社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

ImTCP: 利用可能帯域の計測を可能とする TCP

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　本稿では、エンドホスト間でデータ転送を行うと同時にそれらのデータを利用して利用可能な帯域幅を計測 (インライン計測) する TCP (ImTCP: Inline measurement TCP) 提案する。ImTCP の送信側端末は、データパケットを送信する際にパケットの送信間隔を計測アルゴリズムに基づいて設定し、それらのパケットに対する ACK パケットが送信側に到着する間隔を利用して、利用可能帯域の計測を行う。ImTCP は TCP Reno の送信側のみの変更で実現することができるため、導入が容易であるという利点を持つ。シミュレーションを用いた性能評価結果から、計測機能が TCP のデータ転送性能を低下させず、かつ外部トラヒックに対して影響を与えることなく、1-4 RTT に一回という高い頻度で計測結果を導出することを示した。

キーワード　計測、利用可能帯域、TCP (Transmission Control Protocol)、インライン計測

ImTCP: TCP with an inline network measurement mechanism

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 introduce a new version of TCP (ImTCP: Inline Measurement TCP), which can make use of the data it transfers through a network path binding two end hosts to measure the bandwidth available in the path (inline measurement). ImTCP adjusts the transmission intervals of some data packets then utilize the arrival ACK packets to perform the measurement. ImTCP can be realized based RenoTCP by only changing the sender program. The simulation results show that the measurement mechanism does not degrade the performance of TCP's data transmission, does not give extra effect on surrounding traffic while yielding measurement results in short intervals such as 1-4 RTTs.

Key words　Measurement, Available Bandwidth, TCP (Transmission Control Protocol), Inline Network Measurement

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。特にネットワークリンク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。帯域に関しては以下の三つの概念が利用されている。一つ目は物理的な帯域幅であり、これ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。しかし、多数のフローが共有しているネットワークの資源量を示すためには物理的な帯域よりも、次の二つの指標の方ががより重要な意味を持つ。それらはネットワーク上の二つのエンドホストを結ぶネットワークパス (一連のネットワークリンク）に対して定義される、TCP の最大転送スループット (BTC: Bulk Transfer Capacity)、及び利用可能帯域である。TCP の最大転送スループットとは、二つのエンドホストの間に一本の TCP コネクションが追加された時に、その コネクションが最大で獲得可能なスループットを示す ([1, 2] で定義されている)。この指標は TCP のバージョンやネットワーク上のクロストラヒックの特性に依存する。一方、利用可能帯域とは、ネットワークパスがどのぐらい空いているかを示す指標であり、用いられるデータ転送のプロトコルの特性に依存しない。この指標はネットワーク資源量の状況を把握するためにがよく用いられる。

利用可能帯域に関する情報はさまなところで利用される。まず、データ転送プロトコルが自身の転送レートを利用可の帯域に応じて調整することにより、より高いスループットを実現す

ることが可能である。たとえば TCP コネクションが利用可能帯域が多くあることを認識できたとき、ウインドサイズをより速く大きくすることによって、転送スピードを向上させることができる [3]。また、ネットワークプロトコル層の上位層に構築される各種サービスオーバーレイネットワーク においても利用可能帯域に関する情報は非常に重要である。サービスオーバーレイネットワークには Peer-to-Peer (P2P) ネットワーク [4, 5], Grid ネットワーク [6, 7], コンテンツ配布ネットワーク (CDN: Content Delivery/Distribution Network) [8-10] や IP-VPNs [11] などがあり、それらのネットワークにおいては、利用可能帯域に関する情報は、下記のように利用されることが考えられる。

- P2P ネットワークにおいて、資源発見手続きによって複数のピアが同じ資源を有することがわかった際に、どのピアの資源を利用するかを決定する
- データグリッドにおいて、複数のサイトが同じデータを有する時に、どのサイトからデータをコピーするかを決定する
- CDN において、バックアップまたはキャッシュされるデータのような優先度の低いデータを転送する時、それよりも優先度の高いデータの転送 (Web アクセストラヒックなど) に影響を与えないように、転送速度を決定する

そのほか、ネットワークトのポロジー設計においても、既存のネットワークの利用可能帯域に関する情報は重要である。また、ネットワークプロバイダの課金問題、ネットワークにおける障害発生箇所の特定などの際にも、利用可能帯域に関する情報がよく用いられる [12]。

エンドホスト間の利用可能帯域を知るには、ネットワーク内部のルータとエンドホストが協力して行う手法も考えられるが [13]、ネットワーク内の多くの (理想的にはすべての) ルータに機能を追加する必要があり、かつルータにかかる処理負荷が増大するため、現実的な手法ではないと考えられる。それに対して、エンドホストのみの処理によって利用可能帯域を計測する方法がよく利用され、数多くの計測ツールが提案されている [14-17]。

エンドホスト間で利用可能帯域の計測を行うためには、以下に述べる問題点を克服する必要がある。 まず、利用可能帯域の変化の傾向は計測間隔によって大きく変わるため [2]、利用可能帯域の情報を獲得するためには、できるだけ小さい間隔で計測を行う必要がある。また、利用可能帯域の真の値が、計測用のトラヒック自身の影響を大きく受けることが問題となる。利用可能帯域の計測を行うためには、少なくとも一時的に計測用トラヒックによって空いている帯域を使い切る必要があるため、その際に外部トラヒックに影響を与えてしまうことが考えられる。その結果、外部トラヒックの転送レートが変化するため、利用可能帯域の正確な計測を行うことができない。例えば、初期の計測ツールである Cprobe [18] は高いレートで計測用パケットをバースト的に転送するため、計測結果が利用可能な帯域幅でなく、ADR (Asymptotic Dispersion Rate) という別の値になることが知られている [19]。これら二つの問題点はトレードオフの関係にある。すなわち、計測頻度を高くすると、ネットワークへの影響が大きくなる。一方、計測頻度を減少してネットワークに影響を与えないようにすると、計測結果が不正確となり、計測頻度が低下する。既存の計測ツールは、このトレードオフのバランスをさまざまな方法でとっていると言うことができる。

本研究では上記のトレードオフ問題を解決し、二つの問題点を同時に解決する手法を提案する。すなわち、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利用可能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う。この手法により、計測用のパケットをネットワーク内に送出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。また、TCP コネクションがデータを転送している限り、計測が高い頻度で継続的に行われるため、利用可能帯域の変化をすばやく反映できる。

[20, 21] において我々は、インライン計測に適した計測アルゴリズムを提案した。提案した計測アルゴリズムは、少ない計測パケット数で計測の初期段階から利用可能帯域の計測結果を導出することが可能である。本稿では、提案した計測アルゴリズムを適用した TCP (ImTCP) を提案する。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する。また、その際に発生するいくつかの制御パラメータの設定方法についての議論を行う。また、ImTCP が持つ計測ツールとしての性能 (計測結果、計測頻度、ネットワークへの影響) とデータ転送プロトコルとしての性能 (コネクション間の公平性、リンク利用率、既存の TCP との公平性) に関する評価を行う。シミュレーション結果から、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を示す。また、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を明らかにする。

以下、2 章においてインラインネットワーク計測の関連研究について述べる。3 章にでは TCP Reno をベースに ImTCP を実現する方法について述べる。4 章では ImTCP の計測結果やデータ転送性能についてシミュレーション結果を用いて考察を行う。最後に 5 章でまとめと今後の課題について述べる。

2. インラインネットワーク計測

インライン計測の発想は、従来の TCP にも存在する。なぜなら、従来の TCP は輻輳制御機構により、利用可能帯域の大きさを推測し、それに応じて自分自身の送信速度を調節する機能を持つためである。その意味で、従来の TCP も利用可能帯域を計測しているといえる。また TCP は転送データと ACK パケットを基に、パケットの伝送遅延を計測している [22]。また、TCP のプロトコルとしての本来の目的であるデータ転送を離れて、パケットロスを計測するツールや利用可能帯域を計測するツールに変更させる手法もあった。それらは Sting [23] (パケットロスの計測) や Sprobe [24] (ボトルネックリンクの物理的な帯域幅の計測) である。

TCP が用いているインライン計測アルゴリズムは非常に単純で、効果的ではない。特に TCP は利用可能帯域をすべて使い切るまではその値を認識できない。そのため、利用可能帯域が大きい場合には、正しい計測結果が得られるまでに長い時間がかかる。TCP によりよい利用可能帯域の計測アルゴリズムを導入し、計測結果を TCP コネクションの輻輳制御に利用することによって、高い転送スループットを得る手法は、[25] で最初に提案された。 [25] で提案された手法は受動的な受動な計測方式である。つまり、TCP の送信側において ACK パケットの到着間隔を観察することにより、利用可能帯域を推測する手法である。しかし、計測アルゴリズム自体が非常に単純で、計測結果の精度が低いことが指摘されている [26]。同様の手法として [26] で提案されている TCPW が挙げられる。TCPW は ACK パケットの到着間隔を観測することによって利用可能帯域の計測を行うが、計測結果に対してさまざまなフィルタリング処理を行うことにより、計測結果の精度が [25] と比較して大きく向上している。しかし、ACK パケットの到着間隔を受動的に観察し計測するため、利用可能帯域の変化を速やかにかつ正確に計測できないという本質的な問題を持つ。

これに対して我々の研究においてはまず、インライン計測によって利用可能帯域を推測するための能動的計測手法を提案した [20, 21]。計測のアルゴリズムは既存の能動的計測手法ツール [14, 15] をベースにしているが、TCP コネクション内で利用できるように改善している。提案したアルゴリズムは、[14, 15] の手法とは違い、インライン計測では計測を連続的に行うため、過去の計測結果の統計データに基づいて次の計測の範囲を絞ることによって、素早くかつ少ない量の計測用トラヒックで計測結果を導出することができる。

本稿では、その計測アルゴリズムを利用する ImTCP の提案を行う。ImTCP は ACK パケットの到着間隔を監視するだけではなく、データパケットを送信する間隔も同時に調整する。それにより、より詳細かつ正確な計測データを得ることができるため、計測精度の向上を期待することができる。また、すべての変更が TCP の送信側で行われるため、上記提案方式 [25, 27] と同じコストで実現できると考えられる。

3. ImTCP

3.1 実現方針

TCP によるデータ転送においては、送信側が受信側にパケットを送信し、それを受信した送信側が ACK パケットを送り返

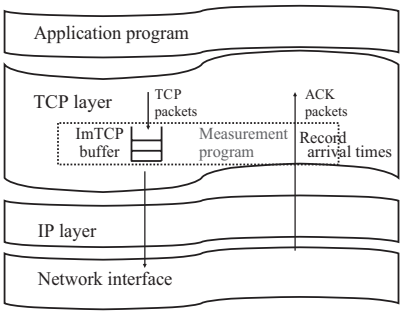

図 1　計測プログラムの位置 (TCP 送信側において)

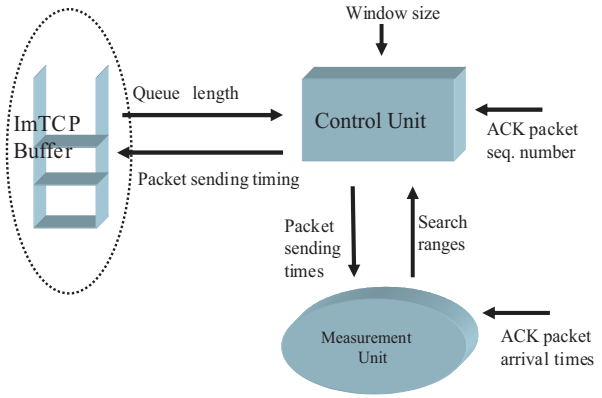

図 2　計測プログラムの構造

す。この性質を利用することにより、送信側で設定した送信間隔に対して、その ACK パケットの到着間隔がどう変化するかを観察することによって、送受信側端末の間の利用可能な帯域幅を計測することが可能である。TCP へ組み込む計測アルゴリズムの詳細については、 [20, 21] を参照されたい。

本研究では、現在もっとも普及している TCP Reno をベースに、送信側 TCP に計測プログラムを組み込むことによって、インライン計測機能をもつ ImTCP を実現する。計測プログラムは TCP のウインドウサイズの値を参照するが、TCP 層におけるパケット処理プロセスには干渉しない。そのため、計測プログラムは TCP 層のもっとも下、すなわち IP 層とのインタフェース部分に存在する (図 1)。TCP 層おいて 1 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 層の間に用意する FIFO バッファ (ImTCP バッファと呼ぶ) にいったん格納する。また、ACK パケットが到着した際には、計測プログラムはその到着時刻を記録する。

ImTCP が計測を行う際には、ImTCP 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間隔で IP 層に渡し、それらのパケットに対応する ACK パケットの到着間隔を利用して、利用可能帯域の算出を行う。計測を行わない場合は計測プログラムは ImTCP バッファに到着したパケットを即座に IP 層に渡す。提案している計測アルゴリズムは、前回までの計測結果を使用して次回の計測範囲を決定するため、前回の計測結果が完全に終了して (計測のために送信したデータパケットに対する ACK パケットが到着して) から次回の計測を開始する。すなわち、ImTCP は 1 RTT で最大で一度の計測を行う。ウインドサイズが小さい場合は、1 RTT の間に計測に必要なすべての計測ストリームを生成できないため、複数 (通常 2-4) の RTT で一度の計測が行われる。また、ウインドサイズが計測ストリームを構成するために必要なパケット数よりも小さい場合は、計測を行わないものとする。

3.2　パケット蓄積機構

計測プログラムは図 2 に示されるように、3 つの部分から構成される。ImTCP バッファは TCP 計測アルゴリズムに応じて TCP のデータパケットの送信間隔を調整するため、パケットを一時的に格納するバッファである。ImTCP バッファは制御部 (Control Unit) によって管理される。制御部はパケットを IP 層への渡すタイミングを ImTCP バッファに知らせる。制御部は計測部 (Measurement Unit) から計測区間を取得し、それに応じて、計測ストリーム内のパケットの送信間隔を決定する。計測部は送信した計測ストリームの ACK パケットの到着間隔を監視し、それに基づいて利用可能帯域の算出を行う。計測部において行われる処理内容、つまり計測アルゴリズムに関しては [20, 21] に示している。ここでは、制御部の詳細について述べる。制御部は 4 つの状態、STORE PACKET、PASS PACKET, SEND STREAM と EMPTY BUFFER を持つ。初期状態は STORE PACKET である。以下にそれぞれの状態においての動作を説明する。

- **STORE PACKET 状態**
  - 計測ストリームを構成するためにパケットを ImTCP バッファに蓄積する。すなわち、TCP パケットが ImTCP バッファに到着した際に、そのパケットを IP 層へ渡さずに ImTCP バッファ内に蓄積する。その際、蓄積される最大時間 T を設定する。T の設定方法については 3.3 節で述べる。
  - 蓄積パケット数が m に達すれば SEND STREAM 状態へ移行する。m の設定方法については 3.3 節で述べる。
  - ウインドサイズが N パケット以下になったとき、または時間 T が経過しても 蓄積パケット数が m に達していない場合は、EMPTY BUFFER 状態へ遷移する。
- **EMPTY BUFFER 状態**
  - 現在 ImTCP バッファに蓄積されているパケットを IP 層に渡す。
  - STORE PACKET 状態へ遷移する。
- **SEND STREAM 状態**
  - ImTCP バッファ内の先頭パケットから計測ストリームを構成し、IP 層へ渡す。計測ストリームの送信中に TCP 層から到着するパケットは ImTCP バッファに蓄積する。
  - 送信したストリームが計測における最後のストリームならば、PASS PACKET 状態へ遷移する。そうでない場合は、EMPTY BUFFER 状態へ遷移する。
- **PASS PACKET 状態**
  - ImTCP バッファに蓄積されているパケット、および新たに到着するパケットをすべて IP 層へ渡す。
  - 送信中の計測ストリームを構成するパケットに対する ACK パケットが全て到着したら、STORE PACKET 状態へ遷移する。

3.3　パラメータ設定

3.3.1　計測ストリームの送信を開始する蓄積パケット数 (m)

計測ストリーム内のパケットは計測アルゴリズムによって決定される時刻に送信される必要がある。計測ストリームを構成するのに十分な数のパケットが ImTCP バッファ内に蓄積されてから計測ストリームを送信すると、パケット蓄積に時間がかかるため、TCP の転送スループットが低下する。したがって、ある程度パケットが蓄積された (その個数を m とする) 時に計測ストリームの送信を開始し、送信中に後続のパケットが ImTCP バッファに溜まることを利用することによって、遅延時間を小さくすることが可能となる。しかし、計測ストリームの先頭部分が送信されている間に ImTCP バッファ内の蓄積されるパケットが無くなると、計測ストリームを構成するパケットをすべて送信することができない (これを計測ストリームの送信失敗と呼ぶ)。これを避けるためには、m を適切に設定する必要がある。m を小さく設定すると、計測ストリームの送信が失敗する可能性が高い。逆に m を大きくすると計測ストリームの送信が成功する可能性が高いが ($m = N$ の時、その成功確率は 1)、パケットが ImTCP バッファに滞留する時間が長くなるため、ImTCP の転送スループットが低下する。以下に、適切な m を決定する手法を示す。これは、計測ストリームの送信が成功するか否かによって、m を動的に増減させる手法である。

- $m = N$ を初期値とする。
- ある m の値で、F 回の計測 ($F = 2$ とすると、4〜8 個の計測ストリームの送信に相当) が成功すれば、m を 1 減少させる。m の最小値は 2 とする。
- ストリームの作成に失敗したら、m を 1 増加させ、スト

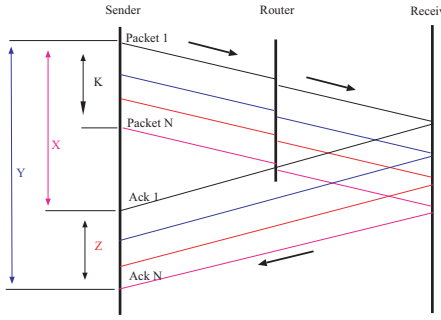

図 3　TCP のパケット転送時間

リームの作成を再度行う。m の最大値は N とする。

3.3.2　計測ストリーム内のパケット数 (N)

ネットワーク状態の変化にともない、利用可能帯域が急激に変化し、計測ストリームの送信レート (R_{send}) が ImTCP バッファにパケットが到着するレート ($R_{arrival}$) よりも低くなると、ImTCP バッファーに多くのパケットが蓄積される。一つの計測ストリームの送信終了後に ImTCP バッファに蓄積されているパケット数は、以下の式で表される。

$$m + N(\frac{R_{arrival}}{R_{send}} - 1) \tag{1}$$

計測ストリームを送信した後、制御部は EMTY BUFFER 状態に遷移し、これらのパケット数を直ちに送信するが、ImTCP の転送スループットを維持するためにはこのパケット数をできるだけ小さくする必要がある。式 (1) 中の m は 3.3.1 節に示した手法で決定されるため、蓄積パケット数を小さくするためには、N を小さくする必要がある。N は計測精度に大きな影響を与え、N を大きくすることによって計測精度が向上することは明白である。N を小さくすることによって、計測に必要なパケット数が減少するため、計測頻度が高くなる。インラインネットワーク計測では、計測頻度、および TCP の転送スループットを下げないことが優先されるため、N は可能な限り小さくする必要がある。

[20, 21] において行った計測アルゴリズムの評価結果より、計測結果が信頼できるためには $5 \leqq N$ である必要があることが明らかになっているため、本方式においては $N = 5$ と設定する。

3.3.3　パケット蓄積のためのタイマ (T)

ImTCP バッファでパケットが長時間蓄積すると、TCP の転送速度が低下するため、タイムアウト時間 T を設定し、ある程度時間が経過しても m 個のパケットが蓄積されない場合には、蓄積を中止して、蓄積されているパケットを直ちに転送する。T の設定に際しては、計測の頻度と TCP の転送速度との間に存在するトレードオフに留意する必要がある。すなわち、T が小さければパケットが長時間蓄積することがないため、転送スループットを高く維持することができる。しかし、計測ストリームの生成に失敗する確率が高くなるため、計測頻度が低下する。一方、T を大きく設定すると、計測ストリームが高い確率で生成できるため、計測頻度が高くなるが、パケットが長時間蓄積される場合があるため、TCP の転送速度が低下することが考えられる。以下では、TCP におけるの RTO (再送時間タイムアウト) 計算方法 [28] に基づいた手法を提案する。

TCP のパケットの RTT が平均 A_{RTT}、 分散 D_{RTT} の正規分布 $N(A_{RTT}, D_{RTT})$ にしたがうと仮定する。A_{RTT} および D_{RTT} は、RTO 算出アルゴリズムから導出される。ここで、以下のように X、Y、Z を定義する。

- X: TCP パケットの RTT
- Y: N パケットの最初のパケットが送信されてから、最後のパケットの ACK が受信されるまでの時間
- Z: N 個の連続した ACK パケットが受信側に到着するのに必要な時間。ここで、ACK パケットが到着すると、一つの TCP データパケットが即座に生成され、ImTCP バッファに送られるとする (送信側に常に送信するデータが用意されている)。そのため、この時間も N 個のパケットが ImTCP に集まる時間となる。

図 3 は 4 つのパケットから構成される 1 つの計測ストリームが送信され、ACK パケットが返送される様子を示したものであり、X、Y、Z の関係をあわせて示している。以下では、Z の分布を導出することによって、T の適切な値を決定する。図 3 から、

$$Z = Y - X \tag{2}$$

となる。上記の仮定から、X は正規分布 $N(A_{RTT}, D_{RTT})$ にしたがう。 Y は 計測ストリーム内の最初のパケットが送信されてから、最後のパケットが送信されるまでの時間 (この時間を K とする) と RTT の和であることから、Y が正規分布 $N(A_{RTT} + K, D_{RTT})$ に従うことがわかる。よって、式 (2) により、Z が正規分布 $N(K, 2 \cdot D_{RTT})$ にしたがう。ただし K は、以下のように近似する。

$$K = \frac{M}{A}(N - 1) \tag{3}$$

ここで、M はパケットサイズ、A は現在の利用可能帯域であり、最新の計測結果を利用する。

Z の分布と式 (3) から 、T を以下のように決定する。

$$T = \frac{M}{A}(N - 1) + 4 \cdot D_{RTT}$$

これは正規分布の性質を利用し、計測ストリームが生成できる確率が 98%となるように設定したものである。これによって、高い確率で計測ストリームを生成し、かつできる限りパケットの滞留時間を短くするように T を決定する。

3.3.4　計 測 頻 度

計測アルゴリズムでは以前の計測結果を利用して、次の計測の計測区間を決定する [20, 21]。そのため、前回の計測に用いられたデータパケットに対する ACK パケットが受信側端末に到着してから、次回の計測を始める。TCP コネクションのウィンドサイズが大きい場合は複数の計測を並行して行うことができるが、並列計測は処理が複雑になり、計測ストリーム間の相互作用による転送スループットの低下など、問題点が多く存在する。TCP コネクションのウィンドサイズが大きい場合は、1 RTT で一度の計測に必要な計測ストリームがすべて送信される。この時、一つの計測ストリームの転送が開始されてから終了するまで (計測ストリームの最初のパケットが送信されてから、計測ストリームの最後のパケットの ACK パケットが受信されるまで) の時間は (1 RTT ＋計測ストリームの送信時間) となる。また、TCP コネクションのウィンドサイズが小さい場合には、1 RTT で一度の計測が終了せず、1 RTT に一つの計測ストリームのみを送信する場合もある。この場合は、一度の計測に最大 4RTT を必要とする。すなわち、計測頻度は ImTCP が設定するパラメータではなく、TCP コネクションのウィンドウサイズ、計測区間の大きさなどから決定される値となり、通常は 1-4 RTT に一度の計測が行われる。

4.　シミュレーション結果

本章では、ns-2 を用いたシミュレーション結果を示し、提案手法である ImTCP が持つ、利用可能帯域の計測精度、およびデータ転送プロトコルとしての性能に関する評価を行う。

4.1　計 測 精 度

まず、図 4 に示すネットワークモデルを用いてシミュレーションを行う。ここでは、背景トラヒックとして UDP によるデータ転送を用いて、計測精度の評価を行う。図 5 は、利用可能帯域が 0 秒から 50 秒までは 60 Mbps、50 秒から 100 秒までは 40 Mbps、100 秒から 150 秒までは 60 Mbps、150 秒から 200 秒までは 20 Mbps、200 秒から 300 秒までは 60 Mbps となるように、背景トラヒックを変化させたときの、計測結果を示している。図中の Search Range は、計測で用いる計測区間を示している。この図から、利用可能帯域の変化に追随して、高い精度の計測結果が得られていることがわかる。またここで得られた結果は、 [20, 21] で示した計測アルゴリズムの性能評価結果とほぼ同じ精度を示している。このことから、本稿で示した

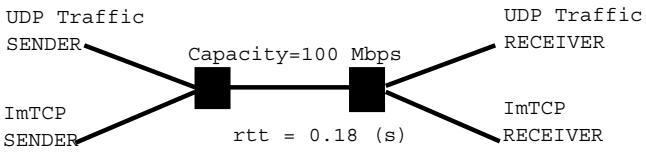

図 4　シミュレーショントポロジー (1)

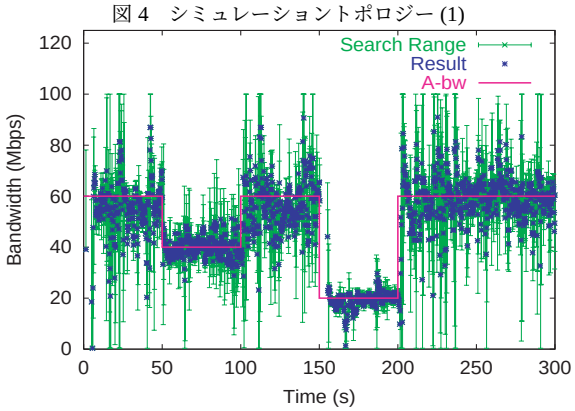

図 5　ImTCP の計測結果

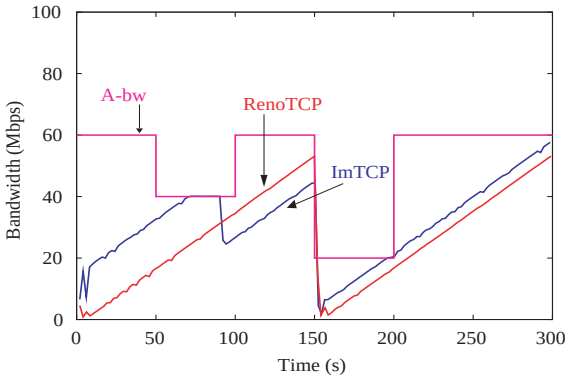

図 6　ImTCP および TCP Reno のスループット

ImTCP 方式によって、計測アルゴリズムの精度を下げることなく、TCP コネクションによるインライン計測を可能にしていることがわかる。

図 6 は、シミュレーション時間中の ImTCP コネクションの転送スループットの変化を示している。図中には、従来の TCP Reno を用いて転送を行った場合のスループットをあわせて示している。図から、計測中の転送スループットは、従来の TCP Reno とほぼ同等であることがわかる。また、図 5、6 から、転送スループットが真の利用可能帯域に達していない期間においても、利用可能帯域を正確に導出していることがわかる。これは、TCPW が行っている計測とは根本的に異なる計測結果である。

4.2　外部トラヒックへの影響

次に、ImTCP が背景トラヒックに与える影響について評価する。ここでは、図 7 に示すネットワークモデルを用い、背景トラヒックとして Web トラヒックを与える。図 8 は、ImTCP を用いた場合と、TCP Reno を用いた場合の、Web ドキュメントの転送時間分布を示したものである。図中の Web only は、ネットワーク内に背景トラヒックである Web トラヒックのみが存在する場合の転送時間分布を示している。図から、ImTCP と TCP Reno が背景トラヒックに与える影響はほぼ同等であることがわかる。また、この間の平均スループットは ImTCP の場合が 25.2 Mbps、TCP Reno の場合は 24.1 Mbps であった。これらの結果から、ImTCP が背景トラヒックに余分な影響を与えることなく、かつ自身の転送スループットを劣化させることなく、利用可能帯域の計測を可能にしていると言える。

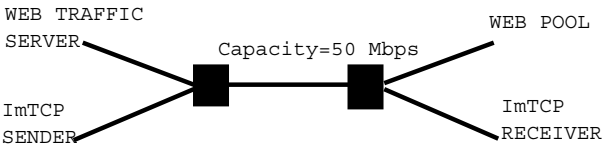

図 7　シミュレーショントポロジー (2)

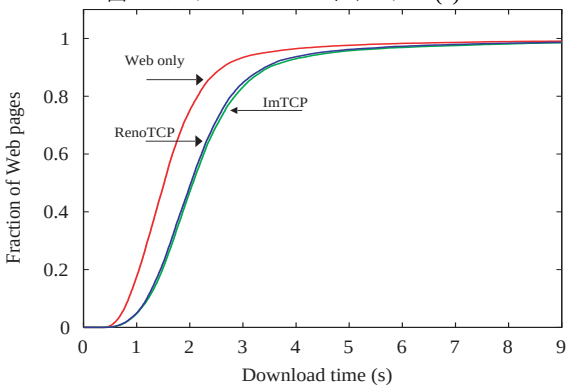

図 8　WEB ページダウンロード時間の比較

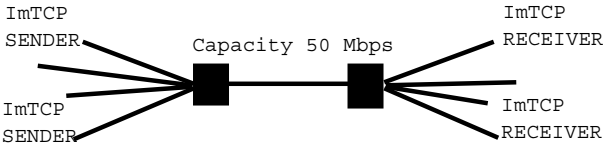

図 9　シミュレーショントポロジー (3)

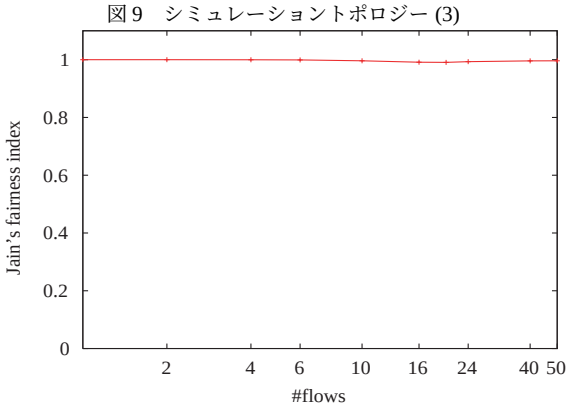

図 10　ImTCP コネクション間の公平性

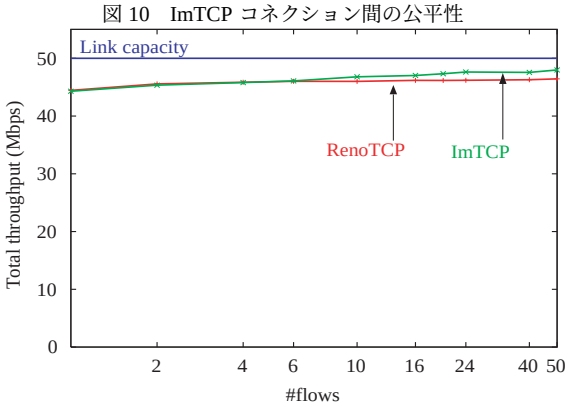

図 11　リンクの利用率

4.3 帯域の利用率と公平性

最後に、ImTCP が持つデータ転送プロトコルとしての重要な性質である、帯域の利用率と公平性に関する評価を行う。ここでは図 9 に示すネットワークトポロジーを用い、ネットワーク内には ImTCP (TCP Reno）のみが存在し、背景トラヒックは存在しないものとする。

図 10，11 は、ネットワーク内の ImTCP (TCP Reno) のコネクション数を変化させたときの、コネクション間の公平性とリンク帯域の利用率を示している。ここで公平性には、Jain's Fairness Index [29] を用いている。これらの図から、ImTCP は TCP Reno とほぼ同等のリンク利用率を達成することが可能であり、その際のコネクション間の公平性は非常に高いことがわかる。

5. おわりに

本稿ではデータ転送中に得られる情報からエンドホスト間の利用可能帯域を推測するインラインネットワーク計測方式を用いた ImTCP を提案し、その性能をシミュレーションによって評価した。その結果、ImTCP は計測ツールとしての高い性能を持ち、データ転送プロトコルとして TCP Reno とほぼ同等の性能を示すことが明らかとなった。今後の課題としては実ネットワークを用いた性能評価を行い、その有効性を検証することがあげられる。

謝　辞

本研究の一部は、総務省戦略的情報通信研究開発推進制度における特定領域重点型研究開発プロジェクト「ユビキタスインターネットのための高位レイヤスイッチング技術の研究開発」、および及び文部科学省科学研究費基盤研究 (A)(2) (15200004) によっている。ここに記して謝意を表す。

文　献

[1] M. Allman, "Measuring end-to-end bulk transfer capacity," in *Proceedings of ACM SIGCOMM Internet Measurement Workshop 2001*, ACM SIGCOMM, Nov. 2001.

[2] R. Prasad, M. Murray, C.Dovrolis and K.Claffy, "Bandwidth estimation: Metrics, measurement techniques, and tools,," *IEEE Network*, Nov. 2003.

[3] R.Wang, G.Pau, K.Yamada, M.Sanadidi and M.Gerla, "TCP startup performance in large bandwidth delay netwroks," in *Proceedings of INFOCOM '04*, 2004.

[4] A. Rao, K. Lakshminarayanan, S.Surana, R.Karp and I.Stoica, "Load balancing in structured P2P systems," in *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, Feb. 2003.

[5] F. Dabek, B. Zhao, P. Druschel, J. Kubiatowicz and I. Stoica, "Towards a common API for structured peer-to-peer overlays," in *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, Feb. 2003.

[6] Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman, "Grid information services for distributed resource sharing," in *Proceedings of the tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press*, Aug. 2001.

[7] Y. Zhao and Y. Hu, "GRESS - a Grid replica selection service," in *Proceedings of the 16 International Conference on Parallel and Distributed Computing Systems (PDCS-2003)*, Aug. 2003.

[8] Akamai Home Page, `http://www.akamai.com/`.

[9] Exodus Home Page, `http://www.exodus.com/`.

[10] G. Pierre and M. van Steen, "Design and implementation of a user-centered content delivery network," in *Proceedings of the third IEEE Workshop on Internet Applications*, June 2003.

[11] J. Jha and A. Sood, "An architectural framework for management of IP-VPNs," in *Proceedings of the 3rd Asia-Pacific Network Operations and Management Symposium*, Sept. 1999.

[12] The Internet Bandwidth Tester (TPTEST), `http://tptest.sourceforge.net/about.php`.

[13] T. Oetiker and D. Rand., "Multi router traffic grapher," `http://people.ee.ethz.ch/~oetiker/webtools/mrtg/`.

[14] 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.

[15] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil and L. Cottrell, "PathChirp: Efficient available bandwidth estimation for network paths," in *Proceedings of Passive and Active Measurement Workshop 2003* , 2003.

[16] J.Strauss, D.Katabi and F.Kaashoek, "A measurement study of available bandwidth estimation tools," in *Proceedings of the conference on Internet measurement conference*, 2003.

[17] R.Anjali, C.Scoglio, L.Chen, I.Akyildiz and G.Uhl, "ABEst: An available bandwidth estimator within an autonomous system," in *Proceedings of IEEE Globecom 2002*, Nov. 2002.

[18] 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.

[19] C. Dovrolis and D. Moore, "What do packet dispersion techniques measure?," in *Proceedings of IEEE INFOCOM 2001*, pp. 22–26, Apr. 2001.

[20] 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.

[21] Cao Le Thanh Man, 長谷川剛, 村田正幸, "サービスオーバーレイネットワークのためのインラインネットワーク計測に関する一検討," Tech. Rep. IN-03-176, 電子情報通信学会技術研究報告, Jan. 2003.

[22] M. Gerla, Y. Sanadidi, R. Wang, A. Zanella, C. Casetti and S. Mascolo, "TCP Vegas: New techniques for congestion detection and avoidance," in *Proceedings of the SIGCOMM '94 Symposium*, pp. 24–35, Aug. 1994.

[23] S. Savage, "Sting: A TCP-based network measurement tool," in *Proceedings of USITS '99*, Oct. 1999.

[24] Sprobe, `http://sprobe.cs.washington.edu/`.

[25] J. C. Hoe, "Improving the start-up behavior of a congestion control sheme for TCP," in *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, vol. 26,4, pp. 270–280, ACM Press, 1996.

[26] Mark Allman and Vern Paxson, "On estimating end-to-end network path properties," in *Proceedings of SIGCOMM '99*, pp. 263–274, 1999.

[27] M. Gerla, M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti and S. Mascolo, "TCP Westwood: Congestion window control using bandwidth estimation," in *Proceedings of IEEE Globecom 2001*, pp. 1698–1702, Nov. 2001.

[28] R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley, 1994.

[29] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley-Interscience, 1991.