

シンクライアントトラヒックの性能向上のための 遅延解析とTCP層最適化

小川祐紀雄[†] 長谷川 剛^{††} 村田 正幸^{†††}

[†] 日立製作所 システム開発研究所 〒215-0013 神奈川県川崎市麻生区王禅寺 1099

^{††} 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: [†]yukio.ogawa.xq@hitachi.com, ^{††}hasegawa@cmc.osaka-u.ac.jp, ^{†††}murata@ist.osaka-u.ac.jp

あらまし シンクライアントシステムとは、本稿では、クライアントからキーボード・マウスイベントを送信し、サーバから画面情報を受信するシステムを指す。本稿においては、シンクライアントシステムで発生するトラヒックを、文字情報に相当するインタラクティブデータフローと、ウィンドウなどの画面情報に相当するバルクデータフローからなる二状態系で把握し、利用者の操作性能を向上させるために、通信全体の性能を維持しつつ前者の遅延を最小化することを目的とする。実システムから抽出したトラヒックを用いてシミュレーションを行い、インタラクティブデータフローの遅延の要因は、ボトルネックリンクに接続するルータでのキューイング遅延と、サーバのTCP送信バッファにおける送信遅延であることを示す。さらに、インタラクティブデータフローの優先制御を行い、かつ、バルクデータフローの再送タイムアウトの発生を抑制する手法を提案し、各データフローがTCPコネクションを共有する場合と個別に持つ場合それぞれにおいて、遅延が改善されることを示す。

キーワード シンクライアント, インタラクティブ, バルク, 二状態系, 優先制御, 再送タイムアウト

Delay Analysis and Transport-Layer Optimization for Improving Performance of Thin-Client Traffic

Yukio OGAWA[†], Go HASEGAWA^{††}, and Masayuki MURATA^{†††}

[†] Systems Development Laboratory, Hitachi, Ltd.

Ohzenji 1099, Asao-ku, Kawasaki-shi, Kanagawa 215-0013 Japan

^{††} Cyber Media Center, Osaka University Machikaneyama 1-32, Toyonaka-shi, Osaka 560-0043 Japan

^{†††} Graduate School of Information Science and Technology, Osaka University

Yamadaoka 1-5, Suita-shi, Osaka 565-0871 Japan

E-mail: [†]yukio.ogawa.xq@hitachi.com, ^{††}hasegawa@cmc.osaka-u.ac.jp, ^{†††}murata@ist.osaka-u.ac.jp

Abstract Thin-client system is a system in which a client transmits a user event such as a mouse or keyboard input to a server and the server returns screen updates of a desktop application to the client. Network traffic in a thin-client system, that is, thin client traffic, is recognized by using two state system, that is consist of interactive data flows and bulk data flows. The interactive data flows mainly transfer character information, and the bulk data flows transfer screen update information. Our purpose is to minimize the latency time of the interactive data flows as well as to keep the performance of whole data flows in order to improve user's operation performance. We run the traffic simulation using the real field data, and show that the latency time of an interactive data flow is almost the sum of queuing delay at the router connecting to the bottleneck link and TCP's transmit delay at the server. We propose techniques of priority queuing of the interactive data flows as well as restraining the retransmission time out of the bulk data flows, and we show the effectiveness of the techniques when the two data flows exist together in a TCP connection as well as when each data flow has its own TCP connection.

Key words thin client, interactive data, bulk data, two state system, priority queuing, retransmission time out

1. はじめに

シンクライアントシステムとは、クライアント・サーバ型のシステムであり、本稿では、サーバのリモートデスクトップサービスを利用して、データやアプリケーションを持たないクライアントからキーボードやマウスのイベント情報をサーバに送信し、サーバから処理結果を表す画面情報を受信する画面転送型のシステムを指す(図1)。データセンタにPCを集約し、オフィスや自宅などからネットワーク経由でPCに接続しデスクトップ環境を利用する形態で企業などに導入されている。データや計算資源と利用者を分離することにより、コンピュータ資源の集中管理、PCの盗難や紛失による情報漏洩リスクの低減、柔軟なサテライトオフィス勤務や在宅勤務などの就業形態を実現する企業情報システムとして注目されている[1]。

シンクライアントシステムを実現するアプリケーションプロトコルには、X11[2]、Citrix[®] Independent Computing Architecture(ICA)[3]、Microsoft[®] Remote Desktop Protocol(RDP)[4]などがある。シンクライアント通信は、持続的接続(persistent connection)を用いる対話型のTCP(Transmission Control Protocol)通信であり、クライアントからの要求に対して、文字情報やウィンドウ画面情報などの様々な大きさのデータを即時に返す。従来、シンクライアントシステムで発生するトラフィック(シンクライアントトラフィック)に関して、代表的なアプリケーションソフトの操作時における性能評価結果が報告されており、広域ネットワーク環境での利用時などに、ネットワーク遅延が問題になることが指摘されている[5],[6]。

また、即時に処理結果を返す必要のあるアプリケーションプロトコルにおいて、TCPのNagleアルゴリズムと遅延ACK(delayed acknowledgement)に起因するデータのバッファリングが性能に影響を与えることが報告されている[7],[8]。例えば、X11では、利用者の操作性能を向上させるために小さなデータを遅延なく転送することが必要であり、これらの設定をオフにする必要がある[9]。

また、持続的接続を用いるTCPにHTTP(Hypertext Transfer Protocol)/1.1がある[10],[11]。このHTTPにおいて、TCPのスロースタート再スタート(slow-start restart)により、アイドル時間後に輻輳ウィンドウサイズが初期化されるため、転送性能が向上しないことが示されている[12],[13]。

また、転送データサイズの異なる複数のTCPフローが競合し不公平が発生することも報告されている。インターネット通信の大半はウェブサイト閲覧などのための小さなサイズのデータフローであるが、それらは、少数の大きなサイズのバルクデータフローのためにキューイング遅延やパケットロスを起こすことがあり、DiffServ(Differentiated Services)やRED(Random Early Detection)などの適用によりこれらの問題を解決する手法が示されている[14]~[16]。

以上の関連研究を踏まえ、シンクライアントトラフィックの性能向上手法を検討する。シンクライアントトラフィックは、キーボード入力に対する文字情報出力などに相当するよりインタラクティブな特性のデータフロー(インタラクティブデータフ

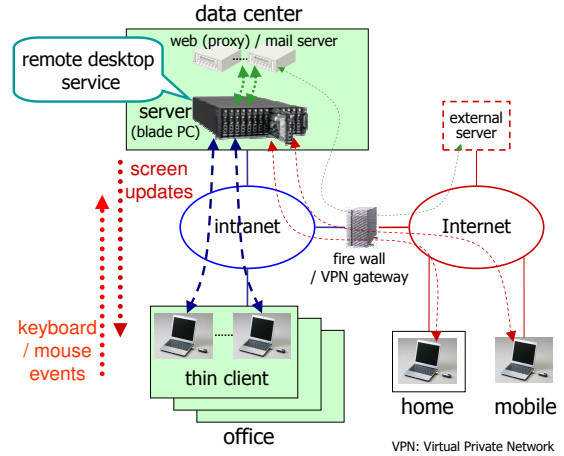


図1 シンクライアントシステムの構成

ロー)と、マウス入力に対するウィンドウ画面情報出力などに相当するよりバルク転送的な特性のデータフロー(バルクデータフロー)が一コネクション中に混在したトラフィックと捉えられる。我々のこれまでの研究において、それぞれのデータフローに対してパケットロスに対する耐性向上を課題に検討を行い、前者のフローに対してはパケットの複製を送信することにより、後者のフローに対してデータセグメントの再構成を行うことにより、課題を解決できることを示した[17],[18]。

本報告では、さらに、これらのデータフローの競合状態を対象に検討を行う。利用者はキーボード入力に対する文字情報出力の遅延に対してより敏感であるため、利用者の操作性能の向上のために、トラフィック全体としての性能を維持しつつインタラクティブデータフローの遅延を最小化することを目的とする。実トラフィックから抽出した評価用データを用いてシミュレーションを実施し、インタラクティブデータフローの遅延は、ボトルネックリンクに接続するルータでのキューイング遅延と、サーバのTCP送信バッファにおける送信遅延に起因することを示す。さらに、それらの改善のために、従来手法のルータでの優先制御に加え、TCPの再送タイムアウト発生を抑制する手法を提案する。また、TCP送信バッファでの遅延の根本的な原因は、両データフローが一コネクション中に共存することであるため、各データフローがTCPコネクションを共有する場合と個別に持つ場合に分けて評価を行う。

以下、2章ではシンクライアントトラフィックのモデル化手法について説明する。3章でインタラクティブデータフローの性能劣化の要因をシミュレーションによって明らかにし、4章において性能向上手法を提案しその評価結果を示す。最後に5章で本稿のまとめと今後の課題を述べる。

2. シンクライアントトラフィックのモデル化

2.1 二状態系によるモデル化

シンクライアントトラフィックを、図2に示すように、文字情報などに相当するインタラクティブデータフローと、ウィンドウ画面情報などに相当するバルクデータフローからなる二状態

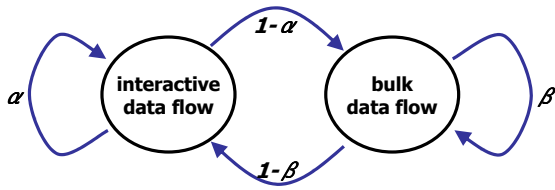


図 2 シンククライアントトラフィックのモデル化

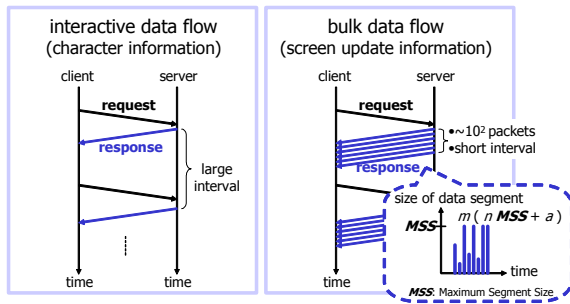


図 3 インタラクティブ/バルクデータフローの特徴

表 1 評価用トラフィックの特徴

(一評価用トラフィック (300 秒) あたりの平均値)

	interactive data flow	bulk data flow
flows	934	198
packets	934	1,466 (7.4/flow)
bytes	128,188 (137.2/packet)	1,373,366 (936.8/packet)

系によりモデル化する。図 3 に示すように、インタラクティブデータフローは、一つの要求パケットと一つの応答パケットの組からなる。一方、バルクデータフローは、一つの要求パケットと最大 10^2 個程度の応答パケットとの組からなる。これらのデータフローの識別は、応答パケットの到着間隔を基に行う。アプリケーションプロトコルに Microsoft[®] RDP を使用した約 200 台のサーバとクライアントの組からなる試行システムに対する約一ヶ月間のトラフィック観察結果に基づき、応答パケットの到着間隔に関する閾値を $10^{-2.2}$ sec (約 6.3 msec) と設定する [18]。二状態系における状態遷移確率は、上述の観察結果より、イントラネットからの接続では $\alpha = 0.91$, $\beta = 0.58$, インターネットからの接続では $\alpha = 0.94$, $\beta = 0.59$ であった。これらの値の違いは、各状況で利用するアプリケーションソフトの違いなどによると考えられる。

2.2 評価用トラフィックの抽出

次章以降で実施するシミュレーションのための評価用トラフィックとして、個々の評価用トラフィックがシンククライアントトラフィックの平均的な特徴を備えるよう、各トラフィック中のインタラクティブ/バルクデータフローの状態遷移確率が前節で求めた値に近いトラフィックを選ぶ。また、実トラフィックから抽出する際には、遅延と帯域の影響の少ない、イントラネット経由で接続したトラフィックを用いる。このとき、大半は同一拠点内からの接続であり、遅延は 1 msec 以下、帯域はエンド・ツー・

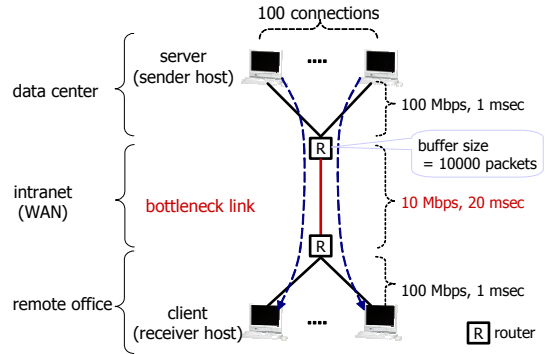


図 4 シミュレーションモデル

エンドでほぼ 100 Mbps である。一評価用トラフィックの条件を、時間区間が 300 sec, 前節で求めた状態遷移確率との差が 0.01 以内、含まれるフロー数が 100 以上とし、これらの条件を満たす約 100 個のトラフィックを抽出した (表 1)。

3. インタラクティブデータフローの遅延解析

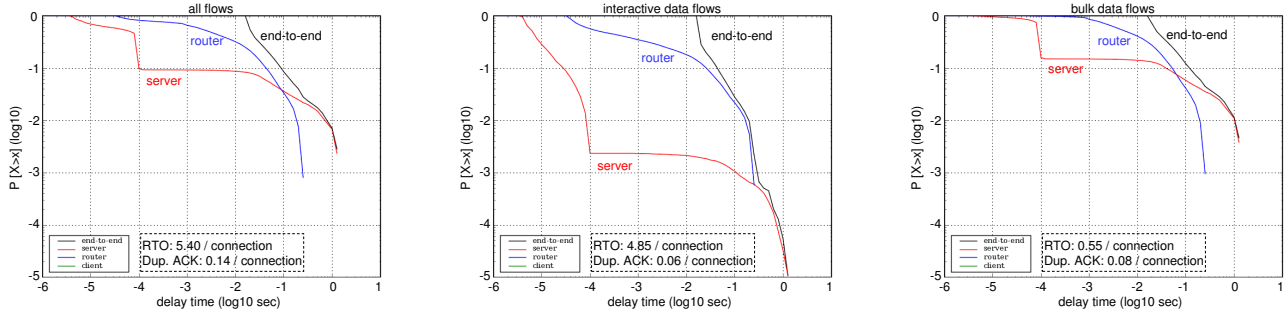
3.1 対象システム構成とシミュレーションモデル

サーバに対して同時に多数の通信が行われる場合のシステム構成として、データセンタからイントラネットを経て多数のシンククライアントが設置された遠隔オフィスに接続する構成を対象とする。イントラネットには、東京～大阪間といった数 10 msec の遅延のある広域ネットワークを考える。図 1 に示すシンククライアントシステムの構成の特徴により、サーバからメール・ウェブサーバなどの業務サーバへの通信は、基本的にデータセンタ内で閉じており、広域ネットワークを転送されるトラフィックはほぼシンククライアントトラフィックのみである。以上の構成を図 4 に示すモデルで表す。サーバ (送信ホスト) とクライアント (受信ホスト) を 100 組とし、これらの 100 コネクションが同時に通信する。各ホストから広域ネットワーク (ボトルネックリンク) に接続するルータまでの回線の帯域を 100 Mbps, 伝播遅延時間を 1 msec, 広域ネットワークでは、帯域を 10 Mbps, 伝播遅延時間を 20 msec に設定する。ルータのバッファサイズは、パケットドロップが発生しない十分大きな値とする。サーバおよびクライアントの設定は、即時に応答を返信するために Nagle アルゴリズムと遅延 ACK の設定はオフに設定し、スループット向上のためにスロースタート再スタートもオフに設定する [18]。

3.2 インタラクティブデータフローの遅延発生箇所の解析

サーバのアプリケーションが送信してからクライアントのアプリケーションが受信するまでのエンド・ツー・エンドの遅延において、回線の伝送遅延以外の遅延要因を以下に挙げる。なお、各ノードでのパケット処理遅延は除いている。

- サーバでの遅延 - TCP 送信バッファにおけるバッファリング遅延, 送信遅延
- ルータでの遅延 - キューイング遅延, 送信遅延
- クライアントでの遅延 - TCP 受信バッファにおけるバッファリング遅延



(a) トラフィック全体 (b) インタラクティブデータフローのみ (c) バルクデータフローのみ

図 5 遅延分布

これらのうち、クライアントでのバッファリング遅延は、行頭ブロッキング (head-of-line blocking) と呼ばれる。パケットドロップなどによってパケットの順序が入れ替わって受信されない限りは発生しないため、本報告では改善対象としない。

図 5 に、ns-2 [19] によりシミュレーションを行った結果を示す。トラフィック全体、インタラクティブデータフローのみ、バルクデータフローのみに分けて、エンド・ツー・エンド、サーバ、ボトルネックリンク接続ルータ、クライアントでの各遅延について、パケット毎の遅延時間の累積補分布 (CCDF: Complementary Cumulative Distribution Function) を示している。グラフにコネクション当たりの再送タイムアウト (RTO: Retransmission Time Out) と重複 ACK (duplicate ACK) の発生回数 (輻輳ウィンドウサイズ (congestion window size) の縮小回数) を追記している。図 5(b) に示されるように、インタラクティブデータフローのエンド・ツー・エンドでの遅延の要因は、ボトルネックリンク接続ルータおよびサーバでの遅延である。インタラクティブデータフローで発生するエンド・ツー・エンドの遅延に関して約 0.3 sec より小さな遅延が 99.9% を占め、その部分はルータでの遅延が支配的である。一方、それ以上の大きな遅延も 0.1% 程度と割合は小さいが存在しており、その部分はサーバでの遅延が支配的である。

ボトルネックリンク接続ルータでの遅延は、ルータのバッファにおいてパケットの出力速度より入力速度の方が大きく待ち行列が発生することによる。サーバでの遅延は、TCP 送信バッファからの出力速度よりアプリケーションからの入力速度の方が大きい場合に発生する。バルクデータフローの TCP 送信バッファへの入力速度は特に大きいためその送信遅延が後続のインタラクティブデータフローに及ぶこともある。また、図 6 に例示するように、ルータのキュー長の時間変化に対して、サーバの送信間隔が大きく往復遅延時間 (RTT: Round Trip Time) の測定回数が少ないため、ルータでの遅延時間に対してサーバでの再送タイムアウト値の再計算が追随せず、ルータでパケット廃棄が発生していないにも関わらずサーバで再送タイムアウトが発生している。このような再送タイムアウトの発生により、サーバでの輻輳ウィンドウサイズが初期化され送信遅延がさらに増加する。なお、重複 ACK 受信による輻輳ウィンドウサイ

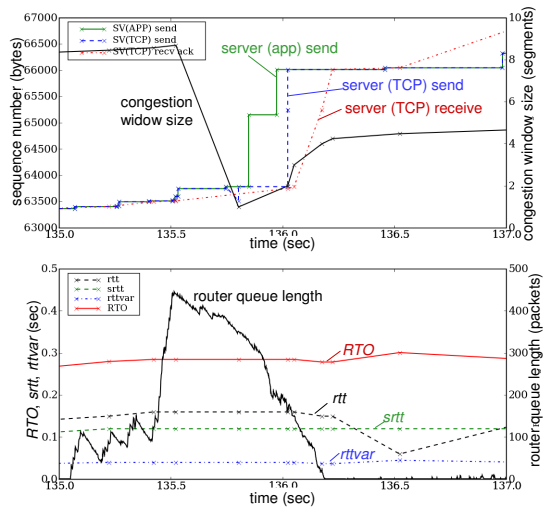


図 6 再送タイムアウト発生によるサーバでの遅延の例

ズの縮小も発生しているが、これは、複数のパケットが再送タイムアウト発生により連続して再送信されると、クライアントではオリジナルのパケットの受信後、複数の再送パケットを受信し、そのつど最大受信シーケンス番号に基づく ACK を返信するという動作による。

4. インタラクティブデータフローの性能向上

4.1 性能向上手法の検討

前章の評価結果より、インタラクティブデータフローの遅延の最大要因は、ボトルネックリンクに接続するルータでのキューイング遅延であることがわかったので、本稿では、ルータでの優先制御によりインタラクティブデータフローの遅延を減少させる。ルータに各データフロー用のバッファを設け、インタラクティブデータフローをバルクデータフローに対して優先して送信する。各データフローの識別のためのラベル付けは、サーバのアプリケーション層で実施する。

上記の優先送信により、インタラクティブデータフローの再送タイムアウトは発生せず、同フロー自身に起因するサーバでの遅延は起こりにくいと考えられる。したがって、サーバでの

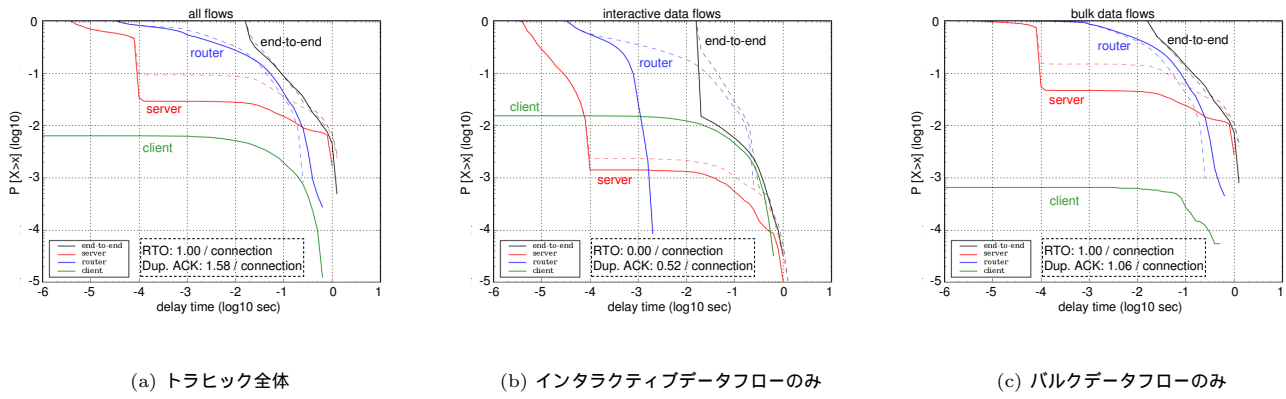


図 7 インタラクティブ/バルクデータフローが同一コネクションに共存するときの評価結果

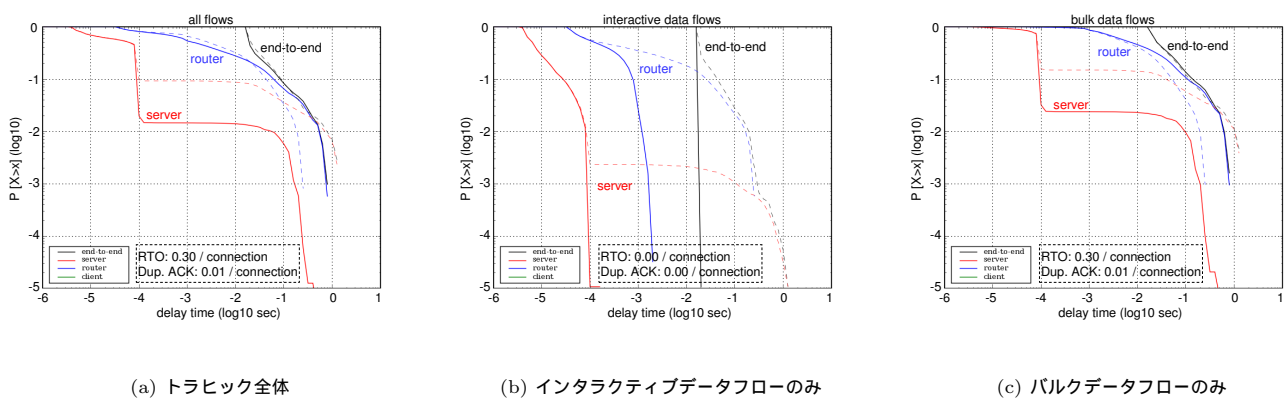


図 8 インタラクティブ/バルクデータフローに個別にコネクションを割り当てたときの評価結果

遅延を減少させるためには、バルクデータフローの送信遅延が後続のインタラクティブデータフローに影響しないようにする必要がある。これは、一つの TCP コネクション中に二つのデータフローが共存することに根本的な原因があるため、以下の二つのアプローチに分けて検討を行う。

- (1) インタラクティブ/バルクデータフローが同一 TCP コネクションに共存 (現アプリケーションプロトコル仕様を前提)
 - (2) インタラクティブ/バルクデータフローの TCP コネクションを分離 (アプリケーションプロトコルの改変を前提)
- アプローチ (1) においては、バルクデータフローの送信遅延を抑えるために輻輳ウィンドウサイズの大きさを保つことが必要である。そのため、再送タイムアウト発生の抑制手法を検討する。再送タイムアウト値 RTO は次式で計算される [6]。

$$srtt \leftarrow (1 - g)srtt + g \cdot rtt \quad (1)$$

$$rttvar \leftarrow (1 - h)rttvar + h|rtt - srtt| \quad (2)$$

$$RTO \leftarrow srtt + 4rttvar \quad (3)$$

rtt は往復遅延時間, $srtt$ は加重平均した往復時間, $rttvar$ は加重平均した平均偏差, g, h は加重平均のパラメータで $g = 1/8, h = 1/4$ である。提案方式においては、アイドル時間後のルータのキュー長の変化に対応するために、アイドル時間後に再送タイムアウト時間をリセットする。つまり、スロー

スタート再スタート発生のタイミングで $srtt$ および $rttvar$ を初期化する。ただし、このとき輻輳ウィンドウサイズは初期化しない。さらに、ルータのキュー長の変化に追従するよう、より最近の往復遅延時間を再送タイムアウト値に反映させる。そのため $g = 7/8$ に設定する。

アプローチ (2) のコネクションを分離し個別に割り当てる場合においては、サーバでの送信時にバルクデータフローがインタラクティブデータフローに影響することは無いが、バルクデータフロー自身の性能を向上させるために、同様に再送タイムアウト発生の抑制を行う。

4.2 バルクデータフローと共存する場合の評価

アプローチ (1) の二つのデータフローが同一コネクションに共存する場合の制御結果を図 7 に示す。比較のために、図 5 の制御なしでの分布を点線で示している。

図 7(b) に示すように、インタラクティブデータフローにおいて、エンド・ツー・エンドの遅延の大半を占める約 0.3 sec 以下の遅延は、ルータでの優先制御に伴い改善される。ただし、代わりにクライアントでの遅延が発生している。これは、ルータでの優先制御により先に送信されたバルクデータフローのバケットを追い越すことがあり、行頭ブロッキングが発生するからである。サーバでの遅延も若干改善されているが、これは、バルクデータフローの再送タイムアウト発生を抑制しているか

らであり（優先制御のみの適用時は，フロー全体での再送タイムアウト発生回数は 2.87 回 / コネクション），再送タイムアウト発生抑制手法を適用しないとサーバでの遅延は悪化する．

バルクデータフローの遅延は，図 7(c) に示すように，制御の適用後も維持されている．ルータでの遅延は，非優先であってもインタラクティブデータフローのデータサイズが小さいため大きな遅延の増加とはならない．サーバでの遅延も，再送タイムアウト発生の抑制により改善されている．なお，行頭ブロッキング（クライアントでの遅延）が発生しているのは，TCP 送信バッファからの送信時にバルクデータフローのセグメントが先行するインタラクティブデータフローのセグメントと再構成されて一つのセグメントとなり，ルータで優先送信され追い越しが発生したためである．

4.3 バルクデータフローと分離した場合の評価

アプローチ (2) の二つのデータフローに対してそれぞれ個別にコネクションを割り当てた場合の制御結果を図 8 に示す．このとき，クライアント側アプリケーションは，もともとのコネクション時の各データフロー間のパケットの順序を維持しなくてもよいとする．

インタラクティブデータフローの遅延は，図 8(b) に示すように，ほぼ回線の伝送遅延のみである．バルクデータフローとコネクションを分離したことにより，ルータでのパケットの追い越しは発生せず，したがって，クライアントでの遅延は発生しない．サーバにおいてバルクデータフローの送信遅延の影響を受けることもない．

バルクデータフローの遅延は，図 8(c) に示すように，制御前の値を維持している．ルータでの遅延が増加するのは，インタラクティブデータフローに対して非優先で転送されることに加えて，再送タイムアウト回数が減少し輻輳ウィンドウサイズが縮小せず，バースト性が増加することによる．サーバでの遅延は，再送タイムアウト発生の抑制により改善されている．

5. おわりに

本稿では，シンクライアントトラフィック全体の通信性能を維持しつつインタラクティブデータフローの遅延を最小化する TCP 層の制御方式について検討した．実トラフィックを用いてシミュレーションを行い，インタラクティブデータフローの遅延の大半はボトルネックリンク（広域ネットワーク）に接続するルータでのキューイング遅延であるが，その他にサーバでの TCP 送信バッファにおける遅延も存在することを示した．このサーバでの遅延には再送タイムアウト発生が影響している．実際に広域ネットワークを利用する場合は，アプリケーションプロトコルの設定を変更しデータサイズを減らすことが考えられるので，バースト性が緩和され再送タイムアウト発生回数は少なくなると考えられるものの，本稿のシステム構成に近い条件下では再送タイムアウトが発生する可能性が高いと予想される．

現在のプロトコルの仕様のようインタラクティブデータフローがバルクデータフローとコネクションを共有する場合，サーバでの遅延をより小さくするためには輻輳ウィンドウサイズを大きく保つことが必要である．本稿では，再送タイムアウ

ト値のアイドル時間後の初期化や計算パラメータ変更により再送タイムアウト発生を抑制することで実現できることを示した．一方，インタラクティブデータフローの遅延からバルクデータフローの影響を完全に除くためには，アプリケーションプロトコルを改変しコネクションを分離することが必要であることも示した．今後の課題としては，ルータのバッファサイズが十分に大きくない場合の評価と制御方式の検討などが挙げられる．

文 献

- [1] 新井，溝口，“モバイルセキュリティを強化したシンクライアントソリューション”，情報処理，vol. 47, no. 10, pp. 1127–1136, October 2006.
- [2] R. W. Scheifler and J. Gettys: *X Window System: Core and Extension Protocols : X Version 11, Releases 6 and 6.1*, Butterworth-Heinemann, 1997.
- [3] Citrix Systems, <http://www.citrix.com/>
- [4] MSDN library, “Understanding the Remote Desktop Protocol (RDP)”, <http://support.microsoft.com/kb/186607>
- [5] A. Lai and J. Nieh: “On the Performance of Wide-Area Thin-Client Computing”, *ACM Transactions on Computer Systems*, vol. 24, no. 2, pp. 175–209, May 2006.
- [6] N. Tolia, D. G. Andersen and M. Satyanarayanan: “Quantifying Interactive User Experience on Thin Clients”, *IEEE Computer Society Press Computer*, vol. 39, no 3, pp. 46–52, March 2006.
- [7] G. Minshall, Y. Saito, J. Mogul, and B. Verghese: “Application performance pitfalls and TCP’s Nagle algorithm”, in *Proc. of 2nd Workshop on Internet Server Performance*, pp. 36–44, May 1999.
- [8] J. C. Mogul and G. Minshall: “Rethinking the TCP Nagle algorithm”, *ACM SIGCOMM Computer Communication Review*, vol.31, no. 1, pp. 6–20, January 2001.
- [9] W. R. Stevens: *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley, 1994.
- [10] J. C. Mogul: “The case for persistent-connection HTTP”, *ACM SIGCOMM Computer Communication Review*, vol. 25, no. 4, pp. 299–313, October 1995.
- [11] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach and T. Berners-Lee: “Hypertext Transfer Protocol – HTTP/1.1”, *RFC 2616*, June 1999.
- [12] J. Heidemann: “Performance Interactions Between P-HTTP and TCP Implementations”, *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 2, pp. 65–73, April 1997.
- [13] J. Heidemann, K. Obraczka and J. Touch: “Modeling the Performance of HTTP Over Several Transport Protocols”, *ACM/IEEE Transactions on Networking*, vol. 5, no. 5, pp. 616–630, October 1997.
- [14] X. Chen and J. Heidemann: “Preferential Treatment for Short Flows to Reduce Web Latency”, *Computer Networks*, vol. 41, no. 6, pp. 779–794, April 2003.
- [15] L. Guo and I. Matta: “The War Between Mice and Elephants”, *Technical Report BU-CS-2001-005*, May 2001.
- [16] K. Tokuda, G. Hasegawa and M. Murata: “Analysis and Improvement of the Fairness between Long-/Short-lived TCP Connections”, in *Proc. of IFIP/IEEE PfHNS 2002*, April 2002.
- [17] 小川，長谷川，村田，“シンクライアントトラフィックの性能向上手法の検討”，電子情報通信学会情報ネットワーク研究会（口頭発表），March 2007.
- [18] Y. Ogawa, G. Hasegawa and M. Murata: “Transport-layer optimization for thin-client systems”, in *Proc. of CQR 2007 Workshop*, May 2007.
- [19] The VINT Project, “The Network Simulator - ns-2”, available from <http://www.isi.edu/nsnam/ns/>.