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.

TCP プロキシ機構の実ネットワーク上での実装評価

山根木果奈† 浜 崇之†† 長谷川 剛††† 村田 正幸†††† 下西 英之†† 村瀬 勉††

† 大阪大学 基礎工学部 〒 560-8531 大阪府豊中市待兼山町 1-3
†† NEC システムプラットフォーム研究所 〒 211–8666 神奈川県川崎市中原区下沼部 1753
††† 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
†††† 大阪大学 大学院情報科学研究科 〒 565-0871 大阪府吹田市山田丘 1-5
E-mail: yamanegi@ics.es.osaka-u.ac.jp, t-hama@cb.jp.nec.com, hasegawa@cmc.osaka-u.ac.jp,
murata@ist.osaka-u.ac.jp, h-shimonishi@cd.jp.nec.com, t-murase@ap.jp.nec.com

あらまし TCP プロキシ機構は、通常データ転送のために設定されるエンド端末間の TCP コネクションを、ネットワーク内のノードにおいて分割し、複数の分割 TCP コネクションを用いて中継データ転送を行う技術である。TCP プロキシ機構を用いることで、TCP コネクションの見かけのフィードバックループが小さくなるため、データ転送スループットが向上することが期待される。本稿では、実ネットワークでのデータ転送実験を通じて TCP プロキシ機構の性能評価を行う。その結果、従来の TCP によるデータ転送では十分なスループットが得られない状況においても、エンド端末の TCP の輻輳制御機構のパラメータやアルゴリズムを変更することなく高いスループットが得られることが明らかにする。また、TCP プロキシ間の TCP コネクションにおいて高速データ転送を可能とする輻輳制御方式を用いることで、さらに高いデータ転送性能が得られることを示す。

キーワード TCP (Transmission Control Protocol), 実装, TCP プロキシ, High-Speed TCP

Implementation Experiments of TCP Proxy Mechanism

Kana YAMANEGI†, Takayuki HAMA††, Go HASEGAWA†††,
Masayuki MURATA††††, Hideyuki SHIMONISHI††, and Tutomu MURASE††

† School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka, 560-8531, Japan
†† System Platforms Reseach Laboratories, NEC Corporation,
1753 Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa, 211–8666, Japan
††† Cybermedia Center, Osaka University, 1-32 Machikaneyama, Toyonaka, Osaka, 560-0043, Japan
†††† Graduate school of Information Science and Technology, 1-5 Yamadaoka, Suita, Osaka, 565-0871, Japan
E-mail: yamanegi@ics.es.osaka-u.ac.jp, t-hama@cb.jp.nec.com, hasegawa@cmc.osaka-u.ac.jp,
murata@ist.osaka-u.ac.jp, h-shimonishi@cd.jp.nec.com, t-murase@ap.jp.nec.com

Abstract TCP overlay network, which controls data transmission quality at the transport layer, is paid much attention as the users' demands for diversified services in the Internet becomes large. In the TCP overlay network, a TCP proxy is a fundamental mechanism which splits a TCP connection between sender and receiver hosts into multiple TCP connections at some nodes in the network and relays data packets from the sender host to the receiver host via the split TCP connections. In this paper, we investigate the performance of the TCP proxy mechanism through experiments using the actual public network. We show the results that the TCP proxy mechanism can enhance the data transfer throughput without any change at the endhost's TCP/IP protocol stack. We also evaluate the performance of gentle High-Speed TCP, proposed in the previous work, when it is used on the TCP connection between the TCP proxy nodes.

Key words TCP (Transmission Control Protocol), Implementation, TCP Proxy, High-Speed TCP

1. はじめに

ADSL や FTTH といった広帯域アクセス網技術の進展により、近年ますますインターネットが発展し、ユーザ数の爆発的な増加に伴い、要求されるサービスが多様化している。それらの中には、エンドホスト間のスループットなどに関して高いネットワーク品質を要求するサービスもあるが、現在のインターネットはベストエフォート型であり、ユーザの要求品質を満たすことはできない。この問題を解決し、IP 層において品質制御を行う技術として IntServ [1] や DiffServ [2] などが存在する。例えば DiffServ では、サービスの種類によってルータにおけるパケット処理の優先順位を決定することによって、各フローの通信品質の差別化を行うことを目的としている。しかしながら、IntServ や DiffServ を実現するためには、フローが通過するすべてのルータに品質制御機能が実装されている必要があり、ネットワーク規模に対するスケーラビリティ、導入コストなどの面から実現は困難であると考えられる。一方、CDN (Contents Delivery Network) におけるプロキシキャッシュサーバなどのように、品質制御をアプリケーション層で行う技術も研究されている [3] が、各アプリケーションに特化した複雑な制御を必要とする、所望の性能を得るためのパラメータセッティング等が困難である、などの問題がある。

そこで我々は、IP 層やアプリケーション層において品質制御を行うのではなく、IP 層においては従来のルーティングなど必要最低限の機能のみを提供し、品質制御をトランスポート層において行う TCP オーバレイネットワーク [4] に関する研究を行っている。TCP オーバレイネットワークにおいては、通常エンドホスト間に設定される TCP コネクションをネットワーク内のノード (TCP プロキシ) で終端し、分割されたコネクションごとにパケットを中継しながら転送を行う (図 1)。これにより、TCP コネクションのフィードバックループを小さくすることが可能になるため、スループットの向上を期待することができる。また、TCP オーバレイネットワークを構築することによって、ネットワーク環境の違いを吸収することが可能になるため、要求されるサービス品質に応じた制御を行うことが可能になる。例えば、送受信ホスト間に無線ネットワークが含まれる場合、一般的には TCP コネクションのスループットは大幅に低下する。しかし、無線ネットワーク部分でデータ転送が独立するように、その前後でコネクション分割を行うことにより、性能劣化を最小限に抑えることが可能である。

これまでの研究では、TCP コネクション分割機構による効果を、シミュレーションや数学的解析によって示してきた [5–7]。その結果、TCP コネクション分割機構を用いることにより、エンドホスト間にコネクション設定する場合と比較して平均スループットが向上すること、またファイル転送遅延時間を短縮することができることが明らかになった。そこで本稿では、TCP プロキシ機構の実ネットワークでの性能評価実験を通じて、TCP オーバレイネットワークアーキテクチャが実ネットワークにおいても効果的であることを示す。

一方、高速ネットワークにおける TCP 性能の問題を解決するために、高速 TCP と呼ばれる新たな輻輳制御手法に関する研究が近年盛んに行われている [8–12]。これらの手法は帯域遅延積が大きいネットワークにおいて高いデータ転送スループットを獲得することができる反面、送受信エンドホストの TCP を置き換える必要があるという問題点を持つ。しかし、TCP オーバレイネットワークにおいて、TCP プロキシに高速 TCP の機能を持たせることにより、送受信エンドホストの書き換えを行うことなく、高速・高遅延ネットワークにおいても高いスループットが得られると考えられる。そこで本稿では、TCP プロキシに高速 TCP を実装し、データ転送実験による性能評価を行うことで、高速 TCP の効果および TCP オーバレイネットワークとの組み合わせによる効果を明らかにする。

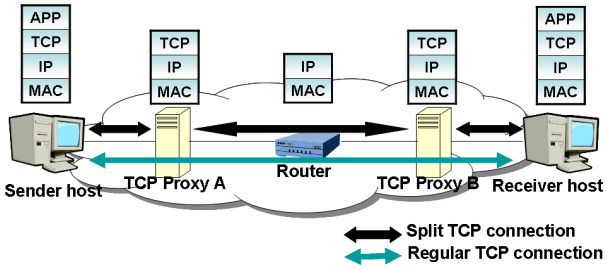

図 1　TCP オーバレイネットワーク

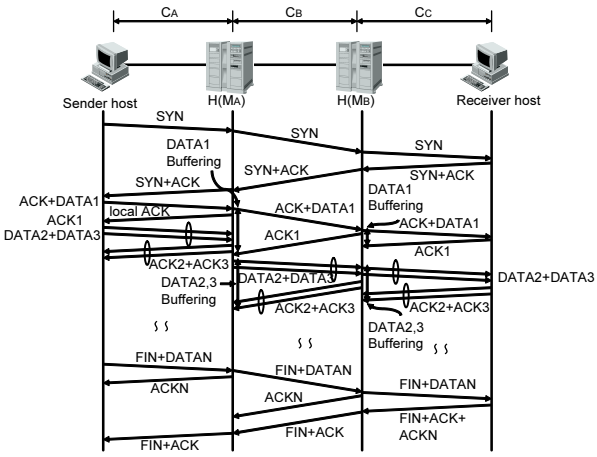

図 2　TCP コネクション分割機構

本稿の構成は以下のとおりである。2 章では関連研究である TCP プロキシおよび高速 TCP の説明を行い、3 章では実験で用いるネットワークおよび実験方法を説明する。さらに 4 章では、基礎実験により得られた実験ネットワークの基本的な性質を述べ、5 章で TCP プロキシおよび高速 TCP の効果を評価する。最後に 6 章でまとめと今後の課題を述べる。

2. 関連研究

2.1 TCP コネクション分割機構

TCP コネクション分割機構は、エンドホスト間に設定される TCP コネクションをネットワーク内のノードで終端することによって TCP コネクションを複数に分割し、分割されたコネクションごとにパケットを中継しながらデータ転送を行うことを実現する機構であり、TCP オーバレイネットワークを構築するための基本的な技術要素である。TCP プロキシは、その TCP コネクションの分割およびパケットの中継転送を行うネットワーク内のノードである (図 1)。また TCP プロキシは、受信側ホストに代わって擬似的な ACK パケットを返信する機能を有している。これにより、送信側ホストは受信側ホストからの ACK パケットを待つことなく新たなデータパケットの送信が可能となるため、送信側 TCP が持つ見かけのラウンドトリップ時間が小さくなり、データ転送効率が向上する。さらに、コネクショ

ン分割をしている TCP プロキシは、データパケットに対する ACK パケットを受信するまでそのデータパケットをバッファリングするため (図 2)、例えば TCP プロキシと受信側ホスト間でパケットが廃棄された場合に、TCP プロキシから廃棄パケットを再送することを可能にしている。

2.2 高 速 TCP

TCP Reno バージョンに基づいて実装されている現在の OS で用いられている TCP は、リンク帯域が 100Mbps を超えるような高速ネットワークにおいては、十分なスループットを得ることができないことが知られている [8]。この問題に対して、TCP の輻輳制御アルゴリズムを改善することで高いスループットを獲得する、高速 TCP と呼ばれる手法が数多く提案されている [8–12]。本稿では、それらのうち文献 [8] において提案されている High Speed TCP (HSTCP)、およびその改良である Gentle High Speed TCP (gHSTCP) [11] を対象とし、TCP プロキシに実装しデータ転送実験を行うことによって、TCP プロキシに高速 TCP を適用することの有効性を検証する。

2.2.1 High Speed TCP (HSTCP)

TCP Reno が高速・高遅延ネットワークにおいて十分なスループットが得られないもっとも大きな原因は、TCP コネクションの転送速度を決定する輻輳ウィンドウサイズの増加速度が遅く (1 ラウンドトリップ時間あたり 1 パケット)、かつパケット廃棄を検知した際の減少幅が大きい (半減) ことである。この問題に対し、[8] において提案されている HSTCP においては、現在の輻輳ウィンドウサイズが大きい場合には、その増加速度を大きくし、減少幅を小さくする。これにより、帯域遅延積の大きなネットワークにおいても輻輳ウィンドウサイズが大きく維持されるため、高いスループットを得ることができる。HSTCP の詳細に関しては [8] を参照されたい。

しかし、[11] において、輻輳ウィンドウサイズの増加速度が原因でパケット廃棄がバースト的に発生する、共存する TCP Reno コネクションのスループットを大幅に低下させるなどの問題点が指摘されている。

2.2.2 Gentle High Speed TCP (gHSTCP)

上述の問題点を解決するための手法として [11] において提案されている gHSTCP においては、輻輳ウィンドウサイズの増加・減少速度を自身の輻輳ウィンドウサイズの値のみに応じて変化させるのではなく、送信データパケットのラウンドトリップ時間の増加傾向を検知することによってネットワークの輻輳状態を監視し、輻輳時には TCP Reno と同じアルゴリズムを用いて輻輳ウィンドウサイズを更新する。これにより、ルータバッファにおけるパケット廃棄率の低下や、共存する TCP Reno との公平性の向上が期待される。gHSTCP のアルゴリズムの詳細に関しては [11] を参照されたい。

3. 実 験 環 境

本章では、本稿における実験に用いた実験ネットワーク、および実験方法の概略について述べる。

3.1 実験ネットワーク

本実験では、NEC (神奈川県川崎市) と大阪大学 (大阪府豊中市) 間に図 3 に示すような通信回線を準備し、通信実験を行う。このとき、エンドホスト間にコネクション設定され、TCP プロキシを用いない場合 (ケース 1) とエンド端末間に設定されるコネクションを TCP proxy A および TCP proxy B で分割する場合 (ケース 2) を考え、データ転送性能の比較を行う。ケース 1

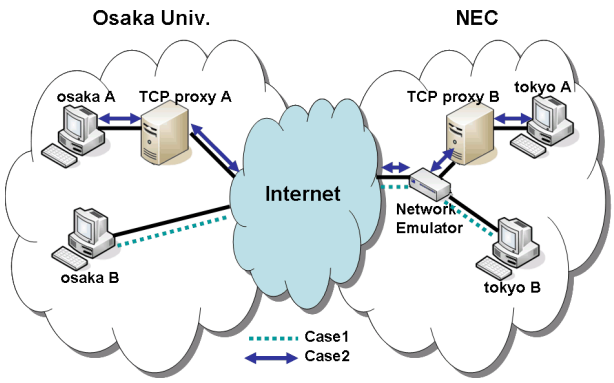

図 3 実 験 環 境

では、阪大側ホスト osaka B と NEC 側ホスト tokyo B を送受信 TCP ホストとして用い、インターネットを介して通信が行われる。ケース 2 では、阪大側ホスト osaka A と NEC 側ホスト tokyo A を送受信 TCP ホストとして用い、TCP proxy A および TCP proxy B を経由して通信が行われる。したがって、ケース 2 においては、エンド間の TCP コネクションが 3 本に分割され (osaka A - TCP proxy A、TCP proxy A - TCP proxy B、および TCP proxy B - tokyo A)、TCP コネクション分割機構を用いて中継転送される。また、NEC 側のネットワーク内にネットワークエミュレータを導入し、NEC と大阪大学間の遅延を大きく設定することを可能にしている。以下では、東京-大阪間相当の通信としてネットワークエミュレータにおいて遅延を発生させない場合、および東京-沖縄間相当の通信としてネットワークエミュレータにおいて 25msec の遅延を発生させる場合の実験結果を示す。

3.2 実 験 方 法

本実験では、帯域測定ツール iperf [13] を用いてエンドホスト間に TCP トラヒックを流し、受信側エンドホストでスループットを測定する。ここでスループットは、受信側エンドホストが単位時間あたりに受信するデータ量と定義する。また TCP の挙動を解析するために、ケース 1 では送信側エンドホスト、ケース 2 では送信側エンドホストに接続されている TCP プロキシの /proc/net/tcp のログを取得し、ラウンドトリップ時間 (RTT) および輻輳ウィンドウサイズ (cwnd) を測定する。

また、以降の実験では、大阪大学側のエンドホスト (osaka A、osaka B) を送信側 TCP ホスト、NEC 側のエンドホスト (tokyo A、tokyo B) を受信側 TCP ホストとして用いた実験結果を示す。これは、予備実験により、大阪大学-NEC 間のネットワークにおいて、大阪大学 → NEC 向きの回線の方がスループットが高く、かつ背景トラヒックも少ないことが明らかとなったためである。

4. 実ネットワーク環境調査

まず、本稿で用いる実験ネットワークの性質を調べるために、通常の TCP トラヒックをネットワークへ流した場合の結果を示す。図 4 は、TCP データ転送を行った場合の、平均スループットの 2 時間における時間的変化を表している。ここでは、iperf によって TCP コネクションを 1 本、2 本、5 本、10 本、および 20 本設定し、2 分間のデータ転送を 12 回行った場合の平均スループットを示している。この図から、コネクション数を 5 本以上にしてもスループットがほとんど向上しないことから、本

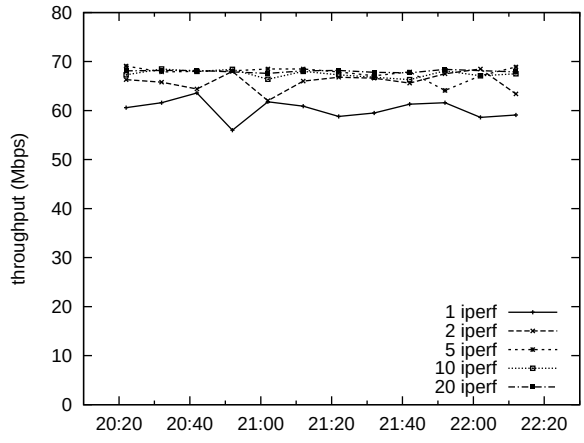

図 4　平均スループット

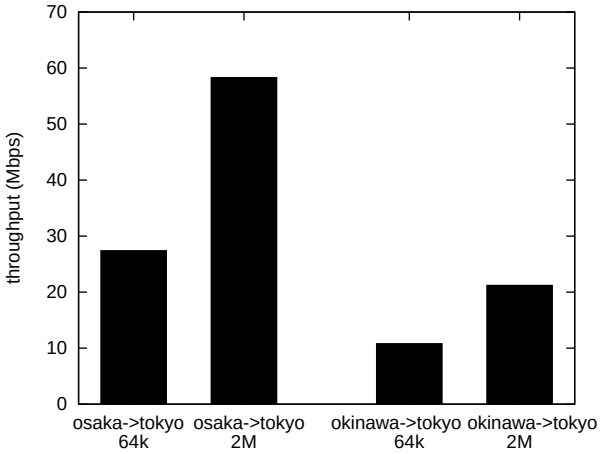

図 6　TCP プロキシを用いない場合のスループット

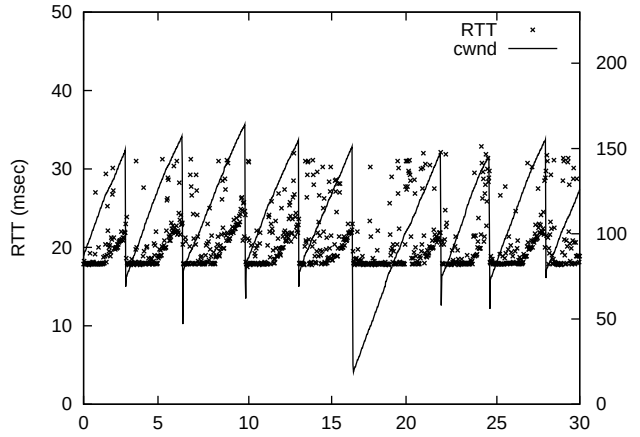

図 5　RTT と cwnd の変動

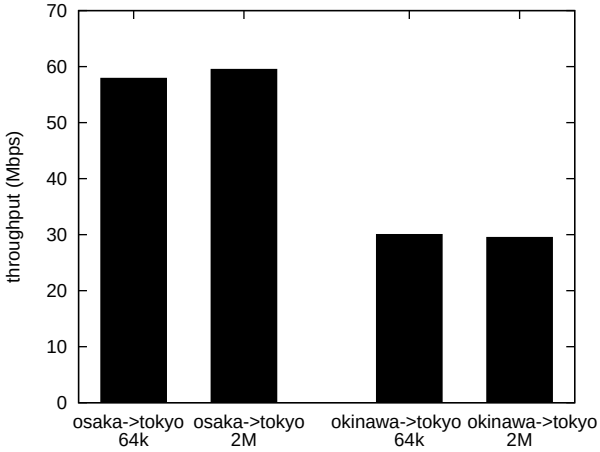

図 7　TCP プロキシを用いる場合のスループット

稿で用いた実験ネットワークにおけるボトルネックリンクの帯域が約 70Mbps であることがわかる。

図 5 は、1 本の TCP コネクションを用いたデータ転送における、TCP コネクションのラウンドトリップ時間 (RTT) と輻輳ウィンドウサイズ (cwnd) の変化をそれぞれ示したものである。この結果から、本実験ネットワークの性質について以下ように推測することができる。

- ウィンドウサイズの増減にともない RTT が増減していることから、ネットワーク輻輳によってボトルネックリンクの出力バッファにパケットが蓄積され、最終的にバッファ溢れが発生している。
- RTT の最小値が約 18msec となっていることから、エンドホスト間の往復伝播遅延時間が 18msec である。
- 最大の RTT が約 30msec となっていることから、ボトルネックリンクの出力バッファが約 12msec (リンク帯域を 70Mbps と考えると約 100KBytes に相当) である。
- パケット廃棄によってウィンドウサイズが小さくなる直前の RTT が安定しており、ほぼ最大 RTT に等しいことから、ボトルネックリンクの出力バッファには Tail Drop 方式が使われている。
- ほとんどの場合においてパケット廃棄時にウィンドウサイズが半減していることから、バッファ溢れによってパケットが廃棄される場合に、1 つのパケットが廃棄されている。

これらの性質は、従来行われている ns-2 [14] などを用いたシミュレーションによって得られる結果とほぼ同じであるといえる。したがって、われわれがこれまでに行ってきたシミュレーションによる TCP プロキシ機構および高速 TCP の評価結果 [5–7, 11] とほぼ同様の結果が、本ネットワークを用いた実験においても得られると考えられる。

5. 実験結果と考察

本章では、データ転送実験結果を示し考察を行うことで、TCP プロキシ機構および高速 TCP の有効性について議論する。

5.1 TCP プロキシの効果

図 6 および図 7 は、それぞれ TCP プロキシ機構を用いない場合と用いる場合について、送受信エンドホストの TCP ソケットバッファサイズおよび送受信エンドホスト間の遅延時間を変

化させた時の、データ転送の平均スループットを示している。実験は、2 分間のデータ転送を 5 回行い、その平均値を示している。TCP プロキシを用いない場合はケース 1 に相当し、送受信エンドホスト間に TCP コネクションを 1 本設定しデータ転送を行っている。一方 TCP プロキシを用いる場合はケース 2 に相当し、図 3 に示す 2 箇所の TCP プロキシにおいてコネクション分割を行い、3 本の中継コネクションを用いてデータ転送を行っている。なお、TCP プロキシ間の TCP コネクションのソケットバッファサイズは十分大きく設定している。

図 6 から、東京-大阪間の通常のデータ転送においては、TCP コネクションが用いるソケットバッファが小さい場合には、十分なスループットが得られず、リンク帯域を有効に使うためには、ソケットバッファを大きく設定する必要があることがわかる。これは、本実験で用いたネットワークの帯域遅延積は約 150KByte であるため、64KByte のソケットバッファではリンク帯域を使い切ることができないためである。また、東京-沖縄間の実験結果から、エンド端末間の遅延時間が大きくなると、バッファを大きくしてもリンク帯域を使い切ることができないことがわかった。これは、2 章において説明したように、TCP Reno のウィンドウサイズの制御アルゴリズムが、ネットワークが輻輳していない場合には 1RTT に 1 パケット分だけ大きくし、パケット廃棄を検知したときに半減させるため、大きな帯域遅延積を有効に使うことができないためである。

一方、TCP プロキシを導入してコネクション中継を行った場合（図 7）には、東京-大阪間の実験結果から、エンド端末のバッファが小さい場合においても、非常に高いスループットが得られていることがわかる。これは、送信側ネットワーク側の TCP プロキシにおいて TCP コネクションを中継し、ソケットバッファサイズを十分大きくしているため、TCP プロキシ間の TCP コネクションが高いスループットでデータ転送を行うことができること、および送信側エンドホストと TCP プロキシ間の TCP コネクションは帯域遅延積が非常に小さいため、小さなソケットバッファを用いても十分高いスループットが得られることが理由である。すなわち、TCP プロキシを導入することによって、接続されたエンドホストはプロトコルやパラメータを変更することなく、高いスループットを得ることができる。

一方、東京-沖縄間の実験結果から、エンドホスト間の遅延時間が大きい場合においても、エンドホストのソケットバッファサイズを大きく設定しなくても、大きく設定した場合とほぼ同等のスループットが得られていることがわかる。しかし、東京-大阪間のスループットと比較すると、東京-沖縄間のスループットは TCP プロキシを用いた場合においても低い。これは、TCP プロキシが両エンドホストから非常に近い場所に設置されているため、TCP プロキシ間の伝播遅延時間が、エンドホスト間の伝播遅延時間とほぼ等しいことから、分割された TCP コネクションのフィードバックループが小さくならず、TCP プロキシ間の TCP コネクションが十分なスループットを得られないためである。これは、図 6 に示す実験結果のうち、東京-沖縄間のネットワークにおいてはソケットバッファを大きくしてもリンク帯域を使い切ることができないことからも明白である。すなわち、TCP プロキシを設置する場所によっては、TCP プロキシを設置するだけでは十分なスループットが得られないといえる。このことは、[6] においても指摘されており、TCP プロキシの効果は、エンド間の遅延時間を等しく分割するような場所にプロキシを置くことでもっとも大きくなることが示されている。

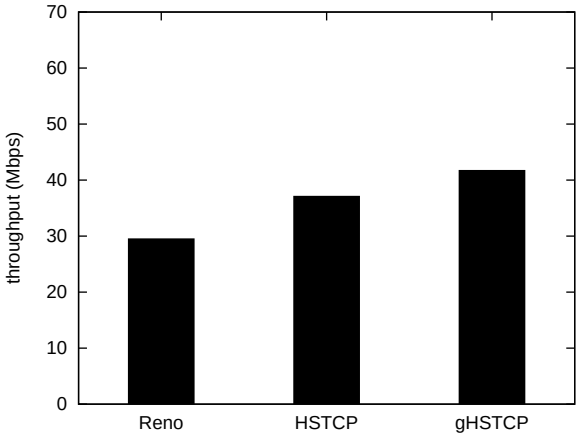

図 8　TCP プロキシ間に高速 TCP を用いた場合のスループット

5.2　高速 TCP の効果

上述の問題を解決するための有効な手段の 1 つとして、TCP プロキシ間の TCP コネクションに高速 TCP を用いることが挙げられる。図 8 に、TCP プロキシ間の TCP コネクションにおいて High Speed TCP (HSTCP) を用いた場合のデータ転送結果を示す。この図から、HSTCP を使うことでエンド間のデータ転送のスループットが向上していることがわかる。また、[11] において提案した HSTCP の改良方式である gHSTCP は、HSTCP よりもさらに高いスループットを得ることができることがわかる。これは、[11] におけるシミュレーション結果に示されているように、gHSTCP を用いることによってバッファ溢れが発生するときのパケット廃棄数が大きく削減されるため、廃棄されたパケットの再送にかかる時間が HSTCP に比べて小さくなることが原因であると考えられる。

しかし、gHSTCP を用いた場合においても、得られると思われる最大スループット (約 70Mbps) は得られていない。これは、送受信エンドホスト間の伝播遅延時間が大きいため、パケット廃棄時にタイムアウトが発生する確率が高いためであると考えられる。この問題に対する解決策に関しては今後の課題としたい。

次に、TCP プロキシを使わないコネクションと、高速 TCP を用いる TCP プロキシを使うコネクションが同時にネットワーク内に存在した場合の、両コネクション間の公平性およびスループットに関する評価を行う。図 9 は、東京-沖縄間のネットワークにおける 2 本のコネクションの TCP プロトコル、およびネットワークの遅延時間を変化させた場合の、それぞれのコネクションの平均スループットを表している。図から、TCP プロキシ間に HSTCP を用いることで、そのコネクションのデータ転送スループットは向上するが、その反面、TCP プロキシを用いない通常の TCP コネクションのスループットを大きく低下させていることがわかる。これは、HSTCP はネットワークの輻輳状況に関係なく、自身のウィンドウサイズに応じてウィンドウサイズの増減量を調整するため、共存する TCP Reno コネクションのスループットを低下させることが原因である。

一方、高速 TCP として gHSTCP を用いた場合には、高いスループットを実現しながら、共存する TCP Reno コネクションのスループットをそれほど低下させていないことがわかる。こ

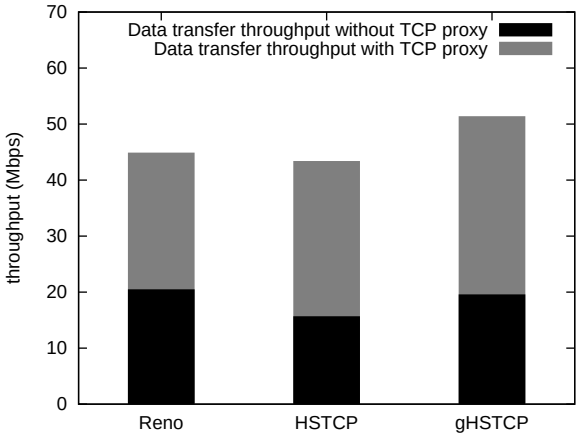

図 9　クロストラヒックとの公平性

れは 2.2.2 節で述べたように、gHSTCP はネットワークの輻輳状態に応じて自身のウィンドウサイズの増減量を調整するためである。これらの結果から、[11] において提案された gHSTCP により、共存する TCP Reno の性能を低下させることなく、自身のデータ転送スループットを大きく向上することが明らかとなった。

6. おわりに

本稿では、TCP プロキシ機構および HSTCP/gHSTCP 方式を実マシン上に実装し、その有効性を NEC (神奈川県川崎市) と大阪大学 (大阪府豊中市) 間の公衆網を経由する実験ネットワークにおけるデータ転送実験を通じて検証した。その結果、TCP プロキシを用いることで、エンドホストの TCP の輻輳制御アルゴリズムやソケットバッファサイズを変更することなく、高いスループットを得ることができることを示した。また、TCP プロキシ間のデータ転送プロトコルとして HSTCP を用いることで、ネットワーク遅延が非常に大きい場合など、従来の TCP Reno では十分なスループットを得ることができない状況においても高いスループットを得ることができる反面、パケット廃棄数や既存の TCP Reno コネクションとの公平性に関して問題が存在すること、および [11] で提案された gHSTCP 方式がそれらの問題点を解決し、HSTCP に比べて高いスループットと公平性を同時に満たすことができることを明らかにした。

今後の課題としては、3 地点以上のネットワークを用いたさらに大規模な公衆網を用いた実験ネットワークにおける検証、gHSTCP 方式のアルゴリズムのパラメータ調整などが挙げられる。また、100Mbps 以上の高速ネットワークを用いた実験を行い、TCP プロキシ機構の高速ネットワークにおける性能評価をあわせて行いたい。

謝　辞

本研究の一部は、総務省における研究プロジェクトである戦略的情報通信研究開発推進制度委託研究「ユビキタスインターネットにおける高位レイヤスイッチの研究開発」によっている。ここに記して謝意を表す。

文　献

[1] J. Wroclawski, “The use of RSVP with IETF integrated services,” *RFC 2210*, Sept. 1997.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An architecture for differentiated service,” *RFC 2475*, Dec. 1998.

[3] S.Williams, M. Abrams, C. R. Standridge, G. Abdulla, and E. A. Fox, “Removal policies in network caches for World-Wide Web documents,” in *Proceedings of ACM SIGCOMM’96*, Aug. 1996.

[4] 村瀬 勉, 下西 英之, 長谷川 洋平, “TCP オーバレイネットワークの提案,” **電子情報通信学会 通信ソサイエティ大会報告** *(B-7-49)*, Sept. 2002.

[5] 牧 一之進, 長谷川 剛, 村田 正幸, 村瀬 勉, “TCP オーバレイネットワークにおける TCP コネクション分割機構の性能解析,” **電子情報通信学会 技術研究報告** *(IN03-198)*, Feb. 2004.

[6] 牧 一之進, 長谷川 剛, 村田 正幸, 村瀬 勉, “TCP オーバレイネットワークの性能解析および評価,” **電子情報通信学会技術研究報告** *(IN2004-96)*, Oct. 2004.

[7] I. Maki, G. Hasegawa, M. Murata, and T. Murase, “Throughput analysis of TCP proxy mechanism,” in *Proceedins of ATNAC 2004*, Dec. 2004.

[8] S. Floyd, “Highspeed TCP for large congestion windows,” *RFC 3649*, Dec. 2003.

[9] C. Jin, D. X. Wei, and S. H. Low, “FAST TCP: motivation, architecture, algorithms, performance,” in *Proceedings of IEEE INFOCOM 2004*, Mar. 2004.

[10] T. Kelly, “Scalable TCP: Improving performance in high-speed wide area networks,” in *Proceedings of PFLDnet ’03: workshop for the purposes of discussion*, Feb. 2003.

[11] Z. Zhang, G. Hasegawa, and M. Murata, “Analysis and improvement of HighSpeed TCP with TailDrop/RED routers,” in *Proceedings of MASCOTS 2004*, Oct. 2004.

[12] T. Iguchi, G. Hasegawa, and M. Murata, “A new congestion control mechanism of TCP with inline network measurement,” *to be presented at ICOIN2005*, Jan. 2005.

[13] NARANR, “NARANR/DAST: Iperf 1.7.0 - The TCP/UDP Bandwidth Measurement Tool,” available from `http://dast.nlanr.net/Projects/Iperf/`.

[14] T. V. Project, “UCB/LBNL/VINT network simulator - ns (version 2),” available from `http://www.isi.edu/nsnam/ns/`.