

通過型高精度 UDP タイムスタンプの開発

町澤 朗彦[†] 鳥山 裕史[†] 岩間 司[†] 金子 明弘[†]

[†] 情報通信研究機構 〒184-8795 東京都小金井市貫井北町 4-2-1

E-mail: †machi@nict.go.jp

あらまし 片方向遅延はネットワークパフォーマンスを示す重要な指標であるが、ネットワークは非均質であり、バスを構成する区間毎の片方向遅延を計測できるシステムが必要である。本論文では、ネットワーク上の任意の位置に挿入し、通過する UDP パケットに高精度なタイムスタンプを逐次挿入する、全く新しい専用ハードウェア (PUTS: cascadable Passing through precision UDP Time-Stamping device) を開発したので報告する。本装置を用いることにより、プローブとデータ収集を同時に行えるため、複数区間毎の片方向遅延時間を、リアルタイムかつ容易に計測することが可能である。PUTS のタイムスタンプは高精度 (解像度 4 ナノ秒) であり、10GbE でもワイヤレートを実現する。

キーワード 高精度タイムスタンプ, 区分的片方向遅延, アクティブ計測, カットスルー, FPGA

Development of a Cascadable Passing Through Precision UDP Time-Stamping Device

Akihiko MACHIZAWA[†], Hiroshi TORIYAMA[†], Tsukasa IWAMA[†], and Akihiro KANEKO[†]

[†] National Institute of Information and Communications Technology, 4-2-1 Nukui-Kitamachi, Koganei, Tokyo 184-8795 Japan

E-mail: †machi@nict.go.jp

Abstract One-Way delay is a key metric in data network performance. Piecewise measurement is required because network is heterogeneous. This article presents a revolutionary dedicated hardware time-stamper, PUTS (cascadable Passing through precision UDP Time-Stamping device), which is inserted into network and overwrites UDP payload data with timestamps. It makes possible to simultaneous probing and data gathering for realtime measurement. The resolution of PUTS time-stamp is 4 nsec and PUTS can handle wire speed on 10GbE.

Key words precision time-stamp, piecewise one-way delay, active measurement, cut through, FPGA

1. はじめに

インターネットの利用が急速に広がっているが、遅延時間はネットワークの状態を反映しており [1] [2]、多くのネットワークモニタリングプロジェクトで遅延時間が計測されている [3]。遅延時間を用いて、インターネットのパフォーマンス推定 [4]、TCP の輻輳制御 [5] [6] あるいは帯域推定 [7] [8] などへの応用も進んでいる。さて、遅延時間には、往復遅延を用いる場合と片方向遅延を用いる場合があるが、遅延は非対称な場合が多いため、片方向遅延が有効であり [9] [10]、現在、IETF OWAMP (One-Way Active Measurement Protocol) [11] の標準化が進められている。また、ネットワークは、様々な回線や接続装置によって構成されており、クロストラフィックも区間毎に異なるなど、非均質となっている。一般に、ユーザはバス途中のルー

タ等にはアクセス権限を有していないため、End-to-End に計測するが、ネットワークの非均質構造を明かにするために、最近、ネットワークトモグラフィとして、End-to-End 計測からネットワーク内部の状態を推定する試みも始まっている [12]。しかし、様々な仮定を必要としており、ホップ・バイ・ホップなデータと併せて精度を高める必要があるであろう。

従来、バス途中の遅延時間を測るには、DAG project [13] や IP Meter [14] などのパケットキャプチャ装置が用いられてきたが、キャプチャ方式では、リアルタイムに遅延時間を計測することは原理的に不可能である。つまり、別途、キャプチャ時のタイムスタンプを集約しなければならない。また、キャプチャデータ集約に伴う、セキュリティ、ポリシー、作業量などの問題を抱えている [15]。

本論文では、ネットワーク上の任意の位置に挿入し、通過する

UDP パケットに高精度なタイムスタンプを逐次挿入する全く新しいシステム (PUTS: cascadable Passing through precision UDP Time-Stamper) を開発したので報告する。PUTS を用いることにより、プローブとデータ収集を同時に行えるため、複数区間毎の片方向遅延時間を、簡便にかつリアルタイムにアクティブ計測することが可能である。なお、1つのパケットに最大 183 個のタイムスタンプを挿入することが可能である (パケットサイズ 1500 バイトの場合)。また、PUTS のタイムスタンプは、ネットワークの広帯域化に対応するため、極めて高精度であり、解像度は 4 ナノ秒、安定度は 10^{-12} (外部周波数源としてルビジウム原子時計を用いた場合) となっている。

一方、PUTS はネットワークに挿入して使用するため、系への影響を極力小さくする必要があり、FPGA によるカットスルー、タイムスタンプのペイロードへの書き込み、更にチェックサム補償方式により、短時間かつ一定値となる通過遅延 (88 ナノ秒) を実現している。また、系への影響を小さくすると共に、セキュリティの問題も有さないため、設置が容易である。

2. 関連研究

2.1 遅延計測

現在進められているネットワークモニタリングプロジェクトでは、Surveyor [16]、RIPE TTM (Test Traffic Measurements) [17] および SATURN [18] で片方向遅延、ANEMOS [19] と NCS (Network Characterization Service) [20] は RTT を計測しているが、全て PC によるアクティブ計測である。

しかし、PC によりソフトウェア的に遅延時間を計測する場合、システム能力の制限によって広帯域では精度が低下し [21]、GbE など広帯域で利用されている”まどめ割込み”も精度を低下させるため [22]、広帯域ネットワークを高精度に測定することはできない。しかし、タイムスケールが異なると、新しい現象が発見されることもあり [23]、より高精度な計測システムが必要である。

そこで、DAG [13] [24] や IP メータ [14] などの専用ハードウェアによるパケットキャプチャを用いた遅延計測システムが提案され [25] [26]、Papagiannaki は DAG を用いて、SPRINT バックボーンを構成する、ある 1 台のルータの通過遅延を計測し、キューイング処理にともなう遅延特性を明かにした [27]。しかし、キャプチャ方式では、2 地点のキャプチャデータを集約する必要があるため、リアルタイムに遅延時間を得ることはできず、集約作業量も大きい。また、ストア・アンド・フォワードデバイスで生じるバッファリング時間のパケットサイズ依存が知られているが [28] [29]、IP メータは、内部にハブのミラーポートと同様の構造を有し、ストア・アンド・フォワード型であるため、ネットワークの特性および測定データに影響が生じてしまうため、専用ハードウェアでも、設計に際しネットワークへの影響に注意する必要がある。

2.2 パスの区分的計測

パスを構成する区間毎の遅延時間計測に関して、IP ヘッダの Time-To-Live (TTL) フィールドを利用したホップ・バイ・ホップ手法が多く用いられているが、ICMP 処理を伴い、近

年のハードウェアルータでは ICMP 処理は slow path を通るため精度が低く [30] [31]、しかも、ICMP Error パケットでは RFC1812 で規定されている返送バイト数に関する "SHOULD" の実装の相違も精度低下を招いている [8]。また、得られる遅延時間は片方向ではなく RTT となる。

一方、キャプチャ方式では、管理者の異なるネットワークにまたがった測定では、セキュリティ、ポリシー、スケーラビリティが課題となっている [15] [32]。

2.3 タイムスタンプフォーマット

タイムスタンプには様々なフォーマットが用いられており、UNIX 系 OS では、timeval 構造体: 秒とマイクロ秒をそれぞれ 32 ビット整数、timespec 構造体: 秒とナノ秒をそれぞれ 32 ビット整数、bintime 構造体: 秒を 32 ビット整数、秒以下を 2^{-64} 秒単位の 64 ビット値としている。ntp [33] では、秒を 32 ビット整数、秒以下を 2^{-32} 秒単位の 32 ビット値としている。IP ヘッダのタイムスタンプオプション (RFC781) [34] では、当日午前 0 時からのミリ秒を 32 ビット値としている。一方、これらの「秒」を単位とする時系とは別に、PC で高精度計測する場合には、プロセッサの動作周波数でカウントアップするカウンタ (PCC: Processor Cycle Counter または TSC: Time Stamp Counter) を用いる時系が使われる場合もある [35]。秒単位タイムスタンプは、異ったタイムスタンプの値の差を直接計算できると思われているが、秒の桁と秒以下の桁を装置内部で 1 つの数値に変換する処理は、多くの演算量を必要とする。一方、一定レートのカウンタは、機器の構成が単純で精度がとりやすく、四則演算に適している。

なお、RFC781 では、複数のタイムスタンプの挿入を可能としているが、タイムスタンプの精度もミリ秒しかなく、しかも、タイムスタンプの数によってパケット長が変化してしまうため、精密計測には不十分である。

3. 通過型タイムスタンプの設計

3.1 通過型構造の原理と特徴

タイムスタンプをプローブパケットに載せて伝送することにより、キャプチャ方式の欠点を解決し、リアルタイムに、しかも簡便に経路装置のデータ収集を行うことを可能とする。図 1 に、本装置を用いたアクティブ計測による区分的片方向遅延計測の原理を示す。送信側では、目的に応じた十分な長さの UDP パケットをプローブパケットとして送出する。パストには、複数の本装置が挿入されており、各地点通過時のタイムスタンプをプローブパケットのペイロード部に書き込むことにより、パケットサイズを変化させることなく複数のタイムスタンプを挿入する。更に、ペイロード書き込みに伴うチェックサム値変化をパケット末尾 2 バイトで補償する。受信側では、各タイムスタンプの差から、該当区間の片方向遅延をリアルタイムに計算することができる。

次に、表 1 に通過型方式とキャプチャ方式を比較する。通過型方式では、プローブパケットにタイムスタンプが挿入されているため、リアルタイムに遅延時間を得ることができるが、キャプチャ方式では、キャプチャされたデータを別途収集した

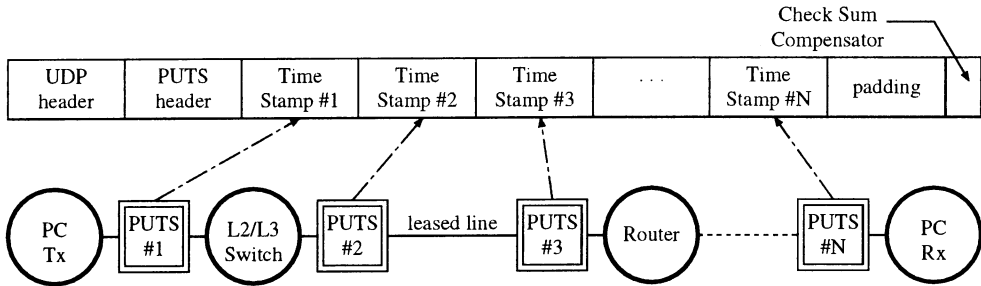


図 1 PUTS 縦列設置による区分的片方向遅延遅延計測

Fig. 1 Picewise One-way Delay Measurement with Cascadable PUTS time-stamp system.

後に、遅延時間を計算する必要があり、リアルタイム計測には適してはいない。

また、個々のパケットに対するタイムスタンプは、異ったキャプチャ装置に保存されているため、それらからデータを収集し、同一パケットに対するデータを抜き出すなどの作業が必要となる。しかも、データ収集時には、データへのアクセス権が必要となるため、セキュリティ上の弱点となる可能性がある。更に、キャプチャリングには通信傍受の側面もあるため、利用には注意が必要となる。一方、キャプチャ方式ではデータを保存するための記憶容量やアクセス用 IP アドレスを必要とする。通過方式では、これらの問題を全て解することが可能であり、アクセス権が不要なため、ゲートウェイの外側に配置した場合には、ローカル側ユーザが全てから共用することも可能であり、高価な専用ハードウェアを有効活用することができる。

一方、通過型方式では、プローブパケットを必要とし、アクティブ計測にしか用いることはできないが、パッシブ方式とされるキャプチャ方式でも、キャプチャデータを収集するためのトラフィックを被計測ネットワークに流す場合には、被計測ネットワークへの影響は避けられない。

3.2 非侵襲性

本装置はネットワークに挿入する使用形態となるため、ネットワークへの影響が最小限となるよう設計する。つまり、通過遅延を一定値かつ最小限とし、更に、ボトルネックとならないこと。具体的には、以下の項目を実現する。

- (1) 遅延ジッタを小さく
- (2) ワイヤレートで動作すること

表 1 通過型とキャプチャの比較

Table 1 Passing through type vs. capturing

	Capturing	Passing through
リアルタイム性	無	有
データ収集作業量	大	小
装置へのアクセス権	要	不要
通信傍受	可	不可
記憶容量	大	不要
IP アドレス	要	不要
プローブパケット	不要	要
データの収集トラフィック	要	不要

上記項目を実現するために、FPGA を用いたハードウェア処理により、ジッタの発生を抑え、ワイヤレートを実現する。また、カットスルー構造を採用し、バッファリング等のパケットサイズ依存性を排除し、通過遅延を短く抑えるとともに、全パケットをパイプラインに通すことにより、処理の有無およびパケット種別によるジッタの発生を抑える。更に、キャリア同期を受信ラインに合わせる。

さて、ペイロードにタイムスタンプを上書きすると、UDP チェックサムが変化してしまう。しかし、カットスルー構造として、滞在時間をパケット長以下に短くした場合、パケット末尾を読み込んだ時点では、すでにチェックサムフィールドはラインに送出された後となる。この問題を解決するために、チェックサム補償方式を導入する。本方式は、チェックサムが常に $FFFFh$ となるよう、パケット末尾 2 バイトの値を調整する方式である。パケット長を L バイトとし、疑似 IP ヘッダ、UDP ヘッダおよび UDP データの先頭より $L-2$ バイト目までの情報から計算されるチェックサム値を C_{L-2} とすれば、チェックサム補償値 m は、以下の式を満たす。

$$C_{L-2} + m = FFFFh \quad (1)$$

したがって、チェックサム補償値は次式により与えられる。

$$m = FFFFh - C_{L-2} \quad (2)$$

チェックサム補償により、パイプラインの段数を低減し、滞在時間を縮小することができる。なお、IPv4 では、チェックサム値を 0 とすることにより、チェックサムによるエラー検出を省くことができるが、IPv6 ではチェックサムは必須である [36] [37] [38]。

3.3 秒単位タイムスタンプと一般化タイムスタンプ

本装置では、秒単位タイムスタンプと PCC 等の任意の一定レートのカウンタがあるが、本論文では、PCC などの一定の速さでカウントアップするタイムスタンプを一般化タイムスタンプと呼ぶこととし、本装置では、両タイムスタンプを利用できるものとする。

さて、予備実験により、1000Base-T 対応スイッチ (非インテリジェントタイプ) の転送遅延は 10 ナノ秒程度であるため、1000Base-T のキャリア周波数である 125MHz を考慮して、タ

タイムスタンプの解像度および処理ジッタを 8 ナノ秒以下とする。また、秒単位タイムスタンプと一般化タイムスタンプ両者の基となるカウンタを駆動するクロックを 1GHz の整数倍または整数の逆数とすれば、カウンタ値のビットシフト演算によりナノ秒単位と容易に変換することができる。今回は、FPGA の性能から 250MHz クロックで駆動し、タイムスタンプのビット長は 64 ビットとする。

秒単位タイムスタンプは、先頭 32 ビットを 1pps カウンタとし、後続 32 ビットは、直前の 1pps 信号からの 250MHz クロック数の 4 倍とする。一方、一般化タイムスタンプは、電源投入時あるいは、電源投入後最初の 1pps でカウンタを 0 にリセットし、以後、250MHz クロックでカウントアップする。

なお、カウンタクロックは固定ではなく、今後、FPGA の性能改善に伴い、更に高いカウントアップ速度によって、より高解像度なタイムスタンプが得られるが、1GHz の整数倍あるいは整数分の 1 とすれば、ナノ秒への変換は容易であろう。ただし、カウントアップ速度の参照方法を用意する必要がある。

また、タイムスタンプ精度は、基準とする発振子の精度によって左右される。温度補償されていない水晶では、数分間でマイクロ秒以上の誤差を生じてしまうため、PUTS では OCXO (または TCXO) を内蔵し、長期間に渡る測定では、GPS や原子時計などの、より高精度な周波数源を外部に接続して使用する。

3.4 設置容易・安全性

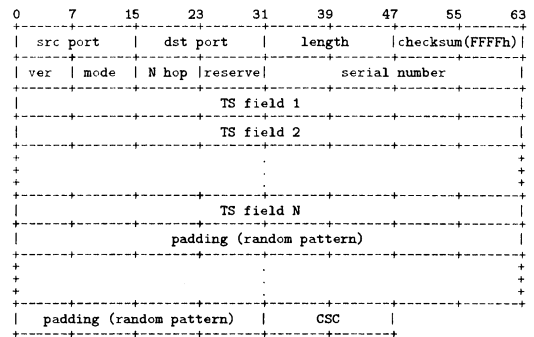
本装置は、より多く設置することにより、より効果を発揮する。もし、全リンクに、PUTS を配すれば、ネットワークの状態推定を極めて簡単に行うことができるであろう。一般に、ネットワークは広域に展開しており、しかも、異った組織によって運営されているネットワークが相互に接続されている。このようなネットワークに本装置を設置するためには、その設置の容易・安全性が重要となる。設置の容易・安全性として、3.2 節の非侵襲性に加えて、以下の項目について考慮した。

- (1) ネットワーク断を起さないこと
 - 電源投入後、速やかに機能すること
 - 故障が少ないこと
- (2) 手間の掛らないこと
 - 起動時に設定が不要なこと
- (3) 設置場所を選ばないこと
 - 小型であること
 - 消費電力が少ないこと
 - 騒音を発生させないこと
 - GPS 受信機が無くても機能すること
- (4) セキュリティホールを含まないこと
 - ログイン不要
 - IP アドレス不要

4. パケットフォーマット

4.1 パケットフォーマット

遅延時間計測用プローブパケットは UDP とし、UDP ヘッダを含むフォーマットは以下の通り。



各フィールドには以下の情報を設定する。

checksum:	PUTS で FFFFh 設定
ver:	バージョン
mode:	タイムスタンプの動作モード指定
N hop:	挿入されている TS フィールドの数
serial number:	通し番号
TS field:	mode に応じて情報をト書き
padding:	ランダムビット列
CSC:	Checksum Compensator

ただし、最低 1 つの TS field を持つ必要がある。したがって、最小パケットサイズは、IP ヘッダ 20 バイト、UDP ヘッダ 8 バイト、UDP ベイロード 18 バイトの合計 46 バイト (IPv4 の場合)。なお、2 点間の時間差を計測するには、TS field が 2 つ必要なため、最低 54 バイトとなる。また、最低バイト数以上であれば、任意サイズが可能である。パケット末尾 2 バイトには、チェックサム補償 CSC (Checksum Compensator) を置き、他はランダムビット列で埋める。ランダムビット列を用いるのは、情報量圧縮符号化が施されている経路でもパケットサイズの大きな変化を防ぐためである。

4.2 動作モード

現在のバージョン (ver = 3) では、表 2 に示す動作モードを用意している。PUTS は、登録したポート番号 (src または dst port) に一致したパケットに対してのみ、モード指定に従ってタイムスタンプ処理を施す。

- 秒単位および一般化タイムスタンプ挿入モード

(Nhop) + 1 番目の TS field に、次のように 64 ビットカウンタ値を上書きし、N hop field の値を 1 つインクリメントする。

表 2 タイムスタンプ処理指示子 (mode)
Table 2 Time-Stamping command (mode)

bit	Description
1	64 ビットカウンタタイムスタンプ挿入モード
2	カウンタレート挿入モード
3	秒単位タイムスタンプ挿入モード
4	イベントタイムスタンプ挿入モード
6	ID 挿入モード
other	予約

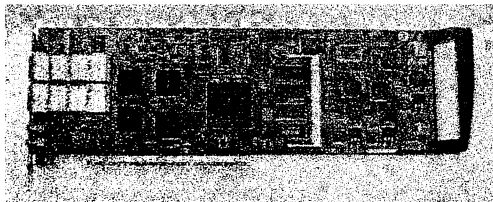


図2 PUTS/X (10GbE 版) の外観
Fig.2 Exterior view of a PUTS/X (10GbE model).

もし、 $(Nhop) + 1$ 番目の TS field の位置が、バケット長を超える場合には、最後の TS field 値に書きこむ。

- カウンタレート挿入モード

タイムスタンプの代わりに、カウンタレートを挿入する。本モードを利用することにより経路する各 PUTS のカウンタレートを知ることができる。

- ID 挿入モード

タイムスタンプの代わりに、8 バイト長の ID を挿入する。本モードを利用することにより経路する PUTS を知ることができる。

- イベントタイムスタンプ挿入モード

時刻情報を有しないセンサーデバイスの発したバケットにタイムスタンプを挿入する。タイムスタンプ挿入と同時に本バケットをクリアすることにより、センサに最も近い PUTS のみがタイムスタンプを挿入することになる。

5. 性能評価

5.1 システムの実装

本装置は FPGA を用いて PCI カードに実装されている (図 2)。回路は 250MHz で動作し、タイムスタンプ用 64 ビットカウンタも 250MHz でカウントアップするため、解像度は 4 ナノ秒となっている。近年、ネットワークのバックボーンは 10GbE 化されてきているため、同アーキテクチャで、10GBase-LR/SR 版 (PUTS/X) と 100/1000Base-T 版 (PUTS/G) を開発した。タイムスタンプ処理対象ポート番号などは、PCI-X バスを介して、ホストコンピュータから設定するが、設定値はフラッシュメモリに保存可能であり、フラッシュメモリに保存された設定で動作する場合には、ホストコンピュータは不要で、単体で機能するため、PCI フォームファクタの設置空間のみで利用可能である。また、ハードディスクや冷却ファンなどの可動部品を有しておらず、故障が少なく、無音で動作する。消費電力は、GbE 版が 9W、10GbE 版が 22W となっている。周波数源はオンボードに OCXO (または TCXO) を搭載し、より精度を必要とする際には、外部より原子時計 (セシウムあるいはルビジウム) または GPS の 10MHz および 1 pps 信号を入力する。起動時間は、電源投入後、GbE 版で約 0.5 秒、10GbE 版で約 1.5 秒である。

5.2 精度

本節では、図 3 の構成を用いて、X の位置に挿入した機器の遅延時間を測定する。PUTS #1 および #2 は、独立したセシウム原子時計により駆動されている。まず、2 枚の GbE 版

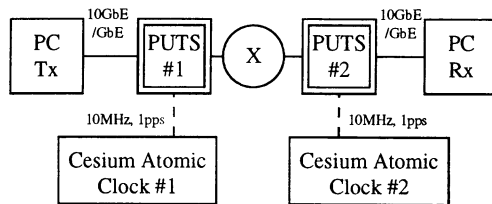


図3 PUTS の精度および遅延
Fig.3 Precision and Latency of PUTS.

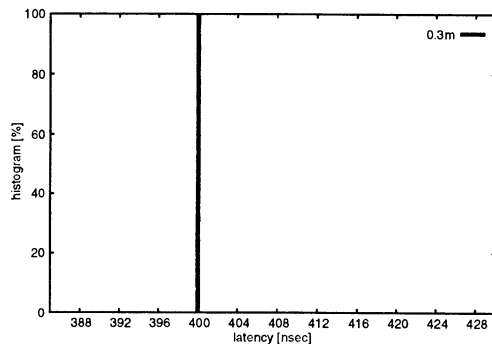


図4 遅延時間の分布 (GbE 版)
Fig.4 Distribution of latency (GbE model)

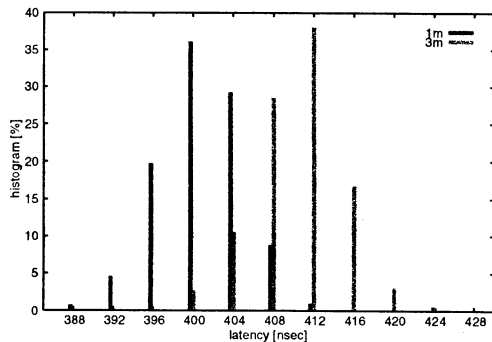


図5 遅延時間の分布 (10GbE 版)
Fig.5 Distribution of latency (10GbE model)

PUTS の間を、長さ 0.3m の UTP ケーブルを用いて、精度を測る。2 枚の PUTS/G のタイムスタンプ差は図 4 に示すように、400nsec 一定であり、ジッタは生じておらず、解像度 (4nsec) の精度で計測可能であることを示している。

次に、2 枚の 10GbE 版 PUTS の間を、長さ 1m および 3m の光ファイバで直結した場合の遅延時間を図 5 に示す。PUTS/X では、XG MII のデータ転送が 156MHz の DDR で行なわれているが、PUTS の動作周波数 250MHz と整数倍となっていないため、タイミング誤差が生じている。長さ 1m のファイバで接続した場合には、遅延時間の平均は 401nsec。3m のファイバの場合は、平均 411nsec。光ファイバ中の光の群速度を 2×10^8 m とすれば、遅延時間の差とファイバ長の差は、一致する。し

たがって、PUTS の測定精度は光ファイバの長さを m 単位で計測可能なレベルであると言える。なお、PUTS/X 一段で生じる遅延およびジッタの標準偏差はそれぞれ 396nsec, 2nsec である。

6. むすび

ネットワーク遅延を測定する、全く新しいタイムスタンプ装置を開発した。本装置を用いることにより、パスを構成する複数区間の片方向遅延を、ナノ秒オーダーの高精度かつリアルタイムにアクティブ計測することが可能となる。なお、本装置は非侵襲性、セキュリティ、設置容易・安全性に優れている。

今後、OWAMP との整合性を図るとともに、インターネットパフォーマンスモニタリングの基盤として普及を図る予定である。また、JGN2 等のリアルネットワークでの継続的計測により、帯域推定、輻輳推定、時刻同期、時計のキャリブレーションなどを行う。

謝辞 コーダ電子(株)野間泉氏と佐武康一郎氏より FPGA 実装に関して多くの助言を頂いた。また、当機構 西永望氏に IPv4 と IPv6 での UDP チェックサムの違いの相違について指摘して頂いた。ここに感謝する。

文 献

- [1] V. Jacobson and M. J. Karels: "Congestion avoidance and control", ACM SIGCOMM 1988, pp. 314-329 (1988).
- [2] V. Paxson: "End-to-end internet packet dynamics", Proc. ACM SIGCOMM 1997, pp. 139-154 (1997).
- [3] T. M. Chen and L. Hu: "Internet performance monitoring", Proc. IEEE (2002).
- [4] L. Carbone, F. Coccetti, P. Dini, R. Percacci and A. Vespignani: "The spectrum of internet performance", Proc. PAM 2003, pp. 131-141 (2003).
- [5] M. Jain and C. Dovrolis: "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp", ACM SIGCOMM 2002 (2002).
- [6] 佐々木, 角田, 太田, 加藤, 根元: "適応型帯域推定と sack を組み合わせた無線ネットワーク向け tcp", 電子情報通信学会論文誌, J87-B, 10, pp. 1657 - 1667 (2004).
- [7] 八木, 堀田, 間瀬: "ボトルネックリンク速度推定ツールの提案と精度検証", 電子情報通信学会論文誌, J87-B, 10, pp. 1636 - 1647 (2004).
- [8] 北口, 町澤, 箱崎, 中川: "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", 電子情報通信学会論文誌, J87-B, 10, pp. 1696 - 1703 (2004).
- [9] K. C. Claffy, G. C. Polyzos and H.-W. Braun: "Measurement considerations for assessing unidirectional latencies", Internetworking: Research and Experience, 4, 3, pp. 121-132 (1993).
- [10] G. Almes, S. Kalidindi and M. Zekauskas: "A one-way delay metric for ippm", RFC 2679, IETF (1999).
- [11] S. Shalunov, B. Teitelbaum, A. Karp, J. W. Boote and M. J. Zekauskas: "A one-way active measurement protocol (owamp)", Rfc, IETF (2004).
- [12] M. Coates, A. Hero, R. Nowak and B. Yu: "Internet tomography", IEEE Signal Processing Magazine (2002).
- [13] J. Michael, S. Donnelly and I. Graham: "Precision timestamping of network packets", Proceedings of the ACM SIGCOMM Internet Measurement Workshop, San Francisco (2001).
- [14] S. KATSUNO, K. YAMAZAKI, T. KUBO, T. ASAMI, K. SUGAUCHI, O. TSUNEHIRO, H. ENOMOTO, K. YOSHIDA and H. ESAKI: "High-speed ip meter him

- and its application in lan/wan environments", Trans. IEEE, E85-D, 8, pp. 1241 - 1249 (2002).
- [15] V. Paxson, A. Adams and M. Mathis: "Experiences with nimi", Proc. PAM 2000, p. 34 (2000).
- [16] S. Kalidindi and M. J. Zekauskas: "Surveyor: An infrastructure for internet performance measurements", Proc. INET 1999 (1999).
- [17] M. Alves, L. Corsello, D. Karrenberg and C. Ogut: "New measurements with the ripe ncc test traffic measurements setup", Proc. PAM 2002, pp. 66-75 (2002).
- [18] J. G. T. Corral and L. Toutain: "End-to-end active measurement architecture in ip networks (saturne)", Proc. PAM 2003, pp. 241-247 (2003).
- [19] A. Danalis and C. Dovrolis: "Anemos: An autonomous network monitoring system", Proc. PAM 2003 (2003).
- [20] G. Jin, G. Yang, B. R. Crowley and D. A. Agarwal: "Network characterization service (ncs)", Proc. IEEE High Performance Distributed Computing 2001 (2001).
- [21] G. Jin and B. L. Tierney: "System capability effects on algorithms for network bandwidth measurement", Proceedings of the Internet Measurement Conference 2003 (IMC 2003), Miami, pp. 27-38 (2003).
- [22] R. Prasad, M. Jain and C. Dovrolis: "Effect of interrupt coalescence on network measurements", Proc. PAM 2004 (2004).
- [23] M. Carson and D. Santay: "Micro-time-scale network measurements and harmonic effects", Proc. PAM 2004, pp. 103-112 (2004).
- [24] S. F. Donnelly: "High precision timing in passive measurements of data networks", Dr thesis of U. Waikato (2002).
- [25] A. Pásztor and D. Veitch: "A precision infrastructure for active probing", Proc. PAM 2001 (2001).
- [26] C. Fraleigh, C. Diot, B. Lyles, S. Moon, P. Owezarski, D. Papagiannaki and F. Tobagi: "Design and deployment of a passive monitoring infrastructure", Proc. PAM 2001, pp. 556-575 (2004).
- [27] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran and C. Diot: "Measurement and analysis of single-hop delay on an ip backbone network", IEEE Journal on Selected Areas in Communications, 21, 6, pp. 908 - 921 (2003).
- [28] R. S. Prasad, C. Dovrolis and B. A. Mah: "The effect of layer-2 switches on pathchar-like tools", Proc. ACM IMW 2002 (2002).
- [29] R. S. Prasad, C. Dovrolis and B. A. Mah: "The effect of layer-2 store-and-forward devices on per-hop capacity estimation", Proceedings of the IEEE INFOCOM 2003, San Francisco (2003).
- [30] S. Savage: "Sting: a tcp-based network measurement tool", Proc. USENIX Symposium on Internet Technologies and Systems 1999 (1999).
- [31] R. Govindan and V. Paxson: "Estimating router icmp generation delays", Proc. PAM 2002, pp. 6-13 (2002).
- [32] D. Agarwal, J. M. Gonzalez, G. Jin and B. Tierney: "An infrastructure for passive network monitoring of application data streams", Proc. PAM 2003 (2003).
- [33] D. L. Mills: "Network time protocol (version 3)", RFC 1305, IETF (1992).
- [34] Z.-S. Su: "A specification of the internet protocol (ip) timestamp option", RFC 781, IETF (1981).
- [35] 町澤, 北口: "割込みハンドラと高精度 pc によるソフトウェアタイムスタンプの精度改善", 電子情報通信学会論文誌, J87-B, 10, pp. 1678 - 1685 (2004).
- [36] R. Braden, D. Borman and C. Partridge: "Computing the internet checksum", RFC 1071, IETF (1988).
- [37] S. Deering and R. Hinden: "Internet protocol, version 6 (ipv6) specification", RFC 2460, IETF (1998).
- [38] J. Postel: "User datagram protocol", RFC 768, IETF (1980).