# パケットロス発生における iSCSI 遠隔ストレージアクセスに関する評価

比嘉　玲華[†]　神坂　紀久子[††]
山口　実靖[†††]　小口　正人[†]

コンピュータシステムにおける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. 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, 現状で iSCSI は高遅延環境になるほど性能が劣化してしまう.iSCSI を用いたストレージアクセス時には TCP パラメータの１つである輻輳ウィンドウとシステム性能が密接に関係しているため,TCP 輻輳ウィンドウ制御アルゴリズムを変えることでスループット向上につながる.

本稿では,iSCSI のパラメータを最適化することにより,iSCSI 遠隔地ストレージアクセスのスループットを向上させられることを示した. さらに, TCP の輻輳アルゴリズムを変えた場合の高遅延環境, パケットロス発生環境における iSCSI ストレージアクセスの性能評価を行った.

# Performance Analysis of iSCSI Remote Storage Access in the case of Packet Loss

Reika Higa,[†] Kikuko Kamisaka,[††] Saneyasu Yamaguchi[†††]
and Masato Oguchi[†]

As the volume of data computer systems process increases, it is important that storage is managed efficiently. Because iSCSI can configure the wide area IP-SAN with low cost and can make easy data backup of remote place, for example data center, iSCSI is expected to be used as Storage Outsoursing service. Throughput has a close relationship with the size of Congestion Window of TCP parameter on storage access using iSCSI. Therefore it is important for the better throughput to optimize Congestion Window Control Algorithm.

In this paper,we have opitimized iSCSI parameter for the better throughput. Moreover,we have changed Congestion Window Control Algorithm and evaluated the performance of iSCSI Storage Access in the case of a long latency and a packet loss environment.

## 1. はじめに

近年，インターネット技術の進展などにより，企業や会社内における個人の所有する情報量が爆発的に増えてきた. また、ストレージ内のデータをネットワークを介した遠隔地へバックアップする要望が多くなってきている. これに伴いストレージの増設，管理コストの増大などによる問題の解決を目的としたストレージ統合に SAN(Storage Area Network) が広く用いられるようになった. SAN とは，サーバとストレージを物理的に切り離し，各ストレージとサーバ間を相互接続してネットワーク化したもので，これにより各サーバにばらばらに分散していたデータの集中管理が実現された.

一般に SAN としてはファイバチャネルを用いる FC-SAN(Fibre Channel - SAN) が利用されている．しかし，FC-SAN はファイバチャネルにより構築されるため高価となり，また接続距離に制約がある．一方, SAN に IP ネットワークを利用した, 接続距離に制限のない IP-SAN として iSCSI が期待されている [1][2]. iSCSI は，これまで DAS(Direct Attached Storage) で使われてきた SCSI コマンドを TCP/IP パケット内にカプセル化することにより，サーバ (Initiator) とストレージ (Target) 間でデータの転送を行う.

iSCSI は,SCSI over TCP/IP over Ethernet という複雑な階層構成のプロトコルスタック構成となる. そのオーバヘッドなどが影響し, iSCSI による通信は特に高遅延環境においては大幅に性能が劣化すること

† お茶の水女子大学
Ochanomizu University
†† 情報通信研究機構
National Institute of Information and Communications Technology
††† 工学院大学
Kogakuin University

がわかっている [3]．一方, iSCSI はパラメータを最適化することによりスループットが向上することが確認されている [4]．

そこで, 本稿では環境に適合するようパラメータの値を最適化した. これによりスループットの向上は確認できたが, 高遅延環境においてはやはり性能の劣化がみられる. その原因として,TCP の輻輳ウィンドウ制御が原因の一つだと考えられるため, 輻輳アルゴリズムを変えてスループットの測定を行った. その結果を基に解析を行うことにより実験の考察を行う.

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み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. さらに, 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多いことが想定できる. そこで, 本研究においては,iSCSI のシーケンシャルライトアクセスの性能向上をめざす.

本稿は, 以下のように構成される．まず,2 章で研究背景,3 章で実験システムについて述べ,4 章で iSCSI のパラメータ設定について述べる．また.5 章で TCP の輻輳ウィンドウ制御アルゴリズムを変化させたときのソケット通信におけるスループット, 6 章ではその場合の iSCSI 通信におけるスループットを示し, 7 章において 5 章と 6 章の考察を述べる. そして, 最後に 8 章でまとめと今後の課題を述べる．

## 2. 研究背景

### 2.1 iSCSI

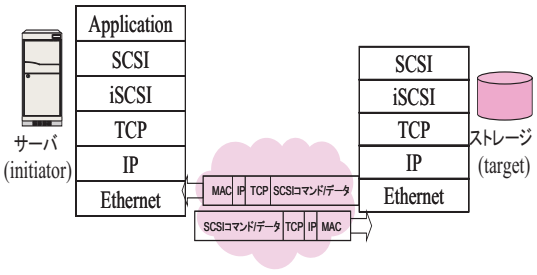


**図 1** iSCSI

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ることから遅延帯域積の問題も挙げられる．そこで下位基盤の TCP/IP 層の適切な制御が求められている．

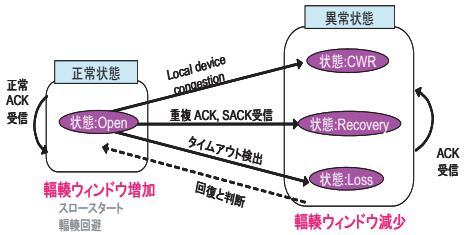


**図 2** LinuxTCP の状態遷移

### 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 が行う機能である．通信開始時にはスロースタートと呼ばれるアルゴリズムに従って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ので，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鋸型のグラフとなる．

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溢れることによる Local Congestion エラーを検出した場合 (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．

Linux における TCP 輻輳ウィンドウ制御アルゴリズムとしてカーネル 2.6 以降では様々なものが実装されており, コマンドを入力するだけでそのアルゴリズムを変えられるようになった．本研究において,Reno,BIC,Westwood,H-TCP の 4 種類のアルゴリズムについて扱う [5][6]．Reno は古典的なアルゴリズムであり,Reno を基にして様々なアルゴリズムが改良されてきた．Reno は輻輳をパケットロスによって検出し，パケットロス発生時の送信レートが利用可能帯

域であるとみなす．具体的には，重複 ACK を三回連続して受け取ると，パケットロスが発生したとみなして，輻輳ウィンドウを半分にする．そして，再度 RTT ごとに一つずつ輻輳ウィンドウを大きくして様子をさぐる．このように，ウィンドウサイズを徐々に大きくして，輻輳を検出したら一気に落とすというアルゴリズムである．

BIC は, 本実験環境における OS のデフォルトのアルゴリズムである．一般的な TCP 輻輳制御は，利用可能帯域に向かってリニアサーチを行うと考えることができるが，BIC はバイナリサーチを行う．

Westwood は, パケットロスが頻繁に起こる環境で効果的であり,Reno を基に無線環境で使う場合を想定して最適化されたアルゴリズムである．

H-TCP は広帯域かつ高遅延の環境で推奨されているアルゴリズムである．輻輳後に素早く元の状態へ回復させることを目指した設計となっている．

## 3. 実験システム

本章では, 本実験で使用した測定ツール, 実験環境および実験手順を示す．

### 3.1 Bonnie++

ハードディスクベンチマークツールとしては Bonnie++1.03 を用いた [7]．これはデータベースのような大規模なファイル操作のスループットを測定可能である．また比較的小さなファイルの作成, 読込み, 削除のスループットも測定する．本研究では, これを用いシーケンシャルライト (連続書き込み) のスループットを測定した．

### 3.2 Iperf

ネットワークのスループットを測定するために Iperf を用いた [8].TCP と UDP プロトコル転送時における帯域幅を測定するツールであり, メモリ・ツー・メモリのデータ転送をネットワークを介して実施し, その結果をレポートする機能を持つ．メモリ・ツー・メモリのデータ転送であるため, ハードディスクの読み書きがボトルネックになることがなく, 真のネットワークスループットを測定することができる．

### 3.3 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、本研究ではネットワーク上のパケットを調べていく．ネットワークからキャプチャしたトラフィックを直接大容量 HDD に書き込む大容量ネットワークアナライザである, ClearSight 社のプロトコルアナライザ Network Recorder を設置し,iSCSI アクセス時のパケットキャプチャを行える環境を整えた．図 3 に Network Recorder のラダー表示の例を示す．パケットの送受信の様子が表示され, 各々の到着時刻やパケット間の差分時間などが示されている．


**図 3** プロトコルアナライザ　ラダー表示

### 3.4 実験環境および実験手順

本研究において,Initiator と Target 間は GigabitEthernet で接続し,TCP/IP 接続を確立した.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 と表 1 に示す．

遅延装置を使い高遅延環境を作り出し, デフォルトの iSCSI とパラメータ設定を変更した iSCSI を起動して iSCSI ストレージアクセスを実行したときのスループット測定を行った．

次に, 遅延装置を使い高遅延環境およびパケットロス発生環境を作り出し,TCP/IP のみで構成されたネットワークにおいてソケット通信を行っている場合（以降, ソケット通信と呼ぶ）と, パラメータを最適化した iSCSI 通信を行なっている場合において, TCP 輻輳ウィンドウ制御アルゴリズムを変えてスループット測定を行った．

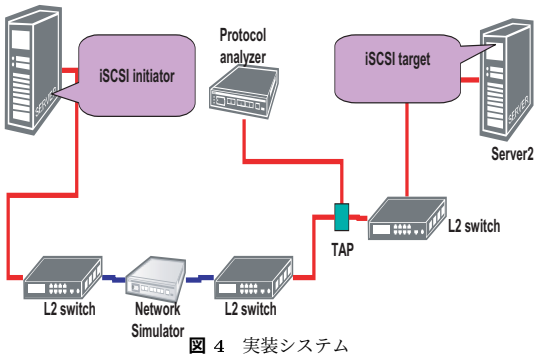


**図 4** 実装システム

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 パラメータ最適化

### 4.1 デフォルト iSCSI とパラメータ最適化 iSCSI スループット測定結果

本実験において,iSCSI のパラメータ設定をライトアクセス時における最適な状態になるように調整した. 変更内容は表 2 の通りである.

表 2　iSCSI パラメータ設定

| | デフォルト | 変更後 |
|---|---|---|
| Target側 | | |
| InitialR2T | Yes | No |
| ImmediateData | No | Yes |
| FirstBurstLength | 65536 | 1048576 |
| MaxBurstLength | 262144 | 1048576 |
| MaxRecvDataSegmentLength | 8192 | 1048576 |
| Initiator側 | | |
| node.conn[0].iscsi.MaxRecvDataSegmentLengt | 131072 | 1048576 |
| node.session.iscsi.FirstBurstLength | 262144 | 1048576 |

遅延装置を使って, 片道遅延時間 0,1,2,4,8,16ms の遅延環境を作り, デフォルトのパラメータを用いた iSCSI とパラメータを最適化した iSCSI のスループットを Bonnie++で測定した. また比較のため, ローカルディスク (SAS) アクセスの性能も共に Bonnie++を使って測定した. この結果を図 5 に示す. ローカルディスクに高速な SAS ディスクをハードウェア RAID0 構成で用いているため, ローカルアクセスが極めて性能が良いことが確認できた. デフォルトのパラメータを用いた iSCSI, パラメータ最適化 iSCSI ともにこれと比較すると性能は低くなるが,iSCSI を用いた場合も低遅延環境においては比較的良好なスループットが出ていることが確認できた. またパラメータ最適化 iSCSI はデフォルトの iSCSI と比べて性能が明らかに良くなっていることが確認できた. しかし, その場合にも高遅延環境においては遅延時間と反比例するようにスループットが劇的に低下している.

図 5 には, 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 に設定したソケット通信のスループットも iSCSI との比較のために示した. 遅延装置を使って, 片道遅延時間 0,1,2,4,8,16ms の遅延環境を作り Iperf で 1000s 間の測定を行った.

その結果, 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ことが確認された. 従って iSCSI 通信の場合も, 高遅延環境においても性能の劣化を極力防ぎ, スループットを保ちたいと考える.

スループットと,TCP の輻輳ウィンドウの値には密接な関係があることが知られている. そこで我々は, スループットの低下原因には TCP の輻輳制御が関連しているのではないかと考えた. 次章以降においては,TCP の輻輳制御アルゴリズムを変えて実験を行う.

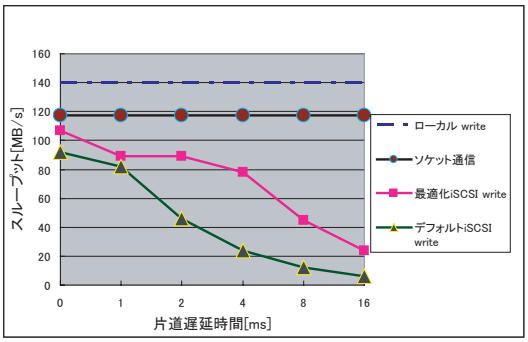


図 5　デフォルト iSCSI, 最適化 iSCSI, ソケット通信のスループット比較

## 5. 異なる TCP 輻輳ウィンドウ制御アルゴリズムを用いたソケット通信におけるスループット測定

TCP 輻輳ウィンドウ制御アルゴリズムとして Reno,BIC,Westwood,H-TCP の 4 種類を用いたときのソケット通信におけるスループットの値をそれぞれ Iperf を使って測定した. 測定は各々5 回ずつ行い, 最高値と最低値を除いた 3 回の平均値であらわす.

### 5.1 高遅延環境におけるスループット測定

測定の際の遅延時間は片道 4ms,8ms,16ms,32ms, 計測時間は 1000s とする. 実験結果を図 6 に示す. なお BIC は図 5 のソケット通信と同じである. また,4 種類のアルゴリズムの立ち上がりの違いを測定するために測定開始から 30s 間のみのスループットの測定も行った. その結果を図 7 に示す.

図 6 からわかるように, 高遅延環境になるにつれてアルゴリズムごとにスループットに違いが表れている. 図 7 でその違いが大きくなっていることから, 違いの原因は主に立ち上がり部分にあると考えられる. そこで, 図 8 から図 11 に片道遅延時間 32ms における各アルゴリズムを用いた場合の測定開始から約 50〜60s 間のスループットの時間変化を示す. これらの図からわかるように, 高遅延環境においては立ち上がりにおいてアルゴリズムごとにスループットが大きく異なるため, 全体としてアルゴリズムごとのスループットの差がでると考えられる.

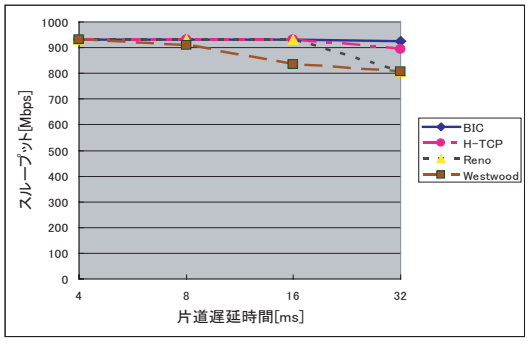


**図 6** 高遅延環境におけるアルゴリズムごとのスループット比較 (測定時間:1000s)

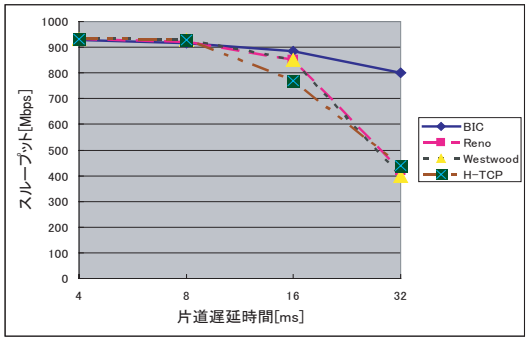


**図 7** 高遅延環境におけるアルゴリズムごとのスループット比較 (測定時間:30s)

### 5.2 パケットロス環境におけるスループット測定

遅延装置を用いパケットロス発生環境を作り出して実験を行った. パケットロス率は約 $1.54 * 10^{-5}$ とした. 測定の際の遅延時間は片道 4ms,8ms,16ms,32ms, 計測時間は 1000s とする. スループット測定結果を図 12 に示す.

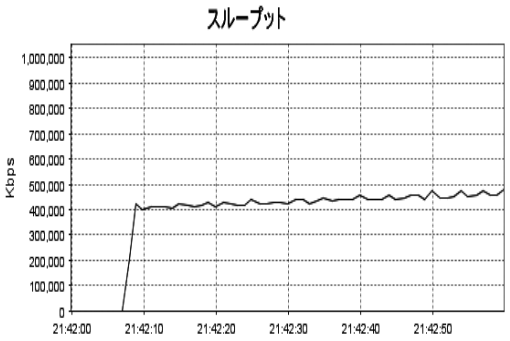


**図 8** 片道遅延時間 32ms の Reno スループット

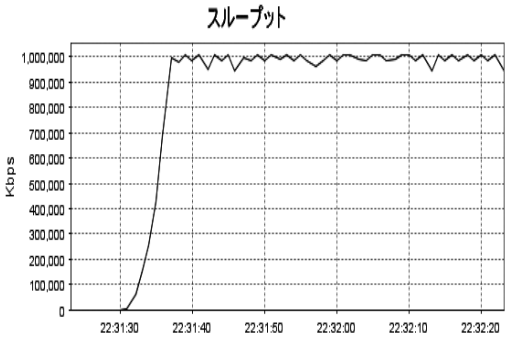


**図 9** 片道遅延時間 32ms の BIC スループット

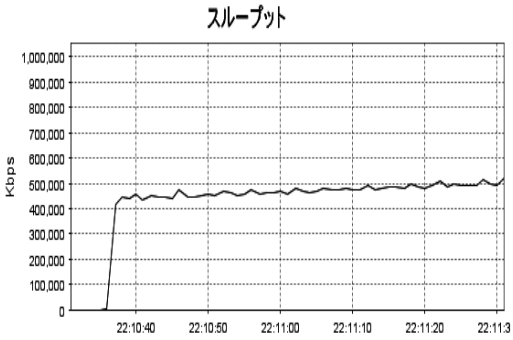


**図 10** 片道遅延時間 32ms の Westwood スループット

低遅延環境においては高いスループットが出ていた BIC などのアルゴリズムも高遅延環境ではスループットが低くなっている. その原因として, 低遅延環境ではパケットを損失してもすぐに回復するが, 高遅延環境だと回復に時間がかかってしまうためパケットロスの影響が大きくなっていると考えられる. 一方パケットロスに強いといわれる Westwood は BIC に比較してスループットが穏やかな低下となっており, 片道遅延時間 32ms では BIC を抜いている.


**図 11** 片道遅延時間 32ms の H-TCP スループット

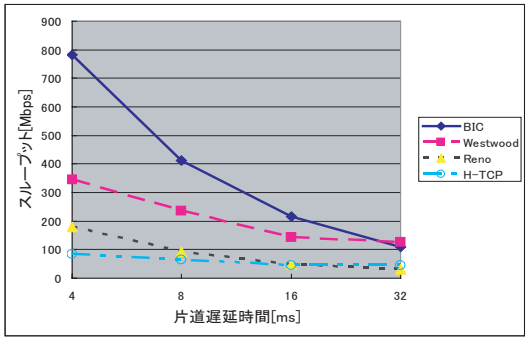


**図 12** パケットロス発生環境におけるアルゴリズムごとのソケット通信スループット比較

## 6. 異なる TCP 輻輳ウィンドウ制御アルゴリズムを用いた iSCSI アクセスにおけるスループット測定

TCP 輻輳ウィンドウ制御アルゴリズムとして Reno,BIC,Westwood,H-TCP の 4 種類を用いたときの iSCSI アクセスにおけるスループットの値をそれぞれ自作のベンチマークツールを使って測定した. アクセスする際のデータのサイズを 16MB とし, 合計で 256MB になるまでアクセスをするように設定した. 測定は各々5 回ずつ行い, 最高値と最低値を除いた 3 回の平均値であらわす.

### 6.1 高遅延環境におけるスループット測定

測定の際の遅延時間は片道 4ms,8ms,16ms,32ms, 計測時間は 1000s とする. ソケット通信で見られたアルゴリズムによるスループットの違い, 立ち上がり方の違いは確認されず, 全てのアルゴリズムで表 3 に示すスループットとなった.

### 6.2 パケットロス環境におけるスループット測定

遅延装置を用いパケットロス率を約 $1.54*10^{-5}$ としたパケットロス発生環境を作り出した. 測定の際の遅延時間は片道 4ms,8ms,16ms,32ms, 計測時間は 1000s とする. パケットロスを発生させることでスループットの大幅な低下は確認できたが, ソケット通信で見られたアルゴリズムによるスループットの違い, 立ち上がり方の違いは確認されず, 全てのアルゴリズムで表 4 に示すスループットとなった.

**表 3** 高遅延環境における iSCSI スループット測定結果

| 4ms | 8ms | 16ms | 32ms |
|---|---|---|---|
| 66MB/s | 36MB/s | 19MB/s | 9.9MB/s |

**表 4** パケットロス環境におけるスループット測定結果

| 4ms | 8ms | 16ms | 32ms |
|---|---|---|---|
| 24MB/s | 13MB/s | 4MB/s | 1.5MB/s |

## 7. 考　　察

本節では,5 章,6 章の実験結果から, ソケット通信時に TCP の輻輳制御アルゴリズムを変えたときの違いと,iSCSI 通信時に TCP の輻輳制御アルゴリズムを変えたときの違いについて考察していく. ソケット通信時にはアルゴリズムごとにスループットの差が生じるが,iSCSI 通信時の高遅延環境, パケットロス環境に両環境においてはアルゴリズムごとの違いが生じていない. その原因としては, 本実験環境において送信可能な iSCSI PDU のブロックサイズが決まってしまっているので性能差が出ないのではないかということが, 表 3 の片道遅延時間 4ms 以降の性能が反比例的に下がっていることから推測できる. すなわちブロックサイズが一定であるため,TCP の輻輳制御アルゴリズムが輻輳ウィンドウの値を変えても, これが使い切られるのに十分な量のパケットが送出されていないと考えられる. また, 表 3 のスループットより片道遅延時間 4ms,8ms,16ms,32ms のときの 1RTT あたりの送信データ量を計算すると, 約 612KB,593KB,562KB,515KB となり,1RTT あたりの iSCSI PDU のブロックサイズが 512KB と推測できる.

そこで,3.3 節で紹介したのプロトコルアナライザを用いてパケットをキャプチャし解析したところ,iSCSI,SCSI ともに最大で 512KB のブロックサイズであった. 高遅延環境, パケットロス環境において TCP の輻輳ウィンドウ制御アルゴリズムごとの性能差を見るためには, ブロックサイズを大きくすることが必要であると考えられる. 今後は, ブロックサイズを大きくし, そのときの性能評価を行う. また, 輻輳ウィンドウの値, エラー

の種類についても考慮していく.

## 8. まとめと今後の課題

本研究では,iSCSI のパラメータを最適に設定したときの高遅延環境における iSCSI ストレージアクセスの変化を評価した. また, TCP の輻輳制御アルゴリズムを変更したときの高遅延環境およびパケットロス環境におけるソケット通信時のスループット測定を行ったうえで,iSCSI ストレージアクセス時のスループットはどのようになるか評価した. その結果,iSCSI のパラメータを最適に設定するとスループットは明らかに良くなるが, 高遅延環境においては性能が劣化してしまうことを確認した. また,TCP の輻輳制御アルゴリズムに異なるものを用いた場合, 高遅延環境およびパケットロス環境において, ソケット通信においては確認することができたアルゴリズムごとのスループット差異は確認されなかった.

この原因については,iSCSI で送信できる PDU のブロックサイズが 512KB と制限されているため輻輳ウィンドウアルゴリズムが変わったとしても輻輳ウィンドウを使い切るのに十分な量のパケットが送出されず, その結果性能差がでない状態であると考えられる.

今後は, ブロックサイズの値がどこで決められているのかを特定してブロックサイズの値を大きく変更した上で, 輻輳ウィンドウの値, エラーの種類を特定し, 輻輳ウィンドウを動的にコントロールして本研究に改良を加えスループットを向上させる手法を検討していく.

## 謝　　辞

本研究は一部, 独立行政法人科学技術復興機構産学共同シーズイノベーション化事業によるものである. また本研究を進めるにあたり, 有用なアドバイスを頂いたソフトバンクテレコム研究所の岡廻隆生氏に深く感謝いたします.

## 参 考 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) 藤原 啓成，若宮 直紀，志賀 賢太:”広域 IP 網を介した iSCSI 通信におけるプロトコルチューニングの一検討”, 第 68 回情報処理学会全国大会, pp.155-156，2006 年 3 月
5) http://linuxgazette.net/135/pfeiffer.html
6) http://acs.lbl.gov/TCP-tuning/linux.html
7) http://www.textuality.com/bonnie/intro.html
8) http://dast.nlanr.net/Projects/Iperf/