

# Predicting End-to-End Delay of the Internet Using Time Series Analysis \*

**Ming Yang X. Rong Li**

Department of Electrical Engineering

University of New Orleans, New Orleans, LA 70148

Phone: 504-280-7416, Email: {myang2, xli}@uno.edu

Nov. 11, 2003

## **Abstract**

This report surveys time series methods that have been used and can be applied in predicting end-to-end delay of the Internet. ARIMA scheme and state-space approach are discussed and compared. Although state-space approach has the advantages in structure and computation, ARIMA modeling is still useful in identifying systems due to the complexity and uncertainty of the Internet. A practical example of using ARMA models to do prediction is given in the report. By converting the time series models (ARIMA or ARMAX) to state-space representation, linear prediction and control theory can be applied more directly. Furthermore, by regarding the network as a hybrid system, multiple-model approach, whose algorithms are mainly based on state-space representation and Kalman filtering, can be used to study the dynamics of the Internet.

## **1 Introduction**

Time series methods have been widely used in many areas: statistics, engineering, economy, medicine, etc. A time series is a collection of observations made sequentially in time. End-to-end Internet delay, as well as round-trip time, when collected at equally spaced intervals over time, are typical time series data. If there is no background information available, time series analysis is a suitable approach since the model fitting procedure does not require any assumption about the internal structure of the observed system.

Queueing theory, as a powerful tool to analyze computer and communication networks for a long time, is a natural choice for delay prediction. However, accurate queueing analysis requires that the distributions of traffic inter-arrival and inter-departure time at each individual link are known, which is rarely the case in the Internet. Even though the distribution of each link is available, the

---

\*This research is sponsored in part by High-Performance Networking Program of the Office of Science, U. S. Department of Energy under Contract DE-AC05-00OR22725 with UT-Battelle, LLC.

computational cost will grow dramatically as the network size increases. Due to these limitations and difficulties in capturing the dynamic behavior of the networks by queueing theory, time series analysis becomes an important alternative. Comparing with analytic models such as queueing models, time series models are cheaper to develop, easier to utilize and update, and in general, less complex to use.

The rest of this report is organized as follows. In Section 2, we discuss the relevant concepts and methodology to the Internet end-to-end delay prediction using time series analysis. In Section 3, we confine our topic on model identification in time series analysis. Section 4 gives a practical example. Section 5 concludes the report and provides our work direction.

## 2 Analysis of the Internet End-to-End Delay Prediction

### 2.1 Prediction Theory and Time Series Analysis

Several important concepts need to be addressed before a precise mathematical statement of the prediction problem and its solution can be given. For example, without criteria it is hard to judge the accuracy of a prediction algorithm.

Here we use a subspace  $\mathbb{M}$  of a Hilbert space  $\mathbb{H}$  to denote the information about the past of a system, any element of  $\mathbb{M}$  is called a *predictor* [13]. Therefore  $\mathbb{M}$  can be viewed as the space of allowable predictors for the future among which we are to find the “best” one.

If  $\hat{X}$  is the predictor for a future random variable  $X$  ( $X \in \mathbb{H}$ ), the prediction error is  $X - \hat{X}$ , and it is desirable to make this error as small as possible using certain error metric. But since  $X - \hat{X}$  is a random quantity and not observable in general, it is natural to pick  $\hat{X}$  so that  $X - \hat{X}$  is small on the average. This can be done, for example, by choosing  $\hat{X}$  so that either  $\Pr\{|X - \hat{X}| \geq \epsilon\}$  or  $E[|X - \hat{X}|^p]$  is small, for appropriate values of  $\epsilon$  or  $p$ .

A popular criterion for the goodness of a predictor is the *mean square error* (MSE):

$$E[|X - \hat{X}|^2] = \|X - \hat{X}\|^2 \quad (1)$$

This criterion is quite satisfactory as far as mathematical tractability of problems related to finding explicit formula for  $\hat{X}$  is concerned. On the other hand, it reflects the requirement that large errors are more serious than small ones. The best predictor of  $X$  based on  $\mathbb{M}$  is an element  $\hat{X} \in \mathbb{M}$  which is closest to  $X$ , that is,

$$\|X - \hat{X}\|^2 \leq \|X - Y\|^2 \quad \text{for all } Y \in M \quad (2)$$

The distance from  $\hat{X}$  to  $X$ , i.e.,

$$\|X - \hat{X}\|^2 = \inf_{Y \in \mathbb{M}} \|X - Y\|^2, \quad (3)$$

is called the *minimum mean square error* (MMSE) of prediction of  $X$  based on  $\mathbb{M}$ .

In practice, it is desirable to develop a prediction theory that leads to simple prediction formulas and requires less statistical information than the full distribution of the past. The linear prediction or the Kolmogorov-Wiener prediction theory (refer to Chapter 12 of [10]) provides such a setting in

which only the knowledge of the past and the first two moments of the distribution of the past are required. In fact for the normal (Gaussian) distribution, the first two moments can fully determine the distribution.

The linear prediction theory is concerned with approximating future in terms of the observed past when the covariance function and the past of a stationary random process (RP) are known. The final goal is to express the predictors in terms of the known past values and the first two moments of the RP. Note in time series analysis usually the *ergodic* assumption of the RP is also required. In [2] Box and Jenkins showed how to find an optimal linear predictor for a particular Auto-Regressive Integrated Moving Average (ARIMA) process. Furthermore the case when autocorrelation function (ACF) is unknown was also considered.

It is clear that such a prediction problem is closely related to a (quality) control problem because if we can predict how a process will behave, we can adjust the process so that the achieved values are, in some sense, as close to the target value as possible. In many applications, the purpose of predicting the Internet end-to-end packet delay is to design a working mechanism so that the Internet can work more stably and more efficiently. In particular, by more accurate prediction, delay-based bandwidth allocation and congestion control can provide further improvements to QoS in heterogeneous networks.

## 2.2 Linear Systems and Control Theory

For either prediction and control purpose, the core work is to identify a model for the system (or process) given observations on the input and output of the system. Although recently there has been an increased interest in time-varying and non-linear systems, much of the literature assumes that the system can be adequately approximated over the range of interest by a linear model whose parameters do not change with time. Numerous useful results in control theory has been obtained for such linear time-invariant (LTI) systems (e.g., [4], [16]).

Control theory was originally concerned mainly with deterministic systems (most of control engineers are still in this area). In the Internet flow control and congestion problems, more attention to stochastic control has been paid. Stochastic control, where the system being controlled is subject to random disturbances, was started with the work on filtering problems by Wiener and Kolmogorov. One major development was the *Kalman filter*, which is a recursive method of estimating the state of a system in the presence of noise (further description for the Kalman filter is in Section 3.3). Kalman filtering has been used in many applications including the control of a space rocket, where the system dynamics are well defined but the disturbances are unknown.

In most control problems, the knowledge of the system structure has been assumed known. But in the Internet dynamics, many issues are involved [14] and their impact is subtle (or hidden). In such a case, statistical work in time series will be very helpful to identify the system. In particular, the Box-Jenkins approach tries to identify a system using observed data. Because for each ARIMA model there exists a corresponding state-space representation (albeit not uniquely), it is possible to make some collaboration between control theory and time series analysis. More specifically, once

the ARIMA model in its state-space representation for the process (system) is obtained, results from control theory can be applied.

### 3 Model Identification in Time Series Analysis

Most time series data of the Internet delay are non-stationary. The methods described in [15] (e.g., ARMA, AR and MA models) are for stationary time series. As we mentioned there: series can be made stationary by operations such as differencing. The ARIMA methodology, develop by Box and Jenkins, is based on such an idea [2]. Once has this been done, the model-fitting techniques described in [15] are still valid. All the remains is to reverse the differencing operations so as to make prediction in levels.

#### 3.1 Autoregressive Integrated Moving Average Models

In practice, a non-stationary time series may be differenced more than once to make it stationary. When prediction is carried out in model fitting the differencing operation must be reversed, and this operation is called *integration* by Box and Jenkins [2]. If  $d$  is the order of differencing needed to produce a stationary and invertible process, the original process is said to be *integrated* of order  $d$  and abbreviated as  $y_t \sim I(d)$ . Similarly the model

$$\phi_p(B)(1 - B)^d y_t = \theta_q(B)\varepsilon_t \quad (4)$$

is called an *autoregressive integrated moving average* process of order  $(p, d, q)$ , and denoted as ARIMA( $p, d, q$ ), where  $\varepsilon_t$  is the disturbance,  $B$  is a backward shift operator, i.e.,

$$By_t = y_{t-1}, B^2 y_t = B \cdot By_t = By_{t-1} = y_{t-2}, \dots \quad (5)$$

and

$$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p \quad (6)$$

$$\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q \quad (7)$$

If  $d > 0$ , the model is clearly non-stationary, as the AR operator  $\phi(B)(1 - B)^d$  has  $d$  roots on the unit circle. An ARMA model can be viewed as a special case of ARIMA models: a stationary model should have  $d = 0$ , then an ARMA( $p, q$ ) is equivalent to an ARIMA( $p, 0, q$ ).

The issues involved in fitting ARMA models were studied in [15]. There leaves one question for the ARIMA methodology: how does order  $d$  affect predictions?

A basic feature of the prediction from stationary models is that they tend towards the mean of the series as the lead time  $l$  (prediction interval) increases. If  $l$  is large, the structure of the model is irrelevant.

Consider an ARMA( $p, q$ ) model

$$\phi_p(B)x_t = \theta_q(B)\varepsilon_t \quad (8)$$

where  $x_t$  is a stationary time series. To find the prediction MSE, note that it has the infinite MA representation [6]

$$x_t = \sum_{j=0}^{\infty} \psi_j \varepsilon_{t-j} \quad (9)$$

where  $\{\psi_j\}$  are parameters and  $\psi_0 = 1$ . Therefore the *optimal predictor*  $l$  step ahead is the expected value of  $x_{T+l}$  conditional on the information at time  $t = T$ , which can be written as

$$\hat{x}_{T+l|T} = E[x_{T+l}|x^T] = \sum_{j=0}^{\infty} \psi_{l+j} \varepsilon_{T-j} \quad (10)$$

where  $x^T$  denotes the information set  $\{x_T, x_{T-1}, \dots\}$ . Then the prediction MSE is

$$\text{MSE}(\hat{x}_{T+l|T}) = \sum_{j=1}^l \psi_{l-j}^2 \sigma^2 \quad (11)$$

where  $\sigma^2$  is the variance of the disturbance  $\varepsilon_t$ .

The MSE of a prediction from an ARIMA model can be obtained similarly as in (11)

$$\text{MSE}(\hat{y}_{T+l|T}) = \sum_{j=1}^l \psi_{l-j}^2 \sigma^2 \quad (12)$$

where  $y_t$  denotes a time series which may be non-stationary. But the  $\psi_j$  coefficients should be calculated by the following equation

$$\varphi(B)\psi(B) = \theta_q(B) \quad (13)$$

where

$$\begin{aligned} \varphi(B) &= \phi_p(B)(1-B)^d = 1 - \varphi_1 B - \varphi_2 B^2 - \dots - \varphi_{p+d} B^{p+d} \\ \psi(B) &= 1 + \psi_1 B + \psi_2 B^2 + \dots \end{aligned}$$

This yields

$$\begin{aligned} \psi_j &= \sum_{i=1}^{\min(j, p+d)} \varphi_i \psi_{j-i} + \theta_j, & 1 \leq j \leq q \\ \psi_j &= \sum_{i=1}^{\min(j, p+d)} \varphi_i \psi_{j-i}, & j > q \end{aligned} \quad (14)$$

The prediction MSEs in the ARIMA models tend to increase rapidly as the lead time  $l$  becomes greater. Thus the main value of such models is in short-term prediction. In fact, because of routing behavior, competing traffic, and available bandwidth etc., end-to-end delays are quite dynamic and the prediction interval cannot be too long.

### 3.2 The Box-Jenkins Seasonal (SARIMA) Model

In practice, many time series contain a seasonal periodic component which repeat every  $s$  observations. As we mentioned in Section 2, some seasonal phenomena do exist in the Internet end-to-end packet delays. How to deal with this *seasonality*? The ARIMA class of models is based on the idea that non-stationary trend movements can be captured implicitly by fitting an ARMA model to differenced

observations. By extending this idea, Box and Jenkins [2] use seasonal differences to handle this seasonality. They define a general multiplicative seasonal ARIMA model (SARIMA) as

$$\phi_p(B)\Phi_P(B^s)(1-B)^d(1-B^s)^D y_t = \theta_q(B)\Theta_Q(B)\varepsilon_t \quad (15)$$

where  $\Phi_P, \Theta_Q$  are polynomials of order  $P, Q$  respectively. In (15) there are not only simply differencing  $(1-B)^d$  (to remove *trend*) but also seasonal differencing  $(1-B^s)^D$  (to remove seasonality). The model in (15) is said to be a SARIMA model of order  $(p, d, q) \times (P, D, Q)_s$ . Usually the values of  $d$  and  $D$  do not need to exceed one.

For the issue of implementation, the model parameters may be estimated by some suitable iterative procedure. Full details are given in [2]. By now many computer programs (in e.g., SAS, MATLAB) can provide good enough estimations using such routines.

### 3.3 State-Space Approach and the Kalman Filter

Another general class of models, state-space models, could also be implemented in the prediction of the Internet end-to-end delay and the identification of their dynamics. Originally, such models were developed by control engineers, particularly for applications concerning navigation systems such as controlling the position of a space rocket. It turned out that they have become the most powerful representation in the control theory. They have also been found to be useful in short-term prediction problem. For our ultimate purpose, state-space models are more suitable, since with such models many existing results in the control theory can be straightforward applied.

The Kalman filter is an important general method of handling state-space models. Essentially, Kalman filtering gives optimal estimates of the current state of a linear dynamic system. It consists of a set of equations for recursively solving the Wiener-Kolmogorov filtering problem based on state-space models.

#### 3.3.1 State-Space Models

The state-space approach is appreciated better through the meaning of the word “state”. The state of a physical system is a (minimal) set of variables needed for predicting its future which summarizes the system’s past in full [10]. For example, to track a satellite in the space, it is crucial to know its position, direction and velocity, so that the state at time  $t$  is at least a 6-dimensional vector. The use of vector-valued processes is quite natural in state-space modeling. Let  $\{x_t\}$  be an  $n_x$ -dimensional process standing for the state of the system,  $\{z_t\}$  be the output of the system with  $n_z$  dimension, and  $\{u_t\}$  be the output of the system with  $n_u$  dimension then the state-space approach can be described by the two following equations (for a stochastic linear system), the first shows the evolution of the state is Markovian in nature and the second is of the form signal (state) plus noise:

$$x_t = F_t x_{t-1} + G_t u_t + \Gamma_t w_t, \quad \text{state equation} \quad (16)$$

$$z_t = H_t x_t + E_t u_t + v_t, \quad \text{observation equation} \quad (17)$$

where

$$F_t = \text{transition matrix } (n_x \times n_x)$$

$$G_t = \text{input gain matrix } (n_x \times n_u)$$

$$H_t = \text{output matrix } (n_z \times n_x)$$

$$E_t = \text{input-output matrix } (n_z \times n_u)$$

and  $\{v_t\}$  and  $\{w_t\}$  are vector-valued white noises (whose dimensions are  $n_v$  and  $n_w$  respectively and means are zero) orthogonal to each other with known covariance  $\Sigma_t = E[w_t w_t']$  and  $R_t = E[v_t v_t']$ ,  $\Gamma_t$  is the noise gain matrix with  $(n_x \times n_w)$ .

### 3.3.2 State-Space Representation of Time Series Models

State-space models for a time series problem are usually arrived at through a structural analysis of its components that make up the series. These components may include trend, seasonal, cycle, together with the explanatory variables, interventions, outliers and missing values. In contrast, the ARIMA modeling is a passive *black box* approach in which model identification relies solely on the data without prior information of the system that generated the data. Which tool is more suitable for our purpose? From a control engineer's point of view, state-space models have more structural advantages than the ARIMA framework. But in the study of the Internet end-to-end delay, unfortunately, there is very little information to build up the state of the system due to the complexity of the networks. A trade-off between these two schemes is to find the best fitting time series model by the ARIMA methods first, then convert to state-space representation so that prediction and control can be done more efficiently.

Consider in general an ARMA system (an ARMA process can be viewed as the *response* of an ARMA system to white noise, refer to [15])

$$y_t + a_1 y_{t-1} + \cdots + a_p y_{t-p} = b_0 u_t + b_1 u_{t-1} + \cdots + b_q u_{t-q} \quad (18)$$

$$\langle u_t \rangle = \text{stationary white noise}$$

Note that (18) can be written as

$$y_t + a_1 y_{t-1} + \cdots + a_m y_{t-m} = b_0 u_t + b_1 u_{t-1} + \cdots + b_m u_{t-m} \quad (19)$$

by setting  $a_{p+1} = a_{p+2} = \dots = a_m = 0$  and  $b_{q+1} = b_{q+2} = \dots = b_m = 0$  if necessary, where  $m = \max\{p, q\}$ . Therefore the state-space representation of this ARMA system in *observable canonical*

form is

$$x_t = \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{m-1} & 0 & 0 & \cdots & 1 \\ -a_m & 0 & 0 & \cdots & 0 \end{bmatrix} x_{t-1} + \begin{bmatrix} b_1 - a_1 b_0 \\ b_2 - a_2 b_0 \\ \vdots \\ b_{m-1} - a_{m-1} b_0 \\ b_m - a_m b_0 \end{bmatrix} u_{t-1} \quad (20)$$

$$y_t = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix} x_t + b_0 u_t \quad (21)$$

and in *controllable canonical form* is

$$x_t = \begin{bmatrix} -a_1 & -a_2 & -a_3 & \cdots & -a_{m-1} & -a_m \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} x_{t-1} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} u_{t-1} \quad (22)$$

$$y_t = \begin{bmatrix} b_1 - a_1 b_0 & b_2 - a_2 b_0 & \cdots & b_m - a_m b_0 \end{bmatrix} x_t + b_0 u_t \quad (23)$$

Note that these representations do not necessarily have a minimum dimension for the state  $x$ .

The state-space representations of the AR and MA systems can be obtained directly by setting  $b_1 = b_2 = \dots = b_m = 0$  and  $a_1 = a_2 = \dots = a_m = 0$ , respectively. However, models of a smaller dimension are preferred. Further details can be found in [10].

### 3.3.3 The Kalman Filter

In state-space modeling, the prime objective is to estimate the signal in the presence of noise. In other words we want to estimate the state vector  $x_t$ . The Kalman filter provides a general and efficient way of doing this. It consists of a set of equations which allow us to update the estimate of  $x_t$  when a new observation becomes available. This procedure has two stages, which are called the *prediction* stage and the *updating* stage respectively.

Consider the equations (16) and (17), the MMSE estimator of  $x$  can be computed sequentially in time using the following recursion:

- **Prediction:**

$$\hat{x}_{t|t-1} = F_{t-1} \hat{x}_{t-1|t-1} + G_{t-1} u_{t-1} \quad (24)$$

$$\hat{z}_{t|t-1} = H_t \hat{x}_{t|t-1} + E_t u_t$$

$$P_{t|t-1} = F_{t-1} P_{t-1|t-1} F_{t-1}^T + \Gamma_{t-1} \Sigma_{t-1} \Gamma_{t-1}^T$$

$$S_t = H_t P_{t|t-1} H_t^T + R_t$$

$$K_t = P_{t|t-1} H_t^T S_t^{-1} \text{ (Kalman Gain Matrix)}$$

- **Updating:**

$$\tilde{z}_{t|t-1} = z_t - \hat{z}_{t|t-1} \text{ (Correction)} \quad (25)$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t \tilde{z}_{t|t-1}$$

$$P_{t|t} = P_{t|t-1} - K_t S_t K_t^T \text{ (Minimum MSE Matrix)}$$

Further topics in Kalman filtering (e.g., initialization) can be found in [8], [10]. A major practical advantage of the Kalman filter is that the calculations are recursive, so that although the current estimates are based on the whole past history of measurements, there is no need for an ever-expanding memory. Another advantage of the Kalman filter is that it converges fairly quickly when there is a constant underlying model, but can also follow the movement of a system where the underlying model is evolving through time [3].

## 4 Practical Example

In [7], there presented a pioneering work of congestion control. Jacobson designed his congestion avoidance algorithm by modeling Round Trip Times (RTTs) based on ARMA models. Following the similar idea, we constructed an experiment to get the Internet end-to-end delay data (RTTs) and did some preliminary data analysis.

### 4.1 Data Collection

In this study we sent probing packets using Internet Control Message Protocol (ICMP) instead of Transmission Control Protocol (TCP) in [7]. More specifically, as in the `ping` program, the source host sends out a series of ICMP Echo Request to the destination host, and the destination host returns ICMP Echo Reply messages. Here an ICMP Echo message is regarded as a probing packet. The original `ping` program sends packets with fixed time interval (one second). We modified it so that variable inter-departure times can be obtained.

The size of a probing packet in ICMP usually is limited when it is sent via routers/switches since some routers/switches will cut it off if its size is larger than their upbounds. For example, if a probing packet is sent from a host in the ECE department Local Area Networks (LAN) of UNO, when the destination is `www.uno.edu` (via a single switch in the LAN), the size cannot be larger than 4 Kbytes; when the target is `www.computrols.com`, the size cannot be larger than 2 Kbytes (multiple routers/switches).

In our experiment, we use a common source host in the ECE department of UNO, which is running windows XP system. Eight different destination hosts were chosen, which include both the ones inside the LAN (without switch and with one single switch) and those outside the LAN (with multiple routers/switches). For each destination, we collected RTTs as a time series data using our modified `ping` program (`pingmachine`). Two different probing packet sizes (512 bytes and 1024 bytes) were used in the experiment, i.e., for each destination, there are two time series data. The timeout as well as

No.	Target	bytes = 512	bytes = 1024
1	enee613-2000.uno.edu	< 0.0001	< 0.0001
2	www.sina.com	0.0273	0.0325
3	www.utd.edu	0.0104	0.0112
4	www.yahoo.com	0.0174	0.0176
5	www.uno.edu	0.0020	0.0017
6	www.google.com	0.1160	0.1698
7	www.wenxuecity.com	0.0081	0.0080
8	216.107.90.145	0.0110	0.0111

Table 1: Packet loss rate

inter-departure time were set as 0.5 second. Each data collection lasted 24 hours. The path for each source-destination (SD) pair was checked by periodically running `tracert` command. It turned out that each route is stable, i.e., *persistent and prevalent* [14].

## 4.2 Packet Loss

Table 1 lists the loss rate of each set of data. The common source host is `Fusion1.uno.edu`. Note that the loss rate in the 6th SD pair is much higher than the others, which means more probing packets were cut off by the nodes between this SD pair than those between the other SD pairs.

## 4.3 Round Trip Times

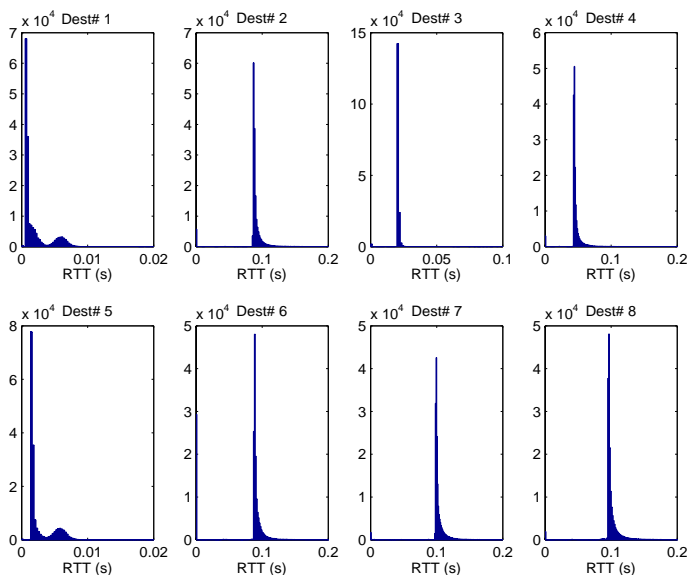


Figure 1: The histograms of the RTTs to different destinations, packet size = 1024 bytes

Figure 1 provides the histograms of the RTTs to different destinations when the packet size is 1024 bytes. There is no clear structure in each distribution.

Figure 2 shows the time plot of a time series data collected from a remote destination ([www.yahoo.edu](http://www.yahoo.edu)). Although this time series is not stationary, in short time period we can still assume it as stationary (i.e., short-term stationary) so that we can use a simple time series model to do prediction.

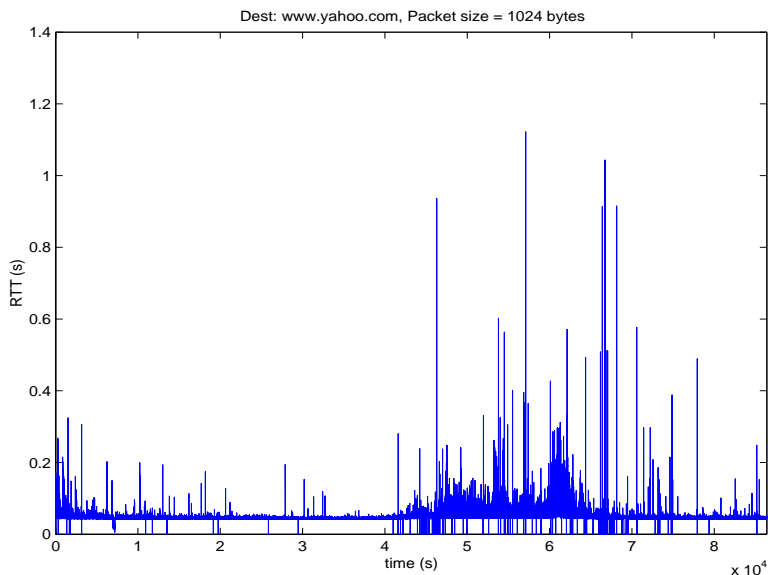


Figure 2: A RTTs data collected from a remote destination

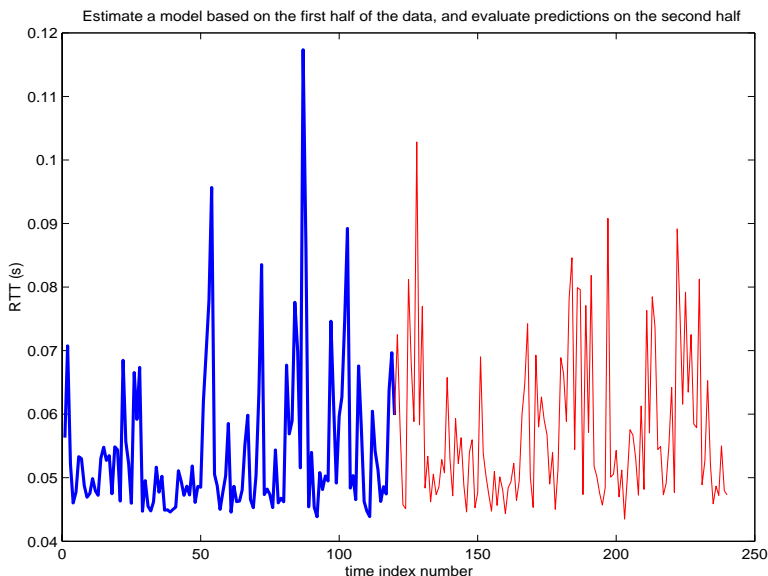


Figure 3: A segment of a RTTs data

The time series models of our data vary in long time range. Here we just give an example to get one specific model from a segment of the data. Figure 3 is an arbitrary segment of the data in

Figure 2. We estimate a model based on the first half of the data, and evaluate predictions on the second half. We set the prediction interval  $l = 2, 10, 20, 30$  samples (i.e., 1, 5, 10, 15 seconds because the sample interval  $\tau = 0.5s$ ) to do  $l$ -step ahead prediction.

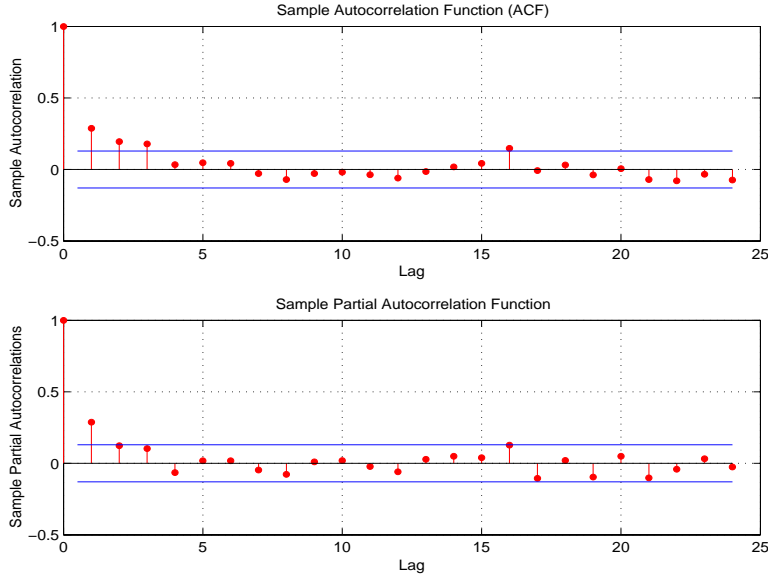


Figure 4: ACF and PACF of a data segment

Figure 4 shows the plots of the sample Autocorrelation Function (ACF) and sample Partial Autocorrelation Function (PACF). The bounds in the figure denote a 95% confidence interval indicating that the sample ACF/PACF estimation errors inside this interval are close to zero by 95% confidence. We choose the model structure as ARMA(1,3) by inspecting the ACF/PACF (the values which fall in the 95% confidence interval are regarded as zero).

The prediction results are compared by the time plots in Figure 5. From the plots we can see the prediction error increases when the prediction interval increases.

At last we check the ACF of the residuals (prediction errors) associated with the data. It turned out (refer to Figure 6) that there are two points outside the 99% confidence interval (the solid region in the figure) but very close to the bounds, which means this ARMA(1,3) model is not ideal but just a useful approximation in practice.

## 5 Conclusion and Discussion

In this report, time series methods for predicting end-to-end Internet delay have been discussed.

Two classes of methods, ARIMA scheme and state-space approach, have been described and compared. For state-space models, there are two key advantages relative to the ARIMA models: in structure and in computation [5]. By determining the *states* of the system, the useful information for prediction has been summarized efficiently. In this sense we say state-space models have better structures than ARIMA models. In addition, state-space models are Markovian in nature and their

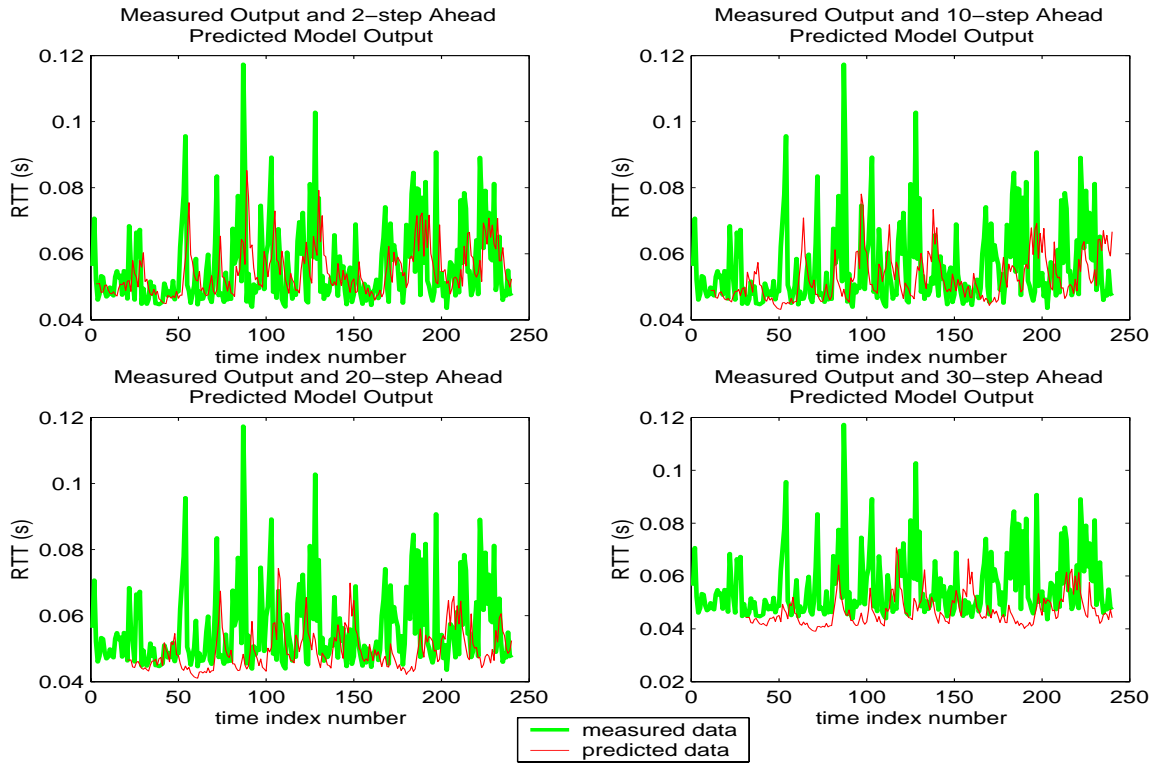


Figure 5: Comparison of prediction results

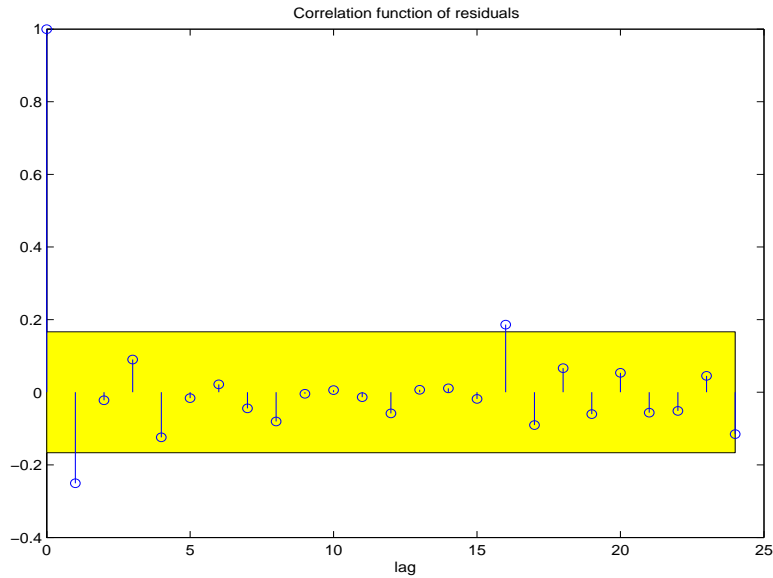


Figure 6: Residual analysis

dynamic or evolution through time is quite transparent. Consequently, the calculations needed to implement such models are recursive (e.g., Kalman filtering) and we can handle increasingly large models and datasets without a disproportionate increase in the computational burden. In contrast,

the computation complexity of the ARIMA modeling usually increases with the size of the datasets and the order of the MA part.

However considering the complexity of the Internet and the unclear impact of each issue involved, *black box* modeling, including ARIMA models and ARIMAX models, is more convenient to use. After the state-space representation of these models have been found, the later work (predicting and evaluating the performance) is almost the same as in control engineering.

The last issue that should be addressed here is that using any single ARIMA model cannot give the whole picture of the network dynamics and the prediction interval is limited (only for short-term prediction). Considering the path properties and the routing behavior, e.g., change of routing path, merge and split of traffic (for details, refer to [1], [11], [12]), it is reasonable to view the Internet as a *hybrid system* [9]. *Hybrid estimation* is a powerful tool for dealing with complex systems because a complex system may be decomposed into simpler subsystems with distinct structures. For the further information, please refer to our fourth report.

## References

- [1] M. Allman and V. Paxson. On estimating end-to-end network path properties. In *SIGCOMM*, pages 263–274, 1999.
- [2] G. E. P. Box and G. M. Jenkins. *Time-Series Analysis: Forecasting and Control*. Holden Day, San Francisco, 1976.
- [3] C. Chatfield. *The Analysis of Time Series: An Introduction*. CRC, Boca Raton, FL, fifth edition, 1996.
- [4] C. Chen. *Linear System Theory and Design*. Oxford Univ. Pr., Aug. 1998.
- [5] J. Durbin. The state space approach to time series analysis and its potential for official statistics. *Australia and New Zealand J. of Statistics*, 42:1–23, 2000.
- [6] A. C. Harvey. *Time Series Models*. The MIT Press, Cambridge, Massachusetts, second edition, 1992.
- [7] V. Jacobson. Congestion avoidance and control. In *Proc. ACM Sigcomm'88*, pages 314–329, Stanford, CA, Aug. 1988.
- [8] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, Englewood Cliffs, NJ, 1993.
- [9] X. R. Li. Hybrid estimation techniques. In C. T. Leondes, editor, *Control and Dynamic Systems: Advances in Theory and Applications*, volume 76, pages 213–287, San Diego, 1996. Academic Press.
- [10] X. R. Li. *Applied Estimation and Filtering*. University of New Orleans, New Orleans, LA, 2002.

- [11] B. A. Mah and A. Downey. Estimating Bandwidth And Other Network Properties. In *ISMA Winter 2000 Workshop*, Dec. 8th 2000.
- [12] V. Paxson. End-to-end routing behavior in the Internet. *IEEE/ACM Transactions on Networking*, 5(5):601–615, 1997.
- [13] M. Pourahmadi. *Foundations of Time Series Analysis and Prediction Theory*. John Wiley & Sons, Inc., 2001.
- [14] M. Yang and X. R. Li. The major issues and factors in end-to-end Internet delay prediction. Technical report, University of New Orleans, Dec. 2002.
- [15] M. Yang and X. R. Li. A survey of the existing methods for predicting end-to-end internet delay. Technical report, University of New Orleans, April 2003.
- [16] K. Zhou and J. C. Doyle. *Essentials Of Robust Control*. Prentice Hall, Upper Saddle River, NJ, 1998.