複数台 Initiator を用いた iSCSI アクセスにおける TCP 輻輳ウィンドウとシステム性能の考察

Analysis of TCP Congestion Window and System Performance on iSCSI Access using Multiple Initiator

豊田 真智子♥　山口 実靖♦
小口 正人♠

Machiko TOYODA　Saneyasu YAMAGUCHI
Masato OGUCHI

ストレージ統合のために登場した SAN において，既存のインフラを利用でき，遠隔ストレージへのアクセスが可能となる iSCSI への期待は大きい．様々な環境での利用が想定される一方で，性能面の弱さも指摘されている．本稿では，iSCSI の利用が想定される環境として，複数台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，輻輳ウィンドウとスループットの測定を行う．そして，測定結果から複数台の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考察する．

Recently, SAN has appeared for the purpose of Storage Consolidation. iSCSI, which can use the existing infrastructure and access remote storages, is expected highly among then. While iSCSI is assumed to be used in various environments, performance issues of iSCSI are pointed out. In this paper, we suppose an environment in which storage is accessed from multiple servers, and measure Congestion Window and throughput in this environment. Moreover, we analyze the effect of storage access from multiple servers based on the experimental result.

1. はじめに

ブロードバンドネットワーク技術の発展により，個人や企業の環境にもインターネットが導入され，それに伴うデータ容量の増加は無視できない重要な問題となっている．ストレージは従来，サーバに直接接続されるDAS（Direct Attached Storage）で管理されてきたが，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．そこで，ストレージを統合することにより，無駄の無い，効率的な管理を行うことを目的として，ストレージ間を高速ネットワークにより接続するSAN（Storege Area Network）が登場した．

SANにはファイバチャネルにより構築するFC-SANと，

♥ 学生会員　お茶の水女子大学大学院 人間文化研究科数理・情報科学専攻　machiko@ogl.is.ocha.ac.jp
♦ 正会員　東京大学 生産技術研究所 sane@tkl.iis.u-tokyo.ac.jp
♠ 正会員　お茶の水女子大学 理学部情報科学科 oguchi@computer.org

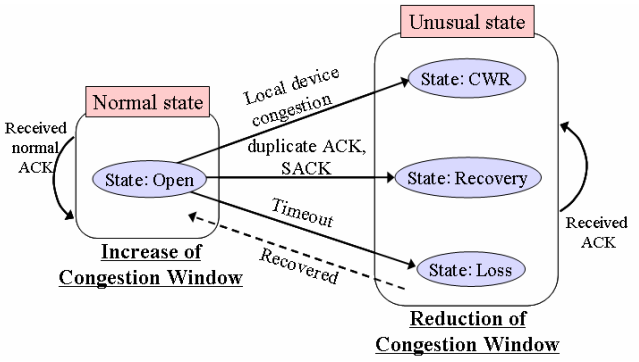

図 1　Linux TCP 実装の状態遷移図
Fig.1 State Transition of Linux TCP

EthernetとTCP/IPを用いて構築するIP-SANがあり，現在までのところ，主にFC-SANが広く利用されている．しかし近年では，既存インフラを利用して手軽に構築可能なIP-SANが次世代SANとしての期待を集めている．その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，将来性が期待されているプロトコルである[1][2]．iSCSIでは，SCSIコマンドをカプセル化することによって，サーバ（Initiator）とストレージ（Target）間のデータアクセスを実現し，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ことがある[3]．

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，複数台の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調べることを目的としているため，遅延は考慮していない．

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最大のパケット数を意味する．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]．

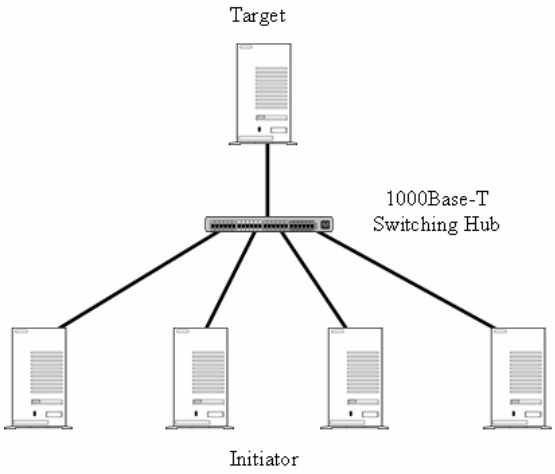

図 2　実験環境概観図
Fig.2 Overview of Experiment Environment

2.2 iSCSI 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，広域エリアにまたがる SAN を実現することが可能となる．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と考えられる．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，各層における処理やネットワークの影響が，iSCSI を用いたストレージアクセスのスループットを劇的に低下させる[3].

また，iSCSIを用いたストレージアクセス時に重要な階層となる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が明らかになった[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 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．個々のマシン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．ディスクリプタ値は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

表 1　使用計算機：Initiator
Table.1 Machine Spec: Initiator

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

表 2　使用計算機：Target
Table.2 Machine Spec: Target

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

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．

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て，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．

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，ディスクリプタ値を変化させてもスループットに大きな変化は見られない．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[4]．しかし今回は遅

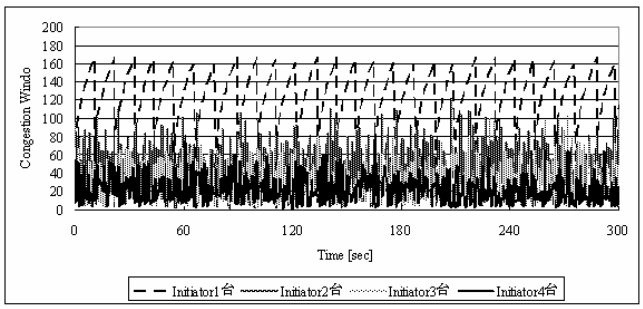

図 3　NIC ディスクリプタ 80 における
輻輳ウィンドウの時間変化
Fig.3 CWND in case of NIC Descriptor 80

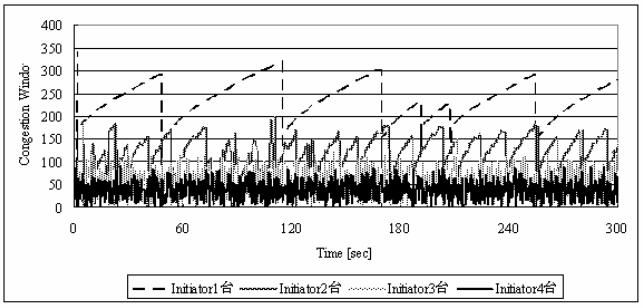

図 4　NIC ディスクリプタ 256（デフォルト）における
輻輳ウィンドウの時間変化
Fig.4 CWND in case of NIC Descriptor 256

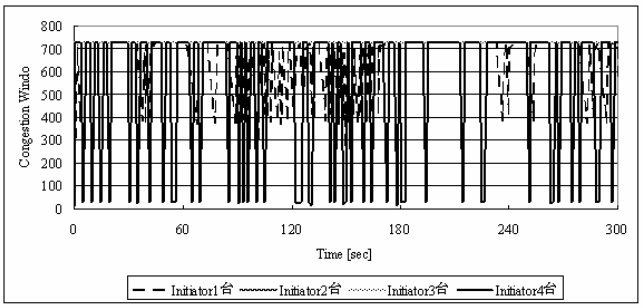

図 5　NIC ディスクリプタ 4096 における
輻輳ウィンドウの時間変化
Fig.5 CWND in case of NIC Descriptor 4096

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

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

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

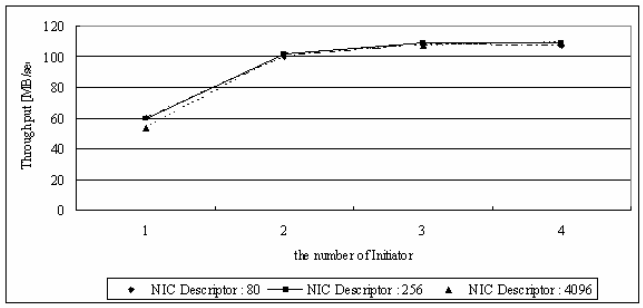

図 6　スループット測定結果
Fig. 6 Result of Throughput

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
Table.3 Throughput in case of NIC Descriptor 80 on each Initiator

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
Table.4 Throughput in case of NIC Descriptor 256 (default) on each Initiator

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
Table.5 Throughput in case of NIC Descriptor 4096 on each Initiator

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

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．輻輳ウィンドウはコネクションごとに存在し，複数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

NIC ディスクリプタを 4096 に設定した実験結果である図 5 において，Initiator を 1 台用いた場合に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 [7]．この場合の輻輳ウィンドウ低下は，通常なら輻輳ウィンドウが増加する正常状態で生じる．これは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ていない．また，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生じていない．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，各ディスクリプタ値の場合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，スループットの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 1 台のみが大きくスループットを低下させた．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. まとめと今後の課題

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から振舞の考察を行った．その結果，送信バッファはコネクション数に応じ，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，その待ち時間が別の Initiator へのデータ転送時間に割り当てられるため，輻輳ウィンドウは一定値を保ち続ける．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が，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 iSCSI の使用が想定される高遅延環境に今回の実験を適用し，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番号 13224014 によるものである．

[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”，先進的計算基盤システムシンポジウム（SACSIS2005）, pp.443-450, 2005年5月.
[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, 2004年9月.
[6] InterOperability Lab Univ, of New Hampshire,
http://www.iol.unh.edu/consortiums/iscsi/
[7] RFC2861
http://www.scit.wlv.ac.uk/rfc/rfc28xx/RFC2861.html

豊田 真智子 Machiko TOYODA

お茶の水女子大学大学院人間文化研究科博士前期課程在学中．2004 年お茶の水女子大学理学部情報科学科卒業．iSCSI を用いたストレージエリアネットワークの研究に従事．日本データベース学会，情報処理学会，電子情報通信学会各学生会員．

山口 実靖 Saneyasu YAMAGUCHI

東京大学生産技術研究所 産学官連携研究員．2002 年東京大学大学院工学系研究科電子情報工学専攻博士課程修了，工学博士．iSCSI を用いたストレージシステムの性能向上の研究に従事．日本データベース学会，情報処理学会，電子情報通信学会各正会員．

小口 正人 Masato OGUCHI

お茶の水女子大学理学部情報科学科助教授．1995 年東京大学大学院工学系研究科電子工学専攻博士課程修了，工学博士．ネットワークコンピューティング・ミドルウェアに関する研究に従事．IEEE，ACM，日本データベース学会，情報処理学会，電子情報通信学会各正会員．