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

2007－ＱＡ１－22（４）
2007／1／19

長距離広帯域ネットワークでの TCP/IP Acknowledge Packet 受信の影響ついて

玉造 潤史† 吉野 剛史‡ 稲葉 真理‡ 平木 敬‡

†東京大学大学院理学系研究科 〒113-0033 東京都文京区本郷 7-3-1
‡東京大学大学院情報理工学系研究科 〒113-0033 東京都文京区本郷 7-3-1
E-mail: {junji, ysn, mary, hiraki}@is.s.u-tokyo.ac.jp

あらまし Long Fat-pipe Network (LFN) としての10Gbpsネットワークが一般化した現在,高速高遅延のTCP通信で性能を引き出すことは高速ネットワーク活用のために重要な問題である. 本稿では,TCP/IP通信における受信完了パケット(Acknowledge Packet)が通信コントロールを支配し,通信性能に影響を与えていることを示す. Ack Packet の振る舞いを調べるため, 疑似的なLFN上で linux-2.6.19.1 カーネルの ack パケットに変更を加え,その変化を調べた. また, ack パケットの通信レートを調整した場合の振る舞いを SC2006 で構築した実ネットワークで実験した結果を示す.

キーワード TCP/IP 通信, Acknowledge Packet, Long Fat Network

Influence of TCP/IP Acknowledge Packet receiving on Long Fat-pipe Network

Junji TAMATSUKURI† Takeshi YOSHINO‡ Mary INABA‡ and Kei HIRAKI‡

† Graduate school of Science, University of Tokyo 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033 Japan
‡ Graduate school of Information Science and Technology, University of Tokyo 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033 Japan
E-mail: {junji, ysn, inagami, sugawara, mary, hiraki}@is.s.u-tokyo.ac.jp

Abstract Recently Long Fat-pipe Network (LFN) with large bandwidth is popular in general. To exploit TCP/IP communication performance on LFN is getting more important to utilize network bandwidth. We show the influence of TCP/IP Ack packet to communication performance in this paper. We used linu-2.6.19.1 TCP/IP stack to analyze Software TCP/IP ack packet behavior modifying in TCP BIC algorithm driver and measured the change of performance. And we measured the behavior of Ack packet rate modification on Real LFN on SC2006.

Keyword TCP/IP, Acknowledge Packet, Long Fat-pipe Network

1. はじめに

10 ギガビットネットワークによる Long Fat-pipe Network (LFN) の構築は 10GBase-LW という SONET-SDH / OC-192 による WAN 物理レイヤをイーサネットとして接続できる方法により既に一般化し利用可能となっている.また,Layer 1 接続性もコントロールできる Cisco ONS-15454 や Nortel HDxC といったネットワークコントローラの普及によりアメリカの National Lambda Rail(NLR)のような 10 ギガビットの L1/L2 ネットワークが構築され,ユーザが利用可能となってきている.そのため,科学研究などでは既に多くの長距離広帯域ネットワークが存在し,利用されている.

長距離広帯域のネットワーク上でも通信はインターネット上と同じ IP が使われ,UDP/IP, TCP/IP が高速のデータ通信プロトコルとして用いられている.現在はまだ 10 ギガビットの高速通信ではユーザがデータコントロールを行う UDP プロトコルを用いられることが多い.これは TCP/IP プロトコルが保障するデータの到着性のために必要なホスト処理能力が大きいことと,ネットワーク上での輻輳を抑制する輻輳ウイン

ドウコントロールを行うために安定なネットワーク通信性能が必須となるためである.特にすべてのパケットロスは輻輳が起きていると判定されるためホストのI/O バスやネットワークインターフェース内部などネットワーク以外の場所で起こるパケットロスもホストにとってはネットワーク上の輻輳として認識され,コントロールされる.輻輳が起こった場合 TCP/IP プロトコルは輻輳ウインドウのサイズを半分にして,送出量を減少し,輻輳からの回避を行う.この振る舞いはエラーのないネットワーク状態においては正しい振る舞いである.しかし、サイズの小さいパケットロスを回避することはすでに selective acknowledge によって実現されている.しかし,一般化したとは言え,10ギガビットの通信速度は非常に高速であるため,パケットロスが起こった場合に影響するサイズも非常に大きくなる.物理回線の特性にも依存するが,10 ギガビットの回線が安定的に性能を発揮することが難しいということと,ネットワーク機材が非常に新しいものとなり安定して性能を得ることが難しいこと,実際のトラフィックによるテストや評価が難しいため,実際のネットワークをエラーフリーで用いることが難しいことが TCP/IP 通信が高速高遅延のネットワークで難しいと認識される主要因となっていると考えられる。 言い換えるとTCP/IP 通信が性能を発揮できるネットワーク環境の前提として、パケットロス要因を排除し,問題のないネットワーク状況を作り出すことが必須であり、これらを実現したとしても,依然として LFN での TCP/IP 通信は難しい.

これまで,我々は 10 ギガビットネットワーク上で構築された LFN を用いて高速 TCP/IP 通信の実験を行ってきた.[1],[2],[4],[5].ここで明らかになったのは,実ネットワーク上では,パケットの到着時間が物理レイヤのフレーミングにより変化し,通信の状況を変化させているということである.特にこの現象はショートパケットである Ack パケットにおいて顕著であり,その影響を可能な限り抑えなければ安定した通信が難しいという結果を示していた[7].この結果は,TCP/IP 通信における困難度は性能を発揮すべきデータパケット側における問題よりもショートパケットとして実現されている,コントロールパケット(Ack パケット)の側が通信の安定性と性能を支配していると考えられることを示している.

本稿ではこのような Ack パケットの特性に着目し,(1)delayed Ack パケットのパケット比率を変化させる(2)Ackパケットの送出レートをハードウエアでコントロールするの 2 通りの方法で通信状況を変化させコントロールとしての通信がデータ通信にどのような作用/影響を与え,どのような結果をもたらすのかを調べた.

2. ソフトウエア TCP/IP における Ack パケットの振る舞い

TCP/IP 通信において通信速度は厳密にはコントロールされておらず,データがネットワーク上にあってコントロールされた状態にあるパケット(Inflight パケット)が輻輳により消失しないようにするためのバッファである輻輳ウインドウ(Congestion Window)のサイズによって制御されている.TCP/IP スタックは,この速度はこの輻輳ウインドウコントロールのサイズを,すでに送信先に到着したパケットの到着を知らせる Ack パケットの到着と,ホスト内部にあるウインドウバッファの空き空間があるかを確認して決定する.輻輳ウインドウが大きくなると,TCP/IP スタックの送信バッファ(send buffer)に書けるデータが増加し,書かれたデータをネットワークインターフェース(NIC)がネットワークに送出する.物理レイヤがイーサネットで実現されている場合,送出の速度はネットワークの物理レイヤの速度であるため,各瞬間瞬間では物理デバイスの速度で送出されている.しかし,輻輳ウインドウのサイズ分しかネットワーク上には送出されないためマクロな平均時間では送信されたデータ/時間の速度で通信しているように観察される.

この通信状態で,輻輳ウインドウを成長させる要因となるのが Ack パケットの到着である.TCP/IP スタックは,Ack パケットから送信先に到着したデータパケットの量が十分であり,送信先の受信バッファが十分であることを確認する.この動作を安定して行うことが送信元のホストで行われた場合に安定して輻輳ウインドウが成長し,平均的通信速度が上昇する.

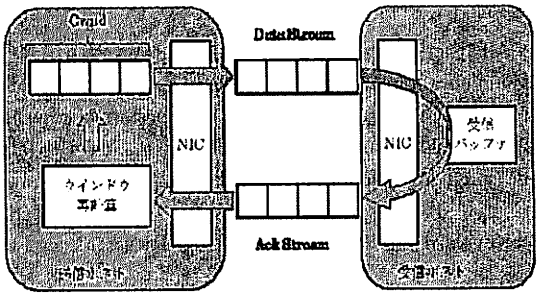

図 1:TCP/IP 通信のコントロール図

しかし,先に述べたようにイーサネットにおける通信速度は物理メディアによって決定されるため,ショートパケットである Ack パケットの流れ(ストリーム)が変化してしまうとサイズが小さいにもデータパケットと同数以上のパケットを受信することになる.NIC

はパケットの到着時に CPU に対して割り込みを発生するため,Ack ストリームの処理はデータパケットの受信側と同数のパケットの受信を行う.送信元のホストでは,データパケットの生成処理のみならず,同数のパケットの到着と輻輳ウインドウの処理とを行わなければならず,性能が向上しない問題を発生する.

この問題を解決するために用いられているのが linux では NAPI というパケットの受信動作の coaleacing であり,Ack パケットの送出時に coaleacing を行う delayed ack である. NAPI は NIC がパケット受信時に発生する割り込みを抑止し,定期的なポーリングをおこなうことで I/O の負荷を減少させてしまうという方法である.一方,delayed ack は TCP/IP における Ack パケット送出を一時的に抑止し,複数個分のパケット受信をまとめて通知する方法である.この方法は本質的に TCP/IP スタック自身の負荷を下げ高負荷となりやすい送信元ホストの受信パケット数と処理を減少させるという効果を持っている.すなわち,delayed ack を用いることで高速で高負荷な状態での通信を実現するには不可避な方法であると考えられる.

問題としては,ナイーブな実装を行った場合に送出元となるホストに到着する Ack パケットの到着数で輻輳ウインドウの再計算を行った場合,ウインドウの成長が遅くなるような状況が考えられる.今回おこなった測定の結果,ウインドウの成長における Ack パケット数への相関はなかった.しかし,輻輳ウインドウのコントロールアルゴリズムは非常に多数提案されており,その実装も多数あり,これらがそれぞれにウインドウ計算を実装しているため想定していないようなウインドウ拡大をしていると思われる場合が多数ある.

2.1. 実験内容

まずはじめに Ack パケット数が送出元ホストとその通信性能に対してどのような影響を持つかを測定するため,TCP/IP スタックが行う delayed ack パケットコントロールを改変して性能の変化を調査した.

用いたのは linux kernel version 2.6.19.1 という最新の安定版カーネルである.Linux カーネルでは,バージョン 2.6.16 以降で TCP/IP スタックのうち輻輳ウインドウコントロールに当たる部分がローダブルな TCP ドライバとして実装されている.また,delayed ack を実装しているのは BIC アルゴリズムとその改変版である CUBIC アルゴリズムだけであり,本実験では,BIC アルゴリズムのドライバを用いて通信を行った.

BIC ドライバでは delayed ack と,スライディングウインドウを合わせて用いており,高負荷時に安定して性能を出せる唯一のドライバである.

ここでドライバにおける packet/Ack の比率を 1/2^n ずつデータパケット,Ack パケットの比率を変化させる改変を行った.

CPU	Dual AMD opteron 250 (送信側), Dual AMD opteron 248 (受信側)
Memory	2GB(Single Memory Bus)
MotherBoard	Rioworks HDAMA rev D
NIC	Chelsio N210
OS	Linux-2.6.19.1 tcp_bic driver

表 1:実験ホスト仕様

通信アプリケーションは iperf version 2.0.2 を用い,NAPI については,データの送出・受信側の割り込みは固定的に 10μs ごとに,Ack 側の割り込みは 100μs となるように設定した.Iperf の実行パラメータは下記の通りである.RTT はウインドウのスケーリング時間が 30 秒以内になるように設定した.アプリケーションバッファは送受信ともそれぞれ最適な性能を示すように調整した結果である.TCP ウインドウバッファは通信最大速度を約 7 Gbps として理論値から求めている.

RTT	200ms
Application Buffer	128kB (送信側) 1024kB (受信側)
TCP Window Buffer	200MB

表 2:通信パラメータ

2.2. 実験環境

測定環境は図 2 のようになっており,2 台のホストを 10Gbps ワイヤレートで通信できるスイッチである Force10 E600 を用いて接続し,その間に疑似的な遅延を挿入するため Anue H シリーズネットワークエミュレータを接続し,TCP 通信の全パケットヘッダを解析するため,ログ採取装置として 10GbE ヘッダキャプチャリングを行うための装置 Tapee (Traffic Analysis Precise Enhancement Engine)[6]を接続した.

TAPEE は,FPGA で精度の高いタイムスタンプをパケットヘッダに打つことができ,複数のパケットヘッダをまとめて送出しなおすことで通常の unix サーバ上で高速な通信であってもパケットの採取が可能である.また,ログの解析も unix サーバ上のプログラムで容易に処理することができるため,測定条件を変えながらの実験を容易にしている.

本接続環境における改変前の通信性能は約 7Gbps である.

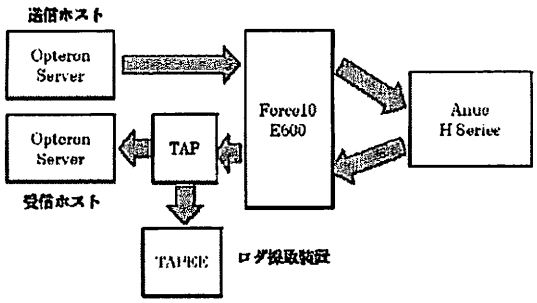

図 2:実験環境構成図

2.3. 実験結果

TCP BIC ドライバの delayed ack 制御部分に改変をくわえ ack の $1/2^n$ ずつ減少させて TCP 通信をおこなった.この n はデータパケットと Ack パケットの比率を決定するシフト計算で用いている値である.この結果が,表 3 である.n を 1 から 6 まで変化させた時のピーク性能と安定性能を示している.すべての場合においてピーク性能は 7.22Gbps となった.また変化したのはピークを過ぎてからのパケットの安定性能である.ここでは Ack パケット数の変化により安定性能だけが変化している.

N	1	2	3	4	5	6
Peak	7.22	7.22	7.22	7.22	7.22	7.22
Stable	7.05	7.02	7.00	6.98	6.95	6.95

表 3:delayed ack を変化させた性能

この表 3 の結果は,ユーザアプリケーションからみたマクロな結果である.

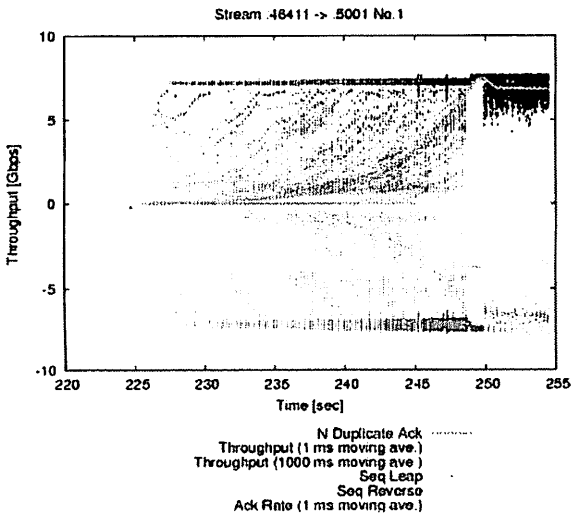

図 3:データと Ack パケットの通信速度

この通信の状況を TAPEE による詳細なデータをグラフ化したものが図 3 である.図 3 において上半分がデータストリームの通信状況であり,下半分が Ack ストリームの速度を示している.Ack 側は Ack パケット内の AckDiff を計算して Data ストリームの速度をグラフ化したものである.太線が 1 秒単位の平均速度を示しており,これは先の表の結果と同じ意味をもっている.分布をもって見えているのがパケットの 1ms あたりの平均速度である.本実験環境ではパケットロスはないので,データパケットストリームを見てみるとスケーリングしている部分(通信開始から約 24 秒間)と,スケーリングが終了してピークに達したとき(約 2 秒間),その後という 3 段階のフェーズに分けることができる.

この同じ通信の Ack パケットを AckDiff を計算せずにパケットカウントをそのままプロットしたものが図 4 である.Ack パケットの個数を見ると 2 つのフェーズにしか分れないことが分かる.これは,ひとつはウインドウが成長している区間(通信開始からの 24 秒間)で,この区間はデータパケット同様分散をもって分布している.ところが,ウインドウの成長後に突然 Ack のパケット個数が減少している.

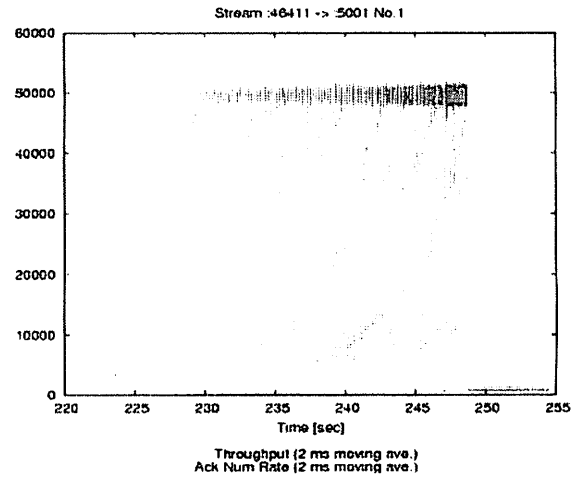

図 4:Ack パケットの個数

Ack パケットの個数の推移から Linux TCP の通信コントロールについて(1)ウインドウのスケーリング部分は delayed ack の機能が効いていない(2)ack の個数の変化があっても通信速度が変わらないことからウインドウコントロールは到着パケット数ではなくデータ量できちんと計算されていることが分かる.

3. 実 LFN 環境における Ack packet コントロールによる通信レート調整

続いて,Ack パケットの送出速度コントロールを行い通信に対してどのような影響を持つのかを調べた.

パケットの送出速度のコントロールには TAPEE を実現しているハードウエアである TGNLE[3]のパケットコントロール機能を用いた.データパケット側は一切変化させずに Ack パケットだけの送出速度を均質にしたものである.本実験は,SC2006 における実験の一部として行われたものであり実 LFN におけるものである.実 LFN における TCP/IP 通信は擬似 LFN 環境よりも難しいが今回はハードウエアによる TCP である TOE(TCP Offload Engine)機能を用いて行っているため顕著な影響は見られなかった.

3.1. 実験環境

実験に用いたネットワークは東京からアメリカ合衆国タンパに到るネットワークである.途中経路として JGN2 および NLR(National Lambda Rail)の L2 ネットワークである FrameNet を用いて L2 の接続として接続されたものである.図 5 にネットワーク構成図を示す.

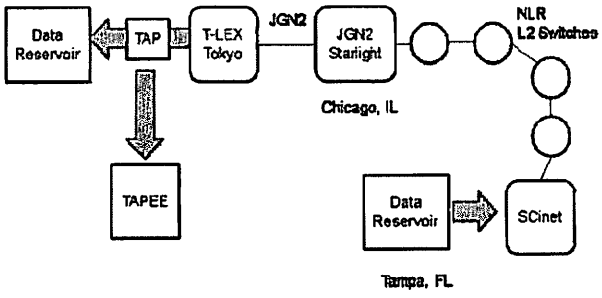

図 5: 海外回線構成図

通信に用いたホストは以下のとおりである.先の実験と同じく Opteron サーバであるが,NIC が TOE 機能をもつ ChelsioT110 で構成されている.ネットワークの RTT は約 250ms である.

アプリケーションは Data Reservoir の rawcp アプリケーションで,32 本のディスクからディスクへの 10Gbps での実転送である.この通信が 1 台のサーバから 2 枚の NIC,2 本の TCP ストリームで実現されている.以下の結果は比較のため 2 本のストリームのうちの片方だけのグラフを示す.TCP 通信であり,最大性能での通信を行うアプリケーションであるため,ディスクデータによるものではあるがネットワーク的には iperf と同じような振る舞いをするアプリケーションである.

3.2. 実験結果

WAN-PHY による帯域限界 9.2Gbps を効果的に使うために 1 本のストリームを最大で約 4Gbps で通信するように設定した.Ack パケットはすべて 60000pps になるようにして送出し,それをタンパから rate コントロールして送信し,東京側で受信する直前の状態を TAPEE で採取して,通信の状況を観測したものである.

その結果が図 6 である.TOE による通信であるため NIC からの送出レートまでコントロールされているためソフトウエアの場合の時のようなパケットの分散は見られない.また,NIC が Ack パケットの受信能力を十分持っているので Duplicate Ack を非常にたくさん出している.この振る舞いはソフトウエアによる TCP にはないものである.

パケットの送出から RTT 時間たつまでの間,ウインドウの成長のために大量のパケットが送出されていることがわかる.その後 Ack パケットの送出レートコントロールが効き通信レートが下がりだすという現象として表れていることが分かる.

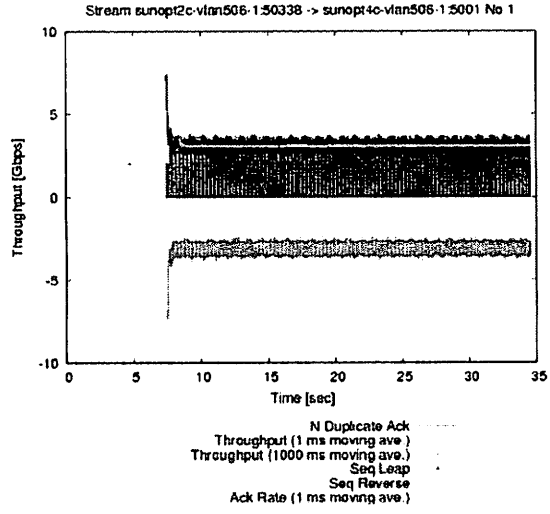

図 6:Ack パケットを rate コントロールした場合

この通信中の Ack パケットの個数をプロットしたものが図 7 である.Ack パケットは通信レートが低いため初めの区間で,パケットが rate コントロールのバッファを満たさずほぼそのまま通過しているが,その後たまり始める.しかし,Ack パケットの到着が遅くなることで,データパケットの通信速度が遅くなり結果としてほぼ一定の通信速度で通信できるようになっている.

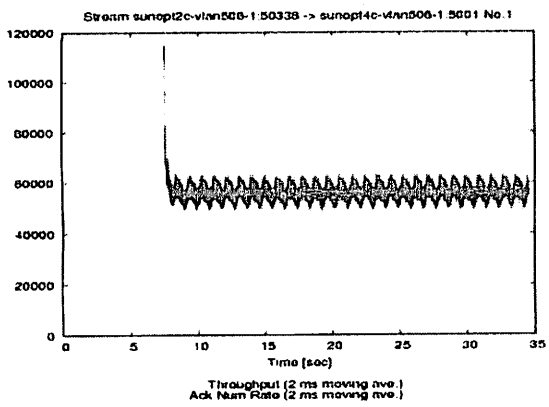

図 7:Ack パケットの個数の変化

4. 考察

これらの結果から TCP/IP の通信性能は Ack パケットの振る舞いが大きく影響しており,Ack パケットの制御をすることで,ハードウエアの TCP であっても速度のコントロールが可能であることが分った.

ソフトウエア TCP の場合,今回 BIC TCP を用いたが,ウインドウスケーリング時の振る舞いは BIC TCP の delayed ack を用いたものでは無いような状況がみられている.これは輻輳ウインドウサイズの計算は BIC TCP によって行われている.BIC TCP では,ある輻輳ウインドウサイズよりも大きくなった場合にウインドウサイズをバイナリサーチによって決定するというアルゴリズムになるように作られており,それまでは Reno 輻輳ウインドウコントロールアルゴリズムに従うはずである.ところが,RTT が大きい場合に通信速度が高いとウインドウサイズがすぐに大きくなりバイナリサーチモードに入ってしまっている.この振る舞いは BIC TCP の設計上は考えられなかった問題が現れてしまっている.これまで,複数の linux カーネルにおいて TCP 通信の実験を行っているが,BIC TCP だけが大きなスケーリング時間を必要としており,その変化が RTT だけに依存している状況があるため,この部分の詳細な検討は必要であると思われる.

ショートパケットの受信性能は徐々に改善しており,先の論文で実験した 2.6.14.7 のカーネルよりも安定的に受信できるようになっていて delayed ack によるパケットの減少が必ずしも必要ではない状況は見られる.そして,ある程度のパケット rate を維持している場合の方が安定した通信を行っているような状況も見られたためより最適な delayed ack のデータパケット/Ack パケット比率があると思われる.

5. まとめ

本稿では TCP 通信の Ack パケットの振る舞いに着目して通信実験を行った. 時間的な問題で delayed ack の制御については実 LFN 環境での実験を行うことができなかったが,既に Ack パケットストリームの変化があることは分かっているので,今後この点については詳細を計測を行いたい.

TCP 通信においてデータパケットの方が通常性能が厳しくなるため性能計測上でも重視されることが多い. しかし本稿では Ack パケットの振る舞いに着目して調査したところ,Ack パケットの通信自体を改善する余地,特に輻輳ウインドウの成長区間については重要な変更が必要であることがわかった.これらの点を踏まえ,今後 TCP 通信の Ack コントロールによる通信性能の向上と通信の安定性向上を目指したい.

謝　辞

本研究の実施に当たって東京大学情報基盤センタ加藤朗先生には多くの支援を頂いた. また,実験のため 10Gbps 回線を JGN2, IEEAF, NLR から提供を受けた.

本研究は,文部科学技術省 科学技術振興調整費「重要課題解決型研究等の推進－分散共有型研究データ利用基盤の整備」,科学技術研究費基盤研究 B(2)15300014「アプリケーショントランスペアレントな大域データインテンシブ機構」,および２１世紀 COE「情報科学技術戦略コア --- 大域ディペンダブル情報基盤で補助された.

文　献

[1] J.Tamatsukuri, K.Inagami, T.Yoshino, Y.Sugawara, M,Inaba and K,Hiraki, "Experimental Results of TCP/IP data transfer On 10Gbps IPv6 Network", 4th Workshop on Protocols for Fast Long-Distance Network (PFLDnet2006), Feb 2006.

[2] 中村誠, 玉造潤史, 菅原豊, 稲葉真理, 平木敬,"擬似ネットワーク環境における TCP/IP の性能評価", 電子情報通信学会技術研究報告 1A2005-7,pp.1-8,Jun 2005.

[3] 菅原豊, 稲葉真理, 平木敬, "細粒度パケット間隔制御の実装と評価", 情報処理学会技術研究報告, OS-100, pp.85-92, Aug 2005.

[4] 玉造潤史, 吉野剛史, 稲垣克史, 菅原豊, 稲葉真理, 平木敬, "Real Long Fat Network における TCP/IPv6 の通信性能評価", 電子情報通信学会技術研究報告 Vol. 106 No.62, IA2006-4, pp. 19-25

[5] 玉造潤史, 吉野剛史, 稲垣克史, 菅原豊, 稲葉真理, 平木敬, "10 ギガビットネットワーク上での高効率 TCP/IP 通信の実現", 情報処理学会研究報告 Vol. 2006-HPC-107,HPC (51), pp.299-304

[6] 吉野剛史, 玉造潤史,稲垣克史, 菅原豊, 稲葉真理, 平木敬, "ハードウェア・エンジンを用いた 10GbE 上の TCP 通信解析", 情報処理学会研究報告 Vol. 2006-HPC-107,HPC (52), pp.305-310

[7] M.Inaba, M.Tanaka, J.Tamatsukuri, K.Hiraki, H.Imai,"Avoidable packet losses on Long Fat Pipe Network; Effects of bottlenecks and intermediate switches", NPC2006, Sep 2006.[