

エンドホスト間のネットワークパスの物理帯域のインライン計測手法

Cao Le Thanh Man[†] 長谷川 剛^{††} 村田 正幸[†]

[†] 大阪大学大学院情報科学研究科
〒 560-0871 大阪府吹田市山田丘 1-5
^{††} 大阪大学サイバーメディアセンター
〒 560-0043 大阪府豊中市待兼山町 1-32

E-mail: [†]{mlt-cao,murata}@ist.osaka-u.ac.jp, ^{††}hasegawa@cmc.osaka-u.ac.jp

あらまし 我々の研究グループでは、新しい TCP のバージョンである ImTCP (Inline measurement TCP) を以前に提案した [1, 2]。ImTCP の送信側はデータパケットの送信間隔を調整し、それらに対応する ACK パケットの受信間隔から利用可能帯域を推測する機能 (インライン計測機能) を持つ。インライン計測は計測用のトラフィックをネットワークに導入することなくアクティブにネットワーク特性を調べることが可能というメリットがある。そこで本研究では、物理帯域をインライン計測によって計測する手法を提案し、ImTCP に組み込むことで、利用可能帯域と物理帯域を同時に計測することを可能にする。本研究で提案する物理帯域の計測手法はパケットペア手法を利用するが、ImTCP が利用可能帯域の情報を把握できる利点を生かし、従来のパケットペアを用いた計測手法よりよい計測結果を出すことが期待できる。特に、従来の手法では計測できない、負荷の高いネットワーク環境において本提案手法は正確な計測結果を導出する。また、本提案手法は計測結果に対する信頼区間も導出する。

キーワード end-to-end 計測、利用可能帯域、物理帯域、インライン計測、パケットペア

Inline Measurement Method for Capacity Bandwidth of End-to-end Network Path

Cao LE THANH MAN[†], Go HASEGAWA^{††}, and Masayuki MURATA[†]

[†] Graduate School of Information Science and Technology, Osaka University
1-3, Yamadagaoka, Suita, Osaka 560-0871, Japan
^{††} Cybermedia Center, Osaka University
1-32, Machikaneyama, Toyonaka, Osaka 560-0043, Japan

E-mail: [†]{mlt-cao,murata}@ist.osaka-u.ac.jp, ^{††}hasegawa@cmc.osaka-u.ac.jp

Abstract We previously proposed a new version of TCP, called Inline measurement TCP (ImTCP), in [1, 2]. The ImTCP sender adjusts the transmission intervals of data packets and then utilizes the arrival intervals of ACK packets for available bandwidth estimation. This type of active measurement is preferred because the obtained results are as accurate as those of other conventional types of active measurement, even though no extra probe traffic is injected onto the network. In the present research, we propose a new capacity measurement function and combine it with ImTCP in order to enable simultaneous measurement of both capacity and available bandwidth in ImTCP. The capacity measurement algorithm is a new packet-pair-based measurement technique that utilizes the estimated available bandwidth values for capacity calculation. This new algorithm promises faster measurement than current packet-pair-based measurement algorithms for various situations and works well for high-load networks, in which current algorithms do not work properly. Moreover, the new algorithm provides a confidence interval for the measurement result.

Key words end-to-end measurement, available bandwidth, capacity, inline measurement, packet pair

1. はじめに

ネットワークサービス品質を向上させるために、ネットワークの基盤となる IP ネットワークの資源状況を把握・有効的に利用することは重要である。特に、エンドホスト間のネットワークパスの帯域に関する情報を得ることによって、さまざまなサービス品質の向上が可能になると考えられる。ネットワーク帯域に関する情報の概念としては物理帯域と利用可能帯域がある。物理帯域とは、ネットワーク設備が導入される際に決定される最大転送速度を指すもので、ネットワークパスの中のもっとも帯域の狭いリンクの帯域となる。一方、ネットワークパスにおいて、どの程度ネットワークが空いているかを示す指標が利用可能帯域である。適応型ネットワーク制御にはこの二つの情報が必要である。さらに、この二つの指標は同時に要求される場合が多い。例えば、データ転送プロトコルが輻輳制御を円滑に行うために、利用可能帯域の情報を必要とすると同時に、ネットワーク帯域の利用を高めるために物理帯域の情報も必要とする。また、サービスオーバレイネットワークにおけるルート選択やサーバ選択においても物理帯域と利用可能帯域の両方を利用するとより効果的である。例えば、利用可能帯域が長期間安定している場合、利用可能帯域に基づいて選択を行うが、利用可能帯域の変動が高いときは物理帯域の利用に切り替えた方がよいと考えられる。また、インターネットサービスプロバイダの課金システムも、利用者に提供されるネットワークの利用可能帯域、物理帯域の両方に基づくことが多い。

エンドホスト間の物理帯域および利用可能帯域を知るために、多数の能動的および受動的な計測手法が存在する。能動的にネットワーク内に計測用のトラヒックを導入し、計測を行う手法の方がより正確な計測結果が得られるが、計測用のトラヒックがネットワーク内の他のトラヒックに影響を与えるという問題がある。例えば、Pathload [3] が一回の計測結果を出すためには、2.5-10 MB までの計測用トラヒックを送信する必要がある。Spruce [4] の場合は、一回の計測結果を出すために必要なトラヒック量は平均 300 KB である。一方、サービスオーバレイネットワークにおけるサーバルート選択、またはデータ転送プロトコルの適応型制御のためには、素早くかつ継続的な帯域に関する情報が必要となる。この場合、上述の能動的な計測を繰り返すことにより、大量の計測用トラヒックが発生し、ネットワーク内の他のトラヒックに影響を与え、ともに計測結果の精度自体の低下の原因ともなる。

我々の研究グループでは、サービスを提供しているエンドホスト間の TCP コネクションを直接用いて、データ転送中に得られる情報からエンドホスト間の帯域を随時推測するインラインネットワーク計測方式の提案を行っている。[1, 2] においてわれわれは、インライン計測アルゴリズムを適用した ImTCP と呼ばれる方式を提案した。ImTCP は Reno TCP をベースに送信側アルゴリズムを変更することによって実現している。ImTCP の送信側はデータパケットの送信間隔を調整し、それらに対応する ACK パケットの受信間隔からネットワーク帯域を推測する。この手法により、計測用のパケットをネットワーク内に新しく導入することなく計測を行うことができるため、計測負荷を最小限に抑えることができる。

ImTCP の利用可能帯域計測アルゴリズムの詳細は [5] を参照されたい。提案方式において利用可能帯域を計測する際には、現在の利用可能帯域値が含まれると考えられる帯域の上限と下限を設定し、その区間の中から利用可能帯域を探す。この区間のことを探索区間と呼ぶ。探索区間を設定することで不必要に高いレートで計測パケットを送出することを避けることができるため、外部トラヒックに与える影響を最小限に抑えることができる。また、計測の精度を保ちながら用いる計測パケット数を減少させることができる。次に、探索区間を大きき等の等しい複数の小区間に分割し、それぞれ的小区間に対応する計測用のパケットストリームを送出する。その際、ストリーム内のパケットの送信間隔を変化させることによって、小区間が持つ帯域幅を全てカバーする。計測用のストリームの ACK パケットが送信側に到着すると、送信側が送出間隔と受信間隔の比較を行い、パケット間隔が大きくなったかどうかを判断する。パケット間隔の増加傾向があることが計測ストリームの送信レートが利用可能帯域よりも大きいことを意味する [3] という原理に基づいて、それぞれの小区間に対応するストリームの増加傾向から、利用可能帯域がどの小区間に属するかを推測する。シミュレーションの結果から、ImTCP の計測機能が外部トラヒックに影響を与えず、1-4 RTT に一度計測結果を導出することができること明らかにした。また、[1, 2] の考察結果により、計測を行うことにより、ImTCP がデータ転送プロトコルとして必要な性質を失わないことが明らかになった。

本稿では、インライン計測で用いられる物理帯域計測アルゴリズムを提案し、それを ImTCP に適用する方法についての説明を行う。提案する計測手法は隣接して送信された二つのパケット (パケットペア) に対応する ACK パケットの到着間隔

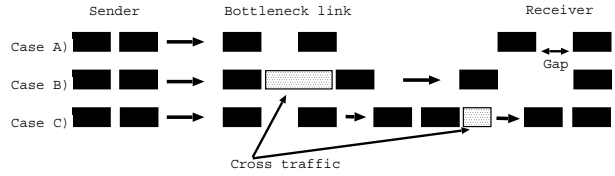


図1 パケットペアの時間間隔の変化

に基づいて物理帯域を推測する。既存のパケットペアに基づく計測手法 [6-8] では、クロストラヒックに割り込まれたパケットペアが不正確な結果を導くため、そのようなパケットペアは結果を計算するためのデータから排除される。それに対して、提案手法はそれらのパケットペアも利用することができる。これにより、提案手法が物理帯域を推測するために用いるデータが従来手法に比べて多くなり、より早くかつ正確な計測結果を得ることができる。

提案手法の基本的なアイデアは、ImTCP がすでに計測可能となっている利用可能帯域の情報を物理帯域の計測に用いることである。利用可能帯域から、パケットペアの間に割り込むクロストラヒック量の平均を推測することにより、クロストラヒックに割り込まれたパケットペアの間隔も物理帯域の推測に用いることができる。さらに、提案手法は導出した結果に対して、信頼区間を算出する。

本稿では、シミュレーション結果より、さまざまなネットワーク環境において ImTCP が常に素早くかつ正確な結果を導出できることが明らかにする。特に、従来の計測手法では計測することができない、ネットワーク負荷が極めて高い環境においても、提案計測手法が正確な結果を出すことができることを示す。

2. パケットペアを用いた物理帯域計測手法

エンドホスト間のネットワークパスの物理帯域を計測するためのツールが、過去に数多く提案されている [6-14]。その中には、異なるサイズの計測用パケットをネットワークパスに流し、それらのパケットの伝送時間を比較することにより、物理帯域を推測する手法がある [10]。また、計測用パケットの TTL (Time To Live) の値を設定し、ネットワークパス上のすべてのリンクの物理帯域を計測する手法も存在する [11-14]。しかし、これらの手法はインライン計測には用いられない。なぜなら、TCP の転送中のデータパケットのサイズを変えたり、TTL の値を設定したりすることによって、TCP の性能を大きく変えてしまうことになるからである。一方、[6-9] に用いられているパケットペアは TCP において容易に実現できるため、本研究ではパケットペアを用いてインライン計測を実現する。

2.1 パケットペア技術

パケットペアを用いた物理帯域計測の原理は、隣接してネットワークパスのボトルネックリンクに到着したパケットペアの後方のパケットが先頭のパケットが処理される間にキューで待たされるため、ボトルネックリンクを出たパケットペアの間の時間間隔がボトルネックリンクに反比例する、というものである。ボトルネックリンク以降のリンクにおいて、パケットペアが再びキューに入らない場合、パケットペアがボトルネックリンクにおいて作られた時間間隔を保つまま受信側に到着する。図 1 の Case A) がこの様子を示している。

このとき、受信側におけるパケットペアの到着間隔から、下記の式でボトルネックリンクの物理帯域 (C) を計算することが可能である。

$$C = \frac{P}{Gap} \quad (1)$$

ここで、 P はパケットのサイズ、 Gap は受信側に到着したときのパケットペアの時間間隔をそれぞれ示す。

パケットペアがネットワークパスを通過する時、上記の場合以外さらに二つの可能性がある。一つ目は、図 1 の Case B) で示すように、ボトルネックリンクにおいてパケットペアがクロストラヒックの packets に割り込まれる可能性がある。この場合、パケットペアの間隔が必要以上に伸ばされ、結果として、式 (1) を用いて物理帯域を計算した結果が正確な値よりも小さくなってしまふ。もう一つは、図 1 の Case C) で示すように、パケットペアが正常に隣接してボトルネックリンクを通過するが、その後、再びほかのキューに入って、パケット間隔が縮小する。この場合、式 (1) が正確な値よりも大きい結果を出してしまふ。

現在のパケットペアを用いた計測手法は上述の Case A) のパケットペアしか利用できない。そのため、パケットペア全体から Case A) のものを特定するために、様々な方法が用いられ

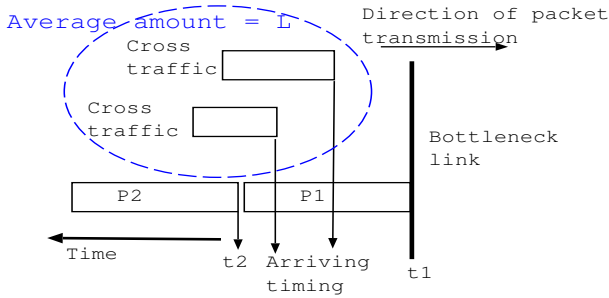


図2 パケットペアとクロストラヒックの到着の様子

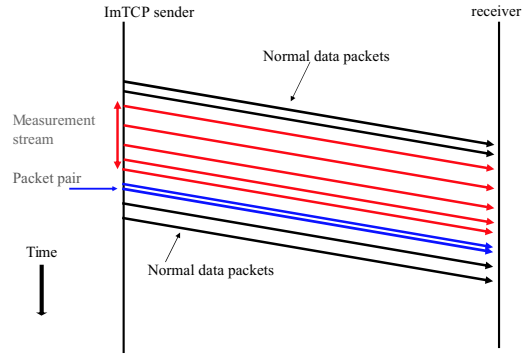


図3 ImTCP におけるパケットペアの作成

た。Pathrate [6] ではパケットペアの間隔のヒストグラムを作成し、Case A) のパケットペアの間隔がもっとも出現頻度が多いという仮定のもとで、Case A) のパケットペアを特定する。また、CapProbe [8] は Case A) のパケットペアが発見できるまでパケットを送り続ける。Case A) のパケットペアであるかどうかはパケットの転送遅延による判断する。

ネットワークパスがほとんど空の状態において、Case A) のパケットペアがもっとも高い頻度で出現する傾向がある。しかし、ネットワークが混雑するときは Case B) のパケットペアの方が Case B)、Case C) よりも多くなる。この場合、CapProbe が確実に Case A) であると判断できるパケットペアを見つけるまでの時間が非常に長く、Pathrate、Nettimer など是不正確な結果を出す。

従来の計測手法と違って、本研究では Case A) と Case B) のパケットペアを両方とも物理帯域の計算に用いる手法を提案する。この手法が可能な理由は、ImTCP が計測している利用可能帯域の情報を利用することである。

2.2 提案アルゴリズム

図2に、パケットペアのボトルネックリンクへ到着する様子を示す。パケットペアの先頭のパケット(P1)の到着時刻を t_1 、後方のパケット(P2)の到着時刻を t_2 とする。 t_1 から t_2 までの間に、クロストラヒックのパケットがボトルネックリンクへ到着することが考えられる。パケットP2はこれらのクロストラヒックのパケットがボトルネックリンクで処理されるまでキューで待つことになる。そのため、ボトルネックリンクを出たときのパケットペアの時間間隔は、後方のパケットP2自身とクロストラヒックの処理される時間の和となる。すなわち、

$$Gap = \frac{P+L}{C} \quad (2)$$

となる。ここで L は t_1 、 t_2 の間にボトルネックリンクへ到着したクロストラヒックの量である。ボトルネックリンクがもっとも利用可能帯域が狭いリンクであると仮定すると、クロストラヒックがボトルネックリンクを流れる量の総和が $C-A$ となる。ここで、 A はその時点の利用可能帯域の値を示す。パケットペアがボトルネックリンクに到着する時の時間間隔を δ ($\delta = t_2 - t_1$) とすると、 L の平均は

$$L = \delta(C - A) \quad (3)$$

となる。(2)式と(3)式から、

$$C = \frac{P + \delta(C - A)}{Gap} \quad \text{または} \quad C = \frac{P - \delta \cdot A}{Gap - \delta} \quad (4)$$

と書くことができる。式(4)により、Case A) および Case B) のパケットペアから物理帯域を計算することができる。次章において、この式を用いた新しい物理帯域の計算方法を提案する。

3. 物理帯域のインライン計測アルゴリズム

3.1 ImTCP におけるパケットペアの作成

ImTCP の送信プログラムは、TCP Reno の送信プログラムにひとつの計測プログラムが配置されることによって、ImTCP の送信プログラムが実現される [2]。計測プログラムは TCP 層の最下、すなわち IP 層とのインタフェース部分に存在する。TCP 層において一つのパケットが処理されると、それを直接に IP 層に渡さず、計測プログラムが TCP 層と IP の層の間に用意している FIFO バッファにいったん格納する。ImTCP が計測を行う際には、バッファ内のパケットを計測アルゴリズムによって決定される時間間隔で IP 層に渡す。計測を行わないときは

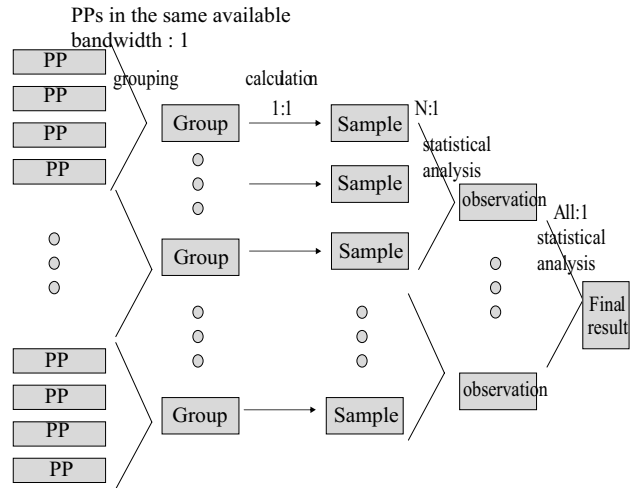


図4 物理帯域の計算アルゴリズム

計測プログラムはバッファへ到着したパケットを即座に IP 層に渡す。

ImTCP が物理帯域と利用可能帯域を同時に計測することを可能とするために、パケットペアの作成及び送信機能を付加する。図3で示されるように、計測プログラムが一つの計測ストリームを送信した後、一つのパケットペアを作成・送信することになる。計測ストリームの送信の間に、数個のパケットが TCP 層からバッファに到着することがあるため、パケットペアを作成するためにさらにパケットを蓄積する時間はほとんど必要としない。そのため、パケットペアを用いた計測機能を導入した ImTCP は、既存の ImTCP と比べての性能の差はほとんどないと言える。

3.2 提案計測手法

図4を用いて、パケットペアの計測結果から物理帯域を推測するアルゴリズムを説明する。提案するアルゴリズムは以下のステップから構成される。

- パケットペアのグルーピング: 利用可能帯域の値が変化していない間に得られたパケットペアの結果を一つのグループにまとめる。グループに属するパケットペアの間隔の平均 \overline{Gap} が求められる。良い平均値を得るために、グループ内のパケットペアの数を3以上とする。そのため、2パケットペア以下のグループは次のグループと合併させる。
- サンプルの計算: それぞれのグループの平均到着間隔の値 \overline{Gap} から、一つの物理帯域のサンプル (sample) を次の式で計算する。

$$C = \begin{cases} \frac{P}{\overline{Gap}}, & \frac{A}{P/\delta} > \lambda \text{ の場合} \quad (5) \\ \frac{P - \delta \cdot A}{\overline{Gap} - \delta}, & \text{ほかの場合} \quad (6) \end{cases}$$

ここで λ は利用可能帯域とパケットペア送信レート P/δ の関係を示す閾値である(パケットペアが送信されてから、ボトルネックリンクに到着するまでの間には間隔が変わらないと仮定する)。利用可能帯域がパケットペアのレートと

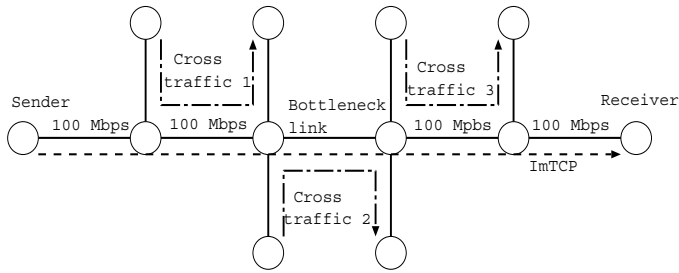


図5 ネットワークトポロジー

ほぼ同じ場合、つまり $\frac{A}{P/\delta} > \lambda$ が満たされるとき、パケットペアがボトルネックリンクを通過する際ほかのパケットに割り込まれる可能性が低い。その場合、パケットペアが Case A) に入るため、式 (5) (式 (1) に基づいた式) を適用する。一方、利用可能帯域がパケットペアの送信レートより小さい場合、つまり $\frac{A}{P/\delta} \leq \lambda$ が満たされるとき、パケットペアが Case B) に入る可能性が高いため、式 (6) を適用する。

● 統計処理

- N 個の物理帯域のサンプルの平均値を計算し、それを一つの観測値 (observation) とする。 N が大きいほど、観測値の信頼度が高いが、一つの観測値の計算するために多数のパケットペアが必要となる。4 章におけるシミュレーションでは、 $N = 10$ を利用する。
- 各観測値から最終的な結果を計算する。また、観測値の分布から、90% の信頼区間を算出する。

4. シミュレーションによる評価

本章では、提案した物理帯域計測手法を ns-2 シミュレータを用いて評価する。また、インラインネットワーク計測の範囲で、提案した手法を適用した場合と CapProbe [8] および Pathrate [6] を適用した場合の比較をあわせて行う。

図 5 にシミュレーションに用いられるネットワークトポロジーを示す。Cross traffic (クロストラヒック) 1 および 3 との送信レートは固定で、それぞれ 5 Mbps、15 Mbps である。クロストラヒックのパケットサイズの分布はインターネットトラヒックの統計データ [15] を用いる。シミュレーション時間は 80 秒である。

4.1 パラメータの設定

● λ の設定

ボトルネックリンクの物理帯域を 90 Mbps、および Cross traffic 2 の送信レートを 5 Mbps に設定した場合の、 λ の値に対する計測結果の変化を示す。図 6(a) と図 6(b) は、それぞれ $\lambda = 0.9$ 、 $\lambda = 0.8$ とした場合の、計測結果の変化を示す。図中のエラーバーは計測結果に対する 90% 信頼区間を表す (信頼区間がつけられていない結果は、パケットペアの数が少なく信頼区間が計算できない場合である)。この場合、ボトルネックリンク上のクロストラヒックが少ないため、式 (5) が使われるべきである。 $\lambda = 0.9$ の設定では、式 (5) が常に使われないため、信頼区間が大きいことからわかるように、計測結果の精度が悪い。逆に、 $\lambda = 0.8$ (または 0.8 よりも小さい値) の設定がよい結果を示している。

次に、物理帯域を 80 Mbps に設定し、Cross traffic 2 を 20 Mbps に設定する。図 7(a) および 7(b) にそれぞれ $\lambda = 0.5$ および $\lambda = 0.8$ とした時の計測結果を示す。この場合、クロストラヒックのレートが高いため、式 (6) が使われる方が好ましい。そのため、 λ の小さい値 (0.5 またはそれ以下) の設定が不正確な結果を出す。また、 $\lambda = 0.8$ はこの場合も良い設定であることが確認できる。

このように、 $\lambda = 0.8$ が上記の二つの場合において良い設定である。また、様々な場合においてもこの設定がよいことがわかったため、以下のシミュレーションでは、 $\lambda = 0.8$ を用いる。

● N の設定

N は観測値を計算するためのサンプル数である。ここでは、物理帯域を 80 Mbps に、Cross traffic 2 を 40 Mbps に設定している環境で考察を行う。図 8(a)、8(b) および 8(c) はそれぞれ $N = 1$ 、10 および 50 の場合の計測結果である。図 8(a) では信頼区間が大きいことがわかり、これは $N = 1$ の場合に計測結果の精度が悪いことを意味する。また、 N が大きい ($N = 50$) 場合においては図 8(c) から、結果を出す

表 1 最初の計測結果を出すために必要なパケットペア数

Capacity	Cross traffic 2	Proposed Algorithm	CapProbe
10 (Mbps)	1 (Mbps)	60 (PPs)	87 (PPs)
10	2	60	85
10	4	60	92
10	5	60	159

のに長い時間がかかることがわかる。図 8(b) から $N = 10$ の設定で良い計測結果が得られることがわかるため、以降この設定を利用する。

4.2 CapProbe との比較

提案手法と CapProbe と公平に比較するために、本研究においては CapProbe を TCP に組み込んだ。TCP に組み込んだ CapProbe と、独立に稼動する CapProbe との唯一の相違点は、本来 CapProbe は、毎回の計測において、計測用のパケットサイズが適宜調整されるのに対し、TCP に導入した場合には、パケットサイズを固定 (TCP のデータサイズ) している点である。この制限は、ImTCP に組み込んだ CapProbe の性能が悪い原因になることが考えられる。

4.2.1 物理帯域が小さく、ネットワーク負荷が低い場合

ボトルネックリンクの物理帯域が 10Mbps で、Cross traffic 2 が 4 Mbps である場合について考察する。図 9(a) が提案手法の計測結果を、図 9(b) に CapProbe の計測結果をそれぞれ示す。両方の結果が良いことがわかる。また、CapProbe の結果の方がより正確であることもわかる。その理由は、CapProbe は正確に Case A) のパケットペアを特定できた場合に、そのパケットペアの間隔のみを用いて物理帯域を計算するためである。CapProbe のもう一つの利点は、提案手法と比べて計算が容易であることが挙げられる。

しかし、CapProbe は最初の計測結果を導出するまで多数のパケットペアを必要とする。表 1 に、Cross traffic 2 の大きさを変化させた時の、最初の計測結果を導出するために必要となったパケットペアの数を示す。表 1 から、CapProbe が最初の結果を出すためにパケットペアを 85 個以上必要としていることがわかり、その数がネットワーク負荷が高くなるとともに大きくなるのがわかる。一方、提案手法はネットワーク負荷によらず、最初の計測結果を出すのに 60 パケットペアだけを必要とする。

4.2.2 物理帯域が大きく、ネットワーク負荷が高い場合

次に、物理帯域を 80 Mbps、Cross traffic 2 の送信レートを 60 Mbps に設定した場合の結果を図 10 で示す。図から、提案手法を用いた計測がよい結果を導出することがわかる。それに対して、CapProbe アルゴリズムを利用した計測の結果が正確でないことがわかる。これは、ネットワーク負荷が高い場合には、ほとんどのパケットペアがクロストラヒックのパケットに割り込まれて、Case B) のパケットペアとなり、Case A) のパケットペアが少ないためである。図 10(b) には、Cross traffic 2 のレートを 50 Mbps とした場合をあわせて示しているが、この結果も正確な物理帯域の値から離れていることがわかる。本来 CapProbe は計測用のパケットサイズを変化させて、計測の性能を向上させるが、インライン計測ではパケットサイズを調整することは好ましくないため、CapProbe はインライン計測には向かないことがわかる。

4.3 Pathrate との比較

次に、Pathrate の計測アルゴリズムを ImTCP で利用することについて考察する。Pathrate はパケットペアの間隔のヒストグラムを必要とするため、ImTCP において収集しているパケットペアの間隔のヒストグラムを作成するようにする。さらに、Pathrate はパケットトレインの計測結果である Average Dispersion Rate (ADR) [6] を必要とするが、パケットトレインを TCP で実現すると、TCP の性能が大幅に影響を受けるため、パケットトレインは独立したツールで計測すること。パケットトレインの結果とパケットペアのヒストグラムから、Pathrate のアルゴリズムに基づいて計測結果を求める。

シミュレーションにおいては、物理帯域が 80 Mbps とし、Cross traffic 2 のレートを様々な値に設定する。またここでは、クロストラヒックのパケットサイズを 400 バイトから 600 バイトの間の一様分布にしたがうものとする。この場合、クロストラヒックのパケットサイズが小さいため、同じ転送レートにお

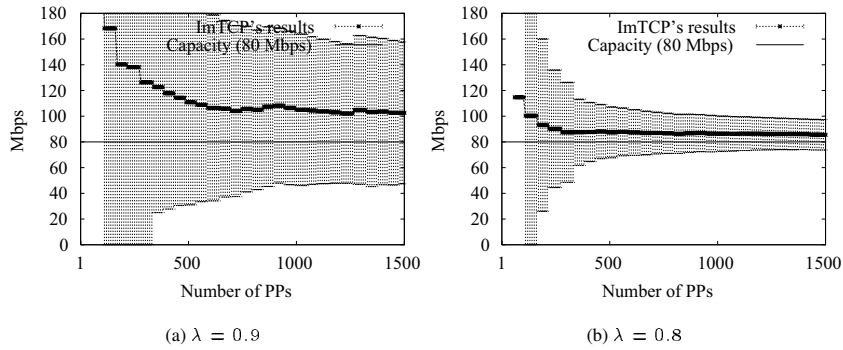


図6 クロストラフィックのレートが5 Mbpsの時の計測結果

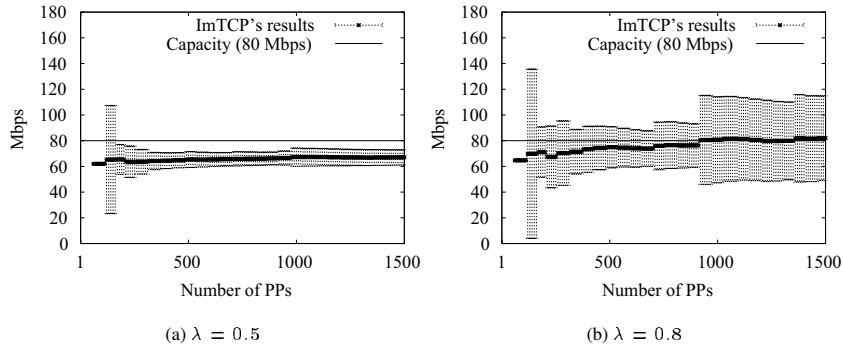


図7 クロストラフィックのレートが20 Mbpsの時の計測結果

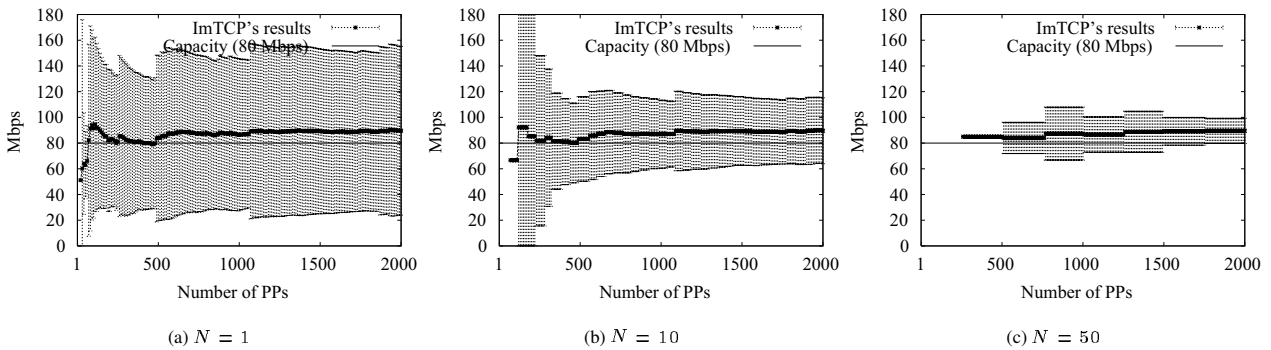


図8 N が変化するときの計測結果

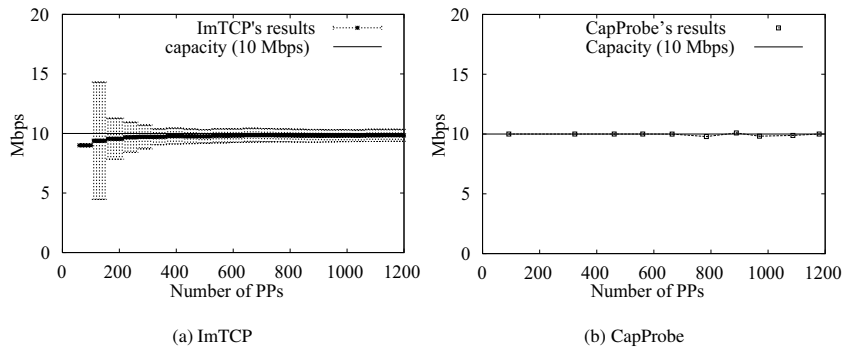


図9 CapProbe との比較結果 (1)

けるパケット数が多くなる。したがって、パケットペアの間にクロストラフィックのパケットが割り込む確率が高くなる。この場合、Pathrateは良い結果を出せないと予測される。この環境において、提案する計測手法を用いた場合およびPathrateを用いた場合の計測結果を表2に示す。この結果から、ネットワーク負荷が高い場合、提案手法がよい計測結果を出すのに対してPathrateが間違った結果を出すことがわかる。

図11(a)および11(b)にCross traffic 2の送信レートが75 Mbpsである場合の、提案手法とPathrateが計測に用いるヒストグラ

ムを示す。図11(a)の“Raw data”のヒストグラムは式(1)を用いたときのパケットペアの計測結果である。一方、図11(b)の“Proposed method”は提案手法によって計算された“observation”の結果のヒストグラムである。この場合、図1に示すCase A)のパケットペアがCase B)のパケットより多いため、図11(a)のヒストグラムが50Mbps付近に集中している(正しい物理帯域の値は80Mbpsである)。それに対して、提案手法が計算した観測値を正しい値である80Mbpsに集中しているため、提案手法が良い結果を導出したものと考えられる。

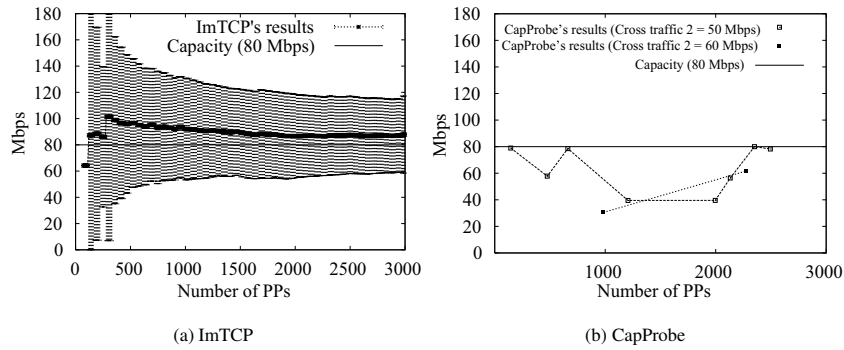


図 10 CapProbe との比較結果 (2)

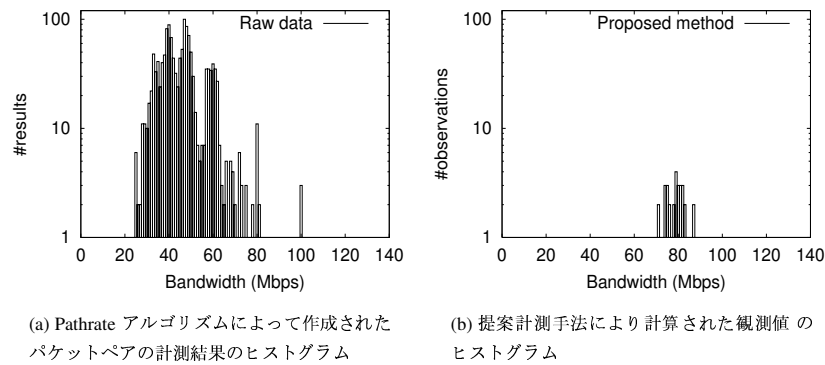


図 11 提案手法および Pathrate のヒストグラム

表 2 提案手法と Pathrate との比較結果

Cross traffic 2	ImTCP(90%confidence interval.)	Pathrate
75 (Mbps)	79.35 (18.26)	49.00
60	80.24 (23.03)	48.00
40	78.32 (26.04)	80.00
10	81.57 (46.98)	80.00

5. 終わりに

本稿では、TCP に適用できる新しい物理帯域の計測手法を提案した。提案手法は ImTCP が計測可能な利用可能帯域の情報を生かし、従来の手法で利用できないパケットペアを計測に用いることができる。シミュレーション結果から、従来手法が正確に動作できない負荷が高いネットワーク環境においても、提案手法が素早く正確な結果を導出することがわかった。

現在、ImTCP を FreeBSD システムに実装しており、実装実験を今後行う予定である。また、TCP 受信側で適用可能な帯域計測手法についても考察する予定である。

謝 辞

本研究の一部は、文部科学省科学研究基礎研究(A)(2)(15200004)によって行われている。ここに記して謝意を表す。

文 献

- [1] Cao Le Thanh Man, Go Hasegawa and Masayuki Murata, "ImTCP: TCP with an inline measurement mechanism for available bandwidth," to appear in *Computer Communications Special Issue: Monitoring and Measurements of IP Networks*, 2005.
- [2] Cao Le Thanh Man, Go Hasegawa and Masayuki Murata, "Available bandwidth measurement via TCP connection," in *Proceedings of the 2nd Workshop on End-to-End Monitoring Techniques and Services E2EMON*, Oct. 2004.
- [3] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput," in *Proceedings of ACM SIGCOMM 2002*, Aug. 2002.

- [4] J. Strauss, D. Katabi and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in *Proceedings of Internet Measurement Conference 2003*, Oct. 2003.
- [5] Cao Man, Go Hasegawa and Masayuki Murata, "A new available bandwidth measurement technique for service overlay networks," in *Proceeding of 6th IFIP/IEEE International Conference on Management of Multimedia Networks and Services Conference, MMNS2003*, pp. 436–448, Sept. 2003.
- [6] C. Dovrolis, P. Ramanathan and D. Moore, "Packet dispersion techniques and capacity estimation," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 963–977, Dec. 2004.
- [7] K. Lai and M. Baker, "Nettimer: A tool for measuring bottleneck link bandwidth," in *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, Mar. 2001.
- [8] R. Kapoor, L. Chen, L. Lao, M. Gerla and M. Sanadidi, "CapProbe: a simple and accurate capacity estimation technique," in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2004.
- [9] R. L. Carter and M. E. Crovella, "Measuring bottleneck link speed in packet-switched networks," Tech. Rep. TR-96-006, Boston University Computer Science Department, Mar. 1996.
- [10] K. Lai and M. Baker, "Measuring link bandwidths using a deterministic model of packet delay," in *Proceedings of ACM Sigcomm*, Aug. 2000.
- [11] Bruce A. Mah, "Pchar," available at <http://www.ca.sandia.gov/~bmah/Software/pchar>.
- [12] V. Jacobson, "Pathchar-A tool to infer characteristics of Internet paths," 1997. available at <http://www.caida.org/tools/utilities/others/pathchar/>.
- [13] A. B. Downey, "Using pathchar to estimate internet link characteristics," in *Proceedings of ACM SIGCOMM*, 1999.
- [14] M. Goutelle and P. Vicat-Blanc, "Study of a non-intrusive method for measuring the end-to-end capacity and useful bandwidth of a path," in *Proceedings of the 2004 IEEE International Conference on Communications*, 2004.
- [15] NLANR web site, <http://moat.nlanr.net/Datacube/>.