

## 高精度時刻 PC による片道遅延時間によるネットワーク帯域推定手法

北口 善明<sup>†,††,†††\*a)</sup> 町澤 朗彦<sup>††</sup> 箱崎 勝也<sup>†††</sup>  
 中川 晋一<sup>††</sup>

The Network Bandwidth Measurement Based on One-Way Delay Using  
 a High-Precision PC

Yoshiaki KITAGUCHI<sup>†,††,†††\*a)</sup>, Akihiko MACHIZAWA<sup>††</sup>, Katsuya HAKOZAKI<sup>†††</sup>,  
 and Shin-ichi NAKAGAWA<sup>††</sup>

あらまし 近年、インターネットの普及に伴いネットワークの帯域推定が重要となってきた。従来の ICMP による往復遅延時間を利用した手法ではクロストラヒックの影響が小さい長所があるが、ICMP 処理に伴う影響により広帯域ネットワークでは精度が低くなることが知られていた。本論文では、ICMP による往復遅延時間を用いずに UDP による片道遅延時間を使った広帯域回線の帯域推定を行う手法を提案する。この手法と併せて片道遅延時間を高精度に計測するために、外部高精度周波数源により駆動される高精度時刻 PC を用いて、従来の手法との精度比較を実施した。その結果、本手法により精度を 1 けたから 2 けた改善することが可能であった。  
 キーワード アクティブネットワーク計測、帯域推定、有効帯域、高精度時刻 PC

### 1. ま え が き

インターネットの普及に伴い、回線の有効利用がますます重要となってきた。ネットワークのスループットが期待されるビットレートに達しない場合、まずその回線の帯域を調査する必要がある。

帯域推定には、様々な手法が使われている。インターネットでは、ADSL 等のブロードバンド計測サイトで、http あるいは ftp などのアプリケーションによるファイル転送時間を測定しているが、TCP を用いているため Window Size が適切に設定されないと、往復遅延時間の大きいコネクションではスループットが上がらない。このように、実際に何バイトのデータを伝送するのにどのくらいの時間がかかるか、あるいは、

単位時間に何バイトのデータを伝送できるか、を測定する netperf [4], iperf [5] などのソフトウェアツールが開発されており、TCP だけではなく遅延時間による性能劣化のない UDP による測定が可能である。しかし、これらの実データを伝送する手法は、バックグラウンドトラヒックやふくそうから影響を受けるだけではなく、回線の空き帯域をすべて埋めつくしてしまい、他のユーザに甚大な影響を与えてしまうため使用には注意が必要である。

そこで、大量のデータを送ることなく、少数パケットで帯域を推定する手法が提案されてきている。Packet pair 計測手法 [2], [3] を用いて、パケット間隔が帯域の逆数に比例することを利用する Cprobe [6] が提案されているが、クロストラヒック（経由ルータ・スイッチでのパケットの合流や離散）により、精度が悪くなる [7]。また、高速なネットワークインタフェース（例えばギガビットイーサネット：GbE）では、パケット受信時の割込み負荷を軽減するために、数パケットまとめて受信する機能が用いられている。そのためパケット間隔を利用することができない場合が存在しており、Packet pair 計測手法の問題点となっている。一方、pathchar [8] と clink [9] はパケット間隔を用いずに、One-Packet 計測手法 [1] に基づいて、パ

<sup>†</sup> 通信・放送機構（現情報通信研究機構），東京都  
 Telecommunications Advancement Organization of Japan,  
 2-31-19 Shiba, Minato-ku, Tokyo, 105-0014 Japan  
<sup>††</sup> 独立行政法人通信総合研究所（現情報通信研究機構），小金井市  
 Communications Research Laboratory, 4-2-1 Nukui-kitamachi,  
 Koganei-shi, 184-8795 Japan  
<sup>†††</sup> 電気通信大学，調布市  
 University of Electro-Communications, 1-5-1 Chofugaoka,  
 Chofu-shi, 182-8585 Japan  
 \* 現在，インテック・ウェブ・アンド・ゲノム・インフォマティクス  
 株式会社  
 a) E-mail: kitaguchi\_yoshiaki@webgen.co.jp

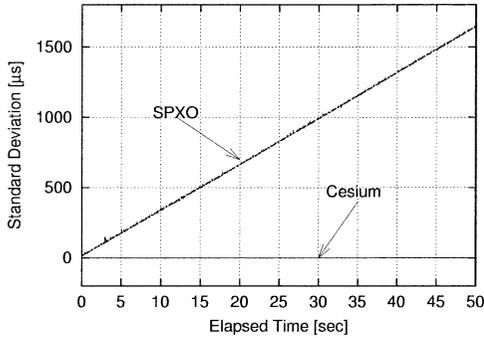


図 1 時刻スキュー値の変化  
Fig. 1 Change of a time skew.

ケットサイズの異なる ICMP パケットの往復遅延時間 (RTT: Round Trip Time) の差異から、帯域推定を行っている。しかし、ICMP 応答処理時間がばらつくため、高速回線では精度が悪くなる [17]。また、スイッチングハブなど、ストアアンドフォワード処理を行い、しかも TTL (Time To Live) 値を変化させない装置 (invisible device) を通過する場合には、帯域が狭く推定されてしまう [10]。

本論文では、クロストラヒックやネットワークインタフェースでのまとめ処理の影響を受けない One-Packet 計測手法の有意性に着目し、ICMP パケットの RTT を用いる手法 (以下 ICMP 手法) の欠点を解決する。その手法として、UDP パケットの片道遅延時間を利用する新しい帯域推定手法 (以下 UDP 手法) を提案する。本手法では、従来の One-Packet 計測手法と同様に、伝送遅延時間がパケットサイズに比例することを利用するが、ICMP 応答は用いずに、UDP パケットの片道遅延時間のみを利用する。ここで片道遅延時間を計測するためには、パケットを送受信する 2 台の PC の時刻スキューが重要になる。通常の PC は温度補償されていない水晶 (SPXO: Simple Packaged Crystal Oscillator) により駆動されているため、動作周波数の変動により、PC 内部の時刻情報には誤差が累積し、計測中にずれが生じる可能性がある。図 1 にセシウム原子時計駆動による PC を基準とした場合の時刻比較を行った結果を示す。計測開始時の基準 PC との時刻差をプロットしたもので SPXO 駆動 PC とセシウム駆動 PC の両者の値を表示している。この図から分かるように、SPXO 駆動の PC では時間の経過とともにドリフトしていく様子が見え、1s 間で約  $33 \mu\text{s}$  も変化していることが分かる。この変動

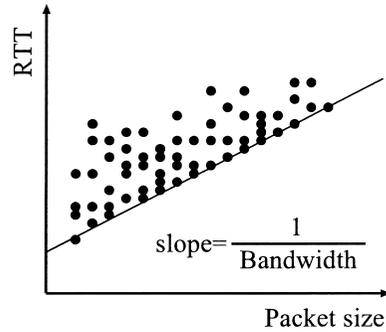


図 2 One-Packet 計測手法での帯域推定方法  
Fig. 2 Concept of One-Packet model.

を NTP による時刻同期 [11] や時刻補償 [12], [13] などで抑制する研究も進められているが、GbE では 1500 バイトパケットでも通過時間は  $12 \mu\text{s}$  にすぎず、ソフトウェア処理では補正しきれない。そこで、我々が先に開発した高精度時刻 PC [14] ~ [16] を用いる。高精度時刻 PC は、セシウム原子時計などの高精度の周波数源から出力される 10 MHz 信号により駆動されるため、計測期間によらず極めて精度の高い時刻スキューを維持することができる。本論文では、UDP 手法と高精度時刻 PC を組み合わせることにより、帯域推定精度が向上することを確認した。

## 2. 提案手法

本論文にて提案する手法では One-Packet 計測手法をもとにしている。この One-Packet 計測手法は Bellovin [1] により提唱された手法で、様々な送信パケットサイズでの転送時間の計測を複数回実施し、得られた計測値から各パケットサイズでの最小転送時間を抽出し、この最小転送時間から回帰直線を求め、通過したリンクの帯域を推定する。帯域は求めた回帰直線の傾きの逆数から得ることができる (図 2 参照)。最小転送時間を用いるのは、計測時に他の要因で遅延が発生した場合の影響を抑えるためである。

パケット転送時間の計測には主に ICMP エラーメッセージを利用した実装が多く、この方法では計測対向地点に特別なソフトウェアを用意する必要がないという利点がある。この ICMP を利用した場合、計測した RTT には送信パケットの送受信時間 ( $T_S$ )、計測対象にて返答処理が行われる時間 ( $T_I$ )、返信パケットの送受信時間 ( $T_R$ ) が含まれる。

$$RTT = T_S + T_I + T_R \quad (1)$$

$T_S$  は式 (2) のように、一定のネットワーク伝搬遅延 (定数  $C$ ) と送信パケットサイズ ( $P_S$ ) にて変化する時間が含まれ、また  $T_R$  には ICMP エラーパケットのサイズ ( $P_R$ ) による時間が含まれる。したがって、リンクの帯域を  $B$  とすると以下の式で表すことができる。

$$T_S = \frac{P_S}{B} + C_S \quad (2)$$

$$T_R = \frac{P_R}{B} + C_R \quad (3)$$

式 (2) と式 (3) から式 (1) は次のようになる。

$$RTT = \frac{P_S}{B} + \frac{P_R}{B} + T_I + C \quad (C = C_S + C_R) \quad (4)$$

この式 (4) を  $P_S$  にて偏微分すると

$$\frac{\partial RTT}{\partial P_S} = \frac{1}{B} + \frac{\partial T_I}{\partial P_S} + \frac{\partial T_R}{\partial P_S} \quad (5)$$

となり、ICMP 処理部分の影響と ICMP エラーパケットの実装の影響を受け、一次関数に近似できない場合が存在することが分かる。また、計測対象の機器の影響で高速なネットワークでは精度が低くなることが知られている [17]。

そこで、上記の問題を回避する方法として、UDP 手法を提案する。UDP パケットに送信時のタイムスタンプを記述し、様々なパケットサイズにて送信する。受信側では受信時のタイムスタンプを取得し、パケットサイズと送受信のタイムスタンプの差を記録する。片道遅延時間を正確に計測するためには計測を行う 2 台の PC の時刻が合っている必要がある。しかし、One-Packet 計測手法ではパケットサイズの変化に対する遅延時間の変化が求めれば良いため絶対時刻の同期は必要ではなく、両 PC の時刻スキューが重要になる。我々の開発した高精度時刻 PC を用いることで時刻スキューの精度を確保している。

この UDP 手法に用いるツールを独自に実装した。UDP 手法では利用するタイムスタンプに絶対時刻は必要でないため、ソフトウェア処理の小さい CPU クロックカウンタを用いることとした。時刻情報への変換は CPU 周波数を用いて行っている。Pentium 以降の Intel CPU では、PC の起動時からの CPU のクロック数をカウントする 64 ビットのカウンタ (TSC: Time Stamp Counter) をもっており、RDTSC 命令で簡単に読み出すことが可能となっている。この RDTSC 命令を使うメリットとしては関数自身の時間コストが低いことが挙げられる。通常の計測ツールに利用される

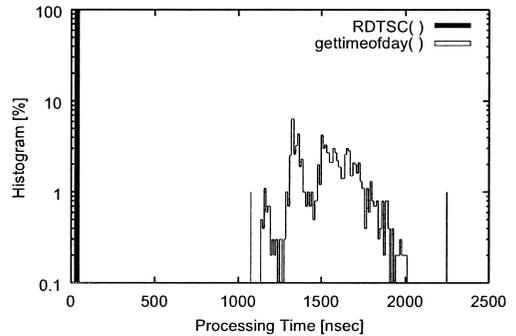


図 3 RDTSC 命令と gettimeofday 関数の時刻コスト  
Fig. 3 Time cost of RDTSC and gettimeofday.

表 1 計測に使用した PC の仕様

Table 1 Specification of PC used for measurement.

OS	RedHat Linux 7.2
Kernel	Linux-2.4.3 + PPSkit
Motherboard	MSI MS-6351 (Ver5)
CPU	Pentium III 996.68 MHz
Memory	512MB
PCI BUS	PCI2.0 32bit 33 MHz
NIC	Intel Pro/1000 XT Server Adapter
NIC driver	e1000 version 4.3.2

絶対時間取得関数: gettimeofday() と時刻コストの比較を行うと、図 3 のような結果を得ることができる。この図は各時刻取得関数の時間コストをそれぞれ 1000 回ずつ計測した場合の分布を表している。ここから gettimeofday() では数  $\mu s$  での変動が観測されることに対して RDTSC 命令は時刻コストも約 50 ns と小さく一定の値を示していることが分かる。本論文で計測対象とする GbE では前章でも述べたようにパケット通過時間が  $\mu s$  オーダと非常に小さいため、時刻取得関数の影響を省く目的から RDTSC を利用することとした。

次章にて、ICMP 手法との比較、計測 PC に高精度時刻 PC と水晶駆動 PC を用いた場合の比較を行う。

### 3. ネットワーク帯域計測結果

提案する UDP 手法の有効性を検証するために「ICMP 手法と UDP 手法の比較」と「高精度時刻 PC を用いる必要性」の 2 種の比較実験を行う。この計測実験に用いた PC のスペックを表 1 にまとめる。対象とするネットワークはファストイーサネット (以下 FE) と GbE (1000 BaseTX) で、それぞれのインタフェースをクロスケーブルで結び、クロストラヒックやスイッチ・ルータ等による影響のない状態で各帯

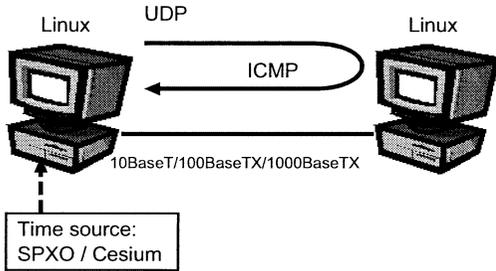


図 4 ICMP 手法によるネットワーク計測時の構成  
Fig. 4 Configuration at the time of ICMP-based network measurement.

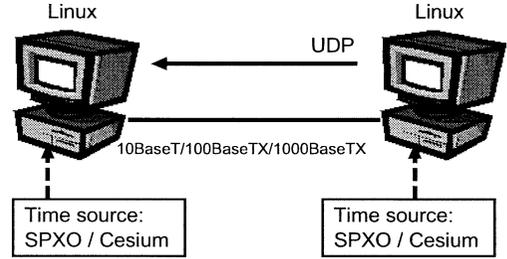


図 5 UDP 手法によるネットワーク計測時の構成  
Fig. 5 Configuration at the time of UDP One-way delay-based network measurement.

域計測の比較を実施する．メディアの変更は使用する GbE 用 NIC のドライバにて実施している．

ICMP 手法と UDP 手法の比較では，PC にセシウム原子時計駆動の高精度時刻 PC (Cesium PC) を利用し，内部クロックの影響がでない環境を用いて計測を行い，次の実験では一般的な水晶駆動 PC (SPXO PC) と高精度時刻 PC との計測精度比較を UDP 手法にて行い，高精度時刻 PC の優位性を示す．

また，計測で求める帯域の推定誤差には，計測に伴う測定データの分散による測定誤差と推定手法及びシステムに依存する系統誤差が含まれるが，この章では測定誤差の精度に焦点をあてて精度の比較を行うこととする．

### 3.1 ICMP 手法と UDP 手法の比較

本論文にて提案する手法の比較手法である ICMP 手法には，一般的に帯域計測に使われている clink を用いた．前章で議論したように，時刻取得関数の影響を小さくするために RDTSC 関数を用いることとし，UDP 手法に用いるツールと同様に clink にて時刻取得関数の置換えを行い，比較計測を実施することとした．計測には，16 バイト刻みのパケットにて各々の試行回数を 10 回として 10 ms 間隔で送信し，16 バイトから 1440 バイトまでの計 900 パケットを用いて，ICMP 手法，UDP 手法の双方で実施した（計測期間は約 10 s）．ICMP 手法での機器構成図を図 4 に，UDP 手法での機器構成図を図 5 にそれぞれ示す．UDP 手法では両端の PC それぞれに独立した時刻源を供給している．

表 2 に両者の計測結果をまとめる．1 列目に UDP 手法の結果，2 列目に ICMP 手法の結果を，また各データは上段が推定帯域，下段が推定帯域の確かさを示す測定誤差を表す．この測定誤差は各パケットサイズの最小値より回帰直線を求めた際の誤差として

表 2 ICMP 手法と UDP 手法の計測結果の比較  
上：帯域，下：標準誤差

Table 2 Comparison of the measurement result of ICMP and UDP One-way delay.  
Up: bandwidth, Down: standard error.

	One-way UDP	ICMP (all)	ICMP (over 600)
FE	79.870 Mbit/s 0.0565%	64 Mbit/s 1.778%	82.9 Mbit/s 0.286%
GbE	279.6 Mbit/s 0.181%	208 Mbit/s 2.987%	370 Mbit/s 6.434%

いる．ICMP 手法での推定帯域が UDP 手法の場合に比べ低い値になっているため，FE，GbE それぞれの観測データをプロットし調査した．FE と GbE における ICMP 手法でのパケットサイズと RTT の相関図を図 6，図 7 にそれぞれ示す．900 個の計測点をすべてプロットし，濃い黒点が回帰直線を求めるために用いる，各パケットサイズにおける最小 RTT のデータを示している．これらの図から，パケットサイズが 600 バイトより小さい点にて RTT の値が大きく変化していることが分かる．この原因を調査したところ，RFC1812 [18] の実装の違いによるものであることが判明した．ICMP エラーパケットのサイズは，IP ヘッダ (20 バイト) + ICMP ヘッダ (8 バイト) + エラーとなったパケットの IP ヘッダを含むパケットデータ (28 バイト) の合計 56 バイトと定義されている [19]．しかし，RFC1812 においてエラーとなったパケットデータの情報を最低 MTU サイズである 576 バイトを上限としてできるだけ付加するべき (SHOULD) としており，この実装を Linux では採用しているためであった．そのため式 (5) の右辺第 3 項の影響を無視できなくなっていた．

この影響を省くために 600 バイト以上のパケットの計測データのみを使い推定帯域を求め，その結果を表 2 の 3 列目に追記する．600 バイト未満のパケット

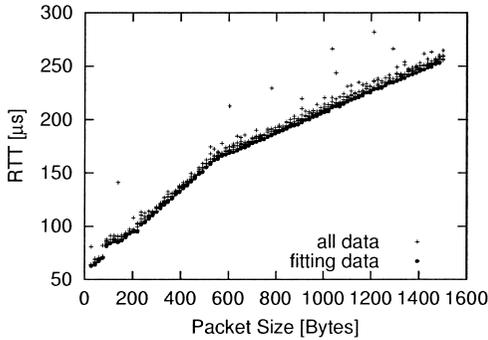


図 6 ICMP 手法でのパケットサイズと RTT の関係 (FE)  
Fig. 6 Correlation of packet size and RTT on ICMP measurement (FE).

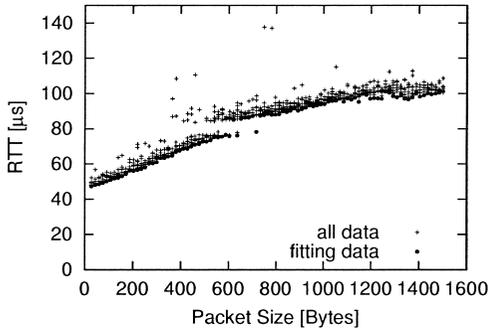


図 7 ICMP 手法でのパケットサイズと RTT の関係 (GbE)  
Fig. 7 Correlation of packet size and RTT on ICMP measurement (GbE).

データを切り捨てることで、FE での推定帯域と標準誤差の値は改善されるがそれでも UDP 手法での標準誤差より小さくなることはなく、また GbE では標準誤差が逆に大きくなる結果であった。GbE での結果は計測点の減少の影響が大きいと考えられる。

以上のように、UDP 手法にて推定帯域の計測精度が測定誤差の比較によって ICMP 手法より向上していることが分かり、FE では 2 けた、GbE では 1 けたの精度改善が見られる。また、各推定帯域はメディアの物理速度より過小評価していることに対しては次章にて詳細に検討する。

### 3.2 高精度時刻 PC を用いる必要性

次に高精度時刻 PC の優位性を一般的な水晶駆動 PC との計測比較にて明らかにする。UDP 手法のみを用い、計測 PC に使用する両端の時刻源を変更してそれぞれ実施した。計測は 3.1 と同じ方法で回帰直線を求め、帯域とその標準誤差を求めた。表 3 の 1 列目

表 3 高精度時刻 PC と水晶駆動 PC の比較  
上：帯域，下：標準誤差

Table 3 Comparison of the measurement result of Cesium PC and SPXO PC.  
Up: bandwidth, Down: standard error.

	Cesium	SPXO
FE (UDP)	79.870 Mbit/s	82.8 Mbit/s
	0.0565%	0.375%
GbE (UDP)	279.5 Mbit/s	360 Mbit/s
	0.181%	5.755%

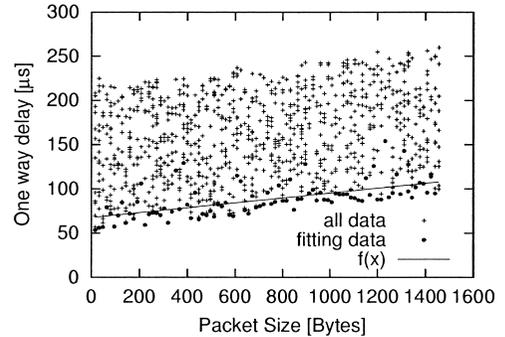


図 8 水晶駆動 PC でのパケットサイズと UDP 片道遅延時間の関係 (GbE)  
Fig. 8 Correlation of packet size and UDP One-way delay on the SPXO PC (GbE).

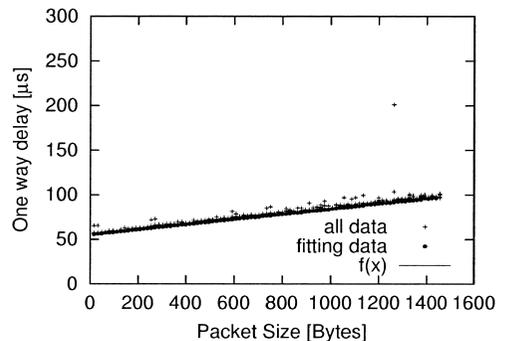


図 9 高精度時刻 PC でのパケットサイズと UDP 片道遅延時間の関係 (GbE)  
Fig. 9 Correlation of packet size and UDP One-way delay on the Cesium PC (GbE).

に高精度時刻 PC、2 列目に水晶駆動 PC の計測結果をそれぞれ示す。

UDP 手法による計測では、両端の PC の時刻スキューが重要で両者の PC の時刻揺らぎが大きく影響する。GbE の計測での水晶駆動 PC と高精度時刻 PC での計測データのプロットを図 8 及び図 9 にそれぞれ示す。この図から分かるように水晶駆動 PC の場合、

各パケットサイズでの計測結果の分布幅が約 170  $\mu\text{s}$  にもなり、その影響で最小片道遅延時間でも約 50  $\mu\text{s}$  の分布幅が発生していることが分かる。これに比べ、高精度時刻 PC では、変動幅が約 20  $\mu\text{s}$  程度と水晶駆動 PC と比べ安定した値を得ることができており、最小値の変動も小さくなっている。この原因としては図 1 に示した経過時間による時刻スキューのドリフトの影響であると考えることができる。水晶駆動 PC では両端の PC それぞれが約 1  $\mu\text{s}/\text{s}$  のドリフト値をもっていることによる影響により、計測期間約 10s の間に時刻差が変化し、計測結果の誤差が高精度時刻 PC に比べて大きくなっていることが分かる。対して精度時刻 PC では、時刻スキューに変化がないため水晶駆動 PC に比べて小さい分布幅に収まっている。また、計測時のパケットサイズはランダムに選んで送信しているため、この分布幅が一様に見える。

これらの観測結果から、高精度時刻 PC を用いることで推定帯域の精度である測定誤差の値が、水晶駆動 PC に比べて 1 けた以上改善されていることが分かる。また、計測結果より、PC 内部時計の精度が計測結果に及ぼす影響が、高速なネットワークほどこの影響は大きくなっていることから、今後利用が一般的となる 10 ギガイーサネット等での計測にて本手法が重要になるといえる。

#### 4. 考 察

3. における計測結果から、ICMP 手法よりも UDP 手法が帯域推定の精度が計測誤差が小さいことから優れており、高精度時刻 PC においてその計測精度向上が可能であることが分かった。高精度時刻 PC と UDP 手法の併用において図 2 に示すような計測値のばらつきを抑えることが可能であったことから、ICMP 手法による影響と利用する PC の時刻精度による計測値のばらつきの要因であると推定できた。しかし、計測で得られた推定帯域を見ると GbE での計測では 400 Mbit/s 以下と低い値でしかないことが分かる。この結果の原因をここで考察する。

UDP 手法での計測に含まれる時間 ( $T_S$ ) には、回線の伝送遅延だけではなく、送信側 PC でタイムスタンプを打ってから実際に回線上にパケットが送出されるまでの時間  $T_\delta$  及びパケットが受信 PC の NIC に取込まれてからタイムスタンプが呼出されるまでの時間  $T_\rho$  も含んでおり、式 (2) は正確には、

$$T_S = T_\delta + \frac{P_S}{B} + T_\rho + C_S \quad (6)$$

となる。ここで、両辺を  $P_S$  で偏微分すると、

$$\frac{\partial T_S}{\partial P_S} = \frac{1}{B} + \frac{\partial(T_\delta + T_\rho)}{\partial P_S} \quad (7)$$

$$\frac{1}{B} = \frac{\partial T_S}{\partial P_S} - \frac{\partial(T_\delta + T_\rho)}{\partial P_S} \quad (8)$$

となる。ここで、式 (8) の右辺の  $\frac{\partial T_S}{\partial P_S}$  は測定データの傾きであり、 $\frac{\partial(T_\delta + T_\rho)}{\partial P_S}$  は使用した PC に依存し、1 ビット当りの遅延時間増加率を表している。この遅延時間増加率を、

$$\alpha = \frac{\partial(T_\delta + T_\rho)}{\partial P_S} \quad (9)$$

とおき、式 (8) より、

$$\alpha = \frac{\partial T_S}{\partial P_S} - \frac{1}{B} \quad (10)$$

となる。この式 (10) を用い、高精度時刻 PC の UDP 片道遅延計測結果を用いて  $\alpha$  を計算すると FE ( $B = 100 \text{ Mbit/s}$ ) で 2.52 (0.0113%)、GbE ( $B = 1000 \text{ Mbit/s}$ ) で 2.57 (0.130%) とそれぞれ求め、ほぼ一定の値であることが分かる (括弧内数値は計測誤差)。この値は NIC に依存するもの (OS におけるデバイスドライバの実装も含む) であると考え、他の NIC (Netgear GA620T と PCI GN-1000TE) を使用して UDP 片道遅延を用いた高精度時刻 PC での計測を実施したところ表 4 のような結果を得た。それぞれの NIC のデバイスドライバには acenic と ns83820 を用いている。この結果から同様に  $\alpha$  を計算した結果を表 5 にまとめると、NIC に依存して値は異なるが、同一 NIC では利用する伝送速度によらず、ほぼ一定の値が得られた。したがって、NIC ごとの影響は、計測に使用する PC の遅延増加率を事前に測定することにより補償可能であることが分かる。ま

表 4 他の NIC を用いた場合の計測結果  
上: 帯域, 下: 標準誤差

Table 4 Measurement result on other NIC.

Up: bandwidth, Down: standard error.

	GA620T	GN-1000TE
FE	79 Mbit/s 0.4042%	77 Mbit/s 0.1375%
GbE	271 Mbit/s 1.388%	258 Mbit/s 0.4153%

表 5 各 NIC における遅延時間増加率  $\alpha$  [nsec/bit]  
 上: 遅延時間増加率, 下: 計測誤差  
 Table 5 Rate of delay increasing on each NIC.  
 Up: bandwidth, Down: error.

	Pro/1000 XT (e1000)	GA620T (acenic)	GN-1000TE (ns83820)	ICMP (e1000)
FE	2.52 0.0113%	2.65 0.0839%	2.84 0.0302%	2.09 0.0609%
GbE	2.57 0.130%	2.69 1.01%	2.88 0.308%	1.71 4.2145%

表 6 遅延時間増加率  $\alpha$  を用いた推定帯域の比較  
 Table 6 Comparison of the bandwidth using by rate  
 of delay increasing.

	UDP (Cesium:e1000)	ICMP (SPXO:e1000)
FE	100.3 Mbit/s	98.0 Mbit/s
GbE	972.3 Mbit/s	1204 Mbit/s

た, 従来の ICMP 手法との比較を行うために, 水晶駆動 PC にて ICMP 手法による計測を行った場合の結果から同様に  $\alpha$  の値を算出したものも表 5 に追記する. ICMP 手法では測定誤差の影響から  $\alpha$  の値の精度が GbE にて幅が大きいことが分かる.

表 6 に,  $\alpha$  の値を用い計測データの補償を行った場合の推定帯域を示す. ただし,  $\alpha$  として, 表 5 の各 NIC の FE と GbE の値の平均値を用い, ICMP 手法と UDP 手法の比較を行っている. この結果から,  $\alpha$  の値を用いることで計測帯域は真値に近い値として求まることが分かり, したがって, UDP 手法の方が系統誤差の面でも ICMP 手法より優れていることが示された.

## 5. む す び

本論文では, 高精度時刻 PC を用いて計測した片道遅延時間により, ネットワーク帯域を推定する新しい手法を提案した. 従来の ICMP を用いた計測手法では計測対象の ICMP の実装による影響が大きく, 高速ネットワークにおいてその精度が十分ではないことを示し, 高精度時刻 PC を用いた UDP による片道遅延時間による帯域計測方法がその問題を回避し, より精度良く計測できることを示した. 提案手法は, 従来の水晶駆動 PC で ICMP を用いた場合に比べ, FE の回線の推定精度が 2 けた程度改善し, GbE の回線でも 1 けたの改善が確認された. 更に, これまで知られていなかったネットワークインタフェースの機種依存特

性を明らかにした.

なお, One-Packet 計測手法の弱点である invisible device の問題は, 本手法でも未解決である. また, ICMP 手法では, リンクの帯域を逐次に推定することができるが, 本手法では逐次推定することができず, 更に改良する必要がある. 更に, 本論文での計測実験を踏まえて, 将来の 10 Gbit/s 以上の高速ネットワーク計測に必要な計測手法及び計測機器の精度の見積りを進めることが重要であると考えている. また今後の課題として, 高精度時刻 PC と UDP 手法の組合せにて発生する計測誤差の原因を追求し, より推定誤差の少ない手法の確立を考えている.

謝辞 本研究をまとめるにあたり, ギガビットネットワーク研究開発プロジェクトの方々には多くの助言を頂き, 特にネットワークの計測手法に関して多くのコメントを頂いた九州工業大学の鶴正氏に深く感謝する.

## 文 献

- [1] S.M. Bellovin, "A best-case network performance model," <http://www.research.att.com/~smb/papers/netmeas.ps>, Feb. 1992.
- [2] J.-C. Bolot, "End-to-end packet delay and loss behavior in the Internet," Proc. ACM SIGCOMM '93, pp.289-298, Sept. 1993.
- [3] R.L. Cartar and M.E. Crovella, "Measuring bottleneck link speed in packet-switched networks," TR-96-006, Boston University Computer Science Department, March 1996.
- [4] R. Jones, "The public netperf homepage," <http://www.netperf.org/>, 1993.
- [5] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs, "iperf: The TCP/UDP bandwidth measurement tool," <http://dast.nlanr.net/Projects/Iperf/>, 1999.
- [6] B. Carter, "Bprobe and cprobe," <http://cs-people.bu.edu/carter/tools/Tools.html>, 1996.
- [7] R.S. Prasad, "BW-meter project," <http://www.pathrate.org/>, 2002.
- [8] V. Jacobson, "Pathchar," <http://www.caida.org/tools/utilities/others/pathchar/>, 1997.
- [9] A.B. Downey, "Clink," <http://rocky.wellesley.edu/downey/clink/>, 1999.
- [10] R.S. Prasad, C. Dovrolis, and B.A. Mah, "The effect of layer-2 switches on pathchar-like tools," Proc. ACM Internet Measurement Workshop 2002, pp.321-322, Nov. 2002.
- [11] D.L. Mills, "Network time protocol (Version 3) specification, implementation," RFC 1305, March 1992.
- [12] V. Paxson, "End-to-end Internet packet dynamics,"

- Proc. ACM SIGCOMM '97, pp.139–152, Sept. 1997.
- [13] L. Zhang, Z. Liu, and C.H. Xia, “Clock synchronization algorithms for network measurements,” Proc. IEEE Infocom 2002, pp.160–169, June 2002.
- [14] H. Okazawa, A. Machizawa, S. Nakagawa, Y. Kitaguchi, T. Asami, and A. Ito, “Advanced NTP synchronization device for Internet monitoring tools,” Proc. INET2001.  
[http://www.isoc.org/inet2001/CD\\_proceedings/T42/inet2001.html](http://www.isoc.org/inet2001/CD_proceedings/T42/inet2001.html), June 2001.
- [15] Y. Kitaguchi, H. Okazawa, S. Shinomiya, Y. Kidawara, K. Hakozaiki, and S. Nakagawa, “Development of a high-accurate time server for measurements of the Internet,” Lecture Note in Computer Science 2344, pp.351–358, Jan. 2002.
- [16] 町澤朗彦, 北口善明, 岡沢治夫, 中川晋一, “ネットワークを用いた CPU 動作周波数期間安定度の精密計測” DICOMO2002 シンポジウム論文集, pp.563–566, July 2002.
- [17] S. Savage, “Sting: A TCP-based network measurement tool,” Proc. USENIX Symposium on Internet Technologies and Systems (USITS '99), pp.71–79, Oct. 1999.
- [18] F. Baker, “Requirements for IP version 4 routers,” RFC 1812, June 1995.
- [19] J. Postel, “Internet control message protocol,” RFC 792, Sept. 1981.

(平成 15 年 5 月 27 日受付, 12 月 8 日再受付)



北口 善明 (正員)

平 7 新潟大・理・物理卒。平 9 同大学院自然科学研究科修士課程了。同年(株)インテック入社。平 12 インテック・ウェブ・アンド・ゲノム・インフォマティクス(株)に転籍, 現在に至る。平 12~16 通信・放送機構(現情報通信研究機構)研究員。主として, ネットワーク運用と計測技術, IPv6 ネットワークの研究開発に従事。情報処理学会会員。



町澤 朗彦 (正員)

昭 59 上智大・理工・電気電子卒。同年郵政省電波研究所(現情報通信研究機構)入所。平 6 科学技術庁に出向し, IMnet 立上げに参与。平 8~11 Univ. Canterbury 客員研究員。平 15 JGN2 立上げに参与。主として, 画像の高効率符号化, 視覚情報処理, 計算機ネットワークの研究に従事。日本認知科学会会員。



箱崎 勝也 (正員)

昭 38 九大・工・電子工卒。同年日本電気(株)入社。中央研究所, ソフトウェア開発グループにおいてシステム性能評価, コンピュータアーキテクチャ, OS, ネットワークの相互接続性等の研究開発に従事。平 6 電気通信大学大学院情報システム学研究科教授。主として, 分散システム技術, マルチメディア応用システム, モバイルコミュニケーションシステムなどの研究に従事。博士(工学)。情報処理学会, IEEE, ACM 各会員。



中川 晋一

昭 63 滋賀医大・医・医学卒。同年(財)近江兄弟社ヴォーリス記念病院医員。平 8 京大大学院医学研究科博士課程了, 同年国立がんセンター研究所がん情報研究部研究員, がん臨床情報研究室長, 平 10 郵政省通信総合研究所(現情報通信研究機構)主任研究員。第 1 研究チームリーダー, 情報通信部次世代インターネットグループリーダーを経て平 14 より情報通信研究機構主任研究員, 平 11 より北陸先端科学技術大学院大学客員助教授。主として情報通信医学, 情報疫学, 次世代インターネット技術に従事。博士(医学)。情報処理学会, 日本循環器内科学会, 日本衛生学会, 日本内科学会, 日本疫学学会各会員。