社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－HPC－116 (30)
2008／8／7

ネットワークトポロジーを考慮した効率的なバンド幅推定手法

長沼 翔[†] 高橋 慧[†] 斎藤秀雄[†]
柴田剛志[†] 田浦健次朗[†] 近山隆[†]

広域分散環境でのデータインテンシブなアプリケーションではデータ転送がボトルネックとなる．しかし広域分散環境ではバンド幅が場所によって大きく異なり，効率的かつ計画的にデータ転送を行う必要がある．この為にはネットワークトポロジーとバンド幅を結びつけた，バンド幅マップの情報が欠かせない．既存の手法では推定に時間がかかるうえ，バンド幅マップのような詳細な情報は得られない．本論文では，ネットワークトポロジーを考慮してバンド幅推定を行うことで高速にバンド幅マップを構築する手法を提案する．

Improving Efficiency of Network Bandwidth Estimation Using Topology Information

Sho Naganuma,[†] Kei Takahashi,[†] Hideo Saito,[†]
Takeshi Shibata,[†] Kenjiro Taura[†] and Takashi Chikayama[†]

Data transfer is a bottleneck to excute data intensive applications in distributed environments. The bandwidth of each link there varies from place to place, however, it is necessary to perform the data transfer efficiently and systematically. To do this, the Bandwidth Map is required, which is information of the network topology combined with the values of bandwidth. Some existing methods to measure bandwidth take a long time to be accomplished, and what is more, we cannot know details such as the Bandwidth Map. In this paper, we propose an efficient and accurate method of measuring and building the Bandwidth Map taking a network topology into account.

1. はじめに

地理的に離れたクラスタをネットワーク接続した広域分散環境が注目されている．これによって安価でありながら莫大な計算機資源を得ることができる．広域分散環境上で並列分散処理を行うことによって，自然言語処理や遺伝子解析などの，これまで望めなかった大規模な処理が可能になっている．このような分散環境において，各ノード間を結ぶリンクはバンド幅の広いリンクと狭いリンクが混在している．並列分散アプリケーションを実行する際にこれらを考慮して通信を行わなければ実行効率は思うようにあがらない．例えばWeb上のドキュメントの言語処理や遺伝子解析はデータインテンシブなアプリケーションと呼ばれ，一般的に計算の量が多く，また処理対象のデータサイズも非常に大きい．これらのアプリケーションは並列分散処理が不可欠である．しかしこれらは非常に大きなデータに対して処理を行うので，処理対象データの移動やコピーを行うなどのデータ転送時間が占める割合が，全体の実行時間に対して，大きくなる．従って各リンクのバンド幅をよく考慮してデータ転送スケジュールを組み，通信相手を選択して効率よくデータ転送やジョブ分配を行うことが重要になる．このように，広域分散環境上で並列分散計算を実行する際には，計算自体のアルゴリズム等の他に，ネットワークの通信性能を考慮したデータ転送・ジョブ分配のスケジューリングが重要である．そしてそのスケジューリングの為には，何らかの手法で，トポロジー上の全てのリンクにバンド幅情報を関連付けた，バンド幅マップ (図1) を知る必要がある． バンド幅マップの情報を活用してデータ転送最適化を実現した例として，高橋らはヘテロなネットワーク環境に対しても安定して効率を高めるできるブロードキャストアルゴリズムを提案している[3]．高橋らのアルゴリズムではデータをパイプライン式に全ノードに配るが，その際に各転送がなるべく他の転送の邪魔にならないようにパイプラインを複数同時に設けて全体のスループットを上げている．こ

† 東京大学
The University of Tokyo

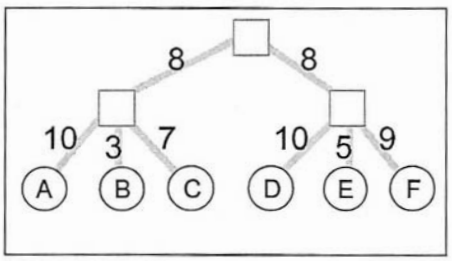

図 1 バンド幅マップ
Fig. 1 A Bandwidth Map

	A	B	C	D	E	F
A	-	3	7	8	5	8
B		-	3	3	3	3
C			-	7	5	7
D				-	5	9
E					-	5
F						-

図 2 バンド幅行列
Fig. 2 A Bandwidth Matrix

のような考えは，バンド幅マップのような詳細な情報を活用することで実現できるようになる．以上のように，ネットワークの通信性能がアプリケーションソフトウェアのパフォーマンスを最適化する為の一つのパラメータとなった近年，リンクのバンド幅を知るということはますます重要になってきている．その他にも，ネットワークの管理やトラブルシューティング，またコンテンツ配信の性能や頑健性の指標として，バンド幅の情報は依然として欠かせないものである．

既存のバンド幅推定手法は数多く存在するが，End-to-End といった一対一の推定のみを対象にしたものが殆んどである．これらの手法は 2 箇所のホスト間の最小のバンド幅をもつリンクの値しか得られないなどの問題があり，分散環境などの複雑なトポロジーを持つネットワークに対してバンド幅マップを得ることは難しい．また，そもそも広域分散環境に対して適用することを想定して設計されていないので，各ノードの組合せに対して逐一に推定を行わなければならないように，N^2 オーダーの時間と手間がかかるという問題点もある．

本研究では広域分散環境におけるバンド幅マップを高速かつ正確に得ることを目的とし，ネットワークトポロジー情報を利用しバンド幅推定を工夫して進めることによってそれを実現する手法を提案する．具体的には，入力されたトポロジー情報を 3 ノードから成る基本セットに分解し，それぞれ並列に推定を進めることによって高速，高スケーラビリティを実現し，トポロジーを考慮してネットワークストリームの流し方を様々に工夫しバンド幅推定をすることによって正確なバンド幅マップの構築を実現する．

本稿の構成は以下の様になっている．2 節で関連研究について述べ，既存手法の問題点を指摘する．3 節で提案する手法について述べる．そして 4 節で提案手法の実装に基づく推定結果を評価する．最後にまとめと今後の課題を述べる．

本論文ではホスト-ルータ間もしくはルータ-ルータ間，さらにホスト-スイッチ間もしくはスイッチ-スイッチ間を接続するポイントツーポイントリンクもしくはブロードキャスト型ネットワークを総称してリンクと呼ぶことにする．実際計算を行うエンドホストやネットワーク機器のルータ，スイッチ等を総称してノードと呼ぶことにする．

2. 関連研究

既存のバンド幅推定手法として，代表的なものに Iperf[4]，Pathchar[1] などがある．これらの手法は，総じて一対一ホスト間の推定を基調とした手法である．すなわち，あるホストに推定サーバをたて，別のあるホストに推定クライアントをたて，この 2 ホストの間でパケットをやりとりし，そのパケットの挙動から 2 ホスト間のバンド幅を推定するという基本概念の手法である．上に挙げた二つ手法の拡張や改良をとげて，他にも様々な手法が，推定精度を追求したり推定負荷の軽減を追求したりといった様々なアプローチで研究されている．しかし，一対一の推定を基調とした手法には，その 2 ホスト間に複数リンクが存在する場合に，ボトルネックリンクの値しか得られない，また推定で得られた値をどのリンクに割り振ったらよいか分からない，といった問題が共通して生じる．例えば，図 1 のようなネットワークがあったとしても，一対一の推定で得られる情報はせいぜい**図 2** のようなバンド幅行列である． バンド幅行列はバンド幅マップから情報量がいくらか削減された表現方法であり，これでは並列分散アプリケーションの厳密な転送・タスクスケジューリングを組めず，ネットワークの管理や運用の観点からしてもイメージ的に乏しい．一対一推定の繰り返しだけでは本研究の目的を達成することはできない．

グリッドの分野でよく知られているパフォーマンス測定ツールに Network Weather Service (NWS)[5] がある．NWS はグリッド上の複数ホストでデーモンを

起動し，CPU やメモリ，ディスク IO などを監視する他，ホスト間のバンド幅も求めている．しかし NWS はトポロジーを考慮しないでバンド幅推定を行うので，推定の衝突が起こるのを恐れ同時に一つのペアしか推定が出来ないようにトークンを回して推定を行い，しかもそれを全ホスト対全ホストについて進めている．この手法はホストの数を N とすると推定にかかる時間は $O(N^2)$ でありスケーラブルとは言えず，またスマートな手法とも言い難い．自動に推定を進めることで手間は省くことができるが，基本としている推定手段が一対一推定なので，結局得られるものはバンド幅行列であり，バンド幅マップのような詳細な情報は得られない．

バンド幅を知るということに関して，推定とは別のアプローチとして，SNMP を用いる手段がある．ネットワーク機器に SNMP で問い合わせることでそれに接続されているリンクのバンド幅を知ることができるので，短時間にバンド幅マップの様な詳細な情報も得ることができる．しかし，依然として SNMP 非対応の機器も多く，ネットワーク上全ての機器についてそれを調査したり置き換えたりすることは現実的ではない．また，MIB の違い等を吸収する上層レイヤーのプロトコルが必要となると考えられるが，それの標準化をどうするかということも考えるべき問題となる．更に考えられる不都合として，SNMP の問い合わせで得られるバンド幅の値は定格値であるということである．定格値を知っても，それはユーザレベルではあまり実用的な値ではない．

3. トポロジーを考慮したバンド幅推定手法

本手法の基本概念は Iperf の考え方を採用する．すなわち，サーバからクライアントへデータを流し，実際に送れたデータサイズをかかった時間で割った値をバンド幅とする．データを流すに当たって，本手法ではトポロジー情報を活用して流し方を様々に工夫することで，既存手法の問題点を克服する．以下の 3.1 と 3.2 で，その流し方の工夫を述べる．この手法を論理ネットワークツリーに適用した場合の処理の流れを 3.3 で説明する．更に，この手法を実際のネットワークに適用した場合を 3.4 で触れる．

以後，本論文ではネットワークトポロジーを表すデータは既知であるとして話を進める．論理ネットワークツリーを表すデータは，白井らにより提案された，ノード間の RTT 測定の結果だけを用いて高速に論理トポロジーのツリーを推定する手法[2] で用意することができる．向きやループを含む実際のネットワークグラフは，例えば traceroute の組合せで用意することができる．

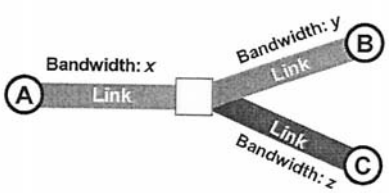

図 3 基本セット
Fig. 3 Elemental Set

3.1 基本セットの推定

図 3 のように 1 スイッチに 3 ノードが接続された構成を基本セットと呼ぶ．ここに 3 本のリンクが存在するが，それぞれのバンド幅 x, y, z の値の組合せ，すなわち基本セットの成し得るバンド幅マップは**表 1** に示す 4 ケースだけ考えればよいことが分かる．このときノード A，B，C の区別を付けず，a < b < c としてよい．今，ノード A とノード B 間のバンド幅推定結果を A-B と表し，ノード A からノード B, C にストリームを枝分かれさせて流したときに A が観測したバンド幅を A-BC と表すとする．A-B，A-C，B-C，A-BC，B-AC，C-AB の推定値は，4 つのケースそれぞれについて**表 2** に示すような値が観測される事が期待される．すなわち，A-B，A-C，B-C，A-BC，B-AC，C-AB 各種推定の結果を表 2 と照らし合わせながら有り得るケースを絞り込んでいけば，3 つのバンド幅の組がどのケースに属しそれぞれ値はどれ程かを推定することができる．これが本手法のバンド幅推定法の一つである．実際に本手法でたどる手順を**図 4** に示す．推定している時点ではノード A，B，C の区別は付いていないことに注意する．

表 1 基本セットのバンド幅マップ
Table 1 Bandwidth Map of Elemental Set

	x	y	z
case 1	a	a	a
case 2	a	a	b
case 3	a	b	b
case 4	a	b	c

表 2 各ケースに期待されるバンド幅推定値
Table 2 Expected Values of Bandwidth for each Case

	A-B	A-C	B-C	A-BC	B-AC	C-AB
case 1	a	a	a	a	a	a
case 2	a	a	a	a	a	min(2a, b)
case 3	a	a	b	a	b	b
case 4	a	a	b	a	b	min(a+b, c)

1 対 1 推定を三種類行う.
二種の異なる値が観測された場合,
　1 対 2 推定を二種類行う.
　異なる値が観測された場合,
　　→ case 4
　等しい値が観測された場合,
　　→ case 3
三つとも等しい値が観測された場合,
　1 対 2 推定を三種類行う.
　二種の異なる値が観測された場合,
　　→ case 2
　三つとも等しい値が観測された場合,
　　→ case 1

図 4　推定手順
Fig. 4　A Procedure of Measurement

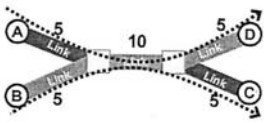

図 5　束ねられたネットワークストリーム
Fig. 5　Bundled Streams of Network Traffic

ところで，Iperf 等の既存手法で図 3 のノード A，B 間を推定したとしても答えは一つの値しか得られない．つまり，得られた一つの値を図 3 中の x, y どちらに割り当てたらよいのか知ることはできない．このような原始的なネットワーク構成でさえも既存手法では正しい答えを表現することができないということが分かる．

3.2　ネットワークストリームを束ねた推定

基本セットで推定を進めても満足に正しい答えが得られない場合がある．例えば**図 5** のようなネットワークを考える．図中の数字はそれぞれのリンクのバンド幅を示す．中央のスイッチ間のリンク (バンド幅 10) を推定しようとしたとき，いずれの左右のノードの組合せ A-D，A-C，B-D，B-C で推定を行っても得られる結果はボトルネックリンクの値，5 となる．そこでノード A からノード D へ、ノード B からノード C へそれぞれ同時にネットワークストリームを束ねるように発生させれば (図 5 中の点線矢印)，両推定とも値 5 が観測され，すなわち中央のリンクはそれらを足した 10，と推定することができる．このように両端がエンドホストでないリンクにおいてはネットワークストリームを束ねて流すことでそのリンクのバンド幅を掴みにいく．これがもう一つの本手法のバンド幅推定法である．Iperf 等の既存手法では図 5 中央のリンクのバンド幅が 10 であることに気づくことはできない．

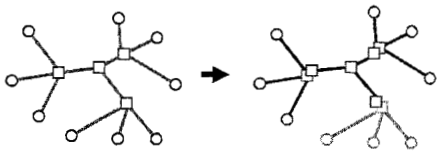

図 6　基本セットに分解されたネットワーク
Fig. 6　A Network as Some Elemental Sets

3.3　論理ネットワークツリーへの適用

論理ネットワークツリーは基本セットの連続としてとらえることができる．与えられた論理ネットワークツリーを基本セットに分解してから処理を開始すれば，各セットを並列に推定することができる．これによってバンド幅マップの構築をより短時間で済ませることができる．Iperf 型のバンド幅推定を複数同時に行う際に注意すべきこととして，同時に推定するホストペアがボトルネックリンクを共有するとお互いの推定が衝突し，正確なバンド幅推定ができないという問題があった．しかし基本セットのような原始的な構造に分解して考えればこの問題は起こらない．実装ではまず与えられたツリーデータの，全ノード (エンドホスト，スイッチ等) を基本セットに分解し，葉の部分の基本セットからボトムアップ式にそれらを並列に推定していく．ここで言っている基本セットの推定とは，3.1 で述べた原理に基づいて三つのリンクのバンド幅の値を決定することである．

具体的に**図 6** で説明する．図中左の様に，4 スイッチ，9 ホストで構成されるネットワークトポロジーが入力されたとする．このネットワークは，1 スイッチ 3 ホストの基本セット三つと，4 スイッチの基本セット一つ，計 4 つの基本セットでできていると考えることができる（図中右）．実装ではまず 1 スイッチ 3 ホストのセット三つを並列に推定し始める．それらが終わると，次に 4 スイッチのセットの推定を始める．ここで三つのサブツリーからそれぞれ 1 ホストを選んでストリームを流すようでは，セットにたどり着く前と後に目的以外のリンクを複数通過するので，それらのボトルネックリンクのバンド幅が観測されてしまい，結果，4 スイッチのセットの推定が誤る可能性がある．そこで本手法は 3.2 で述べた原理によって，複数束ねたストリームを一本のストリームとみなして 4 スイッチのセットの推定を行う．束ねる本数としては必要最小限でできるだけ多くのホストをサブツリーから選ぶ．この例では，3 本のストリームが束ねられることになる．

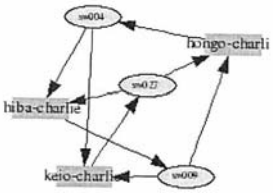

図 7 実ネットワーク
Fig. 7 A Real Network

3.4 実ネットワークへの適用

LAN 内のネットワーク，特にイーサネットで構成されるネットワークはツリー構造である場合が多い．よって，白井らの手法で得た論理ネットワークツリーの上にバンド幅情報を付加したものをバンド幅マップとしても実用性は十分である．ところが現実のネットワーク，特にクラスタ同士を結ぶ WAN 等は循環構造や非対称なリンクを持つことがあり (**図 7**)，これをツリーとして扱うと正しいバンド幅マップは得られない．実ネットワークに本手法を適用する場合，ノードの接続形態の他に通信のルーティング情報も入力し，データを流す方向を制御して基本セットの推定を行うことで，バンド幅マップの構築を実現できる．

4. 評　　価

前節で示した原理に基づいてソフトウェアを実装し，**図 8** のような 11 クラスタ 352 ホストの環境で実験を行った．各クラスタは学術情報ネットワーク SINET3 ☆によって接続されている．入力したトポロジーデータは，白井らの手法で得た論理ネットワークツリーである．得られたバンド幅マップを可視化したものの一部分を，**図 9** に示す．図中の数字は各スイッチ，ルータ同士を結ぶリンクのバンド幅 (Mbps) である．また，バンド幅の大きさによってリンクの太さを変化させている．見易さの為にリーフノードとバンド幅の数値の図示を一部省略している．たとえ地理的に近い位置であっても，バンド幅の大きいリンクと小さいリンクが混在したヘテロなネットワーク環境であることが見て取れる．

4.1 推 定 精 度

バンド幅マップの精度の定量的な評価として，全リンクについて，実際のスループットとの比を**図 10** に示す．本手法で推定した値と Iperf で推定した値の比は，0.95 から 1.05 の間にほぼ収まっており，高い精

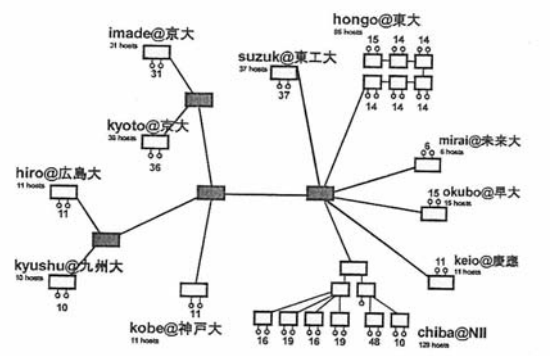

図 8 実験環境
Fig. 8 Experimental Environment

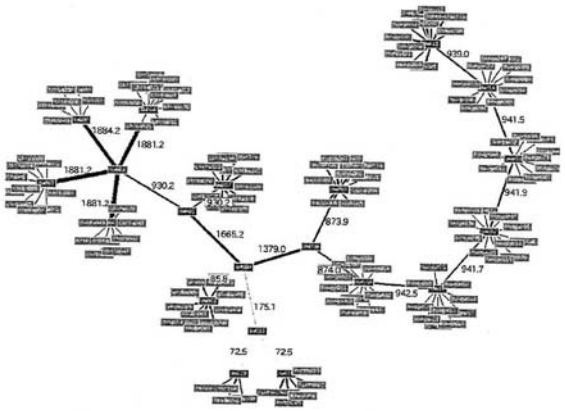

図 9 生成されたバンド幅マップの一部分
Fig. 9 Created Bandwidth Map (selected)

度で推定ができている．推定したリンク中に，End-to-End のリンクよりもバンド幅の太いリンク，すなわちストリームを束にしなくては知り得ないリンクが 12 本，主に WAN 上に存在した．図 9 で言うとバンド幅 1000Mbps を超えているリンクは全てそれにあたる．これらのリンクが実際どれほどのバンド幅をもっているのか，Iperf を同時に起動してストリームを重ねてみたところ，各推定値の和は本手法で推定された値と近い値をとっていることが確かめられた．誤差が生じた推定値が数ヶ所見られたが，これらの多くは NAT を使ってネットワークに接続されているノードまたはそれ付近のノードであった．これらのノードは外との通信を行う際に NAT ルータを経由してしまい，しかも今回の実験環境では NAT ルータは計算ノードも兼ねていた．これが原因で思い通りにネットワークストリームが流れず，基本セットの推定手順が狂い，スト

☆ SINET3
http://www.sinet.ad.jp/

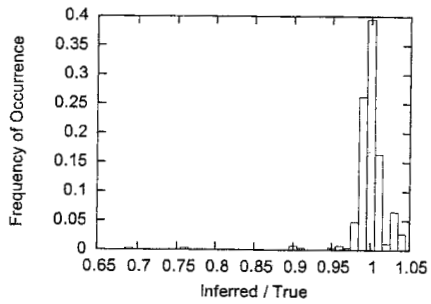

図 10 バンド幅マップの精度
Fig. 10 Accuracy of Bandwidth Map

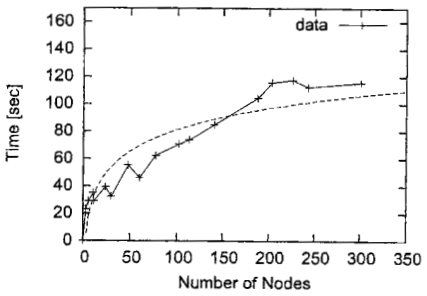

図 11 バンド幅マップ構築にかかる時間
Fig. 11 Time to Build Bandwidth Map

リームの衝突等が起こってしまったことが考えられる．もしくは，ストリームを束ねる際のタイミング合わせに誤差があったなどという原因も考えられる．ネットワークをツリーとして捉えたことによる推定誤差も含まれると思われるが，今回の実験環境においては各クラスタ内などは実際にも全てツリー構造で構築されていたので，その影響があまり見えなかったと考えられる．

4.2 推定時間

バンド幅マップの構築にかかった時間と実行ノード数の関係を**図 11** に示す．図中の点線は log でフィッティングしたものである． 本手法で用いたアルゴリズムはトポロジーを表すツリーの直径に比例した時間を要する．ホスト数を N とするとツリーの直径は $O(logN)$ であるので，ホストの増加に対してスケーラブルな手法と言え，結果からもそれが確かめられた．

5. おわりに

本論文では広域分散環境におけるバンド幅マップを短時間かつ正確に構築する手法を述べた．実験の結果 11 クラスタ，352 ホストを持つ広域分散環境において 120 秒程度でバンド幅マップを構築することができた．得られたバンド幅マップのなかには既存手法では掴むことが不可能なバンド幅も多く存在し，各推定結果は高い精度を持つということも確認した．また我々の手法がツリーのネットワークにおいてはノードの増加に対して高いスケーラビリティを持つことが示された．

今後の課題として，現実のネットワークにも対応させることが考えられる．現実のネットワークとは，向きやループをもった，ツリーではないネットワークトポロジーである．入力を有向グラフとし，ストリームを流す向きも制御して基本セットの推定を進めることで実現することができると考えている．

謝辞 本研究の一部は文部科学省科学研究費補助金特定領域研究「情報爆発に対応する新 IT 基盤研究プラットフォームの構築」の助成を得て行われた．

参 考 文 献

1) Van Jacobson. pathchar - a tool to infer characteristics of internet paths, April 1997.
2) Tatsuya Shirai, Hideo Saito, and Kenjiro Taura. A fast topology inference — a building block for network-aware parallel computing. *In Proceedings of the 16th IEEE International Symposium HPDC 2007*, pp. 11–21, June 2007.
3) Kei Takahashi, Hideo Saito, Takeshi Shibata, and Kenjiro Taura. A stable broadcast algorithm. *to appear the Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid2008)*, March 2008.
4) Ajay Tirumala, Feng Qin, Jon Dugan, and Jim Ferguson. Iperf. http://www.dast.nlanr.net/projects/Iperf/.
5) Rich Wolski, Neil T. Spring, and Jim Hayes. Implementing a performance forecasting system for metacomputing: the network weather service. *Proceedings of the 1997 ACM/IEEE conference on Supercomputing*, pp. 1–19, 1997.