ネットワークの利用を効率化するソフトウェア PSPacer

Linux上で動作するオープンソース・ソフトウェアとして無償公開

グリッド研究センターでは、ネットワークの精密な帯域制御とトラフィックの平滑化を行うソフトウェアPSPacerを開発し、公開した。PSPacerは、これまで専用ハードウェアを用いなければ実現できなかった精密な送信帯域制御を、通常のPC上で行える。送信データ量を定められた帯域を超えないように厳密に制御することによって、ネットワークの途中経路の混雑を防ぎ、ネットワークの利用効率を向上させる。遠距離広帯域通信やストリーム配信の効率化のツールとして広い応用が考えられる。また、Linux上で動作するオープンソース・ソフトウェアであるため無償で利用でき、この技術の適用や改良により新しい研究やビジネスへの展開が期待できる。

PSPacer achieves accurate bandwidth control and smoothing under the Linux operating system on a personal computer (PC) without any additional hardware. PSPacer improves the efficiency of long-distance wide-bandwidth communications, and contributes to improving the quality of streaming delivery by surpressing bursty traffic. PSPacer is released as open source, and available at http://www.gridmpi.org/.

ネットワークを効率よく使うには

計算上は、例えば1Gbpsの帯域(最大伝送容量)を持つ通信路(リンク)があれば、20組の50Mbpsの通信を同時に行えるはずである。ところが、実際にはこのように帯域すべてを使い切ることは難しい。これは、計算機で精密に送信帯域を制御することは困難で、この例では個々の通信を常に確実に50Mbps以下にすることが難しく、瞬間的に合計帯域が1Gbpsを超えることがあるためである。

インターネットで広く用いられるIP通信では、通信されるデータはパケットと呼ばれる固まり単位で送受される。これは、ベルトコンベアに荷物が入った箱(パケット)を次々に乗せて送る様子に例えられる（図1)。例えばギガビットイーサネットのリンクの物理帯域である1Gbpsでデータを送っている状態は、ベルトコンベア上の箱と箱の間に隙間がない状態に相当する（実際には、パケットとパケットの間には、規格により規定される最小の隙間が必要なため、最大伝送容量は1Gbpsよりも若干小さい)。図1（a)と(b)では、いずれも平均すれば物理帯域の1/2の500Mbpsの通信が行われている。しかし、図1（a)では均等な間隔でパケットが送られているのに対し、図1（b)ではパケットの間隔に偏りがあり、一度にまとめてパケットが送られている部分がある。このように局所的にまとめて送られる通信をバーストトラフィックと呼ぶ。

ある場所からパケットが送信され、別の場所で受信されるまでの間には、多くのリンクがあり、それらがネットワークスイッチやルータで相互に接続されている。リンクの帯域(最大伝送容量)は場所によって異なることがある。ここで、2つのリンクからのデータが1つのリンク（いずれも最大伝送容量1Gbps)にまとめられる場合を考える。図1（a)のような通信同士がスイッチでまとめられる場合には、図1（c）のように相互の隙間にパケットが納まるため、スイッチが受け取ったパケットは、ほとんど待たされることなく送り出される。ところが図1（b）のようなバーストトラフィック同士がまとめられると、隙間がない部分が重なってパケットをスムーズに送り出すことができなくなってしまう。このような場合には、スイッチはパケットをバッファメモリに蓄えて送り出せ

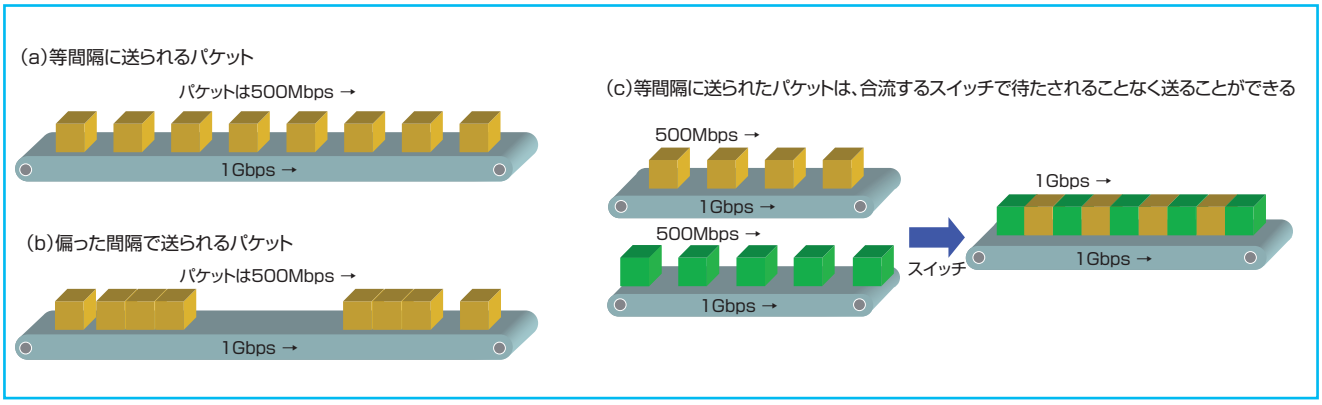

図1　リンク上を伝送されるパケットの概念図

るようになるのを待つが、バッファに蓄えきれなくなると、到着したパケットを破棄してしまう。これをバッファオーバーフローと呼ぶ。バッファオーバーフローによりパケットが破棄されると、例えば映像のストリーム配信では、映像の乱れが生じる。パケットが失われることを防ぐTCPなどのプロトコルが用いられている場合には、再度パケットの伝送を行うため、破棄がなかった場合と比べて余分な通信が発生し、さらに通信量が増えてしまう。バッファメモリには高速なメモリが必要で高価なため、これを少量しか持たないスイッチも多く、図1（a）のように、バーストトラフィックが発生しないように、均等なパケット間隔でデータを送信することが、ネットワークを効率よく利用して安定した通信を実現する鍵となる。

ペーシングとその効果

バーストトラフィックを抑制してバッファオーバーフローを防ぐには、個々のパケットの送信を開始するタイミングを正確に制御する必要がある。このような制御をペーシング（pacing）と呼ぶ。ペーシングは、通信データ量がごく短い時間でも規定の帯域を超えることがないように送信帯域を精密に制御する操作と考えることができる。ペーシングによって、個々の通信が規定の帯域を超えることがないように制御できれば、複数の通信が合わさっても、帯域の変動は最小限に抑えられる。従って、全通信の合計帯域がリンクの最大伝送容量を超えないようにすれば、パケットの破棄は発生せず、効率の良い通信が実現できることになる。

グリッド研究センターでは、ペーシングでバーストトラフィックを抑えることによって効率よく通信が行えることに注目し、当センターで開発した、FPGA（Field Programmable Gate Array）の設定により様々な機能を実現できるハードウェアネットワークテストベッドGtrcNET-1を用いてペーシングを行ってきた。GtrcNET-1は、計算機から送信されたデータをいったんバッファメモリに蓄え、ペーシングしながら送信する。この装置を用いて2003年11月に米国フェニックスで行われた国際会議SC2003において開催された広帯域網利用技術を競うバンド幅チャレンジコンテストに参加し、分散インフラストラクチャ賞を受賞するなど、ペーシングの有効性を示すことができた。しかし、このようなハードウェアがない一般の環境でペーシングを用いることは困難であった。

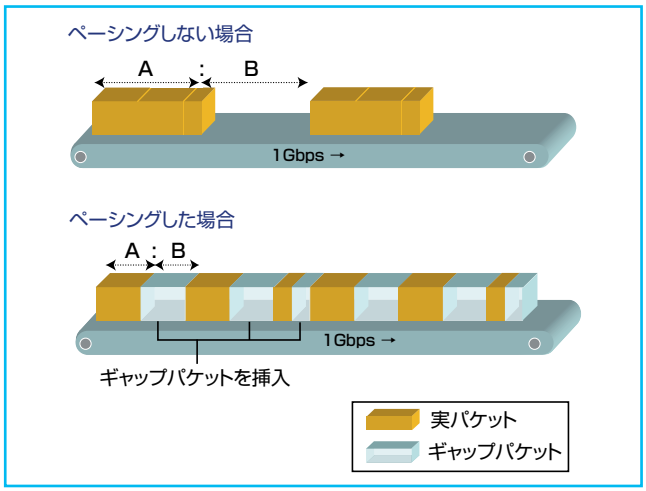

図２　ギャップパケットの挿入によるペーシング

ソフトウェアでペーシングを実現するPSPacer

そこで我々は、ペーシングをさらに広く利用できるようにするために、特別なハードウェアなしにパケットを送信するタイミングを制御し、ペーシングを実現するソフトウェアPSPacerを開発した。

PSPacerの特徴は、実パケットと実パケットの間に、ギャップパケットと呼ぶダミーのパケットを挿入することによって送信間隔を制御する点にある。パケットの送信に要する時間は、パケットのサイズによって正確に決まる。従って、実パケットの間に必要な間隔分の大きさのギャップパケットを送れば、実パケット間の間隔を正確に制御することができる。これは、ベルトコンベアの例で説明すれば、本当に送りたい箱と箱の間にダミーの箱を挿入することによって、箱と箱の間隔を決めることに相当する。間に挿入する、ダミーの箱の大きさを変えれば、間隔を正確に制御することができる。

これまで、ソフトウェアでタイミングを制御するには、タイマ割り込みが用いられてきた。Linuxなどのオペレーティングシステムでは、1ms～10ms程度の間隔のタイマ割り込みが用いられているが、1パケットの送信に要する時間はタイマ割り込みの間隔と比べて短く、ソフトウェアで送信のタイミングを正確に制御することは困難だった。例えば、ギガビットイーサネットで、最小パケットサイズである64バイトのパケットの送信に要する時間は約0.5 μ sである。このように細かい時間単位の制御を行うためにタイマ割り込みの間隔を短くすると、プロセッサの負荷が大きくなってしまい、計算機の計算性能や通信性能に悪影響を及ぼしてしまう。PSPacerでは、ギャップパケットを導入することで、タイマ割り込みなしに正確な送信間隔制御を実現している。

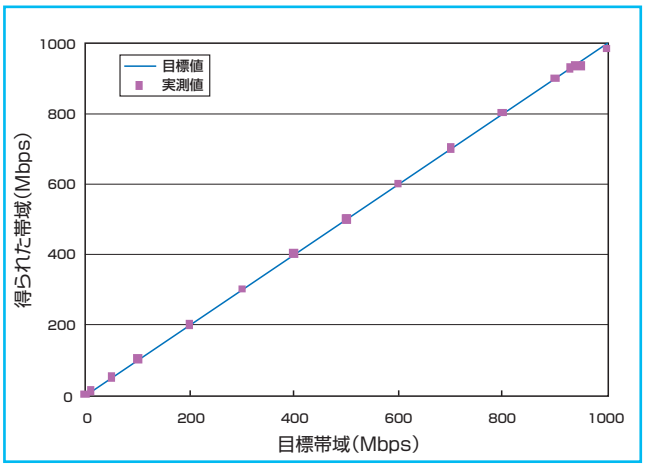

図３　PSPacer による帯域制御の精度

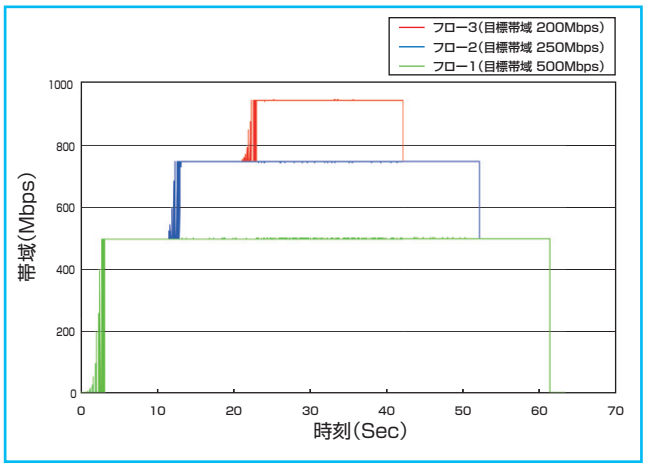

図４　PSPacer により制御された３つのパケットフロー

ギャップパケットは、実際にPCのネットワークインタフェースから送信されるパケットでなくてはならない。その一方、ネットワークをずっと伝搬していってはいけない。つまり、ネットワークインタフェースが接続されているスイッチやルータより先には伝わらないものでなくてはならないのである。そこで、PSPacerでは、ギャップパケットとして、イーサネット規格の一部であるIEEE 802.3xで規定されるPAUSEフレームを用いている。PAUSEフレームは、本来は対向する装置に送信を一定時間停止するよう求めるために用いられる。対向する装置にPAUSE要求が伝わればそれで用済みとなり、その装置（PCが直接接続されているスイッチなど）の入力部で破棄され、それより先に送られることはない。PAUSEフレームには停止時間を指定するフィールドがあり、ギャップパケットでは停止時間を0と指定しているため、対向装置の送信は停止しない。このため、PSPacerを導入すると、PAUSEフレームを用いて対向装置の送信を停止させることはできなくなるが、これ以外の副作用はない。

PSPacerの実装と性能

目標帯域に応じて計算で得られる大きさのギャップパケットを挿入した場合に、実際に観測された送信帯域が図3である。ギャップパケットによって目標どおりの帯域が得られていることがわかる。

PSPacerを用いると、一般的なPCを用いて、IPアドレスとポート番号で識別される100以上のパケットフローをそれぞれ別々にペーシングできる。また、ギガビットイーサネットの場合、パケットフローごとに8Kbps～930Mbpsの範囲で送信帯域を設定できる。図4は、500Mbpsのパケットフローに，250Mbpsと200Mbpsにペーシングしたパケットフローを順に加えていったときの実際の通信帯域の変化を示したグラフである。正確に定められたとおりの通信帯域になっていることがわかる。

PSPacerは、ローダブルカーネルモジュールとして実装されている。そのためカーネルの再構築が不要で、簡単に導入可能であり、デバイスドライバや通信プロトコルにも依存しない。具体的には、Linuxのトラフィック制御機構であるiproute2のQdisc（Queueing Discipline）モジュールとして実装されている。iproute2は、ネットワークトラフィックに対するクラス分け、優先度付け、帯域制御などの機能を提供するためのフレームワークである。Qdiscは、カーネル内のプロトコルスタックとデバイスドライバの間に位置して、ネットワークインタフェースごとの送信キュー（インタフェースキュー）とそのキュー操作アルゴリズムを提供する。

PSPacerは、1パケットフロー、もしくは複数パケットフローをまとめて帯域制御するために、パケットフローをクラスに分けて管理する。パケット送信の要求が発生すると、クラス分けルールにしたがって、いったんクラスごとの送信キューにパケットを保存する。そして、クラスの目標帯域から計算された送信間隔にしたがって、パケットごとの送信タイミング（送信予定時刻）を決定する。送信キューから取り出し送信する際には、現在時刻以前で最も早い送信予定時刻を持つパケットを選択する。現在時刻以前の送信予定時刻を持つパケットが存在しない場合は、ギャップパケットを生成して、実パケットの代わりに送信する。

Linux標準のiproute2を利用しているため、tcコマンドなど、既存のユーティリティやクラス分けフィルタを活用でき、PSPacerと他のQdiscを連携して利用することもできる。PSPacerを導入した後は、IPアドレスやポート番号によるパケットのクラス分けルールを記述し、帯域制御・平滑化を行う通信のインタフェースと帯域を指定する。ネットワークを利用するアプリケーションの変更は不要であり、IPv6にも対応している。

PSPacerの応用

PSPacerを用いると、送信帯域の変動が最小限に抑えられ、バッファオーバーフローの可能性が低減する。このため、ネッ

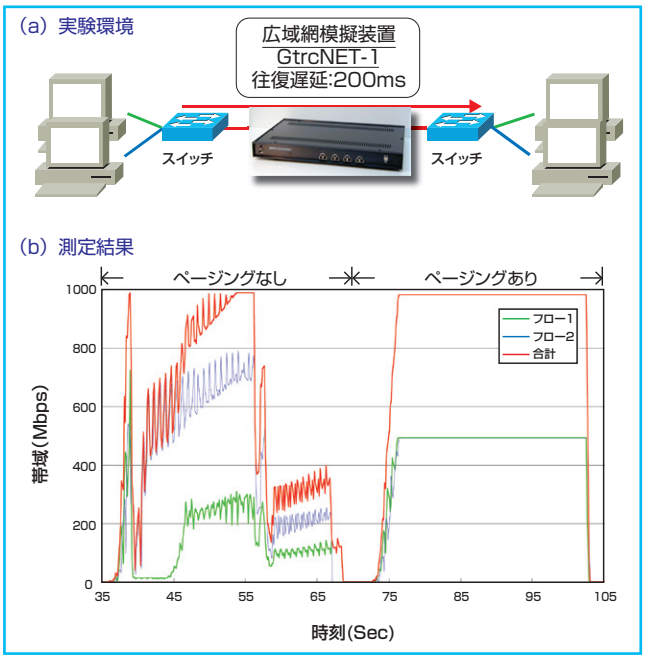

図5 高遅延環境での TCP/IP 通信における PSPacer の効果

トワークの物理帯域の利用効率を飛躍的に増大させることができる。

PSPacerは、以下のような用途で有効である。

(1) 同一リンクを経由する複数の通信がある場合

ストリーム配信などで、ネットワーク上の同一のリンクを用いて複数の通信を行う場合、それぞれの通信のバーストトラフィックが重なって、利用可能な帯域を簡単に超えてしまうことがある。例えば、ストリーム配信業者が、固定帯域のインターネットとの接続リンクを用いて多くのストリームを配信したい場合などがこれにあたる。PSPacerを用いれば個々のストリームに対して帯域を正確に割り当て、ストリーム間の影響をなくして、安定して送信することができる。その結果、配信可能なストリームの数が増え、ネットワークの利用効率が向上する。

(2) 遠距離広帯域のTCP/IP通信

インターネットの通信で広く用いられているTCP/IPでは、往復遅延時間ごとに送信可能なデータの最大量を制御している。このため、遅延が大きい遠距離ネットワークでは、バーストトラフィックが発生しやすくなる。そこでPSPacerを用いれば、日米間のような遅延が非常に大きな環境でもネットワークの帯域を効率よく利用することができる。

図5（a）に示す200msの往復遅延があるネットワーク実験環境で、TCP/IP通信の実験を行った。遅延を模擬する広域網模擬装置として、GtrcNET-1を用いている。送信側の2台の計算機からそれぞれ受信側の別の計算機に向けて送信する。ネットワークは全てギガビットイーサネットである。海底ケーブルを用いた日米間の通信では、150～300ms程度の往復遅延があり、この実験環境はこれを模擬したものである。

図5（b）の「ペーシングなし」の部分は、ペーシングを行わない場合の通信帯域の変化を示している。2組の通信はそれぞれなるべく大きな帯域で送信しようとするため、合計の送信量が中間リンクの帯域を超えてしまい、合流点のスイッチにおいてバッファオーバーフローによるパケット破棄が発生してしまう。TCP/IP通信では、パケットロスが発生すると送信側が送信量を大きく減らすため、全体として通信帯域は安定せず、合計の通信量も不安定でネットワークを効率よく利用できていない。これに対して、図5（b）の「ペーシングあり」の部分では、送信側計算機でそれぞれ490Mbpsにペーシングしている（前述のように、パケット間に必要な最低間隔をとるため、ギガビットイーサネットの実際の通信可能帯域は1Gbpsより若干小さい）。2組の通信は正確に490Mbpsになっており、フロー1とフロー2の線は重なっている。合計での通信帯域も980Mbpsで安定しており、ネットワークを効率よく利用できていることがわかる。

オープンソース・ソフトウェア

PSPacerは、GNU GPL（General Public License）に従ったオープンソース・ソフトウェアとして配布され、http://www.gridmpi.org/からダウンロード可能である。無償で利用可能で、ソースコードが公開されているため、このような技術に関心を持つ人々の協力による更なる改良や、新しい用途への応用とビジネスへの展開が期待できる。

注：本ソフトウェアの開発の一部は、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）により行った。

参考資料

- R. Takano, T. Kudoh, Y. Kodama, M. Matsuda, H. Tezuka, and Y. Ishikawa, “Design and Evaluation of Precise Software Pacing Mechanisms for Fast Long Distance Networks,” PFLDnet 2005, Feb. 2005.
- 高野 了成、工藤 知宏、児玉 祐悦、松田 元彦、手塚 宏史、石川 裕，”ソフトウェアによる精密ペーシング機構の提案と評価，” インターネットコンファレンス 2004, 2004 年 10 月.
- O. Tatebe, H. Ogawa, Y. Kodama, T. Kudoh, S. Sekiguchi, S. Matsuoka, K. Aida, T. Boku, M. Sato, Y. Morita, Y. Kitatsuji, J. Williams, and J. Hicks, “The Second Trans-Pacific Grid Datafarm Testbed and Experiments for SC2003,” IEEE/IPSJ SAINT 2004 Workshops, Jan. 2004.
- Y. Kodama, T. Kudoh, R. Takano, H. Sato, O. Tatebe, and S. Sekiguchi, “GNET-1: Gigabit Ethernet Network Testbed,” IEEE Cluster 2004, Sep.2004.

● 問い合わせ先

独立行政法人 産業技術総合研究所

グリッド研究センター クラスタ技術チーム

研究チーム長　工藤 知宏

主任研究員　児玉 祐悦

高野 了成

E-mail: pspacer@m.aist.go.jp

〒305-8568 茨城県つくば市梅園 1-1-1 中央第二