B-027

高遅延環境下における複数台のiSCSI Initiator使用時のTCP輻輳ウィンドウの考察

A Study of TCP Congestion Window using Multiple iSCSI Initiator in Long-Latency Network

豊田 真智子† 山口 実靖‡ 小口 正人†
Machiko Toyoda Saneyasu Yamaguchi Masato Oguchi

1. まえがき

大容量のデータを蓄積するアプリケーションの登場とインターネット技術の発展により，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．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，高い実績をあげている．ファイバチャネルを用いて構築するFC-SANに変わり，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．このIP-SANのプロトコルにおいて，最も注目されているのがiSCSI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]．また，ストレージアクセス時のスループットは，TCPパラメータの輻輳ウィンドウと密接に関連し，輻輳ウィンドウがのこぎり型の増加減少の変化を繰り返す場合には輻輳ウィンドウの変化に従い，スループットも不安定になることが確認されている[4]．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り，輻輳ウィンドウの振舞を考察することは，性能向上の糸口になると考えられる．

そこで本稿では，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．

2. 複数台のInitiatorを用いた性能測定実験

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側ではなく，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，

†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

表1: 使用計算機：Initiator

CPU	Intel Pentium III 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

表3: 使用計算機：Dummynet

CPU	Intel Xeon 2.4GHz
Main Memory	512MB
OS	FreeBSD4.9 - RELEASE
NIC	Intel PRO/1000MT Server Adapter

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大きくした場合として4096の2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．

2.1 実験環境

本実験は以下の環境で行った．InitiatorとTarget間は1000Base-Tスイッチングハブを介してGigabit Ethernetで接続し，遠隔ストレージアクセスを想定した実験を行うため，Targetとハブの接続途中に人工的な遅延装置であるFreeBSD Dummynet[5]を挿入しTCP/IP接続を確立した．実験環境の概観を図1に示す．Initiator，Target，Dummynetはすべて PC上に構築した．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1，2，3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

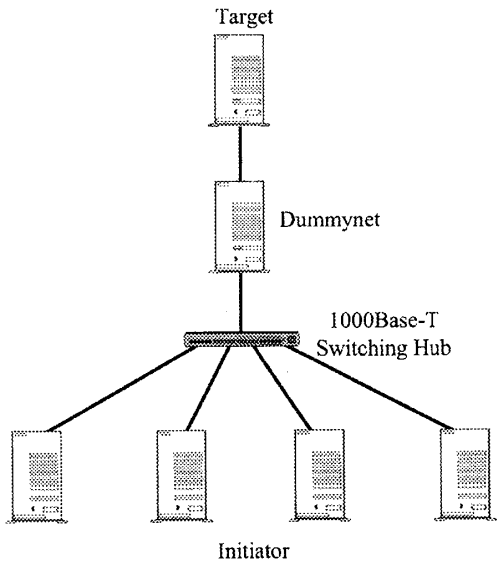

図 1: 実験環境概観図

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．

2.2 実験結果

片道遅延時間 8ms（ラウンドトリップタイム：16ms）に設定し，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 Target で観測した輻輳ウィンドウの時間変化のグラフを図 2，3 に示す．

ディスクリプタ値が 256 の場合は，下位層のデバイスドライババッファが溢れてしまうため（CWR エラー），TCP 実装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，再度増加するというのこぎり型の変化を繰り返している．また，Initiator 数の増加に伴い，輻輳ウィンドウの成長は小さくなり，エラーが増加することが確認される．

一方，ディスクリプタ値が 4096 の場合は一度も輻輳が発生せず，輻輳ウィンドウの上限値は大きい．Linux TCP においては，通信中に一度設定された輻輳ウィンドウは，その値が使い切られない限りは変化しないという特徴を持ち，この時一定値をとる実装となっている．Initiator 台数が 4 台の場合，輻輳ウィンドウが大きく低下しているが，これは輻輳発生によるエラーではなく 4 台目の Initiator に割り当てられている輻輳ウィンドウの値である．これについては次節で詳しく述べる．

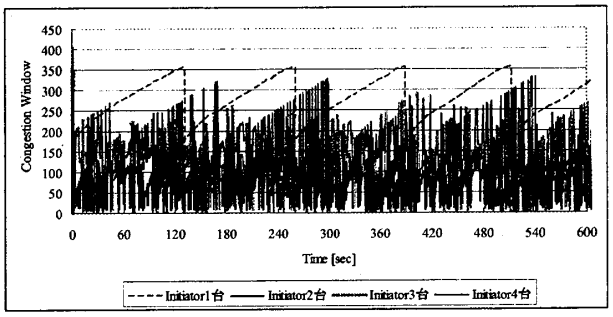

図 2: NIC ディスクリプタ 256（デフォルト）の場合の輻輳ウィンドウの時間変化

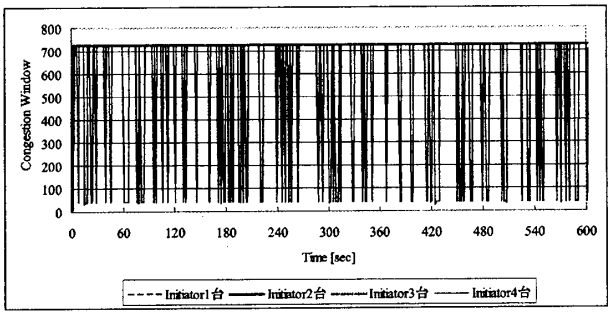

図 3: NIC ディスクリプタ 4096 の場合の輻輳ウィンドウの時間変化

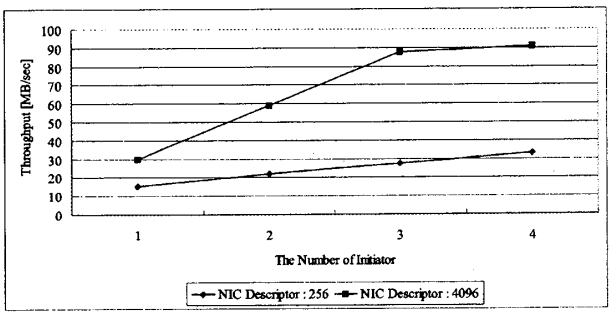

図 4: スループット測定結果

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 4 に，Initiator 台数を変化させた場合の各 Initiator の平均スループットを表 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．ディスクリプタを大きく設定することにより送信バッファ容量が増加し，より多くのデータを一度に送り出すことができるため，各台数におけるスループットはディスクリプタ値が大きい 4096 の場合に大きく向上する．また，ディスクリプタ値 256 の場合は Initiator の台数増加に伴いスループットも増加しているが，ディスクリプタ値 4096 の場合は 3 台以降にあまり変化が無く，ほぼ飽和状態となっていることがわかる．

表4: NICディスクリプタ256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	15.24			
2	10.55	11.47		
3	9.6	9.01	9.11	
4	7.88	7.69	7.23	10.38

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	29.88			
2	29.54	29.69		
3	29.25	29.23	29.21	
4	29.13	29.15	29.13	3.2

3. 考察

NICディスクリプタを256に設定した場合はInitiator台数を増加させるに伴い，輻輳ウィンドウの上限が低下している様子が図2から確認される．ここで，Initiator台数が1台から2台に増加した場合を考える．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，輻輳ウィンドウが減少した場合に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4）．このことから，TargetではNICディスクリプタにより決められる送信バッファを，同時に存在するコネクション数によって分割し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するため，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が，その輻輳ウィンドウの値が各コネクションに適用されて通信が行われるため，Initiator台数が増加すると合計のスループットは向上すると考えられる．

NICディスクリプタを4096に設定した場合，最も大きな変化が見られたのは，表5における4台目のInitiatorのスループットの値である．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各Initiatorのスループットは同等であったが，Initiator台数を4台にした実験においては，そのうち1台のスループットが大きく低下した．また，輻輳ウィンドウの値もInitiator台数が1台から3台の場合にはほぼ一定値をとり，大きな変化はみられないが，Initiator台数を4台にした場合，その値が大きく減少している様子が図3から確認される．この輻輳ウィンドウは，輻輳が検出されたことによる低下ではない．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．

高遅延環境においては応答時間が長くなるため，遅延が少ない場合に比べて性能が大幅に低下する．しかし，送信バッファを大きくすることにより輻輳ウィンドウが大きな値まで成長し，その性能は大幅に向上する．また，同時にTargetにアクセスするInitiator台数を増やすことも，性能向上の大きな要因の一つであることが確認された．

4. まとめと今後の課題

iSCSIストレージアクセスにおいて，高遅延環境下で複数のInitiatorからシーケンシャルリードアクセスを行い，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．輻輳ウィンドウはコネクションごとに設定され，送信バッファもコネクション数に応じて分割されることが確認された．そのため，コネクション数の増加に伴い輻輳ウィンドウの上限は減少する．また高遅延環境においては，送信バッファ容量の増大と，アクセスするInitiator台数の増加によりスループットが大きく向上することがわかった．

今後は，今回の実験から得られた結果を踏まえ，複数台のInitiatorを用いたストレージアクセスにおける性能向上手法を検討していきたい．

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：“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.

[5] L.Rizzo: “dummynet,”
http://info.iet.unipi.it/~luigi/ip_dummynet/

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