

# トラフィック変動に応じた 発見的手法に基づく動的アプリケーション層経路制御手法

松田 一仁<sup>†</sup> 長谷川 剛<sup>††</sup> 村田 正幸<sup>†</sup>

<sup>†</sup> 大阪大学大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘 1-5

<sup>††</sup> 大阪大学サイバーメディアセンター 〒560-0043 大阪府豊中市待兼山町 1-32

E-mail: †{k-matuda,hasegawa,m-murata}@ist.osaka-u.ac.jp

あらまし アプリケーション層においてエンド端末を中継ノードとする経路を候補とした経路制御を行うことにより、エンド端末間のネットワーク性能が向上することが明らかにされている。一方で、個々のユーザが独自に経路選択を行うと、経路の重複によるユーザ性能の悪化や、トランジットリンク利用の増加による ISP 間トランジットコストの増大を招くことが考えられる。我々はこの問題に対して、アプリケーション層における経路選択の最適化問題を定義し、これを分散型焼きなまし法を用いて解くことによって、経路制御を行う手法を提案している。本稿では、エンド端末間のトラフィック量が変動するネットワーク環境を想定して提案手法の拡張および評価を行った。その結果、AL ノードが独自に経路を選択する場合に比べて過負荷な経路の数を 70% 削減出来ることを示した。また、適用するネットワーク環境に応じた提案手法の利用形態についての議論を行った。

キーワード ネットワーク最適化、オーバーレイネットワーク、経路制御、ISP 間トランジットコスト、焼きなまし法

## A dynamic application-level routing method reacting traffic changes based on distributed heuristic algorithm

Kazuhito MATSUDA<sup>†</sup>, Go HASEGAWA<sup>††</sup>, and Masayuki MURATA<sup>†</sup>

<sup>†</sup> Graduate School of Information Science and Technology, Osaka University  
Yamadaoka 1-5, Suita-shi, Osaka, 565-0871 Japan

<sup>††</sup> Cybermedia Center, Osaka University Machikaneyama-cho 1-32, Toyonaka-shi, Osaka, 560-0043 Japan

E-mail: †{k-matuda,hasegawa,m-murata}@ist.osaka-u.ac.jp

**Abstract** Application-level traffic routing that chooses an end-to-end traffic route relaying other end-hosts can improve user-perceived performance. However, selfish route selection performed by each end user can lead to the decrease in the path performance due to overload by route overlaps. It also causes the increase in the inter-ISP transit cost as a result of utilizing more transit links than in the case of native IP routing. To deal these problems, we formulated the application-level route selection as the optimization problem and proposed the method based on distributed simulated annealing to obtain near-optimal solutions to the problem. In this report, we extend the proposed method to react the network traffic changes. We evaluate the proposed method in such an environment and the proposed method can reduce the overloaded routes by 70% compared a method without a coordinated manner. We also discuss some application-scenarios of the proposed method.

**Key words** network optimization, overlay network, routing, inter-ISP transit cost, simulated annealing

### 1. ま え が き

エンド端末間のトラフィックに対して、他のエンド端末を中継するような経路を用いる、アプリケーション層経路制御手法が近年注目されている。この手法により、ユーザの体感性能（以下、ユーザ性能と呼ぶ）である遅延時間や利用可能帯域、パケットロス率などのネットワーク性能が IP 層での経路制御のみを用いる場合に比べて向上することが明らかにされている [1-4]。本稿では以降、アプリケーション層を AL と記述し、アプリ

ケーション層での経路制御を AL ルーティングと呼ぶ。また、AL ルーティングにおいて送信元、宛先、中継端末となるエンド端末を AL ノードと呼ぶ。

一方、AL ルーティングによって制御されたトラフィックによって Internet Service Provider (ISP) 間のトランジットリンクの利用が増加し、ISP の金銭的成本（以降、トランジットコストと呼ぶ）が増大する可能性がある。AL ルーティングにおける経路は IP 層におけるエンド間経路を複数本含んでいるため、このような状況が発生することは大きな問題となる。加えて、

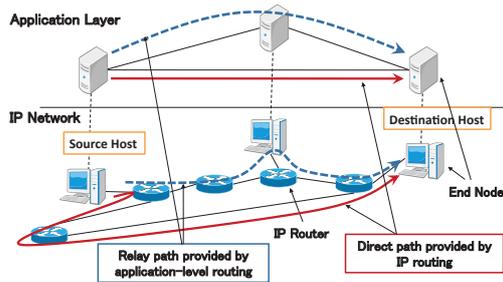


図 1 アプリケーション層経路制御

多数のユーザによって独自に経路選択が行われた場合、利用経路の重複によって輻輳が発生することでユーザ性能が劣化することが考えられる。例えば、複数のオーバーレイネットワークが互いに協調しないで経路制御を行う場合、経路選択が振動することが指摘されている [5]。

我々は [6] において、エンド間計測で得られるネットワーク性能を用いて AL 経路を制御することにより、AL ルーティングによるトランジットコストの増加を低減できる可能性があることを示した。しかし [6] では経路の重複は考慮しておらず、これを考慮した手法の考案が必要であった。そこで [7] において、AL ルーティングにおける経路選択において、経路の重複によるネットワーク性能変化を考慮した最適化問題として定義し、これを分散型焼きなまし法を基とした方法で解く手法を提案した。提案手法を用いることで、AL ノードがネットワーク上に分散した環境において、AL ノード間で情報共有を行い、経路の重複によるネットワーク性能への悪影響を避けた経路選択を行うことができる。しかしながら [7] においては、AL ノード間のトラフィックが変動しない場合を想定しており、トラフィックが変動するネットワーク環境においては更なる拡張およびその評価が必要である。

そこで本稿ではまず、提案手法をトラフィックが変動するネットワーク環境に適合する形に拡張を行う。具体的には [7] で示した最適化問題および分散型焼きなまし法を基とした AL 経路制御手法を簡潔に示した後、トラフィック変動を監視し、変動に対応して経路選択を繰り返し行うアルゴリズムを追加した提案手法を示す。性能評価は、PlanetLab ノードが AL ルーティングを用いる環境を想定して実測データを用いてシミュレーションを行う。トラフィックの変動を再現した評価シナリオを用いて性能評価を行い、ユーザ同士が協調せずに独立して経路選択を行う場合と比較して、提案手法がトラフィックの変動する環境においても有効に機能し、ユーザ性能の向上や輻輳の回避を実現しつつ、トランジットコストの増加を抑制出来ることを示す。また、評価結果を踏まえた提案手法の利用法および昨今のインターネットポロジの変化による提案手法への影響についての議論を行う。

以下、2. 章では本稿で扱う AL ルーティングの利点及び問題点について説明する。3. 章では [7] で定義した AL 経路選択最適化問題およびこれを基にした経路制御手法を示し、その拡張を行う。4. 章では、PlanetLab 環境を想定した提案手法の評価結果を示す。その後、5. 章において提案手法の利用法およびインターネットポロジの変化による影響を議論した後、6. 章で本稿のまとめおよび今後の課題について述べる。

## 2. アプリケーション層経路制御

### 2.1 ユーザ性能の向上

先述の通り、AL ルーティングを用いることでユーザ性能の向上が得られることが明らかにされている。これは主として IP 層における経路制御と AL ルーティングのポリシーの差異によってもたらされる。AL ルーティングがユーザ性能に直結する指標を用いるのに対し、IP 層における経路制御では IP レベルや Autonomous System (AS) レベルのホップ数を基本的な指標として用いる。加えて、IP 層経路制御の提供者である ISP は ISP 間の商業的關係に基づく独自のコスト構造を構築している。ISP が管理する AS は主としてトランジットリンクとピアリングリンクという 2 種類のリンクで接続されている。トランジットリンクはより広範囲への接続性を提供する上流の ISP とそれを利用する下流の ISP とを繋ぐものであり、ピアリングリンクは 2 つの ISP を相互に接続し、ISP 間での直接

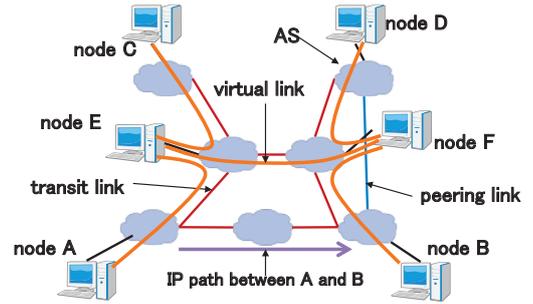


図 2 アプリケーション層経路制御の問題点

のトラフィックのやり取りに利用される。一般的に、トランジットリンクではトラフィック流量に応じた金銭的成本が上流 ISP に支払われ、ピアリングリンクでは設備の維持費以外には金銭的成本が発生しない。ISP はこのようなリンク種類の違いに基づく経路制御を行っている。しかし、IP/AS レベルのホップ数や ISP のコスト構造に基づく経路制御は必ずしもユーザ性能に関係するものではなく、ここに AL ルーティングによってユーザ性能が向上する余地がある。

図 1 は、AL ルーティングによってユーザ性能が向上する例を示している。図 1 において、IP ルータ間のリンクの長さが伝搬遅延時間を表すものとすると、IP レベルのホップ数の観点では IP 層経路制御のみで提供される経路の方が性能が良いが、エンド間の遅延時間の観点では AL ルーティングによって提供される他エンド端末を迂回した経路の方が性能が良い。

### 2.2 問題点

AL ルーティングによってユーザ性能の向上が得られる一方で、経路選択において他 AL ノードの経路を考慮しないとすると、性能のよい AL リンクを複数の経路で重複して利用されることがある。このような状況においては、AL ルーティングによって得られる性能向上が劣化すると考えられる。図 2 にその例を示す。図中の全てのエンド端末は AL ノードであり、それぞれ AL リンクで接続されている。このときノード A からノード B へ、ノード C からノード D へトラフィック要求が発生するとする。ノード E, F 間の AL リンクがよい性能を持つとすると、ノード A, B 間とノード C, D 間の経路において同時にノード E, F 間の AL リンクを利用しようとする考えられる。その結果、両経路共に遅延時間の増大や利用可能帯域の低下等のユーザ性能の劣化が発生する。

さらに、AL ルーティングの利用によって ISP のコスト構造に従わないトラフィックが発生し、ISP 間のトランジットコストが増大することが考えられる。このようなトラフィックが増加するとネットワーク全体のトランジットコストが増大する。例えば図 2 において、それぞれの AL リンクは複数の AS 間リンクを含むとし、AS 間リンクはトランジットリンクまたはピアリングリンクであるとする。このときノード A からノード B へトラフィック要求が発生するとする。IP 層での経路制御のみで提供される経路では通過するトランジットリンクは 2 本である。しかしノード E, F を中継するような経路を選択したとすると、利用するトランジットリンクが 1 本増加するため、その結果ネットワーク全体のトランジットコストが増加する。

## 3. 提案手法

本章では、本稿で想定するネットワークモデルを示した後 [7] で定義した最適化問題を簡潔に示す。その後 [7] で提案した手法を静的なトラフィックに対する AL 経路選択手法として説明する。最後に、これを拡張した動的なトラフィック変動に対応した AL 経路選択手法を示す。

### 3.1 ネットワークモデル

本稿では、図 3 に示すようなネットワークモデルを想定する。下位層ネットワークは複数の AS (ISP に対応するものとする) から構成されており、各 AS は複数の IP ルータによって自身のネットワークを構築している。AS 同士はトランジットリンクまたはピアリングリンクで接続されており、トラフィックがトランジットリンクを通過する際にはトランジットコストが発生する。アプリケーション層経路制御は、下位層ネットワーク上で各 AS に属するエンド端末をノードとして行われる。1 ホップの AL 経路は、下位層ネットワークによって提供されるエン

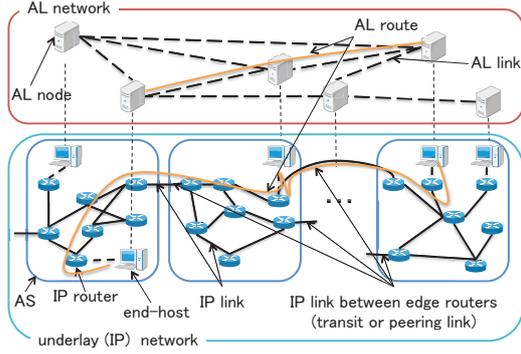


図3 ネットワークモデル

ト間経路に相当する．例えば図3では，AL ネットワーク上に実線で示した2ホップのAL経路は下位層であるIP層のエンド間経路を2本含んでいる．なお，以降，1ホップのAL経路を直接経路と呼ぶ．

### 3.2 AL経路最適化問題

ALノードの総数を $N$ とし，2つのALノードからなるノードペアを考え，それぞれのノードペアは1から $(N-1)N$ までの識別子を持つものとする．アプリケーション層経路制御において，ALノードペアに対してそれぞれAL経路が選択されたとする．このとき，AL経路のルーティング行列 $A^{\text{AL}}$ を次のように定義する．添字は行，列共に $1, 2, \dots, (N-1)N$ の順に並んでいるとする．

$$A^{\text{AL}} = \begin{pmatrix} \text{AL}_1^1 & \cdots & \text{AL}_1^{(N-1)N} \\ \vdots & \ddots & \vdots \\ \text{AL}_{(N-1)N}^1 & \cdots & \text{AL}_{(N-1)N}^{(N-1)N} \end{pmatrix} \quad (1)$$

要素 $\text{AL}_i^j$ が1であれば，ALノードペア $i$ 間のAL経路でALノードペア $i$ 間のALリンクが用いられることを意味する．

各ALリンクのトラヒック量を並べた行列を $\mathcal{Y}$ とする．各ALリンクの遅延時間を並べた行列が $\mathcal{D}$ を引数とする関数 $f_D$ によって決定されるとすると，各ALノード間のAL経路の遅延時間を並べた行列 $D^{\text{AL}} = (d_1^{\text{AL}} \ d_2^{\text{AL}} \ \cdots \ d_{(N-1)N}^{\text{AL}})$ は次式で表せる．各要素 $d_j^{\text{AL}}$ はALノードペア $j$ 間のAL経路の遅延時間を表す．

$$D^{\text{AL}} = f_D(\mathcal{Y})A^{\text{AL}} \quad (2)$$

$\mathcal{Y}$ と $A^{\text{AL}}$ を引数とする関数 $f_B, f_C$ によって，それぞれ各ALノード間のAL経路の利用可能帯域を並べた行列 $B^{\text{AL}} = (b_1^{\text{AL}} \ b_2^{\text{AL}} \ \cdots \ b_{(N-1)N}^{\text{AL}})$ ，各AL経路のトランジットコストを並べた行列 $C^{\text{AL}} = (c_1^{\text{AL}} \ c_2^{\text{AL}} \ \cdots \ c_{(N-1)N}^{\text{AL}})$ が決定されるとすると，それぞれ次式で表せる．各要素 $b_j^{\text{AL}}, c_j^{\text{AL}}$ はそれぞれ，ALノードペア $j$ 間のAL経路の利用可能帯域，トランジットコストを表す．

$$B^{\text{AL}} = f_B(\mathcal{Y}, A^{\text{AL}}) \quad (3)$$

$$C^{\text{AL}} = f_C(\mathcal{Y}, A^{\text{AL}}) \quad (4)$$

提案手法では選択するAL経路のトランジットコストに関して直接経路のトランジットコストを基準とした制限を加える．直接経路のルーティング行列 $A^{\text{DR}}$ は，式(1)を単位行列にしたものに相当する．

直接経路のトランジットコスト $C^{\text{DR}} = (c_1^{\text{DR}} \ c_2^{\text{DR}} \ \cdots \ c_{(N-1)N}^{\text{DR}})$ は， $A^{\text{DR}}$ を用いて次式で計算できる．

$$C^{\text{DR}} = f_C(\mathcal{Y}, A^{\text{DR}}) \quad (5)$$

直接経路のトランジットコストを用いて，トランジットコスト制限は次式のように定義する．

$$c_j^{\text{AL}} \leq c_j^{\text{DR}} \times \alpha \quad (j \in K) \quad (6)$$

$K$ はトラヒック要求を持つALノードペアの集合とする．式(6)は選択するAL経路のトランジットコストを，直接経路のトランジットコストの $\alpha$ 倍以下に制限することを意味する．提案手法ではこれをAL経路を選択する際の制限として加えるこ

とにより，ALルーティングによるトランジットコストの増大を防ぐ．

通信要求を持つALノードペアの遅延時間の平均値を最小化する問題は各ALノード間のAL経路 $r_j (j \in K)$ を変数として次のように与えられる． $R_j^{\text{AL}}$ はALノードペア $j$ が選択できるAL経路の集合とする．

$$\begin{aligned} \text{minimize :} & \quad \left( \sum_{j \in K} d_j^{\text{AL}} \right) / |K| \\ \text{subject to :} & \quad r_j \in R_j^{\text{AL}} \\ & \quad c_j^{\text{AL}} \leq c_j^{\text{DR}} \times \alpha \quad (j \in K) \end{aligned} \quad (7)$$

同様に，通信要求を持つALノードペアの利用可能帯域の平均値を最大化する問題は各ALノード間のAL経路 $r_j (j \in K)$ を変数として次のように与えられる．

$$\begin{aligned} \text{maximize :} & \quad \left( \sum_{j \in K} b_j^{\text{AL}} \right) / |K| \\ \text{subject to :} & \quad r_j \in R_j^{\text{AL}} \\ & \quad c_j^{\text{AL}} \leq c_j^{\text{DR}} \times \alpha \quad (j \in K) \end{aligned} \quad (8)$$

### 3.3 提案手法

[7]ではより広範な状況に対応するため，上述の最適化問題を解く手法として集中型と分散型の2手法を提案したが，本稿では，ALルーティングを各ISPが協調して分散して行う状況を想定し，分散型の手法のみを示す．またこの手法を，トラヒックが変動する環境に対応した形に拡張を行う．

#### 3.3.1 静的なトラヒックに対するAL経路選択手法

提案手法では分散環境に焼きなまし法を適用するため，[8]で提案されている分散型焼きなまし法を用いた．一般に，焼きなまし法の処理は，対象となる問題に対する現在の解（以降，状態と呼ぶ）から微小な変更を加えた解（以降，近傍状態と呼ぶ）を生成し，状態の良さ（以降，コストと呼ぶ）を評価し，近傍状態に移るかどうかを確率的に決定することで進行する．分散焼きなまし法では，複数のノードが状態の一部分を担当し，それを他ノードと交換することで状態全体を共有する．各ノードは共有した状態に関して担当部分のみを変更した近傍状態を生成し，状態遷移の判断を行う．

提案手法のアルゴリズムでは，各ALノードペア間のAL経路の集合を状態とし，選択された経路を用いることによって得られると推定されるネットワーク性能をその状態のコストとする．各ALノードは，自身を送信元とするALリンクおよびAL経路を担当する．各ALノードはALルーティングによるトラヒックがない状態の担当ALリンクのネットワーク性能を計測によって取得し，計測結果をALルーティングの実行開始時に全ALノード間で共有する．加えて，各ALノード間のAL経路およびトラヒック要求量を定期的に情報交換する．各ALノードは交換した情報を用いてALリンクの性能を推定して経路選択を行う．

静的なトラヒックに対するAL経路選択手法のアルゴリズムをAlgorithm 1に示す．アルゴリズム中の添字 $i$ は $i$ 番目のALノードにおいて実行されるものであることを示す．また， $\text{Random}(x, y)$ は $x$ 以上 $y$ 未満の実数を返す関数とする．以下に，アルゴリズム中の各変数および関数について説明する．

#### 初期状態 $S_{\text{init}}$

初期状態は各ALノードにおいて，提案手法を実行する際に最初に用いる状態である．各ALノードは，担当するAL経路に関して直接経路を用いる状態を初期状態とする．

#### 近傍選択関数 $\text{Neighbor}()$

本関数は，状態 $S$ を引数とし，その近傍状態を返す．状態 $S$ の近傍は， $S$ に含まれるAL経路のうち，自身の担当する経路の一部を変更した状態とする．

#### コスト関数 $\text{Cost}()$

本関数は，3.2節で定義した最適化問題に対応した，現在の状態を用いた場合のネットワーク性能を推定して求められる全AL経路の平均遅延時間または平均利用可能帯域をコストとして返す．

#### 遷移確率関数 $\text{Probability}()$

遷移確率の導出を焼きなまし法で一般に用いられている次式を用いて行う． $T, S, S_{\text{tmp}}$ はそれぞれ焼きなまし法における

---

**Algorithm 1** 静的なトラフィックに対する AL 経路選択手法

---

```
1:  $I_i \leftarrow 0, T_i \leftarrow T_{i_{init}}, S_i \leftarrow S_{i_{init}}$ 
2: while  $T_i > F_{th}$  do
3:   Update( $S_i$ )
4:    $S_{i_{tmp}} \leftarrow \text{Neighbor}_{\text{dsa}_1}(S_i)$ 
5:   if  $\text{Cost}(S_i) \geq \text{Cost}(S_{i_{tmp}})$  then
6:      $S_i \leftarrow S_{i_{tmp}}$ 
7:   else
8:      $r_i \leftarrow \text{Random}(0, 1)$ 
9:     if  $r_i < \text{Probability}(T_i, \text{Cost}(S_i), \text{Cost}(S_{i_{tmp}}))$  then
10:       $S_i \leftarrow S_{i_{tmp}}$ 
11:   end if
12: end if
13:  $I_i \leftarrow I_i + 1$ 
14:  $T_i \leftarrow \text{Cooling}(T_i, I_i)$ 
15: if  $I_i \bmod U_i = 0$  then
16:   SendNeighbor( $S_i$ )
17: end if
18: end while
```

---

---

**Algorithm 2** 動的なトラフィック変動に対応した AL 経路選択手法

---

```
1:  $T_i \leftarrow T_{i_{init}}$ 
2:  $C_i \leftarrow 0$ 
3: loop
4:   StaticAlgorithm( $T_i$ )
5:   while  $T_i = 0$  do
6:      $C_i \leftarrow \text{CountChanges}(C_i)$ 
7:     if  $C_i \geq C_{th}$  then
8:        $T_i \leftarrow T_{re}$ 
9:        $C_i \leftarrow 0$ 
10:    end if
11:  end while
12: end loop
```

---

現温度, 現状態, 現状態の近傍状態とする.

$$\text{Probability}(T, S, S_{tmp}) = e^{-\frac{\text{Cost}(S_{tmp}) - \text{Cost}(S)}{T}} \quad (9)$$

初期温度  $T_{init}$  および冷却関数  $\text{Cooling}()$

焼きなまし法においては一般に, 初期温度はコストの増減に関わらず遷移が起こる程度に十分に大きくなければならない [9]. このことを考慮した設定を行う. また冷却関数として, 焼きなまし法で一般に用いられている次式を用いる.

$$\text{Cooling}(T, I) = \gamma T \quad (0 < \gamma < 1) \quad (10)$$

状態更新関数  $\text{Update}()$

本関数は, 実行時まで他に AL ノードから受信した AL 経路およびトラフィック要求量を用いて自身の状態の更新を行う.

経路通知関数  $\text{SendNeighbor}()$  および通知間隔  $U$

本関数は, 自身が変更した経路およびトラフィック要求量を他ノードへ通知する. コストの導出のためには, 他ノードが選択している経路の情報が交換のための通信オーバーヘッドが増大する. このため, 通知間隔  $U$  回の焼きなまし法の反復ごとに情報交換を行う.

### 3.3.2 動的なトラフィックに対する AL 経路選択手法

3.3.1 節で示したアルゴリズムを動的にトラフィックが変動する環境に対応する形に拡張する. 拡張したアルゴリズムでは, まず Algorithm 1 を実行した後, 待機状態に移行する. その後, トラフィック要求量の変化を監視し, 変動が一定分蓄積した際に再度 Algorithm 1 を実行する.

動的なトラフィックに対する AL 経路選択手法のアルゴリズムを Algorithm 2 に示す.  $\text{StaticAlgorithm}()$  は Algorithm 1 を実行することを示す. 以下に, アルゴリズム中の各変数および関数について説明する.

トラフィック変動監視関数  $\text{CountChanges}()$  および閾値  $C_{th}$

トラフィック変動監視関数は, AL ノード間のトラフィック要求量の変動を監視する. 自 AL ノードを送信元とする AL ノードペア間に新たなトラフィック要求が発生した場合には, 直ちに Algorithm 1 を実行する. 一方, 自 AL ノードを送信元とする AL ノードペア間のトラフィック要求が消滅した場合, あるいは, 他ノードを送信元とする AL ノードペア間のトラフィック要求が

発生あるいは消滅した場合には, その変化回数が  $C_{th}$  回に達した場合に Algorithm 1 を実行する.

**Algorithm 1** 再実行時の温度  $T_{re}$

Algorithm 1 の再実行は, 1 回以上実行した後の状態から実行するため, 初期温度と同等あるいはそれよりも低い温度でよい.

## 4. 評価

### 4.1 使用データおよび設定

#### 4.1.1 使用データ

本章では, 3. 章で提案した手法を PlanetLab に参加するノードがアプリケーション層経路制御を利用する状況を想定して評価した結果を示す. 評価のため, データ取得時に接続可能であった 657 の PlanetLab ノードに関してノード間経路に関するデータを取得した. 以下に, 取得したデータとその取得方法を示す.

遅延時間, IP レベルホップ数およびパス

PlanetLab ノード間で traceroute コマンドを実行して取得した. 本稿では 2010 年 10 月 18 日から 19 日にかけて取得したデータを用いた.

利用可能帯域, 物理帯域

Scalable Sensing Service ( $S^3$ ) [10] において公開されている PlanetLab ノード間のネットワーク性能計測結果を利用した.  $S^3$  では PlanetLab ノード間の物理帯域, 遅延時間, 利用可能帯域及びパケット棄却率がおよそ 4 時間毎に計測され, その結果が公開されている. 本稿では 2010 年 10 月 19 日に取得したデータを用いた.

AS レベルホップ数およびパス

IP レベルのパスに Route Views Project [11] において公開されている AS 番号と IP アドレスプレフィックスの対応を当てはめて取得した. 本稿では, 2009 年 4 月 16 日に取得したデータを用いた.

AS 間関係情報

CAIDA [12] において公開されている AS 間の関係情報を利用した. 本稿では 2010 年 1 月 20 日に取得したデータを用いた.

#### 4.1.2 コスト関数

式 (2) - (4) における関数  $f_D, f_B, f_C$  のは次の通りである.  $f_D$  は伝搬遅延時間と, 提案手法の過程で選択された AL 経路を用いた際に発生すると推定されるキューイング遅延時間の和を導出する関数とする. 具体的にはまず, 以下の仮定を置く.

- 全ての AL リンクは下層の IP リンクを共有しない.
- 全ての AL リンクにおいて物理帯域, 利用可能帯域のボトルネックとなる IP リンクは一致しており, エンド間の計測によって取得できる.

• AL リンクのキューイング遅延は主としてボトルネックとなる IP リンクで発生し, 他の IP リンクで発生するキューイング遅延は無視できる.

• 計測結果に含まれるキューイング遅延時間は AL トラフィックによって発生するそれに比べて十分に小さく, 無視できる.

以上の仮定を置くと, 伝搬遅延時間として AL トラフィックがない状態で各 AL ノードが計測した AL リンクの遅延時間を用いることができる. また, AL ノードペア  $j$  間の AL リンクのキューイング遅延時間  $d_j^q$  は, M/M/1 待ち行列モデルに基づいて, 次式で計算することができる.

$$d_j^q = \frac{u_j}{1 - u_j} \times \frac{P}{w_j} \quad (11)$$

ここで,  $u_j$  は AL ノードペア  $j$  間の AL リンクを計測することによって得られた物理帯域, 利用可能帯域, 及び AL ノード間のトラフィック要求量から計算できるボトルネックリンク利用率の推定値,  $w_j$  は物理帯域,  $P$  はパケットサイズの平均値とし, 一般的な最大パケットサイズである 1500 バイトと TCP の ACK パケットのサイズである 40 バイトの平均値 770 バイトを用いる. AL リンク  $j$  の遅延時間  $d_j$  は, 計測によって求められる伝搬遅延時間  $d_j^p$  と式 (11) によって求められるキューイング遅延時間  $d_j^q$  を用いて次式で計算する.

$$d_j = d_j^q + d_j^p \quad (12)$$

$f_B$  は、AL リンクの利用可能帯域に対し Max-min アルゴリズムを適用して AL 経路に割り当てる帯域を導出する関数とする。ノードペア  $i$  間の AL リンクにおいて各 AL 経路に割り当てられる帯域は、次式で決定されるものとする。

$$f_B(i) = (b_i^A - \sum_{j \in S^i} b_j^{AL}) / \{k | b_k^{AL} = 0, k \in S^i\} \quad (13)$$

ここで、 $b_i^A$  は計測によって得られる AL ノードペア  $j$  間の AL リンクの利用可能帯域、 $S^i$  はノードペア  $i$  間の AL リンクを利用するノードペアの集合とする。 $f_B(i)$  によって求められた帯域は、その AL リンクを利用するノードペアのうち、まだ帯域が割り当てられていないペアの AL 経路の帯域となる。すなわち、ノードペア  $j$  間の AL 経路の利用可能帯域を  $b_j^{AL}$  として、次式のように割り当てを行う。

$$b_j^{AL} = f_B(i) \{j | b_j^{AL} = 0, j \in K^i\} \quad (14)$$

式 (13) の計算は、計測した利用可能帯域が小さい AL リンクから順に行われ、その都度帯域を式 (14) に従って割り当てる。

$f_C$  は、利用する AS 間リンクの種類に応じて決まる値に AL ノードペア間のトラフィック要求量を積算してトランジットコストを求める関数とする。ある AL ノードペア  $j$  間の AL 経路のトランジットコスト  $c_j^{AL}$  は、AL ノードペア  $j$  間のトラフィック要求量  $x_j$  を引数として以下のように決定されるものとする。

$$c_j^{AL} = \beta_j x_j \quad (15)$$

ここで、 $\beta_j$  は AL ノードペア  $j$  間の AL 経路のトラフィック量に対するトランジットコストの増加量を決定する係数であり、AL 経路に含まれる AS 間リンク毎にその種類によって設定された値の和とする。すなわち、AL ノードペア  $j$  間の AL 経路上の  $i$  番目の IP リンクに対し、トランジットコストの増加量を決定する値  $v_i^j$  を下記のように与えたとき、

$$v_i^j = \begin{cases} 1 & (\text{トランジットリンクの場合}) \\ 0.05 & (\text{ピアリングリンクの場合}) \\ 0 & (\text{AS 間リンクではない場合}) \end{cases} \quad (16)$$

$\beta_j$  は次式で計算される。 $M_j$  は AL ノードペア  $j$  間の AL 経路に含まれる IP リンクの総数とする。

$$\beta_j = \sum_{i=1}^{M_j} v_i^j \quad (17)$$

なお、遅延時間、利用可能帯域、物理帯域の計測結果が得られない AL リンクに関しては、AL ルーティングにおいて用いないものとする。

#### 4.1.3 その他の設定

提案手法で用いるコスト関数が返す値は初期状態のコストで正規化を行う。他のパラメータの設定は表 1 の通りである。

遅延時間およびトランジットコストの計算を行うにあたり必要なトラフィック要求量は、[13] を参考として決定する。すなわち、100kbps の帯域を要求するトラフィックフローがワイブル分布に従う間隔で発生し、対数正規分布に従う持続時間を持つものとする。各フローの送信元および宛先は、全 AL ノードペアのうちからランダムに決定する。ただし、[13] に示されている各分布のパラメータは単一の AS のアクセスリンクで計測したものに基いているため、多数の AS を含む本稿のネットワークモデルでは妥当ではない。そこで、[13] に示されたパラメータを持つ分布に従うフロー群が多重して発生するとし、この時の重ねる数を多重度と呼ぶことにする。多重度の変化はすなわちトラフィックの総量の変化に相当する。本稿では、この多重度を 5 とする。なお、利用可能帯域を指標とした評価ではフロー数のみを考慮して帯域を割り当てる。

評価においては、上述のトラフィック要求が 10 秒間発生するとし、フローを 1 秒単位で集約し、これを 1 秒間におけるトラフィック要求量とする。すなわち、シミュレーション内時間で 10 秒間の評価を行い、トラフィック要求量の変動は 1 秒毎に全 10 回発生する。経路選択はこのトラフィック要求量の変動に応じて行われ、選択した経路への変更は瞬時に完了するものとする。

評価は、提案手法による経路制御を用いた場合と各 AL ノードが AL リンクの計測結果に基づいて独立して経路選択を行った場合と比較することで行う。また、提案手法として、トランジットコストの制限を行う場合 (式 (7), (8) における  $\alpha = 1$  の

表 1 パラメータ設定

AL ノード数	30
温度 $T_{init}$ および $T_{re}$	0.15
冷却関数	現温度 $\times 0.995$
近傍選択関数で変更する経路数	担当 AL 経路の 1%
AL 経路の最大ホップ数	2
通知間隔 $U$	20
終了温度 $F_{th}$	$10^{-6}$
再実行閾値 $C_{th}$	10

表 2 AL 経路あたりの平均遅延時間

提案手法を用いた場合	218 ms
独立して経路選択をした場合	204 ms

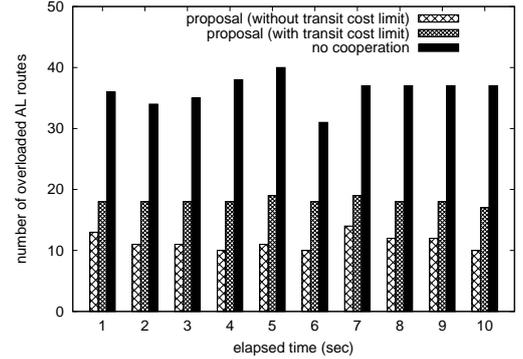


図 4 過負荷な AL 経路数の推移

場合)、および行わない場合 ( $\alpha = \infty$ ) を比較することで、トランジットコスト制限が経路選択に与える影響を評価する。性能評価指標として、経路制御の結果得られる AL 経路の遅延時間、利用可能帯域に加え、発生するトランジットコストおよび過負荷によって遅延時間が無限大に発散する AL リンクを用いている AL 経路 (以降、過負荷な AL 経路と呼ぶ) の数を比較する。これらにより、AL ノード間の協調がネットワーク性能の向上と輻輳を低減し、かつ、トランジットコストの削減に有効であるかを確認する。

#### 4.2 評価結果

表 2 および図 4 にそれぞれ、平均遅延時間を経路選択の指標とし、提案手法 (proposal) および各 AL ノードが独立して経路選択を行う比較手法 (no cooperation) を用いた場合における AL 経路の平均遅延時間と過負荷な AL 経路数の推移を示す。平均遅延時間は過負荷な AL 経路数を除いて計算した値である。図 4 には、提案方式としてトランジットコストを制限する場合としない場合の結果を示しているが、ここでは、基本となる制限しない場合の結果に着目する。これらの結果から、比較手法は、提案手法に比べて平均遅延時間はわずかに小さいものの、多数の過負荷な AL 経路が発生していることがわかる。対して提案手法では、比較手法に比べて、過負荷な AL 経路の選択をおよそ 70% 削減できている。これらの結果から、提案手法を用いることにより、各 AL ノードが独立して経路選択を行う場合に比べ、AL リンクの輻輳の発生を大きく軽減できることがわかる。

また図 4 から、比較手法においては、時間の推移と共に過負荷な AL 経路が増加する箇所があることがわかる。例えば図中の 2 秒から 5 秒にかけて、過負荷な AL 経路が増加している。この変化を詳細に観察した結果、過負荷な AL 経路が増える場合、経路が AL リンクを共有している複数のノードペアのトラフィック要求量が増加していることがわかった。比較手法においては、各 AL ノードは独立して、他ノードの経路選択を考慮せず、計測結果から良い性能を持つ経路を選択するため、過負荷な AL リンクが発生しやすくなる。一方、提案手法を用いた場合においては、トラフィックが増加しても互いに経路の重複を避け、過負荷を避けていることが観察された。このことから、Algorithm 2 で拡張したアルゴリズムがトラフィックの変動に対し有効に機能していることがわかる。

図 5 に、利用可能帯域を経路選択の指標として用いて、提案手法と比較手法における経路選択を行った結果、各ノードペア

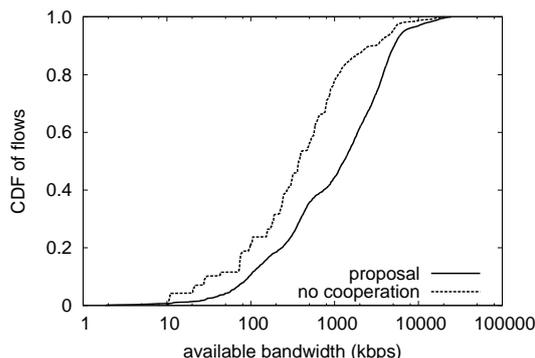


図 5 フローあたりの利用可能帯域の分布

表 3 フローあたりの平均利用可能帯域

提案手法を用いた場合	2285 kbps
独立して経路選択をした場合	1072 kbps

表 4 AL 経路あたりの平均トランジットコスト

トランジットコストを制限しない場合	4033
トランジットコストを制限した場合	3314

のトラフィックが獲得できた利用可能帯域の分布を示す。図から、提案手法を用いることで、ほぼ全てのノードペアにおいて、比較手法よりも大きな利用可能帯域を獲得できていることがわかる。また、それぞれの場合の平均利用可能帯域を表 3 に示す。表から、平均でおおよそ 110% の性能向上が得られることがわかる。これは、提案手法を用いることで経路の重複を大幅に抑えられたことによる結果である。

表 4 に、遅延時間を指標として経路選択を行う提案手法に関して、式 (6) で示したトランジットコストの制限を加えなかった場合と加えた場合の AL 経路あたりの平均トランジットコストの比較結果を示す。表から、式 (6) を制約条件として用いることによって、おおよそ 28% のトランジットコストを削減できていることがわかる。しかし、図 4 に示すように、トランジットコストの制限を行うことによって、過負荷な経路数が大きく増加している。これは、トランジットコストを制限することによって、経路選択において用いることができる AL リンク数が減少するためである。したがって、輻輳の回避を優先した制御を行うためには、トランジットコストの制限を緩めることが有効であると考えられる。

## 5. 議 論

実測データを用いた性能評価から、遅延時間を指標として経路選択を行う場合、提案手法によって経路の遅延時間そのものを大きく改善することはできないが、トラフィックが集中することによる AL リンクの過負荷を避けることが可能であることがわかった。このような提案方式の特性は、ISP にとってもエンドユーザにとっても有用であると考えられる。一方で、利用可能帯域を指標とした経路選択を行った場合、提案手法を用いることによって大きな性能改善を得ることができる。

近年、hyper giants と呼ばれる巨大なコンテンツプロバイダの登場により、インターネットのトポロジに変化が起りつつある [14, 15]。hyper giants は、トランジットリンクを利用することによる金銭的なコストを削減し、特定 ISP への依存度を下げるために、Internet eXchange (IX) を利用して直接ユーザを収容しているエッジ ISP とピアリングリンクを張る方針をとっている。これにより IX の利用が加速されれば、コンテンツプロバイダとだけでなく、エッジ ISP 間でのピアリングリンクの構築が容易となる。エッジ ISP 間にピアリングリンクが増加した場合、提案手法によってこれらのリンクを利用し合うことで、ネットワーク性能の向上およびトランジットコスト削減の効果は更なる有効性を発揮すると考えられる。

## 6. まとめと今後の課題

本稿では、トラフィックが動的に変動するネットワーク環境においてエンド間ネットワーク性能を向上させるために、分散型

焼きなまし法に基づいてアプリケーション層経路制御を行う手法を示した。具体的には、アプリケーション層経路最適化問題の定義を示し、この問題の目的関数に対応したコストを用いて経路制御を行う分散型焼きなまし法を基にした手法の提案を行った。また、PlanetLab 環境を想定した提案手法の性能評価により、動的なトラフィック変動に対して追従して輻輳を回避し、また、エンド間のネットワーク性能の向上およびトランジットコストの削減を行えることを示した。加えて、提案手法の利用形態および昨今のインターネットトポロジの変化による影響について議論した。

今後の課題として、遅延時間を指標とした際の輻輳回避と利用可能帯域を指標とした際の大幅な性能向上を組み合わせる手法を考案する。また、複数の ISP が協調して経路制御を行うといった具体的な利用形態を軸に、各 ISP の得られる利益の釣り合いやネットワーク情報の共有におけるポリシーなどを踏まえた手法の検討を予定している。

## 謝 辞

本研究の一部は、NICT 受託研究「パレート最適ネットワークのためのユーザ・資源管理アプリケーション」によっている。ここに記して謝意を示す。

## 文 献

- [1] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, "Construction of an efficient overlay multicast infrastructure for real-time applications," in *Proceedings of INFOCOM 2003*, Apr. 2003.
- [2] D. G. Andersen, A. C. Snoeren, and H. Balakrishnan, "Best-path vs. multi-path overlay routing," in *Proceedings of IMC 2003*, Oct. 2003.
- [3] C. L. T. Man, G. Hasegawa, and M. Murata, "Monitoring overlay path bandwidth using an inline measurement technique," *IARIA International Journal on Advances in Systems and Measurements*, vol. 1, pp. 50–60, Feb. 2008.
- [4] Y. Zhu, C. Dovrolis, and M. Ammar, "Dynamic overlay routing based on available bandwidth estimation: A simulation study," *Computer Networks Journal*, vol. 50, pp. 739–876, Apr. 2006.
- [5] R. Keralapura, N. Taft, C. nee Chuah, and G. Iannaccone, "Can ISPs take the heat from overlay networks," in *Proceedings of HotNets-III Workshop*, Nov. 2004.
- [6] K. Matsuda, G. Hasegawa, S. Kamei, and M. Murata, "Performance evaluation of a method to reduce inter-ISP transit cost caused by overlay routing," in *Proceedings of NETWORKS 2010*, pp. 250–255, Sept. 2010.
- [7] K. Matsuda, G. Hasegawa, S. Kamei, and M. Murata, "Centralized and distributed heuristic algorithms for application-level traffic routing," in *Proceedings of ICOIN 2012*, Feb. 2012.
- [8] M. Arshad and M. C. Silaghi, "Distributed simulated annealing and comparison to DSA," in *Proceedings of the Fourth Workshop on DCR*, Aug. 2003.
- [9] J. Hromkovic, *Algorithemics for Hard Problems*. Springer, 2005.
- [10] Hewlett-Packard Laboratories Scalable Sensing Service. available at <http://networking.hp.com/s-cube/>.
- [11] University of Oregon Route Views Project. available at <http://www.routeviews.org/>.
- [12] University of California CAIDA. available at <http://www.caida.org/home/>.
- [13] M. Pustisek, I. Humar, and J. Bester, "Empirical analysis and modeling of peer-to-peer traffic flows," in *Proceedings of MELECON2008*, pp. 169–175, May 2008.
- [14] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," *SIGCOMM Comput. Commun. Rev.*, vol. 40, pp. 75–86, Aug. 2010.
- [15] S. Yuval and W. Udi, "Topological trends of Internet content providers," *CoRR*, vol. abs/1201.0834, 2012.