パケット解析と輻輳ウィンドウ解析による遠隔iSCSIアクセスの断続的パケット送出に関する考察

比嘉　玲華† 　松原　幸助†† 　岡廻　隆生†† 　山口　実靖††† 　小口　正人†

† お茶の水女子大学 人間文化創成科学研究科　〒 112–8610 東京都文京区大塚 2–1–1
†† ソフトバンクテレコム株式会社　〒 105–7316 東京都港区東新橋 1–9–1
††† 工学院大学　〒 163-8677 東京都新宿区西新宿 1–24—2
E-mail: †reika@ogl.is.ocha.ac.jp, oguchi@computer.org, †††sane@cc.kogakuin.ac.jp

あらまし　iSCSI は複雑な階層構造を持つため，性能を向上させるためには複数の層にまたがる最適化を行う必要がある．既存研究において複数の層にまたがる最適化を行ったところ，一定の性能向上は達成できたものの，高遅延環境においてはまだなお大きな性能低下が確認された．そこで本研究においては，パケット解析および輻輳ウィンドウ解析を行い，その結果に基づいて，iSCSI 遠隔ストレージアクセスにおける性能劣化の原因を検討していく．特に遠隔 iSCSI アクセス時に観察される断続的なパケット送出に関して，その振舞を詳細に解析し，原因についての考察を行なう．

キーワード　ストレージ，iSCSI, IP-SAN

A study of intermittent packet transfer of iSCSI remote storage access in long latency with packets and congestion window analysis

Reika HIGA†, Kosuke MATSUBARA††, Takao OKAMAWARI††,

Saneyasu YAMAGUCHI†††, and Masato OGUCHI†

† Ochanomizu University　2–1–1, Otsuka, Bunkyo-ku, Tokyo, 112–8610 Japan
†† SOFTBANK TELECOM Corp.　1–9–1, Higashishinbashi, Minato-ku, Tokyo, 105–7316 Japan
††† Kogakuin University　1–24–2 Nishishinjuku, Shinju-ku, Tokyo, 163–8677 Japan
E-mail: †reika@ogl.is.ocha.ac.jp, oguchi@computer.org, †††sane@cc.kogakuin.ac.jp

Abstract　iSCSI has a complex hierarchical structure , SCSI over TCP/IP over Ethernet. As a result, for the purpose of getting the better performance of iSCSI, multiple layers should be controlled. In our previous work, iSCSI remote storage access through multiple layers has been optimized. However, in the case of long latency, drastic performance deterioration has been observed. Thus, in this paper, we have monitored packets and parameters in the kernels including TCP congestion window. Based on the results, we have analyzed the factor of iSCSI performance deterioration.

Key words　storage, iSCSI, IP-SAN

1. はじめに

コンピュータシステムにおける処理データ量の増大に伴い，効率的にストレージを管理したいという要望が高まっている．そこで SAN (Storage Area Network) が登場し，広く用いられるようになった．SAN の中でも次世代 SAN として期待されているのが IP ネットワークを用いた IP-SAN である．iSCSI はその IP-SAN の代表的なプロトコルであり，SCSI コマンドを TCP/IP パケットでカプセル化する規格である [1] [2]．iSCSI を用いることにより広域環境における IP-SAN を低コストで構築でき，遠隔地のデータセンタなどにデータをバックアップすることが容易となるため，ストレージのアウトソーシングといったサービスへの利用が期待されている．

しかし現状において iSCSI は，複雑な階層構成のプロトコルスタックで処理されており，バースト的なデータ転送も多いことから，通常のソケット通信と比較して，特に高遅延環境にお

いては性能の劣化が著しい [3]．さらに下位基盤の TCP/IP 層が提供できる限界性能を超えることはできない．従って iSCSI を用いたストレージアクセスにおいては，iSCSI 層だけではなく複数の層にまたがる制御を施すことによる性能向上が期待される．既存研究において，複数の層にまたがる最適化を行った結果，RTT32ms においてデフォルト時よりも約 4 倍の性能向上が得られた．しかし，高遅延環境においては，なお性能低下が著しいことも確認された．また，4MB のブロックサイズで iSCSI write アクセスを実行しプロセス中の各処理時間を測定することにより，高遅延環境下で性能が劣化する原因を解析したところ，データ転送時間がボトルネックになっていることが確認された．

そこで本研究では，パケット解析，輻輳ウィンドウ解析を行うことで，さらなる iSCSI 遠隔ストレージにおける性能劣化の原因の解明を進める．

また，遠隔バックアップを行う場合，データの書き込み量と読込み量とを比較すると，圧倒的にデータの書き込み量のほうが多いこと，遠隔ストレージ側では標準的なシステムのみを用いることができ，カスタマイズできないことが想定されるため，本研究においては，iSCSI シーケンシャルライトアクセスの性能向上に焦点を絞り，Initiator 側における解析を行う．

2. 研究背景

2.1 iSCSI

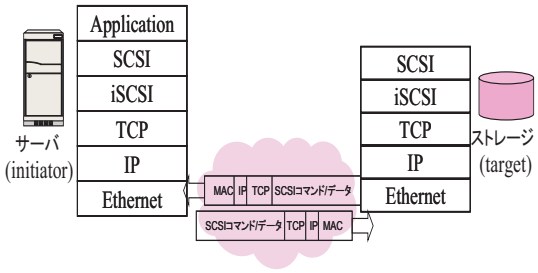

図 1　iSCSI

IP-SAN の代表的なプロトコルに iSCSI がある．iSCSI は SCSI コマンドを TCP/IP パケットでカプセル化する規格で，iSCSI により SAN を IP 機器だけで構築することが可能となる．一方で図 1 のように複雑な階層構成をとることになり，下位のプロトコルの限界性能を超えることはできない．また，iSCSI には長距離アクセスの実現が期待されているが，広帯域な回線を用いた場合には遅延帯域積の問題も存在する．そこで iSCSI 遠隔ストレージアクセスには複数の層にまたがる適切な制御が求められている．

2.2 TCP 輻輳ウィンドウ制御アルゴリズム

輻輳制御はネットワークの混雑解消の方法として TCP が実現する機能である．一般的な TCP の輻輳制御においては，通信開始時にスロースタートと呼ばれるアルゴリズムに従い指数関数的に輻輳ウィンドウが大きくなる．これによりトラフィックが急激に増加するため，ネットワークが輻輳状態になる可能性がある．これを防ぐため，スロースタート閾値という値を用意し，輻輳ウィンドウがその大きさを超えると輻輳回避と呼ばれるフェーズに入り，一次関数的な増え方となる．そしてエラーが検出されると輻輳ウィンドウは急激に低下し，通常これらを繰り返すことで輻輳ウィンドウの振舞の時間変化は一般に鋸型のグラフとなる．

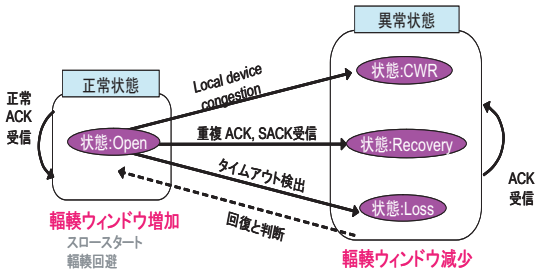

図 2　LinuxTCP の状態遷移

また LinuxOS における TCP の状態遷移を図 2 に示す．LinuxTCP においては，通信時の状態が正常であれば ACK の受信ごとに輻輳ウィンドウは増加するが，エラーが検出されると異常と判断され，輻輳ウィンドウは低下する．輻輳ウィンドウが低下する原因としては，送信側デバイスドライバのバッファが一杯になったことを示す Local Congestion を検出した場合 (CWR)，重複 ACK 又は SACK を受信した場合 (Recovery)，タイムアウトを検出した場合 (Loss) の 3 つが挙げられる．さらに Linux の TCP 実装では，通信中に一度設定された輻輳ウィンドウは，そのウィンドウ値を超えるデータ量が送られない限りは変化しないという特徴を持ち，この時スループットはほぼ一定の値で安定することが確認されている．

3. 実験システム

3.1 プロトコルアナライザ

高遅延環境において性能が著しく劣化する原因を解明するため，本研究ではまず，ネットワーク上を飛来するパケットを調べる．ネットワークからキャプチャしたトラフィックを直接高速アクセス可能な HDD に書き込む大容量ネットワークアナライザである ClearSight 社の Network Recorder [4] を設置し,iSCSI アクセス時のパケットキャプチャを行った．

3.2 TCP 輻輳ウィンドウモニタツール

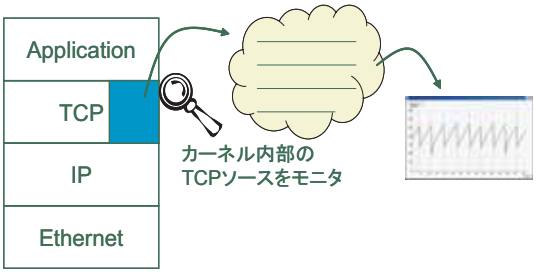

図 3　TCP 輻輳ウィンドウモニタツール

本実験では次に，TCP 輻輳ウィンドウをモニタするツールを構築した．図 3 に示すように，カーネル内部の TCP ソースにモニタ関数を挿入しカーネルを再コンパイルした．これにより

モニタできるようになった値には，輻輳ウィンドウの他，各種エラーイベント (Local device congestion，重複 ACK，SACK 受信，タイムアウト検出) の発生タイミングなどがある．

3.3 実験環境

本研究において,Initiator と Target 間は GigabitEthernet で接続し,TCP/IP コネクションを確立した.Target のストレージには SAS ディスクを用い RAID コントローラによる RAID0 構成で接続した. 使用した実装システムと実験環境を図 4 および表 1 に示す.

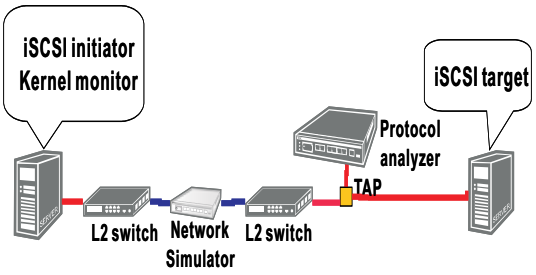

図 4　実装システム概要

表 1　実験環境

OS	Red Hat Enterprise Linux 2.618-8.e.15
CPU	Quad Core Intel Xeon 1.6GHZ
Main Memory	2GB
NIC	Intel PRO/1000PT Server Adaptor on PCI Express
HDD	73GB SAS×2(RAID0)
RAID Controller	SAS5/iR
iSCSI	Initiator : open-iscsi-2.0-865
	Target : iSCSI Enterprise Target(IET)-0.4.15
Network Analyzer	ClearSight Network Recorder
Network Simulator	ANUE

4. iSCSI アクセス時の最適化とデータ転送の解析

iSCSI アクセス時の複数の層にまたがる最適化を行なった上で iSCSI write アクセスモデルを構築しその解析を行なった．その結果，高遅延環境における iSCSI アクセスが理論値よりも低下する理由は，RTT によらずに一定であるはずのデータ転送時間が RTT に比例する値になっていることが原因であるとわかった [5].

4.1 複数の層にまたがる最適化

iSCSI は複雑な階層構造をとなっている．そこで，図 5 のように，複数レイヤにまたがる最適化を行った．SCSI/iSCSI 層においては，iSCSI パラメータ最適化を行い，その結果，RTT32ms の場合に約 4 倍のスループットの向上が確認できた．

ただし SCSI/iSCSI 層の最適化による性能向上はウィンドウサイズにより制限される可能性があるため，次に TCP/IP 層における最適化を試みた．具体的にはスループットと関係の深い輻輳ウィドウの値を決める輻輳ウィンドウ制御アルゴリズムを変更して性能測定を行った．その結果，ソケット通信時には見られたアルゴリズムごとのスループットの違いは iSCSI 利用時には見られなかった．すなわち，本実験環境においては TCP 輻輳制御アルゴリズムの変更は，iSCSI 性能最適化には影響を与えないと言える．この原因としては，TCP の輻輳ウィンドウの違いによる性能向上分が，iSCSI のブロックアクセスのシーケンスに吸収されて消えてしまっていると考えられる．

最後に Ethernet 層における最適化として NIC のパラメータを変更し iSCSI 通信を行ったところ，RTT32ms の場合において約 5%の性能向上が確認できた．

このように iSCSI アクセス時の最適化によって約 4 倍の性能向上が得られた．この結果を，図 6 に示す．また，比較としてソケット通信，ローカルディスクアクセス性能も測定した．ソケット通信の測定には Iperf [6] を，ローカルディスク，iSCSI の測定には bonnie++ [7] を使用した．本実験においては，通信の妨げにならない程度の十分なウィンドウサイズを想定し，広告ウィンドウを 16MB に設定した．

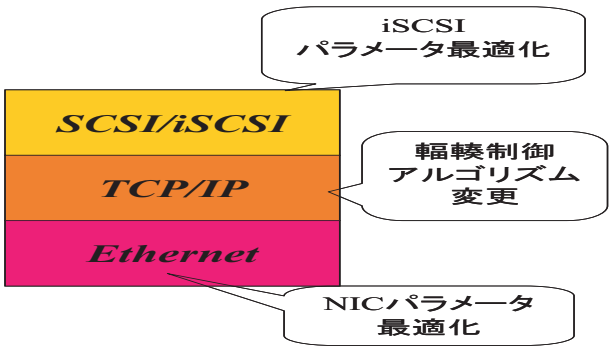

図 5　複数レイヤにまたがる最適化

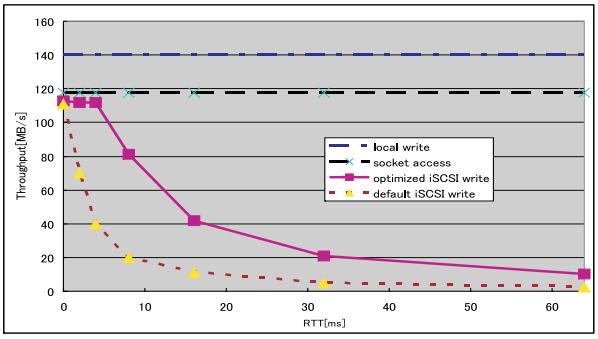

図 6　既存研究におけるスループット比較

図 6 からわかるように，ソケット通信の場合は高遅延環境においてもスループットを保っているのに対して，iSCSI 通信の場合は低遅延環境で保たれていたスループットが高遅延環境において性能低下が著しい．

4.2 iSCSI write アクセスモデルとその解析

4.2.1 iSCSI write アクセスモデル

高遅延環境において性能が低下する原因となるボトルネックを調べていった．

dd コマンドを用いて実行される iSCSI ブロックアクセスのパ

ケットをプロトコルアナライザを用いて解析したところ，様々な大きさの複数のパケットが入り混じって飛んでいたため，モデル化の検証に当たっては sg_dd コマンドを使用した．sg_dd コマンドは，dd コマンドと文法的に互換性があるが，dd コマンドとは異なり，iSCSI アクセスにおいて SCSI レベルで指定したブロックサイズによるアクセスが可能となるコマンドである [8]．カーネルを再構築したことで sg_dd コマンドでのアクセス時に最高で 4096KB のブロックサイズでのアクセスが可能となった．それに伴い，iSCSI パラメータの設定を FirstBurstLength,MaxBurstLength ともに 4,194,304 とした．

4096KB のブロックサイズで write アクセスを実行したときのプロセスは図 7 のようになる．このとき Ta とは Initiator 側における最初のパケット送出から最後のパケット送出までのデータ転送時間，Tb は Target 側で書き込みが終了し Initiator へ書き込みが終了したことを知らせるまでの時間，Tc は次の write が実行されるまでの時間である．遅延装置で設定した遅延時間ごとに Ta，Tb，Tc，RTT を測定することにより，高遅延環境下で性能が劣化する原因を解析する．2048KB，4096KB のブロックサイズで write アクセスを実行した．このときの RTT は 0ms，2ms，5ms，10ms，20ms，50ms である．

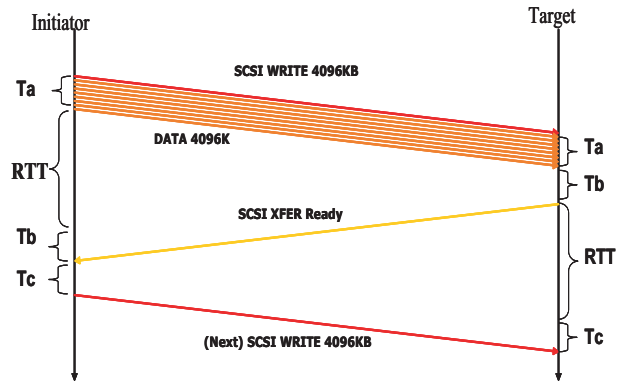

図 7　iSCSI write アクセス実行図

4.2.2　解 析 結 果

Ta，Tb，Tc，RTT をアナライザを用いて測定した結果，Tb と Tc はほぼ定数であること，RTT は遅延装置で設定した値とほぼ等しいということが確認された．しかし，Ta は図 8 に示すように RTT に比例する値で，RTT の増大と共に増加していた．すなわち高遅延環境における iSCSI アクセスが理論値よりも低下する理由は，RTT によらずに一定であるはずのデータ転送時間が RTT に比例する値になっていることが原因であるとわかった．

5.　アナライザを用いたパケット解析

前節に述べた解析結果より，高遅延環境における性能低下の原因が，データ転送処理にあることがわかった．そこで，本節ではアナライザを用いてどのようなパケットがやりとりされているのかを詳細に調べていく．

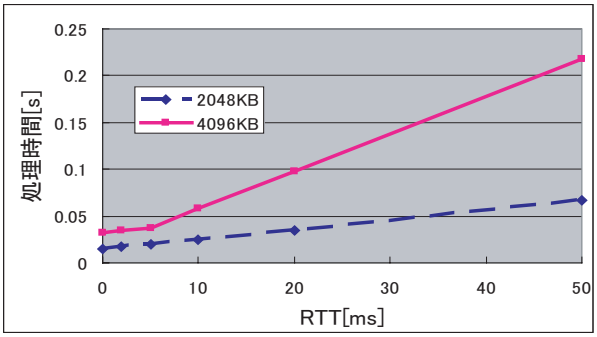

図 8　データ転送時間 Ta の測定結果

5.1　Initiator における送出パケット解析

RTT20ms，ブロックサイズ 4MB の iSCSI アクセスを実行した際の Initiator 側から Target 側に向かって送出されたパケットをアナライザを用いて解析を行った．そのときのパケット解析結果を図 9 に示す．グラフは横軸が時刻、縦軸がパケット番号を表している．パケットと比較するため，write10 コマンドと dataout コマンドの送出タイミングを上部に並べて示した．図 9 からわかることは，write10 コマンドの後に dataout コマンド 15 個が繰り返されていること，write10 パケットの後には 4MB のパケットが繰り返されているということである．

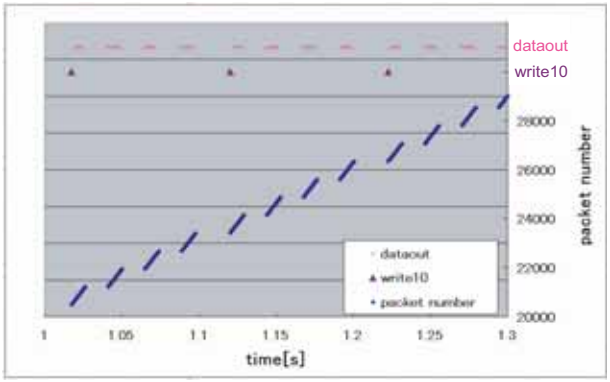

図 9　RTT20ms におけるパケット解析

図 9 周期のうちの一周期を拡大したものを図 10 に示す．

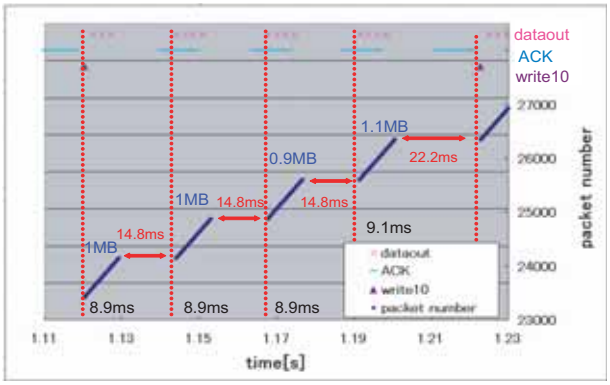

図 10　拡大した RTT20ms におけるパケット解析

図 10 より，短い時間に連続してパケットが送信された後，突然パケットの送出が止まっていること，一定時間の後に再びパ

ケットの連続送信が行われていること，パケットの送出量は約 700 個であること，その送出再開のきっかけとなっているのは write10 コマンド，もしくは dataout コマンドであること，それらパケットの送出から次の再送出までの間隔は RTT にほぼ等しい約 20ms であることがわかる．また，RTT を変化させて同じ実験を行なったところ，パケットの送出から次の再送出再開までの間隔は RTT にほぼ等しい値になった．

5.2 TCP ACK パケットの解析

送出再開の前後にはどのような現象が起こっているのかを解明するために，送出再開の直前のパケットを詳しく調べたところ，Target 側から Initiator 側への TCP ACK のみが存在していた．そのときのパケットを図 10 の上部に並べて示す．図 10 は，Initiator と Target の双方向を流れる全パケットが記載されている．パケット送出再開の前後には Target から Initiator 方向には TCP ACK のみが存在していて，write10 コマンドで送出が再開される場合は全ての ACK が帰ってきて初めてパケット送出再開が行なわれていることが確認された．

5.3 アナライザを用いたパケット解析の考察

RTT20ms，ブロックサイズ 4MB の iSCSI アクセスを実行時のパケットをアナライザを用いてキャプチャし詳細に解析した結果からは，短い時間に連続してパケットが送信された後突然パケットの送出が止まっていること，一定時間の後に write10 コマンド，もしくは dataout コマンドをきっかけに再びパケットの連続送信が行われておりそれらの間隔は RTT に等しい約 20ms であること，また，送信再開の前後には TCP ACK のみが受信されていることもわかった．この結果から，パケットの送信の断続性の原因としては，まず最初に輻輳ウィンドウ切れ，すなはちウィンドウを使い切ったことが可能性として考えられる．そこで，次の節でカーネルモニタを用いてこの時の輻輳ウィンドウの値を調べた．

6. カーネルモニタを用いた輻輳ウィンドウ解析

6.1 輻輳ウィンドウ解析

スループットと輻輳ウィンドウには密接な関係があることが知られている．そこで，Initiator 側でカーネルモニタと tcpdump を使って輻輳ウィンドウの値とパケット送出量の関係を調べた．RTT20ms，ブロックサイズ 4MB の iSCSI アクセスを実行したときの輻輳ウィンドウとパケット解析の結果を図 11 に示す．RTT20ms において 4MB を非同期に送信するには，輻輳ウィンドウは約 3000 が必要であるが，図 11 に示されたように，輻輳ウィンドウは約 1200 であり十分な大きさではないことがわかった．

6.2 輻輳ウィンドウ解析の考察

図 11 を一見すると，輻輳ウィンドウが十分な値でないことより輻輳ウィンドウ切れがパケットの送出停止の原因として考えられるが，図 10 と図 11 より，そのことが原因ではないことが推測される．すなはち図 10 からパケットの一周期あたりの送出量は約 700 であったが，輻輳ウィンドウの値は 1200 であることが図 11 により確認された．このことより，パケットの送出停止は輻輳ウィンドウを使い切ったことが原因でないこと

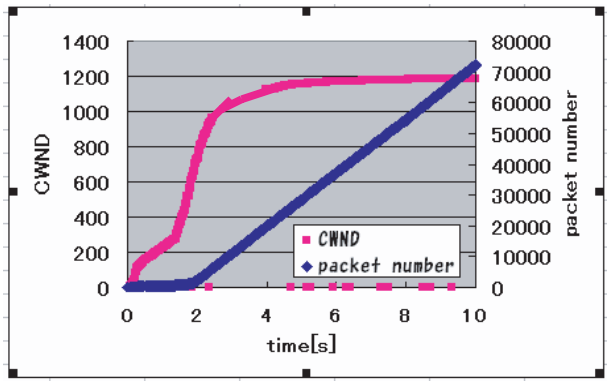

図 11 RTT20ms における輻輳ウィンドウ解析

になる．もし，輻輳ウィンドウが原因でパケットの送出停止が起こったのなら，輻輳ウィンドウ 1200 の分だけのパケット，つまり約 1.8MB のパケットが送出されることが可能だが，最大でも 1.1MB の大きさしか送出されていないからである．

7. 非同期送信しているデータ量の時間変化

この節では，オンザフライの状態のパケット数を調べることで輻輳ウィンドウが本当に余っているのかを確認する．RTT20ms，ブロックサイズ 4MB の iSCSI アクセスを実行したときのパケットをキャプチャし，アナライザを用いて出力したファイルを解析して，ACK を受けずに非同期送信しているデータ量の時間変化を求めた．そのときのグラフを図 12 と図 13 に示す．図 13 は図 12 の拡大図である．これらのグラフは横軸が時刻，縦軸が送信されてまだ ACK が受信されていないパケット数を表す．

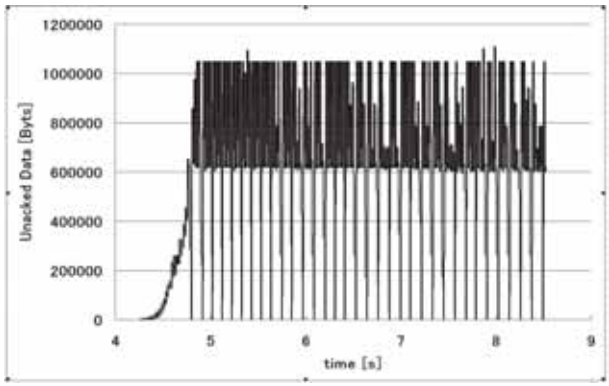

図 12 データ量の時間変化

図 12 と図 13 より，実際にネットワーク上を飛んでいるパケットは最大で約 1.1MB であり，輻輳ウィンドウを使い果たしていない状態であることが確認された．

8. ま と め

本研究では，iSCSI 遠隔ストレージアクセスの性能を高めるために，複数の層にまたがる最適化を行った．その結果，最適化 iSCSI はデフォルト状態の iSCSI と比較して約 4 倍の性能向上を達成することが出来た．しかし，なお高遅延環境におけ

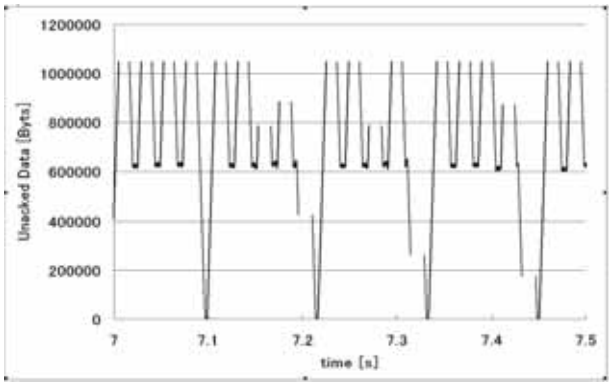

図 13　拡大したデータ量の時間変化

る性能の低下が著しいため，高遅延環境における性能低下の原因をより深く調べるために，iSCSI ブロックアクセスのモデル化を行い解析した結果，高遅延環境における性能低下の原因がデータ転送時間であることが判明した．

本稿において，ネットワーク上を飛びかうパケット解析とカーネル内部の輻輳ウィンドウ解析を行った．その結果，パケットの送出は断続的であり，送信開始の前後には TCP ACK のみが受信されたことが確認された．このことから，iSCSI の送信において停止・開始の制御は TCP レベルで行われていることが推測され輻輳ウィンドウを使い切ったことがパケット送出停止の原因である可能性が考えられる．しかしさらに調べた結果，パケットの 1 周期の送出量はその輻輳ウィンドウの値を使い切る量ではなかったことがわかり，また，送出の再開が write10 コマンド，dataout コマンドをきっかけに始まっているが，パケットの中身であるコマンドの判別は TCP レベルでは不可能である．

従って，iSCSI の送信において停止・開始の制御は TCP レベルで行われているが，その制御は輻輳ウィンドウの値だけによるものではなく，他の要因が存在していることが推測される．

9.　今後の課題

今後の課題としては，更なる解析が必要である．具体的には，カーネルモニタを使用したログの更なる解析である．ログの解析によって，カーネル内部の処理のどの部分に時間を費やしているのかを解析していく．そして，その結果を利用してシステムの改善を行う．

文　　献

[1] iSCSI Specification, http://www.ietf.org/rfc/rfc3720.txt?number=3270
[2] SCSI Specification, http://www.danbbs.dk/˜dino/SCSI/
[3] 山口実靖，小口正人，喜連川優：”高遅延広帯域ネットワーク環境下における iSCSI プロトコルを用いたシーケンシャルストレージアクセスの性能評価ならびにその性能向上手法に関する考察”，電子情報通信学会論文誌 Vol.J87-D-I，No.2，pp.216-231，2004 年 2 月
[4] http://www.toyo.co.jp/clearsight/product/analyzer.html
[5] 比嘉玲華, 松原幸助, 岡廻隆生, 山口実靖, 小口正人：”輻輳ウィンドウ及びパケット解析を用いた iSCSI 遠隔ストレージアクセスの評価”, コンピュータシステム研究会（CPSY），京都，2008 年 12 月
[6] http://dast.nlanr.net/Projects/Iperf/
[7] http://www.textuality.com/bonnie/intro.html
[8] http://sg.torque.net/sg/sg3_utils.html/