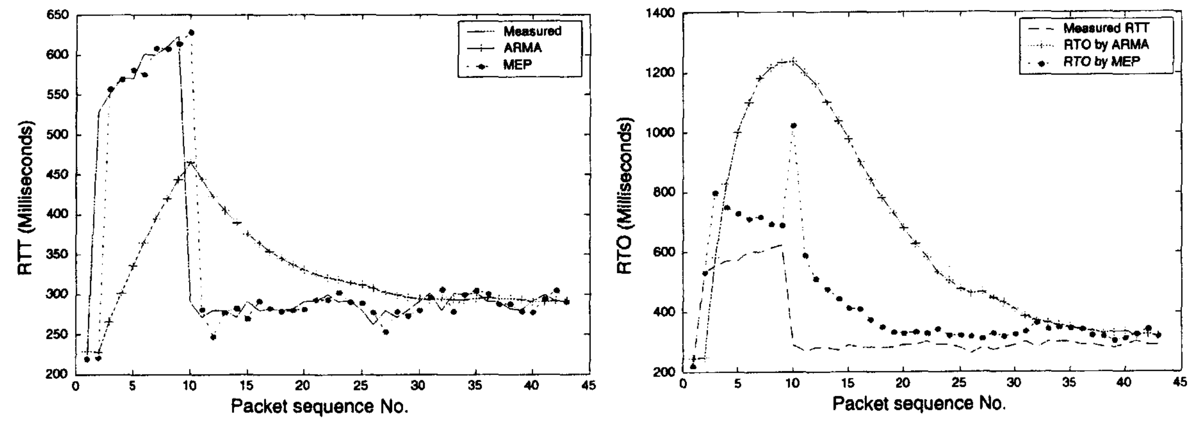
For the scenario shown in Fig. 1(a), the RTT dynamics is dominated by a base plus a noise process with weak patterns. This is the situation on which the traditional ARMA method is targeted. From the plot, the ARMA and the MEP algorithms have similar performance. We cannot determine which scheme is superior for this type of situations. However, for the scenarios shown in Figs. 2(a), 3(a), and 4(a), where the RTT time series show apparent structures and RTT evolves rapidly and significantly, it is clear that the ARMA algorithm cannot track the RTT dynamics since the dynamics of ARMA itself is too slow. On the contrary, the MEP method works very well and much better than ARMA.

Figs. 1(b), 2(b), 3(b), and 4(b) are the corresponding curves of the measured RTT, the estimated RTO using ARMA and the estimated RTO using MEP. From Fig. 1(b), the performance of both ARMA and MEP is at the same level. However, for



(a) (b)

Fig. 1. Scenario I: (a) RTT, (b) RTO.



(a) (b)

Fig. 2. Scenario II: (a) RTT, (b) RTO.

Figs. 2(b), 3(b), and 4(b), the OHs of MEP are much smaller than those of ARMA while the ERRs are approximately at the same level. From Fig. 2(b), both the OH and ERR of MEP is smaller than those of ARMA.

V. CONCLUSIONS

The great advantage of the conventional ARMA algorithm for RTT estimation is its low computing cost, however, this parameter-fixed model is not sufficient for all scenarios and is especially insufficient for rate-based TCP-friendly transport protocols for real-time applications. In this paper, we introduce a parameter-varying model for RTT prediction based on the information theory and the maximum entropy principle. Because the coefficients of the proposed RTT model are updated dynamically, the model is adaptive and can track RTT dynamics quickly. From the experiments, the MEP algorithm has similar performance to ARMA when RTT dynamics is dominated by a base plus a noise process, on which the traditional ARMA method is targeted. However, when RTT evolves rapidly and significantly, the MEP algorithm works much better than ARMA in terms of both RTT prediction and RTO estimation. Since the solution to the coefficient of the MEP model is only a simple polynomial, the computing cost of updating the coefficients will not be a big burden on real-time implementation.

ACKNOWLEDGEMENTS

The authors would like to thank our anonymous JCN reviewers for their valuable comments on the earlier version of this paper.

REFERENCES

[1] J. Postel, *RFC 793: Transmission control protocol*, available at http://www.ibiblio.org /pub/docs/rfc/rfc793.txt, Sept. 1981.

[2] V. Jacobson, "Congestion avoidance and control," in *Proc. ACM SIGCOMM*, Stanford, CA, Aug. 1988, pp. 314–329.

[3] L. Zhang, "Why TCP timers don't work well," in *Proc. ACM SIGCOMM Symp. Comm., Architectures, and Protocols*, Stowe, Vermont, Aug. 1986, pp. 397–405.

[4] J. Padhye *et al.*, *A TCP-friendly rate adjustment protocol for continuous media flows over best effort network*, UMass-CMPSCI Technical Report TR 98-04, Oct. 1998.

[5] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet," in *Proc. IEEE INFOCOM*, New York, NY, Mar. 1999, pp. 1337–1345.

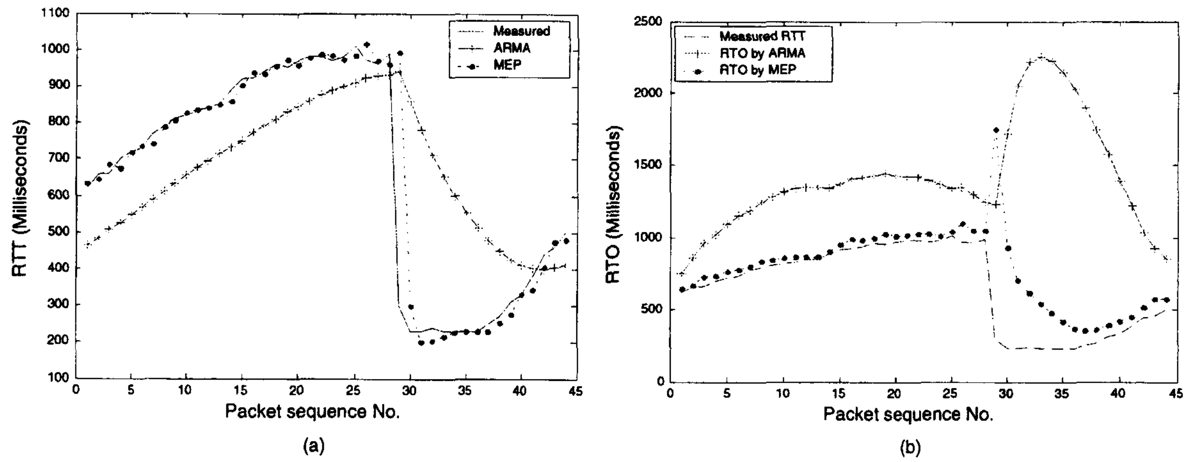


Fig. 3. Scenario III: (a) RTT, (b) RTO.

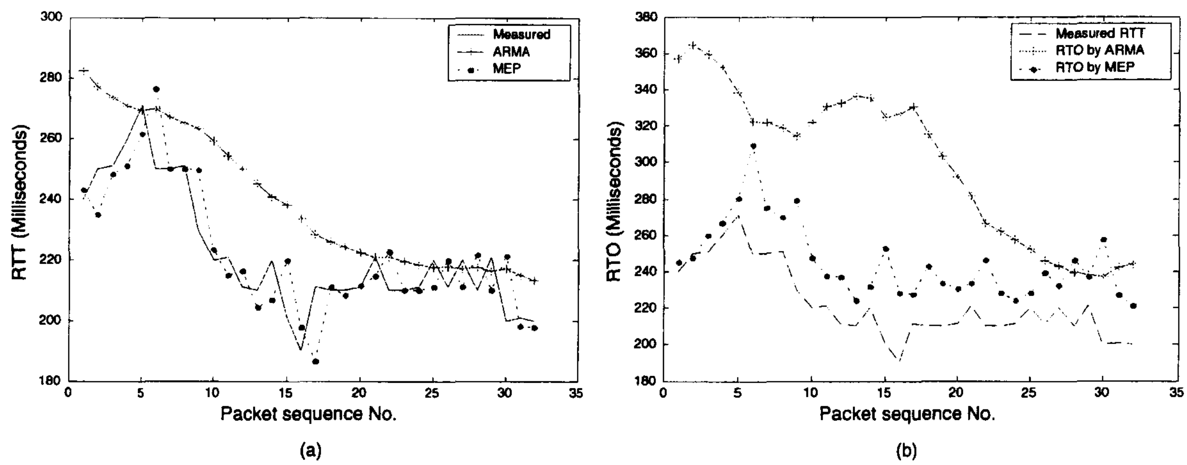
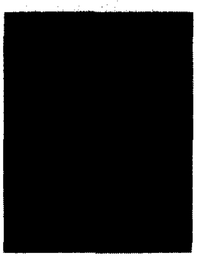


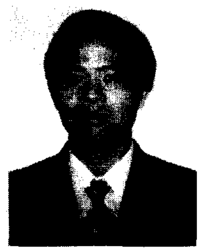
Fig. 4. Scenario IV: (a) RTT, (b) RTO.

[6] P. Liu *et al.*, "A novel data transport protocol for Internet robots," in *Proc. World Congress Intelligent Control and Automation (WCICA'02)*, June 2002, pp. 59–65.
[7] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Trans. Networking*, 7(4): pp. 458–472, 1999.
[8] K. Goldberg and R. Siegwart (editors), *Beyond Webcams: An introduction to online robots*, The MIT Press, Cambridge, Massachusetts, USA, 2002.
[9] M. S. Borella *et al.*, "Self-similarity of Internet packet delay," in *Proc. IEEE ICC'97*, Aug. 1997, pp. 513–517.
[10] Q. Li and D. L. Mills, "Jitter-based delay-boundary prediction of wide-area networks," *IEEE/ACM Trans. Networking*, 9(5): pp. 578–590, 2001.
[11] D. Mills, *RFC889: Internet Delay Experiments*, Dec. 1983.
[12] R. Jain, "Divergence of timeout algorithms for packet retransmissions," in *Proc. Fifth Phoenix Conf. Comp. and Comm.*, Scottsdale, Arizona, Mar. 1986, pp. 174–179.
[13] P. Karn and C. Partridge, "Improving round-trip time estimates in reliable transport protocols," *ACM Trans. Computer Systems*, 9(4): pp. 365–373, 1991.
[14] J.-C. Bolot, "Characterizing end-to-end packet delay and loss in the Internet," *J. High Speed Networks*, 2(3): pp. 305–323, 1993.
[15] A. Acharya and J. Saltz, "A study of Internet round-trip delay," *Technical Report CS-TR-3736*, University of Maryland, Dec. 1996.
[16] M. Allman and V. Paxson, "On estimating end-to-end network path properties," in *Proc. ACM SIGCOM Comput. Commun. Review*, 1999, pp. 263–274.
[17] D. D. Clark, *RFC 813: Window and acknowledgement strategy in TCP*, July 1988.
[18] P. Klan and G. Darbellay, "An information-theoretic adaptive method for time series forecasting," *Neural Network World*, 7(2): pp. 227–238, 1997.
[19] D. Kugiumtzis and M. A. Boudourides, "Chaotic analysis of Internet ping data," *SOEIS Meeting at Bielefeld*, Mar. 27–28, 1998.
[20] S. Guiasu and A. Shenitzer, "Maximum entropy principle," *Mathematical Intelligencer*, 7(1): pp. 42–48, 1985.
[21] E. T. Jaynes, "Information theory and statistical mechanics I," *Physical Review*, 106(4): pp. 620–630, 1957.
[22] J. Uffink, "Can the maximum entropy principle be explained as a consistency requirement?," *Studies in History and Philosophy of Modern Physics*, 26(3): pp. 223-261, 1995.
[23] Available at http://www.topjimmy.net.

Peter Xiaoping Liu is currently an Assistant Professor of Systems and Computer Engineering at Carleton University, Canada. He received his B.S. degree and M.S. degree from Northern Jiaotong University, Beijing, China in 1992 and 1995 respectively, and his Ph.D. degree from the University of Alberta, Canada in 2002. His current interests include network-enabled systems, real-time network computing and distributed systems, network protocols and congestion control, Internet imaging, e-service robotics, control, stochastic processes and time series.



Max Q-H Meng received his Ph.D. degree in Electrical and Computer Engineering in 1992 from the University of Victoria in Canada. He is currently a Professor of Electronic Engineering at the Chinese University of Hong Kong, on leave from the Department of Electrical and Computer Engineering at the University of Alberta in Canada, where he is a Professor and the Director of the ART (Advanced Robotics & Teleoperation) Lab. His research expertise is in the areas of Robotics, Medical Robotics, Biomedical Engineering, Network Enabled Services, Intelligent and Adaptive Systems, Human-Machine Interface, and their medical, industrial, and military applications. He has published extensively in his areas of expertise. He is an editor of the IEEE/ASME Transactions on Mechatronics, an Associate Editor of the Journal of Control and Intelligent Systems, a Technical Editor of Advanced Robotics, the General Chair of the 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA 2001), and the General Chair of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005). Among his awards, he is a recipient of the IEEE Third Millennium Medal award.



Jason Gu is currently an assistant professor in the Department of Electrical and Computer Engineering at Dalhousie University. He received his Bachelor in the Electrical Engineering and Information Science from the University of Science and Technology of China, and the Ph.D. degree in the Electrical and Computer Engineering at University of Alberta. His research interests include robotics, control systems and intelligent systems.