ネットワーク装置の帯域制御機構の評価

児玉祐悦[†]　岡崎史裕[†]
工藤知宏[†]　高野了成[†]

主にグリッド環境での利用を目的とした場合の、スイッチやルータの持つ帯域制御機構の挙動について、詳細に評価を行った。その結果、複数の帯域制御機構の違いや、バッファ量による違い等を明らかとし、実際の利用に関する注意点について考察を行った。

Evaluation of rate control functions of network switches and routers

Yuetsu Kodama ,[†] Fumihiro Okazaki ,[†] Tomohiro Kudoh [†]
and Ryousei Takano [†]

We evaluated rate control functions of network switches and routers, considering their usage in grid computing environments, and made clear the difference among the functions in terms of rate control and amount of buffers. In addition, we discussed on some points on which careful consideration is required for use in Grid environments.

1. はじめに

グリッドとは、地理的に離れた計算機や記憶装置、観測装置などの様々な資源を複数連携させることで大規模な科学技術計算を実行する、次世代情報処理基盤である。グリッドでは、計算機リソースの予約/スケジューリングを可能としているが、それだけでは十分ではない。各計算機資源を接続するネットワークの通信性能と安定性が保証されなければ、各計算機の性能を十分に引き出すことはできず、安定した性能を提供することはできないからである。そのため、ネットワークの帯域を資源としてとらえ、計算機資源と同様に予約/スケジューリングすることが必要である。

産業技術総合研究所グリッド研究センターでは、ネットワークの帯域を予約するための共通インタフェースを定めるための研究開発[1)] を他の機関と共同で行っており、分散した計算機群と帯域を必要に応じて柔軟に予約により確保するスケジューリングシステムを開発している。[1)] で行った実証実験では、高品質で高帯域のネットワークを提供するために、拠点間の帯域を保証できる光ネットワークを利用し、ネットワークの制御には IETF で標準化が進められている GMPLS(Generalized Multi-Protocol Label Switching) を用いて、波長やタイムスロット単位での帯域割り当てを用いた。

† 産業技術総合研究所グリッド研究センター
Grid Technology Research Center, AIST

我々は、このような帯域予約を光パス以外の一般的なネットワークにも適用していきたいと考えている。このためには、波長やタイムスロット単位で管理されていない一般的なネットワークで帯域を確保する仕組みが必要である。すでにネットワークルータなどでは QoS のための各種帯域制御機構が数多く研究/実装されており、これらの機能を利用して帯域を分割して割り当てることが考えられる。しかし、これらの機構による帯域確保は、波長やタイムスロット単位での帯域確保と性質が異なる。

帯域の予約を必要とするようなグリッド応用では、1〜数十程度の比較的少数のストリームが、数百 Mbps〜10Gbps 程度の大きな帯域を利用する形態が多いと考えられる。また、TCP/IP による通信が主であると想定され、ひとつのコネクションが大きな帯域を利用するため、通信路の性質が通信性能に大きな影響を与える可能性がある。我々は、片道遅延が 10 ミリ秒程度の範囲では、1 台のクラスタ内で実行していた MPI プログラムを、そのまま変更せずにグリッド環境で動かすことを想定している[2)]。典型的な MPI プログラムでは通信と計算が周期的に繰り返されることが多く、連続的に転送される通常の TCP/IP ストリームとは異なることにも注意が必要である。

本論文では、グリッドのための帯域確保にルータやスイッチの帯域制御機構を用いた場合にどのような影響があるかを明らかにするための初期評価として、制御帯域に近い、あるいはそれを越えるような少数のトラフィックを流したときの、ルータやスイッチの挙動

について詳細に検討を行う。

2. 実験環境

本論文での実験環境について述べる。将来的には実際のグリッドリソースおよびアプリケーションを用いた評価を行いたいと考えているが、初期的な評価として、スイッチ/ルータ単体の評価を行うために、産総研で開発したネットワークテストベッド GtrcNET-1[3),4)] を用いてトラフィックを生成し、観測する。

2.1 ネットワークテストベッド GtrcNET-1

GtrcNET-1 は、4 ポートの GbE モジュール、4 ポートのメモリ、大規模 FPGA を接続したハードウェア装置であり、FPGA の回路を変更することにより、各種機能を GbE のワイヤレートで行うことが可能である。本実験で用いた GtrcNET-1 の主な機能は以下の通りである。

- トラフィック生成。UDP パケットによる CBR (Constant Bit Rate) トラフィックおよび、任意のバースト長を持つバーストトラフィックを、任意の出力帯域で生成する。トラフィック生成は 4 ポートから独立したパラメータで出力することが可能であり、また、2 ポートからのトラフィック生成を同時に開始することができる。VLAN タグ付のトラフィック生成も可能である。
- トラフィック帯域測定。30 秒から 500 マイクロ秒までの任意の間隔で連続的にトラフィック帯域を測定する。この間隔の制限は、USB を介した測定結果取得の遅延による。内部バッファを用いることにより、4 マイクロ秒までの間隔でバッファがフルになるまで (512 サンプル程度) のトラフィック帯域の測定を行える。さらに、パケット内の指定したバイト位置を ID としてストリーム毎のトラフィック帯域を測定することもできる。
- パケットキャプチャ。パケットのすべてのフィールド、あるいはパケットの先頭から 128 バイトを 4 バイト単位で選択的に、最大 16M バイト分キャプチャすることができる。また、受信時に受信時刻を付加しており、これも一緒にキャプチャする。トラフィック生成でパケット内に送信時刻を埋め込んでおり、送受信の時刻を比較することにより遅延を測定することもできる。時刻分解能は 2^{-24} 秒であり、同一筐体で送受信を行った場合はその精度で、異なる筐体間で送受信を行った場合には GPS を用いた時刻同期によりマイクロ秒の精度で測定できる。
- ネットワークエミュレーション。遅延やパケットロス、ネットワーク帯域などをエミュレーションする。遅延は 2^{-24} 秒単位に最大 1 秒まで指定でき、130 ミリ秒までの遅延の場合はワイヤレートのトラフィックをパケットロスなしにエミュレーションできる。パケットロスは、パケット単位およびビットエラーのエミュレーションが可能。ネットワーク帯域のエミュレーションは IFG(Inter Frame Gap) をパケットサイズに応じて調整することにより行う精密ペーシングと、時間あたりの転送量を制御する token bucket 制御が可能。

表 1 評価したスイッチ/ルータの構成

機種	構成
cisco WS-C3750G-24T	IOS C3750 Software Version 12.1(19)EA1c 24 Gigabit Ethernet/IEEE 802.3 interface(s)
cisco GSR 12404	IOS GS Software, Version 12.0(32)S6 Route Processor Card (PRP-2, 1.2GHz, 1GB mem) 4 port Edge Engine3 Gigabit Ethernet
juniper M120	JUNOS 8.2R1.7 RE-A-1000 Standard M120 Routing Engine (1.0GHz, 2GB mem) FEB-M120 M120 Forwarding Engine Board (10Gbps full duplex) IQ2 Ethernet Services Engine(ESE) Type1 PIC(4-port GbE 4:1 oversubscription)

2.2 評価したネットワークスイッチおよびルータ

エッジスイッチとして、Cisco 社の Catalyst 3750 (以下 C3750) を、ルータとして Cisco 社の GSR 12404 (以下 GSR) および Juniper 社の M120 (以下 M120) の評価を行った。用いた機種の詳細は表 1 の通りである。

2.3 帯域制御方式

大別して、帯域制御方式には以下の leaky bucket 方式と token bucket 方式があると考えられる。

leaky bucket 方式は、入力したパケットを FIFO に格納し、指定した帯域で FIFO からパケットを取り出して出力する。FIFO に格納するときに、パケット分の空きがなければパケットが破棄される。例えば帯域を 50%に制御しているときに連続してパケットが到着する場合を考える。FIFO がいっぱいになるまでは、パケットは破棄されずに FIFO に格納される。FIFO がいっぱいになるとパケットが破棄されるが、次のパケットが到着するまでにパケットが出力されることにより FIFO に空きが生じ、到着したパケットは FIFO に格納される。さらにその次に到着するパケットは、FIFO に空きが無いため破棄される。このように 1 パケットおきに格納と破棄が繰り返される。また、パケットは FIFO に格納されて出力されるのを待つため、遅延が発生する。FIFO が大きいと入力帯域の変動に対応できるが、遅延も大きくなる。

token bucket 方式は、パケットを格納するバッファとは別に帯域制御のためにトークンを用いて制御する。簡単な例としては、token をカウンタ C で管理し、一定時間ごとに C に token を追加する。サイズ p のパケットを受信したら、$C > p$ ならパケットを転送し C を p だけデクリメントする。そうでなかった

らパケットを破棄する。token があればパケットは連続的に転送され、token がなくなると、token が追加されるまで転送は停止する。そのため、ある時間間隔でバースト転送が生じることになる。

実際には、帯域制御の前後にキューを持つかどうか、token の更新頻度やチェック方式等により多くのバリエーションが存在する。各スイッチやルータの実装の詳細については情報が公開されていない。

3. 実験結果

3.1 ポート衝突時の挙動

最初に、帯域制御等を設定していない状態で、2 つの入力ポートから同一出力ポートへのトラフィックが流れたときの、出力ポートのトラフィックについて評価を行った。

図 1 は、C3750 の 2 つの入力ポート (in1,in2) へワイヤレートの帯域で同一宛先の UDP トラフィックを入力したときの出力ポートの帯域を示している。IP 長は 1500 バイトである。GtrcNET-1 で各入力ポートからのトラフィックを 100 ミリ秒ごとに測定した。図で in1 というのが入力ポート 1 からのトラフィック、in2 が入力ポート 2 からのトラフィックである。図から分かるとおり、数秒から 10 秒ほどのタイミングで出力トラフィックが入れ替わるが、ある時点では一方のトラフィックのみが出力され、他方のトラフィックはほとんど流れていない。このように短時間の間には非常に不公平な出力制御が行われている。これはワイヤレートの時に限らず、各入力ポートから同じ帯域のトラフィックが入力される場合も同様である。例えば、各入力が 600Mbps の場合、ポート 1 からのトラフィックが 600Mbps, ポート 2 からのトラフィックが 390Mbps という状況が 10 秒程度続いたあとで、ポート 1 とポート 2 からのトラフィック帯域が逆転するという結果となる。一方、各入力ポートのトラフィックが異なる帯域を使用している場合は、状況が異なる。例えば図 2 は、ポート 1 から 600Mbps, ポート 2 から 900Mbps のトラフィックが入力された場合の出力ポートのトラフィックを示している。図に示すように、ポート 1 からのトラフィックの出力が 360Mbps, ポート 2 からのトラフィックの出力が 630Mbps で、100 ミリ秒間隔でみてもほぼ一定の出力が行われている。

GSR では、2 つの入力ポートからワイヤレートの UDP トラフィックを入力した場合は、図 3 に示すように、ほぼ 500Mbps づつのトラフィックが出力された。より詳細にパケットヘッダをキャプチャして計測したところ、各ポートからのトラフィックはパケット単位で交互に出力されていた。また、スイッチでの遅延が徐々に増加し最大 400 ミリ秒程度に増加した。このとき、パケットが 1 パケットおき、あるいは 2 パケット連続してドロップしていた。また、例えばポート 1 から 600Mbps, ポート 2 から 900Mbps のトラフィックが入力された場合は、ポート 1 からのトラフィックの出力が 360Mbps, ポート 2 からのトラフィックの出力が 630Mbps で、100 ミリ秒間隔でみてもほぼ一定の出力が行われている。

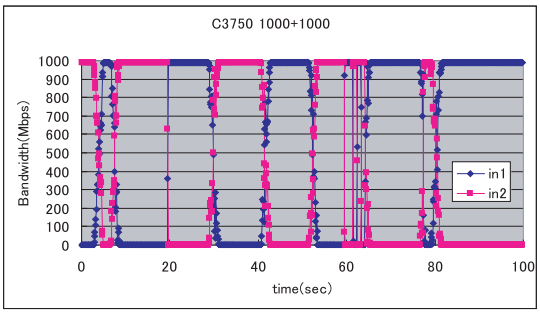

図 1 C3750 1000Mbps+1000Mbps

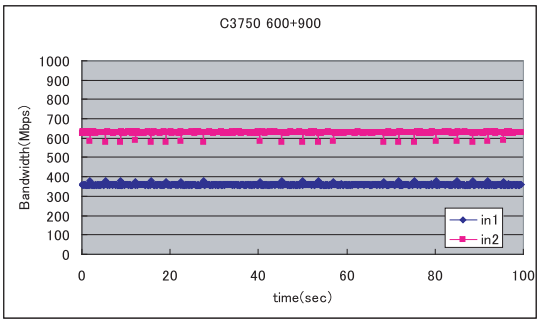

図 2 C3750 600Mbps+900Mbps

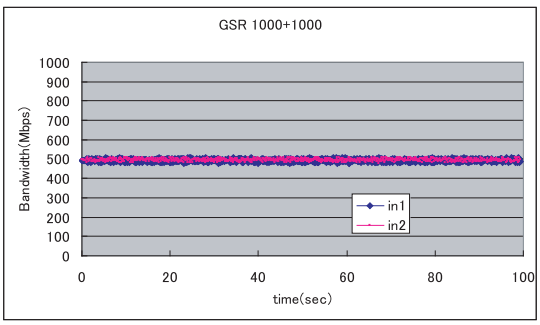

図 3 GSR 1000Mbps+1000Mbps

M120 では、2 つの入力ポートから同一の UDP トラフィックを入力した場合は、GSR と同じく、ほぼ 500Mbps づつのトラフィックが出力された。より詳細にパケットヘッダをキャプチャして計測したところ、各ポートからのトラフィックはパケット単位でほぼ交互に出力されていた。また、スイッチでの遅延が徐々に増加し最大 554 ミリ秒程度に増加した。このとき、各ポート毎にみるとパケットが 1 パケットおき、あるいは 2 パケット連続してドロップしていた。しかし、例えばポート 1 から 600Mbps, ポート 2 から 900Mbps

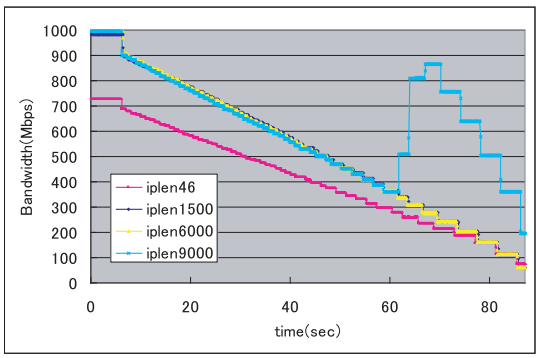

図 4 C3750 出力シェーピングによる帯域制御

のトラフィックが入力された場合は、GSR とは異なり、この場合もほぼ 500Mbps づつのトラフィックが出力された。これは、ここで用いている M120 のインタフェースボードが 4:1 オーバサブスクリプションを行うタイプのボードであり、内部帯域も 1Gbps に制限されているためであると思われる。

3.2 帯域制御時のミクロスコピックな挙動

次に、各スイッチやルータで帯域制御を設定し、ワイヤレートの UDP トラフィックを流した場合の出力トラフィックの挙動について評価を行った。各帯域制御の詳細な挙動を把握するため、パケットをキャプチャして、パケットの送信タイミングやパケットドロップのパターン等について詳細な解析を行った。

C3750 には 2 種類の帯域制御機構がある。1 つは出力シェーピングによるリミット指定、もう一つは入力ポリシングによる帯域指定である。

図 4 は IP 長 46, 1500, 6000, 9000 バイトのワイヤレートトラフィックに対して、出力シェーピングによるリミット指定を 90% から 10%まで 1%づつ減らしていった時の出力帯域を示している。図では約 6 秒付近でリミット指定を開始し、1 秒に 1%づつ減らしている。IP 長 9000 バイト以外は出力帯域が正しく制御されていることが分かる。ただし、リミット指定が 20%以下の時には、4% 刻みでしか制御できない。一方、IP 長 9000 バイトの時にはリミット指定を 40%以下にすると挙動がおかしくなっている。リミット指定を 10%に固定して、IP 長を徐々に変化させたところ、IP 長 6000 を越えたあたりからおかしな出力帯域となっていることが観測されたが、原因は不明である。IP 長が 1500 バイト、リミット指定が 50% の時の出力トラフィックをより詳細に観測すると、先頭でのバースト出力は観測されず、505Mbps に帯域制御されていた。その間、遅延が 22 マイクロ秒から最大 562 マイクロ秒に増加した。遅延が最大となったあとはほぼ 2 パケットに 1 パケットのドロップが観測された。

C3750 で入力ポリシングにより 500Mbps に制限し、IP 長 1500 バイトのワイヤレートトラフィックを流した。出力トラフィックを詳細に観測すると、670 マイクロ秒までワイヤレートでパケットドロップなしに出力され、遅延は 24 マイクロ秒でほぼ一定であった。その後 1 から 3 パケットおきにパケットが転送され、平均 500Mbps に制御されていた。

GSR には入出力それぞれに 2 種類の帯域制御の指定が行える。1 つはシェーピングの指定、もう一つはポリシングでの帯域指定である。

GSR でシェーピングにより 500Mbps に帯域制御し、IP 長 1500 バイトのワイヤレートトラフィックを流した。帯域制御の許容バースト長の指定を省略して、出力パケットを詳細に観測すると、最初 210 ミリ秒はワイヤレートでパケットドロップなしに出力された。その後 1 秒程度は 500 マイクロ秒毎にバースト転送を繰り返して約 506Mbps に帯域制御され、パケットドロップはなかった。その間、遅延は徐々に増加し最大 393 ミリ秒に達すると、20 パケットほど連続してパケットドロップが起きるようになった。許容バースト長として 64K バイトを指定した場合は、最初のバースト長が 840 マイクロ秒に低下した。帯域制御を 250Mbps に変更したり、IP 長を 46 バイト、9000 バイトと変化させたが、500 マイクロ秒毎にバースト転送が行われるのは同じであった。

GSR でポリシングにより 500Mbps に帯域制御し、IP 長 1500 バイトのワイヤレートトラフィックを流した。このとき帯域制御の許容バースト長の指定を省略すると、15.6M バイトが指定された。出力パケットを詳細に観測すると、最初 263 ミリ秒はワイヤレートでパケットドロップなしに出力され、その後 500 マイクロ秒毎にバースト転送を繰り返し、約 506Mbps に帯域制御された。遅延は 48 マイクロ秒で変化なし。帯域を 500Mbps に制御した場合、許容バースト長として指定できる最小値は 256KB であり、この場合最初のワイヤレート転送が 4 ミリ秒となった。また、IP 長が 46 バイトで帯域を 250Mbps に制御した場合、出力は 347Mbps であった。これは、GtrcNET-1 による帯域測定は Ether ヘッダも含んだ値であるのに対し、GSR の帯域指定は IP 部分のみの帯域の指定であるためと思われる。

M120 には、入出力それぞれに 3 種類の帯域制御の指定が行える。1 つめはシェーピングの指定、2 つめはポリシングでの指定、3 つめはトラフィックプロファイルでの指定である。

M120 でシェーピングにより 500Mbps に帯域制御し、IP 長 1500 バイトのワイヤレートトラフィックを流した。許容バースト長の指定はない。出力パケットを詳細に観測すると、最初 100 ミリ秒はワイヤレートでパケットドロップなしに出力され、遅延は 79 マイクロ秒で一定であった。その後 100 マイクロ秒毎にバースト転送を繰り返して、約 510Mbps に帯域制御され、パケットドロップはなかった。その後遅延は

表 2 帯域制御のマイクロスコピックな挙動のまとめ

装置	帯域制御	遅延 (最大)	許容バースト	出力制御
C3750	出力シェーピング	あり (562 マイクロ秒)	なし	1 パケット単位
C3750	入力ポリシング	なし	80KB 固定	1-3 パケット単位
GSR	シェーピング	あり (393 ミリ秒)	指定可	500 マイクロ秒バースト
GSR	ポリシング	なし	指定可	500 マイクロ秒バースト
M120	シェーピング	あり (264 ミリ秒)	6MB 固定	100 マイクロ秒バースト
M120	ポリシング	なし	指定可	1 パケット単位
M120	トラフィックプロファイル	あり (1.9 ミリ秒)	6MB 固定	500 マイクロ秒バースト

徐々に増加し最大 264 ミリ秒に達すると連続してパケットドロップが起きるようになった。

M120 でポリシングにより 500Mbps に帯域制御し、IP 長 1500 バイトのワイヤレートトラフィックを流した。帯域制御の許容バースト長の指定を行うことが必要で、以下では 256K バイトとした。出力パケットを詳細に観測すると、最初 4 ミリ秒はワイヤレートでパケットドロップなしに出力され、その後 1 パケット毎に間隔が空いて帯域が 500Mbps に制御される。遅延の変化はほとんどない。

M120 でトラフィックプロファイルにより 500Mbps に帯域制御し、IP 長 1500 バイトのワイヤレートトラフィックを流した。出力パケットを詳細に観測すると、最初 100 ミリ秒はワイヤレートでパケットドロップなしに出力され、その後 500 マイクロ秒毎にバースト転送を繰り返して約 502Mbps に帯域制御され、パケットドロップはなかった。その後遅延は徐々に増加し最大 1.9 ミリ秒に達すると 20 パケット連続でパケットドロップが起きるようになった。

以上をまとめたものが表 2 である。同じシェーピングやポリシングという機能であっても、出力制御の方式がバースト制御とパケット単位の制御というように違いがみられた。ルータではかなり大きなバッファを利用できることが分かった。一方、大きなバッファを持ち、常に一定の帯域以下に制御したいという我々の要望を満す機能はなかった。また、許容バーストの設定により大きく挙動が変化することが分かった。以下ではもう少しマクロな視点で帯域制御の挙動について検討を行う。

3.3 帯域制御時の間欠トラフィックに対する挙動

前節と同様に帯域制御を行い、間欠トラフィックに対する挙動について評価を行った。本評価は許容バースト長に対する挙動の詳細を評価するため、バースト長を変化させて評価を行った。間欠トラフィックは、前節と同様に GtrcNET-1 により UDP トラフィックとして生成した。IP 長は 1500 バイトとし、バースト長 (ワイヤレート送信パケット数 L) と平均帯域 W を指定する。例えば、L=100, W=500Mbps の時、バースト長は IFG(Inter Frame Gap) やプリアンブルなども含めて 1.23 ミリ秒となり、アイドルの間隔も同じく 1.23 ミリ秒となり、実際の生成バンド幅 (Ether ヘッダを含む) は 494Mbps である。

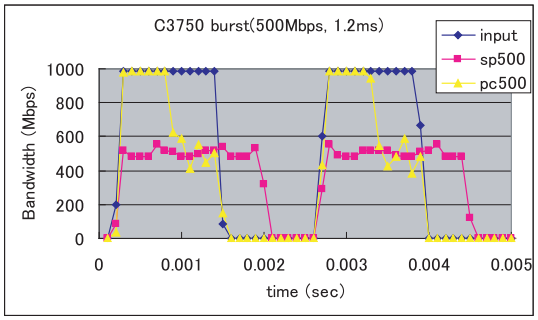

図 5 C3750 間欠トラフィック (L=1.2 ミリ秒, W=500Mbps) 時の出力

C3750 で 出力シェーピングにより 50%に制御し、キューが溢れない程度の間欠トラフィック (L=36 (443 マイクロ秒), W=500Mbps) を入力すると、出力は 505Mbps に帯域制御されたトラフィックが観測された。キューが溢れる程度の間欠トラフィック (L=100 (1.2 ミリ秒), W=500Mbps) を入力すると、1.8 ミリ秒の 505Mbps にペーシングされたトラフィックと 0.6 ミリ秒のアイドル期間が繰り返される (平均 367Mbps)。このときのトラフィックの様子を図 5 に示す。入力トラフィックが input, 出力シェーピングの出力が sp500 であり、100 マイクロ秒毎のバンド幅を示している。

一方、C3750 で入力ポリシングにより 500Mbps に制御し、許容バースト長 (80K バイト) 以下の間欠トラフィック (L=36 (443 マイクロ秒), W=500Mbps) を入力すると、出力は入力と同じ間欠トラフィックが観測された。許容バースト長を越える間欠トラフィック (L=100 (1.2 ミリ秒), W=500Mbps) を入力すると、0.6 ミリ秒のバーストトラフィック、0.6 ミリ秒の 500Mbps に帯域制御されたトラフィック、1.2 ミリ秒のアイドル期間が繰り返される (平均 374Mbps))。このときのトラフィックの様子を図 5 に示す。入力トラフィックが input, 入力ポリシングによる出力が pc500 である。

GSR でシェーピングにより 500Mbps に制御し、間欠トラフィック (L=32000 (394 ミリ秒), W=500Mbps) を入力した場合の様子を、図 6 に示す。図で入力トラフィックが input, 許容バーストを 64K バイトに設定したシェーピングの出力が sp500, 許容バーストを設定しないシェーピングの出力が sp500d であ

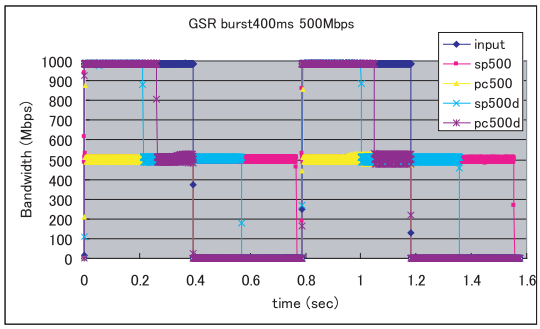

図 6 GSR 間欠トラフィック (L=400 ミリ秒, W=500Mbps) 時の出力

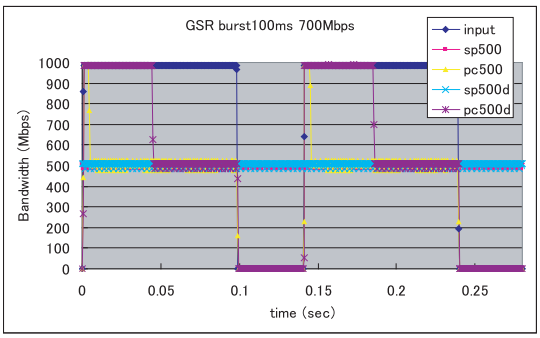

図 7 GSR 間欠トラフィック (L=100 ミリ秒, W=700Mbps) 時の出力

り、500 マイクロ秒毎のバンド幅を示している。sp500 では 1 ミリ秒のバースト転送のあと、入力がアイドルになっても 506Mbps に帯域制御された出力が続く。キューサイズ (96M バイト) は制御できないようである。入力トラフィックの帯域は、正確には IFG 等の分を除いた 493Mbps であり、シェーピングで 506Mbps に帯域制御されているため、出力トラフィックがアイドルになる領域およびその後のバースト転送が生じている。一方、sp500d では 210 ミリ秒のバースト転送のあと 506Mbps に帯域制御された出力が 358 ミリ秒続き、その後アイドルとなり、平均 493Mbps と sp500 と同じである。

GSR でポリシングにより 500Mbps に帯域制御し、上と同じ間欠トラフィックを入力した場合を図 6 に示す。図で入力トラフィックが input, 許容バースト長 256K バイトのポリシングの出力が pc500, 許容バースト長 15.6M バイトのポリシングの出力が pc500d である。pc500 では 4 ミリ秒のバースト転送、入力のある間は 506Mbps に帯域制御され、入力がアイドルになると出力もアイドルになり、平均 255Mbps となる。pc500d では 260 ミリ秒のバースト転送のあと同様に制御され、平均 413Mbps となる。

一方、GSR で 500Mbps に帯域制御し、間欠トラフィック (L=8000 (100 ミリ秒), W=700Mbps) を入力した場合の様子を図 7 に示す。図の凡例は図 6 と同じである。sp500d では、図 6 のようなバーストがみられない。これは入力帯域が制御帯域を越えており、常にバッファがフルの状態になったためと考えられる。また、pc500d ではバーストが 40 ミリ秒ほどに短くなっている。このバースト部の長さはほぼアイドルの長さと同じで、出力トラフィックの平均は 505Mbps となっている。これはアイドルの間に token が増加するが、token bucket がいっぱいになる前に次の転送が始まるため、アイドル時にためた token 分だけバースト転送が起きているためと考えられる。

M120 でシェーピングにより 500Mbps に制御し、間欠トラフィック (L=16000 (200 ミリ秒), W=500Mbps) を入力した場合の様子を、図 8 に示す。

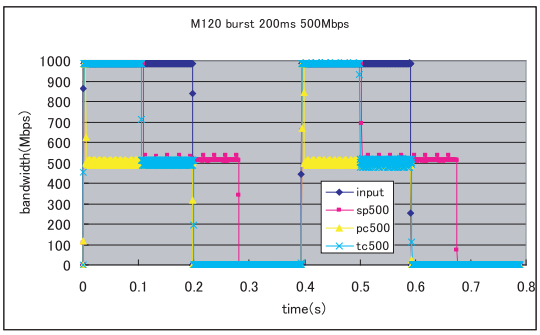

図 8 M120 間欠トラフィック (L=200 ミリ秒, W=500Mbps) 時の出力

図で入力トラフィックが input, シェーピングの出力が sp500 である。入力がアイドルになっても 510Mbps に帯域制御された出力が約 100 ミリ秒間続いたあと、アイドルになり、平均 494Mbps となる。バースト長が 100 ミリ秒の時には、入力と同じバーストトラフィックが出力された。

M120 でポリシングにより 500Mbps に制御し、上と同じ間欠トラフィックを入力した場合の様子を図 8 に示す。図で入力トラフィックが input, ポリシング (許容バースト長 256K バイト) の出力が pc500 である。pc500 では 4 ミリ秒のバースト部のあと、入力のある間は 500Mbps に帯域制御され、入力がアイドルになると出力もアイドルになり、平均 256Mbps となる。

M120 でトラフィックプロファイルにより 500Mbps に帯域制御し、上と同じ間欠トラフィックを入力した場合の様子を、図 8 に示す。図で入力トラフィックが input, トラフィックプロファイルの出力が tc500 である。tc500 では 100 ミリ秒のバースト部のあと、入力のある間は 502Mbps に帯域制御され、入力がアイドルになって 2 ミリ秒後に出力もアイドルになり、平均 383Mbps となる。

また、M120 で 500Mbps に帯域制御し、間欠トラ

表 3 ノードの構成

Node PC	
node	IBM eServer 325
CPU	Opteron/2.0 GHz dual
Ethernet	Broadcom BCM5704
OS	SuSE Enterprise Server 9 (Linux-2.6.17-web100)
sysctl parameters	
net.core.rmem_max	3000000
net.core.wmem_max	3000000
net.ipv4.tcp_rmem	3000000 3000000 3000000
net.ipv4.tcp_wmem	3000000 3000000 3000000
net.ipv4.tcp_no_metrics_save	1
txqueue	10000
initial_ssthresh	1000

フィック (L=8000 (100 ミリ秒), W=700Mbps) を入力した場合、トラフィックプロファイルの出力は、ほぼ GSR の許容バーストが大きい場合のポリシングによる制御と同様に、バースト出力の期間がアイドル期間と同じになり、平均出力帯域がほぼ 500Mbps に制御される。ただし、それ以上に大きなバースト長 (例えば 200 ミリ秒) に対してはバースト部が数ミリ秒になり平均出力は 360Mbps に低下するが、原因は不明である。

間欠トラフィックに対する挙動をまとめると、シェーピングではパケットはいったんバッファに蓄えられるため、パケット破棄を起こさずに出力帯域を制御するのに適した方式である。ただし、遅延が増加する。一方、ポリシングでは許容バーストを越えると直ちにパケットの破棄が起きる。このため、出力を設定帯域に制御するというよりは、入力がある設定帯域を越えないように制御する機能であるといえる。ただし、入力帯域が制御帯域より大きい場合には、入力のアイドル時に対応した分の出力がバースト出力されるため、許容バーストの分だけ出力帯域が制御帯域に近づく。

出力を常にある帯域以下に制御したいという要求に対しては、許容バーストは小さくすることが望ましい。一方、平均出力帯域を制御したいという要求に対しては、許容バーストを大きめに設定することも必要となる。

3.4 帯域制御時の TCP トラフィックに対する挙動

次に、前節と同様に帯域制御を行い、TCP 通信に対する挙動について評価を行った。クラスタとして 32 ノードを 1 台のノンブロック L2 スイッチ (Huawei-3com S5648) で接続し、2 つのクラスタをルータと GtrcNET-1 を通じて接続している。接続はすべて GbE である。GtrcNET-1 では遅延のエミュレート (片道遅延 5 ミリ秒) とストリーム毎のバンド幅を測定している。各ノードの OS やカーネルパラメータ等は表 3 に示す。

各ノードで MPI プログラムを実行し、2 つのクラスタ間で TCP/IP による通信を行う。ここで実行するプログラムは、MPI プログラムを非常に簡略化してモデル化したもので、ノード n からノード n+1 に対して、指定したサイズ L のデータ転送と、指定した時間 W のスリープを、指定した回数 I だけ繰り返す。

表 4 は、2 ノード間の片方向通信で 1GB の転送を行った場合の転送性能を示している。例えば、転送単位が 100MB で転送間隔が 2 秒の時は、100MB 転送する度に 2 秒のスリープを行うが、転送性能として、タグやパケットヘッダを含まないペイロードのデータ量 (8000Mbit) を、スリープの時間を除いた転送にかかった時間の合計で割った値を用いている。表ではルータで帯域制御をせずに 1GB を一度に転送した時の性能に対する相対性能も示している。クラスタ間の通信は本プログラムによる通信のみである。

帯域制御をしていない場合、転送間隔が大きくなるにつれて転送性能が落ちている。これは TCP/IP ではパケット転送を行わない時間に応じて輻輳ウィンドウを減少させるためであり、転送間隔が 0 の時は 2 回目以降の転送は帯域遅延積に対して十分なサイズの輻輳ウィンドウから始まるのに対し、転送間隔が空くと小さな輻輳ウィンドウから始まる。間隔を 3 秒以上にしても変化はなく、2 秒でほぼ輻輳ウインドウが最小になっていると思われる。この輻輳ウィンドウの影響は転送単位が小さいほど顕著に現れる。10MB 単位に転送を行い、転送間隔を 2 秒としたときには、1GB を連続して転送したときに比べて約 1/3 の性能に低下している。転送単位が大きいと、途中で最大性能が出せるくらいに輻輳ウィンドウが大きくなるが、転送単位が小さいと輻輳ウィンドウが十分大きくなる前に転送が終了してしまうためである。

ルータでシェーピングにより 500Mbps に帯域制御を行った場合には、設定通りおよそ 1/2 の転送性能となっている。ただし、許容バーストが大きい場合には設定帯域を越えた転送性能を示している場合がある。これは許容バーストにより間欠トラフィックの先頭で設定帯域を越えた出力が行われるためである。

ルータでポリシングにより 500Mbps に帯域制御を行った場合には、特に許容バーストが大きい場合に転送性能の低下が著しい。例えば、1GB を連続で転送した場合に転送性能が 74Mbps と 1/10 以下に低下している。転送バンド幅を細かな時間間隔で調べたところ、一度転送バンド幅が 1Gbps あたりまで増加したあとに 0 となり数秒間通信が停まっている状況が観測された。通信が停まってしまう原因については現在調査中で理由は不明であるが、許容バーストが大きいため最初はパケット破棄が起らず、輻輳ウィンドウが大きくなったところでパケット破棄が断続的に発生し、多くの再送が一度に発生していることが関係していると考えられる。このような通信が停まってしまうための転送性能低下は、遅延が大きな環境でボトルネックリンクを共有した場合にも観測されており、ポリシン

表 4 GSR 帯域制御を行った場合の 1 ストリームの TCP/IP 通信性能

転送単位	転送間隔	転送性能 (Mbps) (相対性能)				
		帯域制御	シェーピング 500Mbps		ポリシング 500Mbps	
MB	秒	なし	許容バースト 13MB	許容バースト 64KB	許容バースト 16MB	許容バースト 256KB
1000	0	920 (1.00)	484 (0.53)	478 (0.52)	74 (0.08)	318 (0.35)
100	0	901 (0.98)	487 (0.53)	473 (0.51)	88 (0.10)	324 (0.35)
100	1	871 (0.95)	530 (0.58)	468 (0.51)	96 (0.10)	310 (0.34)
100	2	781 (0.85)	509 (0.55)	451 (0.49)	106 (0.12)	306 (0.33)
10	0	746 (0.81)	516 (0.56)	428 (0.47)	274 (0.30)	273 (0.30)
10	1	524 (0.57)	524 (0.57)	372 (0.40)	525 (0.57)	234 (0.25)
10	2	338 (0.37)	339 (0.37)	279 (0.30)	339 (0.37)	204 (0.22)

グに限った話ではない。このような通信性能低下の原因を明らかにし、改善方法を探っていきたいと考えている。

同じポリシングでも許容バーストが小さいときには上記のような著しい性能低下は起きていない。これはパケット破棄が比較的早期に発生するため輻輳ウィンドウが低く抑えられるためだと思われる。この輻輳ウィンドウが小さいままのため、1GB を連続で送ったときでも 318Mbps と設定帯域に比べて低い性能しか出ていない。一方、許容バーストが大きいときには、転送単位が小さく転送間隔が 0 でない場合には、帯域制御なしの場合と同じ性能が出ている。これは許容バーストにより間欠トラフィックの先頭で設定帯域を越えた出力が行われた結果、入力が帯域制御されずにほぼそのまま出力されてしまっているためである。

4. おわりに

本論文では、ルータやスイッチの帯域制御機構について、CBR(Constant Bit Rate) の UDP トラフィックおよび MPI を用いた TCP トラフィックに対する挙動の評価を行った。この評価により、利用時に注意しなければいけない以下のような点について知見を得ることができた。

- スイッチでは、出力ポートでの衝突時に不公平なトラフィックとなる場合がある。
- 帯域制御の細かな挙動は、スイッチ/ルータで異なり、注意が必要である。
- 帯域制御には、パケット破棄がおきにくいシェーピングが有効。ただし、遅延の増加とバッファ長が装置ごとに異なることに注意が必要である。
- ポリシングではパケット破棄がおきやすい。特に許容バーストが大きい場合には、輻輳ウインドウが大きくなってからパケット破棄が起き、通信性能が大幅に低下することがある。これはポリシングに限らず、不連続な大量のパケット破棄が起きたときには同様の現象が観測されているが、原因等詳細は調査中である。

実際のアプリケーションでは、輻輳制御を行う TCP/IP による通信が一般的である。そのため、今後はスイッチやルータの帯域制御がエンドノード間の TCP/IP 通信性能へ与える影響について、より詳細に評価する予定である。また、我々は出力ノードで精密なペーシングを行うソフトウェア PSPacer[5] を開発しており、その効果についても上記の環境で評価を行う予定である。

実際のスイッチ/ルータの実装方式についてはあまり情報が開示されていない。また、帯域制御とキューが不可分に実装されているため、帯域制御方式のみ、あるいはキューサイズのみの影響を十分評価できない。スイッチ/ルータの帯域制御の評価により得られた知見をもとに、スイッチ/ルータの帯域制御の挙動をモデル化し、エミュレータを GtrcNET-1 に実装して詳細に評価して行きたいと考えている。

謝辞

本研究の一部は、文部科学省科学技術振興調整費「グリッド技術による光パス網提供方式の開発」および文部科学省「経済活性化のための重点技術開発プロジェクト」の一環として実施している超高速コンピュータ網形成プロジェクト（NAREGI：National Research Grid Initiative）による。

参 考 文 献

1) http://www.g-lambda.net/
2) M.Matsuda, T.Kudoh, Y.Kodama, R.Takano, and Y.Ishikawa, “TCP Adaptation for MPI on Long-and-Fat Networks,” Proc. of 2005 IEEE International Conference on Cluster Computing (Cluster2005), pp.1-10, 2005
3) Y. Kodama, T. Kudoh, R. Takano, H. Sato, O. Tatebe and S. Sekiguchi, “GNET-1: Gigabit Ethernet Network Testbed,” Proc. of 2004 IEEE International Conference on Cluster Computing (Cluster2004), pp.185-192, 2004.
4) http://www.aist.go.jp/gnet/
5) 高野, 工藤, 児玉, 松田, 岡崎, 石川, ”ギャップパケットを用いたソフトウェアによる精密ペーシング方式,” 情報処理学会論文誌コンピューティングシステム, Vol. 47, No. SIG7(ACS14), pp.194–206, 2006.