社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－HPC－116 (29)
2008／8／7

# 並列TCP ストリーム間協調を目的とした流量調整機構 Stream Equalizer の性能評価

菅原 豊† 吉野剛史‡ 手塚宏史† 稲葉真理† 平木敬†

**要 旨**

近年，グリッド・クラウドコンピューティングでは広帯域・高遅延ネットワーク上で並列TCP ストリームを用いたデータ転送が行われる．現状の TCP 輻輳制御方式では，スロースタート時に並列 TCP ストリームを 1 本のリンクにまとめるスイッチにおいて多数のパケットロスが発生し性能が大きく低下する．本研究では，適切なタイミングでスロースタートフェイズを終了させることにより流量の低下を抑制可能である事を示す．この点を実証するため，ネットワーク経路上でパケットを落とすことでスロースタートを終了させる機構を実装する．評価より，本機構を用いることで最終的なパケットロス数が減少し，流量の低下を抑制できることを示す．

# Performance Evaluation of a Stream Equalizer – A Flow Adjustment Mechanism for Coordination of Parallel TCP Streams

Yutaka Sugawara† Takeshi Yoshino‡ Hiroshi Tezuka† Mary Inaba†
Kei Hiraki†

**Abstract**

In recent grid or cloud computing, data transfer is performed using parallel TCP streams over high-bandwidth and long-latency network. Using recent TCP congestion control algorithms, many packets are lost at the switch that merges parallel TCP streams into one link, and the performance is drastically reduced. In this research, we show that the flow reduction can be reduced by terminating the slow start phase at appropriate timing. To demonstrate it, we implement a mechanism to drop packets at the network path to terminate slow start phase. From evaluations, we show that the resulting number of packet losses is reduced and flow reduction is reduced when this mechanism is used.

## 1 はじめに

近年，ネットワーク技術が進歩し 10 ギガビット・イーサネット (10GbE) や OC-768 などの高速な広域ネットワークが実用化されている．また，グリッド・クラウドコンピューティングなどの新しい応用が登場し，ネットワークの利用形態が変化している．世界規模で分散化したシステムではパケット往復時間 (Round Trip Time, RTT) が数百 ms に及ぶような長距離の通信が行われる場合がある．また，クラスタの普及により複数の協調する TCP ストリーム (並列 TCP ストリーム) を用いたデータ転送がしばしば行われる．

このような広帯域で遅延が大きいネットワーク (Long Fat-pipe Network, LFN) を用いる状況で高い流量を保つことができるような TCP の輻輳制御アルゴリズムが提案されてきた．以前広く用いられていた輻輳制御アルゴリズム TCP Reno では RTT あたりの in-flight データサイズの増え方が遅く，LFN における性能に限界があった．LFN では遅延帯域積が大きいことから高いリンク利用率を保つために必要な in-flight データサイズが大きい．輻輳をでき

†東京大学 情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo
‡グーグルジャパン
Google Japan Inc.

るだけ避けながらより積極的に in-flight データサイズを増やす輻輳制御方式が検討されてきた．現在までに FAST TCP [4], HighSpeed TCP [2], Scalable TCP [5], BIC [7], CUBIC [6] などの輻輳制御アルゴリズムが提案されている．近年の Linux で採用されている BIC や CUBIC では流量の増加関数を工夫することで高い流量増加率とネットワークへの負荷削減を両立するよう設計されている．これらの輻輳制御アルゴリズムの改善により LFN における単一ストリームのスループットが改善された．

しかし，LFN における並列 TCP ストリームの性能には依然として限界がある．現状の輻輳制御方式とスイッチの組み合わせをそのまま用いた場合，複数ホストからのリンクを 1 本のアップリンクへ収容する送信側エッジスイッチにおいて多数のパケットロスが発生することが観察される．その結果，TCP の流量が大幅に低下する．TCP 送信側は ACK からデータパケット不着を判定しているため，スイッチでパケットロスが発生してから実際に流量が減り始めるまで 1RTT の遅延が生じる．そのためパケットロスが連続して発生する．並列 TCP ストリームではシングルストリームと違い送信側エッジスイッチの入力リンク容量の合計が出力リンク容量を上回る．微視的な時間で見ると個々の送信ホストからほぼリンク容量近く達するレートでパケットが送られることで出力リンク容量があふれ，パケットが落ちる傾向がみられる．

この現象は特にスロースタートフェイズで問題になる．TCP Reno や BIC など現在の主要な輻輳制御方式は，通信開始時などで流量を指数関数的に急速に増やすスロースタートフェイズと，輻輳発生の後などでより保守的に流量を増やす輻輳回避フェイズの 2 つのフェイズを持つ．スロースタートフェイズでパケット不着が検出されると輻輳回避フェイズに移行する．輻輳回避フェイズでは輻輳を緩和するためより保守的に流量を調整する．遅延が大きく ACK 到着が遅い LFN では輻輳回避フェイズでの流量変化が遅く，スロースタート直後の状態が長期に渡り影響を与える．スロースタート直後に大きく流量が低下すると回復が遅く，長期に渡りリンク利用率が低下する．スロースタートフェイズでは流量の時間あたりの増加率が大きいためエッジスイッチにおいて並列 TCP ストリームのパケットロスが発生しやすく，この問題が顕著に現れる．

本研究では，LFN に並列 TCP ストリームを流した場合にスロースタートフェイズで発生するパケットロスを減らす方法を議論する．パケットロスを減らすことで TCP の流量を改善することを目的とする．大規模なパケットロスが発生する直前に人為的

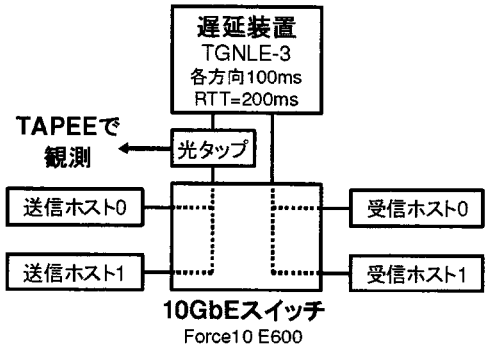


図 1: 実験用ネットワークの構成

に少数のパケットロスを発生させることで，スロースタート，すなわち指数関数的な流量の増加を停止させる．この方法により最終的なパケットロス数を削減可能であることを示す．この点を実証するために，スロースタートを終了させる目的で意図的に少数のパケットを破棄するハードウェア機構を導入する．この機構はネットワークの中間経路に配置される．流量の増加率を監視して一定値を超えた時点でパケットを落としスロースタートを終了させる．本機構をネットワークテストベッド TGNLE-1 に実装し，擬似遅延装置からなる実験環境において評価する．評価の結果より，パケット破棄機構により最終的なパケットロス数が削減される事，その結果 TCP の流量が改善されることを実証する．

本稿では，まず 2 章で LFN における並列 TCP の振る舞いを調べる．この観察結果を元に 3 章でパケット破棄機構を導入する．4 章ではパケット破棄機構をネットワークテストベッド TGNLE-1 を用いて実装し，擬似遅延装置を用いた実験用ネットワークにおいて評価する．5 章では関連研究について述べる．最後に 6 章でまとめを行う．

## 2 LFN における並列 TCP ストリームの振る舞い

10GbE と遅延発生装置からなる実験環境を用いて並列 TCP ストリームのスロースタート時における振る舞いを解析する．評価に用いるネットワークを 図 1 に示す．図中に示すリンクは全て 10GbE である．送信ホスト 2 台と受信ホスト 2 台の間で並列 TCP ストリームを用いた通信を行う．エンドホストの仕様を表 1 に示す．1 ホスト毎に 1 ストリーム，計 2 ストリームを張る．ネットワークの途中経路に

表 1: エンドホストの仕様

| | |
|---|---|
| CPU | Intel Xeon 5160 3.0GHz × 2 |
| OS | Linux 2.6.25.7 (x86_64) |
| 輻輳制御 | BIC, SACK あり |
| NIC | Chelsio T310 (TOE なし) |
| ドライバ | cxgb3 (カーネル付属) |

表 2: iperf のコマンドライン引数

| | |
|---|---|
| 送信側 | -i 1 -c サーバ名 -w 300M -M 8970 -l 192k |
| 受信側 | -i 1 -s -w 600M -M 9150 -l 192k |

TGNLE-3 を挿入する．TGNLE-3 は FPGA(Xilinx Virtex4 XC4VFX60) と 10GbE インタフェース 2 ポートを持つネットワーク実験装置である．片方のポートから受信されたパケットは FPGA で処理されてもう片方のポートから送信される．FPGA ロジックを書き換えることにより様々な機能をワイヤレートで実現可能である．また，各方向毎に 2GB のメモリを持つ．本稿では TGNLE-3 のメモリを用いてパケットをバッファリングし遅延を発生させる．遅延は片方向 100ms，往復 200ms に設定する．この機能により長距離ネットワークの遅延を再現する．

スイッチ Force10 E600 を用いて遅延装置，4 台のホストを接続する．送信ホストから出たパケットは VLAN 設定により一度 E600 の外に出て TGNLE-3 を通過してから再び E600 に戻り受信ホストに到達する．逆方向のトラフィックも同様に TGNLE-3 を通過する．スイッチ E600 と TGNLE-3 送信ホスト側ポートを結ぶリンクに光タップを挿入しトラフィックを観測する．光タップから分岐したファイバを TAPEE [8] に接続しパケットヘッダと時刻を記録する．TAPEE は 10 ギガビット・イーサネット上のトラフィックをワイヤレートで観測するためのハードウェア機構である．受信したパケットからパケットヘッダのみを抽出しタイムスタンプを付加し記録用ホストに送る．ヘッダのみが抽出されデータサイズが削減されるため，記録用ホストでデータを取りこぼさずにリアルタイムで受信可能である．

iperf を用いて TCP 通信を行う．iperf は mmap によりユーザ空間からカーネル空間へのコピーを省いたバージョン [1] を用いた．iperf のコマンドライン引数を送信側，受信側それぞれ表 2 に示す．

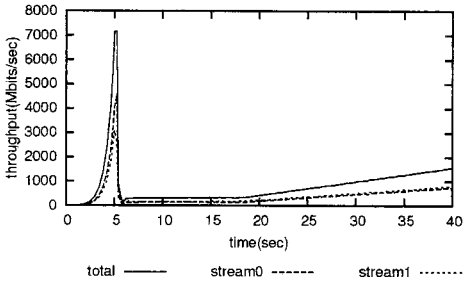


図 2: 各ストリームおよび全体の流量，1RTT(200ms) 区間ごとの平均

## 2.1 TCP 流量の変化

図 2 に TCP 流量の時間変化を示す．流量は 1 RTT(200ms) 区間ごとの平均流量である．太線が 2 本のストリームの合計流量，2 本の点線が各ストリームの流量を示す．5 秒付近までスロースタートフェイズのため流量が増加している．5 秒付近で流量が約 500Mbps に落ちている．20 秒付近まで低下した流量が維持されている．以上のようにスロースタートの終了時に流量が急激に落ち，その後の流量回復が遅いという現象が観測された．その結果，リンク帯域の利用効率が低下している．

## 2.2 パケットロスの解析

スロースタート終了後に流量が低下している箇所におけるパケットロスを解析するため，TAPEE で観測されたパケット列に対してシーケンス番号のとびを調べた．また，スイッチのバッファ溢れの状態を推測するため微視的な流量をプロットした．図 3 に結果を示す．2 本の線が 2 ストリームそれぞれの微視的な流量を示す．ここで示す流量は 1ms 区間ごとの平均値である．また，図中＋印と×印は各ストリームでシーケンス番号のとびが観測された時刻を示す．シーケンス番号のとびについては横軸の時刻のみが意味をもつ．

図から，2 本のストリームが同時にパケットを送っている時間帯にパケットロスが発生していることが示されている．5.15 秒付近で最初のパケットロスが発生している．ここでは 2 本のストリームが同時にパケットを送っている．他の箇所ではいずれのストリームも微視的な流量が 10Gbps に達しているのに対してこの時間帯は 10Gbps に達していない．この

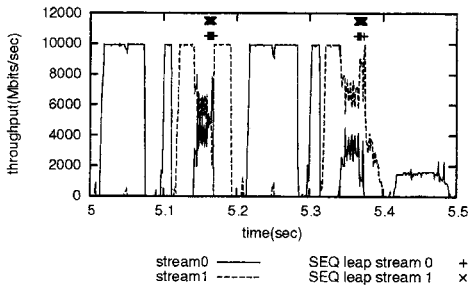


図 3: 各ストリームの微視的な流量 (1ms 区間ごとの平均) とシーケンス番号のとびが観測された時刻

観察から，リンク容量を超えるパケットはバッファリングされ溢れた分が落とされたと推定される．最初のパケットロスが生じてからも流量は減少せず，続けてパケットが落ちている．約 1RTT 後の 5.35 秒付近では再び 2 本のストリームが同時にパケットを送りパケットロスが発生している．その後送信側の流量が下がり始め，パケットロスが終了する．シーケンス番号のとびから推定した結果，2 ストリーム合計で 262 パケット落ちていることが分かった．

以上のようエッジスイッチで 2 本のストリームが同時にトラフィックを送るとパケットロスが発生すること，パケットロスが発生してから実際に流量が減るまで 1RTT 前後の遅延がある事が観測された．その結果多数のパケットが落ちている事が分かった．

# 3 パケット破棄によるスロースタート停止

2 章では，LFN で一度リンクが溢れはじめると実際に流量が絞られるまでに遅延が生じ，多数のパケットが落ちることが観測された．もし，このように多数のパケットが落ちる直前にスロースタートフェイズを抜けて流量の増加を緩和すれば，パケットロスを減らせる可能性がある．本稿ではこの点を実証するため，意図的にパケットロスを起こすことでスロースタートを適切なタイミングで終了させる機構を導入する．本機構は図 4 のように送信側エッジスイッチの出口に挿入して用いる．

本稿では近似的に TCP 流量の増加率からスロースタートの終了時期を決める．エッジスイッチのアップリンクで全ストリームの合計流量が 10Gbps に絞られるので，対応する ACK 番号の進み方も合

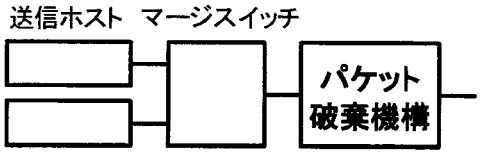


図 4: スロースタートを終了させるためのパケット破棄機構

計で 10Gbps 分になる．したがって，ウィンドウサイズが前の RTT から増加しなければ入力される微視的な流量が 10Gbps を越えることはない．ウィンドウサイズが RTT 毎に増加する場合，この増加分が 10Gbps に加わる．この増加分がスイッチのバッファサイズを越えるとバッファが溢れる可能性がある．本稿ではウィンドウサイズを実際に観測された 1 RTT あたりの流量で近似し，1 RTT あたりの流量増加率からスロースタートの終了時期を決める．本稿の実験環境ではこの方法でパケットロスの直前時期を判定できることを確認した．しかし，理論的にはこの方法の成否はバーストのパタンに依存する．今後，実際の TCP でバーストのパタンが決まるメカニズムなどを分析した上でより的確な方式を検討する予定である．

流量の増加率が一定の値を超えた時点でスロースタートを終了させる．各ストリームから 1 パケットずつを落とすことでスロースタートを終了させる．1 パケットだけを落とすことにより性能低下を最小限に抑えることを目的としている．

## 3.1 TGNLE-1 を用いた実装

TGNLE-1 を用いてパケット破棄機構を実装する．TGNLE-1 は FPGA と 10 ギガビット・イーサネットを搭載したテストベッドである．図 5 にブロック図を示す．片方のポートから受信されたパケットは FPGA により処理された後，もう片方のポートに出力される．本稿では TGNLE-1 を用いて，2 つのポートの間を流れるトラフィックに対して (1) 流量の増加率からパケットを破棄する時期を決める機能，(2) 各ストリームから 1 パケットずつ落としてスロースタートを終了させる機能，の 2 つを実装する．

現時点から過去 1 RTT (200ms) までの平均流量をその直前の RTT の平均流量と比較し，増加率が一定の閾値を超えた場合にパケットを落とす．本稿ではハードウェアを簡単にするため，このチェックを 1/4 RTT 毎のみに行う実装とする．また，今回

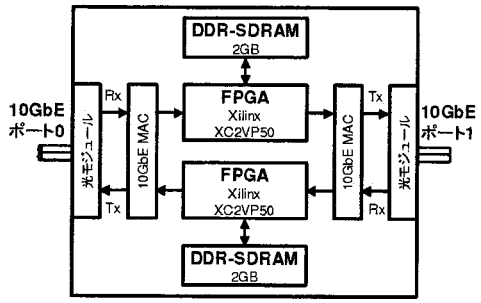


図 5: ネットワークテストベッド TGNLE-1

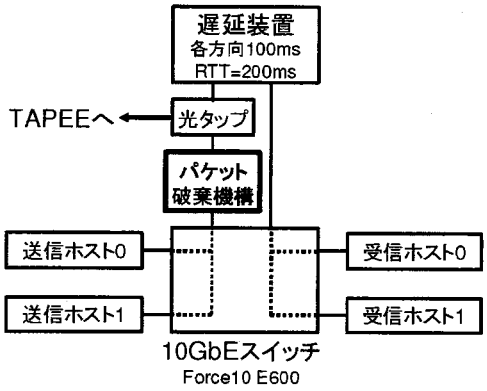


図 6: パケット破棄機構の実験用ネットワーク構成

の実験では送出されるデータパケットサイズは一定なので，実装を簡単にするために流量ではなく単位時間あたりのパケット数から判定を行う．

# 4 評価

TGNLE-1 を用いたパケット破棄機構を実際に適用し並列 TCP ストリームの振る舞いを解析する．2 章で実験に用いたネットワークにパケット破棄機構を挿入する．図 6 のように E600 スイッチの送信側アップリンクポートにパケット破棄機構を挿入する．エンドホスト，スイッチ，遅延装置などの設定は全て 2 章と同じである．

評価に用いる増加率の閾値を実験的に求めた．パケット破棄機構を入れない状態で何度か実験した結果，パケットロスが起きた時点の 1 + 1/4 RTT 前で観測された流量の増加率は最小で約 1750 (パケット/RTT$^2$) であった．そのため，今回の実験では 1750 (パケット/RTT$^2$) を閾値として用いた．こ

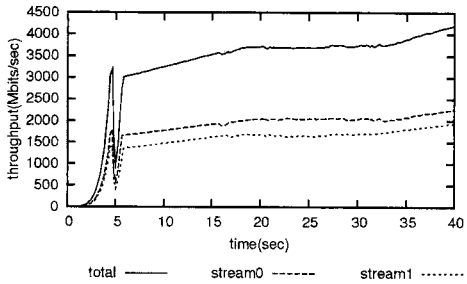


図 7: パケット破棄したときの各ストリームおよび全体の流量，1RTT(200ms) 区間ごとの平均

こで 1 RTT 前ではなく 1+1/4 RTT 前の値を見た理由は，閾値の判定を 1/4 RTT 毎に行うことから最悪の場合 1/4 RTT の遅れが発生するためである．

## 4.1 TCP 流量の変化

図 7 に 1RTT 区間ごとの TCP 平均流量を示す．開始後 5 秒付近までスロースタートフェイズのため流量が増加する．ただし，パケット破棄機構によりパケットが破棄されるため，破棄しない場合 (図 2 と比べて低い流量でスロースタートが終了している．その後再送が始まり一時的に流量が落ちるが，すぐに 2 ストリーム合計約 3Gbps まで回復している．破棄しない場合は 2 ストリーム合計で約 500Mbps であった．このように少数のパケットを予め落とすことでスロースタート直後の流量が回復すること，その結果リンク利用率が改善される事が分かった．

## 4.2 パケットロスの解析

スロースタート終了時刻付近でのパケットロスを解析するため，微視的な流量と観測されたパケット列のシーケンス番号のとびを調べた．図 8 に結果を示す．2 本の線が 2 ストリームそれぞれの 1ms 毎の平均流量，＋印と×印がシーケンス番号のとびが観測された時刻を示す．

図は，各ストリームからパケットは 1 個ずつ，合計 2 個のみ落ちた事を示している．この 2 個のパケットはパケット破棄機構により落とされたものである．この時刻付近ではこの 2 個以外にシーケンス番号のとびは観測されていない．この結果より，少数のパケットを予め落としておくことで大規模なパ

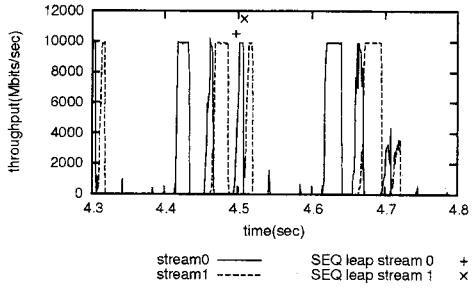


図 8: パケット破棄したときの各ストリームの微視的な流量 (1ms 区間ごとの平均) とシーケンス番号のとびが観測された時刻

ケットロスを回避可能である事が分かった.

## 5 関連研究

LFN で TCP の流量を適切に制御できるよう，輻輳制御アルゴリズムやスイッチのバッファ管理方式が提案されてきた．LFN に適した TCP 輻輳制御方式としては，FAST TCP, Scalable TCP, HighSpeed TCP, BIC, CUBIC 等が提案されてきた．これらのアルゴリズムにより，LFN で単一の TCP ストリームを流した場合の性能が改善された．しかし，本稿で議論したように並列 TCP ストリームを扱う場合はスロースタートフェイズにおいて多数のパケットロスが生じることが課題である．

スイッチのバッファ管理方式としては，Random Early Detection (RED) [3] など Active queue management の手法が提案されている．これらの手法では，バッファ管理の方法を工夫して早期にパケットを落とし，最終的なパケットロス数を削減することを目的としている．これらの方式では実際にリンクが溢れてからパケットロスの制御が始まる．一方，本研究では予測的にパケットを落とす方法について議論した．

## 6 おわりに

本研究では，LFN 上の並列 TCP ストリームがスロースタート時に多数のパケットロスを起こし性能が低下する現象について議論した．10GbE ネットワークに擬似的に遅延を発生させるハードウェアを挿入し LFN の遅延をモデル化した実験を行った．実験を通して，LFN 上の並列 TCP ストリームではパケットロス発生から流量が減少するまで 1 RTT 前後の遅延を要すること，その結果多数のパケットが落ちて性能が大幅に低下する事を示した．次に，適切なタイミングに意図的に少数のパケットを落とすことでスロースタートを終了させる機構を導入した．この機構を入れることによりスロースタートでの大規模なパケットロスを回避可能であること，その結果 TCP の流量が改善される事を示した．この結果より，適切なタイミングでスロースタートを終了させる事で LFN における並列 TCP の性能を改善可能である事が示された．

本稿の結果は，現状の輻輳制御アルゴリズムのスロースタート制御に改善の余地があることを示唆している．また，輻輳制御アルゴリズムだけではなく，スイッチ側で予測的にパケットを落とすことで大規模なパケットロスを防げる可能性を示している．以上の点を踏まえ，今後は LFN に適したスロースタート制御方式について検討を行う予定である．

## 参考文献

[1] http://data-reservoir.adm.s.u-tokyo.ac.jp/press-20070508/.

[2] Floyd, S.: HighSpeed TCP for Large Congestion Windows (2003), RFC3649.

[3] Floyd, S. and Jacobson, V.: Random Early Detection Gateways for Congestion Avoidance, *IEEE/ACM Transactions on Networking (TON)*, Vol. 1, pp. 397–413 (1993).

[4] Jin, C., Wei, D. X. and Low, S. H.: FAST TCP: motivation, architecture, algorithms, performance, in *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004)*, Vol. 4, pp. 2490–2501 (2004).

[5] Kelly, T.: Scalable TCP: Improving Performance in Highspeed Wide Area Networks, in *First International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet 2003)* (2003).

[6] Rhee, I. and Xu, L.: CUBIC: A New TCP-Friendly High-Speed TCP Variant, in *Third International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet 2005)* (2005).

[7] Xu, L., Harfoush, K. and Rhee, I.: Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks, in *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004)*, Vol. 4, pp. 2514–2524 (2004).

[8] Yoshino, T., Tamatsukuri, J., Inagami, K., Sugawara, Y., Inaba, M. and Hiraki, K.: Analysis of 10 Gigabit Ethernet using Hardware Engine for Performance Tuning on Long Fat-pipe Network, in *Fifth International Workshop on Protocols for FAST Long-Distance Networks (PFLDnet 2007)* (2007).