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.

高速ネットワークにおけるインラインネットワーク計測手法の実装に関する検討および性能評価

津川　知朗† 　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　〒 565–0871 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 豊中市待兼山町 1–32
E-mail: †{t-tugawa,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，アクティブな TCP コネクションのデータ転送を利用して帯域を計測するインラインネットワーク計測という概念を提案している．この概念に基づいた計測手法には，利用可能帯域を計測するための余計なトラヒックを必要とせずにアクティブ計測が可能であるという利点が存在する．しかしながら，インラインネットワーク計測手法を汎用コンピュータへ実装する場合，カーネルシステムの時間粒度，ネットワークインターフェースによる割り込み削減機構 （IC: Interrupt Coalescence），受信側 TCP の動作などに関する問題が発生する．本稿では，インラインネットワーク計測手法の実装時に問題となる点について説明し，それらに対する解決策を示す．さらに，本稿ではインラインネットワーク計測手法を FreeBSD 4.10 へ実装し，実験ネットワークを用いた実装実験を行う．実装実験を通じて，我々の提案しているインラインネットワーク計測の概念が実ネットワークにおいても有効であることを示す．

キーワード　インラインネットワーク計測，利用可能帯域，時間粒度，割り込み削減機構 （IC: Interrupt Coalescence）

Implementation issues on inline network measurement algorithms in gigabit networks

Tomoaki TSUGAWA†, Cao LE THANH MAN†, Go HASEGAWA††, and Masayuki MURATA†

† Graduate School of Information Science and Technology, Osaka University　Yamadaoka 1–5, Suita-shi, Osaka 565–0871 Japan
†† Cyber Media Center, Osaka University　Machikaneyama 1–32, Toyonaka-shi, Osaka 560–0043 Japan
E-mail: †{t-tugawa,mlt-cao,murata}@ist.osaka-u.ac.jp, ††hasegawa@cmc.osaka-u.ac.jp

Abstract　We have proposed the concept of *inline network measurement*, which involves the concept of "plugging" an active bandwidth measurement mechanism into an active TCP connection. Mechanisms using this method have the advantage of requiring no extra traffic for measuring available bandwidth. However, when the inline network measurement algorithms are implemented in general-purpose computers, some problems arise, such as the clock resolution of the kernel system, Interrupt Coalescence (IC) deployed in network interface cards, and the behavior of TCP receiver. In the present paper, we explain these difficulties and describe our current solutions. Furthermore, we implement the measurement algorithms and the solutions against those problems in a FreeBSD 4.10 kernel system, and present some results on experimental networks. The experimentally obtained results are used to verify the solutions and to confirm the effectiveness of our concept, inline network measurement, on actual networks.

Key words　inline network measurement, available bandwidth, clock resolution, Interrupt Coalescence (IC)

1. はじめに

近年のネットワーク速度の飛躍的な向上やインターネット利用者数の爆発的な増加にともなってインターネットが急速に発展していくにつれて，提供されるネットワークサービスも多種多様のものとなってきている．例えば，コンテンツ配信を目的とした Contents Delivery Network （CDN），ピア同士の直接的な通信を実現する P2P ネットワーク，ネットワーク上で分散計算環境を提供するグリッドネットワーク，IP ネットワーク上に仮想網を構築する IP-VPN などのサービスオーバレイネットワークが挙げられる．これらのネットワークサービスの品質を向上させるためには，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し，有効に利用することが重要となる．

文献 [1] によると，利用可能帯域とは送受信ホスト間のネットワークパス上で使用されていない帯域のことを表す．利用可能帯域を計測するための手法はこれまでにも多く提案されているが [2–5]，それらの手法の多くは，計測を行う際に多くの計測用パケットを必要とするためにネットワークへ大きな影響を与

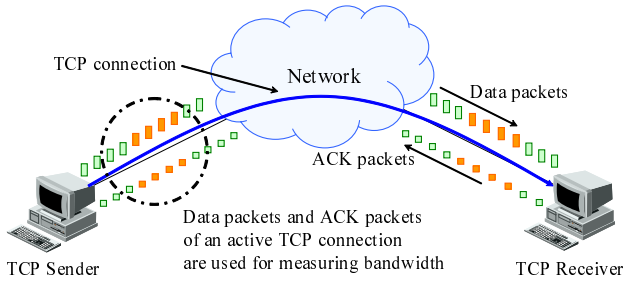


図 1　インラインネットワーク計測

える，計測に長い時間がかかるなどの問題が存在する．これに対して，我々の研究グループではインラインネットワーク計測という計測概念を提案している [6, 7]．これは，図 1 に示されるように，アクティブな TCP コネクションのデータ転送を利用して帯域計測を行うというものである．インラインネットワーク計測に基づく計測手法には，余計な計測用トラヒックを必要とせずに利用可能帯域の計測を行うことができるという利点が存在する．さらに，アクティブな TCP コネクションのデータ転送を利用することにより，送信ホストの修正のみで計測手法を実装することができるという利点も存在する．我々の研究グループでは，この計測概念に基づく計測手法としてこれまでに ImTCP [6] および ICIM [7] を提案している．

アクティブな TCP コネクションを利用した計測手法には前述したような利点が存在する．しかしながら，これらの計測手法を汎用コンピュータへ実装する際には，いくつかの問題が生じる．最も大きな問題は，カーネルシステムの時間粒度である．パケット間隔に基づく計測手法では，パケット間隔を調節するために十分な精度のタイマを必要とするが，カーネルシステムの時間粒度は一般的にアプリケーションよりも荒い．そのため，パケット間隔の調節も荒くなり計測制度が低下する．また，1 Gbps を超えるような高速ネットワークでは，パケット間隔はさらに小さくなり Pathload [3] などの既存の計測ツールもうまく機能しないことが知られている [8]．さらに，高速ネットワークではネットワークインターフェースカード （NIC）で採用されている割り込み削減機構 （IC: Interrupt Coalescence）[9, 10] による問題も生じる．IC とは，CPU への負荷を抑えるために割り込み回数を削減する機構であるが，これによってカーネルシステムで観測されるパケットの到着間隔が変化するため，パケット間隔に基づく計測手法は機能しなくなる．これに対して，我々の研究グループでは，高速ネットワークにおけるこれらの問題を解決した計測手法として Interrupt Coalescence-aware Inline Measurement （ICIM）[7] を提案している．ICIM は，Pathload などのパケット間隔に基づく計測手法とは異なり，IC によってバースト的に転送されるパケット群に含まれるパケット数を調節することにより利用可能帯域の計測を行う．

本稿では，インラインネットワーク計測手法の実装時に問題となる点について説明し，それらに対する解決策を示す．さらに，インラインネットワーク計測手法を実装し，実ネットワーク上で実装実験を行う．実装実験を通じて，帯域が 100 Mbps 以下のような低速なネットワークにおいては，ImTCP が利用可能帯域を計測することができることを確認する．さらに，カーネルシステムの時間粒度に関するパラメータ設定と計測制度の関係についての評価も行う．次に，高速ネットワークにおいて ICIM が利用可能帯域を計測することができることを明らかにし，高速ネットワークにおける，パケット間隔に基づく計測手法に対する優位性を確認する．特に，ICIM はカーネルシステムの時間粒度が荒い場合においても利用可能帯域を計測することができることを示す．

以下，2 章では帯域計測手法をカーネルシステムへ実装する際の問題点について説明する．3 章では，ImTCP および ICIM のアルゴリズムの概要および 2 章の問題点への解決策について説明する．また，それらの計測手法を FreeBSD 4.10 のカーネルシステムに実装する際の実装指針を示す．4 章では，実験ネットワークを用いてこれらの提案方式の評価を行う．最後に，5 章で本稿のまとめと今後の課題を示す．

表 1　*HZ*, 時間粒度, および計測される帯域

HZ	時間粒度 (1 tick) [μs]	計測される帯域 [Mbps] 1 tick	2 ticks	3 ticks	4 ticks	5 ticks
100	10,000	1.2	0.6	0.4	0.3	0.24
1,000	1,000	12	6	4	3	2.4
10,000	100	120	60	40	30	24
20,000	50	240	120	80	60	48
50,000	20	600	300	200	150	120
100,000	10	1,200	600	400	300	240

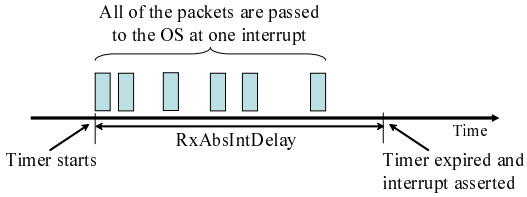


図 2　パケット受信時に動作する絶対タイマ

2.　帯域計測手法の実装時における問題点

本章では，インラインネットワーク計測に基づく計測手法を実システムへ実装する際の問題点について述べる．

2.1　カーネルシステムの時間粒度

パケット間隔に基づく帯域計測手法をカーネルシステムに実装する際には，カーネルシステムの時間粒度が重要となる．パケット間隔に基づく計測手法は，利用可能帯域を推測するためにデータパケットの転送間隔を調節する必要がある．アプリケーションプログラムとして計測手法を実装する場合，アプリケーションプログラムは（例えば，UNIX では `gettimeofday()` を用いて）継続的にシステム時間の確認を行い，適切なタイミングでパケットを送信する．x86 ベースの CPU を採用している Linux では，ハードウェアへ一度アクセスを行うためにおよそ 1.9 μsec の時間を必要する（FreeBSD では，9 μsec）[11]．また，`write()` システムコールは，平均 2 μsec の時間を必要とする．したがって，Linux では，最小 3.9 μsec 間隔でデータパケットを送信することができる．一方，カーネルシステムにおいては，データパケットのパケット間隔の調節はパケット送信プログラムを Interrupt Service Routine （ISR）へ登録することによって実現される．汎用的な UNIX OS においては，このために `hardclock()` システムコールが提供されているが，FreeBSD や Linux では，`hardclock()` は 10 msec 毎に割り込みが発生することによって呼び出される．すなわち，カーネルシステムの時間粒度は通常アプリケーションよりも荒くなる．その結果，計測精度が低下する．

FreeBSD においては，カーネルシステムの時間粒度は *HZ* と呼ばれるパラメータによって決定される．表 1 に，*HZ* の値，カーネルシステムの時間粒度，およびその時間粒度に基づいてパケット間隔を調節した時のデータ転送レートを示す．例えば，FreeBSD のカーネルシステムにおいては，*HZ* の値として通常 100 が用いられているが，表 1 から，この粒度では 1.2Mbps までの帯域しか計測することができないことが分かる．さらに，この表から最大計測可能帯域に近づくほど計測粒度は粗くなることも分かる．したがって，広帯域ネットワークで利用可能帯域の計測を行うためには *HZ* の値を大きくする必要がある．しかしながら，文献 [12] で述べられているように，*HZ* の値を大きくするとタスク切り替えが頻繁に発生するようになり，その処理のオーバヘッドが原因でカーネルの実行速度に影響を与えるようになる．そのため，*HZ* の値を決定する際には，これらのことを考慮して決定する必要がある．

2.2　割り込み削減機構 （IC: Interrupt Coalescence）

1 Gbps を超えるような高速ネットワークにおいては，既存の帯域計測ツールにとって新たな問題が発生する．それは，多くのギガビットネットワーク用のネットワークインターフェースカード （NIC） [9, 10] に採用されている割り込み削減機構 （IC: Interrupt Coalescence）によるものである．IC とは，短時間のうちに到着した複数のパケットをまとめ，一度の割り込みで OS へ渡す技術である．IC を利用しない場合，パケットが NIC

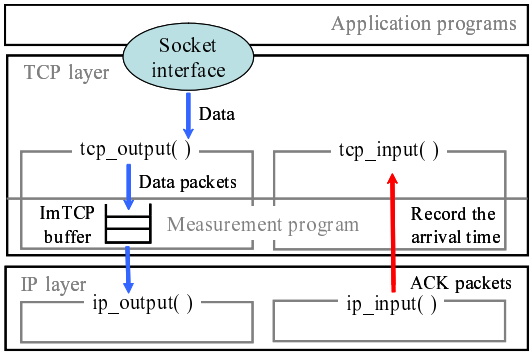


図 3　ImTCP の構成図

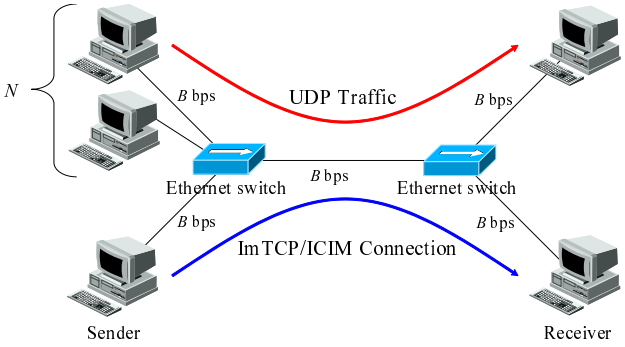


図 4　実験ネットワーク環境

へ到着する度に OS へ割り込みが発生し，CPU へ大きな負荷をかけることになる．言い換えると，IC は CPU への負荷を抑えるために重要な技術である．しかしながら，IC はカーネルで観測されるパケットの到着間隔を変更するため，パケット間隔に基づく帯域計測手法へ非常に大きな影響を与える．

文献 [9] によると，IC には多くのタイマが存在するが，高速なネットワーク環境下では絶対タイマ（absolute timer）が主に機能する．絶対タイマには 2 種類存在し，パケットの転送時，受信時それぞれに発生する割り込みを削減する．パケットの転送時に発生する割り込みは，カーネルシステムへパケットの送信が完了したことを知らせるのみであるため，パケット間隔には影響を与えない．一方，パケットの受信時に発生する割り込みが削減された場合，カーネルシステムで観測されるパケットの受信間隔が変化する．図 2 が示すように，パケット受信時の絶対タイマは，最初のパケットを受信するとタイマが作動する．そして，タイマがゼロに達すると割り込みが発生し，それまでに受信したパケットを全てカーネルシステムへ渡す．タイマ時間のの長さは，FreeBSD では *RxAbsIntDelay* というパラメータによって決定される．したがって，*RxAbsIntDelay* よりも小さな間隔で到着したパケットの間隔は，カーネルシステムではほとんどゼロとして観測される．また，IC によって発生した 2 つのバーストの間隔は *RxAbsIntDelay* 以上となる．したがって，IC が動作している環境においては，パケット間隔に基づく計測手法は計測精度が低下する．

2.3　受信側 TCP の動作

TCP に組み込まれるパケット間隔に基づく計測手法では，受信側 TCP が，受信したデータパケットに対して直ちに ACK パケットを返送する必要がある．しかしながら，今日の多くの TCP の実装では，遅延 ACK オプションが標準で有効になっている [13]．遅延 ACK が有効な場合には，パケット間隔に基づく計測手法はうまく機能しない．したがって，帯域計測手法を TCP に組み込む際には，遅延 ACK の問題を解決する必要がある．この問題への解決策のひとつとして，計測手法に遅延 ACK オプションが有効になっている場合には，計測結果が不正確であることを計測結果を利用する上位アプリケーションに対して通知することが考えられる．

3.　インラインネットワーク計測手法のアルゴリズムおよび実装指針

本章では，インラインネットワーク計測の概念に基づく 2 種類の計測手法の概要を説明する．ひとつは，パケット間隔に基づく計測手法である Inline measurement TCP （ImTCP）である．もう一つは，高速ネットワークを想定して提案されたバースト間隔に基づく計測手法である Interrupt Coalescence-aware Inline Measurement （ICIM）である．さらに，本章ではこれら 2 つの計測手法を FreeBSD の TCP へ実装する際の実装指針についても述べる．

3.1　ImTCP: パケット間隔に基づく計測手法

3.1.1　アルゴリズム

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，受信側が ACK パケットを返送する．ImTCP はこの性質を利用し，送信側で設定したデータパケットの送信間隔に対して，その ACK パケットの到着間隔の変化を観察することによって利用可能帯域の計測を行う．

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ていると考えられる帯域の上限と下限を過去の計測結果を利用して設定し，この区間の中から利用可能帯域を探索する（この区間を探索区間と呼ぶ）．探索区間を設定することで，不必要に高いレートでパケットを送出することが避けられるため，ネットワークに与える影響を最小限に抑えることができる．また，探索する帯域が狭くなるため，計測の精度を保ちながら用いるパケット数を減少させることができる．探索区間は過去の計測結果を基に設定するため，ネットワーク状況の変化に伴い利用可能帯域が急激に変化した場合，探索区間内に利用可能帯域が存在しない場合が存在する．ImTCP では，そのような場合においても，数回の計測で新たな利用可能帯域を発見することができる．ImTCP の詳細なアルゴリズムについては，文献 [6] を参照されたい．

3.1.2　実 装 指 針

図 3 に送信端末のカーネルシステムに実装する ImTCP の構成を示す．通常，アプリケーションから socket インターフェースを通じて渡されたデータパケットは関数 `tcp_output()` によって TCP のプロトコル処理を受けた後，関数 `ip_output()` によって IP のプロトコル処理を受け，ネットワークへ送出される．ImTCP は TCP コネクションの輻輳ウィンドウサイズなどを制御パラメータとして用いるため，TCP 層の下部に実装される必要がある．そのため，計測プログラムは `tcp_output()` 内に実装する．具体的には，カーネルシステムによって確保されたメモリを用いて FIFO バッファ（これ以降，これを ImTCP バッファと呼ぶ）を作成し，TCP のプロトコル処理が終了したパケットを関数 `ip_output()` に渡す前に ImTCP バッファへ格納する．格納されたパケットは計測アルゴリズムに基づいたタイミングで 関数 `ip_output()` へ渡される．このとき，計測プログラムはデータパケットの送信時刻を記録する．

計測プログラムは受信した ACK パケットの処理を行う関数 `tcp_input()` 内にも必要となる．すなわち，受信端末から返送された ACK パケットは関数 `ip_input()` によって IP のプロトコル処理を受け，関数 `tcp_input()` へ渡される．計測プログラムは，ACK パケットの受信時刻を記録し，データパケットの送信時刻と ACK パケットの受信時刻を用いて計測アルゴリズムに基づいて利用可能帯域の推測を行う．

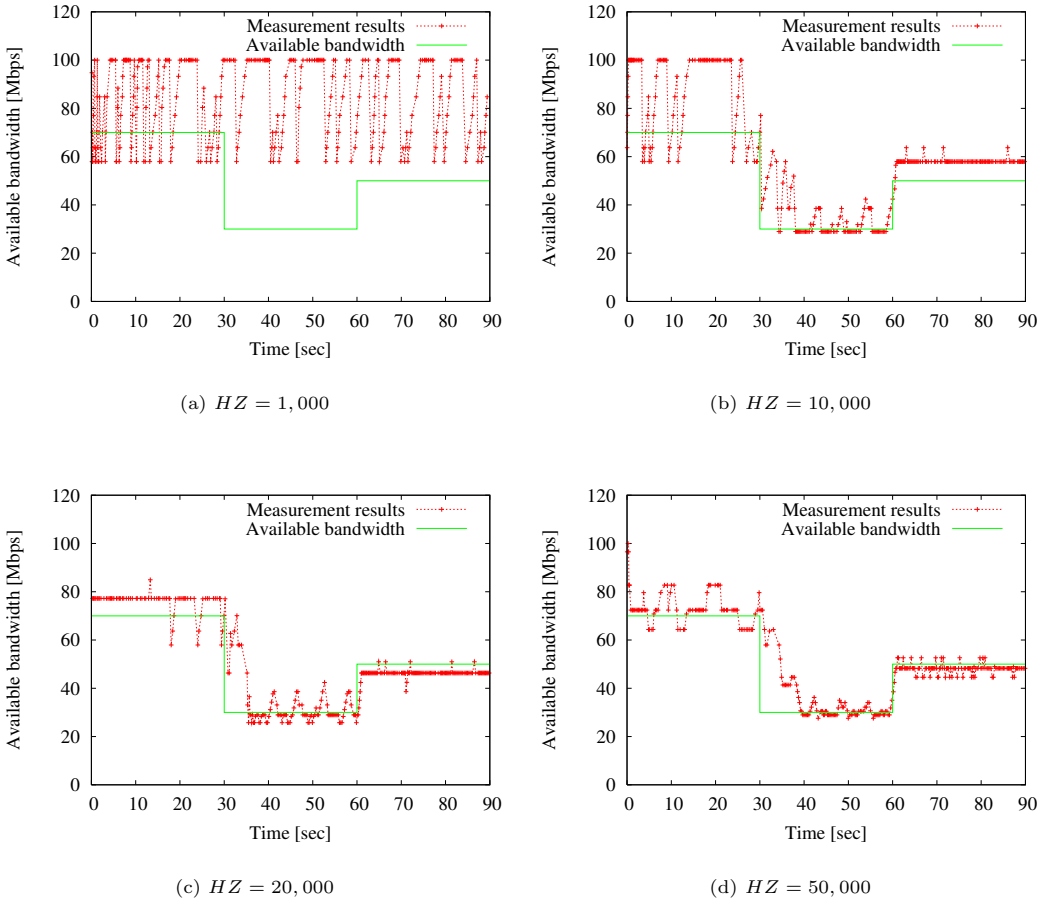
3.2　ICIM: バースト間隔に基づく計測手法

3.2.1　アルゴリズム

2.2 節で述べたように，1 Gbps を超えるような高速ネットワークではパケット間隔に基づく計測手法はうまく機能しない．これに対して，我々の研究グループではバースト間隔に基づく計測手法である ICIM を提案している．ICIM は，IC によって発生するデータパケットのバースト的な転送を利用して送受信ホスト間のネットワークパス上の利用可能帯域を計測する．具体的に言うと，送信側 TCP が IC によってバースト的に転送されるパケット群 （これ以降，これをバーストと呼ぶ）に含まれるパケット数を調節し，そのデータパケットのバーストに対応する ACK パケットのバーストの到着間隔の変化を観察することによって利用可能帯域の計測を行う．また，ICIM においても 3.1.1 節で述べた探索区間の概念を導入する．これにより，ImTCP と同様の利点を享受することができる．ImTCP の詳細なアルゴリズムについては，文献 [7] を参照されたい．

表 2　実験に使用した PC のスペック一覧

	Sender	Receiver	Other endhosts
CPU	Intel Pentium 4 3.0 GHz	Intel Pentium 4 3.0 GHz	Intel Pentium 4 3.4 GHz
Memory	1,024 MB	1,024 MB	1,024 MB
OS	FreeBSD 4.10	FedoraCore 4	FedoraCore 4



(a) $HZ = 1,000$　(b) $HZ = 10,000$

(c) $HZ = 20,000$　(d) $HZ = 50,000$

図 5　ImTCP による計測結果

ICIM の利点の一つとして，受信側 TCP の遅延 ACK オプションが有効になっている場合においても利用可能帯域を計測できることが挙げられる．ICIM では，バーストに含まれるパケット数とそれに対応する ACK パケットの数を TCP のシーケンス番号を利用してバイト単位で比較することによって利用可能帯域を推測する．そのため，遅延 ACK オプションが有効になっている場合においても計測を行うことができる．

3.2.2　実 装 指 針

ICIM を FreeBSD 4.10 カーネルシステムへ実装する際には，3.1.2 節で述べた ImTCP の実装を引き継ぎ，修正を加えることによって実現する．始めに，パケット送信プログラムを修正する．3.2.1 節で述べたように，ICM では，パケットの間隔を調節するのではなく，IC によって発生するバーストに含まれるパケット数を調節することによって利用可能帯域を計測する．したがって，ICIM の計測プログラムは FIFO バッファに格納されているパケット数を計測アルゴリズムに基づいて調節し，それらのパケットを一度に `ip_output()` へ渡す．計測プログラムは，同時にバーストに含まれるパケット数も記録する．

次に，ACK パケットの到着を観察するためのプログラムを修正する．ICIM の計測プログラムは，ACK パケットの到着時間を記録せずに，計測用バーストに対する最後の ACK パケットが `tcp_input()` へ到着した時に，受信した ACK パケットの数を計算する．計測プログラムは，計測用バーストに含まれるデータパケット数とそれに対応する ACK パケット数を比較することによって最新の利用可能帯域を推測する．

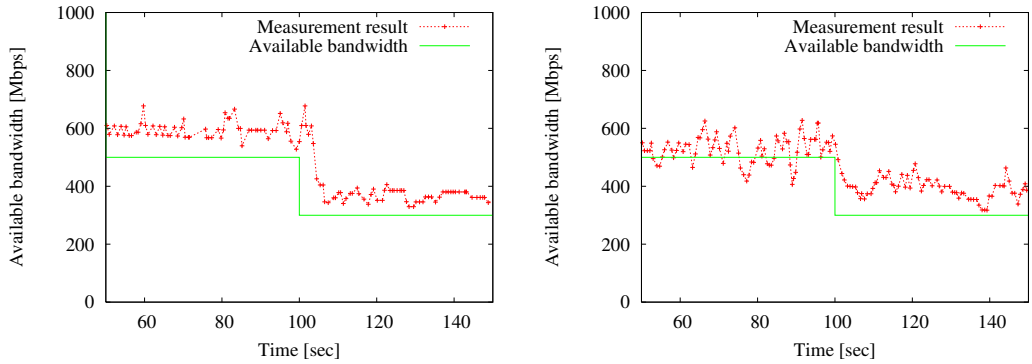
ICIM は，本質的には HZ の値に依存せずに利用可能帯域を計測することができるが，カーネルシステムの時間粒度によってある制限を受ける．ICIM は，2 つのバースト間の到着間隔を観察することによって利用可能帯域を計測する．したがって，カーネルシステムの時間粒度が NIC の絶対タイマよりも荒い場合には，バースト間の間隔を検知することができない．そのため，ICIM を用いて計測を行う場合には，カーネルシステムの時間粒度が絶対タイマ（$RxAbsIntDelay$）よりも荒くならないように HZ を設定する必要がある．この問題に対する解決策として，HZ の値を変更する代わりに，バースト間の間隔を検知する時のみ精度の高い時刻取得関数を用い，それ以外の場合には通常のカーネルシステムが持つ時計を利用するという方法が考えられる．しかしながら，この手法は実際にシステムへ実装するのが困難であるため，ICIM の実装では用いていない．また 4.2 節では，ICIM は，HZ の値が小さくカーネルシステムの時間粒度が荒い場合においても，その精度は低いものの利用可能帯域を計測することができることを示す．

4.　性 能 評 価

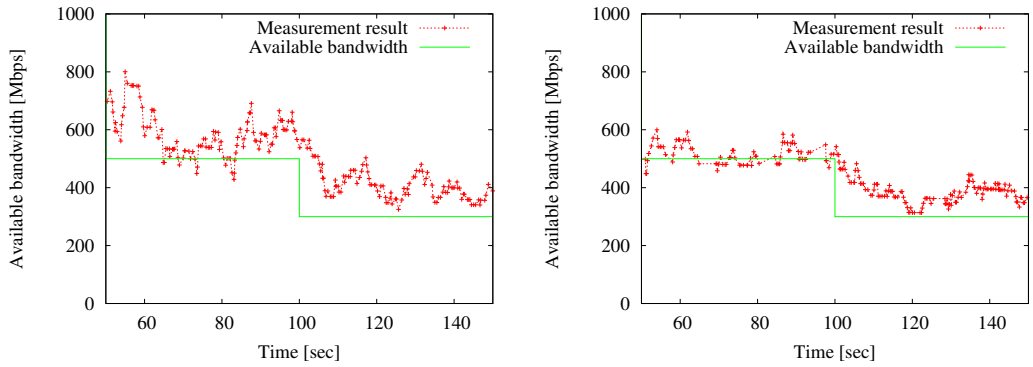
本章では，インラインネットワーク計測の概念に基づく計測手法の実ネットワークにおける有効性について評価を行う．

4.1　低速ネットワークにおける ImTCP の実験結果

本節では，低速ネットワークにおける ImTCP の有効性について評価する．実験ネットワークは，図 4 に示されるようなネットワーク環境であり，$B = 100$ Mbps，$N = 1$ に設定されている．すなわち，実験ネットワーク環境は 100 Mbps のイーサネッ



(a) $HZ = 1,000$　(b) $HZ = 5,000$



(c) $HZ = 10,000$　(d) $HZ = 20,000$

図 6　ICIM による計測結果

トによって構築され，2 台のイーサネットスイッチ介してクロストラヒックを発生させるエンドホスト（Traffic generator），利用可能帯域の計測する送信ホスト（Sender），Sender からのパケットを受信する受信ホスト（Receiver）が接続されている．実験ネットワーク環境を構築する PC のスペックを表 2 に示す．また，クロストラヒックとして UDP トラヒックを発生させる．実験では，発生させるクロストラヒックの量を時間の経過とともに変動させることにより，ボトルネックリンクの利用可能帯域が 0 sec から 30 sec までは 70Mbps，30 sec から 60 sec までは 30 Mbps，60 sec から 90 sec までは 40 Mbps と変化する．

前述したネットワーク環境の下で，*HZ* の値を変化させた時のカーネルシステムの時間粒度と ImTCP の計測精度の関係について観察する．図 5 に *HZ* の値が 1,000，10,000，20,000，50,000 の場合についての計測結果および実際の利用可能帯域の変化を示す．図 5(a) を見ると，*HZ* が 1,000 に設定された場合には，ImTCP は利用可能帯域の計測を行うことができないことが分かる．これは，*HZ* が 1,000 の場合カーネルシステムの時間粒度は 1 msec であり，12 Mbps までの帯域しか計測を行うことができないためである．また，図 5(b) を見ると，利用可能帯域が 70 Mbps の場合に ImTCP はうまく計測することができないことが分かる．*HZ* を 10,000 に設定した場合，計測可能な帯域の上限値は 120 Mbps となる．しかしながら，表 1 から分かるように，計測結果が計測可能な帯域の上限値に近づくにつれて，計測結果の粒度は荒くなる．そのため，実験ネットワーク環境においては $HZ = 10,000$ という設定は，まだ不十分であることが分かる．これに対して，図 5(c)，5(d) から，*HZ* が 20,000 および 50,000 に設定された場合には，ImTCP は利用可能帯域を計測することができることが分かる．したがって，計測精度の観点から見ると，*HZ* は 20,000 以上の値に設定されている必要がある．CPU への負荷を考慮すると，この実験ネットワーク環境の下では，$HZ = 20,000$ が最適なパラメータ設定である．しかしながら，最適なパラメータ設定には，マシンの性能やネットワークの帯域など様々な要因が関わることになる．したがって，*HZ* の値を決定する際には，これらの要因と計測精度との間のトレードオフの問題を考慮して決定しなければならない．

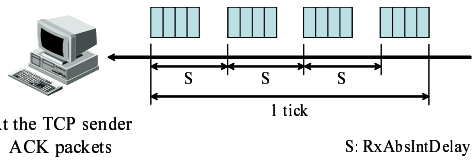


図 7　カーネルシステムの時間粒度が NIC の絶対タイマよりも荒い場合の ACK パケットの到着の様子

4.2　高速ネットワークにおける ICIM の実験結果

本節では，ギガビットネットワーク環境におけるインラインネットワーク計測手法の有効性について評価する．ICIM の基本的な特質に関しては，文献 [7] で明らかにされており，計測精度，他の TCP バージョンとの互換性，他の TCP コネクションとの公平性などに関しては既に確認されている．本節では，ICIM が高速ネットワークにおいて時間粒度が荒い場合においても計測を行うことができることを示す．

実験ネットワークは，図 4 に示されるようなネットワーク環境であり，$B = 1$ Gbps，$N = 2$ に設定されている．すなわち，実験ネットワーク環境は，1 Gbps のイーサネットによって構築され，2 台のイーサネットスイッチ介してクロストラヒックを発生させるエンドホスト（Traffic generator），利用可能帯域の計測する送信ホスト（Sender），Sender からのパケットを受信する受信ホスト（Receiver）が接続されている．Sender および Receiver の NIC には，Intel PRO/1000 アダプタ [14] を用いており，$RxAbsIntDelay = 128$ としている．文献 [14] によ

ると，この値は 131.2 μsec に相当する．また，実験ネットワーク環境を構築する PC のスペックは，表 2 と同様である．また，クロストラヒックとして UDP トラヒックを発生させる．実験では，発生させるクロストラヒックの量を時間の経過とともに変動させることにより，ボトルネックリンクの利用可能帯域が 50 sec から 100 sec までは 500 Mbps，100 sec から 150 sec までは 300 Mbps と変化する．

図 6 に HZ の値が 1,000，5,000，10,000，20,000 の場合についての計測結果および実際の利用可能帯域の変化を示す．これらの図を見ると，ICIM が HZ の値に関わらず利用可能帯域を計測することができることが分かる．表 1 を見ると，パケット間隔に基づく計測手法では，ギガビットネットワークで利用可能帯域を計測するためには，HZ を 100,000 以上に設定する必要がある．したがって，この結果からバースト間隔に基づく計測手法の優位性が明らかになったと言える．

図 6(a) を見ると，$HZ = 1,000$ の場合においても，ICIM は，実際の値よりも高く推測する傾向はあるが，利用可能帯域を計測することができることが分かる．この場合，カーネルシステムの時間粒度は 1,000 μsec になり，$RxAbsIntDelay$ よりも粒度が荒くなる．3. 2. 2 節で説明したように，カーネルシステムの時間粒度が $RxAbsIntDelay$ よりも荒い場合には，ICIM は 2 つのバースト間の間隔を検知することができないため，1 tick（1 tick は $\frac{1}{HZ}$ sec を表す）毎に到着する ACK パケットの数を観察することになる．その結果，ICIM は図 7 に示されるように，1 tick に含まれるデータパケットの数を調節し，それに対応する ACK の数を観察することによって利用可能帯域を推測する．この動作は，Pathload [3] よりも Bprobe [15] に近い手法となる．そのため，文献 [16] で指摘されているように，計測される利用可能帯域は実際の値よりも高くなる傾向がある．

図 6(d) を見ると，$HZ = 20,000$ の場合においても，利用可能帯域が 300 Mbps の時には計測精度が低いことが分かる．これは，利用可能帯域が小さい場合においては，バーストに含まれるパケット数が少ないためである．パケット間隔に基づく計測手法では，データパケットの間隔を調節することによって利用可能帯域を計測する．このとき，利用可能帯域が小さくなるにつれてパケット間隔は大きくなる．そのため，カーネルシステムの時間粒度が同じである場合，利用可能帯域が小さくなるにつれてパケット間隔の調節が容易になり，計測精度が向上する．一方，バースト間隔に基づく計測手法では，バーストに含まれるパケット数を調節することによって利用可能帯域を計測する．しかしながら，利用可能帯域が小さくなるにつれて，バーストに含まれるパケット数は減少する．そのため，利用可能帯域が小さくなるにつれてバーストに含まれるパケット数の調節が難しくなり，計測精度が低下する．

これらの実験結果から，高速ネットワークにおいてインラインネットワーク計測の概念に基づく計測手法を用いて利用可能帯域を計測する場合，バースト間隔に基づく計測手法がパケット間隔に基づく計測手法よりも有効であることが確認できた．利用可能帯域を正確に計測するためには，バースト間隔に基づく計測手法においても HZ の値を高く設定する必要がある．しかしながら，HZ が必要とされている値を満たしていない場合においても，ICIM は，計測精度は低下するが，利用可能帯域を計測することができることを明らかにした．

5. おわりに

本稿では，インラインネットワーク計測手法を実システムへ実装する際の問題点を明らかにし，解決策を示した．さらに，提案手法を実装し，実ネットワーク上で実装実験を行うことによって，インラインネットワーク計測の概念の有効性を確認した．ImTCP を用いた実験では，パケット間隔に基づく計測手法を実装する際のカーネルシステムのパラメータ設定についての指針を示した．また，ICIM の実験から，バースト間隔に基づく計測手法は高速ネットワークにおいても機能することを確認した．さらに，ICIM はカーネルシステムの時間粒度が荒い場合においても，利用可能帯域を計測することができることを示した．これらの実験を通じて，高速ネットワークにおけるパケット間隔に基づく計測手法の限界とバースト間隔に基づく計測手法の優位性を明らかにした．ImTCP および ICIM の実装コードは，`http://www.anarg.jp/imtcp/` で公開している．

今後の課題としては，インラインネットワーク計測の概念に基づく他の帯域手法を提案すること，それらの計測手法をシミュレーションおよび実ネットワーク上で評価を行うことなどが挙げられる．

文　献

[1] R. Prasad, M. Murray, C. Dovrolis, and K. C. Claffy, “Bandwidth estimation: Metrics, measurement techniques, and tools,” *IEEE Network*, vol. 17, pp. 27–35, Dec. 2003.

[2] B. Melander, M. Bjorkman, and P. Gunningberg, “A new end-to-end probing and analysis method for estimating bandwidth bottlenecks,” in *Proceedings of IEEE GLOBECOM 2000*, Nov. 2000.

[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] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell, “pathChirp: Efficient available bandwidth estimation for network paths,” in *Proceedings of Passive and Active Measurement Workshop (PAM 2003)*, Apr. 2003.

[5] A. Shiram, M. Murray, Y. Hyun, N. Brownlee, A. Broido, M. Fomenkov, and K. C. Claffy, “Comparison of public end-to-end bandwidth estimation tools on high-speed links,” in *Proceedings of Passive and Active Measurement Workshop (PAM 2005)*, Mar. 2005.

[6] C. L. T. Man, G. Hasegawa, and M. Murata, “Available bandwidth measurement via TCP connection,” in *Proceedings of IFIP/IEEE Workshop on End-to-End Monitoring Techniques and Services (E2EMON 2004)*, Oct. 2004.

[7] C. L. T. Man, G. Hasegawa, and M. Murata, “ICIM: An inline network measurement mechanism for high-speed networks,” in *Proceedings of IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services (E2EMON 2006)*, Apr. 2006.

[8] R. Prasad, M. Jain, and C. Dovrolis, “Effects of interrupt coalescence on network measurements,” in *Proceedings of Passive and Active Measurement Workshop (PAM 2004)*, Apr. 2004.

[9] Intel, “Interrupt moderation using intel gigabit ethernet controllers.” available at `http://www.intel.com/design/network/applnots/ap450.pdf`.

[10] Syskonnect, “SK-NET GE Gigabit Ethernet Server Adapter.” available at `http://www.syskonnect.com/syskonnect/technology/SK-NET_GE.PDF`.

[11] G. Jin and B. L. Tierney, “System capability effect on algorithms for network bandwidth measurement,” in *Proceedings of Internet Measurement Conference (IMC 2003)*, Oct. 2003.

[12] T. Tsugawa, G. Hasegawa, and M. Murata, “Implementation and evaluation of an inline network measurement algorithm and its application to TCP-based service,” in *Proceedings of IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services (E2EMON 2006)*, Apr. 2006.

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

[14] Intel(R) PRO/1000 Adapter, “README file.” available at `http://support.intel.co.jp/jp/support/network/adapter/1000/linux_readme.htm`.

[15] R. L. Carter and M. E. Crovella, “Measuring bottleneck link speed in packet-switched networks,” *International Journal on Performance Evaluation*, vol. 27–28, pp. 297–318, Oct. 1996.

[16] C. D. P. Ramanathan and D. Moore, “What do packet dispersion techniques measure ?,” in *Proceedings of IEEE INFOCOM 2001*, Apr. 2001.