2005－DBS－137（II）（77）
2005／7／14

複数台の iSCSI Initiator を用いたストレージアクセス時における TCP フローの解析

豊田　真智子 †　　山口　実靖 ‡　　小口　正人 †

ストレージ管理コストを低下させるために SAN が登場し，その実績は高い評価を得ている．近年ではブロードバンドネットワーク技術の発展により，既存のネットワークインフラを利用して構築可能な IP-SAN が次世代 SAN として注目を集めている．その代表的技術である iSCSI プロトコルを用いることで，専用回線を用いることなく，TCP/IP ネットワーク上で遠隔ストレージへのアクセスが可能となるため，iSCSI への期待は大きい．しかし一方で，登場してから間もない新しい技術であるため，性能に関する課題を残している．

本稿では，iSCSI が利用される想定として複数のサーバからストレージにアクセスする環境を取り上げる．また，より詳細な評価を行うために送信バッファサイズを変更し，各環境において，最大 4 台のサーバから iSCSI を用いたストレージアクセスを行い，TCP パラメータである輻輳ウィンドウとスループットの測定を行う．さらに，実験結果から複数サーバがストレージにアクセスを行った場合に確認される影響についての考察を行う．

Analysis of TCP Flow on Storage Access using Multiple iSCSI Initiator

Machiko Toyoda†　　Saneyasu Yamaguchi‡　　Masato Oguchi†

For the purpose of decreacing a storage management cost, SAN has appeared and achieved extensive results. In recent years, as the broadband networks are widely used, IP-SAN configured with the existing network infrastructure is the most promising technologies as the next generation's SAN. iSCSI protocol, which is one of the most promising technologies in IP-SAN, is expected because it doesn't need a private line and can access a remote storage on a TCP/IP network. However, iSCSI protocol has performance issues since it is a relatively new technology.

In this paper, we suppose an environment in which a storage is accessed from multiple servers. We have changed the volume of a send buffer for evaluating in detail, performed a sequential read access using iSCSI from four servers in maximum, and mesured Congestion Window, which is one of the TCP parameters, and throughput. Moreover, we have analyzed the effect of storage access from multiple servers based on the experimental results.

1　はじめに

ブロードバンドインターネット技術が発達し，誰でも気軽にインターネットに接続することができるようになった今日，サーバなどで管理されるデータ容量の増大が無視できない問題となっている．従来 DAS（Direct Attached Storage）により，サーバに直接接続される形態で管理されてきたストレージであったが，データ共有が非効率であり，管理コストが高価であるといった問題点が指摘されてきた．そのため，サーバとストレージ間をストレージ専用の高速ネットワークによって接続する SAN（Storage Area Network）が登場し，DAS に比べて管理コストが削減されるという理由から注目され，SAN へ

† お茶の水女子大学
Ochanomizu University
‡ 東京大学 生産技術研究所
Institute of Industrial Science,
The University of Tokyo

の移行が進んでいる．

現在広く利用されている FC-SAN では，サーバ，ストレージ間をファイバチャネルで接続することにより構築する．しかし，FC 製品が高価であることや，メーカ間での相互接続が難しいといった理由から，気軽に導入することが難しい．そのため，近年では既存の Ethernet と TCP/IP を用いて構築する IP-SAN が次世代 SAN として期待されている．

IP-SAN のデータ転送プロトコルである iSCSI は，2003 年 2 月に IETF により承認され，最も注目されているプロトコルである [1][2]．iSCSI では，サーバ（Initiator）とストレージ（Target）間のデータ通信を SCSI コマンドによって実現し，SCSI over iSCSI over TCP/IP over Ethrenet という複雑なプロトコルスタックを構成する．そのため，各レイヤ間のデータ受け渡しにおけるメモリコピーなどの影響で，大幅に性能が劣化する [3]．iSCSI は遠隔ストレージへのバックアップやストレージのアウトソーシングなどの利用が想定されるため，iSCSI 使用の際の性能低下を最小限に抑えることは非常に重要となる．

我々はこれまで，Initiator と Target を 1 対 1 で接続した環境において，ストレージアクセスの性能測定を行い，性能向上手法を提案してきた [4]．そこで本稿では，これまでの環境をより発展させた形態として，複数の iSCSI Initiator からストレージ（Target）にアクセスする環境を想定し，その際の輻輳ウィンドウとスループットの振舞について考察を行う．また本稿においては，Initiator と Target 間の最も単純な接続形態を用い，ストレージアクセス時に頻繁に確認される影響を取り上げるため，遅延は考慮していない．

本稿は以下のように構成される．まず 2 章で研究背景を述べる．3 章では複数台の Initiator を用いた性能測定実験を行い，その結果を示す．4 章で実験結果の考察を行い，複数台の Initiator を用いることによる影響を議論する．5 章で関連研究について触れ，最後に 6 章でまとめを述べる．

2 研究背景

2.1 輻輳ウィンドウ

TCP パラメータの 1 つである輻輳ウィンドウは，ネットワークの輻輳制御を目的としてデータ送信側が制限する値であり，確認応答パケットである ACK を受信することなく，一度に送信することができる

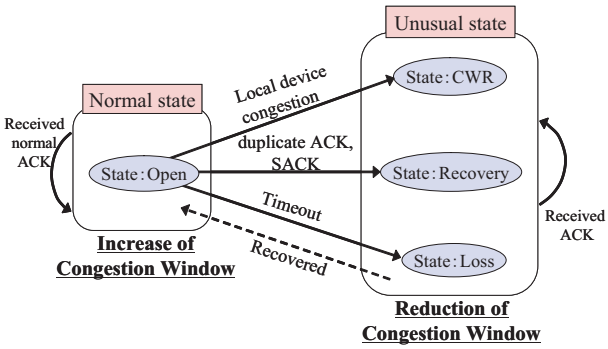

図 1: Linux TCP 実装の状態遷移図

最大のパケット数を意味する．輻輳ウィンドウは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い，ACK を受け取る度に輻輳ウィンドウを 1 ずつ大きくしていく実装となっており，その後輻輳回避アルゴリズムに基づいて制御される．スロースタート，輻輳回避のどちらのフェーズでも輻輳ウィンドウは増加し続ける．しかし，一度輻輳が起こったと判断されると輻輳ウィンドウは急激に減少する．

本実験で用いた Linux OS においては，通信時の状態が正常であれば ACK 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 1）．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，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 [5]．

2.2 iSCSI を用いたストレージアクセスにおける問題点

IP-SAN を構築することにより，より広域エリアにまたがる SAN を実現することが可能となる．また，管理技術者が多い，導入コストが低い，相互接続性が高い，接続距離に制限がないといった点も，IP-SAN の大きなメリットとなっている．

IP-SAN のデータ転送プロトコルであり，従来 DAS で用いられてきた SCSI コマンドと広く普及している TCP/IP ネットワークを用いることができる

iSCSI は，対応製品も増加し，ますます需要が高まるものであると考えられる．具体的には，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，ネットワークにデータ転送を行う．しかし，アプリケーションがデータ転送を要求した場合，そのデータは SCSI 層，iSCSI 層，TCP 層，IP 層，Ethernet 層を通過後，ネットワークを経由し，同様に宛先ホストの各層を通過することになる．そのため，各層での処理やネットワークにおける影響により，iSCSI を用いたストレージアクセスのスループットは劇的に低下する [3].

また我々は，iSCSI を用いたストレージアクセス時に通信において重要な階層となる TCP 層において TCP フローの解析を行った．その結果，TCP パラメータの輻輳ウィンドウとスループットは密接に関連しており，輻輳ウィンドウが増加減少の鋸型の変化を繰り返す場合には，輻輳ウィンドウの変化に伴い，スループットも不安定になることを明らかにした [5]．iSCSI を利用したストレージアクセスの性能向上を検討する場合，輻輳ウィンドウの振舞を考察することは重要な意味を持つと言える．

3 複数台の Initiator を用いた性能測定実験

本節では，複数台のサーバからストレージにアクセスした場合の iSCSI ネットワークの性能を測定するため，最大 4 台の Initiator マシンを用いた実験を行う．Initiator から Target の raw デバイスに対してシーケンシャルリードアクセスを行い，その時の輻輳ウィンドウ，スループットを測定する．各マシンごとの振舞を確認するため，スループットは Target で測定せず，各 Initiator において測定を行った．

また，本実験で用いた NIC（Network Interface Card）は，通信時に TCP 層から受け取ったデータを保持するためのバッファサイズを，ディスクリプタ値を設定することにより変更することが可能となっている．そこで，複数台の Initiator から Target にシーケンシャルリードアクセスを行った場合の性能をより詳細に調べるために，データ送信側である Target の NIC ディスクリプタを変更することで送信バッファサイズを変更して実験を行った．NIC ディスクリプタは 80 から 4096 までの間で変更することができ，デフォルトは 256 に設定されている．そこで，デフォルト値である 256 と，デフォルトよりバッファサイズを小さくした 80，大きくした場合として 4096 の 3 パターンを設定し，各バッファサイズにおける測定を行った．

表 1: 使用計算機：Initiator

CPU	Intel Pentium　800MHz
Main Memory	640MB
OS	Linux2.4.18-3
NIC	Intel PRO/1000MT Server Adapter

表 2: 使用計算機：Target

CPU	Intel Xeon 2.4GHz
Main Memory	512MB
OS	Linux2.4.18-3
NIC	Intel PRO/1000XT Server Adapter

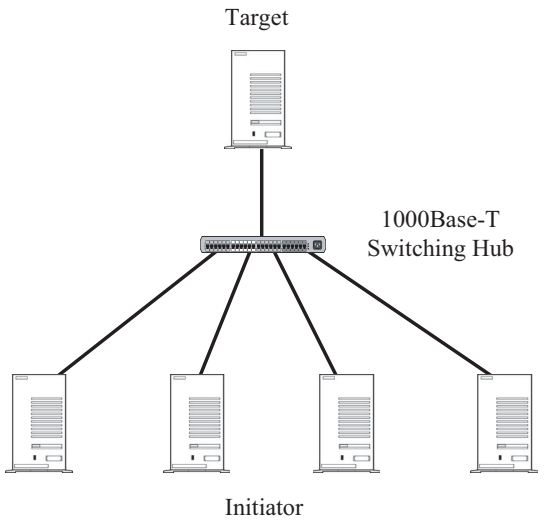

図 2: 実験環境概観図

3.1 実験環境

本実験は以下の環境で行った．Initiator と Target 間は Gigabit Ethernet で接続し，接続途中に 1000Base-T スイッチングハブを挿入し，TCP/IP 接続を確立した．実験環境の概観を図 2 に示す．Initiator，Target はすべて PC 上に構築し，OS には Linux を用いた．iSCSI を利用したストレージアクセスにおけるネットワークの性能を調べるため，Target は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設定した．実験で使用した計算機の環境を表 1，2 に示す．

また，本実験で用いた iSCSI 実装において，Target にはニューハンプシャー大学 InterOperability Lab が提供する UNH IOL reference implementation ver.3 on iSCSI Draft 18[6] を用いた．この UNH 実装では，大きなブロックサイズで read コマンドを発行しても，SCSI 層において要求したブロックサイズより小さなブロックサイズに分割されてしまい，これにより iSCSI ストレージアクセスの性能が大きく低下してしまうことが確認されている [3]．本実験ではこの実装による性能測定への影響を避けるため，Initiator に UNH 実装を用いず，UNH 実装の Initiator と同等の機能を持ち，かつ大きなブロックサイズのデータ転送も行える自作 Initiator を用いて実験を行った．この自作 Initiator は通常のユーザ空間のアプリケーションとして動作し，iSCSI Target と TCP/IP コネクションを確立して iSCSI プロトコルで通信を行うものである．また，ブロックサイズが同じであれば UNH 実装の Initiator とほぼ同じスループットを示すため，自作 Initiator 使用による違いは十分小さいものであり，性能に影響を及ぼさないものであると言える．

3.2 実験結果

Initiator の台数を 1 台から 4 台の間で変更してストレージアクセスを行った実験結果として，各ディスクリプタ値における輻輳ウィンドウの時間変化グラフを図 3，4，5 に示す．

ディスクリプタ値を 80 または 256 に設定した場合は，CWR エラーを検出することにより輻輳が起こったとみなされ，輻輳ウィンドウが急激に減少している．一方ディスクリプタ値を 4096 に設定した場合は，用意されている送信バッファが十分に大きいため，CWR エラーはみられない．また Initiator 数が 1 台の場合は，輻輳ウィンドウは比較的大きな値まで成長し，エラーが生じる間隔も比較的長い．しかし，Target にアクセスする Initiator 数が増加するにつれて輻輳ウィンドウの成長は小さくなり，エラー頻度も高くなることが確認された．

次に各 Initiator のスループットを合計したスループット測定結果を図 6 に，Initiator 数を変化させた場合の各 Initiator の平均スループットを表 3，4，5 に示す．合計スループットは，各環境における Target のスループットであるとみなすことができる．Initiator 台数が 1 台から 2 台に増加した場合，スループットは大きく向上しているが，それ以降はあまり変化がなく，ほぼ飽和状態となっている．また，ディスクリプタを変化させてもスループットに大き

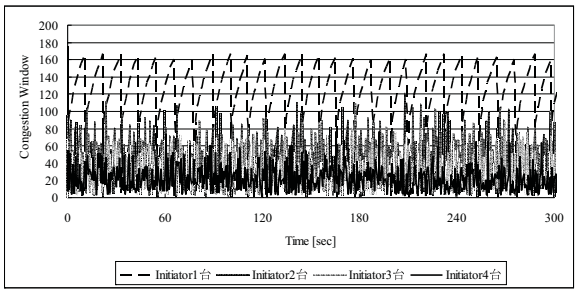

図 3: NIC ディスクリプタ 80 の場合の輻輳ウィンドウの時間変化

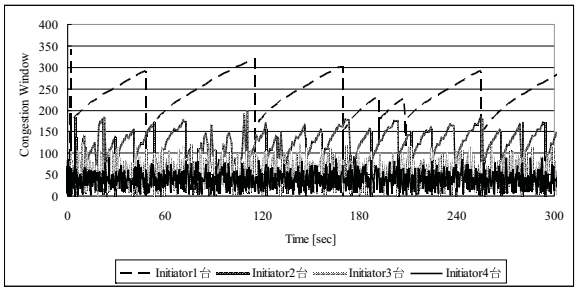

図 4: NIC ディスクリプタ 256（デフォルト）の場合の輻輳ウィンドウの時間変化

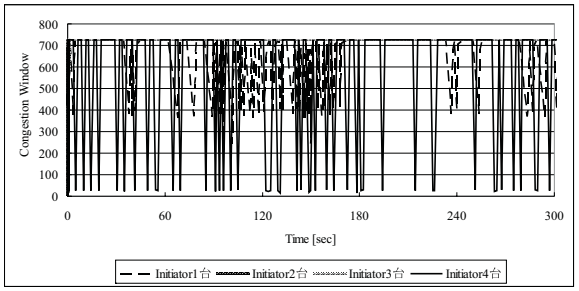

図 5: NIC ディスクリプタ 4096 の場合の輻輳ウィンドウの時間変化

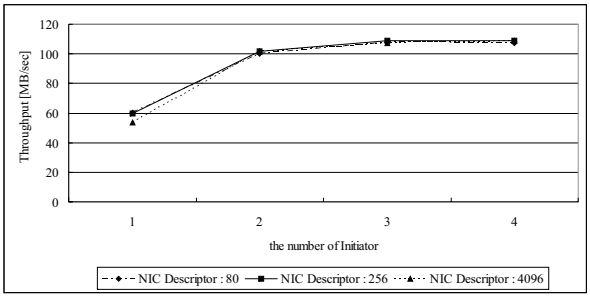

図 6: スループット測定結果

表 3: NIC ディスクリプタ 80 の場合の各 Initiator におけるスループット

Initiator 台数	スループット [MB/sec]			
	Initiator1	Initiator2	Initiator3	Initiator4
1	60.35			
2	49.85	50.00		
3	35.49	35.41	36.69	
4	26.65	26.31	26.59	27.88

表 4: NIC ディスクリプタ 256（デフォルト）の場合の各 Initiator におけるスループット

Initiator 台数	スループット [MB/sec]			
	Initiator1	Initiator2	Initiator3	Initiator4
1	59.45			
2	50.58	51.09		
3	38.09	34.87	35.91	
4	28.38	26.71	26.04	27.78

表 5: NIC ディスクリプタ 4096 の場合の各 Initiator におけるスループット

Initiator 台数	スループット [MB/sec]			
	Initiator1	Initiator2	Initiator3	Initiator4
1	53.89			
2	50.78	50.97		
3	35.66	35.76	35.63	
4	34.97	34.98	34.92	4.29

な変化は見られないということが確認される．我々は遅延が大きな環境においては，スループットが輻輳ウィンドウの成長に依存することを確認している [7]．しかし今回は遅延が存在しない環境において評価することを目的として，Initiator と Target を直結して実験を行ったため，輻輳ウィンドウの大きさにスループットは影響されなかったものと考えられる．

4 考察

本節では，前節の実験結果を CWR エラーが起こるディスクリプタ値 80，256 の場合と CWR エラーが起こらないディスクリプタ値 4096 の場合に分け，その振舞を詳細に考察する．

4.1 NIC ディスクリプタ 80，256 における実験結果の考察

NIC ディスクリプタを 80，256 に設定した場合は，図 3，4 から，Initiator 台数を増加させるに従い，輻輳ウィンドウの上限が大きく低下している様子が確認される．ここで，Initiator 台数が 1 台から 2 台に変化した場合を考える．輻輳ウィンドウは一度に送信できるパケット数を意味しているため，通常は輻輳ウィンドウが減少した場合スループットも減少するはずである．しかし本実験の結果から，輻輳ウィンドウが低下しているのに対し，スループットは大きく向上していることが確認された（図 6）．このことから，Target では同時に存在するコネクション数によって，NIC ディスクリプタにより決められる送信バッファを分割して割り当てていることがわかる．輻輳ウィンドウは各コネクションごとに存在し，2 コネクションの場合はその合計データ量が送信バッファ容量より大きくなった場合に CWR エラーが発生する．そのため，輻輳ウィンドウの上限も Initiator 数が増加するごとに小さくなると言える．

4.2 NIC ディスクリプタ 4096 における実験結果の考察

NIC ディスクリプタを 4096 に設定した場合の結果である図 5 において，Initiator を 1 台用いて行った実験結果から確認される輻輳ウィンドウ低下の原因は，RFC2861 で規定されている処理によるものである [8]．この場合の輻輳ウィンドウ低下は，輻輳が起こったとみなされるどのエラーが原因でもなく，通常輻輳ウィンドウが増加を行うフェーズである正常状態で生じる．送信側が一定時間アイドル状態である場合やアプリケーションによる制限を受けた場合，輻輳ウィンドウの値は現在のネットワーク状態を反映していない無効なものであると判断され，リセット処理が行われるというものである．このリセット処理は Initiator 数が 2 台，3 台の場合は観察されなかった．しかし，リセット処理が確認される通信中，Target から Initiator へ向けてデータは送信され続けており，データ送信の中断などは一切起こっていない．このことから，Target にアクセスする Initiator 数が 1 台の場合，Target にはその処理に余裕があり，何も処理を行わない通信の待ち時間が存在すると考えられる．そのため，TCP 実装がアイドル状態であると判断し，リセット処理を行う．

一方 Initiator の台数が増加すると，1 台の Initiator との通信の待ち時間が別の Initiator へのデータ転送の時間に割り当てられることにより待ち時間は減少し，アイドル状態であると判断される時間に達しないため，輻輳ウィンドウは減少することなく一定値を保ち続ける．この状況は，Initiator 台数が 1 台から 2 台に増加した場合に，各ディスクリプタにおけるスループットが大幅に向上したことと関連するものであると言える．すなわち 1 台の Initiator からのアクセスの場合，Target はデータ処理に比較的余裕があるため，アクセスする Initiator の台数が 2 台に増えると，待ち時間となっていた時間をもう 1 台の Initiator のデータ処理時間に当てる．そのため，スループットは大幅に向上する．しかしこの時点でネットワークもほぼ飽和に近い状態であるため，さらに Initiator の台数を増やしても，ネットワークまたは Target 自身の性能限界のために，さらなる向上はほとんど見られない．

NIC ディスクリプタ 4096 に設定した実験において，一番大きな変化が見られたのは，表 5 における 4 台目の Initiator のスループットである．他のディスクリプタ値における実験の場合も含め，どの実験においても Initiator は同等のスループットであったのに対し，Initiator を 4 台にした実験においてはそのうちの 1 台のみが大きくスループットを低下させている．この時の輻輳ウィンドウは図 5 で確認できる通り，大きく低下している．この輻輳ウィンドウは，リセット処理，輻輳が発生したことによるエラー検出のどちらでもなく，正常な状態で確認されている．このことから，低下した時の小さな輻輳ウィンドウの値は 4 台目の Initiator に割り当てられている輻輳ウィンドウであると考えられる．輻輳ウィンドウが小さいために Target から送信されるデータ量が少なくなり，スループットの向上は見られなかったと言える．

5 関連研究

iSCSI の関連研究としては文献 [9]〜[12]，TCP 輻輳ウィンドウに関連した研究としては文献 [13]〜[15] などが挙げられる．

文献 [9] は iSCSI のソフトウェア実装とハードウェア実装を比較し，CPU 利用率という点を除いてはソフトウェア実装の方が性能が良いという結果を示している．文献 [10] では，iSCSI と NFS の比較を行い，ファイル操作などの一般的な操作におけるパフォーマンスや，ベンチマークを用いた総合的なパフォーマンスを測定している．これらの研究は，iSCSI の性能を知る上では有用な研究であるが，システム内部の振舞について把握し，性能を評価しているものではない．

文献 [11] は，iSCSI の性能が，複数の階層構造を持つことやプロセスの重複により低下するものであると指摘し，“ quanta ”と呼ばれる固定のデータ単位でデータハンドリングを行うこと提案している．iSCSI 性能低下の原因についての着眼点は同じであるが，性能向上のアプローチが異なる．

文献 [12] は，iSCSI Target の内部実装を考慮した iSCSI の性能評価を行っている．性能向上のため，カーネルや iSCSI ソフトウェアの設計に変更を加えるというアプローチであり，既存ソフトウェアには基本的に変更を加えず，ミドルウェアで対応している我々とは手法が異なるものである．また，高遅延環境における評価は行われていない．

文献 [13] は，高速かつ高遅延なネットワークにおける既存 TCP の問題について触れ，パケット送信間隔を調節し，ネットワークへの負荷を抑制するものである．しかし，複数の TCP ストリームを用いることで帯域を有効に利用するというアプローチであるため，iSCSI を対称としている我々とは異なるものである．

文献 [14] では，高遅延環境においては CWR エラーを輻輳とみなし輻輳ウィンドウを減少させると通信性能を低下させるが，並列アプリケーションを実行した場合には CWR エラーを輻輳とみなす方が性能が向上することについて述べ，その解決法を紹介している．また，文献 [15] では，長距離，大容量ネットワークにおける TCP の問題について触れ，新しいトランスポートプロトコルとして提案されている，HSTCP（HighSpeedTCP），Scalable TCP，FAST，SABUL（Simple Avaliable Bandwidth Utilization Library）の輻輳ウィンドウの振舞を紹介し，評価を行っている．これらの文献で紹介されている技術は，輻輳ウィンドウがシステム性能に関連していることについて議論されている点では共通しているが，既存の TCP を改変することで性能向上を目指すものである．従って，広く一般に普及している既存の TCP をそのまま利用した性能向上を目的とする本研究とは異なったものである．

6 まとめ

本稿では，iSCSI ストレージアクセス時に複数の Initiator からシーケンシャルリードアクセスを行い，測定した輻輳ウィンドウとスループットの値からその振舞を考察した．その結果，送信バッファはコネクション数に応じて分割して割り当てられ，輻輳ウィンドウもコネクションごとに設定されることが確認された．そのため，コネクション数が増えるごとに輻輳ウィンドウの上限は減少する．また，アクセスする Initiator 数が 1 台の場合，Target のデータ処理には余裕があり，何も処理を行わない待ち時間が存在する．その結果，TCP 実装がアイドル状態であると判断し，輻輳ウィンドウのリセット処理を行う．しかし Initiator 数が増加すると，待ち時間となっていた時間が別の Initiator へのデータ転送時間に割り当てられることにより待ち時間は減少し，輻輳ウィンドウは一定値を保ち続けることが確認された．また，このような状態であるため，Initiator 数を 1 台から 2 台に増加した場合，スループットは大幅に向上した．しかし，3 台以上ではネットワークが飽和状態となり，スループットの向上はほとんど見られなかった．

今後は今回の実験を iSCSI の使用が想定される高遅延環境に適用し，その性能評価を行いたい．

謝辞

本研究は，一部，文部科学省科学研究費特定領域研究課題番号 13224014 によるものである．

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, February 2004.

[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 ”，先進的計算基盤システムシンポジウム（SACSIS2005），pp.443-450, May 2005.

[5] 豊田真智子，山口実靖，小口正人：“ iSCSI ストレージアクセス時における TCP 輻輳ウィンドウとシステム性能の関連性評価 ”，FIT2004 第 3 回情報科学技術フォーラム，B-004，pp.107-109，September 2004.

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

[7] 豊田真智子，山口実靖，小口正人：“ iSCSI アクセス時の TCP 輻輳ウィンドウ制御のためのアルゴリズムの検討 ”，第 16 回データ工学ワークショップ（DEWS2005），3B-o1，March 2005.

[8] RFC2861 http://www.scit.wlv.ac.uk/rfc/rfc28xx/RFC2861.html

[9] P.Sarkar，S.Uttamchandani，and K.Voruganti：“ Storage over IP: When Does Hardware Support help?, ”*Proc. FAST 2003*，*USENIX Conference on File and Storage Technologies*，pp.231-244，January 2003.

[10] P.Radkov，L.Yin，P.Goyal，P.Sarkar and P.Shenoy：“ Performance Comparison of NFS and iSCSI for IP-Networked Storage，”*Proc. FAST 2002*，*USENIX Conference on File and Storage Technologies*，pp.101-114，March 2004.

[11] P.Gurumohan，S.Narasimhamurthy，J.Hui：“ Quanta Data Storage: A New Storage Paradigm，”*Proc. 12th NASA Goddard Conference on Mass Storage Systems and Technologies*，pp.101-107，April 2004.

[12] 藤田智成，小河原成哲：“ iSCSI ターゲットソフトウェアの解析 ”，先進的計算基盤システムシンポジウム SACSIS2004，pp.335-342，May 2004.

[13] 菅原豊，稲葉真理，平木敬：“ インテリジェント NIC を用いた広帯域ネットワーク向け TCP 通信方式 ”，情報処理学会研究報告 2004-OS-97，SWoPP2004，pp.57-64，August 2004.

[14] 高野了成，石川裕，工藤知宏，松田元彦，児玉祐悦，手塚宏史：“ 並列アプリケーション実行における TCP/IP 通信挙動の解析 ”，IC2003，October 2003.

[15] 熊副和美，堀良彰，鶴正人，尾家祐二：“ JGN を利用した高速トランスポートプロトコルの評価 ”，電子情報通信学会技術研究報告，NS2003-354，IN2003-309，pp.303-308，March 2004.