

モデル予測制御を用いた階層型トラヒックエンジニアリング

大歳 達也[†] 大下 裕一[†] 村田 正幸[†] 高橋 洋介^{††} 石橋 圭介^{††}
塩本 公平^{††} 橋本 智昭^{†††}

[†] 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

^{††} 日本電信電話株式会社 NTT ネットワーク基盤技術研究所 〒180-8585 東京都武蔵野市緑町 3-9-11

^{†††} 大阪大学 大学院基礎工学研究科 〒560-8531 大阪府豊中市待兼山町 1-3

E-mail: [†]{t-otoshi,y-ohsita,murata}@ist.osaka-u.ac.jp,

^{††}{takahashi.yousuke,ishibashi.keisuke,shiomoto.kohei}@lab.ntt.co.jp, ^{†††}thashi@sys.es.osaka-u.ac.jp

あらまし 時間変動の大きなトラヒックに対して、大規模な経路変更の頻発を避けながら、変動に追従するためには、将来の変動傾向を予測しながら、それを考慮した経路変更が必要となる。我々の研究グループでは、モデル予測制御 (MPC) の考え方を適用し、予測誤差が含まれる場合であっても、トラヒックを適切に収容する経路を設定可能なトラヒックエンジニアリング (TE) 手法を提案している。しかし、この手法では、各制御時点において、ネットワーク全体のトラヒック情報の収集、予測、経路計算を行うため、ネットワークの規模が大きくなった際に、そのスケーラビリティが課題となる。この問題を解決するために、ネットワークを階層的に複数の範囲に分割し、分割された各範囲でトラヒック予測及び、経路制御を行うことで、制御負荷を削減しつつ全体の経路制御を実行する手法を提案する。また、シミュレーションにより、提案手法を用いることによって、ネットワーク規模が増大した場合でも計算時間の増大を抑えつつ、最適な経路に近い経路を設定可能であることを示す。

キーワード モデル予測制御, 階層型ルーティング, トラヒックエンジニアリング, トラヒック予測

Hierarchical Traffic Engineering Based on Model Predictive Control

Tatsuya OTOSHI[†], Yuichi OHSITA[†], Masayuki MURATA[†], Yousuke TAKAHASHI^{††}, Keisuke ISHIBASHI^{††}, Kohei SHIOMOTO^{††}, and Tomoaki HASHIMOTO^{†††}

[†] Graduate School of Information Science and Technology, Osaka University 1-5 Yamadaoka, Suita, Osaka, 565-0871, Japan

^{††} NTT Network Technology Laboratories, NTT Corporation 3-9-11, Midori-Cho, Musashino-Shi, Tokyo, 180-8585 Japan

^{†††} Graduate School of Engineering Science, Osaka University 1-3 Machikaneyama-Cho, Toyonaka, Osaka, 560-8531, Japan

E-mail: [†]{t-otoshi,y-ohsita,murata}@ist.osaka-u.ac.jp, ^{††}{takahashi.yousuke,ishibashi.keisuke,shiomoto.kohei}@lab.ntt.co.jp,

^{†††}thashi@sys.es.osaka-u.ac.jp

Abstract Traffic engineering with traffic prediction is one approach to accommodate time-varying traffic stably. In this approach, the routes are calculated so as to avoid congestion based on the predicted traffic. To achieve the robustness against the prediction error, we proposed a traffic engineering method called Model Predictive Traffic Engineering (MP-TE) which is based on the idea of Model Predictive Control (MPC). However, this method has difficulty in scalability. In this method, a central control server repeatedly collects the whole traffic information to correct the traffic prediction and recalculate the whole routes at each time slot. In accordance, load on the central control server significantly increases as the network becomes larger. To solve the scalability problem, we propose a prediction-based hierarchical traffic engineering method in this paper. In this method, we divide the network in multiple areas and aggregate the topology in each area as an upper layer topology. Control servers are deployed in each area and upper layer to calculate the routes inner area and inter area, respectively. By reducing the scale of the topology which one control server manages, the load on each control server is kept low even when the scale of network become large. Through the simulation, we show that the proposal method can reduce the calculation time while the achieved performance is close to that of the centralized control.

Key words Model Predictive Control, Hierarchical Routing, Traffic Engineering, Traffic Prediction

1. はじめに

近年、ストリーミング配信やクラウドサービス等のインターネットを介したサービスが普及するにつれて、ネットワークを流れるトラフィックの時間変動は大きくなっている。バックボーンネットワークは、大きなトラフィック変動が生じた場合にも、輻輳を生じることなく全トラフィックを収容する必要がある。

時間変動のあるトラフィックを収容する単純な方法としては、トラフィック需要に対してリンクの帯域を過剰に増設するオーバープロビジョニング [1,2] がある。オーバープロビジョニングでは、一時的なトラフィックの増加や、ネットワーク機器の故障等に備え、通常時のトラフィックの2倍以上のリンク帯域を用意することが一般的に行われている。この場合、不必要に大きな帯域を用意するため、過剰な設備投資コストを要する。また、近年、リンク利用率の低い不要なポートの電源をオフにすることで、消費電力を削減する等 [3]、限られたリンク帯域でトラフィックを収容することが求められてきている。

ネットワークの資源を効率的に利用することで、限られた資源下においても、輻輳を生じることなくトラフィックを収容する技術として、トラフィックエンジニアリング (TE; Traffic Engineering) と呼ばれる手法に関する研究が進められている [4-7]。これらの研究では、トラフィック変動は定期的なトラフィック観測により取得し、各時刻で観測されたトラフィックに合致した経路設定に移行する手法が検討されてきた。しかしながら、これらの手法では、観測時点に合わせた経路を設定するのみであり、経路変更後に発生したトラフィック変動に対応できず、輻輳を生じる可能性がある。短い制御周期で経路変更を行うことを可能とする手法 [4,5] も検討されているものの、頻繁な経路変更はネットワークの不安定化を招き、TCP スループットの低下及び、遅延・パケットロスの増加につながる。

このような問題を解消するために、我々の研究グループでは、トラフィック予測と連携し、予測された将来のトラフィック変動を考慮した TE 制御である MP-TE (Model Predictive Traffic Engineering) を提案している [8-10]。MP-TE はモデル予測制御 (MPC; Model Predictive Control) と呼ばれるシステム制御の考えに基づいて TE 制御を行うものである。この手法では、制御サーバーが過去に観測されたトラフィック情報を基に、将来の数期先までに到着するトラフィックの時系列を予測し、それぞれの時刻において、トラフィック予測値を収容可能な経路を求める。そして、求めた経路のうちで最も直近の時刻に対応する経路のみを実際にネットワークに設定し、新たなトラフィックの観測を基に予測の修正を行い、再び経路計算を行うことで、以降の経路を設定する。これにより、トラフィックの変動傾向が変化した場合にも、それに応じた経路変更が可能となる。また、経路計算時に、各タイムスロットで行う経路変更を抑えることで、一時的な予測誤差に対して過剰に反応することを避けることができる。

しかしながら、上述の方法では、タイムスロット毎にネットワーク上のすべてのトラフィックの観測とその予測を行い、全体の経路計算を繰り返す必要があり、扱うネットワークの規模が大きくなった場合、タイムスロット内に経路計算を完了できず、制御が破綻する可能性がある。従来、経路制御のスケーラ

リティを向上させる方法として、ネットワークを複数のエリアに分割し、階層的に経路制御を行うことが検討されてきている [11-13]。この場合、各エリアに制御サーバーを配置し、それらが局所的な情報のみを用いてエリア内の経路制御を行う。また、エリア間の経路制御は、制御サーバーがネットワーク全体の情報を用いるのではなく、エリア毎に集約された情報のみを用いて経路の計算を行う。これによって、単独の制御サーバーに負荷が集中することを回避し、スケラビリティの向上が可能となる。そこで、本稿では、MP-TE を階層型経路制御として拡張することで、大規模なネットワークでも適用可能な予測型経路制御手法を提案する。

以降の本稿の構成は次の通りである。2章では、MP-TE の概要について述べ、大域的な経路制御の最適化問題として MP-TE の定式化を行う。3章では、提案手法における階層的なネットワークの分割及び集約方法について述べた後、各階層での経路制御手法について述べる。4章では、提案手法の有効性をシミュレーションにより評価する。5章では、まとめと今後の課題について述べる。

2. モデル予測制御を用いたトラフィックエンジニアリング

2.1 MP-TE の概要

MP-TE [8-10] は、一定のタイムスロット間隔毎に、将来到着すると予測されるトラフィックレートに合わせて、ネットワーク内の経路を動的に変更することにより、輻輳なくトラフィックを収容する手法である。この時、各タイムスロットで得られた新たなトラフィックの観測結果を基に予測を修正し直すことで、トラフィックの変動傾向の変化により、予測誤差が拡大することを避ける。また、各タイムスロットにおいて、著しい経路変更を避けることで、一時的な誤差の影響で大きく誤った経路変更を行うことを避け、誤差が生じた場合でも、大きく輻輳を生じない経路設定を行う。

集中管理型で、ネットワーク全体の情報を用いて大域的な最適化を行う場合、制御サーバーは各タイムスロットにおいて、全 Origin-Destination (OD) 間のトラフィック量を観測し、その将来的なトラフィック量の変化を予測する。そして、予め与えられた OD 間を結ぶ複数の経路候補に対して、OD 間のトラフィックを各経路候補に分割して流す割合を、OD 間のトラフィックの予測値に応じて決めることにより、将来的に到着が予想されるトラフィックを収容可能な経路を計算する。

以降、タイムスロット k における全 f 本の OD フローのトラフィック量をベクトル $\mathbf{x}(k) = {}^t(x_1(k), \dots, x_f(k))$ で表し、その予測値を $\hat{\mathbf{x}}(k)$ で表すこととする。予測値は、直前のスロット t までにおいて得られたトラフィック情報を用いて対象となる予測区間 $[t+1, t+h]$ における各スロットのトラフィック量を予測したものとするが、その予測方法については特定の予測方法を仮定せず、任意の予測方法を用いることができる。また、タイムスロット k において、OD フロー j のトラフィックのうち、経路 i に送出されるものの割合を $R_{i,j}(k)$ と表す。さらに、 $R_{i,j}(k)$ を (i, j) 要素とする $p \times f$ の行列を $R(k)$ と表すこととする。この時、 $R(k)$ で定められた割合に従って、トラフィック量 $\mathbf{x}(k)$ の

全 OD フローをネットワーク内で転送する時、各リンク上を流れるトラフィック $\mathbf{y}(k) = {}^t(y_1(k), \dots, y_l(k))$ は、

$$\mathbf{y}(k) = G \cdot R(k) \cdot \mathbf{x}(k) \quad (1)$$

と表される。ここで、 l は全リンクの本数を表し、 G は各経路候補が経由するリンクの対応関係を表す $l \times p$ 行列で、その (i, j) 成分は、経路候補 j がリンク i を経由するときに 1 それ以外は 0 の値を取る。MP-TE では、 $\mathbf{y}(k)$ が各リンクにおいて目標としている帯域 $\mathbf{C} = {}^t(C_1, \dots, C_l)$ を超えないように $R(k)$ を設定することで、輻輳を回避する。

2.2 大域的最適化問題としての定式化

ここで、集中管理型 MP-TE での経路制御を大域的な最適化問題として定式化しておく。MP-TE の制御の目標は、リンク上のトラフィックを目標帯域内に収容することである。すなわち、各リンクで目標帯域から超過したトラフィック量 $\zeta(k) = {}^t(\zeta_1(k), \dots, \zeta_l(k))$ を最小化することが目的となる。この時、予測誤差の影響を受けて、著しく誤った経路変更を行うことを回避するため、各タイムスロットでの経路変更量 $\Delta R(k) = R(k) - R(k-1)$ も同時に最小化する必要がある。従って、MP-TE はそれらの重み付け和 $(1-w)\|\zeta(k)\|^2 + w\|\Delta R(k)\|^2$ について、予測の対象としている区間 $[t+1, t+h]$ での総和を最小化する問題として、次のように定式化される。

$$\text{minimize : } \sum_{k=t+1}^{t+h} ((1-w)\|\zeta(k)\|^2 + w\|\Delta R(k)\|^2) \quad (2)$$

$$\text{subject to } \forall k, \hat{\mathbf{y}}(k) = G \cdot R(k) \cdot \hat{\mathbf{x}}(k) \quad (3)$$

$$\forall k, l, \zeta_l = \max(\hat{y}_l - C_l, 0) \quad (4)$$

$$\forall k, \forall i, \forall j, R_{i,j}(k) \in [0, 1] \quad (5)$$

$$\forall k, \sum_{i \in \rho(j)} R_{i,j}(k) = 1 \quad (6)$$

ここで、 $\hat{\mathbf{y}}(k)$ は、式 (1) の関係式と OD フローのトラフィックの予測値に基づいた、リンク上のトラフィックの予測値を表している。また、 $\|\cdot\|$ はユークリッドノルムを、 $0 \leq w \leq 1$ は経路変更に対する重みを表し、 w が大きい時、より現状の経路を維持した経路制御を行う。上述の最適化問題の解は、予測対象区間全体に渡る経路の割り当て $R(t+1), \dots, R(t+h)$ であるが、実際にネットワークに投入するのは、直近の $R(t+1)$ のみであり、以降の設定は、新たな観測地を基に予測値 $\hat{\mathbf{x}}(k)$ を修正し、最適化問題を再計算することで決定する。

この最適化問題は、文献 [9,10] と同様にして、凸二次計画問題に置き換えることができる。凸二次計画問題の計算量は、内点法を用いた場合に最悪時で $O(m^{3.5}L)$ である [14]。ここで、 m は制約式の数を表し、 $L(> m^2)$ は問題の入力長を表している。今、ネットワーク内のノード数を n とすると、最悪の場合、 $m = O(n^4)$ であるため、ノード数の増加に応じて計算時間は爆発的に大きくなる。また、MP-TE では、経路計算のみならず、トラフィック情報の収集と、その予測についてもネットワーク中の全 OD フローについて実行する必要があり、計算時間の増大は特に深刻な問題となる。従って、大規模なネットワークにおいては、要求されるタイムスロット内に経路計算を完了することができず、制御が破綻する可能性がある。

3. MP-TE の階層型経路制御への拡張

集中管理型の MP-TE のスケーラビリティの課題を解決するため、制御の対象としているネットワークを複数の連結したエリアに分割し、各エリアとエリア間の階層的な経路制御として、MP-TE を拡張した手法を提案する。この手法では、エリア内での経路制御は物理的なトポロジーを扱った下位層での制御として行い、エリア間の経路制御は、下位層の各エリアを集約した上位層のトポロジー上で行う。はじめに、ネットワークの階層化の方法について述べた後、各階層での経路制御手法について述べる。

3.1 ネットワークの階層化

3.1.1 エリアの分割

エリアの分割方法について、特定の分割方法は仮定しないが、各エリアは連結したネットワークであり、分割前のネットワーク上の各ノードは、いずれか一つのエリアにのみ属するような分割が得られているものとする。この時、エリアの境界に位置するリンクに関しては、どのエリアの内部にも含まれない場合が生じ得るが、このようなリンクは上位層のトポロジーで扱うものとする。

3.1.2 トポロジーの集約

上位層から、下位層の一つのエリアを見た場合に、そのエリア内のトポロジーをどのように集約するかについていくつかの方法が考えられる。最も単純には、エリアを一つのノードあるいはリンクとしてみなす方法があるが、これは上位層が扱う情報を大きく削減することが可能であるものの、必要な情報まで欠損する可能性が高い。他の方法としては、エリアのすべての境界ノードペア間をフルメッシュのリンクで接続したトポロジーに集約することで、リンク数は境界ノード数を B として $O(B^2)$ となるが、少ない情報欠損での集約が可能となる。また、文献 [13] では、スター型のトポロジーと数本の境界ノード間の直接リンク等を用いて、情報の欠損を抑えながら、 $O(B)$ 程度のサイズのトポロジーに集約することが検討されてきた。

分割の階層数または、エリアの分割数を増やすことで、一つのコントローラーが扱う境界ノード数 B は十分に小さく抑えられるため、集約の抽象度を高めて、情報を削減する必要性はないと考えられる。従って、本稿では情報の欠損が少ない境界ノードペア間にフルメッシュの仮想リンクが張られたトポロジーに集約する方法を用いることとする。また、上位層では、下位層のエリアの境界に存在している物理的なリンクについても扱うものとする。図 1 に上位層における下位層のトポロジーの集約例を示す。

上位層で経路制御を行う場合、下位層において混雑したエリアを避けるような経路を設定する必要がある。そのため、下位層のエリア内のトポロジーを抽象化した仮想リンクでは、そのエリアの混雑度を反映した目標帯域を設定する必要がある。基本的に、輻輳を回避するためには、そのエリア内でのポトルネックリンクの混雑度が小さいエリアを経由するように、経路を決定すればよい。しかし、一般に、一組の境界ノード間には複数の経路が存在し、実際にどの経路を経由するかは下位層の経路制御に委ねられる。そのため、複数本の経路の混雑度を一つのリンクの混雑度として集約する際に、どのように集約する

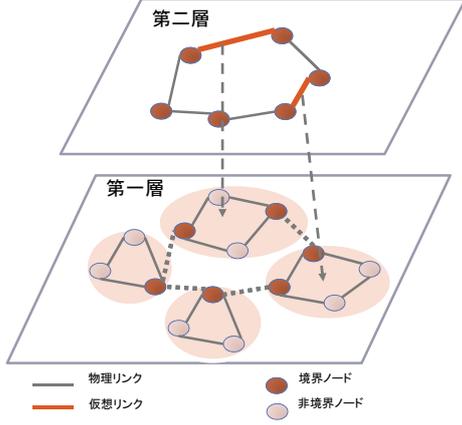


図1 トポロジーの集約
かが問題となる。ここでは、単純な方法として、境界ノード間に存在するすべての経路について最大の空き帯域を仮想リンクの目標帯域として設定することとする。

3.2 各層での経路制御

3.2.1 上位層での経路制御

上位層では下位のエリアをまたぐトラヒック $\mathbf{x}^{up}(t)$ について、それが、どの境界ノードを経由して各エリアを通過するかを決定する。

エリア i に含まれるノードの集合を A_i とすると、エリア i, j 間に流れるトラヒック $x_{i,j}^{up}(t)$ は

$$x_{i,j}^{up}(k) = \sum_{s \in A_i} \sum_{d \in A_j} x_{f(s,d)}(k) \quad (i \neq j) \quad (7)$$

となる。同一エリア ($i = j$ の場合) に流れるトラヒックについては上位層では扱わず、下位層で扱う。 $f(s, d)$ はノード s から d に向かうフローの識別子を表す。 $\mathbf{x}^{up}(k)$ は、この $x_{i,j}^{up}(k)$ を要素として並べたベクトルである。上位層では、タイムスロット t までに観測された $\mathbf{x}^{up}(k)$ を基に、将来到着されると予想されるトラヒック $\hat{\mathbf{x}}^{up}(t+1), \dots, \hat{\mathbf{x}}^{up}(t+h)$ を予測し、これを経路制御に用いる。

上位層で用いる経路は、境界ノード間の物理的、または仮想的なリンクの集合として定義される。物理的なリンクは、もともとの目標帯域がそのまま適用されるが、仮想的なリンクは、前述の通りエリアの混雑度を集約した目標帯域を持っており、リンク i の目標帯域 $C_i^{up}(k)$ は、

$$C_i^{up}(k) = \begin{cases} C_i & (\text{物理リンク}) \\ \max_{p \in P(i)} \min_{l \in L(p)} (C_l - y_l^{low}(k)) & (\text{仮想リンク}) \end{cases} \quad (8)$$

と表される。ここで、 $P(i)$ は仮想リンクとして集約されている境界ノード間のすべてのパスの集合であり、 $L(p)$ はパス p を構成しているエリア内のリンクの集合である。また、 $y_l^{low}(k)$ はリンク l 上のトラヒックのうち、下位層で閉じたフローのトラヒックのみを取り出したもので、上位層自身が流すトラヒックは除いたものである。経路を計算する際には、直前の時刻 $t-1$ における空き帯域が観測されているものとして、その値を時刻 t における目標帯域として用いる。

利用可能な経路が与えられたもとの、それぞれの経路にどの割合でトラヒックを割り当てるかを、次の最適化問題を解くことによって決定する。

$$\text{minimize} : \sum_{k=t+1}^{t+h} ((1-w)\|\zeta^{up}(k)\|^2 + w\|\Delta R^{up}(k)\|^2) \quad (9)$$

$$\text{subject to} : \hat{\mathbf{y}}^{up}(k) = G^{up} \cdot R^{up}(k) \cdot \hat{\mathbf{x}}^{up}(k) \quad (10)$$

$$\zeta_l^{up}(k) = \max(\hat{y}_l^{up}(k) - C_l^{up}(k), 0) \quad (11)$$

$$R^{up}(k) \in [0, 1] \quad (12)$$

$$\sum_{p \in \varphi(f)} R_{p,f}^{up}(k) = 1 \quad (13)$$

ここで、 $\hat{\mathbf{y}}^{up}(k)$ は上位層の各リンクを流れるトラヒック量の予測値を示す。 G^{up} は、その (i, j) 成分が、パス j が、リンク i を経由する時に限り 1 を取り、それ以外は 0 をとる。 $R^{up}(k)$ は、その (i, j) 成分が、フロー j を経路候補 i に割り当てる分割割合を示す。 $\zeta_l^{up}(k)$ はリンク l 上の目標帯域の超過量を示している。

3.2.2 下位層での経路制御

上位層の経路制御では、エリア間のトラヒックが、各エリアのどの境界間を経由するかのみを指定し、それが具体的にエリア内のどのリンクを経由するかまでは指定しない。下位層では、そのようなエリアを中継するトラヒックに加えて、下位層内部で閉じたトラヒックについて、その物理的な経路を確定する。

エリア a における、ノード i, j 間に流れるトラヒック $x_{i,j}^{low_a}(t)$ は、エリア内に閉じたトラヒックと、エリア間を中継するトラヒックの和であり、

$$x_{i,j}^{low_a}(k) = \begin{cases} x_{i,j}(k) + y_{l(i,j)}^{up}(k) & (i \in B_a, j \in B_a) \\ x_{i,j}(k) + \sum_{n,p;d(p)=i} R_{p,f(a(k),a)}^{up}(k) x_{f(n,j)}(k) & (i \in B_a, j \notin B_a) \\ x_{i,j}(k) + \sum_{n,p;s(p)=i} R_{p,f(a,a(k))}^{up}(k) x_{f(i,n)}(k) & (i \notin B_a, j \in B_a) \\ x_{i,j}(k) & (i \notin B_a, j \notin B_a) \end{cases}$$

と表される。 B_a はエリア a の境界ノードの集合を表す。 $l(i, j)$ は境界ノード i, j 間の仮想リンクの識別子、 $a(k)$ はノード k の属するエリアの識別子を表す。 $d(p)$ はパス p のうちで最後に到達する境界ノード、 $s(p)$ は最初に通過する境界ノードを表す。また、 $x_{i,j}^{low_a}(k)$ を要素として並べたベクトルを $\mathbf{x}^{low_a}(k)$ とする。下位層では、上位層の決定する経路 $R^{up}(k)$ や、他のエリアから送出されている OD フローのトラヒック $x_{f(n,j)}$ の情報は直接的に収集しないが、各境界ノード間を流れる合計のトラヒック量として、 $\mathbf{x}^{low_a}(k)$ を計測する。そして、時刻 t までの観測値を基に、将来のトラヒック量 $\mathbf{x}^{low_a}(t+1), \dots, \mathbf{x}^{low_a}(t+h)$ を予測し、その値を経路制御に用いる。

各エリア a で、トラヒック $x_{i,j}^{low_a}(t)$ を各経路に割り当てる割合を次の最適化問題を解くことによって決定する。

$$\text{minimize} : \sum_{k=t+1}^{t+h} ((1-w)\|\zeta^{low_a}(k)\|^2 + w\|\Delta R^{low_a}(k)\|^2) \quad (14)$$

$$\text{subject to} : \hat{\mathbf{y}}^{low_a}(k) = G^{low_a} \cdot R^{low_a}(k) \cdot \hat{\mathbf{x}}^{low_a}(k) \quad (15)$$

$$\zeta_l^{low_a}(k) = \max(\hat{y}_l^{low_a}(k) - C_l(k), 0) \quad (16)$$

$$R^{low_a}(k) \in [0, 1] \quad (17)$$

$$\sum_{p \in \varphi(f)} R_{p,f}^{low_a}(k) = 1 \quad (18)$$

ここで、 $\hat{\mathbf{y}}^{low_a}(k)$ はエリア a 内の各リンクを流れるトラヒック

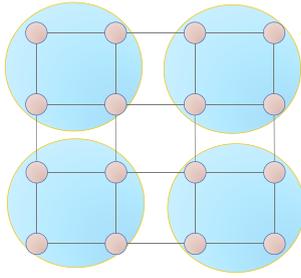


図2 格子状のトポロジー (ノード数:16)
 量の予測値を表す。 G^{low_a} は、その (i, j) 成分が、パス j が、
 リンク i を経由する時に限り 1 を取り、それ以外は 0 をとる。
 $R^{low_a}(k)$ は、その (i, j) 成分が、フロー j を経路候補 i に割り
 当てる分割割合を示す。 $\zeta_l^{low_a}(k)$ はリンク l 上の目標帯域の超
 過量を示している。

4. 評価

4.1 評価方法

4.1.1 評価環境

MPC を適用した階層型 TE の動作をシミュレーションにより評価する。評価には格子状のネットワークトポロジーを用い、ネットワークを十字に区切ることで 4 つのエリアに分割する。図 2 にノード数 16 の場合のトポロジーの例を示す。

評価では、特定のエリア内を流れる一つの OD フロー上のトラヒックが増大した後に、下位層および上位層の経路制御で全体として適切な経路に収束することを確認する。そのため、トラヒックは時間的に変動しない一定値のトラヒックとして与え、ほとんどの OD フローでは均一の 1 単位のトラヒックが流れており、あるエリア内の一つの OD フローではその 10 倍のトラヒックが流すものとする。上位層では、対地間のトラヒックが一定であるため、将来のトラヒックが完全に予測できるものとし、下位層では、上位の経路変更によって、トラヒックパターンが変化するため、予測を行う代わりに、直前の境界ノード間のトラヒックの観測値を代用することとする。予測対象区間の長さは $h = 1$ とした。

輻輳を回避した適切な経路設定が行われていることを評価するために、評価指標としては、ネットワーク全体における最大リンク負荷を用いる。今の評価環境では、トラヒックが固定値であり、最大リンク負荷を最小とするような固定の最適な経路が存在するため、提案手法で求めた経路上での最大リンク負荷が、最適な経路で達成できる最大リンク負荷にどれだけ近い設定が可能であるかを確認することで、提案手法の性能を評価することが可能である。提案手法では、設定された目標帯域に応じて、トラヒックの収容先を決定するため、シミュレーションでは、目標帯域を 0 に設定し、可能な限りリンク負荷を最小化するように経路を設定した上で、最適な経路と最大リンク負荷を比較する。

経路の計算は、4 つの CPU (Xeon E7-4870) を搭載した計算機上で、CPLEX [15] を用いて各階層での経路制御の最適化問題を解くことで求める。

4.2 評価結果

4.2.1 経路の収束性

まず、トラヒックの変化が生じてから、それに対応した経路

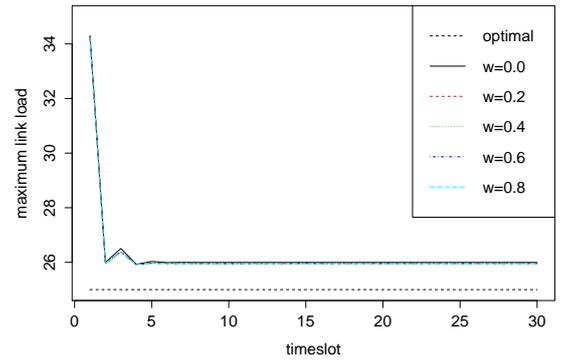


図3 各タイムスロットにおける最大リンク負荷 (ノード数:100) への変更が完了するまでの振る舞いを確認する。図 3 に各タイムスロットで求めた経路にトラヒックを流した場合の最大リンク負荷を示す。トポロジーは、 10×10 の 100 ノードのトポロジーを用いた。破線は最適な経路を設定した場合の最大リンク負荷を示している。

図より、階層型制御は、数ステップで最適な経路に近い経路に収束していることが確認できる。これは、下位層の各エリア内では、内部のトポロジーを正確に把握した上で経路制御を行うため、エリア内で生じるトラヒック変化や、上位での経路変更によるトラヒック変化に対して、素早く対応した経路を設定できるためである。

経路変更に対する重み w を設定することで、各エリアでの経路を急激に変化することを避け、他のエリアの経路制御との協調を促す効果も期待されたが、今回のシミュレーションでは、エリア数が少なく、エリア間でのインタラクションが小さかったために、そのような効果はほとんど確認できなかった。 w の効果としては、文献 [9, 10] で、集中管理型 MP-TE において、 w を適切に設定することで、予測誤差が存在する場合でも適切な経路設定が可能であることが確認されている。そのため、一定値ではなく、変動するトラヒックを実際に予測しながら経路を設定するような場合、 w を適切に設定することで、階層型 MP-TE でも誤差の影響を避けた経路制御が可能となると期待される。

4.2.2 収束した経路の最適性

階層型制御によって、最終的に収束した経路について最適な経路との比較を行う。図 4 にトポロジーのノード数を変化させた場合に、各トポロジーに対し提案手法を適用し、収束した経路での最大リンク負荷を示す。破線は、最適な経路上での最大リンク負荷を示している。図より、提案手法では、ノード数によらず、最適に近い経路に収束していることが確認できる。

4.2.3 計算時間

階層的にネットワークを分割することによる、計算時間削減の効果を確認する。図 5 にノード数を変化させた場合に、各トポロジーの経路計算に要した計算時間を示す。階層型制御では、上位層と下位層の各エリアが並列で経路計算を行っていることを前提として、各タイムスロットで最も計算時間を要したエリアでの計算時間をそのタイムスロットでの計算時間としている。

図より、集中管理型制御では、ノード数の増加に伴って、計算時間が急激に増大していることが確認できる。一方、階層型

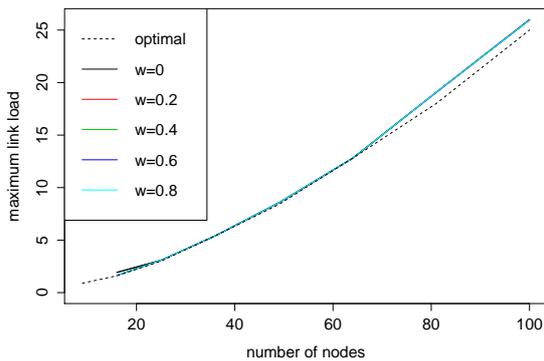


図4 収束した経路での最大リンク負荷とノード数の関係

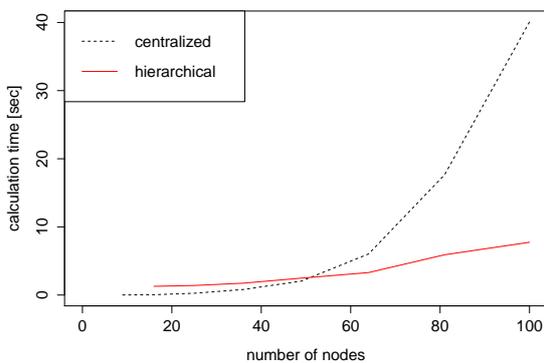


図5 1 スロット当たりの計算時間とノード数の関係
 制御では、ノード数が増加しても、計算時間の増大を抑えることができ、計算時間の削減が行えていることが分かる。これは、エリアを分割・集約することで、一つ当たりの制御サーバーが扱うトポロジーを小さく保っているためである。階層型でも、ノード数の増加に伴って、若干の計算時間の増加がみられるが、これは、今回エリアの分割数を4と固定しているため、ノード数が増加した際に、下位層での一つ当たりのエリア内のノード数が増加しているためである。ノード数の増加に応じて、エリアの分割数を増やし、一つのエリアあたりに属するノード数を減らすことで、さらなる計算時間の削減が可能であると考えられる。ただし、エリア数が増加した場合、エリア間での協調がより難しくなり、経路の収束が遅くなるなどの影響が生じる可能性がある。従って、与えられたネットワークのトポロジーやその規模に対して、適切なエリアの分割数や、具体的な分割の方法について今後検討する必要がある。

5. まとめ

本稿では、MP-TEを階層型経路制御に拡張することにより、大規模なネットワークにも対応可能な予測型経路制御手法を提案した。シミュレーションによる評価では、提案手法によって、ノード数が増加した場合でも、経路計算時間を小さく保ちつつ、適切な経路に十分短い時間で収束可能であることが示された。今後の課題としては、実際にトラヒックの変動の予測と連携する場合に、提案手法に対する予測誤差の影響の評価を行うことと、より現実的なトポロジー上で提案手法を動作させる場合に、適切なエリアの分割数の決定方法と、その具体的な分

割方法の検討を行うことが挙げられる。

謝辞 本研究の一部は戦略的情報通信研究開発推進事業(SCOPE)によっている。

文 献

- [1] C. Graleigh, F. Tobagi, and C. Diot, "Provisioning IP backbone networks to support latency sensitive traffic," in *Proceedings of IEEE INFOCOM 2003*, Apr. 2003, pp. 375–385.
- [2] M. Roughan, "Robust network planning," *Guide to Reliable Internet Services and Applications Computer Communications and Networks*, pp. 137–177, 2010.
- [3] W. Fisher, M. Suchara, and J. Rexford, "Greening backbone networks: reducing energy consumption by shutting off cables in bundled links," in *Proceedings of ACM SIGCOMM Workshop on Green networking*, Aug. 2010, pp. 29–34.
- [4] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: responsive yet stable traffic engineering," in *Proceedings of ACM SIGCOMM 2005*, Aug. 2005, pp. 253–264.
- [5] E. Anwar, C. Jin, L. Steven, and W. Indra, "MATE: MPLS adaptive traffic engineering," in *Proceedings of IEEE INFOCOM 2001*, Apr. 2001, pp. 1300–1309.
- [6] H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg, "COPE: traffic engineering in dynamic networks," in *Proceedings of ACM SIGCOMM 2006*, vol. 36, no. 4, Aug. 2006, pp. 99–110.
- [7] N. Wang, K. H. Ho, G. Pavlou, and M. Howarth, "An overview of routing optimization for Internet traffic engineering," *IEEE Communications Survey & Tutorials*, vol. 10, no. 1, pp. 36–56, first quarter 2008.
- [8] 大歳 達也, 大下 裕一, 村田 正幸, 高橋 洋介, 石橋 圭介, 塩本 公平, 橋本 智昭, "確率的モデル予測制御に基づくトラヒックエンジニアリング," *電子情報通信学会 技術研究報告 (IN2014-81)*, vol. 114, no. 307, pp. 1–6, November 2014.
- [9] 大歳 達也, 大下 裕一, 村田 正幸, 高橋 洋介, 上山 憲昭, 石橋 圭介, 塩本 公平, 橋本 智昭, "トラヒック予測を考慮したトラヒックエンジニアリングの検討と評価," *電子情報通信学会 技術研究報告 (IN2013-78)*, vol. 113, no. 245, pp. 7–12, October 2013.
- [10] —, "モデル予測制御にもとづくトラヒックエンジニアリングの実ネットワークにおけるトラヒックデータを用いた評価," *電子情報通信学会 技術研究報告 (IN2013-194)*, vol. 113, no. 473, pp. 299–304, March 2013.
- [11] Y. Ohsita, T. Miyamura, S. Arakawa, S. Kamamura, D. Shimazaki, K. Shiimoto, A. Hiramatsu, and M. Murata, "Aggregation of traffic information for hierarchical routing reconfiguration," *Computer Networks*, vol. 76, pp. 242–258, Jan. 2015.
- [12] F. Hao and E. W. Zegura, "On scalable QoS routing: performance evaluation of topology aggregation," in *Proceedings of IEEE INFOCOM 2000*, Mar. 2000, pp. 147–156.
- [13] K.-S. Lui, K. Nahrstedt, and S. Chen, "Routing with topology aggregation in delay-bandwidth sensitive networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, pp. 17–29, Feb. 2004.
- [14] S. A. Vavasis, "Complexity theory: quadratic programming," *Encyclopedia of Optimization*, pp. 451–454, 2009.
- [15] "IBM ILOG CPLEX Optimizer," optimization software : <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer>.