Modeling End-to-End Packet Delay Dynamics of the Internet using System Identification

Hiroyuki Ohsaki[a], Masayuki Murata[b], and Hideo Miyahara[a]

[a]Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan

[b]Cybermedia Center, Osaka University
1-30 Machikaneyama, Toyonaka, Osaka 567-0043, Japan

Understanding the end-to-end packet delay dynamics of the Internet is of crucial importance since (1) it directly affects the QoS (Quality of Services) of realtime applications, and (2) it enables us to design an efficient congestion control mechanism for both realtime and non-realtime applications. In this paper, we propose a novel approach for modeling the end-to-end packet delay dynamics of the Internet. The key idea of our approach is treating the network, seen by specific source and destination hosts, as a *black-box*, and modeling the end-to-end packet delay dynamics using *system identification*, having been throughly used in the field of control engineering. The end-to-end packet delay dynamics is modeled as a SISO (Single-Input and Single-Output) system. The input to the system is a packet inter-departure time from the source host, and the output from the system is an end-to-end packet delay variation measured by the destination host. In this paper, the ARX (Auto-Regressive eXogenous) model is used and its coefficients are determined using system identification. Several topics such as model validation and selection of the orders of the ARX model are also discussed. We show that the ARX model accurately captures the end-to-end packet delay dynamics if the orders of the ARX model are appropriately chosen. We also show that the effect of other UDP and TCP traffic can be modeled by white noise by using the end-to-end packet delay variation.

1. Introduction

In the past decade, the Internet has been explosively growing in scale as well as in population after introduction of the WWW (World Wide Web). In January 1997, only 16 million computers were connected to the Internet, but it has jumped to more than 55 million computers in June 1999 [1]. Because of such changing nature of the Internet, nobody knows the current network topology of the Internet. Such uncertainty of the Internet makes it very difficult, but also challenging, to analyze and understand the end-to-end packet behavior of the Internet.

Understanding the end-to-end packet delay dynamics of the Internet is of crucial importance since (1) it directly affects the QoS (Quality of Services) of realtime applications, and (2) it enables us to design an efficient congestion control mechanism for both realtime and non-realtime applications. For non-realtime applications, a delay-based approach for congestion control mechanisms, rather than a loss-based approach as used in TCP (Transmission Control Protocol), has been proposed (e.g., [2, 3]). The main advantage of such a delay-based approach is, if it is properly designed, packet losses can be prevented by anticipating impending congestion from increasing packet delays.

For a long time, the queuing theory has been extensively used as a powerful too to analyze both circuit-switched and packet-switched networks. In general, the queuing theory assumes stationarity of the network, and allows us to obtain several performance measures such as the average packet delay and the average packet loss probability. The stringent limitation of the queuing theory is its impossibility to analyze the *dynamical behavior* of the network. Several measurement-based studies suggest that the end-to-end packet behavior in the Internet is quite dynamical [4-6]. Another approach, being different from the queuing theory, should therefore be taken to investigate the packet delay dynamics of the Internet.

In this paper, we propose a novel approach to model the end-to-end packet delay dynamics of the Internet. The key idea of our approach is treating the network, seen by specific source and destination hosts, as a *black-box*, and modeling the end-to-end packet delay dynamics using *system identification*, having been throughly used in the field of control engineering. The end-to-end packet delay dynamics is modeled as a SISO (Single-Input and Single-Output) system. The input to the system is a packet inter-departure time from the source host, and the output from the system is an end-to-end packet delay variation measured by the destination host. In this paper, the ARX (Auto-Regressive eXogenous) model is used and its coefficients are determined using system identification. Several topics such as model validation and selection of the orders of the ARX model are also discussed.

The main objective of this paper is to construct a mathematical model that can be used, in particular, to design a delay-based congestion control mechanism. Once an appropriate model of the end-to-end packet delay dynamics is obtained, it is possible to apply the *optimal control theory* to design an efficient delay-based congestion control mechanism (for an example application of the optimal control theory, see, e.g., [7, 8]). We use the ARX model in this paper since it is simple but also suitable for application of the control theory. The ARX model is easy to handle, and its coefficients are easily determined with a little computational burden.

This paper is organized as follows. In Section 2, we summarizes related works in the literature. In Section 3, a black-box approach for modeling the end-to-end packet delay dynamics of the Internet is explained, followed by introduction of the ARX model. In Section 4, simulation experiments are performed and a few sets of input and output data are collected. In Section 5, coefficients of the ARX model is determined from the input and output data using system identification. Several topics such as model validation and selection of the orders of the ARX model are also discussed. In Section 6, we discuss several possible applications of our approach. Section 7 concludes this paper with a few remarks.

2. Related Works

In the literature, there have been several measurement-based studies regarding the end-to-end packet delay [4, 5, 9, 10] and the end-to-end path characteristics [6, 11]. In [4], the authors have examined the end-to-end packet delay and loss behavior in the Internet using small UDP probe packets. In [5], the authors have examined the correlation between packet delay and packet loss experienced by a continuous-media traffic source based on measurements of per-packet delays and packet loss. In [9], a large number of TCP measurements have been used to discuss two estimation problems: estimation of the retransmission timer (RTO) for a TCP connection, and estimation of the available bandwidth. In [10], the authors have presented an approach to characterize loss and delay characteristics of a transmission link based on end-to-end multicast measurements. In [6], the packet dynamics of the Internet have been analyzed based on measurements of about 20,000 TCP data transfers. In [11], the routing behavior of the Internet has been analyzed based on measurements of about 40,000 end-to-end traceroute results. However, those studies are limited to statistical behavior of the end-to-end packet delays and/or path characteristics. In other words, the end-to-end packet delay dynamics of the Internet, which is the main concern of this paper, has not been investigated.

Besides analyses of the end-to-end packet delay, another area of measurement-based studies is regarding black-box modeling of the network traffic [12-15]. In [12], the authors have proposed a traffic model for wide-are TCP traffic by characterizing several distributions of, for example, the packet inter-arrival time and the number of bytes transferred. In [13], the authors have proposed a fast algorithm to construct a CMRP (Circulant Modulated Rate Process) for traffic modeling. In [14], CMRP and ARMA (Auto-Regressive Moving Average) have been discussed as a traffic model. In [15], a measurement-based tool for traffic modeling and queuing analysis has been developed, which uses CMRP (Circulant Modulated Poisson Process) for a traffic model. Those studies are closely related to our black-box modeling approach, but there is a significant difference. Those studies have focused on traffic modeling based only on outputs (i.e., observed amount of traffic). On the contrary, this paper focuses on modeling the end-to-end packet delay dynamics based on both inputs (i.e., packet inter-departure time) and outputs (i.e., end-to-end packet delay variation). In other words, this paper focuses on how the end-to-end packet delay of a packet sent from a source host is affected by its past packet transmission process.

3. Black-Box Modeling using ARX Model

As depicted in Fig. 1, the network seen by a specific source and destination pair, including underlying protocol layers (e.g., physical, data-link, and network layers), is considered as a black-box. Our goal of this paper is to model a SISO system describing the end-to-end packet delay dynamics, i.e., the relation between a packet sending process from the source host and its resulting end-to-end packet delay observed at the destination host. Effects of other traffic (i.e., packets coming from other hosts) are modeled as *noise*. As the input to the system, we use a *packet inter-departure time* from the source host, i.e., the interval between two consecutive packet transmissions from the source host. Use of the packet inter-departure time is straightforward since, as suggested by the queuing theory, it directly affects the end-to-end packet delay. On the contrary, as the output from the system, we use an *end-to-end packet delay variation* measured by the destination host, i.e., the difference in two consecutive end-to-end packet delays. We choose the end-to-end packet delay variation, instead of the end-to-end packet delay itself, as the output from the system (see Fig. 2). This is to reduce unstationarity of noise (i.e., effect of other traffic) on the measured end-to-end packet delay since, at a packet-level time scale, the aggregated network traffic is not stationary [16].

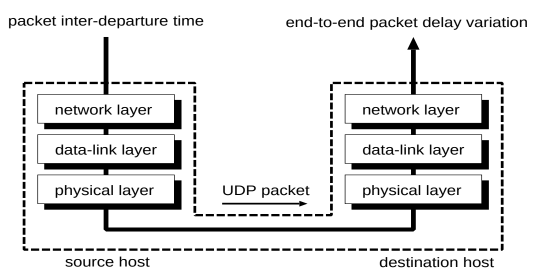


Figure 1. Modeling end-to-end packet delay dynamics as SISO (Single-Input and Single Output) system.

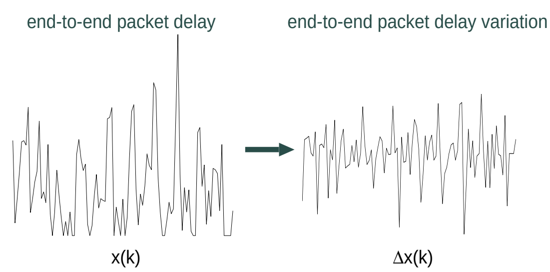


Figure 2. Using end-to-end packet delay variation to reduce unstationarity of noise.

In this paper, the ARX model is used and its coefficients are determined using system identification [17]. Figure 3 illustrates the fundamental concept of using the ARX model for modeling the packet delay dynamics. The input to the ARX model is a packet inter-departure time from the source host, and the output from the ARX model is an end-to-end packet delay variation measured by the destination host. Effects of other traffic (i.e., packets coming from other hosts) are modeled as the noise to the ARX model. Letting $u(k)$ and $y(k)$ be the input and the output data at slot k, the ARX model is defend as

$$\begin{aligned} A(q)\,y(k) &= B(q)\,u(k-n_d)+e(k) \qquad (1)\\ A(q) &= 1+a_1q^{-1}+\ldots+a_{n_a}q^{-n_a}\\ B(q) &= b_1+b_2q^{-1}+\ldots+b_{n_b}q^{-n_b+1}\end{aligned}$$

where $e(k)$ is unmeasurable disturbance (i.e., noise), and q^{-1} is the delay operator; i.e., $q^{-1}u(k) \equiv u(k-1)$. The numbers n_a and n_b are the orders of respective polynomials. The number n_d is the number of delays from the input to the output. For compact notation, ζ is introduced as

$$\zeta = [n_a, n_b, n_d] \qquad (2)$$

In this paper, $u(k)$ and $y(k)$ correspond to the k-th packet inter-departure time and the k-th end-to-end packet delay variation. All coefficients of the polynomials, a_n and b_n, are parameters of the ARX model, and are to be determined from input and output data using system identification. Refer to [17] for the detail of the ARX model.

Our approach of a black-box modeling using the ARX model is distinctive from other black-box approaches for modeling network traffic using the AR (Auto-Regressive) model or the ARMA (Auto-Regressive Moving Average) model [14, 18, 19]. Figure 4 illustrates a typical usage of the AR model or the ARMA model for modeling network traffic. Comparing Figs. 3 and 4, the ARX model has the input whereas either the AR model or the ARMA model does not. In other words, only the ARX model can represent the dynamics, i.e., the relation how the past input data affects the future output data.

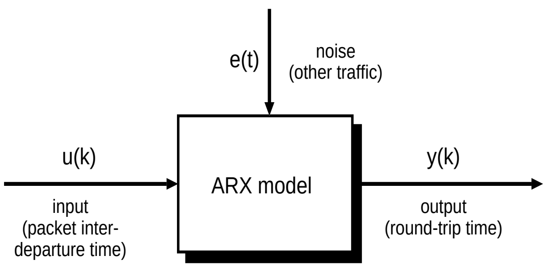


Figure 3. ARX model for modeling end-to-end packet delay dynamics.

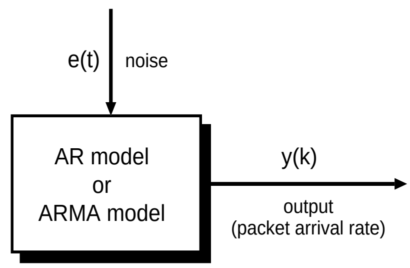


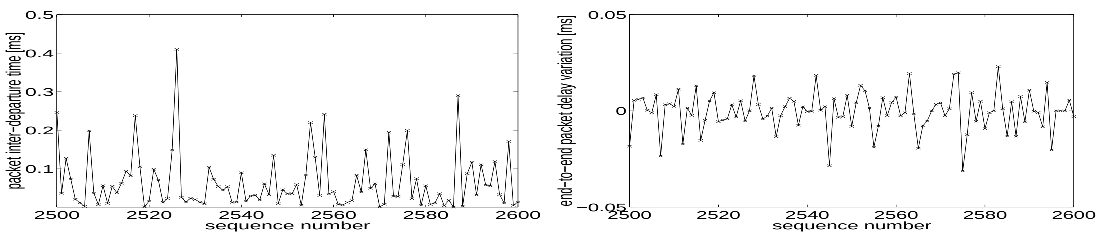
Figure 4. AR model or ARMA model for modeling network traffic.

However, the ARX model has a drawback for modeling the packet delay dynamics; i.e., the ARX model is a linear time-invariant model, so it cannot rigorously capture non-linearity of the packet delay dynamics. However, it should be noted that the ARX model is applicable in various control engineering problems. This is because non-linear dynamical systems operating around the stable point can be well approximated by a linear system [20]. In Section 5, we will investigate how accurately the end-to-end packet delay dynamics can be described by the ARX model.

4. Input and Output Data Collection from Simulation

For determining coefficients of the ARX model, a few sets of input and output data are collected from simulation using *ns2*. The simulation model consists of 10 source–destination pairs and a single bottleneck link. We consider two scenarios: (1) each source host sends only UDP (User Datagram Protocol) packets and (2) each source host sends both UDP and TCP packets. These scenarios will be referred to as the *UDP case* and the *UDP + TCP case*, respectively. In both cases, a source host intermittently generates UDP packets to the destination host. The packet inter-departure time from the source host is randomized according to the exponential distribution. The exponential distribution is used since it is one of ideal input data for system identification [17]. The average transmission rate of UDP packets is set to 128 [Kbit/s]. For each UDP packet received, the destination host records the end-to-end packet delay. In the UDP + TCP case, a source host sends TCP packets as well as UDP packets to the corresponding destination host using TCP Reno. Both routers are Drop-Tail routers having a shared buffer of 100 packets. Other simulation parameters are: the packet size is fixed at 1,000 [byte], the bottleneck link bandwidth is 1.5 [Mbit/s], the propagation delay of the bottleneck link is 1.0 [ms], and propagation delays of access links are ranged from 0.1 to 1.0 [ms].

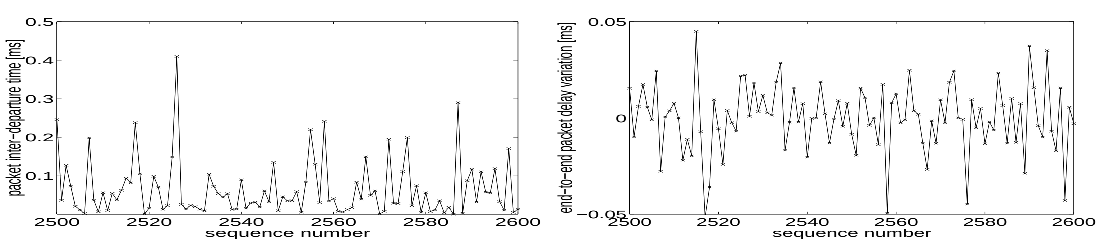
In both the UDP and the UDP + TCP cases, we have collected both the packet inter-departure time $u(k)$ and the end-to-end packet delay variation $y(k)$ sent from a single



(a) Packet inter-departure time $u(k)$ (b) End-to-end packet delay variation $y(k)$

Figure 5. Simulation result (UDP case).

source host. Shown in Figs. 5 and 6 are the inter-departure time $u(k)$ and the end-to-end packet delay variation $y(k)$ for the UDP case and the UDP + TCP case, respectively. Comparison of two figures suggests that the amplitude of the end-to-end packet delay variation gets larger when TCP traffic exists. This is because TCP forces the network to a slightly congested state, leading to a larger waiting time in the router's buffer. On



(a) Packet inter-departure time $u(k)$ (b) End-to-end packet delay variation $y(k)$

Figure 6. Simulation result (UDP + TCP case).

the contrary, as one might expect, a negative correlation between the inter-departure time and the end-to-end packet delay variation can be observed in both cases. Namely, when the packet inter-departure time is small, the end-to-end packet delay variation tends to become large, and vice versa. However, such a negative correlation seems not so strong. This is because the end-to-end delay of a UDP packet is disturbed by other UDP and TCP packets since the bottleneck node has a shared FIFO buffer. To view the correlation between $u(k)$ and $y(k)$ more clearly, scatter plots for input and output data of 1,000 packets are shown in Figs. 7 and 8 for the UDP case and the UDP + TCP case, respectively. In these figure, a least-squared fit line is also plotted. These figures suggest a weak negative correlation between the inter-departure time and the end-to-end packet delay variation. In spite of such a weak correlation, as will be shown in Section 5, the

ARX model can accurately capture the end-to-end packet delay dynamics.

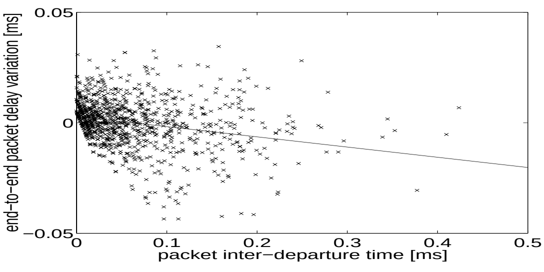


Figure 7. Scatter plot of packet inter-departure time vs. end-to-end packet delay variation (UDP case).

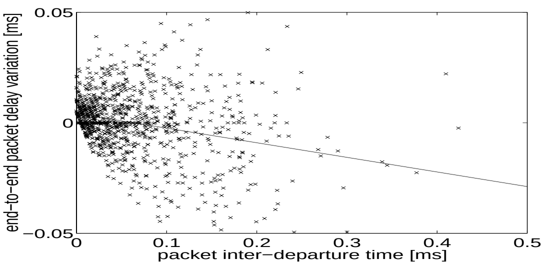


Figure 8. Scatter plot of packet inter-departure time vs. end-to-end packet delay variation (UDP + TCP case).

5. System Identification and Discussions

The system identification problem for the ARX model is formulated as a minimization problem, where the cost function is given by a loss function [17]. Because of space limitation, only the outline are shown in this paper, and interested readers should refer to [17] for more detail. The following numerical examples are obtained using MATLAB and its System Identification Toolbox.

Let θ be a vector of all coefficients and $\psi(k)$ be a vector of all past n_a outputs and n_b inputs, respectively.

$$\theta = [a_1, \ldots, a_{n_a}, b_1, \ldots b_{n_b}]^T \tag{3}$$

$$\psi(k) = [-y(k-1), \ldots, -y(k-n_a), u(k-n_d-1), \ldots, u(k-n_d-n_b)] \tag{4}$$

Using Eq. 1, the output from the ARX model $\hat{y}(k|\theta)$ is given by

$$\hat{y}(k|\theta) = \psi^T(k)\,\theta \tag{5}$$

The loss function $V_N(\theta, Z^N)$ is defined as the sum of all squared prediction errors for N input and output data.

$$V_N(\theta, Z^N) = \frac{1}{N}\sum_{k=1}^{N}(y(k) - \hat{y}(k|\theta))^2 \tag{6}$$

where Z^n is the past input and output data defined as

$$Z^N = \{u(1), y(1), \ldots, u(N), y(N)\} \tag{7}$$

The solution $\hat{\theta}_N$ that minimizes the above loss function is easily obtained by the least squares method:

$$\hat{\theta}_N = \left[\sum_{k=1}^{N}\psi(k)\psi^T(k)\right]^{-1}\sum_{k=1}^{N}\psi(k)y(k) \tag{8}$$

Of all input and output data collected in Section 4, we use the input and output data of 100 packets ($2500 \leq k < 2600$) for coefficients determination and model validation. As an example, when $\zeta = [5, 5, 1]$, coefficients of the ARX model and their standard deviations are obtained from Eq. (8) as Tab. 1 (the UDP case) and Tab. 2 (the UDP + TCP case).

	a_1	a_2	a_3	a_4	a_5
coefficient	0.2617	0.2481	0.2819	0.1332	0.0493
standard deviation	0.1108	0.1137	0.1130	0.1098	0.1069
	b_1	b_2	b_3	b_4	b_5
coefficient	0.0035	0.0201	0.0005	0.0067	0.0100
standard deviation	0.0146	0.0145	0.0147	0.0147	0.0144

Table 1
Coefficients and standard deviations of the ARX model for $\zeta = [5, 5, 1]$ (UDP case).

	a_1	a_2	a_3	a_4	a_5
coefficient	0.1063	0.1279	0.0175	-0.0515	-0.0277
standard deviation	0.1064	0.1069	0.1094	0.1083	0.1079
	b_1	b_2	b_3	b_4	b_5
coefficient	0.0089	-0.0246	0.0058	-0.0111	0.0274
standard deviation	0.0267	0.0265	0.0267	0.0265	0.0253

Table 2
Coefficients and standard deviations of the ARX model for $\zeta = [5, 5, 1]$ (UDP + TCP case).

In what follows, we discuss how accurately the end-to-end packet delay dynamics is modeled by the ARX model. Figures 9 and 10 show comparisons between the measured data (solid line) and the model output (dotted line) for $\zeta = [5, 5, 1]$. More specifically, the measured data is the measured end-to-end packet delay variation $y(k)$, and the model output $y^*(k|\theta)$ is the simulated output from the ARX model, which is defined as

$$y^*(k|\theta) \;=\; \psi^{*T}(k|\theta)\,\theta \tag{9}$$

where

$$\psi^*(k|\theta) \;=\; [-y^*(k-1|\theta), \ldots, -y^*(k-n_a|\theta), u(k-n_d-1), \ldots, u(k-n_d-n_b)] \tag{10}$$

Note the difference between $\hat{y}(k|\theta)$ and $y^*(k|\theta)$; i.e., $\hat{y}(k)$ is a 1-step ahead prediction from the measured inputs and outputs, whereas $y^*(k|\theta)$ is a simulated output only from the measured inputs assuming zero noise. Figures 9 and 10 indicate that the ARX model does not capture the end-to-end packet delay dynamics. In both the UDP and the UDP

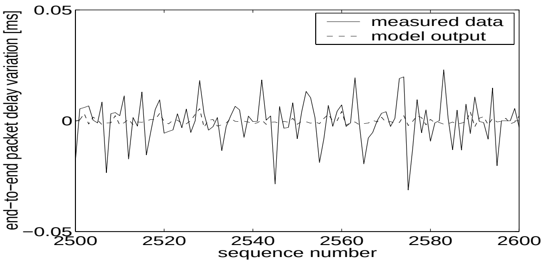


Figure 9. Comparison between measured data $y(k)$ and model output $y^*(k)$ for $\zeta = [5, 5, 1]$ for (UDP case).

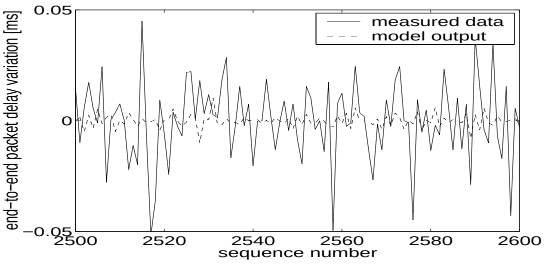


Figure 10. Comparison between measured data $y(k)$ and model output $y^*(k)$ for $\zeta = [5, 5, 1]$ for (UDP + TCP case).

+ TCP cases, the model output $y^*(k)$ is almost unchanged although the measured end-to-end packet delay variation excessively oscillates. Namely, the end-to-end packet delay dynamics cannot be modeled by the ARX model with $\zeta = [5, 5, 1]$.

However, as will be shown below, the ARX model can correctly model the end-to-end packet delay dynamics if ζ is chosen appropriately. We choose $\zeta = [18, 20, 1]$ for the UDP case and $\zeta = [18, 13, 1]$ for the UDP + TCP case. These values are chosen to minimize the AIC (Akaike's Information Theoretic Criterion) [17]. The AIC is defined as

$$AIC \simeq \log\left[(1 + \frac{2n}{N})V_N(\theta, Z^N)\right] \tag{11}$$

where n is the number of unknown parameters, i.e., $n_a + n_b$. Shown in Figs. 11 and 12 are comparisons between the measured data (solid line) and the model output (dotted line) for $\zeta = [18, 20, 1]$ (the UDP case) and $\zeta = [18, 13, 1]$ (the UDP + TCP case), respectively. It can be found that, in both cases, the model output $y^*(k|\theta)$ and the measured output $y(k)$ roughly coincide but slightly differ. This is because the measured end-to-end packet delay variation is disturbed by other traffic, which is unknown so that not included in the model output $y^*(k)$.

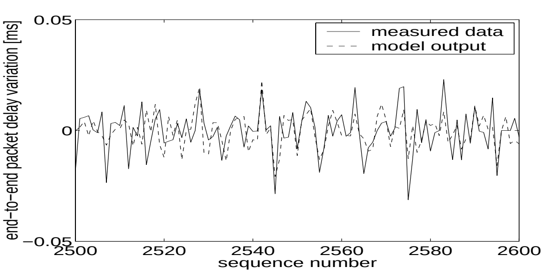


Figure 11. Comparison between measured data $y(k)$ and model output $y^*(k)$ for $\zeta = [18, 20, 1]$ for (UDP case).

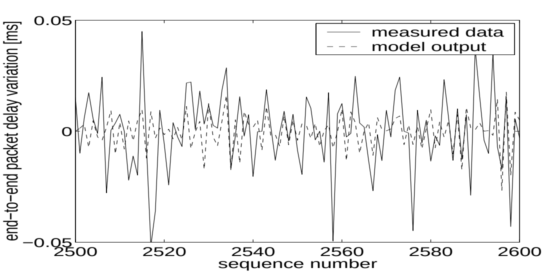


Figure 12. Comparison between measured data $y(k)$ and model output $y^*(k)$ for $\zeta = [18, 13, 1]$ for (UDP + TCP case).

To confirm the goodness of the ARX model obtained, the auto-correlation functions of residuals (i.e., prediction errors $e(k) \equiv y(k) - \hat{y}(k|\theta)$) are plotted in Figs. 13 and 14 for the UDP and the UDP + TCP cases, respectively. Figure 13 corresponds to the UDP case with $\zeta = [18, 20, 1]$, and Fig. 14 corresponds to the UDP + TCP case with $\zeta = [18, 13, 1]$. In these figures, 95% confidence interval is also shown. It is known that the ARX model is inappropriate if any auto-correlation of residuals for a positive lag is outside the confidence interval [17]. These figures show that all residuals are inside the confidence interval, indicating the end-to-end packet delay dynamics is well modeled by the ARX model. This also suggests that the effect of other UDP and TCP traffic can be modeled by a white noise.

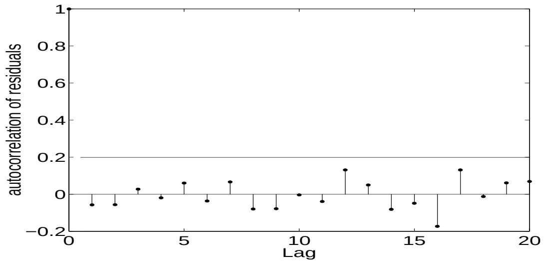


Figure 13. Autocorrelation of residuals $e(k)$ for $\zeta = [18, 20, 1]$ (UDP case).

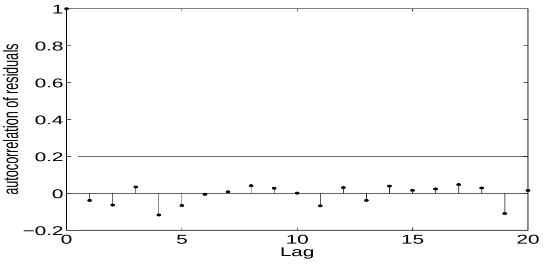


Figure 14. Autocorrelation of residuals $e(k)$ for $\zeta = [18, 13, 1]$ (UDP + TCP case).

Finally, we investigate how the choice of ζ affects the accuracy of the ARX model. In Figs. 15 and 16, the loss function for different values of n_a and n_b ($1 \leq n_a, n_b \leq 20$) is plotted for the UDP case and the UDP + TCP case, respectively. In these figures, n_d is fixed at 1. These figures indicate that the smaller the loss function is, the more accurate the ARX model becomes. It can be found that a larger value of n_a and/or n_b is desirable. In particular, it can be seen from Fig. 15 that n_a should be larger than 10 when there exist only UDP packets in the network.

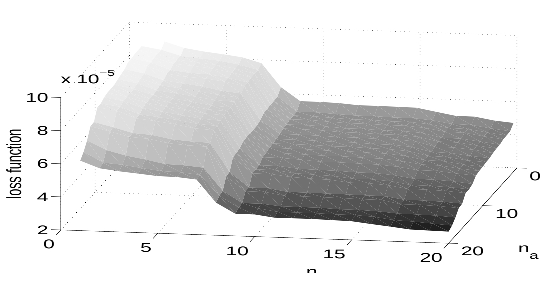


Figure 15. Loss function for different orders of the ARX model for $n_d = 1$ (UDP case).

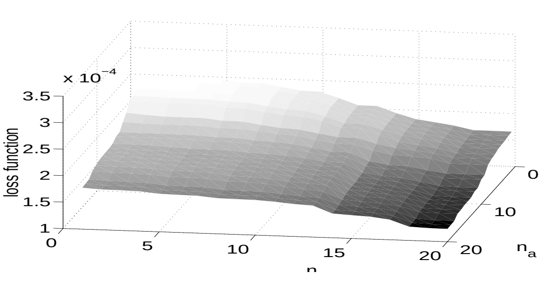


Figure 16. Loss function for different orders of the ARX model for $n_d = 1$ (UDP + TCP case).

6. Application Scenarios of Our Approach

We discuss several possible applications of our approach — modeling the end-to-end packet delay dynamics of the Internet using the ARX model. Although details of these topics are beyond the scope of this paper, it is worthwhile to discuss how our approach is applied to various problems. The first and straightforward application would be to use our approach to *understand* the end-to-end packet delay dynamics of the Internet. We can analyze the end-to-end packet delay dynamics through the ARX model obtained. Because the ARX model is one of LTI (Linear Time Invariant) models, various analysis techniques for LTI models in time- and frequency-domain can be utilized. The second application would be to *predict* the future end-to-end packet delay from the ARX model obtained. As have shown in Section 4, the end-to-end delay of a packet is considerably disturbed by other UDP and TCP packets. Hence, it is almost impossible to predict the far future end-to-end packet delay. However, the ARX model can predict the near future end-to-end packet delay. As noted in 1, the third and possibly most important application would be to *design* a congestion control mechanism. Once the ARX model capturing the end-to-end packet delay dynamics is obtained, it is possible to apply the optimal control theory to design an efficient delay-based congestion control mechanism. Congestion control of the Internet is a difficult problem because of its complexity such as heterogeneity of various network elements and non-negligible propagation delays. However, combination of the ARX model and the optimal control theory would help us to design a more efficient congestion control mechanism.

7. Conclusion

In this paper, we have proposed a novel approach to model the end-to-end packet delay dynamics of the Internet using system identification. We have investigated how accurately the ARX model captures the end-to-end packet delay dynamics. The key idea is to model the network, seen by specific source and destination hosts, as the ARX model. The input to the ARX model is the packet inter-departure time from the source host, and the output from the ARX model is the end-to-end packet delay variation measured by the destination host. We have collected input and output data from simulation experiments using only UDP packets and both UDP and TCP packets. Through several numerical examples, we have shown that the ARX model accurately captures the end-to-end packet delay dynamics if the orders of the ARX model is appropriately chosen. We have also shown that the effect of other UDP and TCP traffic can be well modeled by a white noise as far as the end-to-end packet delay variation is concerned.

As future work, it would be of importance to investigate effectiveness of the ARX model using a through set of input and output data. We are currently working on building the ARX model describing the end-to-end packet delay dynamics based on measured input and output data from the operating LAN and WAN.

REFERENCES

1. S. Floyd and V. Paxson, "Why we don't know how to simulate the Internet," Oct. 1999. available at `http://www.aciri.org/floyd/papers/wsc.ps`.

2. R. Jain, “A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks,” *ACM Computer Communication Review*, vol. 19, pp. 56–71, Oct. 1989.
3. L. S. Brakmo, S. W. O’Malley, and L. L. Peterson, “TCP Vegas: New techniques for congestion detection and avoidance,” in *Proceedings of ACM SIGCOMM ’94*, pp. 24–35, Oct. 1994.
4. J.-C. Bolot, “Characterizing end-to-end packet delay and loss in the Internet,” *Journal of High-Speed Networks*, vol. 2, pp. 305–323, Dec. 1993.
5. S. B. Moon, J. Kurose, P. Skelly, and D. Towsley, “Correlation of packet delay and loss in the Internet,” tech. rep., Department of Computer Science, University of Massachusetts, USA, Jan. 1998.
6. V. Paxson, “End-to-end Internet packet dynamics,” in *Proceedings of ACM SIGCOMM*, pp. 139–152, Sept. 1997.
7. Z. Pan, E. Altman, and T. Bassar, “Robust adaptive flow control in high speed telecommunication networks,” in *Proceedings of the 35th IEEE Conference on Decision and Control*, Dec. 1996.
8. E. Altman and T. Bassar, “Optimal rate control for high speed telecommunication networks,” in *Proceedings of 34th IEEE Conference on Decision and Control*, Oct. 1995.
9. M. Allman and V. Paxon, “On estimating end-to-end network path properties,” *ACM SIGCOMM ’99*, Aug. 1999.
10. A. Adams, T. Bu, R. Caceres, N. Duffield, T. Friedman, J. Horowitz, F. L. Presti, S. B. Moon, V. Paxson, and D. Towsley, “The use of end-to-end multicast measurements for characterizing internal network behavior,” *IEEE Communications*, May 2000.
11. V. Paxson, “End-to-end routing behavior in the Internet,” in *ACM SIGCOMM ’96*, pp. 25–38, May 1996.
12. R. Caceres, P. B. Danzig, S. Jamin, and D. J.Mitzel, “Characteristics of wide-area TCP/IP conversations,” in *ACM SIGCOMM ’91*, pp. 101–112, Sept. 1991.
13. H. Che and S.-Q. Li, “Fast algorithms for measurement-based traffic modeling,” *IEEE Journal on Selected Areas in Communications*, vol. 16, June 1998.
14. L. A. Kulkarni and S.-Q. Li, “Measurement-based traffic modeling: capturing important statistics,” *Journal of Stochastic Model*, vol. 14, no. 5, 1998.
15. S.-Q. Li, S. Park, and D. Arifler, “SMAQ: A measurement-based tool for traffic modeling and queueing analysis Part I - design methodologies and software architecture,” *IEEE Communications Magazine*, vol. 36, Aug. 1998.
16. Y. Zhang, V. Paxon, and S. Schenker, “The stationarity of Internet path properties: routing, loss, and throughput,” tech. rep., ACIRI, May 2000.
17. L. Ljung, *System identification — theory for the user.* Englewood Cliffs, N.J.: Prentice Hall, 1987.
18. O. Ait-Hellal, E. Altman, and T. Basar, “A robust identification algorithm for traffic models in telecommunications,” in *Proceedings of IEEE CDC ’99*, Dec. 1998.
19. A. Sang and S.-Q. Li, “A predictability analysis of network traffic,” in *Proceedings of IEEE INFOCOM 2000*, Mar. 2000.
20. J.-N. Juang, *Applied system identification.* Englewood Cliffs, N.J.: Prentice Hall, 1994.