通過型高精度UDPタイムスタンパの開発

町澤　朗彦[†a)]　鳥山　裕史[†]　岩間　司[†]　金子　明弘[†]

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

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

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るため，パスを構成する区間ごとの片方向遅延を計測する必要がある．従来，経路途中の片方向遅延を高精度に測るには，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ムな遅延時間計測は原理的に不可能であり，しかも，キャプチャデータ集約に伴う作業量及びセキュリティの問題を抱えている．本論文では，アクティブ計測を目的として，ネットワーク上の任意の複数の位置に挿入し，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，複数区間ごとの片方向遅延時間を，リアルタイムかつ容易にアクティブ計測することが可能である．また，PUTSのタイムスタンプは，解像度4ナノ秒，安定度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），と極めて高精度である．なお，本装置はネットワークに挿入して使用するため，非侵襲性と設置容易性を考慮しており，管理組織の異なったネットワーク間でも協調利用することを目指している．

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

1. まえがき

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]．また，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のふくそう制御[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，pingコマンドに代表される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，クロストラヒックも区間ごとに異なるなど，非均質となっている．一般に，ユーザはパ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らかにするために，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，パス途中区間の遅延時間を測るには，DAG project [13]やIPメータ[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，作業量などの問題を抱えている[15]．

本論文では，アクティブ計測を目的として，ネットワーク上の任意の複数の位置に挿入し，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

† 情報通信研究機構，小金井市
National Institute of Information and Communications Technology, 4–2–1 Nukui–Kitamachi, Koganei-shi, 184–8795 Japan
a) E-mail: machi@nict.go.jp

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，複数区間ごと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，一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．

なお，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くする必要があり，FPGA によるワイヤレートでの処理，カットスルー構造，タイムスタンプのペイロードへの上書き，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を実現している．また，系への影響を小さくするとともに，セキュリティの問題も有さないため，設置が容易である．

本論文の構成は以下のとおりである．まず，片方向遅延計測に関連する研究についてまとめ，続いて，**3.** で PUTS の設計，**4.** でプローブパケットフォーマットについて検討する．**5.** では，試作した PUTS ハードウェアを用いて，機能の動作検証するとともに，精度を測定した．更に，**6.** で応用例として，回線利用度推定を行った．

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] 及び 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，すべて PC によるアクティブ計測である．

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 “まとめ割込み” 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，HOTS [24]，DAG [13], [25] や IP メータ [14] などの専用ハードウェアが開発され，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ている [26], [27]．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伴う遅延特性を明かにした [28]．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 [29], [30]，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及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．一方，HOTS は，タイムスタンプ挿入機能を有するネットワークインタフェースであり，End-to-End 計測を目的として開発され，キャプチャ機能を有しないため，区分的計測に用いることはできない．しかも，HOTS はストアアンドフォワード型であり，本論文で提案する PUTS とは全く異なる．

2.2 パスの区分的計測

パスを構成する区間ごと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 [31], [32]，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．なお，“fast path / slow path” の問題は，ICMP だけではなく，IP ヘッダオプションでも見られ [33]，メインパスである “fast path” を対象とする遅延計測では，IP ヘッダオプションの利用を避ける必要がある．

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，スケーラビリティが課題となっている [15], [34].

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：

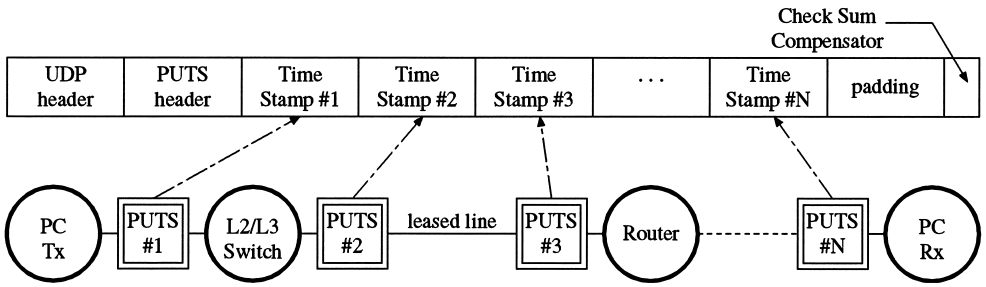


図 1　PUTS 縦列設置による区分的片方向遅延計測
Fig. 1　Picewise one-way delay measurement with cascadable PUTS time-stamp system.

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 [35] 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）[36] 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 [37]．秒単位タイムスタンプは，異なったタイムスタンパの値の差を直接計算できると思われているが，秒のけたと秒以下のけたを装置内部で一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パの設計

本章では，通過型タイムスタンパの備えるべき機能，性能及び実現方法を検討する．本装置は，アクティブ計測により，区間ごとの片道遅延時間を計測するために，通過型の構造を採用し，プローブパケットにタイムスタンプを載せる．**3.1** に通過型構造の特長，**3.2** で通過型構造によるネットワークへの影響を低減するための非侵襲性，更に，**3.3** で本装置のタイムスタンプの精度とタイムスタンプフォーマットについて述べる．また，本装置によりネットワーク遅延を区分的に計測するには，より多地点に設置することが望ましいため，**3.4** で設置容易性について言及する．

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，しかも簡便に経由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．なお，UDP を用いる理由であるが，まず，**2.2** で述べたルータの “fast path / slow path” 問題により，ICMP 及び IP ヘッダオプションだけではなく，UDP 及び TCP 以外のトランスポートプロトコルの使用も避けるべきである．また，片方向遅延を計測する際には，コネクションのオーバヘッドがなく，パケット送出を制御しやすい UDP が適していると考えられるため，UDP のペイロードにタイムスタンプを載せている．

パス上には，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上書きすることにより，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，次節で詳述するように，ペイロード上書き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後に，遅延時間を計

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
データの収集トラヒック	要	不要

算する必要があり，リアルタ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，異な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 アドレスを必要する．通過方式では，これらの問題をすべて解決することが可能である．また，アクセス権が不要なため，スイッチングハブなどのアップリンクポート側に配置することにより，ローカル側ユーザすべ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ヒ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） 通過遅延を短く
（3） ワイヤレートで動作すること

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，キャリヤ同期を受信ラインに合わせることにより，クロックタイミングのずれを抑える．更に，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，処理の有無及びパケット種別によるジッタの発生を抑える．

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 ヘッダ及び 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．

$$C_{L-2} + m = FFFFh \tag{1}$$

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．

$$m = FFFFh - C_{L-2} \tag{2}$$

チェックサム補償により，パケット末尾の到着を待たずにパケット送出することが可能となるため，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 [38]〜[40].

3.3 秒単位タイムスタンプと一般化タイムスタンプ

タイムスタンプには，秒単位のタイムスタンプと PCC 等の任意の一定レートのカウンタがあるが，本装置では，両タイムスタンプを利用できるものとする．なお，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する．

さて，予備実験により，1000 Base-T 対応スイッチ（非インテリジェントタイプ）の遅延ジッタは 10 ナノ秒程度であるため，1000 Base-T のキャリヤ周波数である 125 MHz を考慮して，タイムスタンプの解像度

及び処理ジッタを 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 1 GHz 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 250 MHz 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 1 pps カウンタとし，後続 32 ビットは，直前の 1 pps 信号からの 250 MHz クロック数の 4 倍とすることにより，ナノ秒を表す．一方，一般化タイムスタンプは，電源投入時及び電源投入後最初の 1 pps 入力に対して，カウンタを 0 にリセットし，以後，250 MHz 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，1 GHz 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，遠隔地とのタイムスタンプ比較では絶対時刻との同期が必要であるため，PUTS では，GPS などの高精度な時刻源より 10 MHz 及び 1 PPS を入力して使用することを基本とし，必要とする精度・期間によっては，内蔵の OCXO（または TCXO）のみでも使用可能である．

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，全リンク上に，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，異な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** 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こさないこと
- 電源投入後，速やかに機能すること
- 可動部品を用いず，故障が少ないこと

（2） 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・低消費電力・無音

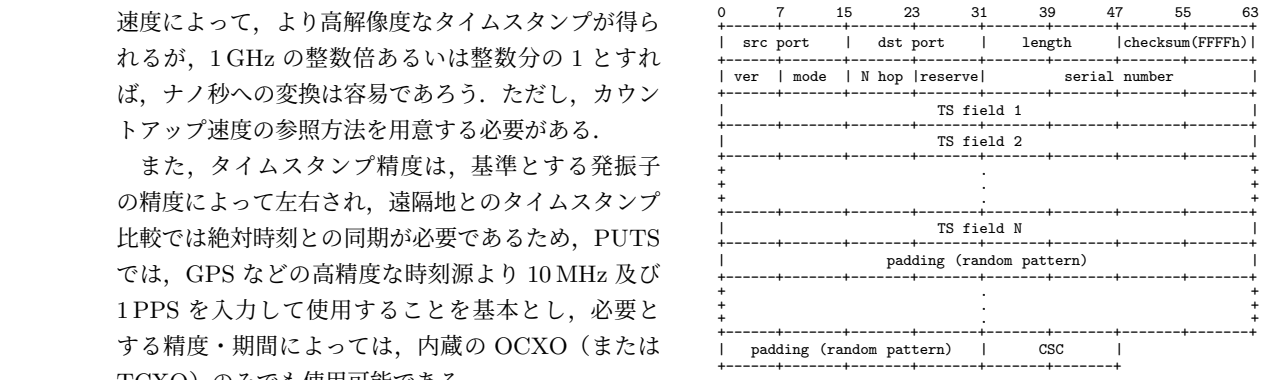
（4） セキュリティホールを含まないこと
- ログイン不要・IP アドレス不要

4. パケットフォーマット

本章では，PUTS のタイムスタンプ挿入対象となるプローブパケットのフォーマットを定義する．また，PUTS は，秒単位または一般化タイムスタンプなどの複数の動作モードを有するが，動作モードもプローブパケットで指定する．

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，PUTS は，事前に登録された UDP ポート番号のパケットに対してのみ，タイムスタンプ挿入処理を施し，他のパケットに対してはそのまま通過させる．UDP ヘッダを含むフォーマットは以下のとおり．



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 に応じて情報を上書き
padding:	ランダムビット列
CSC:	CheckSum Compensator

UDP ペイロード先頭 8 オクテットに PUTS ヘッダ，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．なお，最低

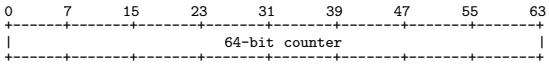
一つの TS field をもつ必要があるが，それ以上であれば，任意サイズのパケット長が可能である．ただし，2 点間の時間差を計測するには，TS field は二つ必要である．

4.2 動作モード

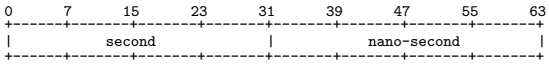
現在のバージョン（ver = 3）では，表 2 に示す動作モードを用意している．PUTS は，登録したポート番号（src または dst port）に一致したパケットに対してのみ，モード指定に従ってタイムスタンプ処理を施す．

- 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 の値を一つインクリメントする．もし，$(Nhop)+1$ 番目の TS field の位置が，パケット長を超える場合には，最後の TS field 値に上書きする．



- 秒単位タイムスタンプ挿入モード一般化タイムスタンプ挿入モードと同様に，64 ビットのタイムスタンプを TS field に上書きするが，先頭 32 ビットは 1 pps カウンタ値，後続 32 ビットは直近 1 pps 入力時よりのナノ秒値とする．



- 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由する各 PUTS のカウンタレート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

タイムスタンプ挿入と同時に本ビットをクリアすることにより，発信元に最も近い PUTS のみがタイムスタンプを挿入することになる．時刻情報を有しないセンサデバイス等の発したパケットにタイムスタンプを挿入する場合に有効である．

- 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由する PUTS を知ることができる．

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 250 MHz で動作し，タイムスタンプ用 64 ビットカウンタも 250 MHz 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 10 GbE 化されてきているため，同一アーキテクチャで，10 GbE-XFP 版（PUTS/X）と 100/1000 Base-T 版（PUTS/G）を開発した．回路は 250 MHz 動作であるが，複数ビット並列に処理することにより，10 Gbit/s でもワイヤレートを実現する．

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 9 W, 10 GbE 版が 22 W 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 10 MHz 及び 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，10 GbE 版で約 1.5 秒である．

さて，全パケットはタイムスタンプ処理の有無にかかわらず，同じパイプラインをカットスルーに通過す

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

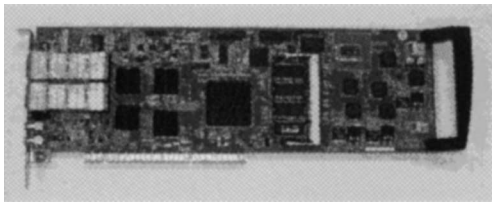


図 2 PUTS/X（10 GbE 版）の外観
Fig. 2 Exterior view of a PUTS/X (10 GbE model).

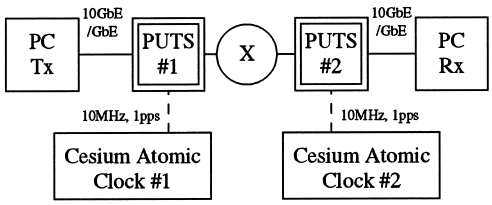


図 3　PUTS の精度及び遅延
Fig. 3　Precision and latency of PUTS.

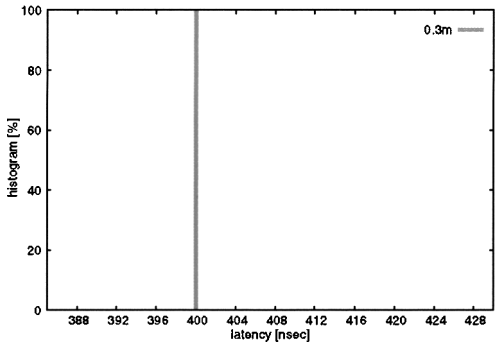


図 4　遅延時間の分布（GbE 版）
Fig. 4　Distribution of latency (GbE model).

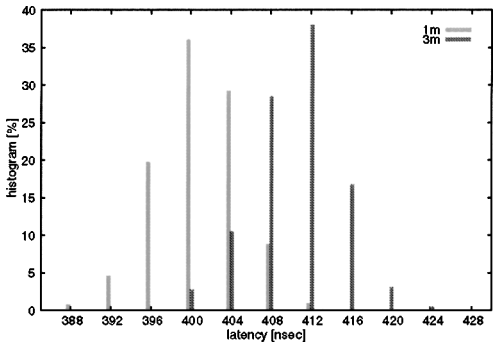


図 5　遅延時間の分布（10GbE 版）
Fig. 5　Distribution of latency (10 GbE model).

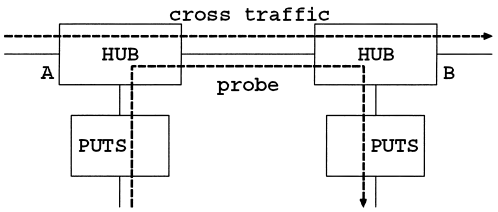


図 6　回線利用度推定テストネットワーク
Fig. 6　Test network whose utilization was estimated.

るため，通過時間は一定となる．また，通過時間は，パケットサイズや負荷の大小の影響も受けない．

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 及び#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 PUTS の間を，長さ 0.3 m 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，400 ns 一定であり，ジッタは生じておらず，解像度（4 ns）の精度で計測可能であることを示している．長さ 0.3 m の UTP では遅延時間はたかだか 2 ns であり，PUTS の解像度以下であるため，PUTS/G 一段で生じる遅延時間は 400 ns とする．なお，FPGA 内部で生じる遅延は，88 ns であり，残りは PHY で生じている．

次に，2 枚の 10 GbE 版 PUTS の間を，長さ 1 m 及び 3 m 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 156 MHz の DDR で行われているが，PUTS の動作周波数 250 MHz 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 1 m のファイバで接続した場合には，遅延時間の平均は 401 ns．3 m のファイバの場合は，平均 411 ns．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いえる．なお，PUTS/X 一段で生じる遅延時間の平均及びジッタの標準偏差はそれぞれ 396 ns，2 ns である．また，FPGA 内部で生じる遅延は，60 ns であり，残りは PHY で生じている．

6. 応　用　例

図 6 の構成で，クロストラヒックの回線利用度を推定する．HUB が FIFO キューイングを行い，クロストラヒックのパケットサイズが一定の場合，プローブパケットのキューイングディレーの分布は一様分布となる（図 7）．図 7 左上は，クロストラヒックの回線占有時間を表す．パケットサイズ P に対して，回線速度を R とすれば，P/R の間，回線を占有する．図 7 左下は，プローブパケットの到着時間によるキューイング時間を示す．クロストラヒックの直後に到着したプローブパケットは，P/R 時間待機し，クロストラヒック送出完了直前に到着したプローブパケットは少ない待機時間で済む．また，回線が空いている時間帯に到着したプローブパケットは，速やかにフォワードされる．これらのキューイング時間からヒストグラムを作成す

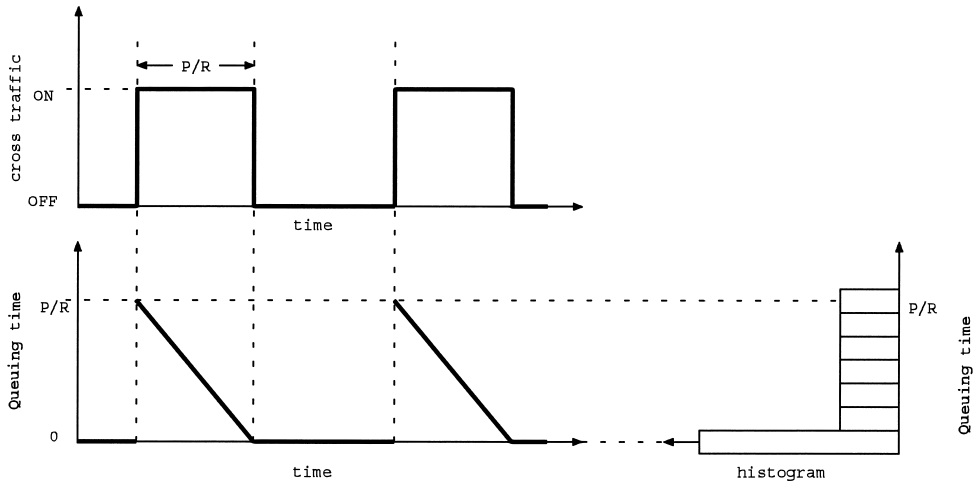


図 7　回線利用度推定の原理
Fig. 7　Priciple of link usage estimation.

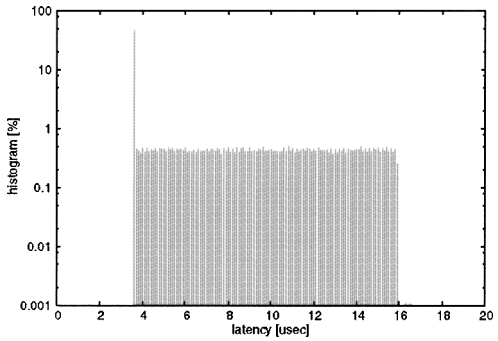


図 8　キューイング遅延分布の実測値
Fig. 8　Histogram of queuing time.

ると，図 7 右下のようになるが，最短遅延時間の割合が回線の空き時間に相当する．

さて，実際に，図 6 の構成（リンクはすべて GbE）でクロストラヒックとして iperf を用いて，UDP 511 Mbps のトラヒックを流した．IP ヘッダ，UDP ヘッダ，Ethernet frame ヘッダ，インタフレームギャップを考慮すると，iperf のトラヒックは，回線容量の 53%を占める．一方，84 バイトの UDP パケットをプローブパケットとして，1 ms 間隔で 10 秒間送出し，片方向遅延時間のヒストグラムを作成した（図 8）．図 8 より，空き帯域は 47%と推定され，極めて高い精度で，回線利用度を推定することが可能である．また，キューイング遅延が，幅 12.3 マイクロ秒の一様分布となっていることから，Ethernet frame ヘッダ及びインタフレームギャップを考慮すれば，クロストラヒックの多くがパケット長 1500 バイトの IP パケットと推定される．

7. むすび

ネットワーク遅延のアクティブ計測を目的として，ネットワーク上の任意の複数の位置に挿入し，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，精度 4 ナノ秒で計測することが可能となる．また，プローブパケットにタイムスタンプを載せて伝送するため，リアルタイムかつ容易に遅延時間を収集することができる，一方，本装置挿入によるネットワークへの影響は，400 ナノ秒ほどの固定遅延が増加するのみで，非侵襲性が高い．なお，本装置は設置容易・安全性に優れており，管理組織の異なるネットワークにまたがった計測にも適している．ただし，組織にまたがった利用では，パケット内容の保証を必要とする場合もあるが，本論文では，OWAMP における非認証（unauthenticated）モードに対応するモードのみの提案となっており，認証モード及び暗号化モードなど，OWAMP の標準化を参考にして開発を進める予定である．

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，本論文では，UDP パケットにタイムスタンプを載せる方式を提案したが，

現在のインターネットの主要なアプリケーションで用いられている TCP の挙動を解析するために，PUTS の TCP への拡張も今後の課題である．更に，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，ふくそう推定，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 チェックサムの扱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, vol.90, no.9, pp.1592–1603, 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, pp.295–308, 2002.

[6] 佐々木貴之, 角田　裕, 太田耕平, 加藤　寧, 根元善章, “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,” 信学論（B），vol.J87-B, no.10, pp.1657–1667, Oct. 2004.

[7] 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,” 信学論（B），vol.J87-B, no.10, pp.1636–1647, Oct. 2004.

[8] 北口善明, 町澤朗彦, 箱崎勝也, 中川晋一, “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,” 信学論（B），vol.J87-B, no.10, pp.1696–1703, Oct. 2004.

[9] K.C. Claffy, G.C. Polyzos, and H.-W. Braun, “Measurement considerations for assessing unidirectional latencies,” Internetworking: Research and Experience, vol.4, no.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),” Internet draft. http://www.ietf.org/internet-drafts/draft-ietf-ippm-owdp-14.txt

[12] M. Coates, A. Hero, R. Nowak, and B. Yu, “Internet tomography,” IEEE Signal Process. Mag., vol.19, pp.47–65, 2002.

[13] J. Micheel, S. Donnelly, and I. Graham, “Precision timestamping of network packets,” Proc. ACM SIGCOMM Internet Measurement Workshop, pp.273–277, 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,” IEICE Trans. Inf. & Syst., vol.E85-D, no.8, pp.1241–1249, Aug. 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. http://www.isoc.org/isoc/conferences/inet/99/proceedings/4h/4h_2.html

[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. http://moat.nlanr.net/PAM2003/PAM2003papers/3707.pdf

[20] G. Jin, G. Yang, B.R. Crowley, and D.A. Agarwal, “Network characterization service (ncs),” Proc. IEEE High Performance Distributed Computing 2001, pp.289–299, 2001.

[21] G. Jin and B.L. Tierney, “System capability effects on algorithms for network bandwidth measurement,” Proc. Internet Measurement Conference 2003 (IMC 2003), pp.27–38, Miami, 2003.

[22] R. Prasad, M. Jain, and C. Dovrolis, “Effect of interrupt coalescence on network measurements,” Proc. PAM 2004, 2004. http://www.pam2004.org/papers/265.pdf

[23] M. Carson and D. Santay, “Micro-time-scale network measurements and harmonic effects,” Proc. PAM 2004, pp.103–112, 2004.

[24] Z. Shu and K. Kobayashi, “Hots: An owamp-compliant hardware packet timestamper,” Proc. PAM 2005, pp.358–361, 2005.

[25] S.F. Donnelly, High precision timing in passive measurements of data networks, Dr thesis of U. Waikato, 2002.

[26] A. Pásztor and D. Veitch, “A precision infrastructure for active probing,” Proc. PAM 2001, 2001. http://citeseer.ist.psu.edu/495791.html

[27] 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.

[28] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot, “Measurement and analysis of single-

hop delay on an ip backbone network," IEEE J. Sel. Areas Commun., vol.21, no.6, pp.908–921, 2003.
[29] R.S. Prasad, C. Dovrolis, and B.A. Mah, "The effect of layer-2 switches on pathchar-like tools," Proc. ACM IMW 2002, pp.321–322, 2002.
[30] R.S. Prasad, C. Dovrolis, and B.A. Mah, "The effect of layer-2 store-and-forward devices on per-hop capacity estimation," Proc. IEEE INFOCOM 2003, pp.2090–2100, San Francisco, 2003.
[31] S. Savage, "Sting: A tcp-based network measurement tool," Proc. USENIX Symposium on Internet Technologies and Systems 1999, 1999. http://www.usenix.org/publications/library/proceedings/usits99/full-papers/savage/savage.pdf
[32] R. Govindan and V. Paxson, "Estimating router icmp generation delays," Proc. PAM 2002, pp.6–13, 2002.
[33] P. Fransson and A. Jonsson, "End-to-end measurements on performance penalties of ipv4 options," Proc. IEEE GLOBECOM 2004, pp.1441–1447, Dallas, 2004.
[34] D. Agarwal, J.M. Gonzalez, G. Jin, and B. Tierney, "An infrastructure for passive network monitoring of application data streams," Proc. PAM 2003, 2003. http://moat.nlanr.net/PAM2003/PAM2003papers/3765.pdf
[35] D.L. Mills, "Network time protocol, (version 3)," RFC 1305, IETF, 1992.
[36] Z.-S. Su, "A specification of the internet protocol (ip) timestamp option," RFC 781, IETF, 1981.
[37] 町澤朗彦，北口善明，"割込みハンドラと高精度 pc によるソフトウェアタイムタンプの精度改善，" 信学論（B），vol.J87-B, no.10, pp.1678–1685, Oct. 2004.
[38] R. Braden, D. Borman, and C. Partridge, "Computing the internet checksum," RFC 1071, IETF, 1988.
[39] S. Deering and R. Hinden, "Internet protocol, version 6 (ipv6) specification," RFC 2460, IETF, 1998.
[40] J. Postel, "User datagram protocol," RFC 768, IETF, 1980.

（平成 17 年 1 月 6 日受付，5 月 10 日再受付）

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．画像の高能率符号化，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．

鳥山　裕史　（正員）

昭 58 名大大学院情報工学専攻博士課程前期課程了．同年郵政省電波研究所（現情報通信研究機構）入所．平 2〜5 ATR 通信システム研究所．平 5〜6 ドイツテレコム研究所客員研究員．画像符号化，情報通信などの研究に従事．

岩間　　司　（正員）

昭 58 山梨大・工・電子卒．昭 60 東工大大学院修士課程了．同年郵政省電波研究所（現情報通信研究機構）入所．以来，電波伝搬特性解析，移動通信のセル構成，標準時，時刻認証基盤技術の研究に従事．現在，電磁波計測部門タイムアプリケーショングループ主任研究員．平 2 本会篠原記念学術奨励賞受賞．IEEE 会員．

金子　明弘　（正員）

昭 57 昭和第一工業高・電気卒．同年郵政省電波研究所（現情報通信研究機構）入所．以来，VLBI，時刻比較，周波数標準等の研究に従事．