



インライン計測手法 ImTCP および その応用手法の実装および性能評価

津川 知朗 < t-tugawa@ist.osaka-u.ac.jp >

長谷川 剛 < hasegawa@cmc.osaka-u.ac.jp >

村田 正幸 < murata@ist.osaka-u.ac.jp >

大阪大学 大学院情報科学研究科

研究の背景と目的

- ◆ インターネットの発展に従いサービスが多様化してきている
 - ✦ ex. CDN, P2P, Grid, IP-VPN, ...
 - ◆ IP ネットワークの資源状況を把握することが重要となってくる
 - ✦ 資源を有効に利用できるようになる
 - ✦ ネットワークサービスの品質を向上させることができる
 - ◆ インライン計測手法 ImTCP およびその応用手法を提案した
 - ✦ コンピュータ上のシミュレーションによって有効性を評価した
 - ◆ 有効性の評価には実ネットワークでの実験が不可欠である
- 
- ◆ 提案手法の実ネットワーク上での有効性を評価する
 - ✦ ImTCP, ImTCP-bg の実装を行う
 - ◆ FreeBSD 4.10 のカーネルシステムへ実装する
 - ✦ 実験ネットワークを用いた実装実験を行う

ImTCP 概要

- ◆ 送受信ホスト間の利用可能帯域の計測を行う
- ◆ TCP コネクションのデータ・ACKパケットを利用する
 - ⊕ 計測アルゴリズムによりデータパケットの送信間隔を調節する
 - ⊕ ACK パケットの受信間隔から利用可能帯域を導出する
 - ➡ 余計な計測パケットを必要とせずに計測を行うことができる
- ◆ 過去の計測結果をもとに探索区間を設定する
 - ⊕ 過去の計測結果から 95% 信頼区間を導出し次回の探索区間とする
 - ➡ 不必要に高い転送レートでパケットを送出することを避けられる
 - ➡ 計測精度を保ちながら必要なパケット数を減少させることができる
- ◆ 探索区間内に利用可能帯域が存在しない場合もある
 - ⊕ 利用可能帯域が急激に変化した場合など
 - ⊕ このような場合でも数回の計測で新たな利用可能帯域を発見できる

[8] Cao Le Thanh Man, Go hasegawa, and Masayuki Murata, ``Available bandwidth measurement via TCP connection,’’ in Proceedings of IFIP/IEEE MMNS 2004 E2EMON Workshop, Oct. 2004.

ImTCP-bg 概要

◆ ImTCP の計測結果に基づくバックグラウンド転送

1. 計測結果に対して平滑化を行う

$$\bar{A} \leftarrow (1 - \gamma) \times \bar{A} + \gamma \times A_{cur} \quad A_{cur} : \text{最新の利用可能帯域}$$

2. 送信側のTCPの輻輳ウィンドウサイズの上限值を決定する

$$maxcwnd = \bar{A} \times RTT_{min} \quad RTT_{min} : RTT \text{ の最小値}$$

◆ ImTCP の計測結果が信頼できない場合がある

➡ RTT を用いた輻輳ウィンドウサイズの制御を併用する

- ⊕ RTT の現在値・最小値から輻輳を検知する

$$\frac{RTT_{cur}}{RTT_{min}} > \delta \quad \delta : \text{しきい値} (1 \leq \delta)$$

$RTT_{cur}, RTT_{min} : RTT \text{ の現在値, 最小値}$

- ⊕ RTT の値に応じて輻輳ウィンドウサイズを減少させる

$$cwnd \leftarrow cwnd \times \frac{RTT_{min}}{RTT_{cur}}$$

[12] Tomoaki Tsugawa, Go Hasegawa, and Masayuki Murata, "Background TCP data transfer with inline network measurement," in Proceedings of APCC 2005, Oct. 2005.

ImTCP・ImTCP-bg 実装方針

- ◆ ImTCP, ImTCP-bg は TCP 層の下部に実装する
 - ✦ TCP コネクションが保持する情報を利用するため
- ◆ ImTCP バッファを作成する
 - ✦ カーネルシステムによって確保されたメモリを用いて作成する
 - ✦ TCP プロトコル処理が終了したパケットを格納する
 - ➡ 計測アルゴリズムに基づいたタイミングで IP 層へ渡す
 - ◆ カーネルシステムのスケジューリング機能を用いて実現する
- ◆ パケットの送信時刻・受信時刻を記録する
 - ✦ データパケットを IP 層へ渡した時刻
 - ✦ IP 層から ACK を受け取った時刻
- ◆ 輻輳ウィンドウサイズは ACK を受信するたびに更新される
- ➡ ImTCP-bg は ACK 処理を行う部分に実装する

実装上の問題点

- ◆ カーネルシステムのタイマ粒度が粗いため計測精度が低下
- ◆ FreeBSD 4.10 では *HZ* パラメータの値によって粒度を変更できる
 - ✦ *HZ*=100 の場合 1.2Mbps まで計測可能
 - ✦ *HZ*=100,000 の場合 1.2Gbps まで計測可能
- ◆ *HZ* が大きすぎると実行速度に影響を与える
 - ✦ タスク切り替えが頻発するようになるため
 - ✦ 例:カーネルプログラムに要した処理時間の比較

カーネルプログラムのコンパイルに要した処理時間 (※1)

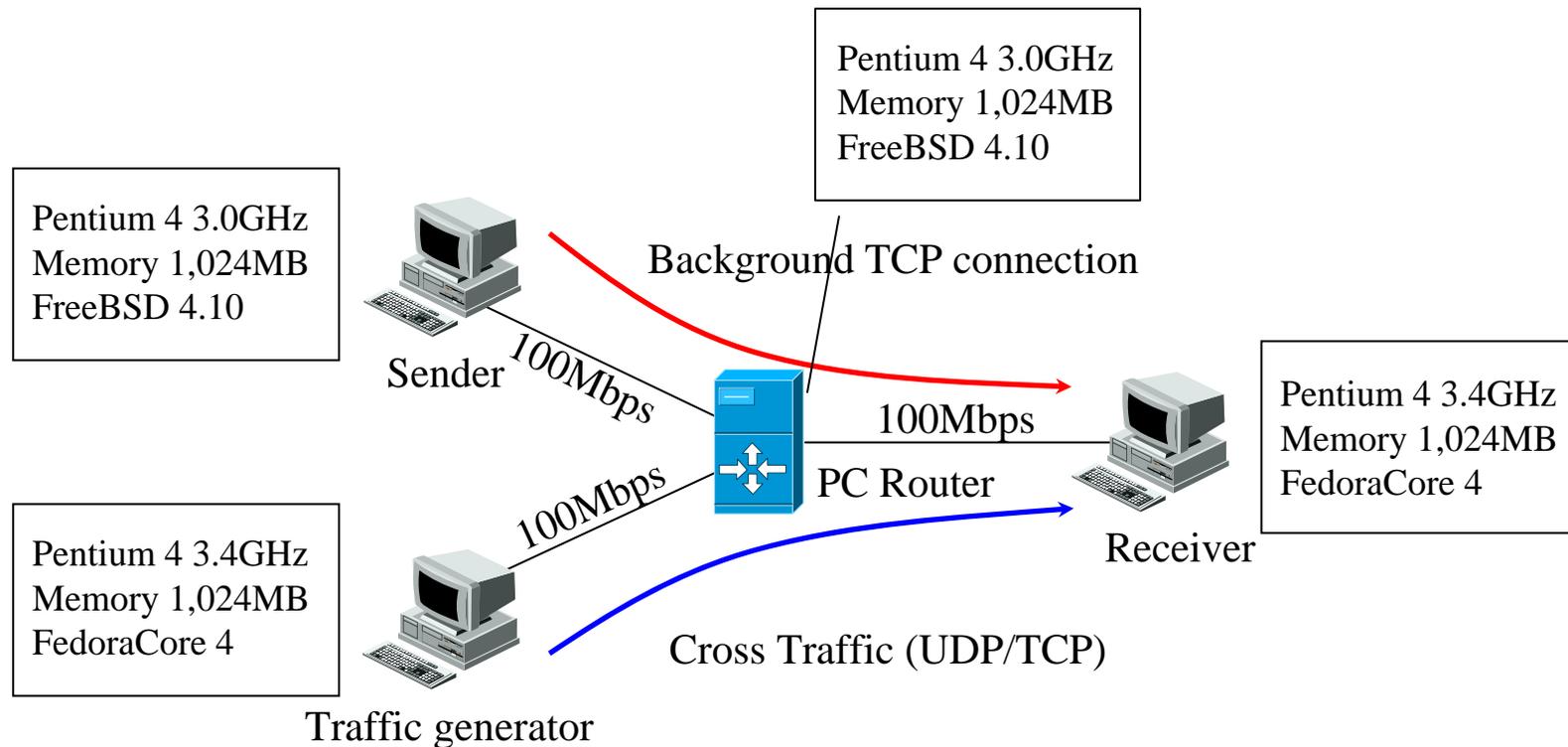
<i>HZ</i>	処理時間 [sec]
100	168.20
1,000	170.09
10,000	183.38

<i>HZ</i>	処理時間 [sec]
20,000	199.78
50,000	277.84
100,000	734.10

- ➡ 計測精度・実行速度を考慮して *HZ* を決定する必要がある

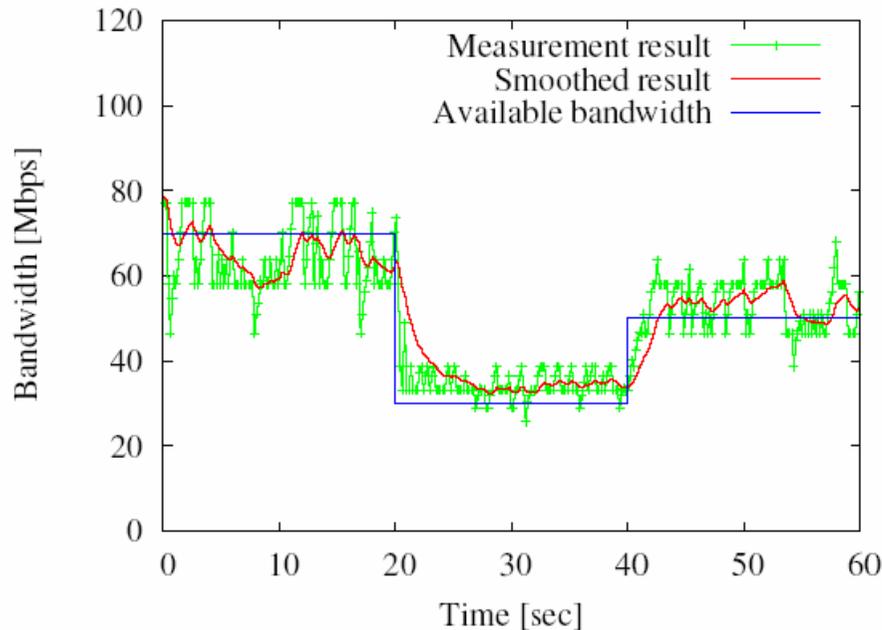
※1 Pentium 4 3.0GHz, Memory 1,024MB, FreeBSD 4.10 の PC を用いて行った

実験ネットワーク環境

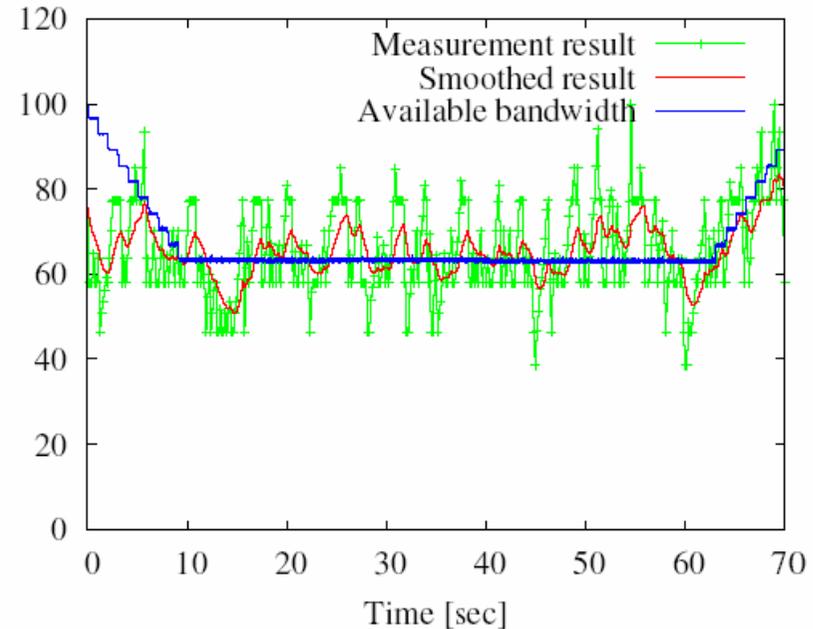


- ◆ DUMMYNET を用いて遅延時間を発生させている
 - ⊕ RTT の最小値が 30msec
- ◆ ImTCP・ImTCP-bg パラメータ設定
 - ⊕ 1 回の計測あたりのストリーム数 : 4, 1 ストリームあたりのパケット数 : 10
 - ⊕ 平滑化パラメータ (γ) : 1/8, RTT Threshold (δ) : 1.4

ImTCP の計測精度と CPU へかかる負荷



クロストラフィックがUDPトラフィックの場合



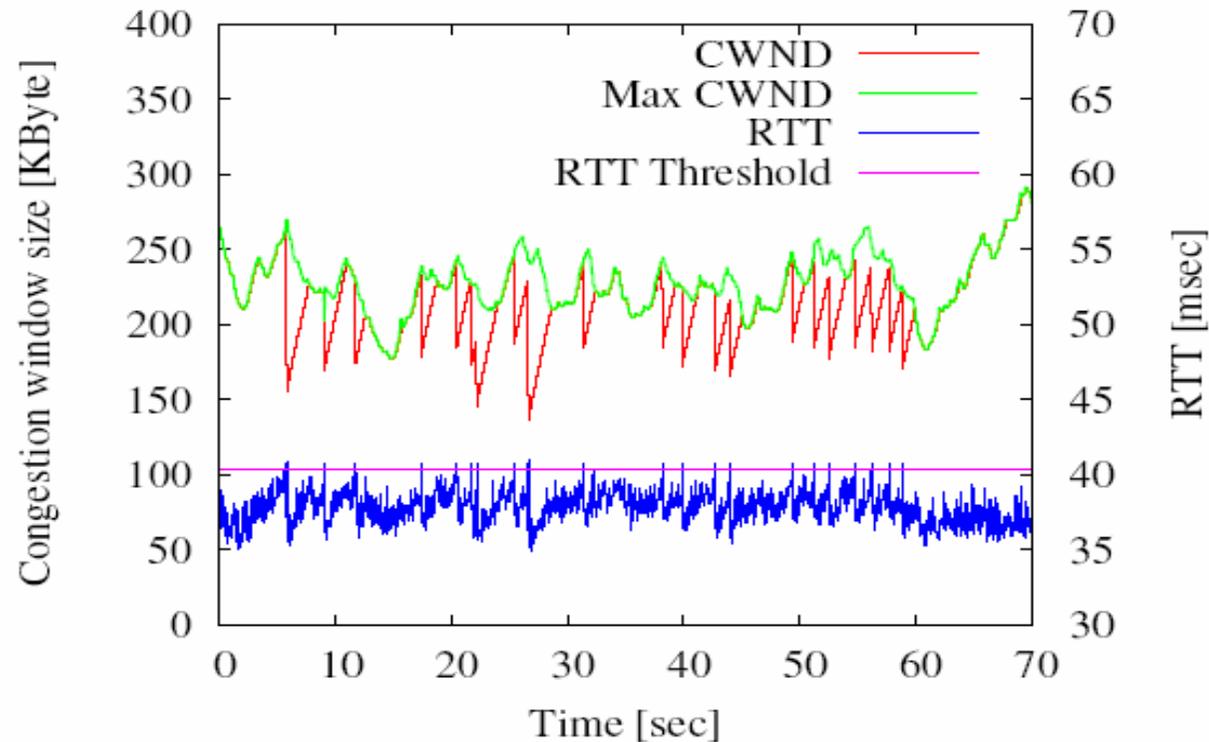
クロストラフィックがTCPトラフィックの場合

CPU へかかる負荷の比較 (HZ=20,000)

	TCP Reno	ImTCP
平均負荷 [%]	18.62	19.12

- ◆ クロストラフィックがどちらの場合でも高い精度で計測することができる
- ◆ CPU へかかる負荷は TCP Reno と比較してほとんど差が見られない
- ➡ ImTCP の計測アルゴリズムは CPU へ大きな負荷を与えずに実現できる

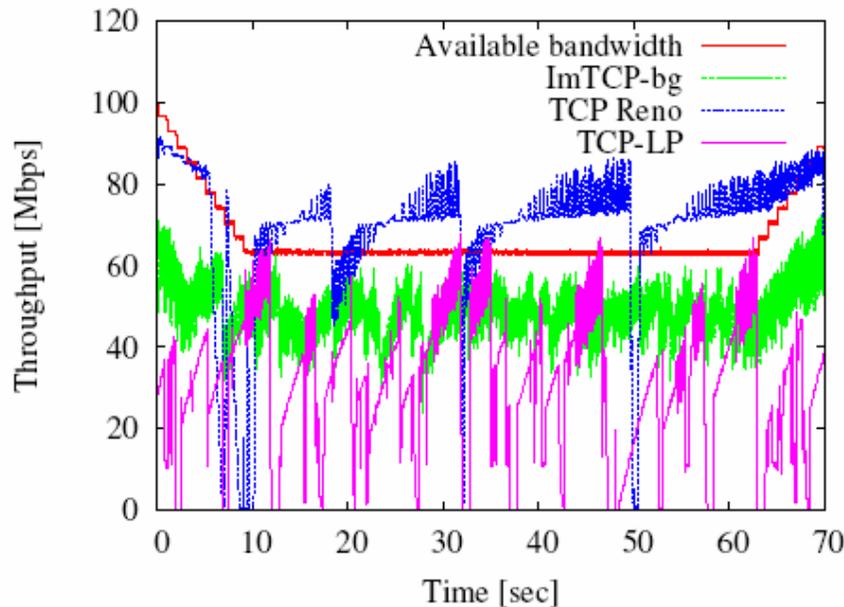
ImTCP-bg の動作確認



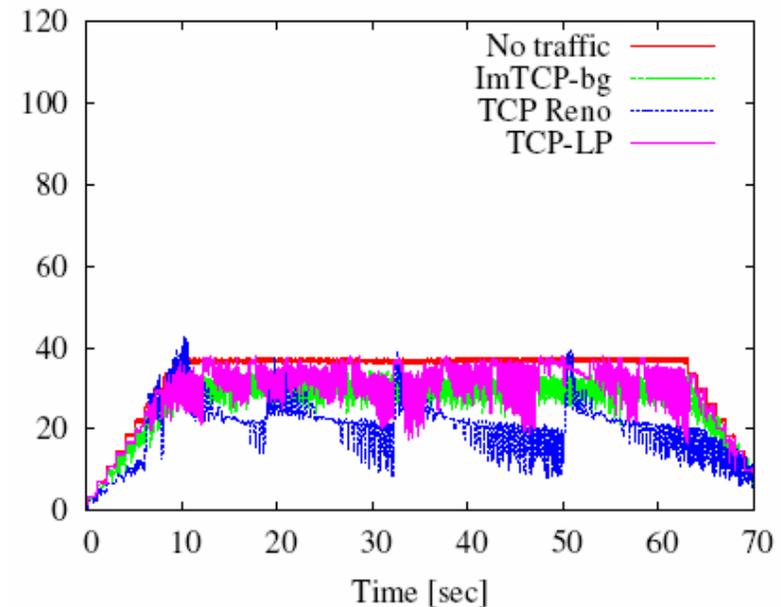
輻輳ウィンドウサイズおよび RTT の変化

- ◆ 輻輳ウィンドウサイズが ImTCP の計測結果を用いて設定された上限値によって抑えられている
- ◆ RTT がしきい値を超えた場合に輻輳ウィンドウサイズを減少させている
- ◆ ImTCP-bg のアルゴリズムに基づいて正しく動作している

ImTCP-bgの性能評価



バックグラウンド転送 TCP のスループットの変化



クロストラヒックのスループットの変化

- ◆ TCP Reno を用いた場合得られたスループットが最も高い
 - ⊕ 空き帯域を大きく越えておりクロストラヒックのスループットも押し下げている
 - ⊕ バックグラウンド転送を行っているとは言えない
- ◆ ImTCP-bg, TCP-LP を用いた場合クロストラヒックのスループットをほとんど押し下げない
- ◆ ImTCP-bg を用いた場合空き帯域に近いスループットを得られている