

ボトルネックリンク速度推定ツールの提案と精度検証

八木 敬宏[†] 塩田 茂雄[†] 間瀬 憲一^{††}

[†]千葉大学工学部 都市環境システム学科 〒263-8522 千葉市稲毛区弥生町 1-33

^{††}新潟大学工学部情報工学科 〒950-2181 新潟市五十嵐 2 の町 8050

E-mail: [†]{yagi,shioda}@hike.tu.chiba-u.ac.jp, ^{††}mase@ie.niigata-u.ac.jp

あらまし 本論文では、試験パケットを用いて、エンドホスト間を結ぶ経路上のボトルネックリンクの存在箇所とその物理速度を推定するツール (NetTracer) を提案する。提案ツールは、Jacobson が考案したリンク速度推定ツール “Pathchar” の動作特徴と類似しているが、スイッチングハブのようにレイヤ 2 でパケットをストア&フォワード処理するノードがリンク上に存在しても、正確な推定が可能であるという特長を有する。本論文では提案ツールの基本的な特徴、動作概要、実装方法を既存のツールと対比させて述べるとともに、提案ツールを利用したネットワーク実験結果を通して、提案ツールの有効性を示す。

キーワード インターネット、帯域測定、ボトルネック、Pathchar、QoS、ネットワークトモグラフィ

A new approach for measuring the bottleneck link bandwidth using probe packets

Takahiro YAGI[†], Shigeo SHIODA[†], and Kenichi MASE^{††}

[†]Urban Environment Systems, Faculty of Engineering, Chiba University
1-33 Yayoi, Inage, Chiba 263-8522, Japan

^{††}Department of Information Engineering, Faculty of Engineering, Niigata University
2-8050, Ikarashi, Niigata 950-2181, Japan

E-mail: [†]{yagi,shioda}@hike.tu.chiba-u.ac.jp, ^{††}mase@ie.niigata-u.ac.jp

Abstract This paper proposes a tool, called *NetTracer*, for finding the location and for measuring the bandwidth of the bottleneck link along a path through the Internet. Although the techniques and implementation of *NetTracer* are similar to those of *Pathchar*, *NetTracer* does not suffer from the *invisible node* problem; that is, unlike *Pathchar*, it could accurately measure the bandwidth of the bottleneck link even when store-and-forward *invisible* nodes, which do not decrement the IP TTL, exist along a path. Preliminary experiments conducted on a network at Chiba university indicate that, in terms of accuracy and robustness, *NetTracer* is superior to currently known tools including *Pathchar*.

Key words Internet, bandwidth measurement, bottleneck link, *Pathchar*, QoS, network tomography

1. ま え が き

インターネットは急速に普及し、いまや我々の社会活動に欠かせない重要なインフラストラクチャの一つとして成長した。インターネットのトラフィックも急増し、テキストのみならず、静止画像、動画、音声などの様々な形態の情報がインターネット上で配信されているが、それに伴い、インターネットの高品質化が次なる技術課題として取り組まれている。例えば、音声/動画を（特に双方向で）リアルタイム配信するためには、インターネット上で音声/動画を配信する際の遅延を許容値（例えば数百ミリ秒）以下に抑える仕組みが求められる。従って、許容値を超えて遅延が発生している場合、その原因を速やかに

調査し、原因を除去する、いわゆる「トラブルシューティング」技術が必要である。このような理由から、外部ホストからインターネット内部の Quality-of-Service (QoS) やトラフィック交流を遠隔推定する「ネットワークトモグラフィ」技術 [11] が近年注目されている。

インターネットに接続された 1 台のホストから送信されたパケットが、インターネットを経由して別のホストに届くのに要する時間は、(1) 伝播遅延、(2) 各ルータでの（アドレス検索等に伴う）処理遅延、(3) 各ルータでのバッファ待ち遅延、(4) 各ルータでのパケット転送遅延の 4 つの遅延時間の和に等しい。一般には、(3) と (4) が支配的な遅延要因となるが、いずれもルータ間を接続するリンク（もしくは LAN のようなサブネット

ワーク)の物理的な速度に大きく依存する。従って、インターネットの両端に位置するホスト間の経路上の各リンクの物理速度、特にボトルネックリンクの物理速度を知ることは、ホスト間の end-to-end 遅延の発生要因を知る上で重要である。

インターネットに接続されたホストから、インターネット内の各リンクの物理速度を遠隔測定するツールとして Pathchar [3] が知られている。Pathchar はルータやホストに特別な機能を追加することなく、(Pathchar をインストールしたホストを端点とする全ての経路上の)リンクの物理速度を測定できる優れたツールであるが、2章で述べるように、幾つかの問題点を抱えている。特に、スイッチングハブのようなデータリンク層でパケットをストア&フォワード処理するノード (invisible ノードと呼ばれる [5]) がリンク上に存在する場合、そのリンクの物理速度を実際よりも大幅に小さく見積もる傾向がある [10]。

本論文では、Pathchar の長所を有し、かつ invisible ノードの存在にもかかわらず、正確にボトルネックリンクの存在場所とその物理速度を測定できるツールを新たに提案するとともに、幾つかのネットワーク実験により、その有効性を示す。

以下、本論文の構成を述べる。まず2章で、ボトルネックリンクの物理速度測定用にこれまで提案されたツールを紹介し、それらの有する問題点について述べる。次いで3章で、提案ツールの使用形態、測定原理、実装方法を既存ツールと対比させて述べる。4章では、ツールの有効性を検証するために実施したネットワーク実験の概要とその結果について述べる。更に、5章で今後の課題を示す。

2. 既存提案

インターネット内の各ルータを接続するリンクの物理速度を測定する手法は、(1)One-packet 推定法、(2)Packet-Pair 推定法の2つに大別できる [5]。

2.1 One-Packet 推定法

2.1.1 基本原理

インターネットに接続された1台のホストから送信されたパケットが、複数台のルータを経由して別のホストに届くまでの時間を考える。ルータはパケットをストア&フォワード処理し、一方、ルータ間のリンク (サブネットワーク) にはパケットをストア&フォワード処理する (レイヤ2以下の) ノードが存在しないとすると、以下の式が成立する [2], [5], [7]。

$$T_n = \sum_{k=0}^{n-1} \left(\frac{s}{b_k} + d_k + Q_k \right). \quad (1)$$

ここで、

T_n : n 番目のルータに届くまでの時間 (確率変数)

s : パケットサイズ (固定値)

b_k : k 番目のルータと $k+1$ 番目のルータ間のリンクの物理速度 (固定値)

d_k : k 番目のルータと $k+1$ 番目のルータ間の伝搬遅延 + k 番目のルータにおける (アドレス検索等の) パケットのフォワーディング処理に伴う遅延 (固定値)

Q_k : k 番目のルータでのバッファ待ち遅延 (確率変数)

(なお、0番目のルータはパケットを送出したホストを表すものとする。) 式 (1) より

$$T_{n+1} - T_n = \frac{s}{b_n} + d_n + Q_n. \quad (2)$$

ここで、 Q_n は他のトラヒックの存在に影響を受ける項であり、

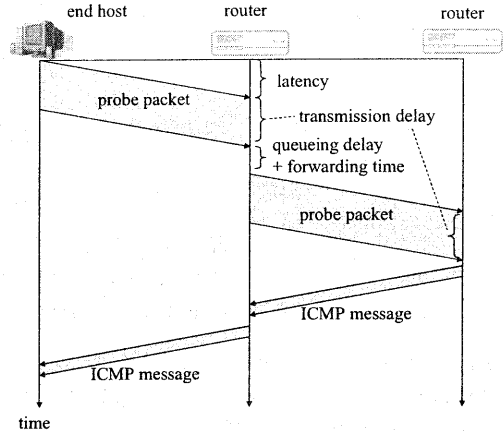


図1 RTT のモデル

ttl を 2 に設定した UDP データが送信され、ルータを通過する毎に ttl が 1 づつ減少し、2 番目のルータが ICMP "time exceeded message" を返す。

測定のたびに異なる値をとる一種の確率変数であるが、その最小値は 0 に等しいことから、

$$\min\{T_{n+1} - T_n\} = \frac{s}{b_n} + d_n. \quad (3)$$

つまり、複数回、 T_{n+1} と T_n の測定を行い、その差の最小値を取れば、その結果はパケットサイズに比例し、かつ比例定数は n 番目のルータと $n+1$ 番目のルータ間のリンクの物理速度の逆数に等しくなることが期待できる。

2.1.2 Pathchar

Jacobson は、2.1.1 章の原理を利用し、Pathchar と呼ばれるリンク速度測定ツールを開発した [3]。Pathchar は、試験パケットを利用した active 型のツールであり、最初に、TTL (Time to Live) を 1 に設定した試験パケットを宛先ホストに送る。すると、(TTL が 1 に設定されているため) 1 つ隣のルータから ICMP "time exceeded" メッセージが返される (図 1)。

試験パケットを送ってから、ICMP メッセージを受信するまでの時間は、試験パケットを送信したホストと隣のルータ間の RTT (Round Trip Time) に相当するが、Pathchar ではこの RTT を複数回測定し、その最小値を求める。2.1.1 で述べた原理から、RTT の最小値は試験パケットのサイズに線形比例し、その逆数が試験パケットを送信したホストと隣のルータ間のリンクの物理速度の推定値となる。Pathchar はこの RTT の最小値を (120 B 以上 1528 B 以下の) 45 通りのサイズの試験パケットを用いて測定し、RTT の最小値をパケットサイズを変数とする一次関数とみなして最小 2 乗法により直線近似し、求めた一次関数の傾きを使って帯域を推定する (図 2)。更に、順次 TTL を増やして同操作を繰り返すことで、宛先ホストに至る経路上の全てのリンクの物理速度を推定する。Pathchar の動作特性は Downey [2] が詳細に評価している。

Remark. なお、式 (3) は片道遅延に関するものであり、RTT とは異なるが、ICMP メッセージを格納したパケットは固定サイズであり、サイズも小さいため、RTT についても式 (3) と同様の関係が成立する。

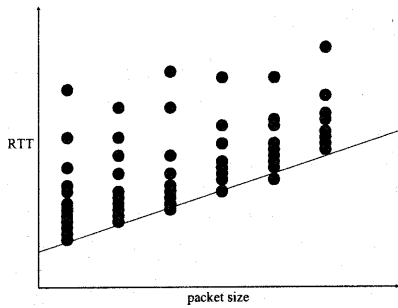


図2 最小 RTT の直線近似各パケットサイズでの RTT の最小値は直線近似可能。

2.1.3 Pathchar の問題点

Pathchar は簡便なツールであるが、以下のような問題点を有する。

- (1) 測定にあたって多くの試験パケットを必要とする (10 ホップの Ethernet の場合、約 10 MB の量の試験パケットが必要 [5])。特に、正確な推定のためには、少なくとも一つの試験パケットをバッファ遅延なしで全経路通過させる必要があり、そのため、経路が長くなるほど、必要な試験パケットの数も増える [2]。
- (2) ルータ間のリンク (サブネットワーク) にパケットをストア&フォワード処理する (レイヤ 2 以下の) ノードが存在すると、正しい推定ができない [10]。例えば、 n 番目のルータと $n+1$ 番目のルータが M 台のスイッチングハブを経由して Ethernet で接続されている場合、式 (3) は以下のように変更される。

$$\min(T_{n+1} - T_n) = \frac{(M+1)s}{b_n} + d_n \quad (4)$$

つまり、2.1.1 章で述べた原理で Ethernet の物理速度を推定すると、その結果は、実際の物理速度の $1/(M+1)$ となる。

その他にも、試験パケットと ICMP メッセージの経路が異なる可能性 [5] や、(ルーチングテーブルの更新に伴い) 測定の途中で経路が変更される可能性 [7] により、推定精度が損なわれるという問題点が指摘されている。

2.1.4 Pathchar の改良

式 (3) で示したように、最小 RTT とパケットサイズとは線形関係が成立するが、実際には有限回の測定から必ずしも最小 RTT が得られるとは限らない。Pathchar は (有限回の測定で得られた) RTT の最小値が正規分布に従うとして、RTT の最小値とパケットサイズとの線形関係を最小二乗法を用いて推定するが、Mah [6] は、RTT の最小値が必ずしも正規分布に従わないケースでも対応できるように、最小二乗法に代えてノンパラメトリック推定法を組み込んだツール (Pchar) を提案している。的場、阿多、村田 [7] は、測定の途中で経路が変更されても、安定した推定結果が得られるツールを提案している。一方、Downey [2] は、試験パケット量を軽減化したツール (clink) を考案している。但し、これらは 2.1.1 章の原理を利用しており、いずれも、invisible ノードにより推定結果が不正確になる問題を抱えている。

2.2 Packet-Pair 推定法

あるホストから、2つの同じサイズ (サイズを s とする) の

パケットを連続してインターネットを介して別のホストに送信する。ホスト間の経路上のリンク物理速度の最小値 (ボトルネックリンクの物理速度) を b_{min} とすると、次の関係が成立する [5]。

$$b_{min} = \frac{s}{T_{nowait}^2 - T_{nowait}^1} \quad (5)$$

ここで T_{nowait}^1 は先行パケットが途中ルータのバッファで待たされることなく送信先ホストに届くまでの時間を、 T_{nowait}^2 は後行パケットが (先行パケット以外の要因により) 途中ルータのバッファで待たされることなく送信先ホストに届くまでの時間をそれぞれ表す。

式 (5) の代わりに、 $b_{min} = s/(T^2 - T^1)$ が成立するとして、2つの試験パケットを連続送信し、ボトルネックリンクの物理速度を推定する方法が、幾つかの文献 (例えば、[1], [8]) において提案されている (T^1 は先行パケットが送信先ホストに届くまでの時間、 T^2 は後行パケットが送信先ホストに届くまでの時間)。また、Lai [4] は、関係式 $b_{min} = s/(T^2 - T^1)$ を利用した passive 型のリンク速度推定ツールを提案している。更に、Lai [5] は、両者のパケットサイズが異なる場合、更には3つ以上のパケットが連続送出される場合に式 (5) を拡張し、ボトルネックリンクのみならず、経路上の全てのリンクの物理速度を推定できるツールを考案している (invisible ノードの問題は解決されていない)。

Packet-Pair 推定法は、特定パス上に存在する「ボトルネックリンクの帯域」のみに興味がある場合に有効であるが、アプリケーションの観点から見ればこのパターンが多い。また、One Packet 推定法と異なり、ルータ間のリンクに invisible ノードが存在しても、正確にボトルネックリンクの速度を推定できる。但し、式 (5) からわかるように、 $b_{min} = s/(T^2 - T^1)$ はあくまで近似式であり、例えば、ボトルネックリンク以降、最初のパケットが経路上のルータで待たされても (この場合、ボトルネックリンク速度が大きめに推定される)、また最初のパケットと2番目のパケットの間に別のパケットが挿入されても (この場合は、ボトルネックリンク速度が小さめに推定される)、推定結果が不正確になる。

3. 提案ツールの概要

本章では、今回提案するボトルネックリンク物理速度推定ツール (以下 NetTracer と呼ぶ) の概要を述べる。

3.1 利用・動作形態

NetTracer は、Pathchar と同様、試験パケットを利用した active 型のツールである。ある二つのホスト (以下、ホスト A、B と呼ぶ) を結ぶ経路上のボトルネックリンクの物理速度とその存在場所を特定したい場合には、例えばホスト A 上で NetTracer を動作させる。NetTracer はホスト A からホスト B に向けて、同サイズの試験パケットをペアで連続送信する。試験パケットの TTL はいずれも同じ値 (例えば n) に設定されており、従って (ホスト A から見て) n 番目のルータから ICMP メッセージが返されるため、試験パケットの送信時刻と ICMP のメッセージの受信時刻の差から、ホスト A と n 番目のルータ間の RTT 値を (連続して) 測定できる。NetTracer は 3.2 章で述べる原理により、この RTT の測定結果からホスト A と n 番目のルータ間のボトルネックリンクの物理速度を推定し、更にこの動作を $n=1$ から順次 n の値を増やして繰り返すことで、ホスト A-B 間のボトルネックリンクの物理速度とボトルネックリンクの存在箇所を特定する。

上述のように、NetTracerはPathcharと類似した動作特徴を有するが、(1)試験パケットとしてペアパケットを利用(Pathcharは単体パケットを利用)、(2)ボトルネックリンクの物理速度と存在場所を推定(Pathcharは経路上の全てのリンクの物理速度を推定)する点でPathcharと異なる。

3.2 ボトルネックリンク物理速度推定原理

一方、NetTracerのボトルネックリンク速度推定法はPacket-Pair推定法のアイディアを利用している。但し、2.2章で述べたように、Packet-Pair推定法が通常利用する関係式 $b_{min} = s/(T^2 - T^1)$ はあくまで近似式であり、ネットワークの状態による推定誤差を含む可能性が高い。

そこで、NetTracerでは、複数回(例えば50回)パケットを連続送出し、先行パケットと後行パケットに関するRTTの最小値をそれぞれ求め、それらがそれぞれ T_{nowait}^1 及び T_{nowait}^2 に等しいとして式(5)に代入し、ボトルネックリンクの物理速度を求める。

なお、有限回の測定から得られた先行パケットと後行パケットのRTTの最小値は、必ずしも T_{nowait}^1 及び T_{nowait}^2 と一致せず、従って、推定結果に誤差が含まれる可能性がある。そのため、Pathcharと同様に、何通りかのサイズの試験パケットを用いて、RTTの最小値の差を測定し、RTTの最小値の差をパケットサイズを変数とする1次関数とみなして最小2乗法により直線近似し、求めた1次関数の傾きを使って帯域を推定する。これにより、一層、推定値の散らばりを抑えることができる。

3.3 実装方法

3.1章で述べたように、NetTracerはPathcharに類似した動作特徴を持つが、今回、ツールを(FreeBSD上で)開発するにあたり、以下で述べるように、Pathcharとは若干異なる形での実装を行った。

Pathcharはtracertouteコマンド[9]をベースにした実装を特徴としている。すなわち、Pathcharは試験パケットとしてUDPパケットを送信し、TTLが0になったルータからICMP“time exceeded”メッセージを受け取ることでRTTを測定する(通常使用されないポート番号を利用するので、パケットが宛先ホストに到達した場合にも、ICMP“unreach”メッセージが返される)。この場合、パケットの送信時刻とICMP“time exceeded”メッセージの受信時刻との差を取ることでRTTを計算するため、プログラム内でパケットの送信時刻を保持しなければならない。その結果、Pathcharでは「試験パケットの送出」と「ICMPメッセージの受信+RTTの算出」をシーケンシャルに実行する必要がある。

一方、NetTracerはpingコマンド[9]をベースにした実装を特徴とする。すなわち試験パケットとしてオプションデータ部分に送信時刻を書き込んだICMP“echo reply”メッセージを送信し、途中ルータもしくは相手先ホストからのICMP“echo”メッセージを受信する。この場合、RTTはICMP“echo”メッセージ受信時にヘッダを解析して、送信時刻を取り出し受信時刻との差を取ることで算出する。従って、「試験パケットの送出」と「ICMPメッセージの受信+RTTの算出」を別プロセスに分けて並列実行できるようになり、実行効率が上がる。実際、標準装備のpingでは、ICMP“echo reply”メッセージの送信とICMP“echo”メッセージの受信は同一プロセスで実行しているが、NetTracerの実装にあたっては、fork()関数を用いて送信用子プロセスと受信用親プロセスに分け、PCの処理時間によるオーバーヘッドの削減と効率の向上を図った。

4. 推定精度評価

本章では、提案ツールの推定精度を検証するために実施した、ネットワーク実験の概要の結果について述べる。

4.1 実験環境

図3に示したような千葉大学内のネットワークにおいて、ホストA-B1間及びホストA-B2間のボトルネックリンクの物理速度を、Pathchar、Packet-pair推定法、NetTracerの3種類の方法で測定した。ホストAとB1は100Base-TXで構築された同一のサブネットに属しており、従って、ホストA-B1間のボトルネックリンクの物理速度は100Mbpsである。なお、ホストA-B1間には2台のスイッチングハブが置かれている。一方、ホストAとホストB2は別のサブネットに属し、ホストB2側のサブネットも100Base-TXのLANであるが、ホストB2のネットワークインタフェースは10Mbpsであるため、ホストA-B2間のボトルネックリンクの物理速度は10Mbpsである。

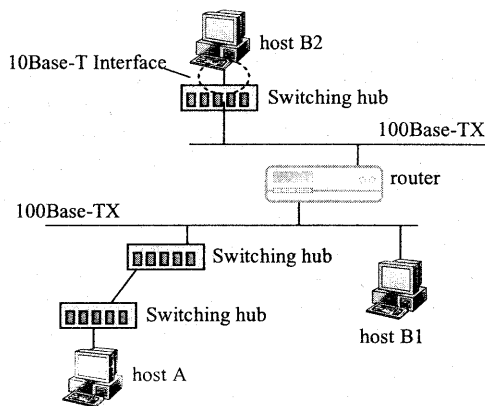


図3 実験環境

4.2 実験結果

4.2.1 ホストA-B1間

表1は、それぞれの推定法により、ホストA-B1間のボトルネックリンク速度を30回推定した結果をまとめたものである。2台のスイッチングハブが存在するため、2.1.3章で述べた理由により、Pathcharの推定結果は実際の物理速度(100Mbps)の約1/3となる。一方、NetTracerとPacket-pair推定法による推定値の平均はいずれも実際の物理速度に比較的近い。但し、Packet-pair推定法の推定結果のバラツキは大きい(最大188.2Mbps、最小19.7Mbps)。

表1 ホストA-B1間のボトルネックリンク速度の推定値

	Pathchar	Packet-pair	Nettracer
Average [Mbps]	33.1	112.1	94.4
Max [Mbps]	34.0	188.2	97.7
Min [Mbps]	33.0	19.7	88.5

このバラツキの詳細を見るために、Packet-pair推定法による

ボトルネックリンク速度の 1000 回の推定結果をヒストグラムに表したものを図 4 に示す。120Mbps 付近に推定値は集中しており、160Mbps 以上の推定結果を与えるケースも少なからず存在する。逆に推定結果が 40Mbps 以下になるケースも多く、推定結果が広範囲に分布していることが読み取れる。

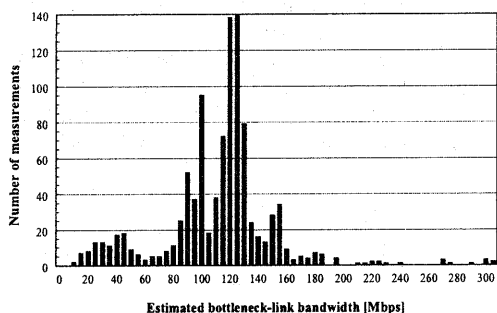


図 4 Packet-Pair による帯域推定値のヒストグラム

NetTracer はペアの packets を用いて複数回 RTT の測定を行い、(先行パケットと後行パケットの RTT の最小値を求めてから、両者の差をとるという意味での) フィルタリングを測定データに施してから帯域推定を行う Packet-pair 推定法の改良版と位置付けることができる。このフィルタリングの効果を見るために、図 5 に、十通りのパケットサイズにおいて、先行パケットの RTT (T^1) と後行パケットの RTT (T^2) の単純な差を取った結果 ($T^2 - T^1$) と、先行パケットと後行パケットの RTT の最小値を求めてから両者の差を取った結果 ($\min\{T^2\} - \min\{T^1\}$) を散布した図を示す。図 5 でわかるように、先行パケットと後行パケットの RTT の差は広範囲に分布する。一方、RTT の最小値の差は、傾きが 100 [Mbps] となる直線上にほぼ分布している。このことから、NetTracer で採用した、(RTT の最小値をとるという意味での) フィルタリングが推定結果を安定させるのに本質的に重要であることがわかる。

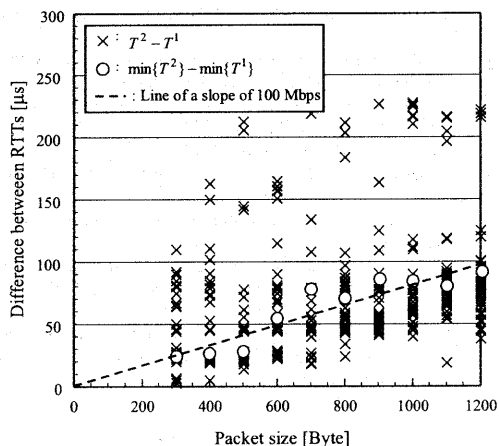


図 5 Packet-Pair による RTT の差と最小 RTT の関係
縦軸は先行パケットと後行パケットの RTT の差を取った値 [マイクロ秒]、横軸はパケットサイズ [Byte]。

4.2.2 ホスト A-B2 間

次に、ホスト A-B2 間のボトルネックリンク速度の推定結果を表 2 に示す。Pathchar の推定結果は実際のボトルネックリンク速度 (10Mbps) の 2/3 程度となる。また、Packet-Pair は、平均的には良好な推定結果を与えるが、推定値のバラツキは大きい。一方、Nettracer はバラツキの少ない、良好な推定結果を与える。

図 6 は、図 4 同様、Packet-pair 推定法による 1000 回の推定結果のヒストグラムを表したものである。推定値は 10Mbps 付近に集中しているものの、推定結果が 20Mbps 以上 (つまり、実際のボトルネックリンク速度の 2 倍以上) になるケースが少なからず存在する。一方、推定結果が 10Mbps 以下のケースも多く、図 4 同様、推定結果は広範囲に分布している。

表 2 ホスト A-B2 間のボトルネックリンク速度の推定値

	Pathchar	Packet-pair	Nettracer
Average [Mbps]	6.9	9.5	9.2
Max [Mbps]	7.0	13.4	9.3
Min [Mbps]	6.8	5.7	9.1

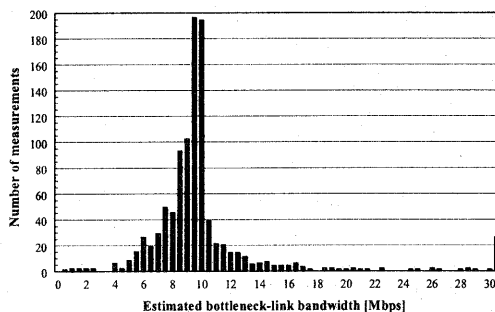


図 6 Packet-Pair による帯域推定値のヒストグラム

図 7 は、先行パケットと後行パケットの単純な RTT の差を取った結果 ($T^2 - T^1$) と、先行パケットと後行パケットの RTT の最小値の差を取った結果 ($\min\{T^2\} - \min\{T^1\}$) を散布図に示したものである。図 5 と同様に、先行パケットと後行パケットの RTT の差は広範囲に分布するが、RTT の最小値の差は、傾きが 10 Mbps に等しい直線上に集中しており、RTT の最小値を求め、その差からリンク速度を推定することの重要性がやはり確認できる。

4.3 広域実験

最後に、図 3 のホスト A と大学外に置かれたサーバ (以下ホスト C とする) 間の経路上のボトルネックリンク速度を NetTracer により実験的に測定した。ホスト C は一般のインターネットサービスプロバイダーに接続されており、ホスト A とホスト C の間には 13 台のルータが存在する (図 8)。ホスト C は ADSL によりインターネットサービスプロバイダーに接続されている。ADSL は下り 8Mbps、上りは約 700kbps である。

なお、(試験パケットのサイズを 1500byte 以下にすると、測定結果が必ずしも安定しない傾向があったため) この実験では

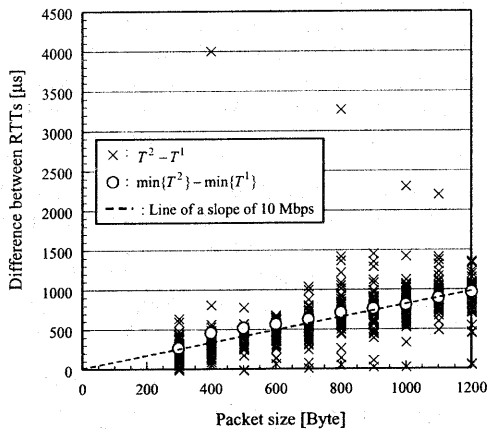


図7 Packet-PairによるRTTの差と最小RTTの関係
縦軸は先行パケットと後行パケットのRTTの差を取った値[マイクロ秒]、横軸はパケットサイズ[Byte]。

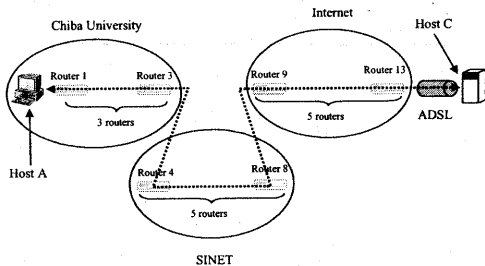


図8 広域実験

試験パケットのサイズは最大4000byte程度にして、3章の原理によりボトルネックリンク速度を測定した。

表3は、ホストAと途中のルータ間のボトルネックリンク速度に関する、ある一回の測定結果を示したものである。

表3 広域実験結果

path	Host A ⇔Router 1	Host A ⇔Router 2	Host A ⇔Router 3	Host A ⇔Router 4
bottleneck link's bandwidth	101.0[Mbps]	30.7[Mbps]	32.5[Mbps]	22.6[Mbps]
path	Host A ⇔Router 5	Host A ⇔Router 6	Host A ⇔Router 7	Host A ⇔Router 8
bottleneck link's bandwidth	54.6 [Mbps]	60.0 [Mbps]	69.0 [Mbps]	82.0 [Mbps]
path	Host A ⇔Router 9	Host A ⇔Router 10	Host A ⇔Router 11	Host A ⇔Router 12
bottleneck link's bandwidth	59.2 [Mbps]	55.9 [Mbps]	43.3 [Mbps]	13.1 [Mbps]
path	Host A ⇔Router 13	Host A ⇔Host C		
bottleneck link's bandwidth	21.6 [Mbps]	0.71 [Mbps]		

ホストA-C間のボトルネックはADSL上り回線の部分と思われるが、その値は正確に測定されている。一方、ホストAと途中のルータ間の経路上のボトルネックリンク速度の測定結果は必ずしも正確でない(例えば、千葉大学内の回線速度および

千葉大学とSINET間の回線速度は、実際には、100Mbps以上である)。また、この実験では、NetTracerはホストA側にインストールされており、従って、ホストCに近づくにつれ、ボトルネックリンクの速度は次第に小さくなるはずであるが、表3からは、そのような単調減少の傾向は見られない。

このように測定結果が不正確であることの原因の一つとして、(1500バイト以上のサイズの試験パケットを利用したことにより)特に、受信側ルータでのパケットの再構築/分割処理に時間がかかり、再構築/分割処理がむしろボトルネックになって、肝心の回線速度が測定されないこと等が考えられる。広域網で提案ツールを使用することを想定した、何らかの改善が必要である。

5. 結 論

本論文では、試験パケットを用いて、ホスト間を結ぶ経路上のボトルネックリンクの物理速度を推定するツール(NetTracer)を提案し、ネットワーク実験によりその有効性を検証した。本ツールにより、経路上にスイッチングハブを含むEthernetが存在しても、正確にボトルネックリンクを測定することが可能である。但し、本ツールはactive型の測定ツールであり、多くの測定用トラフィックをネットワークに印加する必要がある。測定に要するトラフィック量の軽減化が今後の課題である。

謝辞 提案ツールのFreeBSD上への実装とネットワーク実験の実施にあたり、有益な御助言・御指導をいただいた千葉大学工学部の松垣泰彦助手、池田宏明教授、ならびに千葉大学総合メディア基盤センタの岡野正明技術専門職員に深謝申し上げます。

文 献

- [1] J.-C. Bolot, "End-to-end packet delay and loss behavior in the Internet," *ACM SIGCOMM '93*, 1993.
- [2] A. B. Downey, "Using pathchar to estimate Internet link characteristics," *ACM SIGCOMM '99*, pp. 241-250, 1999.
- [3] V. Jacobson, "pathchar," <ftp://ftp.ee.lbl.gov/pathchar>.
- [4] K. Lai and M. Baker, "Measuring bandwidth," *IEEE INFOCOM '99*, 1999.
- [5] K. Lai and M. Baker, "Measuring link bandwidths using a deterministic model of packet delay," *ACM SIGCOMM '00*, pp. 283-194, 2000.
- [6] B. A. Mah, "pchar: A tool for measuring Internet path characteristics," <http://www.ca.sandia.gov/bmah/Software/pchar>.
- [7] 的場一峰, 阿多信吾, 村田正幸, "インターネットにおける統計的手法に基づいた帯域測定," 電子情報通信学会技術研究報告, SSE99-168, IN99-131, pp. 43-48, 2000.
- [8] V. Paxson, "End-to-end Internet packet dynamics," *IEEE/ACM Trans. Networking*, pp. 277-292 1999.
- [9] W. R. Stevens, "TCP/IP Illustrated, Volume 1: The Protocols," Addison-Wesley, 1994.
- [10] 八木敬宏, 塩田茂雄, 松垣泰彦, 池田宏明, "10Base-T/100Base-TX ベース LAN における Pathchar の特性評価," 電子情報通信学会ソサエティ大会, B-16-12, 2001.
- [11] Y. Vardi, "Network tomography: estimating source-destination traffic intensities from link data," *Journal of the American Statistical Association*, pp. 365-377, 1996.