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

2006－DBS－140 (II) (41)
2006／7／13

iSCSIアクセス時のVPN環境における TCP輻輳ウィンドウ制御手法の検討

千島 望† 豊田 真智子‡ 山口 実靖* 小口 正人†

†お茶の水女子大学 ‡NTT 情報流通プラットフォーム研究所 *工学院大学

ストレージの管理コスト低減等の目的でSANの導入が進んでおり，IPネットワークを利用したIP-SANとしてiSCSIが期待されている．現在までのところ，SANは主にサーバサイト内のみでしか利用されていない．しかし遠隔バックアップ等を目的として，離れたサイトのサーバとストレージをSANで接続することが望まれている．

そこで本稿では，VPNを利用することにより，ローカルな環境で利用されているiSCSIを広域ネットワークに適用することを考え，その実験システムを実装した．この環境において，TCPパラメータである輻輳ウィンドウの振舞の違いを観察し，それに伴うスループットへの影響を検討した．さらに，VPN環境が輻輳ウィンドウ制御手法によりどのような影響を与えるか評価した．

A Study of Controlling TCP Congestion Window on iSCSI Access through VPN

Nozomi Chishima† Machiko Toyoda‡ Saneyasu Yamaguchi* Masato Oguchi†

†Ochanomizu University ‡NTT Information Sharing Laboratory Group *Kogakuin University

The introduction of SAN is advanced for the purpose of the storage management cost reduction, and iSCSI is expected as IP-SAN which uses IP network. SAN is chiefly used only in the server site currently. However, it is expected to connect a server site with distant storage on SAN for the purpose of remote backup.

Therefore, we have throught iSCSI used in a local environment can be applied to the WAN using VPN, and an experimental system for it has been implemented. In this experimental environment, we have observed a difference of behavior of the congestion window, which is one of the TCP parameters, and examined its influence on throughput Furthermore, we have evaluated how the VPN environment influences the congestion window control technique.

1 はじめに

近年，インターネット技術の進展などにより，ユーザが蓄積し利用するデータ容量が爆発的に増加している．これに伴いストレージの増設，管理コストの増大が問題となっている．そこでストレージネットワークが登場し，その代表的なものとしてFC-SAN(Fibre Channel - 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．一方，SANにIPネットワークを利用した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．

現状において，SANは主にサーバサイト内のみ

でしか使用されていない．しかし遠隔バックアップ等を目的として，離れたサイトのサーバとストレージをSANで接続することが望まれている．そこでVPN(Virtual Private Network)を利用することにより，ローカル環境で使用されているiSCSIを用いて広域ネットワーク上でリモートアクセスを行うことを検討する．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]．また，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ず，最大限の性能が発揮できるようTCPパラメータなどを制御することが求められる．

本稿の構成は以下の通りである．2章で研究背景を述べ，3章でInitiatorとTarget 1対1通信の実装による性能評価結果を，4章で複数台Initiatorの実装による性能評価結果を示し，最後に5章でまとめる．

2 研究背景

2.1 VPN

VPNは，インターネットや通信事業者が持つ公衆ネットワークを使って，拠点間を仮想的に閉じたネットワークで接続する技術である．安価であるという公衆網のメリットを活かしつつ，機密性の低さを暗号化等の別の方法で補うことにより，「実質的な専用網」を実現できるということがVPNの利点である．一方，専用網と異なりネットワークの品質は保証されない場合が多い．

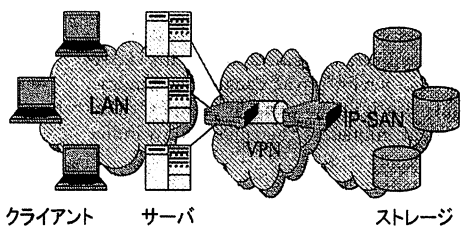

図 1: VPN利用モデル

iSCSIを用いて遠隔バックアップなどを行うには，VPNルータで接続したリモート環境にネットワークストレージを設置し，広域ネットワーク内のVPN越しにアクセスを行うという方法が考えられる(図1)．この場合，VPNルータを通ることによってネットワークの帯域幅が制限され，スループットが著しく低下することが有り得る．iSCSIは通常ギガビットクラス以上の太いネットワーク上で用いられるが，途中にVPNルータの暗号化処理速度などによりスループットが決まる細い回線が挟まることにより，トラフィックとして大いに性質の異なるものになると考えられる．従ってiSCSIが最大限の性能が発揮できるようにTCPパラメータなどを制御することが求められる．

2.2 Linux TCP 実装

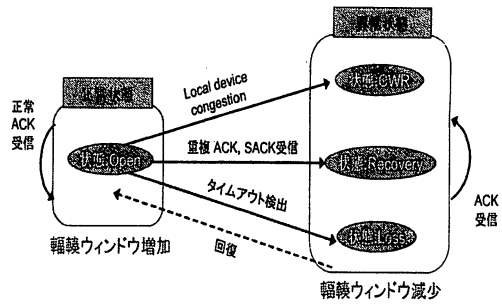

図 2: LinuxTCPの状態遷移

TCPでは，通信能力の制御にウィンドウサイズという概念を用いている．ウィンドウサイズとは，ホストが確認応答パケット(Acknowledgement:ACK)なしに一度に送信できるデータの量で，TCPヘッダにその情報が含まれる．また，データの送信側では輻輳ウィンドウ，受信側では広告ウィンドウという値が決定され，このどちらか小さい方がウィンドウサイズとして用いられる．広告ウィンドウは現在の受信バッファの空き容量を示しており，ACKのヘッダに含まれて送信側に送られる．一方，輻輳ウィンドウは送信側の制御パラメータで，ネットワークの混雑を回避するため送信側が自主的に制限する値である．輻輳制御ではこの輻輳ウィンドウが利用されている．

本実験で用いたLinuxOS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(図2)．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または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．また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，そのウィンドウの値を使い切

ら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．

2.3 既存研究

我々は，これまでにiSCSIストレージアクセスにおいて，輻輳ウィンドウ値を動的にコントロールする手法を提案し，実装と評価を行ってきた[4]．この手法は，まずTargetのOSのカーネルに輻輳ウィンドウモニタ関数を挿入し，これによりモニタした輻輳ウィンドウの変化を観察して，Initiatorにその値を通知する．通知を受けたInitiatorは輻輳ウィンドウの値に基づきブロックサイズを再指定して，シーケンシャルリードアクセスを行うというものである．この手法を適用し輻輳ウィンドウを限界値で一定に保った場合には，高遅延環境において最大28%のスループットの向上が確認された．

また，iSCSIを用いたアプリケーション実行性能とTCPパラメータの相関関係の評価も行った[5]．その結果，広告ウィンドウの値を制限することで，輻輳ウィンドウの値も制限でき，それによって実行性能にも影響が出ることが確認された．

3 1対1通信時の性能評価

本章では，InitiatorとTargetの1対1通信時における輻輳ウィンドウの振舞や性能を評価する．

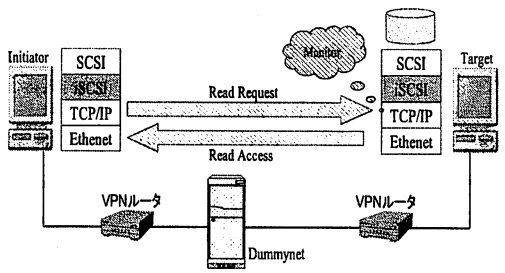

図3: 1対1通信時の実験環境

本実験では，iSCSIストレージアクセスにおいて，VPNを利用した時のTCPパラメータである輻輳ウィンドウと，その輻輳ウィンドウ制御によるストレージアクセスの性能を評価するために，図3に示す実験環境を構築した．iSCSIストレージアクセスを行うInitiatorとストレージを提供するTargetの間にFujitsu Si-R180 VPNルータを2台挟み，さらに，遠距離アクセスを想定して，人工的な遅延装置であるFreeBSD Dummynetを挿入した[6]．InitiatorとTargetには，OSはLinux2.4.18-3，CPUはIntel Xeon 2.4GHz，Main Memoryは512MB DDR SDRAM，NICはIntel Pro/1000XT Server Adapter on PCI-X (64bit,100MHz)，iSCSIはUNH IOL reference implementation ver.3 on iSCSI Draft 18を用いた[7]．この実験環境において，TCP輻輳ウィンドウの影響を見るため，モニタ関数を挿入しカーネルを再コンパイルした．そしてVPN接続環境において，1対1のiSCSIシーケンシャルリードアクセス時のデータを収集し，その性質を調べた．

本実験ではストレージアクセスのみの性能を評価するため，Initiator側ではrawデバイスを使用することにより，キャッシュの影響を排除した．また，iSCSIストレージアクセスにおけるネットワーク性能に焦点を当てて評価を行うため，TargetはUNH実装が提供するメモリモードで動作させ，ディスクアクセスを伴わないようにした．

3.1 輻輳ウィンドウへの影響

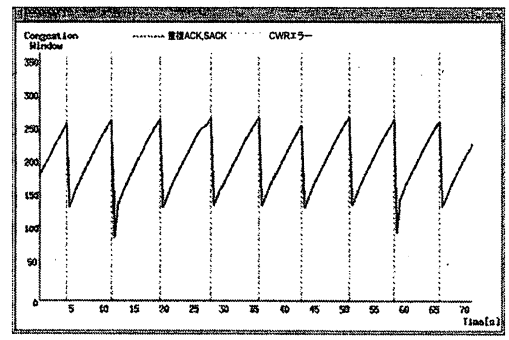

図4: 輻輳ウィンドウの変化（VPNなし）

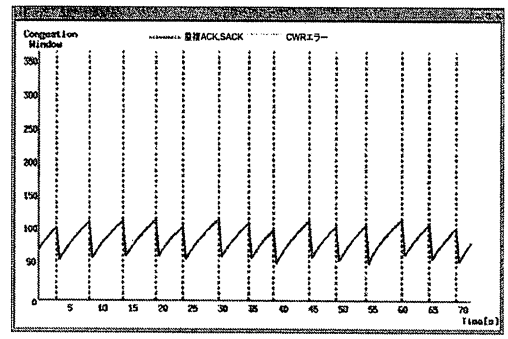

図5: 輻輳ウィンドウの変化（VPNあり）

図4，5はTCP輻輳ウィンドウをモニタした際の時間変化の様子である．図4はVPNルータを挟ま

ず，Dummynet のみを配置し，iSCSI シーケンシャルリードアクセスの通信を行った時の輻輳ウィンドウをモニタした様子である．また，図 4 に示された細かい縦の破線は Local device congestion(CWR) エラーが起こったことを表しており，これは送信側のデバイスドライバのバッファが溢れることによるエラーである．輻輳ウィンドウは約 250 パケットまで増加した後，CWR エラーが検出され輻輳ウィンドウが急激に減少している．

図 5 は VPN ルータ 2 台と Dummynet を挟んだ時の輻輳ウィンドウをモニタした様子である．また，図 5 に示された太い縦の破線は重複 ACK，SACK を受信したことによるエラーが起きたことを示し，これはパケットロスによるものである．輻輳ウィンドウは約 120 パケットまで増加した後，エラーが検出されている．

このように，VPN ルータを挟むことによって，輻輳ウィンドウの上限値は低下し，また，エラーの種類も変化した．

VPN を挟まない時は，途中のネットワークにより通信が制限されることなく，輻輳ウィンドウが高い値まで増加している．そして送信側のデバイスドライバのバッファが限界に達すると CWR エラーが起こり，輻輳ウィンドウが急激に減少，という動作を繰り返して鋸型のグラフになっている．これに対し，VPN ルータを挟んだ時は，送信側のバッファより先に通信経路途中のルータが限界に達するため，輻輳ウィンドウが十分に高い値になる前にパケットロスで低下し，これを繰り返して低い位置での鋸型のグラフとなっている．

3.2 スループットへの影響

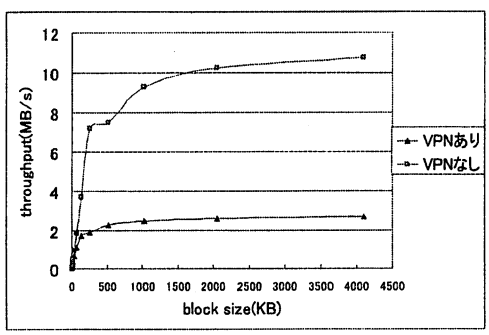

図 6: ブロックサイズ変更時のスループット

図 6 は iSCSI ストレージアクセスにおいてブロックサイズを変化させた時のスループットの値である．

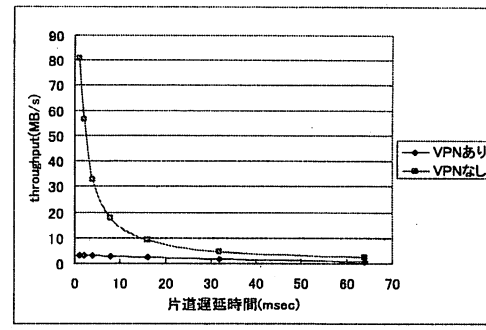

図 7: 片道遅延時間変更時のスループット

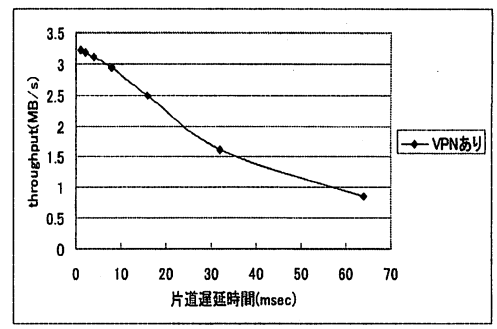

図 8: 片道遅延時間変更時のスループット（拡大図）

この実験において，片道遅延時間は 16ms に設定した．グラフ上側が VPN 接続を用いない直接接続通信の場合，下側が VPN 接続の場合である．グラフからもわかるように，VPN 接続の場合ブロックサイズが 500KB を過ぎた時からスループットにあまり変化はなく，ほぼ 2.5MB/s で落ち着いた．直接接続通信の場合はブロックサイズが 2000KB を過ぎた時からスループットにあまり変化はなく，スループットは 10.5MB/s 位で落ち着いた．また，VPN 接続にすることで，スループットは著しく低下することが確認された．

図 7 は iSCSI ストレージアクセスにおいて片道遅延時間を変えた時のスループットの値である．この実験において，ブロックサイズは 1024KB に設定した．また図 8 は VPN 接続の場合のみの結果を拡大したものである．これらのグラフより，遅延時間を長くするとスループットが著しく低下することがわかる．また，直接接続通信の場合には遅延時間が短い時スループットは急激に減少しており，VPN 接続環境においては急激な減少は見られない．片道遅延時間が 64ms の場合，直接接続通信の場合では約

97% も性能が低下しているのに対し，VPN 接続の場合は約 86% 低下している．いずれの場合においても，遅延時間が長い環境では，性能が著しく低下することがわかる．

3.3 輻輳ウィンドウコントロール手法の適用

3.3.1 実験概要

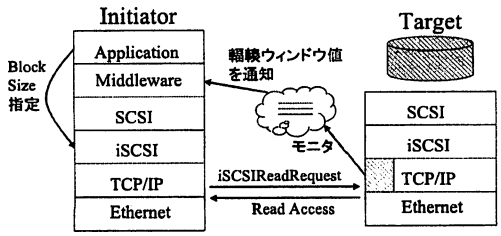

図 9: コントロール手法適用概要

我々は，これまで 1 対 1 接続 iSCSI ストレージアクセス時に，輻輳ウィンドウ値を動的にコントロールする手法を提案した [4]．この手法を，VPN 接続環境において適用する．図 9 は，輻輳ウィンドウコントロール手法の概要図で，iSCSI シーケンシャルリードアクセス時に Target の輻輳ウィンドウをモニタし，CWR エラーが起きた時，Initiator に輻輳ウィンドウ値を通知する．通知を受けた Initiator はミドルウェアでブロックサイズを決定し，アプリケーションがブロックサイズを再指定する．その値を受け Initiator から Target にシーケンシャルリードコマンドを送信し，ストレージアクセスを行う．Target は Initiator に向けて要求されたブロックサイズのデータ転送を実行する．この処理を繰り返すことで，輻輳ウィンドウは CWR エラーが起こらない限界値で一定に保たれる．

3.3.2 実験結果

図 10 に，この手法を用いた場合の実験結果として，片道遅延時間 16ms の環境における輻輳ウィンドウ，ブロックサイズ，スループットの時間変化を示す．コントロール手法適用により，ブロックサイズが変更され，これにより鋸型の変化を繰り返していた輻輳ウィンドウとスループットがやがて一定になる．輻輳ウィンドウが一定値となった後のスループットは，鋸型の変化をする時に比べ大幅に向上している．

次に，VPN 接続環境において同様のコントロール手法を用いた場合の結果として，図 11 に片道遅延時間 16ms の環境における輻輳ウィンドウ，ブロックサイズ，スループットの時間変化を示し，図 12 にこの場合の 100s までの結果を拡大したものを示す．この時，ブロックサイズはずっと変化せず，輻輳ウィンドウは一定値にならずに鋸型の変化を繰り返している．それに伴いスループットも鋸型の変化を繰り返している．また，全体のスループットも 2.5MB/s とコントロール手法適用前と比べてほぼ変化していない．VPN 接続環境では前節で述べた通り，CWR エラーは起こらず，代わりにパケットロスによる輻輳ウィンドウ低下が頻繁に起こっている．そのため CWR エラーの検出を元にブロックサイズを変更させる従来の輻輳ウィンドウコントロール手法は VPN 接続環境には対応しきれていない．従ってパケットロスによる輻輳ウィンドウ低下にも対応できるよう，従来手法を改良することが必要である．

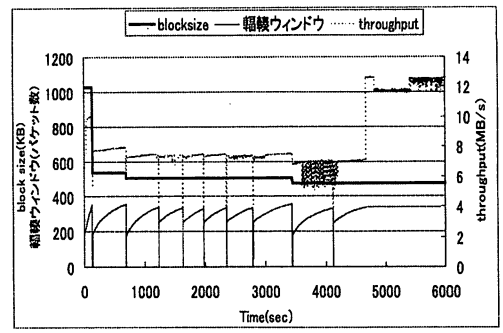

図 10: コントロール手法適用（VPN なし）

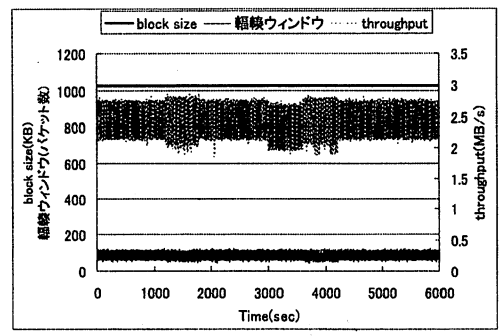

図 11: コントロール手法適用（VPN あり）

4 複数台 Initiator 通信時の性能評価

本章では，Initiator が複数台の場合の輻輳ウィンドウの振舞や性能を評価する．実験環境は図 13 のようになっており、前章での実験環境で

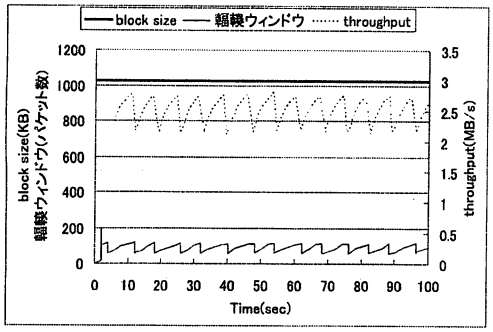

図 12: コントロール手法適用：拡大図（VPN あり）

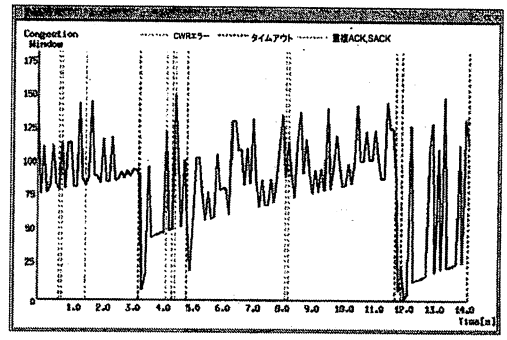

図 14: 輻輳ウィンドウの変化（VPN なし）

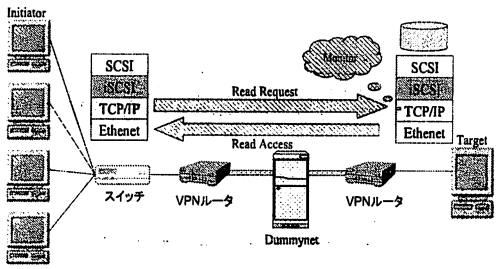

図 13: 複数台 Initiator の実験環境

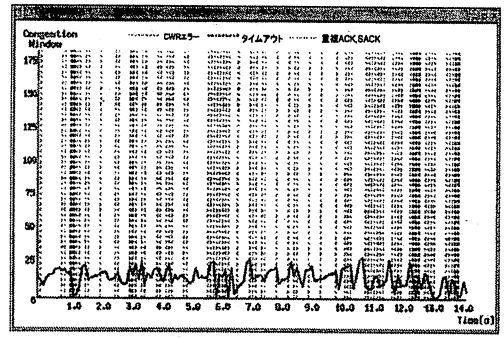

図 15: 輻輳ウィンドウの変化（VPN あり）

Initiator を 4 台に増やしたものである．Initiator は OS は Linux2.4.18-3，CPU は Intel Pentium11 800MHz，Main Memory は 640MB，NIC は Intel PRO/1000MT Server Adapter を用いた．他の実験機器は，1 対 1 通信実験の場合と同じである．

本実験環境では Initiator 4 台をスイッチに接続し，スイッチと VPN ルータを接続している．そして，Target の VPN ルータとの間に人工的な遅延装置である FreeBSD Dummynet を挿入し，片道遅延時間を 8ms に設定した．

この実験環境において，iSCSI シーケンシャルリードアクセス時の TCP 輻輳ウィンドウの振舞を観察した．

4.1 輻輳ウィンドウへの影響

図 14，15 は複数台 Initiator 時の iSCSI シーケンシャルリードアクセス通信を行った場合の輻輳ウィンドウの変化の様子である．また，図中に表れる破線はエラーによるものであり，1 番細い破線が SACK の受信によるエラー，2 番目に細い破線が CWR エラー，1 番太い破線がタイムアウトによるエラーでが起こったことを示している．

図 14 は VPN を利用しない時のグラフであり，輻輳ウィンドウは SACK のエラーを度々起こしながらも，徐々に成長していき，CWR エラーを検出すると，急激に減少している．一方図 15 は VPN を利用した時のグラフで，輻輳ウィンドウは SACK のエラーを頻繁に起こし，連続した SACK のエラーや，タイムアウトによるエラーにより，ほとんど輻輳ウィンドウが成長していないうちに減少している．また，他のエラーよりも少ないが，CWR エラーも時々発生し輻輳ウィンドウを低下させている．

このように，複数台の Initiator にした場合でも，エラーの種類とそれに伴う輻輳ウィンドウの振舞が変化していることがわかる．

4.2 輻輳ウィンドウコントロール手法の適用

我々は，これまでに複数台 Initiator における輻輳ウィンドウコントロール手法も提案し，実装と評価を行ってきた [8]．これは各サーバの輻輳ウィンドウを動的にコントロールして，スループットを均一化する手法で，Initiator が Target から CWR エラー通知を受けると，ブロックサイズをコントロールするものである．

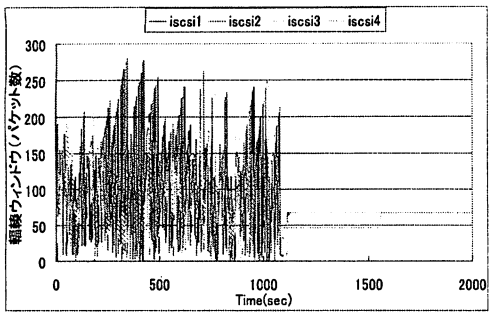

図 16: 輻輳ウィンドウの変化（VPN なし）

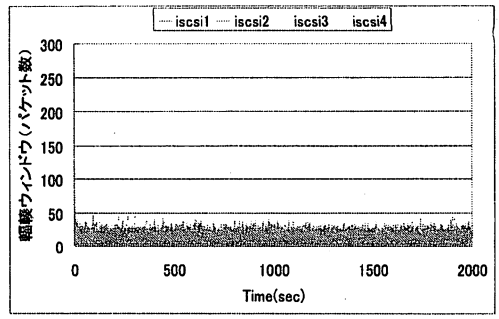

図 18: 輻輳ウィンドウの変化（VPN あり）

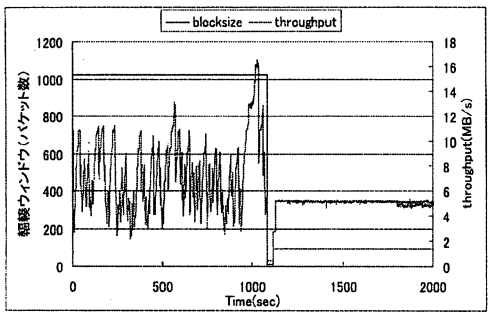

図 17: スループットの変化（VPN なし）

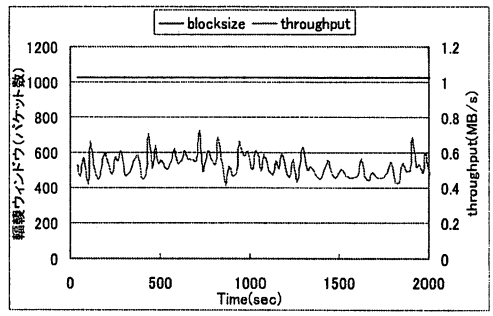

図 19: スループットの変化（VPN あり）

図 16 は VPN を利用しない時の各々の輻輳ウィンドウの変化の様子で，図 17 はその時の Initiator 1 台のスループットとブロックサイズの値である．ブロックサイズが最初の 1024 からコントロールされていく様子が見える．また，それに伴い輻輳ウィンドウやスループットも一定の値をとっていくようになることが確認できた．

図 18 は VPN を利用した時の各々の輻輳ウィンドウの変化の様子で，図 19 はその時の Initiator 1 台のスループットとブロックサイズの値である．この時ブロックサイズはコントロールされず，1024 のままである．そして，輻輳ウィンドウやスループットも一定にはならない．前項で述べたように，輻輳ウィンドウは小さくエラーも頻繁に起こっているのがわかる．

これは，輻輳のエラーの種類が違うことから説明できる．我々が提案したコントロール手法は CWR エラーが起きた時にブロックサイズをコントロールするアルゴリズムになっているので，タイムアウト等のエラーにも対応できるよう改良する必要がある．

5 まとめ

本稿では，VPN 利用時の iSCSI ストレージアクセスによる TCP 輻輳ウィンドウの振舞を観察した．VPN を利用することによって，これまでとはエラーの種類が大きく変わることが確認できた．これは，VPN ルータを通り帯域幅が急激に狭くなることにより，トラフィックの性質が大きく変わるためであると考えられる．その結果，我々が従来提案してきた輻輳ウィンドウコントロール手法は，VPN 利用時にも対応できるよう改良する必要があるということがわかった．

さらに，Initiator を複数にするとエラーが頻繁に起こり，スループットも著しく低下した．VPN 接続環境を用いた場合には，従来の実験では殆ど起こらなかった SACK やタイムアウト等のエラーが頻繁に観察された．

今後は，VPN 利用時に性能が向上するような TCP パラメータのコントロール手法を提案していきたい．また，多対多の環境で複数経路を利用してアクセスできるシステムを構築し実験する予定である．

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] 豊田 真智子，山口 実靖，小口 正人："高遅延ネットワーク環境における iSCSI リードアクセス時の TCP 輻輳ウィンドウ制御手法の性能評価"，SACSIS 2005，pp.443-450，2005 年 5 月

[5] 千島 望，豊田 真智子，山口 実靖，小口 正人:"iSCSI における TCP パラメータとアプリケーション実行性能の相関関係評価" 第 68 回情報処理学会全国大会，pp.131-132，2006 年 3 月

[6] L.Rizzo "dummynet",
http://info.iet.unipi.it/~luigi/ip_dummynet/

[7] InterOperability Lab, Univ,of New Hampshire,
http://www.iol.unh.edu/consortiums/iscsi/

[8] 豊田 真智子，山口 実靖，小口 正人： "複数台 iSCSI Initiator を用いた高遅延ネットワーク環境における TCP 輻輳ウィンドウ制御手法の性能評価"，DEWS2006，7C-o4，2006 年 3 月