

Measurement-Based Modeling of Internet Round-Trip Time Dynamics Using System Identification

Hiroyuki Ohsaki¹, Mitsushige Morita², and Masayuki Murata¹

¹ Cybermedia Center, Osaka University
1-30 Machikaneyama, Toyonaka, Osaka, Japan
{osaki, murata}@cmc.osaka-u.ac.jp

² Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka, Japan
m-morita@ics.es.osaka-u.ac.jp

Abstract. Understanding the end-to-end packet delay dynamics of the Internet is of crucial importance since it directly affects the QoS (Quality of Services) of various applications, and it enables us to design an efficient congestion control mechanism. In our previous studies, we have measured round-trip time of the Internet, and have modeled its dynamics by the ARX (Auto-Regressive eXogenous) model using system identification. As input and output data for the ARX model, we have used the packet inter-departure time from a source host and the corresponding round-trip time variation measured by the source host. In the current paper, for improving the model accuracy, we instead use the packet transmission rate from the source host and the average round-trip time measured by the source host. Using input and output data measured in working LAN and WAN environments, we model the round-trip time dynamics by determining coefficients of the ARX model using system identification. Through numerical examples, we show that in LAN environment, the round-trip time dynamics can be accurately modeled by the ARX model. We also show that in WAN environment, the round-trip time dynamics can be accurately modeled when the bottleneck link is shared by a small number of users.

1 Introduction

In the past decade, the Internet has been explosively growing in scale as well as in population after the 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 56 million computers in July 1999 [1]. Because of the 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 various 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.

In [4,5], we have proposed a novel approach for modeling the end-to-end packet delay dynamics of the Internet using system identification. In [4,5], we have regarded the network, seen by a specific source host, as a dynamic SISO (Single-Input and Single Output) system. We have modeled the round-trip time dynamics using the ARX (Auto-Regressive eXogenous) model. In those studies, the input to the system was the packet inter-departure time from the source host, and the output was the round-trip time variation between two adjacent packets. Using measured data obtained in wired and wireless LAN environments, we have investigated how accurately the ARX model can capture the round-trip time dynamics of the Internet. We have found that the ARX model can capture the round-trip time dynamics when the network is moderately congested. We have also found that, when the network is not congested or the measured round-trip time is noisy, the ARX model fails to capture the dynamics.

This paper is a direct extension of [4,5], and has three major changes: (1) refined definition of input and output data for improving the model accuracy, (2) experimentations in LAN and WAN environments, and (3) use of two model validation methods in time domain and frequency domain. The first change is to refine the definition of the input and the output for the ARX model. The input to the system is changed to an *instantaneous* packet transmission rate from the source host during a fixed sampling interval. Also the output is changed to an *instantaneous* average round-trip time observed by the source host during a fixed sampling interval. In [4,5], the sampling interval is not fixed since it is dependent on the packet sending/receiving process at the source host. On the contrary, in this paper, the sampling interval is fixed, so that the model accuracy is expected to be improved since system identification originally assumes a fixed sampling interval. The objective of the second change is to investigate how the model accuracy is related to a network configuration. We collect input and output data for system identification in LAN and WAN environments, and build a model for the round-trip time dynamics. The third change is to evaluate the model accuracy in a more rigorous manner. We evaluate the model accuracy in frequency domain as well as in time domain. In [5], the accuracy of the ARX model was evaluated only in time domain; that is, we have compared the simulated outputs from the ARX model (i.e., round-trip times) with the actual round-trip times. In this paper, we also examine the model accuracy in frequency domain using a spectral analysis. Through numerical examples, we show that in LAN environment, the round-trip time dynamics can be accurately modeled by the ARX model. We also show that in WAN environment, the round-trip time dynamics can be accurately modeled when the bottleneck link is shared by a small number of users.

This paper is organized as follows. In Section 2, a black-box approach for modeling the round-trip time dynamics of the Internet is explained. In Section 3, we discuss several measurement methods of the round-trip time, in particular, for collecting input and output data for system identification. We also explain three network environments in which input and output data, used for the model identification and for the model validation, are collected. Section 4 shows several measurement and modeling results, and discuss how accurately the ARX model can capture the round-trip time dynamics in various network configurations. Section 5 concludes this paper with a few remarks.

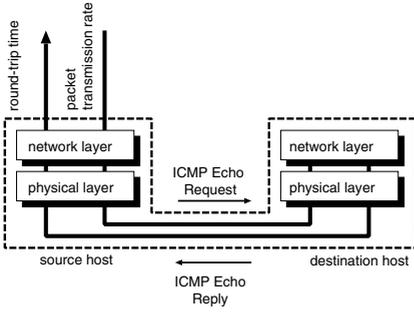


Fig. 1. Modeling round-trip time dynamics as SISO system.

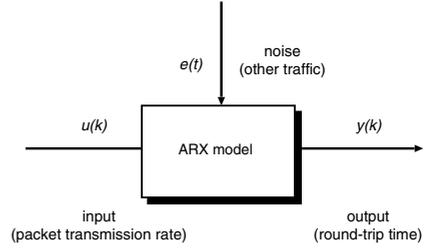


Fig. 2. ARX model for modeling round-trip time dynamics.

2 Black-Box Modeling Using ARX Model

As depicted in Fig. 1, the network seen by a specific source host, 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 round-trip time dynamics: i.e., the relation between a packet sending process from the source host and its resulting round-trip time observed at the source host. Effects of other traffic (i.e., packets coming from other hosts) are modeled as *noise*. As the input to the system, we use an *instantaneous* packet transmission rate from the source host: i.e., the packet transmission rate during a fixed sampling interval. As the output from the system, we use an *instantaneous* average round-trip time measured by the source host: i.e., the average round-trip time during a fixed sampling interval.

In this paper, the ARX model is used and its coefficients are determined using system identification [6]. Figure 2 illustrates a fundamental concept of using the ARX model for capturing the round-trip time dynamics. The input to the ARX model is a packet transmission rate from the source host, and the output from the ARX model is a round-trip time measured by the source 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 at slot k , respectively, the ARX model is defined as

$$\begin{aligned}
 A(q) y(k) &= B(q) u(k - n_d) + e(k) \\
 A(q) &= 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a} \\
 B(q) &= b_1 + b_2 q^{-1} + \dots + b_{n_b} q^{-n_b + 1}
 \end{aligned}$$

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 polynomials. The number n_d corresponds to delays from the input to the output. All coefficients of the polynomials, a_n and b_n , are parameters of the ARX model, and are to be identified from input and output data. Refer to [6] for the detail of the ARX model and system identification. For compact notation, ζ and θ are introduced as

$$\begin{aligned}
 \zeta &= [n_a, n_b, n_d] \\
 \theta &= [a_1, \dots, a_{n_a}, b_1, \dots, b_{n_b}]^T
 \end{aligned}$$

In [5], we have defined the input as the packet inter-departure time from the source host, and the output as the round-trip time variation measured by the source host. Although the ARX model, with such input and output definition, can capture the round-trip time dynamics to some extent, the model accuracy is not good. It is possibly because of the non-fixed sampling interval. Namely, use of a fixed sampling interval is generally assumed in system identification, however, in [5], the sampling interval is not fixed since it is dependent on the packet sending/receiving process at the source host. In this paper, we therefore use a fixed sampling interval for improving the model accuracy; that is, the input to the system is the packet transmission rate during a fixed sampling interval, and the output from the system is the average round-trip time during a fixed sampling interval. More specifically, the input $u(k)$ and the output $y(k)$ are defined as follows. Let $t_s(i)$ be the time at which the i th packet is injected into the network, and $t_r(i)$ be the time at which the i th ACK packet is received by the source host. We further introduce $l(i)$ as the size of the i th packet including the IP header, and T as the sampling interval. Then, $u(k)$ and $y(k)$ are defined as

$$u(k) = \frac{\sum_{i \in \phi_s(k)} l(i)}{T}$$

$$y(k) = \frac{\sum_{i \in \phi_r(k)} (t_r(i) - t_s(i))}{|\phi_r(k)|}$$

where $\phi_s(k)$ (or $\phi_r(k)$) is the set of packet numbers sent (or received) during k th sampling interval; i.e.,

$$\phi_s(k) \equiv \{n : kT \leq t_s(n) < (k+1)T\}$$

$$\phi_r(k) \equiv \{n : kT \leq t_r(n) < (k+1)T\}$$

3 Data Collection Using ICMP Packet

3.1 Measurement Method

For collecting input and output data from a real network, it is necessary to send a series of probe packets into the network, and to measure their resulting round-trip times. For sending a probe packet, one of the following protocols can be used.

- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)
- ICMP (Internet Control Message Protocol)

In what follows, we briefly discuss advantages and disadvantages of these protocols for sending a probe packet to collect input and output data, in particular, for system identification.

TCP has a feedback-based congestion control mechanism, which controls the packet sending process from a source host according to the congestion status of the network. Since it is an ACK-based protocol, it is easy for the source host to measure the round-trip time for each packet. However, because of such a feedback-based mechanism, TCP is not suitable for sending a probe packet for two reasons. First, although the input (i.e., the

packet transmission rate) should contain diverse frequencies for system identification purposes, the packet transmission rate of TCP would have limited frequencies. Second, regardless of many system identification techniques assuming an independence between the input and the output, the independence assumption cannot be satisfied with TCP since the packet transmission rate is dependent on the past round-trip times.

On the contrary, UDP has no feedback-based control. The packet transmission rate of UDP can be freely controlled. However, UDP is a one-way protocol. The destination host must perform some procedure to measure the round-trip time for each packet at the sender side. One possible way is to use *ICMP Destination Unreachable* message as in the *traceroute* program [7]. When the host receives a UDP packet to an unreachable port, it returns *ICMP Destination Unreachable* message to the source host. The source host can therefore measure the round-trip time by observing the elapsed time between the UDP packet transmission and the receipt of the corresponding ICMP packet. However, as specified in [8], generation of *ICMP Destination Unreachable* messages is limited to a low rate. Use of *ICMP Destination Unreachable* message is therefore not desirable to collect the input and output data for system identification.

ICMP is a protocol to exchange control messages such as routing information and node failures [9]. Since ICMP has no feedback-based control, the inter-departure time of ICMP packets can be freely controlled. Also it is easy to measure the round-trip time at the source host by using *ICMP Echo Request* and *ICMP Echo Reply* messages, as in the *ping* program. Although some network devices limit the rate of ICMP packets because of malicious use of them [10], such as a DoS (Denial of Service) attack, many network devices respond to *ICMP Echo Request* message and do not limit the rate of them.

In this paper, we therefore choose *ICMP Echo* message as a probe packet. More specifically, the source host sends a series of *ICMP Echo Request* messages to the destination host, and the destination host returns *ICMP Echo Reply* messages. We have modified the *ping* program to dynamically change the packet inter-departure time (originally fixed at one second). The destination host copies the payload of the received *ICMP Echo Request* message to the returning *ICMP Echo Reply* message. Thus, the *ICMP Echo Reply* packet contains the timestamp placed by the source host at its transmission time. This enables precise measurement of the round-trip time at the source host. Instead of measuring *ICMP Echo Request/Reply* packet sending/receiving time at the source, a *measurement host* is prepared (Fig. 3). It is for achieving reliable data measurement even when the source host sends or receives packets at a very high rate. As shown in Fig. 3, the *Ether TAP* copies all packets carried on the link, and sends copies to the measurement host; that is, all *ICMP Echo Request/Reply* packets sent from/to the source host are also delivered to the measurement host.

We use an active measurement approach for collecting data by sending probe packets to the network. This is because we want to know how accurately the ARX model can represent the round-trip time dynamics of the Internet. However, we intend to apply a passive measurement approach, which measures data by monitoring packets being transmitted in the network.

3.2 Network Environments

As the number of routers between source and destination hosts increases, the noise (i.e., effect of other traffic and measurement errors) contained in the output becomes large.

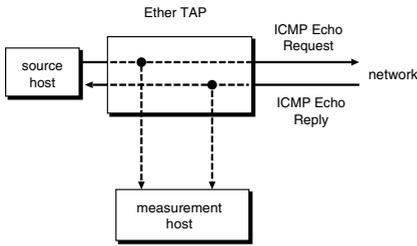


Fig. 3. Measurement host for reliable data measurement.

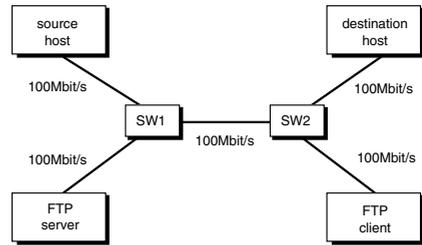


Fig. 4. Network N1 (LAN)

Besides, the dominant part of the round-trip time is a queuing delay at the bottleneck router. It is therefore important to choose network configurations, in which the input and the output are collected, by taking account of the number of routers and the location of the bottleneck link. In this paper, we measure packet sending/receiving times in three network configurations including LAN and WAN environments, and obtain the input $u(k)$ and the output $y(k)$. In LAN environment, it is expected that the ARX model can accurately model the round-trip time dynamics since the network topology is rather simple and the measured data would suffer little observation noise. On the contrary, in WAN environment, it is expected that the model accuracy is degraded compared to that in LAN environment since the network topology is complex. We use two network configurations for WAN environment. The difference in these WAN configurations is the location of the bottleneck link. In this paper, the following three network configurations (i.e., **N1**, **N2**, and **N3**) are used for collecting input and output data.

- Network N1 (LAN)

The network **N1** is LAN environment of a simple network configuration (Fig. 4). There exist two switches (SW1 and SW2) between source and destination hosts. All hosts and switches are connected to 100 Mbps LAN. The link between SW1 and SW2 also carries background traffic, as well as ICMP Echo Request/Replay packets exchanged between source and destination hosts. Namely, a bulk FTP transfer from a server (connected to SW1) to a client (connected to SW2) is performed during data collection.

- Network N2 (WAN with the bottlenecked access link)

The network **N2** is WAN environment of a complex network configuration, and the access link is the bottleneck between source and destination hosts (Fig. 5). The source host is connected to the Internet via 100 Mbps LAN, and the destination host is connected via 56 Kbps dial-up PPP link. At the time of measurement, the number of hops between source and destination hosts was 16, and the average round-trip time was 319.7 ms.

- Network N3 (WAN with the non-bottlenecked access link)

The network **N3** is WAN environment, and the access link is not the bottleneck between source and destination hosts (Fig. 6). The source host is connected to the Internet via 100 Mbps LAN. We have chosen `www.so-net.ne.jp` as the destination host. At the time of measurement, the number of hops between source and destination hosts was 16, and the average round-trip time was 36.89 ms.

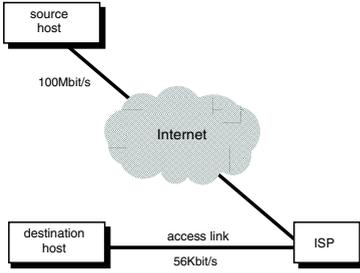


Fig. 5. Network N2 (WAN with the bottlenecked access link)

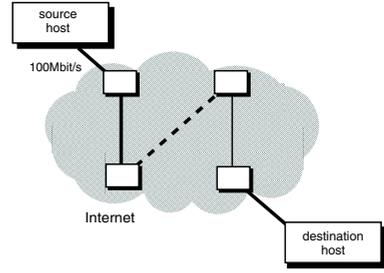


Fig. 6. Network N3: (WAN with the non-bottlenecked access link)

In the above three network configurations, we measured the packet sending/receiving time at the measurement host. The source host sent 20,000 ICMP Echo Request packets, and the timestamp of each ICMP Echo Request/Replay packet is recorded by the measurement host. The data collection was done at midnight of October 18, 2001. As we have explained in Section 2, the input $u(k)$ and the output $y(k)$ for system identification is calculated from measured packet sending/receiving times. We empirically choose the sampling interval T in each network configuration; that is, T is chosen for each sampling period to contain about five samples. In this paper, the packet inter-departure time from the source host is randomly changed, there might be a sampling period in which no packet is sent or received. If no packet is sent (or received) during k th sampling period, the input $u(k)$ (or the output $y(k)$) is not defined. In such a case, we use the minimum value of all past input (or output) data; i.e.,

$$u(k) = \min_{0 \leq i < k} (u(i))$$

$$y(k) = \min_{0 \leq i < k} (y(i))$$

4 Modeling from Measured Data

4.1 Choice of Model Orders and Number of Samples

In this paper, the orders of the ARX model are fixed at $n_a = 5$ and $n_b = 5$ in all three network configurations. This is for comparing the model accuracy in each network configuration. The delay from the input to the output, n_d , is determined from the average round-trip time. This is because the packet sending rate at a specific time would have influence on the packet receiving process after the round-trip time. By letting N be the total number of round-trip time samples, n_d is determined as

$$n_d = \left\lfloor \frac{\sum_{k=1}^N y(k)}{N T} \right\rfloor \quad (1)$$

For evaluating the model accuracy, we use two validation methods: (1) a validation method using simulation, and (2) a validation method in frequency domain. The first

method is to compare the simulated output from the ARX model, where zero noise is assumed (i.e., $e(k) = 0$), with the actual output [6]. The simulated output from the ARX model is defined as

$$\begin{aligned} y^*(k|\theta) &= \psi^{*T}(k|\theta) \theta \\ \psi^*(k|\theta) &= [-y^*(k-1|\theta), \dots, -y^*(k-n_a|\theta), \\ &\quad u(k-n_d-1), \dots, u(k-n_d-n_b)] \end{aligned}$$

The ARX model is thought to be accurate if the simulated output from the ARX model coincides to the actual output.

The second method is to compare the frequency response of the ARX model with the frequency response estimated by the spectral analysis [6]. The bode plot is used for visually comparing those frequency responses. The bode plot illustrates the gain and the phase of a dynamic system at different frequencies. When a linear stable dynamic system has a sinusoid input signal

$$u(t) = A \sin(\omega t),$$

the output from the system can be written as

$$y(k) = B \sin(\omega + \phi)$$

The gain and the phase of the system at the frequency ω are B/A and ϕ , respectively. The frequency response of the ARX model is easily obtained by deriving its corresponding transfer function. On the contrary, the spectral analysis directly estimates the frequency response from measured input and output data. The ARX model is thought to be accurate if frequency responses of the ARX model and the spectral analysis are identical.

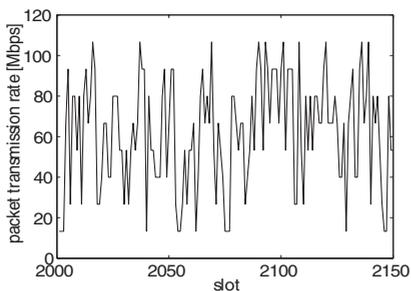
From all input and output data obtained in Section 3, two datasets for determining parameters of the ARX model (i.e., model identification) and for validating the model accuracy (i.e., model validation) are extracted. The number of input and output data used for system identification directly affects the model accuracy. We use 150 input and output data for both model identification and model validation; i.e., we use input and output data from 2,001 to 2,150 for model identification, and from 2,201 to 2,350 for model validation.

4.2 Modeling Results and Discussions

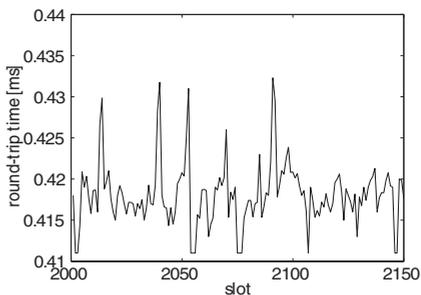
- Network **N1** (LAN)

Figure 7 shows (a) the input (the packet transmission rate from the source host), (b) the output (the average round-trip time), (c) comparison of the simulated output and the actual output, and (d) comparison of frequency responses of the ARX model and the spectral analysis for the network **N1**. Note that (a) and (b) are input and output data not for model validation, but for model identification. In this case, the average packet transmission rate from the source host was 66.6 Mbps, and the average round-trip time was 0.42 ms. The average throughput of the FTP transfer was 5.2 Mbps. Also the sampling interval is $T = 0.9$ ms, and the delay of the ARX model is $n_d = 0$ according to Eq. (1).

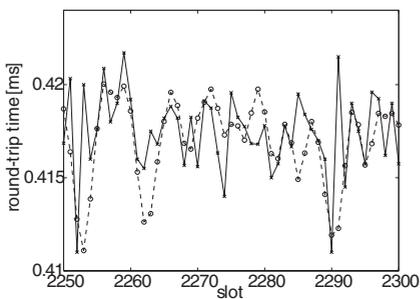
Figure 7(c) shows good agreement between the simulated output and the actual output. Figure 7(d) also shows good agreement between frequency responses of the



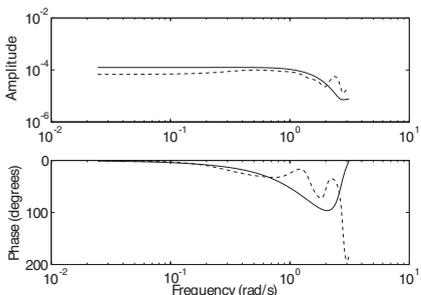
(a) Input data (packet transmission rate)



(b) Output data (average round-trip time)



(c) Comparison with simulation (solid: measured output, dotted: simulated output)



(d) Comparison in frequency domain (solid: ARX model, dotted: spectrum analysis)

Fig. 7. Network N1 (LAN)

ARX model and the spectral analysis. Although frequency responses at a high frequency are different, such disagreement would be caused by inaccuracy of the spectral analysis, in particular, at a high frequency [6]. From these observations, we conclude that in the network N1, the round-trip time dynamics can be accurately modeled by the ARX model. One of possible explanations for this phenomenon is that in LAN environment, the packet transmission rate from the source host directly affects the packet waiting time at the bottleneck link, resulting in a strong correlation between the packet transmission rate and the round-trip time.

Recall that the average packet transmission rate in this experiment is rather high (i.e., 66.6 Mbps). Although results are not included here due to space limitation, the ARX model cannot capture the round-trip time dynamics when the average packet transmission rate was 20 Mbps. This phenomenon is possibly because of little packet waiting time at the bottleneck link.

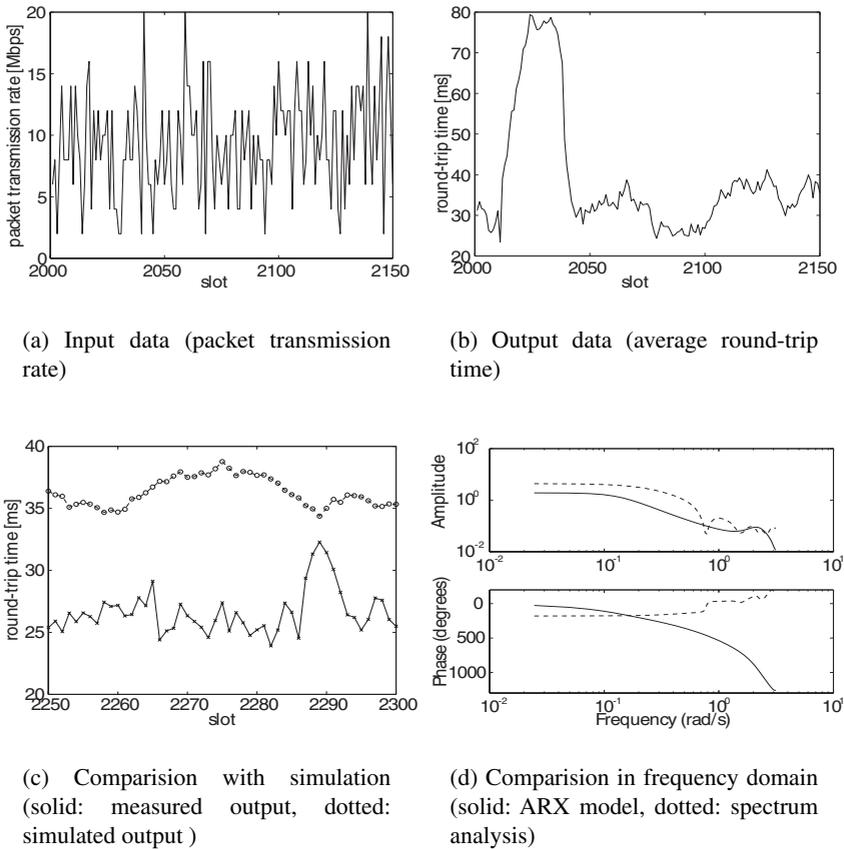


Fig. 8. Network **N3** (WAN with the non-bottlenecked access link)

- Network **N2** (WAN with the bottlenecked access link)

Figure 9 shows input and output data for model identification, and results of system identification for the network **N2**. In this experiment, the average packet transmission rate from the source host was 31.8 Kbps, and the average round-trip time was 319.7 ms. The sampling interval is $T = 125$ ms, and the delay of the ARX model is $n_d = 2$ according to Eq. (1). The network **N2** is WAN environment where there are 15 routers between source and destination hosts. Intuitively, it is expected that the modeling the round-trip time dynamics is more difficult than in the network **N1**. However, Fig. 9(c) indicates that the simulated output and the actual output well coincide. In addition, frequency responses in Fig. 9(d) shows good agreement between the ARX model and the spectral analysis. Hence, the round-trip time dynamics can be accurately modeled by the ARX model in the network **N2**. This phenomenon can be explained by the fact that the access link is the bottleneck between source and destination hosts. Namely, as the packet transmission rate

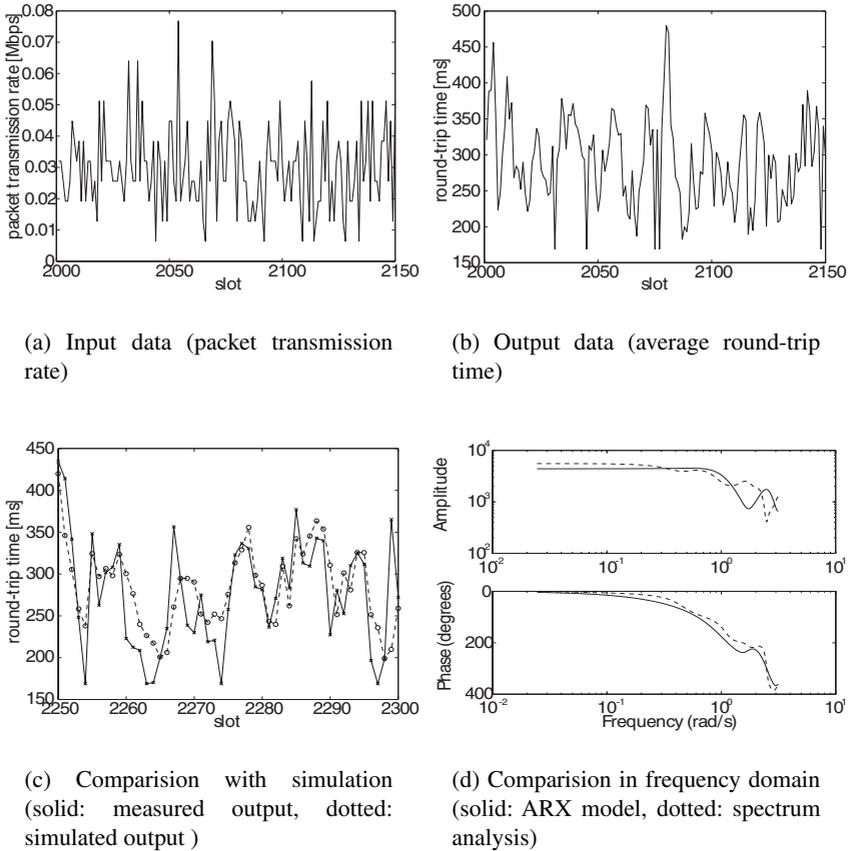


Fig. 9. Network N2 (WAN with the bottlenecked access link)

from the source changes, the packet waiting time also changes at the bottleneck link. In the network N2, since the access link is the bottleneck, the round-trip time suffers little disturbance from other traffic. Therefore, there exists a strong correlation between the packet transmission rate and the round-trip time, resulting in an accurate ARX model. From these observations and discussions, we conclude that in WAN environment, the round-trip time dynamics can be accurately modeled when the bottleneck link is shared by a small number of users.

- Network N3 (WAN with the non-bottlenecked access link)

Figure 8 shows input and output data for model identification, and results of system identification for the network N3, where the average packet transmission rate from the source host was 9.55 Mbps, and the average round-trip time was 36.89 ms. The sampling interval is $T = 6$ ms, and the delay of the ARX model is $n_d = 6$. Figure 8(c) shows that the simulated output and the actual output are completely different. Also, frequency

responses in Fig. 8(c) disagree. Thus, in the network **N3**, the ARX model fails to capture the round-trip time dynamics. This inaccuracy would be caused by a large noise; that is, in the network **N3**, the network topology is rather complex, and the bottleneck link would be shared by many users. Hence, the packet transmission rate from the source host has little impact on the round-trip time. In other words, there is a very weak correlation between the packet transmission rate and the round-trip time, resulting in an inaccurate ARX model. Although we have done several experiments for different destination hosts, the ARX model cannot capture the round-trip time dynamics in any case. From these observations, we conclude that the ARX model is unable to model the round-trip time dynamics when the network configuration is complex and the packet transmission rate has little effect on the round-trip time.

5 Conclusion

In this paper, we have modeled the round-trip time dynamics of the Internet by the ARX model using system identification. As input and output data for the ARX model, we have used the packet transmission rate from the source host and the average round-trip time measured by the source host. Using input and output data measured in working LAN and WAN environments, we have investigated how accurately the ARX model can capture the round-trip time dynamics. Through numerical examples, we have shown that in LAN environment, the round-trip time dynamics can be accurately modeled by the ARX model. We have also shown that in WAN environment, the round-trip time dynamics can be accurately modeled when the bottleneck link is shared by a small number of users.

As a future work, we are currently working to improve the model accuracy by using more complicated models such as ARMAX (Auto-Regressive Moving Average eXogenous) model. We are planning to design an efficient delay-based congestion control mechanism by utilizing the ARX model, which captures the round-trip time dynamics.

References

1. Internet Software Consortium, "Internet domain survey." available at <http://www.isc.org/ds/>.
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. H. Ohsaki, M. Murata, and H. Miyahara, "Modeling end-to-end packet delay dynamics of the Internet using system identification," in *Proceedings of Seventeenth International Teletraffic Congress*, pp. 1027–1038, Dec. 2001.
5. H. Ohsaki, M. Morita, and M. Murata, "On modeling round-trip time dynamics of the Internet using system identification," to be presented at *the 16th International Conference on Information Networking (ICOIN-16)*, Jan. 2002.
6. L. Ljung, *System identification — theory for the user*. Englewood Cliffs, N.J.: Prentice Hall, 1987.
7. S. Hares, "Essential tools for the OSI Internet," *Request for Comments (RFC) 1574*, Feb. 1994.

8. F. Baker, "Requirements for IP version 4 routers," *Request for Comments (RFC) 1812*, June 1995.
9. J. Postel, "Internet control message protocol," *Request for Comments (RFC) 792*, Sept. 1981.
10. S. Savage, "Sting: A TCP-based network measurement tool," in *USENIX Symposium on Internet Technologies and Systems*, pp. 71–79, Oct. 1999.