広範なトラフィック要求に対応する負荷分散経路計算アルゴリズム

小原泰弘[†] 今泉英明[††,☆] 加藤朗[††]
中村修[†††] 村井純[†††]

インターネットではトラフィック要求が変化した後の輻輳回避を考慮していない．本研究では広範なトラフィック要求に対応することにより，トラフィック要求が変化しても通信性能が低下しないネットワークを実現する．トラフィック要求に前提を置かない負荷分散経路計算アルゴリズム，Load Balancing Routing Algorithm（LBRA）を設計し，シミュレーションにより広範なトラフィック要求に応えることを評価した．LBRA は，ホップバイホップネットワークにおいて，すべてのノードから終点に対する実現可能な通信量を最大化する．このため，ある 2 地点間のトラフィックが飛びぬけて多い場合，既存の最短ホップ経路制御より輻輳の可能性が低い．ランダムなトラフィック要求の場合は，最短ホップ経路制御と同程度の回線利用率となる．しかし，すべての回線の帯域が同一であるトポロジでは，最短ホップ経路制御と比較して輻輳する回線数が増加し，問題があることが分かった．

A Load Balancing Routing Algorithm for Wide Range of Traffic Demands

Yasuhiro Ohara,[†] Hideaki Imaizumi,[††,☆] Akira Kato,[††]
Osamu Nakamura[†††] and Jun Murai[†††]

The congestion avoidance after a traffic demand change is not considered in the current Internet. This research aims to achieve a network without performance degradation after change of the traffic demands. A Load Balancing Routing Algorithm (LBRA) that does not presume any traffic demands is proposed, and is shown by simulation that it supports a wide range of traffic demands. LBRA maximizes a minimum feasible communication bandwidth from each node to the destination in the network. Hence the possibility of congestion is relatively low in LBRA when a traffic demands have a large traffic flow on a source-destination pair. For random model traffic demands, LBRA shows equivalent link utilization with InvCap Dijkstra. A particular network topology with constant link bandwidth was found problematic on LBRA, where the number of congested links increases compared to existing standard minimum-hop shortest path routing.

1. はじめに

ネットワークの回線利用率を最適化し，ネットワーク全体の通信性能を向上させる目的で通信経路を操作する技術を，トラフィックエンジニアリングという[1)]．現在インターネットサービスプロバイダはオフライン方式のトラフィックエンジニアリング技術を利用している[2),3)]．オフライン方式では，トラフィック要求の量をあらかじめ計測しておく．そして，この過去のトラフィック要求の最大値をもとに，最適な経路を計算し設定する．しかし，オフライン方式のトラフィックエンジニアリングでは，つねに変化する実際のトラフィック要求に対応することはできない．また，回線の切断などネットワークトポロジの変化に対応できない．

オフライン方式のトラフィックエンジニアリングの効果は，あらかじめ計測したトラフィック要求の精度に依存する．しかし，あらかじめ計測したトラフィック要求と実際のトラフィック要求は，以下の 3 つの理由から完全に同一とはならない．

第 1 に，トラフィック要求を正確に計測することは困難である．ネットワーク全体の正確なトラフィック要求は，ネットワークのすべての地点で観測しなければ得られない．このため，特にネットワークの規模が拡大するにつれ，ネットワーク全体のトラフィック要求を正確に計測することは困難になる．

† 慶應義塾大学大学院政策・メディア研究科
Graduate School of Media and Governance, Keio University
†† 東京大学
The University of Tokyo
††† 慶應義塾大学環境情報学部
Faculty of Environmental Information, Keio University
☆ 現在，東京大学大学院工学系研究科特任助手

第 2 に，実際のトラフィック要求はつねに変化している．トラフィック要求は同日の朝と夜でも異なるため，ある特定の時刻でのトラフィック要求は，別の時刻のトラフィック要求とは異なったものとなる．また，突発的な要求が生じる場合もある．

第 3 に，トラフィック要求はさまざまな外的要因からも変化する．回線の切断などネットワークトポロジの変化だけでなく，他の組織の BGP [4] の経路変更や，突発的な DoS（Denial of Service）攻撃など，外的要因によってトラフィック要求は容易に急激に変化する．

このようなオフライン方式の問題点に対応するため，オンライン方式のトラフィックエンジニアリング手法が研究されている．しかし，既存のオンライン方式の手法は，次に述べるように課題が多い．

まず，多くの手法は MPLS [5] など回線交換型のネットワークを前提としており，ホップバイホップネットワークでは利用できない．また，MPLS を利用したオンライン方式のトラフィックエンジニアリングでは，機能がネットワークの一部に集中化され，単一障害点（Single point of failure）となる可能性がある．さらに，MPLS を利用する技術の多くは，複数の LSP（Label Switched Path）があらかじめ設定されていることを前提としているため，予期された回線切断にしか対応できない．多くのノード数やパス数を持つ大規模なネットワークでは，あらかじめ設定する必要のある LSP の数が膨大になるため，実現が困難であるなどの問題点がある．

MPLS を利用しないオンライン方式のトラフィックエンジニアリングではトラフィックの変化から経路振動が持続することが古くから知られており，安定性が問題視される [6),7)]．また，終点に対する単一の経路しか計算しないため，ある 2 地点間のトラフィックが飛びぬけて多いような，極端なトラフィック要求をサポートできない．

つねに変化するトラフィック要求に対して，通信性能の良い安定した経路を提供するためには，時間による変化に対応できる，突発的な要求を許容するなど，広い範囲のトラフィック要求に経路制御機構が対応する必要がある．これは，幅広い種類のトラフィック要求に対して輻輳しない経路を提供することによって実現できる．以上を実現するため，ネットワークトポロジに適した負荷分散経路（複数経路）をあらかじめ計算する手法を提案する．提案する経路計算アルゴリズムを，Load Balancing Routing Algorithm（LBRA）と呼ぶ．

LBRA の要件は以下の 2 点である．

(1) 特定のトラフィック要求を前提としない．
(2) 広範な種類のトラフィック要求に対して良好な通信性能を提供する．

本稿では，特定のトラフィック要求を前提としない経路制御機構の設計を述べる．その後，2 地点間のトラフィックが飛びぬけて高いような特異なトラフィック要求と，ランダムなトラフィックモデルに対してどのような性能を発揮するかを調べ，広範なトラフィック要求に対応するかを評価する．

本稿の構成は以下である．関連研究を 2 章に示す．提案するアルゴリズム LBRA の概要を 3 章に示す．4 章では，仮想ネットワークにおけるシミュレーションによって，各ネットワーク回線の帯域利用率を調べ，ダイクストラ法による最短経路と比較し評価する．本稿を 5 章でまとめる．アルゴリズムの詳細を付録 A.1 に，部分正当性を付録 A.2 に，停止性を付録 A.3 に，計算量を付録 A.4 に示す．

2. 関 連 研 究

グラフ理論において，ネットワークのある 2 点間の流量を最大化する手法に，ford-fulkerson 法がある [8)]．しかし，ford-fulkerson 法は特定の 2 点間の最大流量を実現するのみで，ほかの 2 点間の流量はまったく考慮されず，実際の通信ネットワークの利用には適さない．

Fortz ら [9),10)] は，特定のネットワークトポロジとトラフィック要求に対して，発見的手法により近似解としての経路制御メトリックを計算した．しかし，対応するトラフィック要求は限定されたものであり，回線切断などのネットワークトポロジ変化後は，利用効率が大幅に低下する．

Applegate ら [11)] は，線形計画モデルを構築し．トラフィック要求とその近接部分を含む周辺に対して効率的な経路制御を計算した．本手法は，特定のトラフィック要求を前提とする．また，実際のネットワークではこの手法の計算時間は膨大なものとなる．回線切断時の利用率については十分に調査されていない．

MATE [12)] や TeXCP [3)] は，MPLS ネットワークにおいて，あらかじめ設定された LSP の間で負荷分散を実行し利用効率を改善する．回線の障害を考慮しつつ，大規模なネットワークで必要な LSP をあらかじめ設定しておくことは難しい．複雑で大規模なネットワークの上での動作は調べられていない．

ホップバイホップネットワークにおけるオンライン方式に PBR [13)] がある．これは特定のトラフィック要求を前提とせず，トラフィック要求の変化や回線断に適

応する，安定した経路制御を実現する．この手法は単一の経路を計算するもので，始点終点が同じトラフィックをネットワーク上で分割しないため，ある 2 点間のトラフィックが飛びぬけて大きいようなトラフィック要求に対応できない．

単一経路計算の問題点を解決するものに，複数経路計算がある．MPDA [14]，MDVA [15] は最小遅延経路制御を実現するために複数経路を計算する．これは最短な複数経路を計算することが目的であり，リンクメトリックを設定することによって最大帯域を目的とした経路制御を実現するのは困難である．そのため，広範なトラフィック要求には対応できない．

3. 提案するアルゴリズム：LBRA

3.1 目　　的

トラフィック要求を前提とするトラフィックエンジニアリング手法では，実際のトラフィック要求が前提と異なれば，期待した利用効率が得られない．本研究では特定のトラフィック要求を前提とせず，できるだけ広い範囲のトラフィック要求に対して，輻輳しない経路を提供する．具体的には，ネットワーク上のすべての始点終点の組において，既存の単一経路制御手法より実現可能な通信量を増加させることを目的とする．

既存の単一経路制御手法では，ある 2 地点間の最大流量は，最短経路上の各回線の帯域幅に依存する．最短経路上にない回線は，帯域に余裕があっても利用されない．2 地点間の最大流量は最短経路の帯域幅に制限される．最短経路の帯域幅を超える流量が 2 地点間に流れると，回線は容易に輻輳する．このことから，既存の単一経路制御手法では，ある 2 地点間のトラフィックが飛びぬけて高いトラフィック要求には対応できない．

理論的には，グラフ構造上の 2 地点間の最大流量は，最短経路の容量のみではなく，代替経路の本数とその容量（最小カット容量）で決定される．ford-fulkerson 法では，複数経路を活用して 2 地点間の最大流量を計算できる．しかし，既存のホップバイホップネットワークでは，複数経路が活用できないために，このような最大流量が実現できない．また，前述のように，ホップバイホップネットワークではある 2 地点間の最大流量は他の 2 地点間の最大流量と矛盾した経路を持つ．ここからも，実際のネットワークでは 2 地点間の最大流量の理論値を利用することはできない．

3.2 概　　要

LBRA はトラフィック要求に前提を置かないため，経路制御機構はネットワークトポロジ（グラフ構造と回線容量）のみから経路を計算する．ここで輻輳を回避するための最善の策は，すべての始点終点の組において最大通信量が最小となる組の通信容量を最大化する経路を計算することである．LBRA はそのような経路を計算する．

ホップバイホップネットワークでは，経路制御アルゴリズムはネットワークグラフ構造の部分グラフとなる有向グラフを各終点ごとに作成する．たとえば，**図 1** の (a) のネットワークがあるとき，既存の単一経路制御機構では，t 向け経路に (b) のような部分有向グラフを作成する．ノードからの外向きに張られる辺は基本的に 1 本であり，これが各ノードの経路表に設定される．この例では，ノード e のみが例外的に t 向け経路にネクストホップ s と c を利用しており，複数経路の同時利用（ECMP: Equal Cost Multi Path）となっている．

しかし，(c) は正しいホップバイホップ経路ではない．(c) では，s から t 向けへの通信パケットは，$s \to d \to b \to e \to s$ という経路ループをたどる可能性がある．このような経路ループを回避するため，LBRA ではパスベクタ型経路ループ回避アルゴリズム [4] を採用している．

通常の単一経路制御 (b) では，s から t への通信は $s \to d \to b \to t$ をたどる．t に接続する回線の帯域がすべて 100，それ以外の回線の帯域が無限であるとすると，当然 $s \to t$ で最大に利用可能な通信帯域は 100 である．$d \to t$ も同様に最大通信可能帯域 100 である．しかし，もし経路を (d) のように設定すれば，$s \to t$，$d \to t$ の最大通信可能帯域はそれぞれ 300，200 に増加する．LBRA はこのように，各ノードから終点への通信可能帯域を最大にするような経路を計算する．

LBRA は，各終点に対して個別に，ネットワーク全体ですべてのリンクを含む DAG（Directed Acyclic Graph）を作成する．つまり，すべてのリンクが終点への負荷分散に利用されるように，経路を計算する．LBRA は，特定の終点への経路に関して，ループが構成されないように，各リンクに対して向きを与える．すべてのリンクに向きが与えられると，各ノードからその終点への実現可能な通信量が計算できる．これを，ノードの可能通信量と呼ぶ．

次に，各リンクの向きを変更することによって，ノードから（その終点に対する）可能通信量の増減を調査していく．増加する場合は，リンクの向きを変更し，ノードの可能通信量を更新する．このノードの可能通信量の変化によって，他のリンクの向きをさらに変更

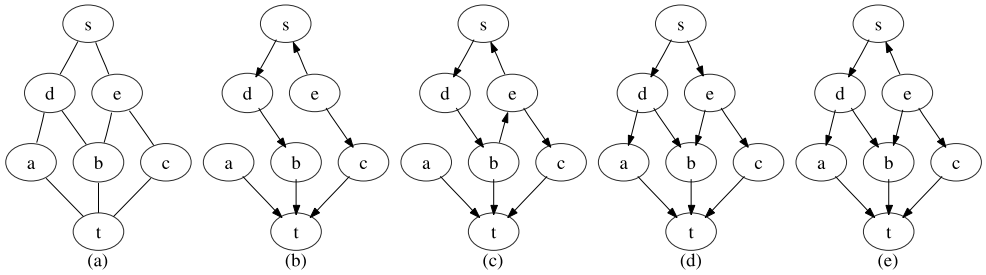

図 1 経路制御トポロジ例
Fig. 1 An example routing topology.

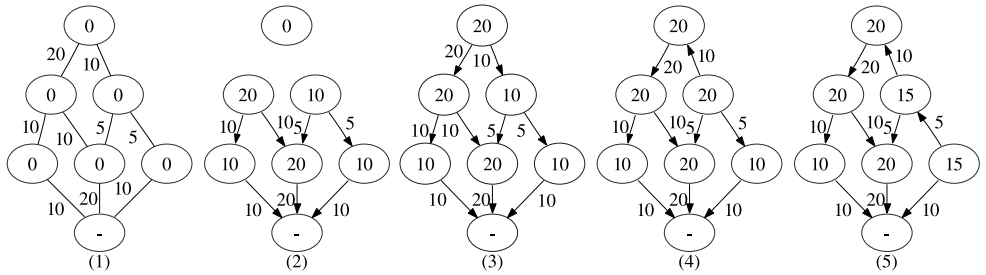

図 2 アルゴリズム動作例
Fig. 2 An example process of the algorithm.

する必要がある可能性がある．そのため，LBRA はまたすべてのリンクの向きを検査する．これを繰り返すことによって，LBRA はすべての始点終点の組で最小の可能通信量を最大化する．たとえば 図 1 (d) において $e \to b$，$e \to c$ の帯域がとても狭いような場合には，$s - e$ 間のリンクの向きを変更し，$e \to t$ のパスを増やすことによって $e \to t$ の通信帯域を増加させる（図中 (e)）．

この状態では，ノードは終点に対し複数のネクストホップを持つ．LBRA の目的は，各ノードの各終点に対しネクストホップ集合を計算することである．そして，その複数のネクストホップのそれぞれには，分割割合が設定される．この分割割合に従って，ノードはトラフィックを経路上に分散する．LBRA は，隣接ノードの通信可能帯域の比をトラフィック分割割合として利用する．付録 A.1 に LBRA の全体計算アルゴリズムを示す．

LBRA では，計算途中で各ノードの通信可能帯域を加算していく．通信可能帯域を加算する際，加算する複数の経路が，経路途中で合流している可能性がある．これを検知するために，LBRA は各ノードから終点へ到達する際の中継ノードリストと，各ノードの可能通信量を保持する．この中継ノードリストはパスの数の増大に影響されないよう，パスに存在する中継ノードを集約したリストとなっている．そのため，LBRA では経路が途中でどのように合流しているかを判断できない．LBRA では，加算しようとする 2 つの経路の中継ノードリストに共通なノードがあれば，経路が合流していると判断され，加算後の通信可能帯域は最大でも共通ノードのそれ以下となる．

しかし，この方法では可能通信量を少なくしすぎてしまう場合がある．たとえば，図 1 (d) において，s から t への可能通信量は，$b \to t$ の可能通信量となってしまい，$a \to t$ や $c \to t$ の通信帯域が考慮されない．このような状況を緩和するため，LBRA では可能通信量を少なくとも経由する隣接ノードの可能通信量の最大値にする．図 1 の例では，s から t への可能通信量は，d が集約する $a \to t$ と $b \to t$ の和か，e が集約する $b \to t$ と $c \to t$ の和か，どちらか最大のものとなる．

図 1 のトポロジに帯域を設定した例を，**図 2** に示す．これを利用して，LBRA の動作例を示す．(1) はリンクの帯域と，各ノードの初期の可能通信量を表している．(2) では，t に対して，図の上側のノードが下側のノードからリンクの帯域分の可能通信量を受け継いでいる．複数のノードから受け継いだ可能通信量は和をとって自身の可能通信量とする．(3) では s が d から 20，e から 10 受け取っているが，双方の中継ノードリストに共通のノード b があるため，s の可能通信量は 20 に制限される．その後，(4) では $s \to e$ のリンクが $s \leftarrow e$ となり，e の可能通信量が 20 に増える．最後に (5) で $e \to c$ が $e \leftarrow c$ となり，e と c

表 1 BRITE トポロジ生成：名称と設定
Table 1 Topology generation using BRITE: name and configuration.

名称	生成モデル	ノード数	リンク数	ノード配置	帯域割当て
BA-20-HT-C	Barabási-Albert	20	37	Heavy Tail	Constant
WA-20-HT-C	Waxman	20	40	Heavy Tail	Constant
WA-20-RN-C	Waxman	20	40	Random	Constant
BA-20-HT-U	Barabási-Albert	20	37	Heavy Tail	Uniform
WA-20-HT-U	Waxman	20	40	Heavy Tail	Uniform
WA-20-RN-U	Waxman	20	40	Random	Uniform

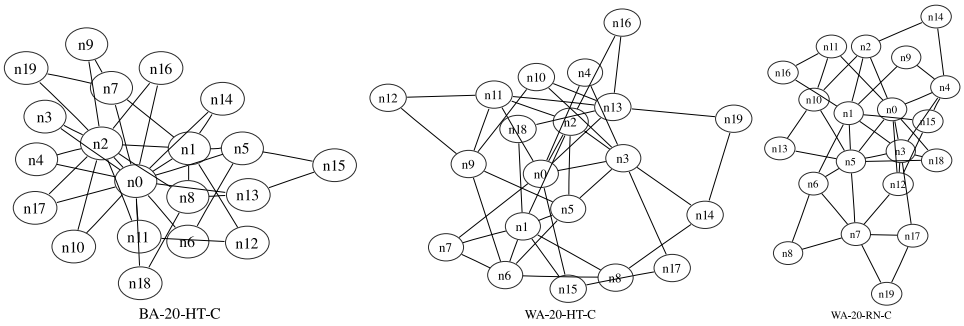

図 3 シミュレーションネットワークのグラフ構造：BA-20-HT-C，WA-20-HT-C，WA-20-RN-C
Fig. 3 Illustration of the graph structure in the simulation: BA-20-HT-C, WA-20-HT-C, WA-20-RN-C.

の可能通信量がともに 15 となる．このように，可能通信量が増加するようにリンクの向きを調整し，リンクの向きの変更により可能通信量が増加できなくなった時点で収束する．

4. 評　　価

まず，仮想的なネットワークトポロジの上で，LBRA の計算時間を評価した．ネットワークトポロジを 4.1 節に，計算時間の結果を 4.2 節に示す．

次に，そのネットワークトポロジと典型的なトラフィックモデルを利用して，LBRA がどれほど回線の輻輳を発生させるのかをシミュレーションによって評価する．比較対象となる既存の経路制御機構のモデルを 4.4 節に示す．トラフィックモデルは，ある 1 組の 2 点間に多くのトラフィック要求が存在するモデルを 4.5 節で評価し，多数の始点終点の組にランダムなトラフィック要求を配置したモデルを 4.6 節で評価した．

4.1 ネットワークトポロジ

ネットワークトポロジは，BRITE[16] を用いてノード数 20 の仮想ネットワークを生成した．各ネットワークの名称とその設定を **表 1** に示す．グラフの生成モデルは Barabási-Albert（BA）と Waxman（WA）を利用した．各リンクに対する帯域の割当てには Constant と Uniform を用いた．Constant では，すべてのリンクの帯域幅は一定で，1000 の帯域を持つ．Uniform では，各リンクの帯域幅は 10 から 1000 の範囲で一様に分布する．一定帯域を C と示し，一様帯域を U と示す．C と U では，グラフ構造は同一である．

ノード配置は各ノードを仮想平面に配置する際のモデルである．一様に配置するモデル（Random）と，平面をセルに区切って，各セル内に配置されるノード数が Heavy Tail 分布となるモデルを利用した．BA モデルでは，Heavy Tail であるか Random であるかによってグラフ構造に変化がないため，Heavy Tail のみ調査した．WA モデルではノード間の距離が接続確率に影響するため，Heavy Tail と Random でグラフ構造が変化する．結果として生成される 3 種類のグラフ構造を，**図 3** に示す．シミュレーションの結果で BA モデルであるか WA モデルであるかに明確な相違が見られない場合は，BA モデルのみ示したが，WA モデルでも同等の結果となった．

4.2 計算時間による評価

Intel Pentium D 3.20 GHz，メモリ 4 G バイトの計算機でそれぞれのトポロジに対する LBRA の計算時間を **表 2** に示す．値は，1000 回試行した結果の平均と標準偏差である．

リンク切断に対応するためには，LBRA をオンラインで動作させる必要がある．LBRA の計算時間は 1 秒以内に収まっており，オンラインで実現するのに現実的な値を示している．

表 2 LBRA 計算時間
Table 2 Time taken by LBRA computation.

トポロジ	平均時間 (ms)	標準偏差
BA-20-HT-C	337.8	23.5
WA-20-HT-C	477.7	30.1
WA-20-RN-C	462.6	24.8

4.3 ネットワークモデル

4.1 節で述べた各トポロジにおいて，各経路制御アルゴリズムが計算した経路を利用した場合に，あるトラフィック要求における各リンクの帯域利用率を評価した．

トラフィック要求は，始点終点の行列にトラフィック量としての数値を割り当てることによって作成した．ある始点終点の組の間のトラフィックを，トラフィックフローと呼ぶ．トラフィックフローは実際には 2 地点間における多くの通信フローの集約である．始点終点間のトラフィックフローは利用される各経路に分散され，各リンクに配置される．リンクの帯域利用率は，リンクに割り当てられたトラフィックフローの帯域の総和の回線帯域幅に対する割合である．

リンク帯域利用率が 1 を超えるものは輻輳を意味するが，TCP の性能低下によるトラフィックの減少など，パケットロスによるフィードバックは考慮していない．

シミュレーションで想定するルータのトラフィック転送機能は終点への複数経路（マルチパス）をサポートしており，それぞれのノードは自分が転送するトラフィックを経路上に分割して配置できる．分割の割合は，比較対象の経路制御の場合には等分割とし，LBRA の場合は LBRA が指定する割合で分割できるものとする．

シミュレーションでは，トラフィック量を単純にこの割合で分割しリンク上に配置した．そのため，各通信パケットの遅延やジッタは考慮していない．また，パケットの到着順序も考慮していない．実際のネットワークでは，パケットの到着順序が通信のスループットに影響する場合がある．パケットの到着順序の乱れによるスループットの低下を防ぐには，パケットの始点，終点フィールドの組のハッシュ値ごとにトラフィックを分割するなどの既存の技術が有効である．

4.4 比較対象：InvCap Dijkstra

比較対象の経路計算手法は，回線の帯域幅に反比例するリンクメトリックを利用した，dijkstra アルゴリズムによる最短経路である．これを InvCap Dijkstra と呼ぶ．

回線の帯域幅は BRITE によってあらかじめ生成される．これを使い，InvCap Dijkstra ではリンクメトリックを以下の式で計算する．

$$metric = 100000/bandwidth$$

このため，一定の帯域を持つネットワークトポロジでは，すべてのリンクメトリックは同一（値 100）となる．これは，すべてのリンクメトリックを 1 に設定することと同義であり，最短ホップ経路制御と同じ結果になる．

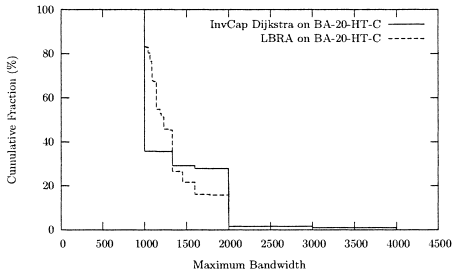

図 4 通信帯域幅とそれが可能な始点終点の組の割合：トポロジ BA-20-HT-C における InvCap Dijkstra と LBRA の比較
Fig. 4 Cumulative fraction of source-destination pair regarding maximum feasible bandwidth: comparison of InvCap Dijkstra and LBRA on a constant bandwidth BA network.

4.5 各始点終点間の最大帯域での比較

本節では，ある特定の 1 組の始点終点間に多くのトラフィックが存在するトラフィック要求を想定し，ほかにトラフィックが存在しない場合に各始点終点の組でどれだけの帯域幅の通信が可能かを比較する．

トポロジ BA-20-HT-C において InvCap Dijkstra と LBRA を利用して経路を計算した．その際のすべての始点終点間での通信可能な最大帯域幅を**図 4** に示す．通信可能な帯域幅は，特定の始点終点間にのみトラフィックを配置し，ネットワーク全体でリンク帯域利用率が 1 を超えないものを指す．配置するトラフィック量を増加させ，リンク帯域利用率が 1 となるリンクが現れた際のトラフィック量を，通信可能な最大帯域幅と呼ぶ．x 軸は通信可能な最大帯域幅を，y 軸は全始点終点の組合せのうちどれほどの割合の始点終点の組がその帯域幅で通信可能であったかを示す．InvCap Dijkstra では帯域幅 1200 のトラフィックを配置してリンク帯域利用率が 1 を超えない始点終点の組合せは全体の 35.5% であったが，LBRA においては 52.9% の始点終点の組において通信可能であった．帯域幅 1800 付近では，この関係は逆転している．LBRA では 15.7% の始点終点の組がこの帯域幅で通信可能であったが，InvCap Dijkstra では 27.9% の始点終点の組でこの帯域の通信が可能である．

トポロジ BA-20-HT-C では，リンクの帯域幅が一

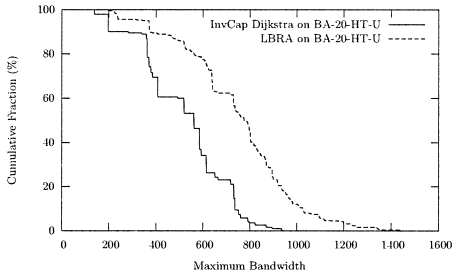

図 5 通信帯域幅とそれが可能な始点終点の組の割合：トポロジ BA-20-HT-U における InvCap Dijkstra と LBRA の比較
Fig. 5 Cumulative fraction of source-destination pair regarding maximum feasible bandwidth: comparison of InvCap Dijkstra and LBRA on a uniform bandwidth BA network.

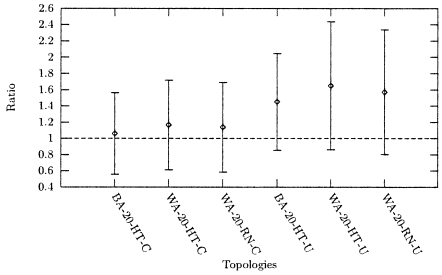

図 6 各トポロジにおける最大通信可能帯域幅の比較：InvCap Dijkstra を基準とした LBRA の増加割合
Fig. 6 Ratio between maximum feasible bandwidths of InvCap Dijkstra and LBRA: Comparison on each topology (Ratio = Max BW in LBRA/Max BW in InvCap Dijkstra).

定であるので，各リンクに割り当てられる InvCap Dijkstra のリンクのメトリックも一定となる．このため InvCap Dijkstra では，複数経路のコストが同一になる場合がランダムなコストが設定された回線よりも増え，ECMP が多く計算できる．このため，InvCap Dijkstra は LBRA と同等かそれより良い結果を得る．しかし，現実のネットワークではリンクの帯域が一定であることは稀であり，リンクメトリックを一定にすることができない状況も多く存在する．そのため，実際のネットワークでこの InvCap Dijkstra の結果が得られる場合は少ないと考えられる．

次に，トポロジ BA-20-HT-U における結果を **図 5** に示す．ネットワークのリンク帯域幅が 10 から 1000 に分散するため，InvCap Dijkstra のリンクメトリックはそれに応じて変化し，BA-20-HT-C において顕著だった ECMP は BA-20-HT-U では現れない．InvCap Dijkstra では，通信可能帯域幅が 600 となる始点終点の組は 34.2 % であったが，LBRA では 77.4%であった．始点終点の組に依存せず，LBRA は全体的に通信可能な最大帯域幅を改善している．

図 6 に，それぞれのトポロジの各始点終点での最大帯域幅の比を求めた．たとえば，同一のトポロジで同一の始点終点の組において，InvCap Dijkstra の通信可能帯域幅が 1000 であり，LBRA のそれが 1200 だった場合，LBRA の帯域幅を InvCap Dijkstra のもので割り，結果の比を 1.2 とする．始点終点の組によってこれらの値は変化するので，平均と標準偏差を求めた．

比較結果は，リンクの帯域が一定（C）か一様（U）かによって変化しているのが分かる．この理由は，リンクメトリックが一定かつリンクの帯域幅が一定である場合には ECMP が最もよく働くためである．トポロジの生成モデルなどには相関は見られない．帯域が一様（U）である場合には，LBRA は平均で性能を 1.45 から 1.65 ほど改善している．

4.6 ランダムなトラフィック要求のモデルに対する比較

多地点間にランダムな値のトラフィック量が存在するトラフィック要求に対してリンク帯域利用率を調査するために，Fortz ら[9),10)] のトラフィックモデル（fortz-model）を利用した．fortz-model はノード間の距離とホットスポットノードを考慮した，ランダムなモデルである．fortz-model によってトラフィック要求を 1000 個生成し，その各トラフィック要求を BA-20-HT-C と BA-20-HT-U の 2 つのトポロジに対してそれぞれ適用し，全リンク中最大となった帯域利用率を観測した．BA-20-HT-C に対する観測結果を **図 7** に，BA-20-HT-U に対する観測結果を **図 8** に示す．それぞれの図では，左の図が散布図であり，ある 1 つのトラフィック要求に対して x 軸が InvCap Dijkstra でのリンク帯域利用率の最大値，y 軸が LBRA での最大値を表している．つまり，左上にプロットされた点は，InvCap Dijkstra ではリンク帯域利用率の最大値が低いにもかかわらず，LBRA ではリンク帯域利用率の最大値が高い場合を指す．右の図は最大リンク帯域利用率の全トラフィック要求に対する累積度数分布である．

図 7 はリンク帯域が一定の場合であるが，LBRA はつねに InvCap Dijkstra より高いリンク利用率を示している．つまり，多くのリンク帯域を消費しており，帯域利用効率が悪い．これは，一定帯域かつリンクメトリックが一定のトポロジでは ECMP が多く計算されるために，InvCap Dijkstra の性能が良いというのが第 1 の理由である．

もう 1 つの理由は，LBRA の性質が原因である．LBRA は，自分に関係するトラフィックをネットワー

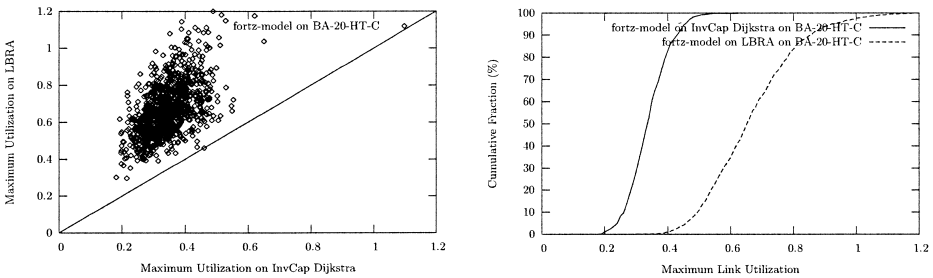

図 7 fortz-model トラフィックにおける最大リンク帯域利用率：トポロジ BA-20-HT-C における InvCap Dijkstra と LBRA の比較
Fig. 7 Comparison of maximum link utilizations of the InvCap Dijkstra and LBRA on a constant bandwidth BA network.

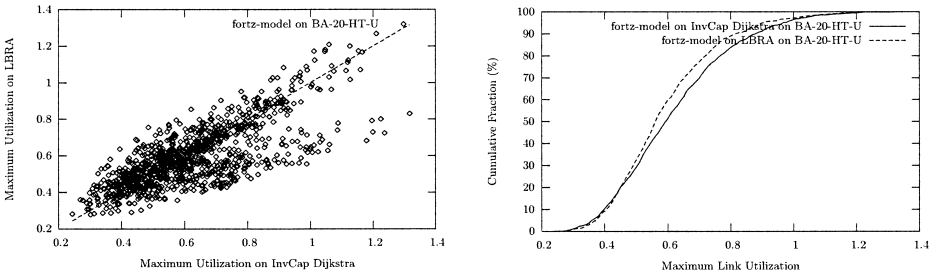

図 8 fortz-model トラフィックにおける最大リンク帯域利用率：トポロジ BA-20-HT-U における InvCap Dijkstra と LBRA の比較
Fig. 8 Comparison of maximum link utilizations of the InvCap Dijkstra and LBRA on a uniform bandwidth BA network.

ク全体に分散するような経路を計算するので，すべてのトラフィックがネットワーク上に分散される．そのため，1 つのリンクに着目すると，多くのトラフィックがそのリンクの帯域を消費しようとする．最短経路制御であれば，遠回りとなるためそのリンクを利用しなかったようなトラフィックフローも，LBRA ではいくらかのトラフィックに分割され，そのリンクの帯域消費に貢献する．このようにして，複数経路を利用した負荷分散を積極的に行い，同量のトラフィックに対して最短経路利用時より多くの帯域を消費する．このような観点では，LBRA は帯域の利用効率が悪い．

これらの理由から，図 7 左側の散布図では，InvCap Dijkstra（x 軸）が最大リンク利用率を 0.2 から 0.6 に収めているのに対して，LBRA（y 軸）は 0.3 から 1.2 にまで広がっている．図 7 右側の累積度数分布では，InvCap Dijkstra はトラフィック要求の 99.8%以上を 0.6 以下の最大リンク帯域利用率で抑えているのに対し，LBRA では最大リンク帯域利用率を 0.6 以下に抑えられたのはトラフィック要求の試行のうち 34.7%のみであった．InvCap Dijkstra が効率的にトラフィックを配送している理由は，4.5 節と同様に，すべてのリンクメトリックが同一であるために ECMP が多く計算されるからである．

次に，一様に分布したリンク帯域のトポロジでの評価を 図 8 に示す．ここでは，LBRA は InvCap Dijkstra と同等かそれ以上の帯域利用効率となった．InvCap Dijkstra が最大リンク利用率を LBRA より低く抑えたのはトラフィック要求の試行全体の 48.5%であり，LBRA が InvCap を下回ったのは 51.5%であった．図 8 右側の累積分布からも，両者の帯域利用効率はほぼ同等であることが分かる．

これらの評価から，リンク帯域が一定でリンクメトリックが同一であるような特異なネットワークでは，LBRA は既存の InvCap Dijkstra 経路制御よりリンク帯域利用効率が大幅に悪化する．しかし，リンク帯域が一様に分布したトポロジでは，多地点間の多数の通信からなるランダムモデルのトラフィック要求においても，LBRA は InvCap Dijkstra と同等以上の帯域利用効率を実現することが分かった．

5. ま と め

本稿では，ネットワークのある 2 点間のトラフィック量が飛びぬけて多い，または同じトラフィックの始点終点となる 2 点間が異なる（変化する）などの広範

なトラフィック要求に対応する，新しい経路制御アルゴリズム LBRA を示した．シミュレーションによって，リンクメトリックが帯域と反比例の値に設定した既存の最短経路制御と比較評価した．シミュレーションは複数の仮想ネットワーク上で，全始点終点間の最大流量を調べるものと，ランダムなトラフィックモデルを適用するものを行った．

LBRA は，ネットワーク上の多くの始点終点の組に対して，既存の経路制御機構に比べ輻輳しにくい経路を提供する．ネットワーク上のある 1 組の始点終点間のトラフィック量が極端に高いトラフィック要求に対して，良好な通信性能を提供する．ある時刻で存在する極端に高いトラフィックを持つ始点終点の組が 1 組であれば，多くの始点終点間に対応できる，始点終点の組が変化しても対応できるという点から，既存の経路制御機構より広範なトラフィック要求に対応する．

しかし，LBRA はトラフィックによる負荷を分割し一部のトラフィックを遠回りの経路で配送しようとするため，各始点終点間のトラフィックが最短経路以外の多くのリンクに搭載される傾向にある．そのため，ネットワーク上に同程度の帯域のトラフィックが多数存在する場合，既存の経路制御機構に比べ帯域利用効率が悪い．特に，回線の帯域がすべて同一であるネットワークで，リンクメトリックをすべて同一に設定した場合の既存の最短経路制御に比べて，LBRA は容易に回線を輻輳させる．

しかし，回線の帯域幅が一様に分布するようなネットワークでは，同程度の帯域のトラフィック要求が始点終点間に多数存在しても，既存の経路制御手法と同等以上の性能を提供する．具体的には，ランダムなトラフィック要求を発生させた際に回線が輻輳する確率は，ほぼ同等以下となった．

本研究の今後の課題として，より多角的な評価がある．本研究の評価は，各シミュレーション結果において，最大の帯域利用率を持つ 1 本の回線にだけ着目している．今後，トラフィックによる負荷の平滑化などの観点から評価する必要がある．また，トポロジの種類とトラフィック要求の種類に対する相関も調査する予定である．さらに，LBRA を実際のネットワークで利用するには分散計算として実行するのが適しているが，その手法とプロトコルの設計は今後の課題である．

参 考 文 献

1) Awduche, D., Chiu, A., Elwalid, A., Widjaja, I. and Xiao, X.: Overview and Principles of Internet Traffic Engineering, RFC 3272 (2002).
2) Xiao, X., Hannan, A., Bailey, B. and Ni, L.M.: Traffic Engineering with MPLS in theInternet, *IEEE network* (2000).
3) Kandula, S., Katabi, D., Davie, B. and Charny, A.: Walking the tightrope: responsive yet stable traffic engineering, *SIGCOMM Comput. Commun. Rev.*, Vol.35, No.4, pp.253–264 (2005).
4) Rekhter, Y., Li, T. and Hares, S.: A Border Gateway Protocol 4 (BGP-4), RFC 4271, IETF (2006).
5) Rosen, E., Viswanathan, A. and Callon, R.: Multiprotocol Label Switching Architecture, RFC 3031, IETF (2001).
6) Khanna, A. and Zinky, J.: The Revised ARPANET Routing Metric, *SIGCOMM*, pp.45–56 (1989).
7) Anderson, E.J. and Anderson, T.E.: On the Stability of Adaptive Routing in the Presence of Congestion Control, *INFOCOM '03* (2003).
8) 滝根哲哉，伊藤大雄，西尾章治郎：ネットワーク設計理論，岩波書店 (2001).
9) Fortz, B. and Thorup, M.: Internet traffic engineering by optimizing OSPF weights, *IEEE INFOCOM*, Vol.2, pp.519–528 (2000).
10) Fortz, B. and Thorup, M.: Optimizing OSPF/IS-IS Weights in a Changing World, *IEEE J. Selected Areas in Communications*, Vol.20, No.4, pp.756–767 (2002).
11) Applegate, D. and Cohen, E.: Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs, *SIGCOMM '03: Proc. 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, New York, NY, USA, pp.313–324, ACM Press (2003).
12) Elwalid, A., Jin, C., Low, S. and Widjaja, I.: MATE: Multipath adaptive traffic engineering, *Comput. Networks*, Vol.40, No.6, pp.695–709 (2002).
13) Basu, A., Lin, A. and Ramanathan, S.: Routing using potentials: A dynamic traffic-aware routing algorithm, *SIGCOMM '03: Proc. 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, New York, NY, USA, pp.37–48, ACM Press (2003).
14) Vutukury, S. and Garcia-Luna-Aceves, J.J.: A simple approximation to minimum-delay routing, *SIGCOMM '99: Proc. conference on Applications, technologies, architectures, and protocols for computer communication*, New York, NY, USA, pp.227–238, ACM Press (1999).

15) Vutukury, S. and Garcia-Luna-Aceves, J.J.: MDVA: A Distance-Vector Multipath Routing Protocol, *INFOCOM*, pp.557–564 (2001).
16) Medina, A., Lakhina, A., Matta, I., and Byers, J.: BRITE: An Approach to Universal Topology Generation, *International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems- MASCOTS '01* (2001).

付　　録

A.1 アルゴリズム

対象とするネットワークをグラフ $G=(V,E)$ として扱う．V はノード集合，E はリンクの集合であり，関数 cap は $cap((i,j))$ でリンク (i,j) の帯域幅を返す．N^i はノード i の隣接ノード集合を示す．

ノード i における終点 j への経路表エントリは $\{(x_1,r_1),(x_2,r_2),...,(x_k,r_k)\}$ となる．これを $R^i_j=\{(x,r)|x\in N^i\}$ と表す．ここで，x はネクストホップとなるノードであり，r はネクストホップノード x へのトラフィック分割割合を表す．ノード i における終点 j へのネクストホップノードの集合を $S^i_j(\subseteq N^i)$ と表す．LBRA アルゴリズムは，グラフ構造 G を入力とし，終点ごとにリンクにパケット転送の向きを与えた有向グラフを計算する．そして，この有向グラフを利用して，全ノードの経路表 $R=\{R^i_j|i,j\in V\}$ を求める．

LBRA は，経路ループの回避にパスベクタ型アルゴリズムを利用する．ノード i から終点ノード j に到達するための中継ノード集合を P^i_j，ノード i から終点ノード j に対して現在可能な最大通信量を b^i_j と表す．$W=\{S^i_j|i,j\in V\}$，$Q=\{P^i_j|i,j\in V\}$，$B=\{b^i_j|i,j\in V\}$ と決める．リンク容量および可能通信量は非負の整数である．

```
1: procedure INITIALIZE(G, W, Q, B, R)
2:     for i ∈ V do
3:         for j ∈ V do
4:             R^i_j ← φ
5:             S^i_j ← φ
6:             P^i_j ← φ
7:             b^i_j ← 0
8:         done
9:     done
10: end procedure
```

```
1: procedure CALCBW(i, j, G, S^i_j, Q, B, R)
2:     if (i = j) then
3:         return ∞
4:     end if
5:     for each nexthop x ∈ S^i_j do
6:         if (i ∈ P^x_j) then
7:             return 0
8:         end if
9:     done
10:    bw ← Σ_{x∈S^i_j} {min{cap(i,x), b^x_j}}
11:    for each common node c ∈ ⋂ P^x_j do
12:        bw ← min{bw, b^c_j}
13:    done
14:    for each neighbor x ∈ S^i_j do
15:        bw ← max{bw, min{cap(i,x), b^x_j}}
16:    done
17:    return bw
18: end procedure
```

```
1: procedure CALCNODELIST(i, j, G, S^i_j, Q, B, R)
2:     P' ← {i} ∪ {⋃_{x∈S^i_j} P^x_j}
3:     return P'
4: end procedure
```

```
1: procedure UPDATE(i, j, G, S^i_j, Q, B, R)
2:     b^i_j ← CALCBW (i, j, G, S^i_j, Q, B, R)
3:     P^i_j ← CALCNODELIST (i, j, G, S^i_j, Q, B, R)
4:     R^i_j ← φ
5:     for k ∈ S^i_j do
6:         r ← b^k_j / (Σ_{x∈S^i_j} b^x_j)
7:         R^i_j ← R^i_j ∪ (k, r)
8:     done
9: end procedure
```

```
1: procedure DECIDE(e, j, G, W, Q, B, R)
2:     (i, k) ← e
3:     S^i_j ← S^i_j \ {k}
4:     UPDATE (i, j, G, S^i_j, Q, B, R)
5:     bi' ← CALCBW (i, j, G, S^i_j ∪ {k}, Q, B, R)
6:     δi ← bi' − b^i_j
7:     S^k_j ← S^k_j \ {i}
8:     UPDATE (k, j, G, S^k_j, Q, B, R)
9:     bk' ← CALCBW (k, j, G, S^k_j ∪ {i}, Q, B, R)
10:    δk ← bk' − b^k_j
11:    if ((bi' = 0) ∧ (bk' = 0)) then
12:        return
13:    end if
14:    if ((δi ≦ 0) ∧ (δk > 0)) then
```

```
        S_j^k ← S_j^k ∪ {i}
    else if ((δi > 0) ∧ (δk ≦ 0)) then
        S_j^i ← S_j^i ∪ {k}
    else if (b_j^i < b_j^k) then
        S_j^i ← S_j^i ∪ {k}
    else if (b_j^i > b_j^k) then
        S_j^k ← S_j^k ∪ {i}
    else if (δi > δk) then
        S_j^i ← S_j^i ∪ {k}
    else if (δi < δk) then
        S_j^k ← S_j^k ∪ {i}
    else if ((nodeid(j) ≡ 0 (mod 2)) then
        if (nodeid(i) < nodeid(k)) then
            S_j^i ← S_j^i ∪ {k}
        else if (nodeid(i) > nodeid(k)) then
            S_j^k ← S_j^k ∪ {i}
        end if
    else if ((nodeid(j) ≡ 1 (mod 2)) then
        if (nodeid(i) > nodeid(k)) then
            S_j^i ← S_j^i ∪ {k}
        else if (nodeid(i) < nodeid(k)) then
            S_j^k ← S_j^k ∪ {i}
        end if
    end if
    UPDATE (i, j, G, S_j^i, Q, B, R)
    UPDATE (k, j, G, S_j^k, Q, B, R)
end procedure
procedure LBRA(G)
    INITIALIZE (G, W^0, Q^0, B^0, R)
    for each destination j ∈ V do
        X ← {j}
        T ← φ
        h ← 1
        W^h ← W^{h-1}
        k ← 1
        Q^k ← Q^{k-1}
        B^k ← B^{k-1}
        while T ≠ E do
            e ← {(u,v)|u ∈ X, (u,v) ∈ E, (u,v) ∉ T}
            DECIDE(e, j, G, W^h, Q^k, B^k, R)
            X ← X ∪ {u,v}
            T ← T ∪ {e}
            while W^h ≠ W^{h-1} do
                h ← h + 1
                W^h ← W^{h-1}
                for each link e' ∈ T do
                    DECIDE(e', j, G, W^h, Q^k, B^k, R)
                    if (W^h ≠ W^{h-1}) then
                        while (Q^k ≠ Q^{k-1}) ∨ (B^k ≠ B^{k-1}) do
                            k ← k + 1
                            Q^k ← Q^{k-1}
                            B^k ← B^{k-1}
                            for each link (u', v') ∈ T do
                                UPDATE (u', j, G, W^h, Q^k, B^k, R)
                                UPDATE (v', j, G, W^h, Q^k, B^k, R)
                            done
                        end while
                        ( A_j check )
                    end if
                done
            end while
        end while
    done
end procedure
```

A.2 部分正当性

これ以降，ある互いに隣接するノード u, v ($v \in N^u$) および終点 j について，u のネクストホップ S_j^u に v が含まれるとき，リンク (u, v) は $u \to v$ の向きが与えられた，と表現する．このリンクの向きは，パケットが終点に向けて転送される方向と同等である．この有向リンクが閉路を構成する場合は，経路制御ループを意味する．

任意のグラフが与えられたとき，LBRA は各終点に対してすべてのリンクを利用する非ループの複数経路を計算する．アルゴリズムの各ステップで計算される経路がループを持たないこと，アルゴリズムが停止した場合にはすべてのリンクに向きが与えられていることを証明し，LBRA の部分正当性を示す．

定理 A.2.1. 任意のグラフ $G = (V, E)$，リンク容量 $cap(e)$，終点 $j \in V$ に対して，アルゴリズムの各ステップで，*LBRA* は経路制御ループを持たない．

証明. 背理法により証明する．

グラフ $G = (V, E)$ について，終点 j について経路制御ループが存在すると仮定する．経路制御ループのうち最後に向きが決定されたリンクを (u, v) とする．

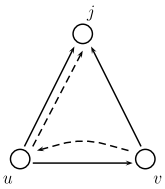

図 9 経路制御ループの概念図
Fig. 9 A routing loop.

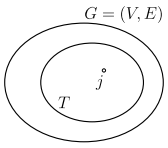

図 10 LBRA アルゴリズムの G, T, 終点 j の関係
Fig. 10 Relation between G, T and destination j in LBRA.

向きは $u \to v$ であるとして一般性を失わない．経路制御ループの仮定から，v は j に到達するために u を経由する（**図 9**）ので，中継ノードリスト P_j^v は u を含む．アルゴリズムから，CalcBw$(u, j, G', ...)$ はネクストホップ v の P_j^v が u を含むため 0 を返す．$i \leftarrow u, k \leftarrow v$ としたときの δi は $bi' = 0$ から 0 以下になるはずである．その場合は $S_j^k \leftarrow S_j^k \cup \{i\}$ つまり $S_j^v \leftarrow S_j^v \cup \{u\}$ となるはずであり，リンク (u, v) の向きが $u \to v$ と決定されたことに矛盾する． □

LBRA が全リンクに向きを与えることを証明するために，まず補題として，可能通信量 0 のノードはネクストホップ集合に追加されないこと，アルゴリズムの各ステップで一度向きが与えられたリンクから向きがなくなることがないことを示す．

補題 A.2.2. 可能通信量 *0* のノードはネクストホップ集合には追加されない．

証明. Decide() において，ノード k の可能通信量が 0 であれば，i が終点 j に対して k をネクストホップとした際の可能通信量の増加分 δi は 0 になるはずである．アルゴリズムから，$\delta i = 0$ の場合には $S_j^i \leftarrow S_j^i \cup \{k\}$ とならない． □

補題 A.2.3. アルゴリズムの各ステップで，一度向きが与えられたリンクから向きがなくなることはない．

証明. アルゴリズムから，Decide () がリンクに向きを与えないのは，どちらの向きに決定してもノードの可能通信量が双方とも 0 となる場合のみである．CalcBw () によって計算されるノードの可能通信量が 0 となるのは，i のネクストホップ x の中継ノードリスト $P_j^x (\in Q)$ がそのノード自身 (i) を含む場合（経路制御ループとなる場合）か，i のすべてのネクストホップの可能通信量の総和 $\sum_{x \in S_j^i} b_j^x$ が 0 である場合のどちらかとなる．グラフの中で向きが決定されていたリンクのうち，初めて向きがなくなったリンクを (u, v) であると仮定し，矛盾を導く．

ノード u, v が互いを中継ノードリストに含む場合は可能通信量が双方とも 0 になる．しかしこれは経路制御ループであり，定理 A.2.1 に矛盾する．

v の中継ノードリスト P_j^v に u が含まれ，u のすべてのネクストホップの可能通信量の総和 $\sum_{x \in S_j^u} b_j^x$ が 0 となる場合も，可能通信量が双方とも 0 となる．この場合は，u のすべてのネクストホップの可能通信量の総和が 0 であるため，u の可能通信量自体も 0 となり，v のネクストホップ集合 S_j^v が可能通信量 0 のノード u を含むことが補題 A.2.2 に矛盾する．

残る場合は u と v のネクストホップの可能通信量の総和がどちらも 0 になる場合である．

一度リンクの向きが決められたノード u, v のネクストホップ集合の和 $S_j^u \cup S_j^v$ は，u, v 以外のノードを含む．なぜならば，u, v のネクストホップ集合に u, v 以外のノードが含まれないのは，経路制御ループであるか，u, v からそれ以外のノードへの到達性がない状態である．u, v からそれ以外のノードへの到達性がないことは，グラフの中で他に向きのないリンクがあることを示し，初めて向きがなくなったリンクが (u, v) であるという仮定に反する．

ノード u, v のネクストホップ集合の和 $S_j^u \cup S_j^v$ に u, v 以外のノードが含まれることは，可能通信量 0 のノードはネクストホップ集合に追加されないこと（補題 A.2.2）から，$S_j^u \cup S_j^v$ に含まれる u, v 以外のすべてのノードの可能通信量の和が 0 であるという仮定に矛盾する． □

定理 A.2.4. 任意のグラフ $G = (V, E)$，リンク容量 $cap(e)$，終点 $j \in V$ に対して，*LBRA* は全リンクに向きを与える

証明. 計算途中の G, T, j の関係の概念図を**図 10** に示す．アルゴリズムから，リンク集合 T に属するリンクには，一度向きが与えられる．補題 A.2.3 から，リンクの向きは消えない．アルゴリズムが停止する条件 $T = E$ から，アルゴリズムの停止時にはすべてのリンクに向きが与えられる． □

定理 A.2.1 により LBRA の計算結果となる経路が経路制御ループを含まないこと，定理 A.2.4 によりすべてのリンクに向きが計算できることが示された．この 2 つにより，LBRA の部分正当性が示された．

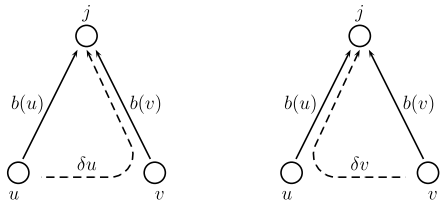

図 11 リンクの向きの変更
Fig. 11 Changing routing direction between u and v.

A.3 停 止 性

まず LBRA の停止性を示すための補題を示す.

補題 A.3.1. *LBRA* がリンクの向きを変更する場合, リンクに接続する 2 つのノードの可能通信量の最小値がかならず増加する.

証明. リンク (u,v) について, $u \to v$ の向きが $u \leftarrow v$ に変更され, ノード u, v の可能通信量の最小値が, リンクの向き変更の前後で増加しなかったと仮定する.

リンクの向きが変更される直前の状態において, ノード u, v がリンク (u,v) を利用せずに終点 j に到達する際の可能通信量をそれぞれ $b(u)$, $b(v)$ とする (**図 11**). ノード u, v がリンク (u,v) を利用することによって増加させる可能通信量を δu, δv とするとき, ノード u の可能通信量は $b(u)+\delta u$ から $b(u)$ に, ノード v の可能通信量は $b(v)$ から $b(v)+\delta v$ に変化する.

リンクの向きが $u \to v$ から $u \leftarrow v$ に変更されたことから $b(u) > b(v)$ である. リンクに接続するノードの可能通信量の最小値がノード u であったと仮定すると, $b(u)+\delta u < b(v)$ となるが, $b(u) > b(v)$ と組み合わせると $b(u)+\delta u < b(v) < b(u)$ となり, 可能通信量の増加分 δu が非負の整数であることに矛盾する. ここから, リンク (u,v) の向きが変更される前の可能通信量の最小値はノード v であり, $b(u)+\delta u > b(v)$ を得る.

リンクの向きが $u \leftarrow v$ に変更された後は, ノード u の可能通信量は $b(u)$, ノード v の可能通信量は $b(v)+\delta v$ となる. リンクの向きが変更されたことから $b(u) > b(v)$, δv が非負の整数であることから $b(v)+\delta v > b(v)$ であるため, どちらが最小の場合も, 可能通信量の最小値は以前の値 $b(v)$ から増加する. □

次に, アルゴリズムが停止することを示す.

定理 A.3.2. 任意のグラフ $G=(V,E)$, リンク容量 $cap(e)$, 終点 $j \in V$ に対して, *LBRA* は停止する.

証明. ある量を定義し, アルゴリズムの各ステップでその量がかならず減少すること, 量が無限に減少せず下限があることから, アルゴリズムの停止性が示せる. 終点 j に対して, 量 A_j を次のように定義する.

$$A_j = (N^0, N^a, N^b, \ldots, N^z, N^\infty)$$

ここで, N^k は, 可能通信量 k を持つノードの個数である. 可能なすべての可能通信量の値に対する N^k を組にしたものが, A_j となる. 初期状態は, $A_j = (|V|-1, 0, 0, \ldots, 0, 1)$ となり, 終点ノード j 自身のみが終点に対して ∞ の可能通信量を持つ.

N^a, N^b の a や b は, 考えられる可能通信量の値すべてを並べたものとなる. これは, 最大でリンク容量の全組合せとなる.

補題 A.3.1 から, アルゴリズムの各ステップでリンクの向きが変更されたときは必ず, 変更の前後でリンクに接続する 2 つのノードのうち可能通信量が最小のものの値を増加させる. これを A_j として考えると, ノードの数は 2 つ移動する可能性があるが, 移動するもののうち可能通信量が最小のもの (つまり, 最も左側で移動するもの) は必ず右側に移動する. つまり, 辞書式順序として必ず減少する. A_j はアルゴリズム 31 行目で確認できる.

A_j は無限に減少しない. 終点を除いて, ノードの可能通信量がとりうる最大値は $\sum_{e \in E} cap(e)$ を超えない. このため, LBRA は停止する. □

A.4 計 算 量

LBRA が題材としているのは, 全始点から 1 つの終点への帯域の最小値を最大化するような非循環有向グラフ (DAG: Directed Acyclic Graph) の計算である. LBRA の計算量は任意のグラフから作成できる DAG の数などに制限されるが, これは未知である. 我々の知る限り LBRA の計算には最悪すべてのリンクの向きの場合の数を考慮する必要があり, 計算量は指数関数オーダ, $O(2^{|E|})$ となる. LBRA は計算の途中ステップにも循環 (経路制御ループ) を許さず, 発見的手法で帯域の最大化を目指す貪欲アルゴリズムといえる.

(平成 18 年 7 月 7 日受付)
(平成 19 年 1 月 9 日採録)

小原 泰弘

2001 年慶應義塾大学大学院政策・メディア研究科修士課程を修了し, 同年より同大学院政策・メディア研究科後期博士課程所属. 研究分野はインターネットルーティング.

今泉　英明（正会員）

2005 年慶應義塾大学大学院政策・メディア研究科後期博士課程を修了し，同年博士（政策・メディア）を取得．2005 年 4 月より東京大学大学院情報理工学系研究科特任助手を経て，2006 年 4 月より同大学院新領域創成科学研究科特任助手，同年 11 月より同大学院工学系研究科特任助手．研究分野は超高帯域高速データリンクを扱う次世代ルータ技術および通信品質保証技術．

加藤　　朗（正会員）

東京工業大学大学院理工学研究科博士後期課程を単位取得退学．1990 年慶應義塾大学環境情報学部助手，1994 年東京大学大型計算機センター助手を経て，2002 年から同情報基盤センター助教授．大学院時代から計算機ネットワークの研究・運用に従事し，WIDE Project の設立に参画．博士（政策・メディア）．

中村　　修（正会員）

1982 年慶應義塾大学工学部数理工学科卒業．1984 年同大学院理工学研究科修士課程数理工学専攻修了．1993 年同大学院より博士（工学）取得．1993 年 9 月より同大学環境情報学部助手として就任．2000 年 4 月より同大学環境情報学部助教授，2006 年 4 月より同大学環境情報学部教授．

村井　　純（フェロー）

1955 年生．1984 年慶應義塾大学大学院工学研究科博士課程修了．1987 年博士号取得．1984 年東京工業大学総合情報処理センター助手，1987 年東京大学大型計算機センター助手．1990 年慶應義塾大学環境情報学部助教授を経て 1997 年より同教授．2005 年 5 月より学校法人慶應義塾常任理事．1984 年 JUNET を設立．1988 年 WIDE プロジェクトを設立し，今日までその代表として指導にあたる．2005 年 Internet Society より Jonathan B. Postel Service Award 受賞，社団法人情報処理学会フェロー，日本学術会議第 20 期会員．著書『インターネット』,『インターネット II』（岩波新書）,『インターネットの不思議、探検隊！』（太郎次郎社エディタス）．