

## 第二のルーティングテーブルを用いた柔軟な負荷分散手法

山本 徹<sup>†</sup> 吉廣 卓哉<sup>‡</sup>

<sup>†</sup>和歌山大学大学院システム工学研究科

<sup>‡</sup>和歌山大学システム工学部

〒640-8510 和歌山県和歌山市栄谷 930

E-mail: <sup>†,‡</sup>{s051058,tac}@sys.wakayama-u.ac.jp

あらまし 現在インターネットで使用されているルーティングプロトコルでは、ネットワークの管理者がリンクに対して設定したコスト（メトリック）を基に計算された最短パスを用いてパケット転送が行われるため、現状では輻輳が特定の部分に偏りやすく、ネットワーク資源を有効に活用することが難しい。そこで、本研究ではリンクステート型 IGP ルーティングプロトコルを拡張し、各ルータが第二のルーティングテーブルを持つことで、どのルータからも 2 本の転送経路を利用して負荷分散を行う枠組みを提案する。これにより、各ルータは自分で計測した負荷情報を用いて、各宛先に対する 2 つの経路を状況に応じて使い分けることで、従来よりも柔軟な負荷分散を実現した。本稿では、この枠組みを ns-2 上に実装し、シミュレーション実験を行った結果を報告する。

キーワード インターネット、負荷分散、トラヒックエンジニアリング、経路制御、ルーティングテーブル

## A Flexible Load-balancing Method Using a Secondary Routing Table

Toru YAMAMOTO<sup>†</sup>, Takuya YOSHIHIRO<sup>‡</sup>

<sup>†</sup> Graduate School of Systems Engineering, Wakayama University

<sup>‡</sup> Faculty of Systems Engineering, Wakayama University

930 Sakaedani, Wakayama, 640-8510 Japan

E-mail: <sup>†,‡</sup>{s051058,tac}@sys.wakayama-u.ac.jp

**Abstract** In the current Internet, routers use only one routing table that is computed automatically based on the link costs configured by network administrators. However, this mechanism prevents us from making a well-balanced use of network resources, because packets tend to pass several particular links under the shortest-path based routing environment. In this paper, we expand a link-state type IGP routing protocol with a secondary routing table so that each router have two disjoint paths for every destination. By using those two paths, each router can make a good selection of packet forwarding paths based on the locally measured traffic load information. We implemented our proposed method on ns-2, and report the simulation results.

**Keyword** Internet, Load-Balancing, Traffic Engineering, Routing, Routing Tables

### 1. はじめに

近年インターネットの普及・発展に伴い、インターネット上での IP 電話や映像・音声の配信、インターネット放送など多様なサービスが出現してきた。それに伴ってネットワークを流れるトラヒック量も増加しており、通信品質の改善がより一層求められるようになってきている。しかし、現在一般的に用いられるルーティングプロトコルでは、管理者により設定されたコスト値に基づく最短経路のみを用いた通信をするため、ネットワークにおいて負荷の発生箇所に偏りが起こりやすく、効率的な資源の活用が行われていない。

この状況を回避する手法として、リンクのトラヒック流量に応じて動的に経路を制御し、負荷を分散させることが考えられる。このような技術は一般的にトラヒックエンジニアリングと呼ばれており、これまでに様々な手法が提案・検討されている [1]。また、現在イ

ンターネットで用いられているルーティングプロトコル OSPF (Open Shortest Path First) [2] を拡張し、ネットワーク内のリンク負荷に応じて動的にリンクのコスト値を変更する経路制御手法も提案されている [3-6]。しかし、この手法を用いても、ある宛先ノードへパケットが転送される経路は一つだけであるため、特定リンクに多くのトラヒックが発生した状況では負荷を分散できない。また、ネットワークのトラヒック流量は常に変動しており頻繁に状況が変化するため、これを指標とした経路制御を行う場合には、経路収束の安定性が懸念される。

このような状況においても適切に負荷分散を行うためには、宛先までのパスを複数用意することが必要になる。そこで本研究では、次章で説明する障害回復モデルを適用し、パケット転送に用いるパスをネットワークの負荷状況に応じて柔軟に選択できる枠組みを

提案する。本枠組みでは、各ルータに第二のルーティングテーブルを追加する程度の拡張により、どのルータからでも代替経路が利用できるという特徴があり、従来手法よりも格段に柔軟な負荷分散が可能である。本稿では、この枠組みを用いた負荷分散プロトコルに必要な条件について検討したうえで、シミュレーション実験を通じて本手法の有効性を確認した。

以下、2章では第二のルーティングテーブルを用いた障害回復モデルの詳細について説明する。3章では障害回復モデルをプロトコル化するにあたって必要な条件を検討したうえで、この枠組みを用いた負荷分散手法について説明する。4章では、シミュレーション実験の結果について述べた後、5章では考察を行う。最後に6章でまとめを行う。

## 2. 第二のルーティングテーブル

本章では、本研究で適用する障害回復モデルの詳細について説明する。

### 2.1. 障害回復モデル

OSPFのような最短経路ルーティングにおいて、瞬時にリンクの故障回復を可能とする障害回復モデル[7]が提案されている。このモデルを用いて、どのような2枝連結ネットワーク（任意の2点間に、リンクを共有しない2本の経路が存在するネットワーク）においても、任意の1リンク故障に対応できるような第二のルーティングテーブルを構成できる。

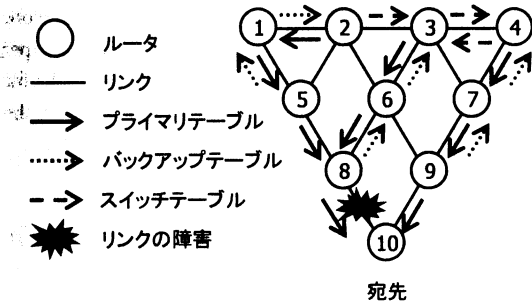


図1 障害回復モデル

本障害回復モデルを図1に示す。本モデルでは、ネットワーク上の全リンクが双方向リンクであることを前提としている。図中のプライマリテーブルとは、通常の最短経路計算により求めた、各ルータから宛先（この場合はルータ10）に対する最短経路木を表している。つまり、プライマリテーブルとは、通常のルーティングプロトコルで計算される第一のルーティングテーブルのことであり、図1ではルータから出る実線矢印により表される。

本モデルでは、これに第二のルーティングテーブルを追加する。図中のバックアップテーブル（点線）とスイッチテーブル（破線）が第二のテーブルであり、ルータは各宛先に対してこのどちらかを持つ。文献[7]により提案されたアルゴリズムを用いて第二のテーブルを計算すると、各ルータから、次のように代替経路を利用することが可能である。

図1において、各矢印はパケットの宛先をルータ10と仮定した場合のパケット転送先を示している。この状況で、例えばルータ8とルータ10の間にリンク障害が発生した場合には、通常のルーティングではルータ8にてパケットが破棄されるが、本モデルでは、ルータ8が障害を検出すると同時に、次のように迂回経路を用いて障害リンクを避けるパケット転送を行うことが可能である。例として、ルータ1からルータ10へのパケットが障害を避けて転送される様子を次に示す。

- (1). 1→5→8（通常の最短経路ルーティングと同様に、プライマリテーブルを用いて転送される）
- (2). 8→6（ルータ8がリンク障害を検知し、使用テーブルをプライマリからバックアップに切り替えることでルータ6に転送する。）
- (3). 6→3（一度バックアップを利用した後は、続けてバックアップを使い、ルータ3に転送する。）
- (4). 3→4（同様に、ルータ3がスイッチを使用し、ルータ4に転送する。）
- (5). 4→7（スイッチを使用した後にはプライマリテーブルに切り替える。ルータ7に転送する。）
- (6). 7→9→10（その後は通常のルーティングと同様にプライマリテーブルを使用しルータ9へ、続いてルータ10へ転送される。）

上記はルータ8-10間の障害例であったが、この障害回復モデルではどの1リンクの障害であっても、各ルータが他のルータと通信することなしに瞬時に判断してパケットの迂回が可能であるため、非常に高速な障害対応が可能である。

### 2.2. 経路の切替フラグ

障害を検知したルータが使用するルーティングテーブルを切り替えることで迂回経路を利用することは前述のとおりである。ここで各ルータは、どのパケットがどのテーブルを使用するのかを判断できなければならない。本モデルでは、このために、パケットヘッダに1ビットの「切替フラグ」を追加する。つまり、各ルータがパケット受信時に切替フラグを見て、どのルーティングテーブルを使用すればよいかを判断する。切替フラグは通常「0」に設定しておき、必要に応じて「1」に切り替える。具体的には、ルータは次のように動作すればよい。

- 切替フラグが“0”の時
  - プライマリへ転送
  - 迂回路を使う場合はフラグを“1”にしてバックアップまたはスイッチへ転送
- 切替フラグが“1”の時
  - バックアップまたはスイッチへ転送
  - スイッチならばフラグを“0”にして転送

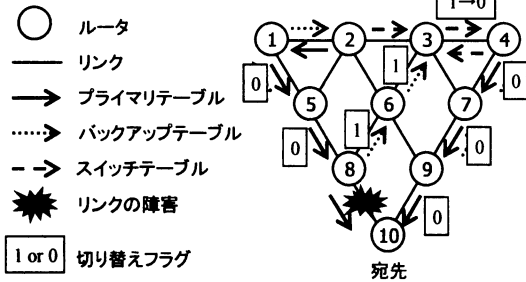


図 2 障害回復モデルにおける切替フラグの変化

図 2 に、ルータ 1 からルータ 10 へ転送されるパケットの切替フラグの状態を示す。

- (1) 1→5→8 (切替フラグは“0”のままである。)
- (2) 8→6 (ルータ 8 は、切替フラグを“0”から“1”に変更する。)
- (3) 6→3 (切替フラグは“1”のままである。)
- (4) 3→4 (ルータ 3 は、切替フラグを“1”から“0”に変更する。)
- (5) 4→7 (切替フラグは“0”のままである。)
- (6) 7→9→10 (同様に、切替フラグは“0”のままである。)

### 3. 負荷分散プロトコル

本章では、まず 2 章で紹介した障害回復モデルを実現するために、経路制御プロトコルが備える条件について説明する。次にこのプロトコル上で負荷分散を行う際の方針と個々のルータの挙動について説明する。

#### 3.1. 経路制御プロトコル

2 章で紹介した障害回復モデルを実現するには考慮すべき点が存在する。本節では以下に示した 4 つの課題と解決策について述べる。

- A) パケットループ問題
- B) フラグの実装方法
- C) AS 境界でのフラグの扱い
- D) プロトコルにおけるテーブル計算手順

まず A) パケットループ問題とは、2 箇所以上のリンクが故障した場合に、パケットが同じ場所をループする問題である。前述の例ではルータ 8-10 間のリンク故障を想定したが、同時にルータ 7-9 間のリンクも故障

した場合には、パケットは 1→5→8→6→3→4→7→4→3→6→8→6 の順に転送され、ループしてしまう。

これを解決するために、一度でも迂回路を利用したパケットはそれ以後は迂回させない仕組みを実装する。このためには、切替フラグとは別に「迂回フラグ」を用意すればよい。迂回路を利用したことがなければこのフラグを“0”に設定しておき、バックアップに切り替えられた際に“1”とすれば、各ルータがこのフラグを用いて判断できるためループ問題は回避できる。

B) これら 2 ビットのフラグをパケットヘッダに用意する必要がある。これは、IPv4 であれば TOS (Type of Service) フィールドを用いて、IPv6 であれば拡張ヘッダを用いることで、既存プロトコルへのパケットフォーマットを変更することなしに実装できる。

C) 実装されたフラグが“1”のままクリアされずに他の AS (Autonomous System) に渡ると、他の AS 内でのサービス妨害や、エラーパケットとしての誤認識を招く可能性がある。また、隣接 AS からのパケットのフラグが初めから“1”であれば、自 AS 内のパケット転送に悪影響を及ぼす恐れがある。これを回避するために、AS 境界(入口と出口)でフラグ状態を確認し、“1”であれば“0”にクリアする必要がある。これにより、他の AS に影響を与えることなく本枠組みを運用できる。

D) 本枠組みは、リンクステート型ルーティングプロトコルを拡張することで実装できる。ここでは、どのように拡張すれば問題なく動作するかという点について説明する。

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、まずフラッディングによって経路情報の交換が行われ、次に Dijkstra 法を用いて最短路が計算される。本枠組みは、最短路が計算された後に、第二のテーブルを計算する作業を追加するだけで、問題なく動作する。第二のテーブルは全体のトポロジとプライマリテーブルに使われている最短路木から計算できるが、この計算が全ルータで終了し、安定するまでの間の動作が懸念される。

最短路が計算されてから第二のテーブルが計算されるまでの間には、ネットワーク上に、第二のテーブルが計算されたルータとそうでないルータが混在する。あるルータが迂回路にパケットを転送しても、迂回路上に、第二のテーブルが未計算のルータがあれば、そのパケットが通るはずの迂回路は利用することができない。しかしこの場合にも、最低限パケットが宛先に到達することができれば、第二のテーブルを追加することによる通信性能の劣化はあまり問題とはならない。

そこで、本枠組みでは、第二のテーブルが計算され次第、各ルータで使用して良いこととし、もし第二のテーブルが未計算のルータに、切替フラグが“1”のパケ

ットが到着した場合には、切替フラグを”0”に、迂回フラグを”1”にして、プライマリテーブルへ転送することとする。これにより、そのルータから最短路を通って宛先に到着することができ、パケットループ問題も発生しない。このように、各ルータが独立して動作することで、同期処理による複雑化を避け、プロトコルを単純に保つことができる。

### 3.2. 自己計測値に基づく負荷分散

既存の IP ネットワークではルーティングプロトコルによって計算された最短路を使用するため、ネットワークの一部にトラフィックが集中することが多い。そこでトラフィックを分散させるべく、各ルータに隣接リンクのトラフィック流量を監視させ、その流量に応じてパスを切り替える手法を提案する。つまり、輻輳が発生し、パケットロスが発生しそうな状態を検出し、破棄されそうな一部のパケットのみを迂回路に転送する。もちろん、輻輳が回避されると全パケットがプライマリパスを通る状態に戻る。そうすることで、従来手法ではロスするパケットを救出し、迂回路の空き帯域を使って宛先まで到達できるようにすることで、ネットワーク資源の利用効率を向上できる。

図3に障害回復モデルを応用した、負荷分散の提案手法を示す。通常はプライマリパスを利用するためパケットは1→5→8→10の順で宛先へ到着する。しかし、ルータ8-10間に輻輳が発生した場合、一部のパケットがルータ8から迂回路に転送され1→5→8→6→3→4→7→9→10の経路で宛先に到着することになる。

提案する負荷分散手法は障害回復モデルを応用したものである。障害回復モデルはどのリンク障害でもパスを切り替えることで対応できた。それと同様に、提案手法もどのリンク間の輻輳でもルータ各自の判断で迂回路に転送することができる。

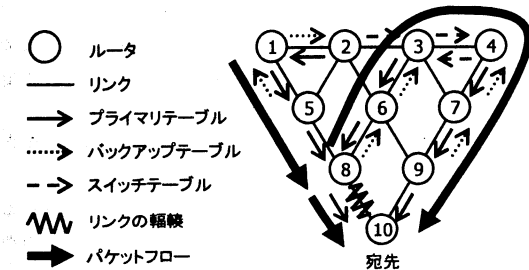


図3 障害回復モデルを応用した負荷分散手法

## 4. シミュレーション

本研究では既存研究である障害回復モデルを応用した負荷分散手法の提案を行い、ネットワークシミュレータの ns-2[8]上にて実装して性能評価する。本章ではシミュレーションモデル、および性能の評価方法につ

いて述べる。

### 4.1. シミュレーションモデル

本節ではシミュレーションを実施する環境と、実験モデルについて説明する。始めに、計算機の仕様と ns-2 の設定内容を以下に示し、次に実験の目的とそのモデル、実験のパラメータを説明する。

- PCサーバ (Fedora Core)
  - CPU : Intel Pentium 4 3.00GHz
  - メモリ : 768MB
- ns-2 の設定内容
  - 各リンクの帯域は 1Mbps
  - 各リンクの遅延は 10ms
  - 各リンクのコストは 1 で統一
  - ルータの出力キューは FIFO 型
  - 出力キュー容量の 90% 超で迂回路へ転送

本実験では、メッシュ型のネットワーク (図4) を実験モデルとし、網内を通過するようなトラフィックを発生させた。ネットワーク負荷が低負荷から高負荷へと変化する過程において、提案手法がどのような性能を示すかを調べることを目的とした。

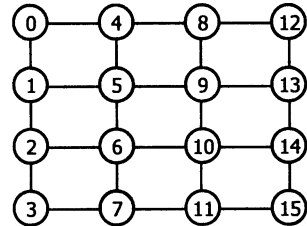


図4 実験ネットワーク

実験においてトラフィックとして発生させるフローは次のように決定される。まず、送信ノードを左側のルータ 0,1,2,3 の中から、次にその対となる宛先ノードを右側のルータ 12,13,14,15 の中から共にランダムに選択する。フローを発生する度に送信ノード、宛先ノードをランダムに選び、図5において左端から右端への右向きのフローを発生させた。一定時間間隔で、一定時間持続するフローを発生させることで、少しずつトラフィックを増加させた後、フロー数が一定に保たれる期間の通信状況を観察した。これらのフローのパラメータは以下の通りである。

- フローの種類 : UDP の CBR (Constant Bit Rate)
- 発生レート : 500kbps
- フロー発生間隔 : 5 秒
- 通信持続時間 : 1 フローあたり 30 秒
- 最大負荷量 : 3Mbps (フロー6 本分)

実験では右向きのフローが発生するため、右向きのリンクについてネットワーク全体の帯域は 1Mbps×4 の 4Mbps であり、これ以上は流れない。よって、負荷の

最大量が3Mbpsであることはネットワークの負荷が上がり過ぎない状況である。また発生レートが500kbpsであることから、3本のフローが同じリンクを通るときに1Mbpsの帯域を越えるため、うまく迂回させないとパケットが破棄される状況である。つまりこれらのパラメータからわかるように、ネットワーク全体の負荷を上げず、一部で輻輳が起こる状況において提案手法の有効性を確認することが本実験の狙いである。

## 4.2. 性能評価方法

前節で説明した条件にてシミュレーションを行い、提案手法の有効性を確認するために、調査・比較する対象を以下の方式とした。

- OSPF
- OSPF-ECMP (Equal-Cost Multi-Path)
- 提案手法

OSPFは通常の最短路を用いた経路制御を行う。それに対し、ECMPは宛先に対して同じコストで到達できるような複数経路が存在する場合に、各経路に等しくトラフィックを分散させる手法である。OSPFではECMPが一つの機能として提供されておりOSPF-ECMPと称されることが多い[2][9]。

一方、本実験の性能評価指標は総スループットとした。本研究では、集中するトラフィックを分散させることで、他の空きリンクの帯域を有効活用することであるため、その有効活用度合いを調べるためには、ネットワーク全体のトラフィック量、つまり総スループットが重要な指標となると考えられる。スループットはネットワーク上のあるフローによって宛先へ転送された、単位時間当たりのデータ量である。つまり、総スループットとは、フロー群のスループットを合計したものであり、転送中の総データ量と考えることができる。

## 5. 結果・考察

本章では提案した負荷分散手法の効果を確認するために、前章で説明したシミュレーション実験の結果を示し、考察を行う。送信、受信ノードをランダム選択することから、偶然による誤った評価を避けるために、実験は5回行った。そのうち、典型的な2回の実験結果を図5と図6に示し、その後、実験5回の平均(図7)を示すことで、提案手法の有効性を確認する。

### 5.1. 実験結果

図5は1回目の実験における総スループットの時間推移を示したものである。この結果より、総じて提案手法が他の2方式よりも良好な総スループットを示していることがわかる。これは、あるリンクにトラフィックが集中した状況で、提案手法が混雑状況に応じて柔軟に転送パスを切り替えたためである。また、OSPFと提案手法を比較すると、31秒の時点で約1.4倍、OSPF-ECMPとの比較では99秒に、約1.3倍になる性

能が確認できた。50~60秒にかけて提案手法の性能が下がる傾向が見られたが、提案手法だけでなく他の2方式も性能が下がっている。これはあるリンクへのトラフィックが集中したことが原因であるが、提案手法がこの状況を完全に回避できなかったのは、迂回路でも輻輳が発生したため、結局はパケットが破棄されたことが原因である。

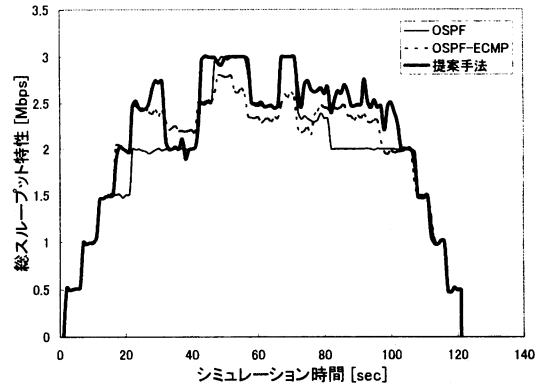


図5 総スループットの時間推移(実験1)

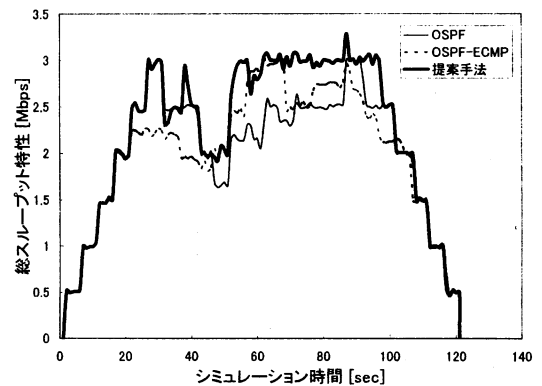


図6 総スループットの時間推移(実験2)

図6は2回目の実験における総スループットの時間推移を示したものである。この図からも、提案手法がほぼ全ての時間帯で他の2方式よりもよい総スループットを示すことがわかる。また、図7に見られる総スループットの最大値は、最大送信レートである3Mbpsを上回っている。これは、OSPFなどでは破棄されるはずだったパケットが迂回路へ流れるため、ネットワーク全体におけるパケット転送量が一時的に増加していることを意味している。これより、提案手法によってパケット破棄を減らし、総スループット性能が向上した。またOSPFと提案手法の比較においては

55秒の時点で約1.4倍、OSPF-ECMPとの比較では38秒に約1.5倍となる良好な結果を確認できた。

図7は5回行った実験の総スループットの平均値を表している。この時間推移を見ることで提案手法の総合的な性能を判断できる。この結果においても、ほぼ全ての時間帯において提案手法の優位性を確認できる。以上により、各実験の結果と、その平均のどちらみても、提案手法が既存の2方式よりも良好な総スループットを示すことが確認された。

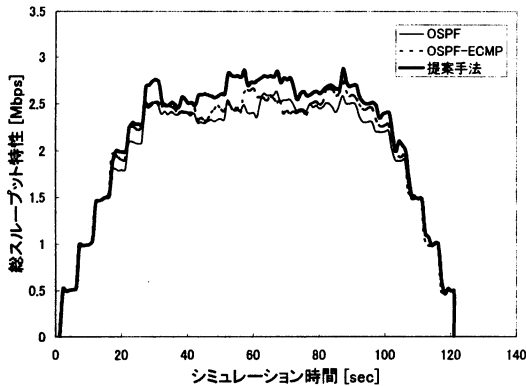


図7 総スループット特性の時間推移  
(5回行った実験の平均)

## 5.2. 考察

本実験では一部のリンクに輻輳が起るようランダムに複数フローを発生させ、提案手法の有効性を検証した。複数回行った実験でも、提案手法がほぼ全時間帯において良好な総スループット特性を示すことが確認できた。但し、一部時間帯のみではあるが、OSPFやOSPF-ECMPが優勢な時間帯が存在した。これは迂回したパケットが迂回先で再び輻輳し、破棄されることで、通常の最短路ルーティングよりも余分に資源を消費しているためと予想される。

ところで、本実験では、各リンクのコストを"1"に固定していた。これはECMPが有効に働くような経路が多数存在したことを意味する。つまり、ECMPは各経路にトラフィックを分散できるため、今回の実験がECMPにとって最もネットワーク資源を有効に活用できる状況であったといえる。その状況にも関わらず提案手法はOSPF-ECMPよりも有効といえることは今回の実験結果が示すとおりである。また、ECMPはコストの問題があるため、利用できる場面に限られるが、提案手法はコストに依存しない。つまり、実際の多様なネットワークにおいても、提案手法が総スループット特性を向上させ、ネットワーク資源の有効活用に貢献できると考える。

## 6. まとめ

本研究ではネットワーク資源の有効活用を目的とし、複数パスにトラフィックを分散させる手法を提案した。この手法は、既存研究の障害回復モデルを応用したものであり、各ルータがパケットの転送パスを切替えることでトラフィック分散を図るものである。

提案手法ではOSPFのような既存のルーティングプロトコルで最短路のプライマリパスが計算された後に、追加で計算されるバックアップの迂回路を用いてトラフィックを分散させる。各ルータには隣接リンクのトラフィック流量を監視させ、プライマリパスが高負荷な場合にのみ迂回路のパスへ転送させる。そうすることで輻輳の発生しやすいプライマリパスの負荷軽減や、パケットロスの低減を目指した。

また、ns-2への障害回復モデルと提案手法の実装を行い、シミュレーション実験を行った。この実験で提案手法がリンクの空き帯域を有効活用でき、その結果として提案手法の優位性を確認した。

今後の課題として、様々な状況や、多様なトポロジでのシミュレーション実験を行いたい。また、従来手法では問題にならない、迂回路へ流すことによる他リンクの通信品質の悪化についても検討する必要があるだろう。

## 文 献

- [1] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering," IETF RFC 3272, May.2002.
- [2] John Moy, "OSPF Version 2," IETF RFC 2328, Apr.1998.
- [3] G. Apostolopoulos, S.Kamat, D. Williams, R. Guerin, A. Orda, and T. Przygienda, "QoS Routing Mechanisms and OSPF Extensions," IETF RFC 2676, Aug.1999.
- [4] R. Guerin, Ariel Orda, and D. Williams, "QoS Routing Mechanisms and OSPF Extensions," Proceedings of 2nd Global Internet Miniconference (joint with Globecom'97), Nov.1997.
- [5] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi, "Quality of Service Based Routing: A Performance Perspective," Proceedings of SIGCOMM'98, 1998.
- [6] G. Apostolopoulos, R. Guerin, and S. Kamat, "Implementation and Performance Measurements of QoS Routing Extensions to OSPF," Proceedings of INFOCOM'99, Mar.1999.
- [7] Hiro Ito, Kazuo Iwama, Yasuo Okabe and Takuya Yoshihiro, "Single Backup Table Schemes for Shortest-Path Routing," Theoretical Computer Science, Vol.333, pp.347-353, 2005.
- [8] The VINT Project, The Network Simulator - ns-2, <http://www.isi.edu/nsnam/ns/>.
- [9] C. Hopps, "Analysis of an Equal-Cost Multi-Path Algorithm," IETF RFC 2992, Nov.2000.