インターネットにおけるパケット伝送遅延時間の測定およびシステム同定によるモデル化に関する検討

森田 光茂† 大崎 博之‡ 村田 正幸‡

†大阪大学 大学院基礎工学研究科
〒 560-8531 大阪府豊中市待兼山町 1-3
Phone: 06-6850-6616, Fax: 06-6850-6589
E-mail: m-morita@ics.es.osaka-u.ac.jp

‡大阪大学 サイバーメディアセンター
〒 560-0043 大阪府豊中市待兼山町 1-30
Phone: 06-6879-8793, Fax: 06-6879-8794
E-mail: {oosaki,murata}@cmc.osaka-u.ac.jp

あらまし インターネットにおける、エンド–エンド間のパケット伝送遅延の特性を知ることは、アプリケーションの QoS (サービス品質) を向上し、効率的な輻輳制御を実現するためにきわめて重要である。本稿では、インターネットにおけるパケット伝送遅延時間を測定し、システム同定を用いてパケット伝送遅延時間の動的な特性をモデル化する。まず、システム同定で用いる入出力データとして、送信側ホストからのパケット送信間隔およびラウンドトリップ時間を測定する。往復パケット伝送遅延時間の測定には、ICMP (Internet Control Message Protocol) を利用する。次に、送信側ホストから見たネットワークをブラックボックスと考え、1 入力 1 出力の動的システムとしてモデル化する。システムへの入力として送信側ホストからのパケット送信間隔を、システムからの出力としてラウンドトリップ時間の差分を用いる。本稿では、ARX (Auto-Regressive eXogeneous) モデルに対してシステム同定を適用し、測定結果からモデルのパラメータを決定する。複数のネットワーク環境における測定結果を用いてモデル化を行い、ARX モデルによりラウンドトリップ時間がどの程度モデル化できるか検討する。

和文キーワード パケット伝送遅延時間、トラヒック測定、システム同定、ARX モデル

On Measuring the End-to-End Packet Delay and its Dynamics Modeling using System Identification

Mitsushige Morita† Hiroyuki Ohsaki‡ Masayuki Murata‡

†Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan
Phone: +81-6-6850-6616, Fax: +81-6-6850-6589
E-mail: m-morita@ics.es.osaka-u.ac.jp

‡Cybermedia Center, Osaka University
1-30 Machikaneyama, Toyonaka, Osaka 567-0043, Japan
Phone: +81-6-6879-8793, Fax: +81-6-6879-8794
E-mail: {oosaki, murata}@cmc.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 realtime services, and it enables us to design an efficient congestion control mechanism. In this paper, we measure the end-to-end packet delay of the Internet, and build a model representing its dynamics using the system identification. We first measure, as the input and output data for the system identification, the packet inter-departure time from a source host and the corresponding round-trip time measured by the source host. ICMP (Internet Control Message Protocol) is utilized to measure the round-trip time for each packet. We next model the network, seen by a specific source host, as a dynamic SISO (Single-Input and Single-Output) system. Namely, the input to the system is the packet inter-departure time from the source host, and the output from the system is the round-trip time variation between two adjacent packets. Using measurement results obtained from several network environments, we show how accurately the packet delay dynamics of the Internet can be modeled with the system identification.

key words End-to-End Packet Delay, Traffic Measurement, System Identification, ARX Model

1 はじめに

インターネットにおける、エンド–エンド間のパケット伝送遅延時間の特性を知ることは、アプリケーションの QoS (サービス品質) を向上し、効率的な輻輳制御を実現するためにきわめて重要である。例えば、TCP (Transmission Control Protocol) において、パケット棄却の有無によって輻輳を検出するのではなく、パケット伝送遅延時間の変動からネットワークの輻輳を検出する方式が提案されている [1]。このようなパケット伝送遅延時間を利用した輻輳制御では、パケット伝送遅延時間の増加から輻輳の発生を予備的に検知することにより、ネットワーク内でのパケット棄却を防ぐことが可能となる。

従来、回線交換ネットワークおよびパケット交換ネットワークの性能評価には、待ち行列理論が広く用いられてきた。一般に、待ち行列理論はネットワークの状態が定常であることを仮定し、平均パケット伝送遅延時間やパケット棄却率などの性能指標を導出することができる。しかし、パケット単位の細かなタイムスケールで考えた場合、インターネットは定常ではない [2, 3, 4]。このため、特にインターネットにおける TCP のような、フィードバックシステムにおいてパケットごとの動的な特性を解析するためには、待ち行列理論とは異なるアプローチが必要となる。例えば、送信側ホストからのパケット送出レートを動的に変更した場合、エンド– エンド間のパケット伝送遅延時間が将来どのように変化するかといった、フィードバックシステムにおけるパケット伝送遅延時間の動的な特性はこれまでほとんど明らかにされていない。

我々はこれまで、文献 [5] において、制御工学の分野で広く用いられているシステム同定 [6] を用いることにより、エンド–エンド間のパケット伝送遅延時間の特性をモデル化する方法を提案した。ここでは、送信側ホストから見たネットワークを 1 入力 1 出力の動的システムと考え、パケット伝送遅延時間の特性を ARX (Auto-Regressive eXogenous) モデルでモデル化している。システムへの入力として送信側ホストからのパケット送信間隔を、システムからの出力としてパケット伝送遅延時間の差分を用いている。文献 [5] では、シミュレーションで得られた入出力データを用いて、パケット伝送遅延時間の特性をモデル化した。そこで本稿では、パケット送信間隔およびパケット伝送遅延時間を実際のネットワーク上で測定し、これを用いてパケット伝送遅延時間特性のモデル化を行う。

本稿では、まず、システム同定で用いる入出力データとして、送信側ホストからのパケット送信間隔およびラウンドトリップ時間 (往復パケット伝送遅延時間) を測定する。なお、文献 [5] では、送信側ホストおよび受信側ホストから見たネットワークをブラックボックスとしてモデル化しているが、本稿では送信側ホストから見たネットワークをブラックボックスとしてモデル化する。このため、エンド–エンド間のパケット伝送遅延時間ではなく、ラウンドトリップ時間 (往復パケット伝送遅延時間) がシステムからの出力となる。ラウンドトリップ時間の測定には、ICMP (Internet Control Message Protocol) を利用する。現在、ほとんどのネットワーク機器が ICMP ECHO パケットに応答するため、さまざまな送信側ホストに対してラウンドトリップ時間を測定することが可能となる。本稿では、複数のネットワーク環境 (有線 LAN および無線 LAN) においてシステム同定で用いる入出力データを測定する。次に、ARX モデルに対してシステム同定を適用し、測定結果からモデルのパラメータを決定する。測定結果を用いてモデル化を行い、ラウンドトリップ時間の動的な特性をどの程度適切にモデル化できるかを明らかにする。

本稿の構成は以下の通りである。まず、2 章では送信側ホストから見たネットワークをどのようにブラックボックスとしてモデル化するかを説明する。また、ARX モデルの概要についてもここで説明する。3 章では、ARX モデルを用いたシステム同定の概要を説明する。4 章では、システム同定に用いる入出力データの測定方法を説明する。さらに、5 章では複数のネットワーク環境における測定結果を用いてモデル化を行い、システム同定によりラウンドトリップ時間がどの程度モデル化できるかを明らかにする。最後に 6 章において、本稿のまとめと今後の課題を述べる。

2 パケット伝送遅延時間のモデル化

2.1 モデルへの入出力

本稿では、送信側ホストから見たネットワーク全体をブラックボックスととらえ、システム同定を用いてパケット伝送遅延時間の動的な特性をモデル化する。文献 [5] では、送信側ホストおよび受信側ホストから見たネットワークをブラックボックスとしてモデル化している。本稿では、送信側ホストから見たネットワークを、1 入力 1 出力の動的システムとしてモデル化する (図 1)。システム同定では、観測されたシステムへの入力と出力のみにもとづいてモデルを作成する。このため、システム同定を用いてモデル化する場合、システムへの入力と出力を適切に選択する必要がある。システム同定では、

1. 同定に用いる入力および出力の間に相関があること
2. 同定対象となるシステムがフィードバック系でないこと
3. システムに対するノイズの平均が 0 で、なおかつ定常であること

が望ましい [6]。本稿のように、パケット伝送遅延時間を出力とするモデルを作成する場合には、

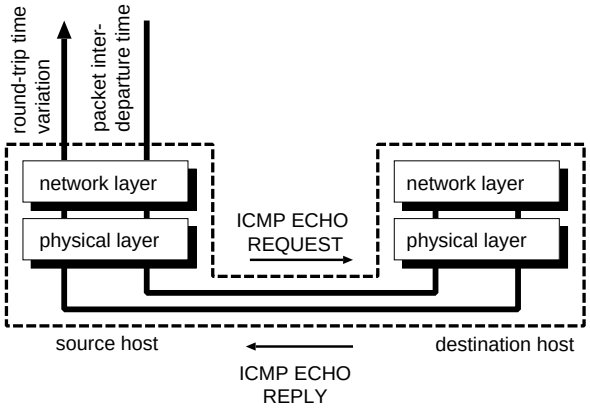


図 1: パケット伝送遅延時間のブラックボックスモデル

1. システムへの入力はパケット伝送遅延時間に影響を与える
2. システムへの入力は過去のパケット伝送遅延時間とは独立であること
3. ネットワーク内部で発生するノイズの平均が 0 で、なおかつ定常であること

と、読みかえることができる。

パケット伝送遅延時間はネットワークの輻輳状況に応じて変化する [3] ため、1. よりシステムへの入力は送信側ホストからのパケット送信レートが自然である。一般に、送信側ホストからのパケット送信レートが大きい時、ネットワークが輻輳するためパケット伝送遅延時間が大きくなる。一方、送信側ホストからのパケット送信レートが小さい時は、ネットワークがより輻輳しにくくなるため、パケット伝送遅延時間が比較的小さくなる。ただし、パケット単位といったタイムスケールにおいて、パケット伝送遅延時間がどのように変動するかをモデル化するためには、パケット送信レートといった平均的な値を使うことはできない。そこで本稿では、各パケットの送信間隔をシステムへの入力として用いる。

2. は、システム同定がシステムの入出力に相関がないことを利用しているために生じる制約である。フィードバック系に対してもシステム同定を適用することは可能であるが、実際のシステムへの適用は容易ではない。インターネットにおいては、TCP においてフィードバック型の輻輳制御が行われている。このため、システム同定に用いる入力としては、TCP のパケット送信間隔は適切ではない。

3. は、システムからの出力をどのように決定するかに関連する。例えば、システムからの出力を、送信側ホストで観測したラウンドトリップ時間とした場合を考える。この時、送信側ホストで観測するラウンドトリップ時間は、ネットワーク内部で他のトラヒックの影響を受ける。システム同定では、このような他の要因をすべてノイズとしてモデル化する。他のトラヒックがパケット伝送遅延時間に与える影響は、ルータのバッファ内パケット数の変動に与える影響と考えることができる。しかし、このような他のトラヒックの影響は定常ではないため、そのままではシステムからの出力として使用できない。そこで本稿では、文献 [5] と同じように、連続するラウンドトリップ時間の差分を、システムからの出力として使用する (図 2)。

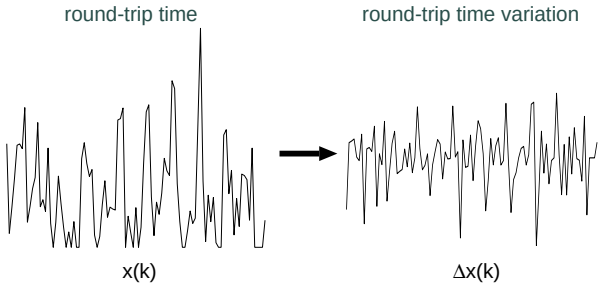


図 2: ラウンドトリップ時間の差分

2.2 ARX (Auto-Regressive eXogenous) モデル

本稿では、ARX モデルを用いて、パケット伝送遅延時間の特性をモデル化する (図 3)。前節で述べたように、ARX モデルへの入力と出力は、それぞれ送信側ホストからのパケット送信間隔と、送信側ホストで観測されたラウンドトリップ時間の差分である。また、他のトラヒックがパケット伝送遅延時間に与える影響を、ARX モデルに加わるノイズとしてモデル化する。

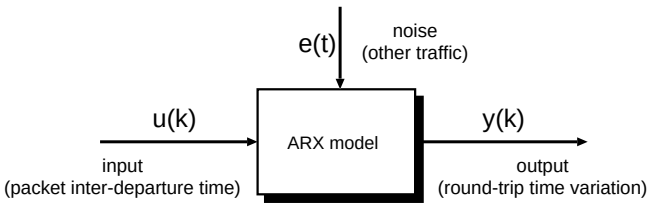


図 3: ARX モデル

一般に、$u(k)$ および $y(k)$ を、それぞれ k 番目のスロットのおける入力および出力とすれば、ARX モデルは以下のように定義される [6]。

$$A(q)y(k) = B(q)u(k-n_d) + e(k) \quad (1)$$

$$A(q) = 1 + a_1 q^{-1} + \ldots + a_{n_a} q^{-n_a} \quad (2)$$

$$B(q) = b_1 + b_2 q^{-1} + \ldots + b_{n_b} q^{-n_b+1} \quad (3)$$

ここで、$e(k)$ は測定不可能な雑音であり、q^{-1} はシフトオペレータ、すなわち、

$$q^{-1}u(k) \equiv u(k-1) \quad (4)$$

また、n_a と n_b はそれぞれ多項式の次数であり、n_d は入力と出力の間の遅れである。a_i および b_j ($1 \leq i \leq$

$n_a, 1 \leq j \leq n_b$) は ARX モデルのパラメータである。本稿では、$u(k)$ は送信側ホストから送信された k 番目のパケットと、$k+1$ 番目のパケットの送信間隔を意味する。また、$y(k)$ は、k 番目のパケットのラウンドトリップ時間と $k+1$ 番目のパケットのラウンドトリップ時間との差となる。

ARX モデルは、線型な時不変のモデルである。このため、パケット伝送遅延時間の動的特性に含まれる非線型性をモデル化することはできない。しかし、たとえ非線型なシステムであっても、平衡点の近傍であれば線型システムで十分に近似することが可能である。また、ARX モデルは構造が比較的単純であるため、次章で述べるように、ARX モデルのパラメータを容易に決定できるという利点がある。

3 システム同定

3.1 最小 2 乗法によるパラメータ決定

システム同定を用いて、パケット伝送遅延時間の動的な特性をモデル化するという問題は、入出力データ $u(k)$ および $y(k)$ を用いて、ARX モデルのパラメータ a_i および b_j を決定するという問題に帰着する [6]。以下では、a_i および b_j を

$$\theta = [a_1, \ldots, a_{n_a}, b_1, \ldots, b_{n_b}] \tag{5}$$

のように表記する。この時、ARX モデルの 1 段先予測は以下のように定義される。

$$\hat{y}(k|\theta) \equiv \psi(k)\,\theta \tag{6}$$

ただし、

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

である。実際の出力と ARX モデルの 1 段先予測との差を予測誤差と呼び、

$$\varepsilon(k|\theta) \equiv y(k) - \hat{y}(k|\theta) \tag{7}$$

のように定義される。システム同定では、この予測誤差が小さくなるような ARX モデルのパラメータ θ を決定する。特に、損失関数

$$J_N(\theta) \equiv \frac{\sum_{k=1}^{N} \varepsilon(k|\theta)^2}{N} \tag{8}$$

が最小となるように θ を決定する最小 2 乗法が広く用いられている。ここで N は、システム同定に用いた入出力データの数 (サンプル数) である。最小 2 乗法では、損失関数 $J_N(\theta)$ が θ に関する 2 次式になるため、損失関数を最小化する θ を簡単に求めることができる。ただし、最小 2 乗法は、入出力データに含まれる異常値に非常に敏感であるため、入出力データに異常値が含まれていると正しいモデル化ができないといった問題がある。そのため、実際に同定を行なう前に、入出力データから異常値を取り除いておく必要がある。

3.2 逐次最小 2 乗法によるオンライン同定

最小 2 乗法を用いたパラメータ推定法には、入出力データの測定後にシステム同定を行うのではなく、入出力データを測定しながら逐次的にシステム同定が可能な逐次最小 2 乗法がある [6]。逐次最小 2 乗法では、以下のようにして逐次的に ARX モデルのパラメータを推定する。

初期値:

$$\begin{aligned}\hat{\theta}(0) &= 0 \\ \mathbf{P}(0) &= \gamma \mathbf{I} \ (\gamma > 0、\mathbf{I} \text{は単位行列})\end{aligned}$$

入出力データ測定ごと:

$$\begin{aligned}\hat{\theta}(k) &= \hat{\theta}(k-1) + \mathbf{P}(k)\psi(k)[y(k) - \psi^T(k)\hat{\theta}(k-1)] \\ \mathbf{P}(k) &= P(k-1) - \frac{\mathbf{P}(k-1)\psi(k)\psi^T(k)\mathbf{P}(k-1)}{1 + \psi^T(k)\mathbf{P}(k-1)\psi(k)}\end{aligned}$$

最小 2 乗法は逆行列の計算を必要とするが、逐次最小 2 乗法では逆行列の計算も不要となり、計算が非常に簡単になるという利点がある。ただし、逐次最小 2 乗法では、ARX モデルのパラメータ θ が収束するまで計算を行う必要があるため、通常の最小 2 乗法に比べてより多くの入出力データを必要とする。

4 ICMP による入出力データ測定

システム同定に用いる入出力データを測定するためには、実際にネットワークにパケットを送信し、それらのパケットのラウンドトリップ時間を測定する必要がある。測定のために送信するパケットとして、

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

などが考えられる。

システム同定に用いる入力としては、さまざまな周波数成分が含まれている白色雑音が理想的である。しかし、TCP ではフィードバック型の輻輳制御が行われているため、パケット送信間隔を自由に変更することはできない。また、2 章で述べたように、フィードバック型輻輳制御のために、システム同定に用いる入力としては適切ではない。なお、TCP は双方向の通信であり、パケットが到着したことを確認するための ACK (ACKnowledgment) パケットが、受信側ホストから送信側ホストに向けて返送される。これを利用すれば、ラウンドトリップ時間の測定は可能である。

一方、UDP では輻輳制御が行なわれないため、送信側ホストからのパケット送信間隔を自由に指定することが

可能である。ただし、UDP は片方向の通信であるため、ラウンドトリップ時間を測定するためには、受信側ホストで何らかの処理を行う必要がある。traceroute プログラムで行われているように、ICMP ポート到達不能パケットを利用して、送信側ホストだけでラウンドトリップ時間を測定することは可能である。これは、受信側ホストの未使用ポートに UDP パケットを送信すると、受信側ホストは ICMP ポート到達不能パケットを送信側ホストに返送するからである。しかし、ICMP ポート到達不能パケットの数は制限されているため [7]、システム同定の入出力データ測定には使用できない。

ICMP は、ネットワークの障害通知などに用いられる、制御用のパケットを通信するためのプロトコルである [8]。ICMP は UDP と同じく輻輳制御が行われないため、送信側ホストからのパケット送信間隔を自由に指定することが可能である。また、ping プログラムで行われているように、ICMP ECHO パケットと ICMP ECHO REPLY パケットを利用して、送信側ホストにおいてラウンドトリップ時間を測定することが可能である。ほとんどのネットワーク機器が ICMP ECHO パケットに応答するため、ICMP を用いればほとんどの送信側ホストに対してラウンドトリップ時間を測定することができる。

本稿では、以上のような理由から、ICMP パケットを利用して入出力データの測定を行った。ping プログラムを一部改造することにより、送信側ホストにおいてパケット送信間隔およびパケット伝送遅延時間を測定した。送信側ホストの動作アルゴリズムの概要は以下の通りである。

ICMP パケット送信部のアルゴリズム:

S1) 1,500 バイトの ICMP ECHO メッセージを送信する。ICMP パケットのデータ部分に現在時刻 (送信時刻) を記録しておく。

S2) パケット送信間隔を指数分布によって決定し、次の ICMP パケット送信をスケジューリングする。

S3) S1 に戻る。

ICMP パケット受信部アルゴリズム:

R1) ICMP ECHO REPLY パケットの到着を待つ。

R2) ICMP パケットのデータ部分に記録された送信時刻を取り出す。

R3) 現在時刻から ICMP パケットのラウンドトリップ時間を計算する。

R4) 直前の ICMP パケットの送信時刻から、パケット送信間隔を計算する。

R5) R1 に戻る。

受信側ホストから返送される ICMP ECHO REQUEST パケットには、送信した ICMP ECHO パケットのデータ部分がそのままコピーされる。そこで、これを利用してパケット送信間隔およびパケット伝送遅延時間を測定している。

5 測定した入出力データによるモデル化

5.1 測定したネットワーク環境

本稿では、以下のような異なる 3 種類のネットワーク環境において、システム同定で用いる入出力データを測定した。なお、送信側ホストおよび受信側ホストともに、Linux オペレーティング・システムを使用した。

N1) 有線 LAN 100Mbit/s (他のトラヒックなし)

N2) 有線 LAN 100Mbit/s (他のトラヒックあり)

N3) 無線 LAN 11Mbit/s + 有線 LAN 100Mbit/s

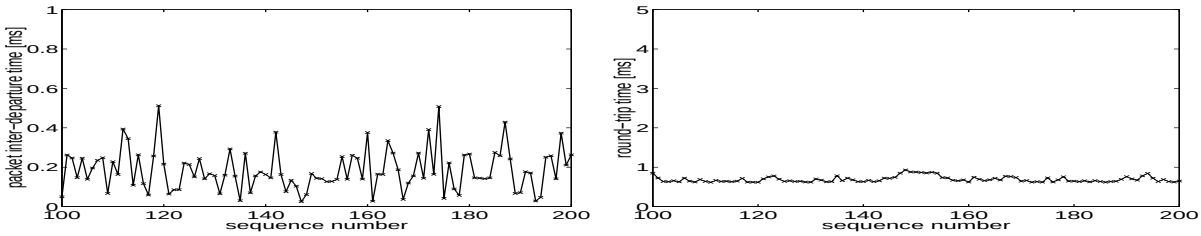
ネットワーク N1 では、送信側ホストと受信側ホストが単一のスイッチングハブで接続された、100Mbit/s の LAN 環境である。単一のスイッチングハブによって接続しているため、送信側ホストで測定するラウンドトリップ時間は、他のトラヒックの影響を受けない。このため、他のトラヒックの影響というノイズが存在しない場合に、どの程度ラウンドトリップ時間の動的な特性がモデル化できるかを調べることができる。

ネットワーク N2 では、送信側ホストと受信側ホストの間に 4 段のスイッチングハブが接続された、100Mbit/s の LAN 環境である。途中のスイッチングハブには他のトラヒックが流れているため、受信側ホストで測定するラウンドトリップ時間は、これらのトラヒックの影響によって変動する。これにより、他のトラヒックがどの程度ノイズとしてモデル化できるかを調べることができる。

ネットワーク N3 では、送信側ホストは 11Mbit/s の無線 LAN によって基地局に接続されている。そこから 3 段のスイッチングハブを経由して、100Mbit/s の有線 LAN によって受信側ホストに接続されている。この場合、送信側ホストと受信側ホストの間に非常に低速なネットワークが存在するため、ラウンドトリップ時間時間がネットワーク N1 やネットワーク N2 と比べて非常に大きくなる。

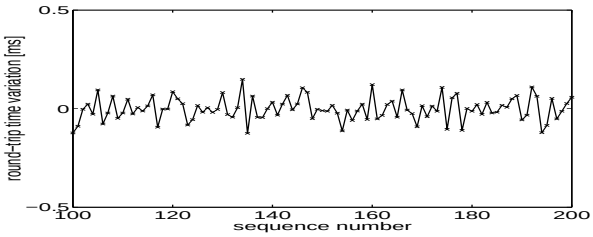
5.2 ネットワーク N1

図 4 に、ネットワーク N1 における入出力データの測定結果およびシステム同定の結果を示す。左上から順に、(a) ICMP パケットの送信間隔、(b) ラウンドトリップ時間、(c) ラウンドトリップ時間の差分、(d) 実際の出力と ARX モデルの出力との比較を示している。この時の平均パケット送信レートは 59.6 Mbit/s、平均ラウンドトリップ時間は 1.4 ms であった。測定した入出力データのうち、100 サンプルを用いて、最小 2 乗法によりシステム同定を行った。なお、ARX モデルの次数および遅れを $n_a = 5$、$n_b = 5$、$n_d = 1$ とした。図 4(d) において ARX モデルの出力とは、システム同定によって得られた ARX モデルに、実際に入力データを与えて出力をシミュレートした結果である。ARX モデルに加わるノイズを 0 と

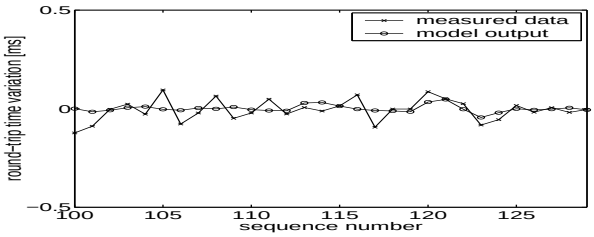


(a) パケット送信間隔

(b) ラウンドトリップ時間



(c) ラウンドトリップ時間の差分



(d) 実際の出力とモデル出力との比較

図 4: ネットワーク N1 (有線 LAN) での結果

仮定して、ARX モデルの出力を計算している。ノイズの特性が不明であるため、たとえラウンドトリップ時間の変動を正確にあらわす ARX モデルが得られたとしても、実際の出力と ARX モデルの出力が完全に一致することはない。

図 4 より、パケット送信間隔が変動しているにもかかわらず、ラウンドトリップ時間が 0.2 ms 程度しか変動していないことが分かる。これは、ネットワーク N1 では単一のスイッチングハブを使用しているため、他のトラヒックの影響をまったく受けないためである。ラウンドトリップ時間のわずかな変動は、送信側ホストおよび受信側ホストにおける処理遅延の変動、もしくは測定に用いた Linux オペレーティング・システムのタイマの精度によるものと考えられる。図 4(d) より、ラウンドトリップ時間の変動がうまくモデル化できていないことがわかる。これは入力と出力の相関がほとんどないためと考えられる。

5.3 ネットワーク N2

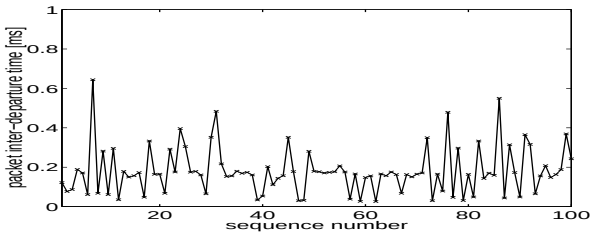
ネットワーク N2 における測定結果および同定結果を図 5 に示す。この時の平均パケット送信レートは 55.7 Mbit/s、平均ラウンドトリップ時間は 4.2 ms であった。図 5 と同じように、測定した入出力データのうち 100 サンプルを用いてシステム同定を行った。ARX モデルの次数および遅れは $n_a = 5$、$n_b = 5$、$n_d = 1$ である。図 5(c) より、ネットワーク N1 の場合と比べて、ラウンドトリップ時間が大きく変動していることがわかる。これは、他のトラヒックの影響によってネットワークが輻輳しているためと考えられる。図 5(c) より、ARX モデルによって、ラウンドトリップ時間の変動の傾向が、おおよそモデル化できていることがわかる。しかし、他のトラヒックによる影響 (ノイズ) をと仮定しているため、実際の出力と ARX モデルの出力が大きくずれる場合がある。これは、他のトラヒックがラウンドトリップ時間の変動に与える影響が、送信側ホストからのパケット送信間隔がラウンドトリップ時間の変動に与える影響と同じ程度に大きいことを示している。

5.4 ネットワーク N3

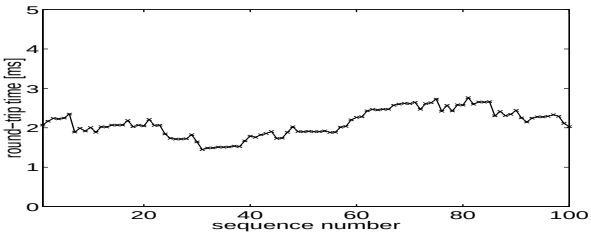
ネットワーク N3 における測定結果および同定結果を図 6 に示す。この時の平均パケット送信レートは 2.12 Mbit/s、平均ラウンドトリップ時間は 633.2 ms であった。同定に用いた入出力データ数や ARX モデルの次数は図 4 と同じである。図 6(c) より、ネットワーク N1 やネットワーク N2 と比べて、ラウンドトリップ時間が大きく変動している (最大 15 ms 程度) ことがわかる。図 5(d) より、ネットワーク N2 と同じ程度に、ラウンドトリップ時間の変動の傾向がモデル化できていることがわかる。しかし、ここでも他のトラヒックによる影響が大きいため、ラウンドトリップ時間の急激な変動はモデル化できていない。

5.5 モデルの次数およびサンプル数の影響

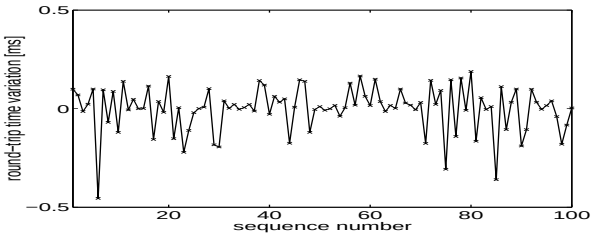
これまで、測定した入出力データのうち 100 サンプルを使用し、ARX モデルの次数および遅れを $n_a = 5$、$n_b = 5$、$n_d = 1$ としてモデル化を行った。しかし、シ



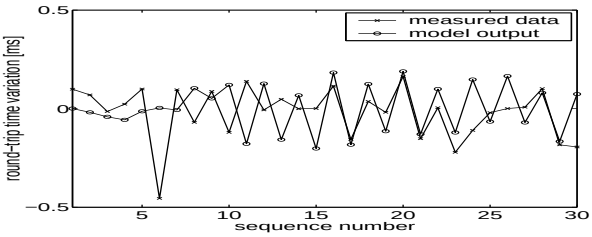
(a) パケット送信間隔



(b) ラウンドトリップ時間



(c) ラウンドトリップ時間の差分



(d) 実際の出力とモデル出力との比較

図 5: ネットワーク N2 (有線 LAN) での結果

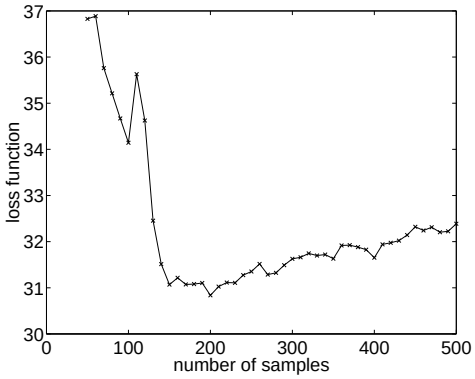


図 7: 損失関数の最小値とサンプル数の関係

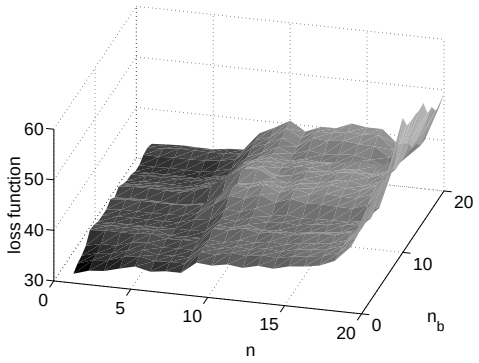


図 8: 損失関数の最小値と ARX モデルの次数の関係

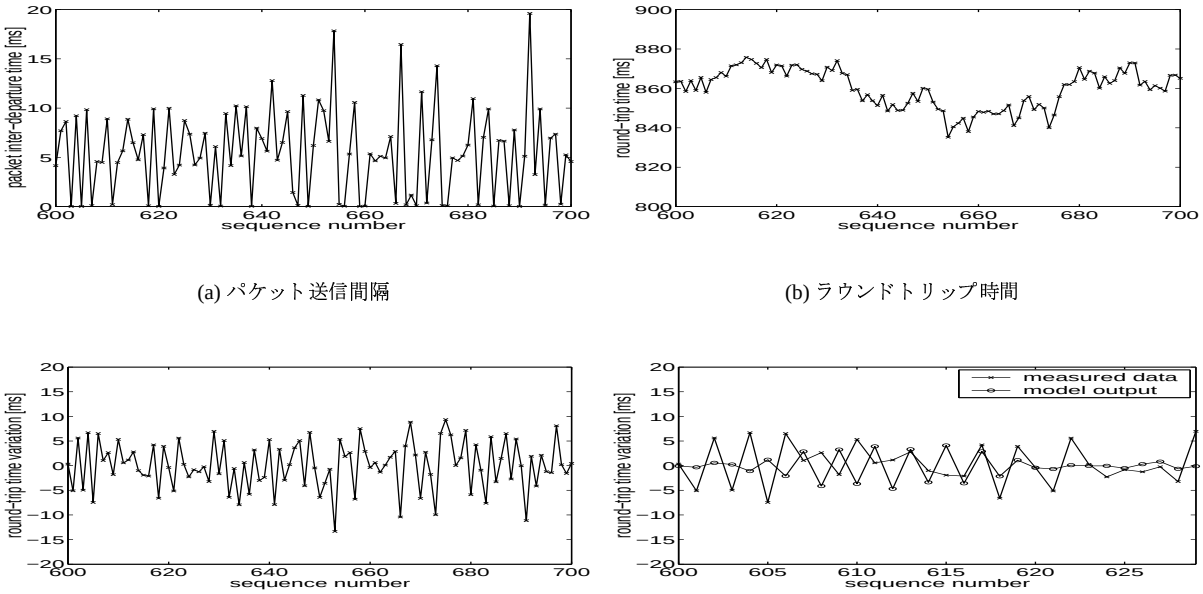
ステム同定に使用するサンプル数や、ARX モデルの次数および遅れをどのように選択するかによって、得られるモデルの精度が変化する。このため、実際には損失関数の値が最小となるように、サンプル数などを適切に選択する必要がある。

そこで図 7 に、ネットワーク N3 における入出力データを使用した場合の、損失関数 $J_N(\theta)$ とサンプル数との関係を示す。ARX モデルの次数および遅れを $n_a = 5$、$n_b = 5$、$n_d = 1$ と固定し、システム同定に使用するサンプル数を変化させた時の損失関数の値を示している。この図から、サンプル数が多くなれば、これにともない損失関数も減少する傾向があることがわかる。結果は省略するが、ネットワーク N1 およびネットワーク N2 で測定した入出力データについても同様の結果が得られた。

また図 8 に、ARX モデルの次数 n_a、n_b と損失関数の最小値との関係を示す。ここでは、システム同定に用いるサンプル数を 100 と固定し、ARX モデルの次数 n_a、n_b をそれぞれ 1 から 20 まで変化させている。この図から、ARX モデルの次数 n_a を大きくするにつれ、損失関数の値が大きくなっていることがわかる。一方、ARX モデルの次数 n_b は損失関数の値に大きな影響を与えないこともわかる。一般に、モデルの次数が大きくなるにつれモデルの精度は向上するが、システム同定に要する計算量が増え、またモデルが不安定になりやすくなる。このため実際には、モデルの精度と計算量というトレードオフを考慮して次数を決定する必要がある。

6 まとめと今後の課題

本稿では、送信側ホストから見たネットワークをブラックボックスと考え、パケット伝送遅延時間の動的な特性



(a) パケット送信間隔

(b) ラウンドトリップ時間

(c) ラウンドトリップ時間の差分

(d) 実際の出力とモデル出力との比較

図 6: ネットワーク N3 (無線 LAN + 有線 LAN) での結果

を、システム同定によりどのようにモデル化できるかを検討した。送信側ホストから見たネットワークを、1 入力 1 出力の ARX モデルと考え、異なるネットワーク環境下で測定した入出力データを用いてモデルを作成した。その結果、システム同定を用いることにより、有線 LAN および 無線 LAN の両方において、ラウンドトリップ時間の変動をある程度モデル化できることがわかった。しかし、ラウンドトリップ時間は他のトラヒックの影響によって大きく変動するため、ラウンドトリップ時間の急激な変動まではモデル化できないことがわかった。

本稿では、有線および無線 LAN 環境において、パケット送信間隔およびパケット伝送遅延時間を測定した。今後は、WAN 環境において測定した入出力データにより、ラウンドトリップ時間の変動がどの程度モデル化できるかを検討する必要がある。ただし、パケット伝送遅延時間をより正確にモデル化するためには、大量の ICMP パケットを送出することによって、ネットワークをある程度輻輳させる必要がある。このため、小量のパケットにより入出力データを測定する方法を検討する必要がある。

謝辞

本研究の一部は、日本学術振興会未来開拓学術研究推進事業における研究プロジェクト「高度マルチメディア応用システム構築のための先進的ネットワークアーキテクチャの研究」(JSPS-RFTF97R16301) によっている。ここに記して謝意を表す。

参考文献

[1] 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, October 1994.

[2] 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.

[3] 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.

[4] V. Paxson, "End-to-end Internet packet dynamics," in *Proceedings of ACM SIGCOMM*, pp. 139–152, Sept. 1997.

[5] H. Ohsaki, M. Murata, and H. Miyahara, "Modeling end-to-end packet delay dynamics of the Internet using system identification," submitted to *International Teletraffic Congress 17*, Jan. 2001.

[6] L.Ljung, *System Identification – theory for the user*. Englewood Cliffs,N.J.: Prentice Hall, 1987.

[7] F. Baker, "Requirements for IP version 4 routers," *Request for Comments (RFC) 1812*, June 1995.

[8] J. Postel, "Internet control message protocol," *Request for Comments (RFC) 792*, Sept. 1981.