

## 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<sup>†</sup> 塩田 茂雄<sup>†</sup> 間瀬 憲一<sup>††</sup>

A Measurement Methodology for End-to-End Bottleneck-Link Bandwidth

Takahiro YAGI<sup>†</sup>, Shigeo SHIODA<sup>†</sup>, and Kenichi MASE<sup>††</sup>

あらまし 本論文では、エンド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 (RateTracer) を提案する。提案ツールは、試験パケットをネットワークに印加し、そのレスポンス (ICMP メッセージ) の受信時刻に基づいて推定を行う active 型のツールであり、通常の Packet-Pair 推定法の推定原理を改良して用いることで、Packet-Pair 推定法に比べて高精度で安定した推定が可能であるという特徴を有する。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、ネットワーク実験を通して、提案ツールの有効性を示す。

キーワード インターネット、品質管理、ネットワーク計測、ボトルネックリンク、パケットペア

## 1. まえがき

インターネットは急速に普及し、今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、社会的に重要性の高いトラフィックや、IP 電話のように通信品質に敏感なアプリケーションのトラフィックも増えるにつれ、インターネットの品質管理の重要性が広く認識されるようになってきた。

通信ネットワークの品質管理は従来より存在する確立された業務の一つであるが [1]、インターネットの品質管理に際しては、インターネット特有の事情に配慮しなければならない。例えば、インターネットは様々なネットワークの集合体であり、ユーザは特定のインターネットサービス事業者とのみ契約するものの、インターネット利用中の通信経路は契約事業者以外のネットワークにもまたがって形成されることが多い。インターネットサービス事業者が自社ユーザのエンドツーエンド品質を管理するために、自社ネットワーク内の各リンクのトラフィック量・品質・物理的特性を把握することはもちろんであるが、例えば品質に関する

ユーザクレームが発生した場合、エンドツーエンドの通信経路上に存在する他事業者網の様子を知ることが必要になる。ところが、他事業者網のトポロジー、各リンクのトラフィック量・品質・物理的特性を直接測ることは、通常、不可能であるため、何らかの間接的な手法でこれらを推定せざるを得ない。

これと同様の事情は、例えば、インターネット上に Virtual Private Network (VPN) を構築する場合、若しくは通信事業者の IP-VPN サービス等を利用して VPN を構築する場合にも発生する。VPN の管理者は、ある対地間に大きな遅延やロスが発生した場合、まずは、その対地間で十分な物理帯域が確保されているかを確認する必要があるが、VPN の管理者がインターネット若しくは IP-VPN のバックボーンの物理帯域を直接知ることはできないため、これらを間接的に調査できる手段があると便利である。このような背景から、ネットワークの各リンクの状態を間接的に計測・推定する技術の研究開発が急速に進められている [2]。

本論文は、上記技術のうち、端末間の通信経路上のボトルネックリンクの物理速度 (帯域) を遠隔推定する技術について提案を行うものである。リンクの物理速度を推定する技術は、上述したようにネットワークの品質管理上重要であると同時に、例えば、ネットワーク内に分散する複数のプロキシ/コピーサーバから最適なサーバを選択する際に、または動画像等をストリーミング配信する場合の配信レイトを設定する

<sup>†</sup> 千葉大学工学部, 千葉市  
Faculty of Engineering, Chiba University, 1-33 Inage, Yayoi-ku, Chiba-shi, 263-8522 Japan

<sup>††</sup> 新潟大学工学部, 新潟市  
Faculty of Engineering, Niigata University, 2-8050 Ikarashi, Niigata-shi, 950-2181 Japan

際にご利用することができる。そのため、これまでに、Pathchar [3], Pchar [4], Nettimer [5] 等、数多くの推定原理・ツールが提案されているが、いずれも推定精度等にいくつかの問題点を抱えている。

本論文では、これら既存手法の長所を踏まえながら、より優れた推定精度を有すると考えられる手法を新たに提案するとともに、実際に Linux 上で動作するツールを作成し、いくつかのネットワーク実験を通して、提案手法の有効性を示す。

以下、本論文の構成を述べる。まず 2. 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、それらの特長及び問題点について述べる。次いで 3. で、提案ツールの測定原理、実装方法を述べる。4. 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. で今後の課題を示す。

## 2. 既存提案

インターネットを介したホスト間の通信には多くのルータが存在する。ホスト-ルータ間若しくはルータ-ルータ間は、ADSL のようなポイントツーポイントリンクで接続される場合もあれば、イーサネットのようなブロードキャスト型ネットワークを介して接続される場合もある。本論文では、このホスト-ルータ間若しくはルータ-ルータ間を接続するポイントツーポイントリンク若しくはブロードキャスト型ネットワークを総称して「リンク」と呼ぶこととする。

1. で述べたように、ホストを結ぶ通信経路上の各リンクの物理速度を測定する原理・ツールは数多く提案されており、これらは、(1) One-Packet 推定法、(2) Packet-Pair 推定法の二つに大別できる [6], [7]。

### 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 [6], [8], [9]。

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

ここで、

$T_n$ :  $n$  番目のルータに到達するまでの時間（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) において  $Q_n$  はクロストラヒックの存在に影響される項であり、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、

$$\min\{T_n\} = \sum_{k=0}^{n-1} \left( \frac{s}{b_k} + d_k \right). \quad (2)$$

したがって

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

つまり、何らかの方法により、 $T_{n+1}$  と  $T_n$  の最小値がそれぞれ測定できるならば、両者の差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い。

#### 2.1.2 Pathchar

Jacobson は、2.1.1 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、UDP パケットを利用した active 型のツールであり、例えば TTL (Time to Live) を  $n$  に設定した UDP パケットをあて先ホストに送る。すると、(TTL が  $n$  に設定されているため) 送信元ホストから見て  $n$  番目のルータから “ICMP Time Exceeded Message” が返される (図 1)。

2.1.1 と同様の考察により、TTL を  $n$  に設定した UDP 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  $\hat{T}_n$  についても、式 (3) と同様の関係式が成立することを確認できる。Pathchar は複数の UDP パケットを用いて  $\hat{T}_n$  を複数回測定し、その最小値を求める。 $(\hat{T}_n$  の複数回の測定値の最小値が  $\hat{T}_n$  の真の最小値である保証はないが、仮に、これが  $\hat{T}_n$  の真の最小値であるならば) UDP のパケットサイズ

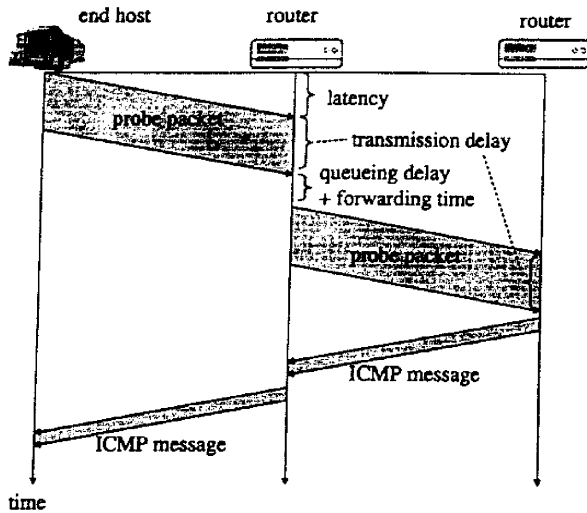


図1 Pathcharの推定原理  
Fig. 1 Estimation methodology of Pathchar.

を変えて、 $\bar{T}_n$ の最小値を異なる $n$ について測定することで、2.1.1で述べた原理から、あて先ホストに至る経路上のすべてのリンクの物理速度が推定できる。Pathcharの動作特性はDowney [8]が詳細に評価している。

### 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) TTLとパケットサイズの両方を変えてレスポンスタイム $\bar{T}_n$ の最小値を測る必要があるため、結果的に、多くの試験用パケットが必要になる(10ホップのイーサネットの場合、約10MByteの量の試験パケットが必要[6])。

(2) ルータ間のリン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(invisibleノードと呼ばれる[6])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イーサネットに接続されている場合、式(3)は以下のように変更される。

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

つまり、2.1.1で述べた原理でイーサネット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、(ルーティ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ことにより、推定精度が損なわれる等の問題点が指摘されている[9]。

### 2.1.4 Pathcharの改良

(2.1.2でも述べたように) $\bar{T}_n$ の有限回の測定値の最小値が $\bar{T}_n$ の真の最小値である保証はない。Pathcharでは、 $\bar{T}_n$ の有限回の測定値の最小値が、 $\bar{T}_n$ の真の最小値からずれる可能性があることを考慮し、そのずれが正規分布に従うことを仮定して、最小二乗法を用いてリンクの物理速度を推定している。Mah [4]は、このずれが必ずしも正規分布に従わないケースにも対応できるよう、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)を提案した。Pasztor, Veitch [11]は、 $\bar{T}_n$ の最小値でなく平均値に基づいてリンクの物理速度を推定する手法を提案している。的場, 阿多, 村田[9]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 [8]は、試験パケット量を軽減したツール(clink)を考案している。ただし、これらは2.1.1の原理を利用しており、いずれも、invisibleノードにより正しい推定値が得られない問題を抱えている。

## 2.2 Packet-Pair 推定法

### 2.2.1 基本原理

Packet-Pair 推定法は、その簡便さから、従来より広く使用されてきた手法である[12]~[15]。以下、その原理を説明する。時刻0に、二つの同サイズのパケットをあるホストから別のホストに向けて連続送信し、各パケットが途中ルータに到達する時刻を考える。仮に、クロストラヒックがないとすると、両ホストを結ぶ通信経路上のボトルネックリンクの手前で、後行パケットは先行パケットに続いてバッファリングされるため、ボトルネックリンク以降、両パケットの到着時刻は後行パケットのボトルネックリンクでの処理時間分、広がることになる(図2)。つまり、以下の式が(近似的に)成立する。

$$T_n^2 - T_n^1 \approx \frac{s}{\min\{b_1, \dots, b_{n-1}\}}. \quad (5)$$

ここで、 $T_n^1$ 及び $T_n^2$ は先行パケット及び後行パケットの $n$ 番目のルータへの到達時刻を表す。つまり、Packet-Pair 推定法は、何らかの方法により $T_n^1$ 及び $T_n^2$ (若しくは式(5)と同様の関係が成立する物理量)を測定し、その差でパケットサイズを割ることで、ボトルネックリンクの物理速度を推定するものである。

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速度」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

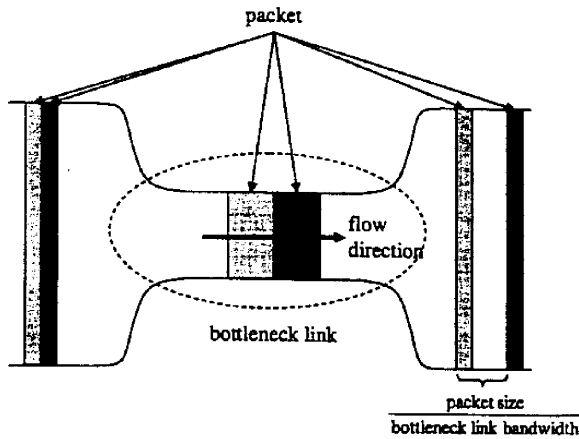


図 2 Packet-Pair 推定法の原理 [6]

Fig. 2 Estimation methodology of packet-pair technique [6].

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。

### 2.2.2 Packet-Pair 推定法の改良案

3.1 で述べるように、Packet-Pair 推定法の原理となる式 (5) は、パケットのバッファ待ち遅延がない場合にのみ成立する。実際には、パケットペアの到達時間の差分にはバッファ待ち遅延が含まれるため、Packet-Pair 推定法にはこれに起因する推定誤差が発生する。

このため、Packet-Pair 推定法の改良案がいくつか提案されてきた。Crater, Crovella [16] は Packet-Pair (正確には Packet-train [16]) 推定法によりボトルネックリンク速度の推定値が分布する範囲を複数回求め、例えば分布範囲がオーバーラップする箇所を抽出し、最終的な推定値とするツール (Bprobe) を提案している。Lai, Baker [5] は、Packet-Pair 推定法によるボトルネックリンク速度の推定値の分布密度関数を求め、分布密度関数の最大ピークの場所を推定結果として与えるツール (Nettimer) を提案している。一方、Dovrolis ら [17] は、Nettimer に類似した推定技術を利用したクライアント-サーバ型のツール (Pathrate) を提案している。これらは、Packet-Pair 推定法で得られた複数の推定結果から真の推定値を抽出する手法に工夫をこらしたものであるが、いずれもヒューリスティックな方法論に頼っており、必ずしも明快な理論的な裏付けをもたない。特に、恒常的に負荷の大きいリンクでは、例えば MTU サイズのパケット  $N$  個分のバッファ待ち遅延が発生しやすい傾向があるため、単に分布密度関数のピーク値から推定する手法では誤推定が発生する。3. で述べるように、提案手法では、

バッファ待ち遅延の影響をあらかじめ除外することによって推定精度の向上を達成している。

## 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 RateTracer) の概要を述べる。

### 3.1 基本原理

RateTracer は、基本的には、(Pathchar の考え方を取り入れて) Packet-Pair 推定法の原理を改良し、精度を高めた推定ツールである。以下、基本原理を説明する。時刻 0 に、二つの同サイズの packets をあるホストから別のホストに向けて連続送信した場合、各パケットの途中ルータへの到達時刻を考える。2. で説明した変数に加え、以下の変数を導入する。

- $Q_k^1$ : 先行パケットの  $k$  番目のルータでのバッファ待ち遅延 (確率変数)
- $Q_k^2$ : 先行パケットと後行パケットの間に挿入された別のパケットにより生ずる、後行パケットの  $k$  番目のルータでのバッファ待ち遅延 (確率変数)
- $\tilde{Q}_k^2$ : 先行パケットにより生ずる、後行パケットの  $k$  番目のルータでのバッファ待ち遅延 (確率変数)

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。

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

$$T_n^2 = \sum_{k=0}^{n-1} \left( \frac{s}{b_k} + d_k + Q_k^2 + \tilde{Q}_k^2 \right). \quad (6)$$

ここで [6],

$$\tilde{Q}_k^2 = \max\{0, T_{k+1}^1 - d_{k+1} - T_k^2\}. \quad (7)$$

$Q_k^1$  及び  $Q_k^2$  はクロストラヒックの影響で発生する項であるが、その最小値は 0 に等しいことから、

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

$$\min\{T_n^2\} = \sum_{k=0}^{n-1} \left( \frac{s}{b_k} + d_k + \tilde{Q}_k^2 \right). \quad (8)$$

式 (8) から、次の関係が成立することが導かれる [6].

$$\min\{T_n^2\} - \min\{T_n^1\} = \frac{s}{\min\{b_1, \dots, b_{n-1}\}} \quad (9)$$

式(9)は、クロストラヒックに影響されない(つまり、先行パケットにバッファ待ち遅延が発生せず、先行パケットと後行パケットの間に別のパケットが挿入されない)場合に、 $T_n^1$ と $T_n^2$ はその最小値をとり、更なる場合には、Packet-Pair推定法が仮定する関係式(式(5))が厳密に成立することを示している。3.2で述べるように、本論文で提案するツールは、 $T_n^1$ と $T_n^2$ (に類似した量)をそれぞれ複数回測定し、複数回の測定結果の最小値を式(9)に代入してボトルネックリンク速度を推定することで、通常のPacket-Pair推定法の精度を改善させたものである。

(なお、式(9)は、スイッチングハブのように(レイヤ2以下で)パケットをストア&フォワード処理するノードがあっても成立する。)

[注3.1]  $T_n^1$ と $T_n^2$ の複数回の測定結果の最小値が真の最小値である保証はなく、これが真の最小値であると仮定してボトルネックリンク速度を推定する本方式には、(特に測定対象リンクの負荷が大きい場合)この仮定に起因する推定誤差が発生する。2.1.4でいくつか紹介したように、この推定誤差の除去法についてはいくつかの既存提案があるが、本研究では、まずは実装の簡明さを優先し、(複数回の測定結果の最小値を真の最小値と仮定する)最も単純な「Pathchar-like」な方式を採用した。

### 3.2 実現方法

式(9)の原理に基づいてボトルネックリンク速度推定ツールを実現する方法には、いくつかの選択肢が存在する。本研究では、利便性を考慮し、単体ホストで動作するツールとして実装することとした。すなわち、試験パケットをペアで連続送信し、それぞれのパケットに対するネットワークからの応答(ICMPメッセージ)の受信時刻の差から、式(9)によりボトルネックリンク速度を推定する形態である。

単体ホスト上のツールとしての実装方法にも複数の選択肢が存在するが、本研究では以下のようなTracerouteベース[18]の方式を採用した。まず、ツールを実装したホストから、同サイズのUDPパケットをペアで連続送信する。UDP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、したがって $n$ ホップ目のルータからそれぞれのパケットに対して“ICMP Time Exceeded Message”(エンドホストからは“ICMP Port Unreachable Message”)が返

される。先行UDPパケット及び後行UDPパケットを送ってから、各UDPパケットに対するICMPメッセージを受信するまでの時間をそれぞれ $\hat{T}_n^1$ 及び $\hat{T}_n^2$ とすると、 $\hat{T}_n^1$ と $\hat{T}_n^2$ の最小値についても式(9)と同じ関係式が成立するため、UDPペアパケットの連続送信を $n$ を固定して例えば100回行い、(100回中の) $\hat{T}_n^1$ 及び $\hat{T}_n^2$ の最小値をそれぞれ求める。次いで、得られた最小値が、真の最小値に等しいと仮定して式(9)に代入し、ホストと $n$ 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する(注1)。

なお、試験パケットのサイズはパスMTUに設定することが望ましいが、パスMTUが分かるとは限らないため、本ツールでは試験パケットのデフォルト値を(IPヘッダ込みで)1500バイトに設定している。

[注3.2] Tracerouteベースの実装の場合、行きの経路(ツールを実装したホストから相手ホストに向かう経路)では1500バイトのUDPパケットが用いられ、帰りの経路ではICMPパケットが用いられる。ICMPパケットはUDPパケットに比べて小さいため(56バイト)、途中ルータ若しくはエンドホストから送信された直後のICMPペアパケットの時間間隔は、ボトルネックリンクの影響を受けず、そのまま保持されて、ツールを実装したホストに到達する。したがって、ADSLのように速度が非対称なリンクが経路上に存在する場合、基本的にツールを実装したホストから測定対象ホストに向かう方向のリンク速度のみが測定対象になり、逆方向のリンクの速度はボトルネックリンク速度としては検出されない。

一方、Pingベースの実装、すなわち、ICMPエコー要求/エコー応答メッセージを用いた実装[19]の場合、行き/帰りの経路ともに同じサイズのパケットが用いられるため、行き/帰り両経路通しての最小速度リンク(ボトルネックリンク)が抽出される。しかし、ICMPパケットは、各ルータにおける転送優先度が、TCPやUDPパケットに比べて、低く設定されていることが多い。そのため、先行パケットと後行パケットの間にクロストラヒックのパケットが挿入される可能性が高く、上述したTracerouteベースの実装方法に比べて推定精度の点で不利である。

[注3.3] 本提案方式により、1500バイトの試験パケットを用いて、100Mbit/s程度のボトルネックリン

(注1): 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、ボトルネックリンクの物理速度のみならずボトルネックリンクの存在箇所も同時に特定することができる。

クを測定するためには、 $1500 \times 8 / 100 \times 10^3 = 0.12 \text{ ms}$ 程度の時刻精度が要求される。このレベルの時刻精度を通常の Linux カーネルで長期にわたり保持することは困難であるが、連続するパケットの到着時刻の差を測定するという本方式の目的のためには、短期的にこのレベルの時刻精度が保証されていればよく、実際に4. で示すように、通常の Linux カーネル上に実装したツールにより、100 Mbit/s 程度のリンク速度を実用上十分な精度で測定できることを確認している。

### 3.3 提案方法の特徴

提案ツールでは、パケットサイズを MTU に固定して測定すればよく、試験パケットのサイズを45通り変えて測定を行う Pathchar に比べると、必要な試験パケット量は少ない。また、提案ツールは、先行パケットと後行パケットに関するレスポンスタイム  $\hat{T}_n^1$  と  $\hat{T}_n^2$  を複数回測定し、(それぞれ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結果に施してから推定を行う Packet-Pair 推定法の簡易な改良版と見ることが出来る。このフィルタリング手順は、(Nettimer や Pathrate 等の Packet-Pair ベースのツールに必要な)  $\hat{T}_n^1$  と  $\hat{T}_n^2$  の差の分布密度関数の同定に比べると単純であり、そのため実装も容易である。

なお、試験パケットのパケットフラグメンテーションが起きても、原理的には同じメカニズムでボトルネックリンク速度は測定できるが、パケットフラグメンテーションが生じるケースでは測定誤差が発生しやすい。IPsec, PPPoE, EoMPLS などのトンネリング技術を用いて構築された VPN ではパケットフラグメンテーションが生じやすく、このような場合、本ツールは必ずしも十分な推定精度が得られないことを付記しておく。ATM やロードバランサーでマルチリンクが多重化されている場合も同様である。

## 4. 提案ツールの特性評価

本章では、提案ツールの推定精度等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。

### 4.1 学内 LAN 環境での実験

#### 4.1.1 実験環境

図3に示したような千葉大学内の運用中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、(単純な) Packet-Pair 推定法<sup>(注2)</sup>、Nettimer、RateTracer

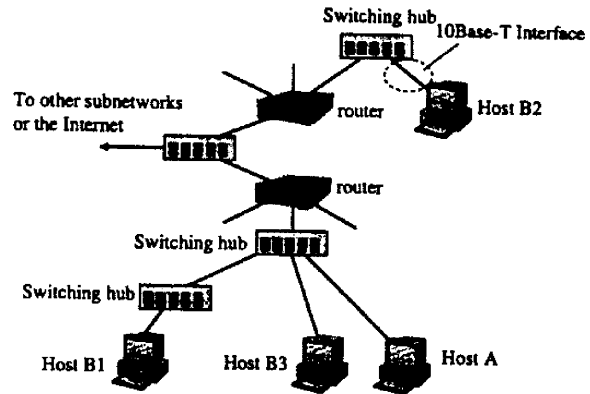


図3 実験環境

Fig. 3 Network-experimental environment in Chiba University.

の4種類の方法で測定した。Pathchar, Packet-Pair 推定法, Nettimer, RateTracer はいずれもホスト A (Linux) にインストールした (なお、他の代表的な既存ツールに Pathrate [17] があるが、Pathrate はクライアント-サーバ型のツールであり、他のツールと実装方法が異なるため、今回、比較対象とはしなかった)。

ホスト A とホスト B1 は 100Base-TX で構築された同一のサブネットに属しており、したがって、ホスト A-B1 間のボトルネックリンクの物理速度は 100 Mbit/s 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 10 Mbit/s 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 10 Mbit/s である。ホスト A が属するサブネットとホスト B2 が属するサブネットは 2 台のルータを介して接続されている。なお、ホスト A が属するサブネットは 15 人程度、ホスト B2 が属するサブネットは数十人程度が常時利用している。

#### 4.1.2 実験結果：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 で述べた理由により、Pathchar の推定結果は実際の物理速度 (100 Mbit/s) の約 1/3 となる。それに比べると、単純な Packet-Pair 推定法による平均推定値は実際の物理速度に比較的近いが、推

(注2)：本章では、Packet-Pair 推定法をホスト A から UDP のペーパケットを送り、ICMP Port Unreachable Message 等の受信時刻の差からボトルネックリンク速度を推定する手法として定義した。

表 1 ホスト A-B1 間のボトルネックリンク速度の推定値  
Table 1 Estimated bottleneck-link bandwidth between hosts A and B1.

	Pathchar	Packet-pair	Nettimer	RateTracer
Average [Mbps]	33.1	112.1	98.9	104.4
Max [Mbps]	34.0	188.2	117.6	107.1
Min [Mbps]	33.0	19.7	41.2	96.5

表 2 ホスト A-B1 間のボトルネックリンク速度の推定値 (Iperf 使用時)

Table 2 Estimated bottleneck-link bandwidth between hosts A and B1 when the network is heavily loaded.

	Pathchar	Packet-pair	Nettimer	RateTracer
Average [Mbps]	31.2	88.7	90.1	93.9
Max [Mbps]	31.0	180.5	99.5	97.6
Min [Mbps]	32.0	12.0	40.2	83.3

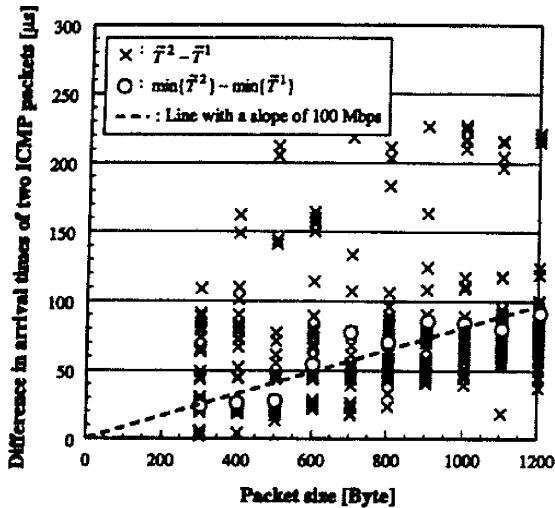


図 4 先行パケットと後行パケットのレスポンスタイムの差の散布図  
Fig. 4 Difference in RTTs between the first and second packets.

表 3 ホスト A-B2 間のボトルネックリンク速度の推定値  
Table 3 Estimated bottleneck-link bandwidth between hosts A and B2.

	Pathchar	Packet-pair	Nettimer	RateTracer
Average [Mbps]	5.5	241.0	44.8	10.4
Max [Mbps]	5.6	3925.3	46.5	14.0
Min [Mbps]	5.4	7.1	36.3	9.1

定結果のばらつきは大きい (最大 188.2 Mbit/s, 最小 19.7 Mbit/s)。これらに比べて, Nettimer と RateTracer の推定結果は, いずれも正確でかつ安定している。

RateTracer は、「先行パケットと後行パケットのレスポンスタイム<sup>(注3)</sup>を複数回測定し, それぞれの最小値をとる」というフィルタリングを測定データに施してから推定を行う Packet-Pair 推定法の改良版と位置づけることができる。このフィルタリングの効果を見るために, 図 4 に, 10 通りのパケットサイズについて, 先行パケットに関するレスポンスタイム ( $\bar{T}^1$ ) と後行パケットに関するレスポンスタイム ( $\bar{T}^2$ ) の単純な差をとった結果 ( $\bar{T}^2 - \bar{T}^1$ ) と, 先行パケットと後行パケットのレスポンスタイムの最小値を求めてから両者の差をとった結果 ( $\min\{\bar{T}^2\} - \min\{\bar{T}^1\}$ ) の散布図を示す。図 4 から分かるように,  $\bar{T}^2 - \bar{T}^1$  の測定値は広い範囲に分布する。一方,  $\min\{\bar{T}^2\} - \min\{\bar{T}^1\}$  は傾きが 100 Mbit/s の逆数に等しい直線上にほぼ分布

している。このことから, RateTracer で採用した (レスポンスタイム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上で本質的に重要であることが分かる。

更に, 参考のために, ホスト B3 からホスト B1 に Iperf [20] で 50 Mbit/s の負荷を掛けながら, 同様の測定を行った結果を表 2 に示した。表 1 と表 2 の結果には大きな差が見られず, 少なくともこの構成においては, リンク負荷の程度が測定結果に与える影響は比較的小さかった。

#### 4.1.3 実験結果: 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 3 に示す。Pathchar の推定結果は, 実際のボトルネックリンク速度 (10 Mbit/s) の半分程度の値であった。加えて, Packet-Pair 推定法と Nettimer の推定結果の平均値は, とともに実際のボトルネックリンクの速度とは大きく異なる。これらに比べて, RateTracer の推定結果は正確である。

Nettimer の推定誤差の原因を説明するために, 図 5 に, 単純な Packet-Pair 推定法による 400 回の推定結果のヒストグラムを示した。Nettimer は, 大まかにいうと, 単純な Packet-Pair 推定法による複数回のポ

(注3): ここでは, レスポンスタイムを「UDP 若しくは ICMP Echo Message を送ってから, ICMP Time Exceeded Message 若しくは ICMP Echo Reply Message を受信するまでの時間」の意味で用いる。

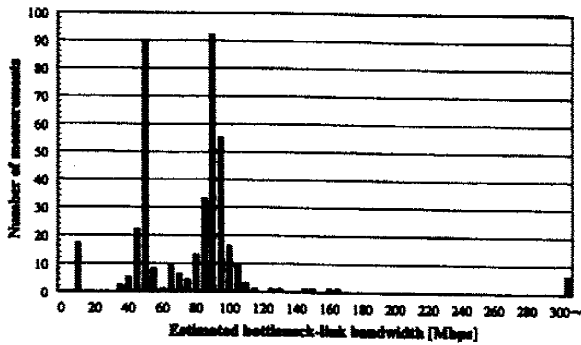


図5 Packet-Pairによる帯域推定値のヒストグラム  
Fig.5 Histogram of estimated bottleneck-link bandwidth by packet pair technique.

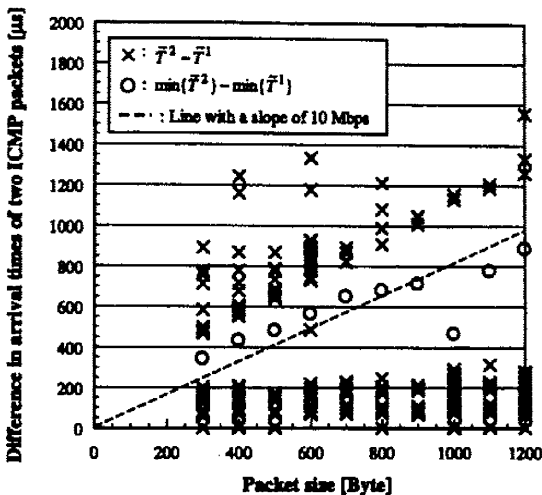


図6 先行パケットと後行パケットのレスポンスタイムの差の散布図  
Fig.6 Difference in RTTs concerning the first and second packets.

トルネックリンク速度の推定結果からそのヒストグラム(分布密度関数)を推定し、ヒストグラムがピークをとる場所を最終的な推定値とする方法である。図5に見られるように、ヒストグラムは10 Mbit/s付近に小さなピークをもつが、50 Mbit/s付近と90 Mbit/s付近により大きなピークをもつことを確認できる。50 Mbit/s及び90 Mbit/s付近のピークは、ホストB2から返される先頭のICMPパケットがスイッチングハブ若しくはルータのバッファで待たされるためによるものと推測できるが、Nettimerが45 Mbit/s付近の値を誤推定する傾向にあるのは、この50 Mbit/s付近のピークを、ボトルネックリンク速度と誤って推定するためと考えられる。

図6は、先行パケットと後行パケットに関するレスポンスタイムの単純な差をとった結果( $\bar{T}^2 - \bar{T}^1$ )と、先行パケットと後行パケットのレスポンスタイムの最

表4 推定に要する試験パケット量  
Table 4 The amount of packets required for estimation.

Tool	The amount of traffic
Pathchar	1065 kB
Nettimer	7.4 kB
RateTracer	303 kB

小値の差をとった結果( $\min\{\bar{T}^2\} - \min\{\bar{T}^1\}$ )を散布図に示したものである。図5と同様に、 $\bar{T}^2 - \bar{T}^1$ の実測値は広範囲に分布するが、 $\min\{\bar{T}^2\} - \min\{\bar{T}^1\}$ は傾きが10 Mbit/sの逆数に等しい直線付近に分布しており、レスポンスタイムの最小値を求め、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。

#### 4.1.4 推定に要する試験パケット量

更に、この学内LAN環境において、Pathchar, Nettimer, RateTracerの3種類のツールで使用する試験パケット量をそれぞれtcpdumpツールを用いて計測した。結果を表4に示す(ここで、RateTracerは100個のペアパケットを用いた場合の結果である)。試験パケット量はPathcharが最も多く、Nettimerが最も少ない。RateTracerは両者の中間であるが、Nettimerの40倍程度の量の試験パケットを使用しており、この軽減化が今後の課題である。なお、RateTracerについては、ペアパケットを送受信するたびに推定値を更新し、ペアパケットN回分、推定結果が変わらなかった場合に、動作を停止させるようなツールへの改良も行っており、N=30に設定すると、100個のペアパケットを用いた場合と同等の推定精度を保持しつつ、試験パケット量を半減できることを確認している[21]。

また、ペアパケットを送受信するたびに推定値を更新するだけでなく、それまで得られた推定値の平均と分散を求め、平均と分散に基づいて例えば95%信頼度区間を計算し、信頼度区間が一定値以下に収まった時点で測定を終了するようなツールへの改良も考えられる。この場合、推定値のみならず、推定値の信頼度区間も同時に求めることができる。

#### 4.2 広域実験

次に、図3のホストAと大学外に置かれた3台のノード(以下、C1, C2, C3とする)間の経路上のボトルネックリンク速度をPathchar, Nettimer, RateTracerにより測定した(図7)。C1は一般のイ



インターネットサービスプロバイダに ADSL 回線 (下り 12 Mbit/s, 上りは約 700 kbit/s) により接続されたホスト, C2 は Abilene (Internet2 の基幹伝送網: 米国) 内のルータ, C3 は GEMnet (NTT が保有する国際実験ネットワーク) 内のホストである. 千葉大学内のリンク速度はすべて 100 Mbit/s であり, Router 1 と Router 2 の間には, 一部, ATM 回線が使われている. 一方, 図 7 の SINET 及び Abilene 内のすべてのリンク速度は 100 Mbit/s 以上である. また, GEMnet 内の一部 (Router 10 ⇔ Router 11) に人為的に ATM による 10 Mbit/s のボトルネックリンクを形成した. GEMnet 内の Router 9 ⇔ Router 10 間も 110 Mbit/s の ATM 回線であり, Router 11 ⇔ Host C3 間は Fast Ether である. 一般のサービスプロバイダ内のリンク速度は不明である.

4.2.1 広域実験結果: A-C1 間

ホスト A-C1 間のボトルネックリンク速度の 10 回の推定結果の平均値を表 5 に示す. 表には, 経路上の

各ルータとホスト A 間のボトルネックリンク速度の推定結果及び RTT<sup>(注4)</sup> も併せて記した. なお, Pathchar は各リンクの物理速度を直接計測するツールであるため, Pathchar の欄には, Pathchar で計測したそれぞれリンクの物理速度のうち最小のリンク速度をボトルネック速度として表示した. 更に, Pathchar では, 6 ホップ目以降一部の経路 (Router 6 ⇒ Router 7, Router 7 ⇒ Router 8, Router 10 ⇒ Router 11) の物理リンク速度が計測できなかったため, 6 ホップ目以降については, 計測不能であったリンクを除く最小リンク速度を暫定的なボトルネックリンク速度として括弧付きの数値で示した. Nettimer については, 最終ホップを除く 8 ホップ目以降のボトルネックリンク速度は計測できなかったため, 該当欄には数値を記載しなかった. また, 「actual bottleneck link bandwidth」の欄には, 各ホップまでのボトルネックリンク速度を

(注4): RTT は, UDP パケットを送信してから, ICMP Time Exceeded Message を受信するまでの時間として計測した.

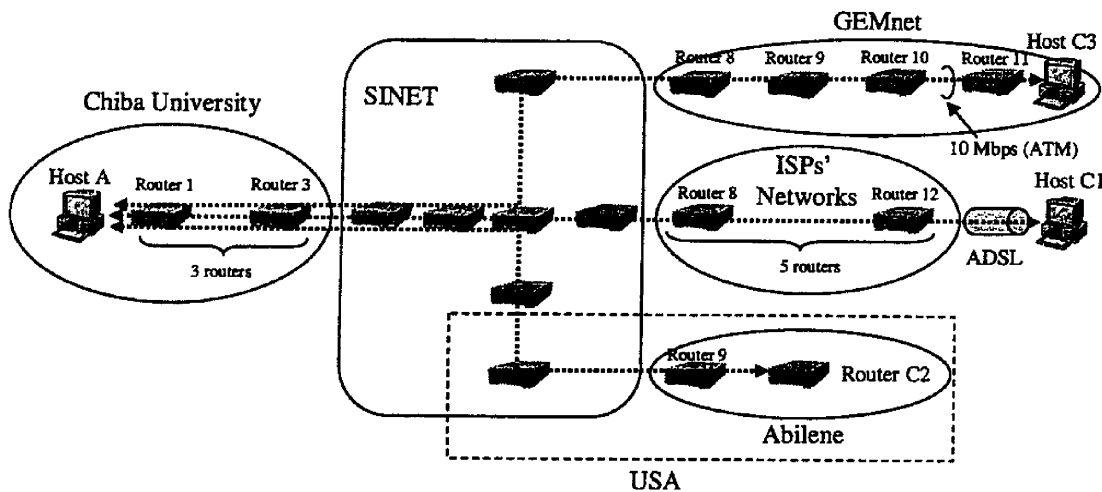


図 7 広域実験  
Fig. 7 Experimental network environment outside of Chiba University.

表 5 広域実験結果 (A-C1): 平均推定値  
Table 5 Estimated bottleneck-link bandwidth between A-C1.

path	Host A ⇔ Router 1	Host A ⇔ Router 2	Host A ⇔ Router 3	Host A ⇔ Router 4	Host A ⇔ Router 5	Host A ⇔ Router 6	Host A ⇔ Router 7	Host A ⇔ Router 8	Host A ⇔ Router 9	Host A ⇔ Router 10	Host A ⇔ Router 11	Host A ⇔ Router 12	Host A ⇔ Host C1
Rate/Denom [Mbps]	97.3	160.7	86.2	85.5	98.9	98.2	113.8	104.8	99.8	88.9	115.5	50.8	11.9
Pathchar [Mbps]	49.0	15.0	15.0	15.0	15.0	(15.0)	(15.0)	(15.0)	(15.0)	(15.0)	(15.0)	(15.0)	(3.0)
Nettimer [Mbps]	35.5	41.3	46.5	28.9	34.7	27.2	44.3	-	-	-	-	-	4.9
Actual bottleneck link bandwidth [Mbps]	100	100	100	100	100	100	100	(≤100)	(≤100)	(≤100)	(≤100)	(≤100)	12
RTT [ms]	0.791	2.65	2.98	3.22	4.06	4.83	4.65	5.12	5.19	5.28	5.50	5.70	28.9

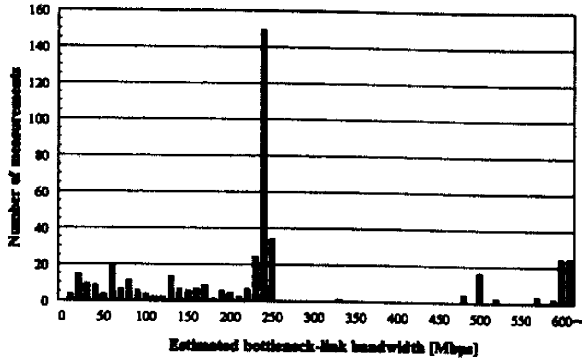


図 8 Packet-Pair 推定法によるホスト A-Router8 間ボトルネックリンク速度推定値のヒストグラム

Fig. 8 Histogram of estimated bottleneck-link bandwidth between host A and router 8 by packet pair technique.

表示したが、8 ホップ以降は実際のボトルネックリンク速度が (100 Mbit/s 以下であるという点を除き) 分からないため、「( $\leq 100$ )」と表示した。

ホスト A-C1 間のボトルネックリンクは ADSL 回線の部分であり、特に、A  $\Rightarrow$  C1 方向の経路上のボトルネックリンクの速度は 12Mbit/s である。RateTracer はこのボトルネックリンク速度を正確に推定しているが、Pathchar と Nettimer による推定値は真値の半分以下の値である<sup>(注5)</sup>。

ホスト A から 8 ホップ目 (Router 8) までの経路のボトルネックリンク速度は 100 Mbit/s である。この 8 ホップ目までについて見ると、RateTracer による推定結果は、Host A  $\Rightarrow$  Router 2 を除き、おおむね正確である<sup>(注6)</sup>。これに比べて、他の二つのツールの推定精度は落ちる。特に、既に述べたように、Pathchar では一部の経路 (Router 6  $\Rightarrow$  Router 7, Router 7  $\Rightarrow$  Router 8, Router 10  $\Rightarrow$  Router 11) のリンク速度が計測不能であり、Nettimer についても、最終ホップを除く 8 ホップ目以降は計測不能であった。

Nettimer が 8 ホップ目以降計測できない原因を分析するため、単純な Packet-Pair 推定法により Host A  $\Rightarrow$  Router 8 間のボトルネックリンク速度を 400 回推定し、推定結果のヒストグラムを図 8 に示した。推定結果は広範囲に分布する。また、240 Mbit/s 付近にヒストグラムのピークが存在するが、それ以外には特徴的なピークは見当たらない。(240 Mbit/s 付近にピークをもつなど) 推定値が 100 Mbit/s 以上に偏るのは、ボトルネックリンク以降の往路若しくは復路で先頭パケットにバッファ待ち遅延が発生するためと推測できるが、このように、バッファ待ち遅延の影響により真

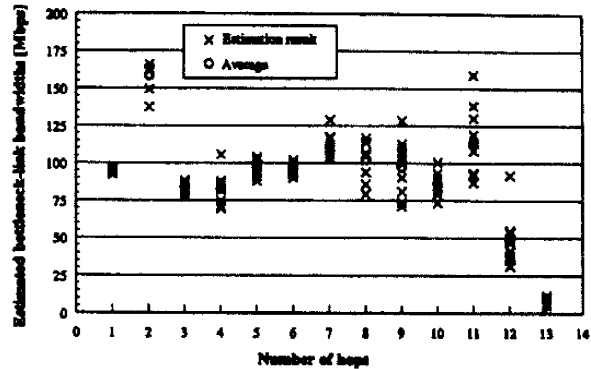


図 9 RateTracer による A-C1 間のボトルネックリンク速度推定値の散布図

Fig. 9 Estimated bottleneck-link bandwidths between A-C1 by RateTracer.

値はほとんどマスクされてしまい、(Nettimer のように) ヒストグラムの情報からボトルネックリンク速度に関する情報を抽出するのは実際には困難であることが確認できる。

なお、ISP のバックボーンネットワーク内のリンク速度は、通常、100 Mbit/s 以上であり、したがって、Host A  $\Rightarrow$  Router 12 間のボトルネックリンク速度は本来 100 Mbit/s であると思われるが、この間のボトルネックリンク速度の RateTracer による推定結果は平均 50.8 Mbit/s であった。この誤差の原因は判明していないが、Router 11  $\Rightarrow$  Router 12 にレイトを 50 Mbit/s 程度に絞った ATM 回線が使われている可能性もある。

図 9 には、各ホップまでのボトルネックリンク速度の RateTracer による 10 回の推定結果の散布図を示した。7 ホップ目までは推定結果は比較的安定しているが、8 ホップ目以降は推定結果の分散が大きくなる。これは、ホップ数が増えると、クロストラヒックの影響を十分フィルタリングできなくなるためと思われる。なお、最終リンク (ボトルネックリンク) の物理速度の推定結果は安定している。

#### 4.2.2 広域実験結果：A-C2 間

表 6 はホスト A-ルータ C2 間のボトルネックリンク速度の 10 回の推定結果をまとめたものである。このように、国際回線を含む経路についても RateTracer

(注5)：ホスト A から見て上り方向の経路上のボトルネックリンクの速度は 0.7 Mbit/s であり、Ping ベースのツールを用いると、その値もおおむね正確に推定できる [19]。

(注6)：Host A  $\Rightarrow$  Router 2 間の推定結果が実際と食い違う原因は判明していないが、Router 1 と Router 2 の間には 156 Mbit/s の ATM 回線が途中に使用されており、これが原因している可能性がある。

表 6 広域実験結果 (A-C2 間): 平均推定値  
Table 6 Estimated bottleneck-link bandwidth between A-C2.

path	Host A⇔ Router 1	Host A⇔ Router 2	Host A⇔ Router 3	Host A⇔ Router 4	Host A⇔ Router 5	Host A⇔ Router 6	Host A⇔ Router 7	Host A⇔ Router 8	Host A⇔ Router 9	Host A⇔ Router C2
RateTracer [Mbps]	106.7	149.3	93.1	121.5	98.2	101.6	97.2	116.2	98.7	101.7
Pathchar [Mbps]	31.0	15.0	15.0	15.0	15.0	15.0	15.0	(15.0)	(15.0)	(15.0)
Nettimer [Mbps]	36.9	40.8	40.6	24.4	29.1	33.0	45.2	48.5	45.6	46.3
Actual bottleneck link bandwidth [Mbps]	100	100	100	100	100	100	100	100	100	100
RTT [ms]	0.798	2.62	2.98	3.27	4.10	4.89	4.90	180.4	195.0	201.4

表 7 広域実験結果 (A-C3 間): 平均推定値  
Table 7 Estimated bottleneck-link bandwidth between A-C3.

path	Host A⇔ Router 1	Host A⇔ Router 2	Host A⇔ Router 3	Host A⇔ Router 4	Host A⇔ Router 5	Host A⇔ Router 6	Host A⇔ Router 7	Host A⇔ Router 8	Host A⇔ Router 9	Host A⇔ Router 10	Host A⇔ Router 11	Host A⇔ Host C3
RateTracer [Mbps]	105.1	153.8	94.6	121.8	98.3	100.5	109.2	109.6	105.3	126.9	14.5	11.6
Pathchar [Mbps]	31.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	7.1	7.1
Nettimer [Mbps]	34.9	41.9	42.0	26.3	27.7	30.0	26.6	31.3	45.6	44.3	4.4	23.5
Actual bottleneck link bandwidth [Mbps]	100	100	100	100	100	100	100	100	100	100	10	10
RTT [ms]	0.794	2.62	2.99	3.28	4.12	4.91	5.00	5.57	5.81	6.54	11.9	9.83

の推定結果はおおむね良好であった。Nettimer はすべてのホップについて計測できたが、推定結果は真値のほぼ半分であった。Pathchar による推定結果はこれらより更に落ちる。特に、Pathchar では Router7 ⇔ Router 8 間のリンク速度が計測できず、それ以降のリンクについても測定が成功したのは 10 回中 2~3 回程度であった。

#### 4.2.3 広域実験結果: A-C3 間

表 7 はホスト A-ルータ C3 間のボトルネックリンク速度の 10 回の推定結果をまとめたものである。ホスト A-C3 間には最終ホップの一つ手前 (Router 10-Router 11) に 10 Mbit/s の ATM 回線があり、これがホスト A-C3 間のボトルネックリンクとなる。各ツールによるホスト A-C3 間のボトルネックリンク速度の推定値を比較すると、Nettimer の推定値 (23.5 Mbit/s) の誤差が最も大きい (真値の倍以上)。一方、RateTracer は真値より若干大きい推定値 (11.6 Mbit/s)、Pathchar は逆に真値より若干小さい推定値 (7.1 Mbit/s) を与える。ATM ヘッダを考慮すると、実質的なボトルネックリンク速度は 9 Mbit/s 程度であり、その意味で、RateTracer より Pathchar の推定値の方が若干正確である。また、

ホスト A-Router 12 間についても、Pathchar の方が RateTracer よりも正確な推定値を与えている。このように、RateTracer は特に ATM 回線のリンク速度を若干大きめに誤推定する傾向が認められる。ただし、Router 11 及びその手前のルータまでの経路については、ホスト A-Router 2 間を除き、RateTracer が最も正確である。

なお、ホスト A-Router 11 間の RTT が、ホスト A-C3 間の RTT を上回っているのは、Router 11 での ICMP 処理に時間がかかっているためと思われる。

## 5. む す び

本論文では、試験パケットを用いて、end-to-end 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 (RateTracer) 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。我々の提案は、Packet-Pair 推定法の単純な改良であり、実装も容易である。また、いくつかの既存ツールに比べて、より優れた推定精度をもつことが確認できた。したがって、提案ツールは、例えばトラブルシューティングの際の有望なツールとして十分期待できると思われる。今後の課題としては、利用する試験パケット量の軽減化、クライアント-サー

バ型ツールとしての実装等がある。

**謝辞** GEMnet での実験を快く御許可頂いた NTT サービスインテグレーション基盤研究所の魚瀬尚郎主幹研究員に深謝申し上げます。また、実験の実施にあたっては、同研究所の井上規昭研究員、NTT アドバンステクノロジーの山田隆主査、田島弘隆主任をはじめ多くの方々のサポートを頂きました。この場を借りて御礼申し上げます。また、有益なコメントを頂いた査読者の方々に感謝致します。

### 文 献

- [1] 間瀬憲一 (編著), マルチメディアネットワークとコミュニケーション品質, 電子情報通信学会, 1998.
- [2] 鶴 正人, 尾家祐二, “インターネット計測における推定技術の新潮流,” 信学技報, IN2003-06, 2003.
- [3] V. Jacobson, “Pathchar,” <ftp://ftp.ee.lbl.gov/pathchar>
- [4] B.A. Mah, “Pchar: A tool for measuring Internet path characteristics,” <http://www.ca.sandia.gov/bmah/Software/pchar>
- [5] K. Lai and M. Baker, “Nettimer: A tool for measuring bottleneck link bandwidth,” Proc. USENIX Symposium on Internet Technologies and Systems, pp.122-133, 2001.
- [6] K. Lai and M. Baker, “Measuring link bandwidths using a deterministic model of packet delay,” ACM SIGCOMM '00, pp.283-194, 2000.
- [7] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, “Bandwidth estimation: Metrics, measurement technique, and tools,” IEEE Network Magazine, vol.17, pp.27-35, 2003.
- [8] A.B. Downey, “Using pathchar to estimate Internet link characteristics,” ACM SIGCOMM '99, pp.241-250, 1999.
- [9] 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,” 信学技報, SSE99-168, IN99-131, 2000.
- [10] 八木敬宏, 塩田茂雄, 檜垣泰彦, 池田宏明, “10Base-T/100Base-TX ベース LAN における Pathchar の特性評価,” 2001 信学ソ大 (通信), B-16-12, 2001.
- [11] P. Pasztor and D. Veitch, “Active probing using packet quarters,” ACM Internet Measurement Workshop 2002, pp.293-305, 2002.
- [12] J.-C. Bolot, “End-to-end packet delay and loss behavior in the Internet,” ACM SIGCOMM '93, pp.289-298, 1993.
- [13] V. Jacobson, “Congestion avoidance and control,” ACM SIGCOMM '88, pp.314-329, 1988.
- [14] S. Keshav “A control-theoretical approach to flow control,” ACM SIGCOMM '91, pp.113-121, 1991.
- [15] V. Paxson “End-to-end Internet packet dynamics,” IEEE/ACM Trans. Netw., pp.277-292, 1999.
- [16] R. Carter and M.E. Crovella, “Measuring bottleneck link speed in packet-switched networks,” Per-

- formance Evaluation, vol.27 & 28, pp.297-318, 1996.
- [17] C. Dovrolis, P. Ramanathan, and D. Moore, “What do packets dispersion techniques measure?,” IEEE INFOCOM, pp.905-914, 2001.
- [18] W.R. Stevens, UNIX Network Programming, Networking APIs: Sockets and XTI Volume 1, Prentice Hall, 1998.
- [19] 八木敬宏, 塩田茂雄, 間瀬憲一, “ボトルネックリンク速度推定ツールの提案と精度検証,” 信学技報, IN2002-199/IA2002-55, 2003.
- [20] “Iperf,” <http://dast.nlanr.net/Projects/Iperf1.1.1/release.html>
- [21] 八木敬宏, ボトルネックリンク速度推定ツールの提案と精度検証, 修士論文, 千葉大学自然科学研究科, 2003.

(平成 16 年 1 月 9 日受付, 5 月 6 日再受付)



八木 敬宏 (正員)

平 14 千葉大・工・都市環境システム卒。  
平 16 同大大学院修士課程了。同年 (株) 日立製作所入社。現在, 同社公共システム事業部勤務。電子政府システムや電子自治体システムに関する研究・開発に従事。平 16 本会情報ネットワーク研究賞受賞。



塩田 茂雄 (正員)

昭 61 早大・理工・物理卒。昭 63 東大大学院修士課程了。同年 NTT 入社。平 13 より, 千葉大・工・助教授。インターネット運用・管理技術, システム性能評価に関する研究・開発に従事。博士 (工学)。平 15 本会ネットワークシステム研究賞。平 16 本会情報ネットワーク研究賞受賞。ACM, IEEE, 日本 OR 学会各会員。



間瀬 憲一 (正員:フェロー)

昭 45 早大・理工・電気通信卒。昭 47 同大大学院修士課程了。同年電電公社武蔵野電気通信研究所入所。以来, 通信網構成法, ダイナミックルーティング, ネットワーク設計法, 通信品質, コンピュータネットワーク, モバイルアドホックネットワーク等の研究に従事。昭 53~54 米国コロンビア大客員研究員。現在, 新潟大学大学院自然科学研究科教授。同工学部附属国際情報通信教育研究センター長。工博。平 5 本会論文賞, 平 9 第 13 回電気通信普及財団テレコムシステム技術賞受賞。著書「マルチメディアネットワークとコミュニケーション品質」(共著), 「無線 LAN とユビキタスネットワーク」(共著) など。