

## 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  
<sup>††</sup>大阪大学 サイバーメディアセンター

### 研究背景

- アプリケーション層経路制御
  - エンド端末間ネットワーク性能指標 (遅延時間、利用可能帯域等) を用いて経路選択
  - 他エンド端末を経由する経路によりネットワーク性能向上の余地がある
  - IP 層の経路制御はルーター・AS ホップ数、及び ISP 間の契約関係によって経路選択が行われるため

- アプリケーション層リンク (AL リンク)
  - アプリケーション層経路制御のノード (AL ノード) 間にはられる仮想的なリンク
- アプリケーション層経路 (AL 経路)
  - 1 本以上 AL リンクを用いる AL ノード間の経路
  - 1 ホップの経路を直接経路、それ以外の経路を迂回経路と呼ぶ

12/07/19 2012/7 IN 研究会 2

### アプリケーション層経路制御の問題点

- 他の AL ノードの経路選択状況を考慮せずに経路を決定すると AL リンクを複数の AL 経路で共有する可能性がある
  - 遅延時間の増加や利用可能帯域の低下等、AL 経路の性能の劣化がおこる
- IP 層の経路制御と比べて通過する IP リンクが増加する可能性がある
  - 利用するトランジットリンクが増加しネットワーク全体のトランジットコストが増大する

A-B 間で E-F 間の AL リンクを使用しているとき、C-D 間がこれを考慮せずに経路を選択すると E-F 間のリンクを共有する

A-B 間の IP 層経路制御による経路と比べ A-E-F-B という経路は通過するトランジットリンクの数が増加する

12/07/19 2012/7 IN 研究会 3

### 研究の目的

- 他ユーザの経路選択状況を考慮しないアプリケーション層経路制御はユーザ性能 (遅延時間、利用可能帯域など) が劣化する可能性がある
- アプリケーション層で複数ホップの経路を用いることにより、利用するトランジットリンクが増加し、トランジットコストが増加する可能性がある

↓

他 AL ノードの経路選択状況を考慮してネットワーク性能の向上やトランジットコストの削減を得られる経路選択を行うアプリケーション層経路制御手法の提案

- アプリケーション層経路選択を最適化問題として定義する
- 定義した最適化問題を分散型発見的手法で解くアプリケーション層経路制御手法を提案する
  - 複数の ISP に分散した AL ノードが独立して処理を行う状況を想定

12/07/19 2012/7 IN 研究会 4

### ネットワークモデル

- AS に属するエンドホストが AL ノードとなり、AL ネットワークを構築
- AL リンクは IP 層ネットワークにおけるエンド間の経路に相当する

12/07/19 2012/7 IN 研究会 5

### アプリケーション層経路最適化問題

- ユーザ性能を目的関数とする最適化問題
- AL ノードペア間の遅延時間の平均値を最小化
- トランジットコストに関する制約条件を含める

$$\text{minimize : } \left( \sum_{j \in K} d_j^{AL} \right) / |K|$$

$$\text{subject to : } r_j \in R_j^{AL}$$

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

1 で直接経路以下 無限大で制限なし

- $K$ : 通信要求を持つノードペアの集合
- $R_j^{AL}$ : AL ノードペア  $j$  が利用できる経路の集合
- $d_j^{AL}$ : AL ノードペア  $j$  間 AL 経路の遅延時間
- $r_j$ : AL ノードペア  $j$  が利用する AL 経路
- $c_j^{AL}$ : AL ノードペア  $j$  間 AL 経路のトランジットコスト
- $c_j^{DR}$ : AL ノードペア  $j$  間直接経路のトランジットコスト
- $\alpha$ : トランジットコスト制限の程度を決める定数

- 各 AL ノードペア間の遅延時間が相互に影響を及ぼさない場合は、それぞれの遅延時間を最小化する問題に等しい
- 実際には各 AL ノードペアが利用する AL パスの重なりにより、遅延時間 ( $d_j^{AL}$ ) が相互に影響する
- 総当たりで解くことが現実的な時間では不可能になるため、発見的手法を用いる

12/07/19 2012/7 IN 研究会 6

### 発見的手法に基づくAL 経路制御手法

- ・焼きなまし法を利用して最適化問題の制約条件を満たす解を求め、アプリケーション層経路制御における経路選択に適用
- ・アプリケーション層経路制御におけるトラヒック (AL トラヒック) が不在状態の AL リンクの各種ネットワーク性能を計測によって取得
- ・各 AL ノードペアの AL トラヒック要求量は取得できるとする
- ・1 ホップの AL 経路 (直接経路) を全 AL ノードペアが選択した状態を初期状態として焼きなましを開始
  - ・ある状態の近傍状態は、その状態の一部の AL ノードペアに関してランダムに経路を変更した状態とする
- ・ある状態に関して、その経路選択で得られるネットワーク性能を推定する
  - ・ネットワーク性能の推定は、計測した AL リンクの性能と AL トラヒック要求量から行う
- ・推定したネットワーク性能をコストとし、これにもとづいて状態遷移を行う
- ・一般的な焼きなまし法と同様、与えた温度と冷却関数により一定回数の焼きなましを得て終了

### AL 経路制御手法における分散焼きなまし法の利用

- ・分散焼きなまし法 (Distributed Simulated Annealing) [8] を利用
  - ・各 AL ノードは、独立して経路選択を実行
  - ・各 AL ノードは自身を送信元とする AL リンクのネットワーク性能のみを計測
  - ・各 AL ノードは自身を送信元とする AL 経路のみを変更したものを近傍状態とする
  - ・各 AL ノードは他 AL ノードと通信を行い、ネットワーク性能計測結果、AL 経路の選択状況、AL トラヒック要求量の情報を共有する
    - ・取得した他ノードの情報をういて焼きなましに用いる状態を更新し、コストを計算する
    - ・一度の状態遷移判断毎に情報交換を行うと通信オーバーヘッドが大きいため、一定回数毎に一度だけ情報交換を行う

[8] M. Arshad and M. C. Silaghi, "Distributed simulated annealing and comparison to DSA," in Proceedings of the Fourth Workshop on Distributed Constraint Reasoning, Aug. 2003.

### 分散焼きなまし法に基づくAL 経路制御手法

- ・焼きなまし法を経路選択に適用し、分散処理の仕組みを追加

- ・交換した情報を元に現在の状態を更新する

- ・自分の担当する部分のみを一部変更して近傍状態を作成する
  - ・本手法では、自身を送信元とする経路のみを変更する

- ・定期的を選択経路およびトラヒック要求量を他ノードに送信する

```

Algorithm 1 静的なトラヒックに対する AL 経路選択手法
1:  $T_i \leftarrow 0, T_i \leftarrow T_{init}, S_i \leftarrow S_{init}$ 
2: while  $T_i > T_{th}$  do
3:   Update( $S_i$ )
4:    $S_{temp} \leftarrow \text{Neighbor}_{\text{best}}(S_i)$ 
5:   if  $\text{Cost}(S_i) \geq \text{Cost}(S_{temp})$  then
6:      $S_i \leftarrow S_{temp}$ 
7:   else
8:      $r_i \leftarrow \text{Random}(0, 1)$ 
9:     if  $r_i < \text{Probability}(T_i, \text{Cost}(S_i), \text{Cost}(S_{temp}))$  then
10:       $S_i \leftarrow S_{temp}$ 
11:   end if
12: end if
13:  $I_i \leftarrow I_i + 1$ 
14:  $T_i \leftarrow \text{Cooling}(T, I_i)$ 
15: if  $I_i \bmod U_i = 0$  then
16:   SendNeighbor( $S_i$ )
17: end if
18: end while
    
```

### AL トラヒック要求量の変動に対応する拡張

- ・AL トラヒック要求量の変動に継続的に対応できるように拡張

- ・静的なトラヒックに対するアルゴリズムを実行

- ・トラヒック要求量の変化をカウント

- ・閾値を設定し、これを越えた際に再度温度を上昇させ、静的なトラヒックに対するアルゴリズムを再実行
  - ・自身を送信元とするトラヒック要求が発生した場合は即時、そうでない場合は  $C_{th}$  回の変化で再実行
  - ・再実行の初期状態は、前回の経路選択の結果を用いる

```

Algorithm 1 AL トラヒック要求量の変動に対応した拡張
1:  $T_i \leftarrow T_{init}$ 
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
    
```

### AL リンク・経路のネットワーク性能推定方法

- ・遅延時間
  - ・AL ノードペア間にトラヒック要求量に対し、物理帯域とリンク利用率を用いて M/M/1 待ち行列モデルに基づいてキューイング遅延時間を算出する
  - ・キューイング遅延時間と計測によって得た伝搬遅延時間の和を AL リンクの遅延時間とする
  - ・AL 経路を構成する各 AL リンクの遅延時間の和をその AL 経路の遅延時間とする
- ・利用可能帯域
  - ・トラヒック要求を持つ AL ノードペア間で計測によって取得した AL リンクの利用可能帯域を max-min アルゴリズムに従って均等に共有する
  - ・AL 経路を構成する各 AL リンクの利用可能帯域のうち最小のものをその AL 経路の利用可能帯域とする
- ・トランジットコスト
  - ・AL リンクに含まれる AS 間リンクを同一 ISP 内リンク、ピアリングリンク、トランジットリンクに分類し、それぞれに設定した係数の和を単位トラヒックあたりのトランジットコストとする
  - ・AL ノードペア間にトラヒック要求量に対し、通過する AL リンクの係数の和とトラヒック要求量の積を AL 経路のトランジットコストとする

### 性能評価

- ・PlanetLab ノードがアプリケーション層経路制御を用いることを想定し、以下のノード間情報を用いて評価を行う

- ・遅延時間, IP 層経路
  - ・PlanetLab ノード間で traceroute を行うことで取得
- ・利用可能帯域, 物理帯域
  - ・S-cube [10] が公開しているデータを利用
- ・AS レベルの経路
  - ・上記で取得した IP 層経路に Route Views Project [11] で公開しているアドレスプレフィックスと AS 番号の対応データを当てはめて取得
- ・AS 間のリンク種別
  - ・上記 traceroute の結果と CAIDA [12] が公開している AS 間関係情報 (トランジットリンク, ピアリングリンク) を用いて取得

[10] Hewlett-Packard Laboratories Scalable Sensing Service, available at <http://networking.hpl.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/>.

### 評価の設定

- ・コスト関数
  - ・トラフィック要求を持つ AL 経路の平均遅延時間, または平均利用可能帯域
- ・AL トラフィックを除いた状態の AL リンクのネットワーク性能は, 全 AL ノードにおいて計測・共有済みとする
- ・近傍選択関数
  - ・対象とする AL 経路 (自身を送信元とする経路) のうち 1% の経路を変更したものを近傍状態とする
- ・情報交換の頻度
  - ・焼きなましの反復処理 20 回につき 1 回, 選択した AL 経路および AL トラフィック要求量の情報を交換する
- ・AL トラフィック要求量
  - ・遅延時間を指標とした場合は AS 間リンクの計測データ [14] を参考に設定
    - ・パラメータを変更し, シミュレーション内時間で 2 秒毎に変動が起こるとした
  - ・利用可能帯域を指標とした場合は一定割合のノードペアがトラフィック要求を持つと想定
    - ・10%, 80%, 40% の順に 2 秒毎で割合が変動するとした

[14] C. Lebovitz, 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.

2012/7 IN 研究会 13

### 評価結果: 遅延時間を経路選択指標とした場合

平均遅延時間

トラフィック 1 3 10 5

過負荷経路数

- ・ランダムに選んだ PlanetLab ノード 30 ノードを AL ノードとし, その性能を評価
  - ・比較手法 (non-cooperation method) は, ノードペア間で協調せず, それぞれが独立して AL リンクの計測結果から最小の遅延時間をもつ経路を選択することした
  - ・平均遅延時間, および利用可能帯域を越えたトラフィック要求量により過負荷となる経路の数を評価指標とした

平均遅延時間は, 提案手法が比較手法に比べて僅かに大きい

過負荷な経路を除いた平均を導出しているため

過負荷な経路の数は, 提案手法により大きな削減効果を得られている

12/07/19 2012/7 IN 研究会 14

### 評価結果: 遅延時間を経路選択指標とした場合 経路変更の様子

- ・提案手法と比較手法の経路変更の様子を比較
  - ・選択した AL 経路が利用している AL リンクに関して, 他ノードペアが重複して利用している数および AL リンクの物理帯域に対するトラフィック量の割合を示す
    - ・例) (1, 0.81) は, 重複利用数 1 で, 物理帯域のうちトラフィックが 81% を占めることを表す

Table 1: Samples of changes in selected AL routes

and traffic value	proposed method	non-cooperation method
2 sec	planetlab1.lf.umitsc.it (1, 0.81)	planetlab1.lf.umitsc.it (2, 0.84)
300 kbps	ricepl-1.cs.rice.edu (1, 0.65)	ricepl-1.cs.rice.edu (3, 1.39)
3 sec	planetlab1-ids.umitsc.it (2, 0.12)	planetlab2.cs.columbia.edu (2, 0.08)
1600 kbps		

2 秒時点では, 両手法とも直接経路を選択している

3 秒時点では, トラフィック要求量の増加に伴い経路を変更

- ・提案手法では, 経路の重複を避け, 利用率の低い経路を利用しているのに対し, 比較手法では重複が多く利用率が 1 を越えている

提案手法を用いることにより, 重複を避け, 利用率の低い AL リンクを用いた経路を選択することができ AL リンクにおける輻輳を回避できる

12/07/19 2012/7 IN 研究会 15

### 評価結果: 遅延時間を指標とした場合 トラフィックコストの制限を加えた評価

Table 1: Average value of transit cost of the AL routes

proposed method ( $\alpha=0.7$ )	proposed method ( $\alpha=1$ )
5,186	4,260

直接経路以下という制限を加えることにより 制限なしの場合に比べてトラフィックコストを 20% 程度削減できる

過負荷経路数

トラフィック 1 3 10 5

- ・制限なしの場合より, 過負荷な AL 経路の数が少数ではあるが増加する
- ・トラフィックコストの制限により 利用できる AL リンクの数が増加するため
- ・輻輳を避けることに着目すると厳しすぎるトラフィックコスト制限は避けたほうが良い

12/07/19 2012/7 IN 研究会 16

### 評価結果: 利用可能帯域を経路選択指標とした場合

平均利用可能帯域

トラフィック要求を持つノードペアの割合

10% 80% 40%

- ・ランダムに選んだ PlanetLab ノード 30 ノードを AL ノードとし, その性能を評価
  - ・比較手法 (non-cooperation method) は, ノードペア間で協調せず, それぞれが独立して AL リンクの計測結果から最大の利用可能帯域をもつ経路を選択することした
  - ・各秒における平均利用可能帯域を評価指標とした

すべての時刻において, 提案手法により 比較手法と比べて高い性能を得られている

トラフィック要求があるノード数の割合を変更しても 性能の変化が少ない

遅延時間の場合と同じく経路の重複を避けた経路変更を行なっているため

12/07/19 2012/7 IN 研究会 17

### まとめと今後の課題

- ・まとめ
    - ・エンド間性能の向上やトラフィックコストの削減を得られる経路を選択するアプリケーション層経路制御手法を提案
      - ・分散焼きなまし法に基づき, 各ノードが独立して経路制御を実行
      - ・トラフィック要求量変動する環境を想定して, トラフィック要求量の変動に対し継続的に経路選択を行えるよう拡張
    - ・PlanetLab 環境を想定した評価により提案手法の性能を確認
      - ・遅延時間を指標とすることにより, AL リンクの輻輳を大幅に回避できる
      - ・利用可能帯域を指標とすることにより, 大幅に利用可能帯域を向上できる
  - ・今後の課題
    - ・遅延時間を指標とした場合の輻輳回避と利用可能帯域を指標とした場合の帯域向上を組み合わせた経路制御の考案
    - ・提案手法を複数の ISP が協調して利用するシナリオを想定しての, 情報共有のポリシーや ISP 間の利益調整の仕組みの考案
- 12/07/19 2012/7 IN 研究会 18