

## 複数経路通信における経路制御に関する検討

穂坂 怜†

相田 仁†

†東京大学大学院 新領域創成科学研究科

## 1 はじめに

近年インターネットは社会基盤となり、多くの重要な通信がその上で行われている。通信の信頼性の向上は非常に重要な課題である。ネットワーク上での通信の信頼性を向上させる技術の1つに、物理的に別(disjoint)のパスを利用する方法がある。本稿では代表的な disjoint なパスの探索手法と同等なパスを探索する手法を提案する。その上で、適切な disjoint なパスへパケットを流す手法について検討を行う。

## 2 disjoint パスの探索手法

disjoint パス探索とは、特定のノード間でノードやリンクを共有しないようなパスの組を探すことである。ノードを共有しないことを node-disjoint、リンクを共有しないことを link-disjoint と呼ぶ。本稿では disjoint は node-disjoint を指し、パスの本数は2本とする。

## 2.1 従来手法

これまでに disjoint なパス対を探索する様々なアルゴリズムが提案されている[1]。それらは求める最適性の違いにより次の3つに大別される。「パス対のメトリックの和の最小化(以下、MinSum法)」、「長い方のパスの最小化(以下、MinLong法)」、「短い方のパスを最小化(以下、2-Dijkstra法)」である。MinSum法は多くの研究がなされており  $O(2N^2)$  の計算量で解ける多項式時間アルゴリズムが提案されている。ここで  $N$  はノード数である。MinLong法もまた多くの研究がされているがNP困難であることが知られている。

disjoint なパス対を探す最も単純なアルゴリズムはエンドノード間の最短路を含む2-Dijkstra法である。1本目のパスに最短路を選択し、それに用いたノード及びパスを対象となるネットワークから全て取り除き、そのグラフ上で再び最短路を選択し2本目とする。上に挙げた他の2つの手法に比べ、Dijkstraの最短路アルゴリズム( $O(M + N \log N)$ ,  $M$  はリンク数)を用いるため高速に計算が可能である。

一方で、2-Dijkstra法は大きな2つの問題を抱えている。1つ目は、探索された disjoint なパス対の最適性

の甘さである。MinSum法、MinLong法がパス対全体のバランスを考慮しているのに対し、2-Dijkstra法は1本目の最適性しか考慮していない。2つ目の問題は、1本目のパスに必ず最短路を選択してしまうために、1本目のパスと disjoint な2本目のパスを選択できない場合があることである。2-Dijkstra法では1本目のパスで用いられるノード及びリンクを削除することで、エンドノード間が非連結となり2本目が選択できない場合がある。2つ目の問題を回避する手法を2.2で提案する。

## 2.2 Hybrid-Dijkstraの提案

本手法は高速なDijkstraの最短路アルゴリズムを用いつつ、2-Dijkstraとは別の disjoint なパス対を探索する。以下の手順で disjoint なパス対を選択する。1) 最短路に用いられるリンクをグラフから削除する。2) そのグラフにおいて最短路を探索し1本目のパスとする。3) 元のグラフから1本目のパスに用いられるノード及び関連リンクを削除し、そのグラフでの最短路を2本目とする。2-Dijkstra法とこの手法を補完的に用いる(以下、Hybrid-Dijkstra法)ことで、2本目が選択できない事態の発生を抑えることができる。Hybrid-Dijkstra法はDijkstraの最短路アルゴリズム( $O(M + N \log N)$ )に比例した計算量に収まり、一般のOSPFドメインに属するノード数が40から100であることを考慮するとMinSum法の計算量( $O(2N^2)$ )よりも少なく済むと考えられる。

## 2.3 3手法の比較

仮想的なネットワークトポロジ上でMinSum法、2-Dijkstra法、Hybrid-Dijkstra法の各手法により探索されるパス対を比較する。

## 2.3.1 比較の条件

ネットワークトポロジはBRITEを用いてノード数20の仮想ネットワーク(以下、BA20)を生成した。グラフの生成モデルはBarabasi-Albert(BA)を利用した。各リンクに対する帯域の割り当ては10から1000の範囲で一様に分布するUniformを用いた。生成されたグラフ構造を図1に示す。リンクの帯域幅に反比例するリンクメトリックを利用した。リンクの帯域幅は

Routing control for multi-path communication

†Satoshi HOSAKA †Hitoshi AIDA

†Graduate School of Frontier Sciences, The University of Tokyo

BRITEによってあらかじめ生成される．各リンクのリンクメトリックを以下の式で計算する．

$$metric = 100000/bandwidth$$

### 2.3.2 コストの和ならびに長い方のコストの比較

生成したトポロジ BA20 において全点間で MinSum 法，2-Dijkstra 法，Hybrid-Dijkstra 法で disjoint なパス対を選択するシミュレーションを行った．各手法でのパス対のコスト和ならびに長い方のコストの累積度数分布を図 2 に示す．Hybrid-Dijkstra 法が 2-Dijkstra 法と比較して同等以上のパス対を選択できていることが分かる．また，コスト和，長い方のコストともに Hybrid-Dijkstra 法は MinSum 法と同等の disjoint なパス対を選択できていることが分かる．

### 2.4 経路探索法の提案

2.3.2 で得られた結果から経路探索法を提案する．現在のインターネットにおいて最短路以外にパケットを流すには工夫が必要となる．2-Dijkstra 法のように disjoint なパス対に最短路が含まれている場合，そのパスへは容易にパケットを流すことができる．また，2.3.2 で述べたように 2-Dijkstra 法および Hybrid-Dijkstra で発見されるパス対は MinSum 法のそれと同等の最適性を持つ．これらのことからまず 2-Dijkstra 法でパス対を探索するのが望ましいと考えられる．2-Dijkstra 法で 2 本目が見つからなかった場合は，次に計算量の少ない Hybrid-Dijkstra 法，それでも見つからない場合は MinSum 法で disjoint なパス対を探索するのが妥当であると言える．

## 3 disjoint パスへのルーティングの検討

OSPF のようなリンクステート型のルーティングプロトコルでは最短路を最適なパスとする．そのため，2.4 で述べたように探索した disjoint なパス対へパケットをルーティングして欲しい場合は工夫が必要となる．古くからその手法として挙げられるのがソースルーティングである．しかし，ソースルーティングはルータへの負荷が問題視されており，所望パスのノード全てを指定することは望ましくない．また，loose ソースルーティングと全てのノードを指定する strict ソースルーティングを組み合わせる必要があるケースも存在することが分かっている．これらのことを踏まえ，経由地の指定の方法ならびに指定ノード数を最小化する方法について検討を進めている．

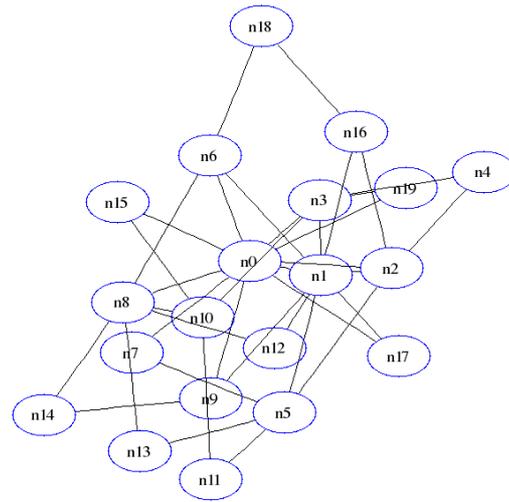


図 1: BA20 のグラフ構造 (各リンクの帯域幅は 10 から 1000 の間で一様に分布)

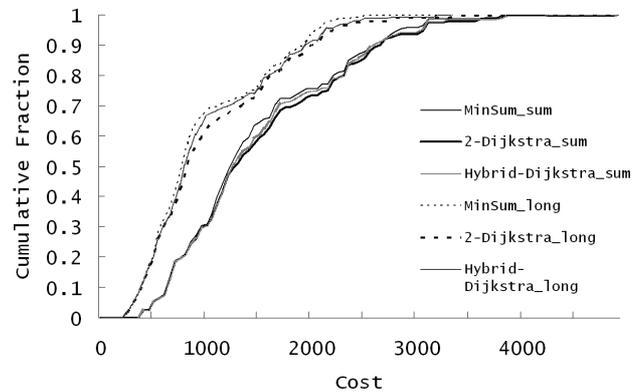


図 2: トポロジ BA20 における各手法でのパス対の和ならびに長い方のコストの比較

## 4 まとめ

2-Dijkstra 法の 2 本目のパスが取れない問題を改善した Hybrid-Dijkstra 法を提案し，提案手法が MinSum 法と比較してパス対のコストの和ならびに長い方のパスのコストにおいて同等の disjoint なパス対を選択することを示した．そして，探索したパス対にパケットを流す方法としてソースルーティングを用いた手法について述べた．経由地の指定方法，指定ノード数を最小化する方法について引き続き検討していく予定である．

## 参考文献

- [1] Randeep Bhatia, Murali Kodialam, and T. V. Lakshman, "Finding disjoint paths with related path costs," *Journal of Combinatorial Optimization*, Volume 12, Numbers 1-2, pp83-96, September, 2006.