ネットワークトポロジ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．既存の手法では測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[†]

The 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 an information of the network topology combined with the values of bandwidth. Some existing methods to measure bandwidth take a long time to finish, and what is more, we cannot know details such as the Bandwidth Map. In this paper, we propose an efficient and fast 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グリッドと呼ばれる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．グリッド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．ある並列分散プログラムを実装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を行うことが重要になる．以上のようにグリッド上で並列分散計算を実行する際にはアルゴリズム等の他にネットワークを考慮したデータ転送のスケジューリングが重要であり，そしてそのスケジューリングの為には，何らかの手法で，トポロジ情報上の全てのリンクにバンド幅情報を関連付けた，バンド幅マップを構築しておく必要がある．その他にもグリッド環境の管理，トラブルシューティング，コンテンツ配信の性能や頑健性向上の為にはバンド幅の情報は欠かせない．

既存のバンド幅推定手法は数多く存在するが，End-to-End の測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，グリッド環境などの複雑なトポロジを持つネットワークに対してバンド幅マップを構築することは難しい．ま

† 東京大学
The University of Tokyo

た，そもそもグリッド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測定を行わなければならないように多大な時間と手間がかかるという問題点もある．

本研究ではグリッド環境におけるバンド幅マップを高速かつ正確に構築することを目的とし，ネットワークトポロジ情報を利用しバンド幅測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．入力されたトポロジ情報を 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化を実現し，トポロジ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．また本論文では構築したバンド幅マップの応用例として，誤りを含む入力トポロジデータの修正方法と大きいデータのブロードキャスト最適化方法を示す．

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 節で本研究で当たる問題の設定を明らかにし，4 節で提案する手法について述べる．そして 5 節で提案手法の実装に基づく推定結果を評価し，6 節で推定結果を用いた応用例について述べる．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

既存のバンド幅測定手法で代表的なものに Iperf[5] がある．Iperf は各ノードのキュー待ち遅延やフォワーディング処理遅延等のネットワーク機器の遅延は一切考えず，データ転送時間がデータサイズとバンド幅の比で定まるとしている．よってバンド幅は送ったデータサイズを転送時間で割ることで得ることができる．Iperf ではデータサイズと転送時間を大きくすることで前提条件で無視していた遅延誤差を小さくし，正確でばらつきの小さい測定結果を得ることができる．しかも実際にデータを送って測定されるバンド幅の値は実用上という観点からしても信頼性が高い．しかし測定する 2 つのノード間に複数のリンクが介在する場合この手法ではそれらのボトルネックリンクの値しか得らず，それらのうちどのリンクに得られた値を割り振ったらよいのか分からない．すなわち，複雑なネットワーク構成をもつグリッドの様な環境でのバンド幅マップの構築は Iperf では難しい．また異なる N ペアの測定を同時に Iperf で行っているつもりでもそれらの通信経路が共有されてしまうと結果は $1/N$ となってしまう．通信経路の共有はグリッド環境下ではしばしば起こることなので，複数同時進行で Iperf の測定をする際には注意が必要となる．

Pathchar[1] は上で述べた各ノードのキュー待ち遅延やフォワーディング処理遅延等も含めて詳しくネットワークをモデル化し，2 ノード間で多数やりとりしたパケットの到着時間からその間にあるリンクのバンド幅を算出する手法である．Pathchar のネットワークモデルやバンド幅算出方法などのアイデアに基づいたバンド幅推定手法は現在でも様々研究されている．この手法は原理上 2 ノード間に複数のリンクが存在する場合でもそれぞれのリンクのバンド幅を推定することができる．しかしこの手法は計算上大量のパケットのやりとりが必要なので一つのノードペア間の推定に数十分オーダーの時間がかかってしまう．また推定のもととなる数値にキュー待ち遅延時間やフォワーディング処理遅延時間等の確率的な値を用いることになるので推定結果の精度は低く，ばらつきも大きいという問題点がある．更に，推定方法の特性上，スイッチングハブ等のネットワークレイヤ 2 以下のノードが 2 ノード間に存在すると正しい推定ができないという問題点もある．グリッド環境では多段にスイッチ接続されたネットワーク構成をとることが多く，この手法で正しい値が得られる確証は小さい．またグリッドのような測定箇所の多い環境下で測定時間が長くかかってしまう手法をとることは現実的ではない．

Pathchar の考え方を拡張した Packet-Pair 推定法がある．この手法は二つの同サイズのパケットをあるノードから別のノードへ連続送信し，各パケットの到着時刻の変化から 2 ノード間のバンド幅を算出する方法である．Pathchar に比べ推定にかかる時間が短いほか，ネットワークレイヤ 2 以下のノードをまたいでも正確に推定できるという特徴を持つ．しかし原理上 2 ノード間のボトルネックリンクしか測ることが出来ず，Pathchar と同様の理由により推定結果の精度は低くばらつきも大きい．従ってこれらの手法もグリッド環境でバンド幅マップを構築するのには不向きである．Packet-pair 推定法を原理とした代表的なプログラムには Nettimer がある．

グリッド環境を念頭に置いたパフォーマンス測定ツールに NWS[6] がある．NWS は複数ホストで起動し，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 はトポロジを考慮しないでバンド幅測定を行うので，測定の衝突が起こるのを恐れ同時に一つのペアしか測定が出来ないようにトークンを回して測定を行い，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 とすると測定にかかる時間は $O(N^2)$ でありスケーラブルとは言えず，しかもバンド幅マップのような詳細な情報も得られない．

3. 問題設定

まずリンクのバンド幅を測定するということについて以下に述べる．一般的に二つのノードとそれらを繋ぐリンクにはバンド幅と遅延の二つの要素値が割り当てられる．バンド幅とはそのリンクを単位時間当たりに通ることのできるデータサイズである．バンド幅の値はリンクによって数 Mbps から数 Gbps までと様々であるが，現在イーサネットでは 1Gbps のバンド幅が広く使われている．遅延とはノードからもう一方のノードへパケットが届くまでの時間である．これには両端のノードでパケットをプロセスするのにかかるオーバーヘッドや光や電気の信号の伝搬にかかる時間が含まれる．この値は送るパケットのサイズに依らず，またそれぞれのノードやリンクで固有の値をとる．遅延の値はリンクや経由するノードによって数マイクロ秒から数ミリ秒までと様々である．実際のネットーワークは他にも様々な要素があり複雑だが，このような簡単なネットワークモデルでもデータを送るのにかかる時間を知ることに関しては良く表されている．このモデルでは，リンク l のバンド幅を $v(l)$，遅延を $t(l)$，送信するデータのサイズを S，送信にかかった時間を T とすると，$v(l)$ は次の式で得られる．

$$v(l) = \frac{S}{T - t(l)} \tag{1}$$

右辺分母の $t(l)$ は一般に数マイクロ秒から数ミリ秒の値をとる．例えば T を 1 秒とすれば $t(l)$ との間に 1000 倍程の差がつくことになる．そこで T の値を操作し $t(l)$ が無視できる程の値に設定すればバンド幅は単純に S を T で割った値で求められる．ただし T を操作すれば送るデータサイズ S は T の関数 $S(T)$ となる．本研究では T の値を 1 秒から 2，3 秒の範囲で設定し，バンド幅を次の式で定めることとする．この計算でバンド幅を求める方法は，2 節で紹介した Iperf と同等な方法である．

$$v(l) = \frac{S(T)}{T} \tag{2}$$

一般的にネットワークや計算ノードに与える負荷とバンド幅測定結果の精度はトレードオフになっていることが多く，現在もより良いバンド幅測定手法を求めて数多く研究が進められている．この方法でバンド幅を求めることは，T 秒間の間ネットワークをほぼ 100%占領することになり，Iperf 等の Active Measurement の明らかな欠点である．そうでありながらそれで得られる値が実用的な意味で非常に正確で有用なために，これによって測定を行う必要性は非常に高い．

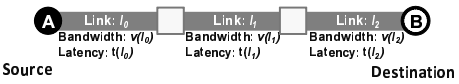

図 1 複数のリンクで接続されるノード
Fig. 1 Nodes connected with several links serially

通常，送信ノードと受信ノードの間に複数のルータまたはスイッチが存在する (**図 1**)．この図の場合 (2) で求めたバンド幅は $min\{v(l_0), v(l_1), v(l_2)\}$ を求めていることになるが，ホスト群の接続形態に関わらないルータやスイッチ，つまりホスト群の分岐に関わらないルータやスイッチしか 2 ノード間に無い場合であれば，それらのノード間にはバンド幅 $min\{v(l_0), v(l_1), v(l_2)\}$ を持つリンクが一本あると考えることにしても実用上問題は無い．

次にネットワークトポロジについて，本研究では通信に参加するノード（エンドホスト）と中継するだけのノード（スイッチ，ルータ等）の二種類のみで構築される，ツリーとして取り扱う．LAN 内のネットワーク，特にイーサネットのトポロジは通常ツリー構造である．従ってクラスタ内ではツリーモデルを適用しても差し支えない場合がほとんどである．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，それらをツリーとして扱って良いか否かには議論の余地がある．しかし WAN では一般的に LAN より広いバンド幅を持つリンクで構成されていることが多く，この場合，バンド幅測定という観点からすると，クラスタ間を結ぶ WAN をツリーとして表現しても問題無いと言うことができる．具体的には**図 2** のようにツリーに置き換える． この時，バンド幅 x, y, z の値がそれぞれ a, b, c のいずれよりも大きい場合，置き換えてもよく実用上も問題は無い．なぜなら，WAN に到るまでの経路がボトルネックとなり，x, y, z の値は見えなくなるからである．非対称なリンクの存在を考える場合，本手法は各リンクについて方向を考えて二度推定し，有向リンクとして二つの値を関連付けることでツリーを構築できると考えられる (ただしこの場合，単純に考えただけでも 2 倍の推定時間を要することとな

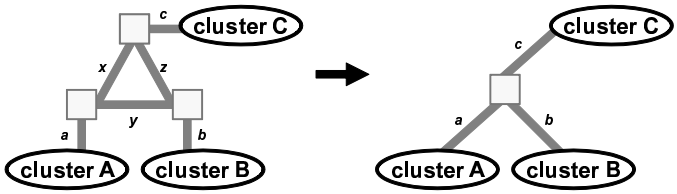

図 2 ツリーで表現しなおしたネットワーク
Fig. 2 Network Topology Replaced with Tree Structure

る)．更にツリーにすることで冗長化されたリンクは一本のリンクとして表現されてしまうが，パラレルに接続された各リンクの和のバンド幅を持つリンクがそこに一本あるとみなすことにしても実用上都合の悪いことは無い．以上のように実際のネットワークにおいてもツリーモデルの適用範囲は広い．

バンド幅マップの情報を用いたアプリケーションは，対象とするホスト群の分岐の関係と，ノード間の通信性能の情報が重要であり，ホスト間の詳細な経路や分岐に関与しないスイッチを通るか通らないかなどの情報は特に問題にはならない．従って我々はネットワークを論理トポロジのツリーとして扱い，その上にバンド幅マップが構築できればよいと考える．

4. トポロジを考慮したバンド幅推定手法

本手法は 3 節で示したネットワークモデルに基づき 2 節で紹介した Iperf のように実際にネットワークストリームを流す基本原理でバンド幅を推定していく．ストリームを流すに当たって本手法ではトポロジ情報を用いて流しかたを様々に工夫することで，以下の 4.1 と 4.2 で述べる二つの特徴的な推定手法を実現している．本手法はこれら二つを組み合わせて進められる．その実行の全体的な流れを 4.3 で説明する．以後，本論文ではネットワークトポロジを表すデータは既知であるとして話を進める．トポロジデータは，一つの方法として，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のツリーを推定する手法[3] で用意することができる．

4.1 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 間のバンド幅測定結果を 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 の測定値は，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 各種測定の結果を表 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．これが本手法の独特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．

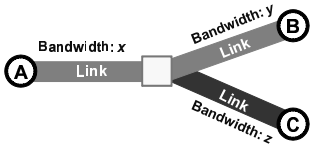

図 3 基本セット
Fig. 3 Elemental Set

ツリーネットワークトポロジを基本セットの連続として捉え，与えられたトポロジ情報を基本セットに分解して推定を開始すれば，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

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測定値
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 測定を三種類行う.
二種の異なる値が観測された場合,
　1 対 2 測定を二種類行う.
　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 測定を三種類行う.
　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　測定手順
Fig. 4　A Procedure of Measurement

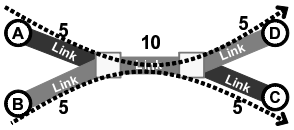

図 5　束ねられたネットワークストリーム
Fig. 5　Bundled Streams of Network Traffic

ド幅測定を並列に行う際に注意すべきこととして，同時に測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測定が邪魔をし正確なバンド幅測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．実装ではまずネットワークトポロジを表すツリーの，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．

ところで，Iperf 等の既存手法で図 3 のノード A, B 間を測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．

4.2　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) を測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 で測定し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 へそれぞれ一斉にネットワークストリームを束ねるように発

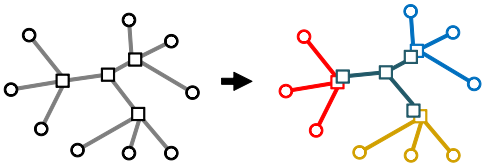

図 6　基本セットに分解されたネットワーク
Fig. 6　A Network as Some Elemental Sets

生させれば (図 5 中の点線矢印)，両測定とも値 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独特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．

4.3　実行の全体的な流れ

本実装にトポロジデータを入力すると，まずエンドホストやスイッチ等の中継ノードも含む全ノードを基本セットに分解する. そして葉の基本セットを並列に推定し，以後ボトムアップ式にツリーの中心へ辿り基本セットを推定していく．基本セットを推定するとは，4.1 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 4.2 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．

5.　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 7** のような 6 クラスタ 333 ホストの環境で実験を

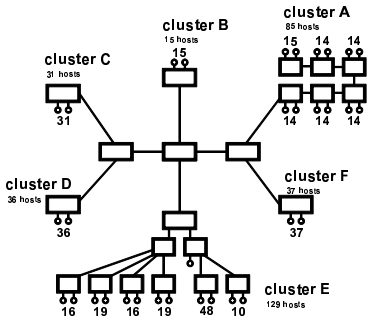

図 7 実験環境
Fig. 7 Experimental Environment

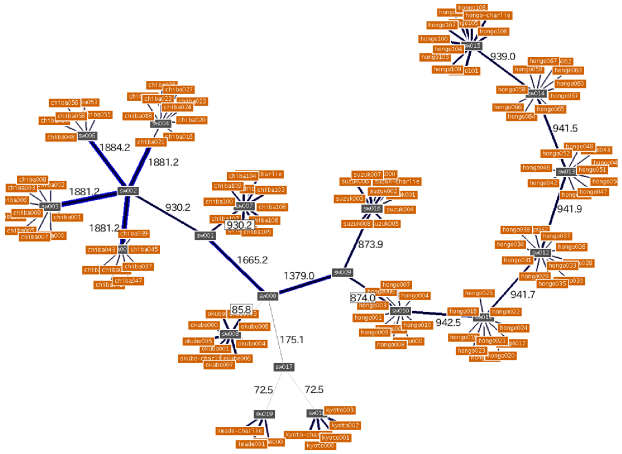

図 8 生成されたバンド幅マップ
Fig. 8 Created Bandwidth Map

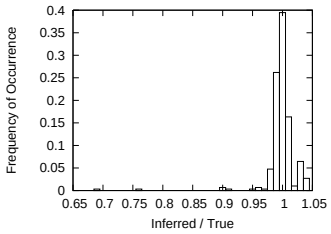

図 9 バンド幅マップの精度
Fig. 9 Accuracy of Bandwidth Map

行った．Cluster A，B，F は東京に，E は千葉に，C，D は京都に置かれている．Cluster C，D は数台のホストがグローバル IP アドレスを持ち，他のホストは NAT により構成されている．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．エンドホストは 301 台ありスイッチ，ルータは合わせて 20 台，そして推定すべきリンクは 320 本であった．この環境全体に対して実行したところ，終了にまでかかった時間は 1 分 55 秒であった．得られたバンド幅マップを可視化したものを**図 8** 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．

5.1 推定精度

バンド幅マップの精度の定量的な評価として，320 本の全リンクについて，実際のスループットとの比を**図 9** に示す． 本手法で推定した値と Iperf で測定した値の比は，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 8 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，各測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，しかも実験環境では NAT ルータは計算ノードも兼ねていた．この状況は，3 節で設定した“中継するだけのノード”という仮定に反している．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，ストリームの衝突等が起こってしまったことが考えられる．

5.2 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 10** 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．

☆ SINET3
http://www.sinet.ad.jp/

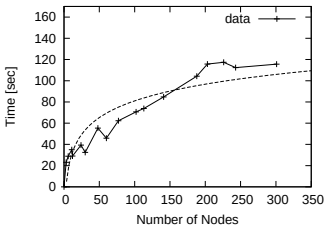

図 10 バンド幅マップ構築にかかる時間
Fig. 10 Time to Build Bandwidth Map

6. 推定したバンド幅マップの応用例

6.1 誤りを含む入力トポロジデータの修正方法

ネットワーク構成を記述するための主要なパラメータには遅延，バンド幅，そしてトポロジの三つがある．白井らはこれらのうち遅延の値を基にトポロジを推定する方法を提案している．そこで我々はバンド幅の値も考慮に入れてトポロジを推定する方法を以下のように考える．白井らのアルゴリズムは一つのスイッチに N ノード接続された形を，二つのスイッチにそれぞれ K，L（$K+L=N$）ノード接続されている形に分解して推定してしまうケースがある．これは同スイッチ内でも微妙な遅延の揺らぎがあるが，それを手入力の閾値で判断しているために起こる．このような誤りを含んだトポロジデータを我々の手法の入力として実行すると，分解された二つのスイッチ間のリンクを推定するために $min(K,L)$ 本のネットワークストリームの束を作ってこのリンクに流そうとする．この時，異様に大きいバンド幅が観測された場合や束ねれば束ねる程流れてしまうような場合，そこのリンクは実際には無く，その両端のスイッチをマージして一つのスイッチと考えられるのではないか，と疑いをかける．このようにして誤ってスイッチを分解して推定されたトポロジデータを修正することができると考える．関連する研究として，Shao らはバンド幅の値からトポロジを推定する手法を提案している[2)]．Shao らの手法もバンド幅測定の競合の様子からホストの接続形態の推定を試みている．

実際に一つのクラスタ内，図 7 中の cluster E で我々の考えを適用した．まず白井らのアルゴリズムでこのクラスタのトポロジを推定したものが**図 11** である． 次にこの推定結果を本手法に入力し，トポロジの修正をかけたものが**図 12** である．今回，疑うべきスイッチがなくなるまでに本手法の三回の実行が必要となった． そして正解のトポロジを手入力でだしたものが**図 13** である．

図 11 の図は枝わかれが多く，ネットワークがどの様になっているのかが一目では分りづらい．これはスイッチの誤った分解が多段に重なりあってしまったためである．一方この節で考えた手法で冗長なスイッチをマージした図 12 は図として非常に見やすくなり，現実のトポロジの図 13 に近い形になっていることがわかる．修正の結果，図 12 では右下に大きなホストの塊ができてしまったが，これは現実の図 13 の右下の二つの塊がスタッカブルスイッチで接続されているために物理的にはスイッチで二つに分かれて接続されていてもバンド幅の性能的には一つのスイッチにホストが集まっていると考えられてしまったためである．

6.2 大きいデータのブロードキャスト最適化方法

効率的なブロードキャストアルゴリズムとして高橋

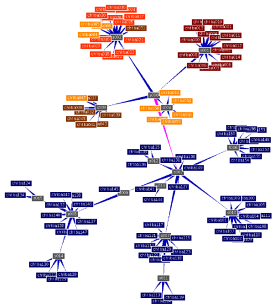

図 11 トポロジ推定結果
Fig. 11 An Inferred Topology

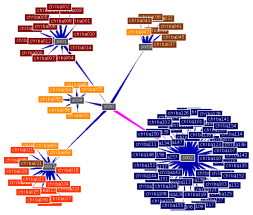

図 12 修正をかけたトポロジ図
Fig. 12 A Figure of Fixed Network Topology

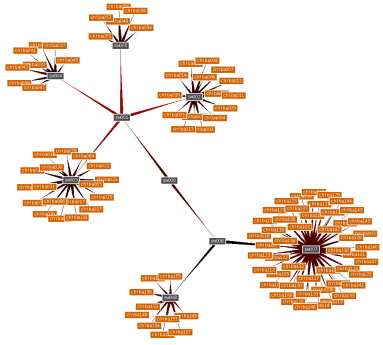

図 13 cluster E の正確なトポロジ図
Fig. 13 Real Network Topology of cluster E

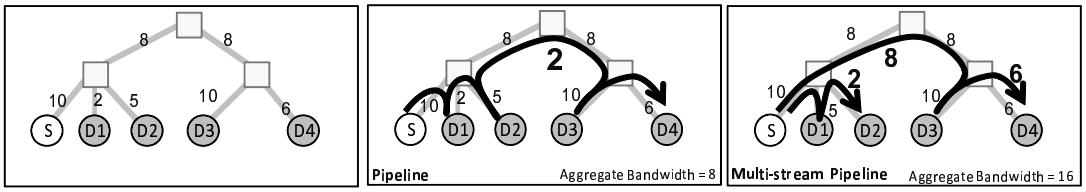

図 14 安定したブロードキャスト最適化方法
Fig. 14 A Stable Broadcast Optimization

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[4]. 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設けて全体のスループットを上げている．簡単な例として**図 14** に示す． 図中左の様なネットワークがあったとする. 各リンクに振られている数字はバンド幅を示す．今，ノード S から他の 4 つのノードへデータを配布することを考える. 転送のパイプラインを一つしか用いないと通過するリンクの最小のバンド幅がそのパイプラインの全体のスループットとなる (図中真中). しかし各リンクのバンド幅が分かっていれば，複数のパイプラインを用いてもリンクの共有による影響が出ないなどといったことが予め分かるので，図中右のように二本のパイプラインを用いてブロードキャストの性能をあげることができる．このような技巧的な考えは，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．尚，更に複雑な例やアルゴリズムの詳細と正当性の証明が論文中で挙げられている．

7. おわりに

本論文ではグリッド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アルゴリズムとその結果を用いた応用例としてトポロジ情報を修正する方法を述べた．実験の結果 6 クラスタ，301 ホスト，20 スイッチ，そして 320 リンクを持つグリッド環境において 1 分 55 秒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．

性能改善の余地として一つに，基本的なバンド幅測定部の精度向上，つまり Iperf そのものの精度向上の追求がある．現段階では各測定の所要時間を，精度が悪くならない範囲の値で手設定で定めている．この時間を出来るだけ小さくすることは全体の推定にかかる時間を小さくする鍵になっている．しかし測定の精度と掛ける時間はトレードオフになっているので，それらのバランスをとるように所要時間を適応的に決定するといったようなよりよい設定方法を考える必要がある．今後はまずこれらの問題にあたり，推定精度を上げつつ推定時間を狭めていく．更にバンド幅と遅延，トポロジの三つのパラメータが判明したところでそれらの活用・応用方法についても考えていく．

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) Gary Shao, Fran Berman, and Rich Wolski. Using effective network views to promote distributed application performance. *In Proceedings of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applications*, pp. 2649–2656, 1999.
3) 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.
4) 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.
5) Ajay Tirumala, Feng Qin, Jon Dugan, and Jim Ferguson. Iperf. http://www.dast.nlanr.net/projects/Iperf/.
6) 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.