

# モデル予測制御にもとづくトラフィックエンジニアリングの 実ネットワークにおけるトラフィックデータを用いた評価

大歳 達也<sup>†</sup> 大下 裕一<sup>†</sup> 村田 正幸<sup>†</sup> 高橋 洋介<sup>††</sup> 上山 憲昭<sup>††</sup>  
石橋 圭介<sup>††</sup> 塩本 公平<sup>††</sup> 橋本 智昭<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> 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

<sup>†††</sup> 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,kamiyama.noriaki,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フィックを収容するには、トラフィックの時間変動やネットワークの状態の変化に対して動的に経路を変更するトラフィックエンジニアリング (TE;Traffic Engineering) と呼ばれる手法が有効である。しかし、各時刻のトラフィック量のみを考慮した TE では、変動への応答の遅れや、大規模な経路変更の頻発を生じ、トラフィックを安定して収容することができない。そのため、動的な TE を行う際にも、将来にわたるトラフィック変動を予測し、それを踏まえた制御が必要となる。しかしながら、予測されたトラフィックには誤差が含まれる。本稿では、モデル予測制御 (MPC;Model Predictive Control) に基づき、将来のトラフィック予測を考慮に入れつつ、予測誤差に対してロバストな TE 手法を検討する。実ネットワークのトラフィックトレースを用いたシミュレーション評価により、MPC を TE に適用することで、予測誤差の影響を吸収して輻輳を回避した適切な経路設定が可能となることを示す。

**キーワード** モデル予測制御, トラフィックエンジニアリング, トラフィック予測, マルチパスルーティング

## Evaluation of Traffic Engineering Based on Model Predictive Control Using Traffic Trace in Actual Network

Tatsuya OTOSHI<sup>†</sup>, Yuichi OHSITA<sup>†</sup>, Masayuki MURATA<sup>†</sup>, Yousuke TAKAHASHI<sup>††</sup>, Noriaki KAMIYAMA<sup>††</sup>, Keisuke ISHIBASHI<sup>††</sup>, Kohei SHIOMOTO<sup>††</sup>, and Tomoaki HASHIMOTO<sup>†††</sup>

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

<sup>††</sup> NTT Network Technology Laboratories, NTT Corporation 3-9-11, Midori-Cho, Musashino-Shi, Tokyo, 180-8585 Japan

<sup>†††</sup> 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,kamiyama.noriaki,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. However, predicted traffic includes prediction errors. We proposed a traffic engineering method with traffic prediction, which is robust to prediction errors. To achieve the robust control against the prediction errors, our method uses the idea of the Model Predictive Control(MPC), which is a method of process control with predicting the system dynamics. In each control cycle, our method calculates the routes with the new predicted traffic which is corrected by the feedback from the observation. In addition, our method avoids the large routing change to achieve the stable routing even if the prediction error may happen. In this paper, through simulation with the actual traffic trace, we clarify that our method avoids the congestion even if the predicted traffic includes errors.

**Key words** Model Predictive Control, Traffic Engineering, Traffic Prediction, Multi-path Routing

## 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。経路制御を分散化することにより、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方法として、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 が考えられる。ネットワーク内のトラフィック予測に関しては、ミリ秒から秒オーダーの短期変動を対象としたもの [8,9] から、日周期変動を対象としたもの [10,11] や数ヶ月から年に渡る長期変動を対象としたもの [12] まで、様々な時間粒度を対象に研究が行われている。また、TE との連携を前提として、日周期変動を予測しつつも短期変動による変動の上端を予測する手法 [13] も検討されている。しかしながら、いずれの予測手法を用いた場合でも、突発的なトラフィック変動も発生するため、予測されたトラフィック変動には誤差が含まれる。特に、予測対象が遠い将来になるほど、その予測誤差は大きくなる。そのため、予測を用いた TE は、予測誤差について考慮したものである必要がある。

本稿では、トラフィック予測と TE を連携させる上で課題となる予測誤差に対してロバストな TE 手法の検討を行う。近年、システム制御の分野において、予測に基づいた制御がモデル予測制御 (MPC; Model Predictive Control) の枠組みで理論的にも研究が進められており [14]、これを TE に応用する。MPC

は、対象システムの将来の出力値を目標値に近づけるように、将来の予測をしながら入力値を定める手法である。各制御時では、一定期間先の将来の入力値を計算するものの、実際にはシステムには直近の入力のみを投入する。そして、その後、出力値をフィードバックとして用い、予測を修正し、再度適切な入力値の計算を行う。これにより、将来の予測を考慮しつつ、予測の誤差にもフィードバックにより対応可能となる。また、各時刻での入力値の変更を抑えることで、一時的な予測誤差に過剰に反応することを防ぐことができる。我々の研究グループでは、このような特徴を持つ MPC を TE 制御に応用することで、予測したトラフィック変動に追従しつつも、予測誤差に対してロバストな TE 制御を提案している [15]。既に単純なネットワーク環境における基本的な動作の確認は行っているものの、実際のネットワーク環境に適用した際の有効性についての議論は未だなされていない。そこで、本稿では、実際のバックボーンネットワークにおけるトラフィックのトレースデータを用いたシミュレーション評価により、提案手法が実際のネットワーク環境においても輻輳を回避した適切な経路設定が可能であることを示す。

以降の本稿の構成は次の通りである。2章では、TE とトラフィック予測のそれぞれについての概要を述べる。3章では、MPC を適用した TE について述べ、4章では、実際のトラフィックトレースを用いたシミュレーション評価について述べる。5章で、まとめと今後の課題について述べる。

## 2. トラフィックエンジニアリングとトラフィック予測

### 2.1 トラフィックエンジニアリング

TE は、ネットワーク内の経路を各タイムスロットのトラフィックに合わせて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技術である。本稿では、TE のうち、各タイムスロットのトラフィックの経路は管理サーバーにおいて集中的に計算されるものとして、集中型の TE について扱う。

集中管理型の TE では、各タイムスロットにおけるトラフィック量を把握し、そのタイムスロットのトラフィック量に適した経路を管理サーバーが計算し、ネットワークに投入する。本稿では、ネットワークの Origin-Destination (OD) 間を結ぶ複数の候補経路が予め与えられた上で、その OD 間のトラフィック量のうち、各候補経路に流す割合を、OD 間のトラフィック量に応じて決めることにより、各タイムスロットに合わせた経路を計算するものとする。

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需要をベクトル  $\mathbf{x}(k) = {}^t(x_1(k), \dots, x_f(k))$  と表す。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。また、 $R_{i,j}(k)$  を要素とする  $p \times f$  行列を  $R(k)$  と表す。 $R(k)$  に従って、OD フローのトラフィック需要  $\mathbf{x}(k)$  を転送する時、全  $l$  本のリンク上を流れるトラフィック  $\mathbf{y}(k) = {}^t(y_1, \dots, y_l)$  は

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

として表される。ここで、 $G$  は各候補経路が経由するリンクを

表す  $l \times p$  行列で、その  $(i, j)$  成分は候補経路  $j$  がリンク  $i$  を經由する時に 1 それ以外は 0 を取る。TE では、 $\mathbf{y}(k)$  を各リンクにおいて収容可能なトラヒック量以下とするように  $R(k)$  を定めることにより、輻輳を回避する。以降、各リンクで適切に収容可能なトラヒック量の上限を  $\mathbf{c} = {}^t(c_1, \dots, c_l)$  とする。

## 2.2 トラヒック予測を用いたトラヒックエンジニアリング

従来の TE では、タイムスロット  $t$  において計測したトラヒック需要  $\mathbf{x}(t)$  を元に、次のタイムスロット  $t+1$  における経路  $R(t+1)$  を設定する。しかしながら、トラヒック変動のため  $\mathbf{x}(t) \neq \mathbf{x}(t+1)$  であり、このように設定された  $R(t+1)$  は、タイムスロット  $t+1$  のトラヒックに合致したものとはならない。特にトラヒック変動が激しい場合は、 $\mathbf{x}(t)$  と  $\mathbf{x}(t+1)$  の乖離が激しく、経路  $R(t+1)$  では輻輳を回避できない。そのため、 $R(t+1)$  の計算には、 $\mathbf{x}(t)$  ではなく、タイムスロット  $t+1$  のトラヒックの予測値  $\hat{\mathbf{x}}(t+1)$  を用いて計算することが必要となる。

さらに、将来のトラヒック変動の予測を用いることにより、頻繁な経路変更を必要としない安定的なトラヒック収容が可能となる。例えば、タイムスロット  $t+1$  からタイムスロット  $t+h$  までのトラヒック量の予測をもとに、いずれのタイムスロットでも輻輳を発生しないような経路を計算することにより、タイムスロット  $t+1$  から  $t+h$  まで経路変更をすることなく、輻輳を回避することができる。しかしながら、トラヒック予測には、予測誤差が含まれる。特に先の将来の予測ほど誤差が増大する。その結果、輻輳が生じてしまう可能性がある。このため、予測を用いた TE では、単に予測結果に合わせて経路を計算するだけでなく、予測誤差に対してロバストとなるような設定が必要となる。

## 3. モデル予測制御のトラヒックエンジニアリングへの適用

### 3.1 モデル予測制御の概要

MPC は、システムの動作（出力）が設定したターゲットに近づくように、システムのパラメータ（入力）を最適化するのは従来のシステム制御と同様であるが、その際に現時刻だけではなく、将来のある一定の区間（予測ホライズン）におけるシステムの振る舞いを考慮した上で、入力の最適化を行う。すなわち、時刻  $k$  の出力を  $\mathbf{y}(k)$ 、ターゲットを  $\mathbf{r}_y(k)$  とすると、現時刻  $t$  において、将来の  $h$  ステップ先の区間  $[t+1, t+h]$  全体に渡ってターゲットからのずれ  $J_1 = \sum_{k=t+1}^{t+h} \|\mathbf{y}(k) - \mathbf{r}_y(k)\|^2$  を最小化する入力列を

$$(\mathbf{u}(t+1), \dots, \mathbf{u}(t+h)) = \arg \min_{\mathbf{u}(t+1), \dots, \mathbf{u}(t+h)} J_1 \quad (2)$$

として求める。ここで、 $\|\cdot\|$  はユークリッドノルムを表す。

上記の最適化を行う際には、時刻  $t$  において、入力  $\mathbf{u}(k)$  が与えられた際の出力  $\mathbf{y}(k)$  を予測することが必要となる。システムに入力を与えた時の出力の時間発展関係はシステムモデルとして表され、次式のようなシステムの各時刻での状態  $\mathbf{z}(k)$  も考慮した状態空間モデルが主に用いられる。

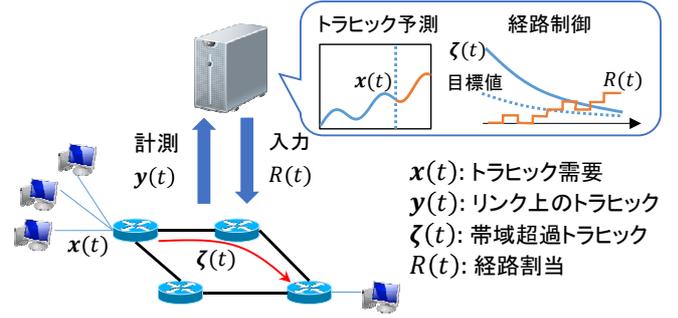


図1 MPCを適用したTEの概要図

$$\mathbf{z}(k+1) = \phi(k, \mathbf{z}(k), \mathbf{u}(k)) \quad (3)$$

$$\mathbf{y}(k) = \psi(k, \mathbf{z}(k), \mathbf{u}(k)) \quad (4)$$

ここで、 $\phi, \psi$  は入力・出力・状態の関係性を表す関数である

しかし、システムのモデル化の際に誤差が含まれることがあり、それに基づいた出力の予測値  $\hat{\mathbf{y}}(k)$  は誤差が生じ得る。特に、先の将来の予測ほど予測誤差が大きくなるため、予測ホライズン  $[t+1, t+h]$  において最適化した入力  $\mathbf{u}(t+1), \dots, \mathbf{u}(t+h)$  の内、実際にシステムに投入するのは直近の  $\mathbf{u}(t+1)$  のみとする。以降の入力設定は、観測フィードバックを元に予測値を修正し、再度入力の最適化を行うことで決定する。

また、予測誤差の影響を受けて過剰に入力変更を行い、システムが不安定になる可能性がある。そこで、各時刻での入力の変更量  $\Delta \mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1)$  を抑えることで、システムを安定化させる。これは、予測ホライズン  $[t+1, t+h]$  全体における入力の変更量  $J_2 = \sum_{k=t+1}^{t+h} \|\Delta \mathbf{u}(k)\|$  を式 (2) の目的関数  $J_1$  と同時に最小化することで可能である。すなわち、 $J_1$  と  $J_2$  のバランスを表すパラメータ  $w$  を導入して、予測ホライズンにおける入力列を式 (2) の代わりに

$$(\mathbf{u}(t+1), \dots, \mathbf{u}(t+h)) = \arg \min_{\mathbf{u}(t+1), \dots, \mathbf{u}(t+h)} J_1 + w J_2 \quad (5)$$

として求める。ここで、 $w$  はその値が大きほど入力の変更を避け、現状の入力を維持した入力設計を行う。

### 3.2 モデル予測制御のトラヒックエンジニアリングへの適用方法

#### 3.2.1 概要

図1に本稿で用いる MPC を適用した TE の概要を示す。TE では、各タイムスロットの経路  $R(k)$  が MPC における入力に該当し、その結果リンク上に流れるトラヒック  $\mathbf{y}(k)$  が出力に該当する。制御の目的は、 $\mathbf{y}(k)$  を目標帯域内に抑え、輻輳を回避することである。すなわち、タイムスロット  $k$  において、各経路において、リンクの目標帯域以上に送出されたトラヒック量  $\zeta(k) = {}^t(\zeta_1(k), \dots, \zeta_p(k))$  を 0 に近づけることが制御目標となる。リンク上に流れるタイムスロット  $k+1$  のトラヒックの予測値  $\hat{\mathbf{y}}(k+1) = {}^t(\hat{y}_1(k+1), \dots, \hat{y}_l(k+1))$  は、式 (1) を基にトラヒック需要の予測値  $\hat{\mathbf{x}}(k)$  を用いて

$$\hat{\mathbf{y}}(k+1) = G \cdot R(k+1) \cdot \hat{\mathbf{x}}(k+1) \quad (6)$$

として表される。文献 [15] では、輻輳時にタイムスロット間をまたぐ遅延が生じるとして  $\hat{\mathbf{y}}(k)$  をモデル化していたが、こ

では、遅延が生じて、タイムスロット内に全てのトラヒックの転送が完了するものとしている。

各経路において、リンクの目標帯域以上に送出されたトラヒック量  $\zeta(k)$  は、各リンクでの帯域超過  $[\mathbf{y}(k) - \mathbf{c}]^+$  によって決定される。ここでは、各リンクでの帯域超過は、そのリンクを経由する各経路に均等に分散されるものとする。そのため、経路  $j$  での帯域超過量  $\zeta'_j(k)$  は、経由するリンクの内、ボトルネックリンクで発生した帯域超過を、当該リンクを経由するトラヒックの本数で割った値となる。すなわち、

$$\zeta'_j(k) = \max_{i \in j} [y_i(k) - c_i]^+ / n_i \quad (7)$$

と表される。ここで、 $n_i$  はリンク  $i$  を経由する候補経路の本数を表し、 $[x]^+$  は  $x \geq 0$  の時  $x$  を取り、それ以外は 0 を取る。また、超過量  $\zeta_j(k)$  を実際の最大リンク帯域で規格化した値を、パスの混雑度  $\zeta'_j(k) = \zeta_j(k) / \max_l c_l$  として定義する。

### 3.2.2 最適化問題としての定式化

経路の設定では、予測ホライズン全体に渡って、帯域超過量  $J_1 = \sum_{k=t+1}^{t+h} \|\zeta(k)\|^2$  を最小化することで、帯域超過を即座に解消する経路を設定する。この時、予測ホライズン全体での経路割当の変更量  $J_2 = \sum_{k=t+1}^{t+h} \|\Delta R(k)\|$  も同時に最小化することで、急激な経路変更を避け、安定した経路変更を実現する。

本稿の TE において、各タイムスロット  $t$  で管理サーバーが解く最適化問題を以下に示す。

$$\text{minimize} : \sum_{k=t+1}^{t+h} ((1-w)\|\zeta(k)\| + w\|\Delta R(k)\|) \quad (8)$$

$$\text{subject to} : \forall k, \forall p, n_l \zeta_p(k) = \max_{l \in p} \frac{[\hat{y}_l(k) - c_l]^+}{\max_l c_l} \quad (9)$$

$$\forall k, \hat{\mathbf{y}}(k) = G \cdot R(k) \cdot \hat{\mathbf{x}}(k) \quad (10)$$

$$\forall k, \forall i, \forall j, R_{i,j}(k) \in [0, 1] \quad (11)$$

$$\forall k, \sum_{i \in \varphi(j)} R_{i,j}(k) = 1 \quad (12)$$

ここで、 $\mathbf{c}, \hat{\mathbf{x}}(k), G, n_l$  は最適化問題に与える変数であり、 $\zeta(k), R(k), \hat{\mathbf{y}}(k)$  は最適化問題を解くことによって得られる変数である。 $0 \leq w \leq 1$  は経路変更に対する重みで、値が大きい方が、現在の経路を維持した経路制御を行う。最適化問題を解くことで、予測ホライズン全体に渡る入力  $R(t+1), \dots, R(t+h)$  が得られるが、実際に投入するのは、直近の設定  $R(t+1)$  のみで、以降の設定は、新たな観測値を基に予測値  $\hat{\mathbf{x}}(k)$  を修正して最適化問題を再計算することにより決定する。

式 (8)–(12) の最適化問題は、非線形な制約条件 (9) を含んでいるが、これを線形な条件に置き換えることで凸二次計画問題に置き換えることができる。すなわち、 $[y_l(k) - c_l]^+$  をスラック変数  $S_l(k) \geq 0$  を導入することで線形制約式  $[y_l(k) - c_l]^+ = y_l(k) - c_l + S_l(k)$  に置き換え、 $\max_{l \in p}$  の演算を行うかわりに、候補経路に含まれる任意のリンク  $l$  についての不等式制約に置き換えることで、次式で表される凸二次計画問題に置き換えることができる。

$$\text{minimize} : \sum_{k=t+1}^{t+h} ((1-w)\|\zeta(k)\| + w\|\Delta R(k)\|) \quad (13)$$

$$\text{subject to} : \forall k, \forall p, \forall l \in p, n_l \zeta_p(k) \geq \alpha_l(k) / \max_l c_l \quad (14)$$

$$\forall k, \forall l, \alpha_l(k) = \hat{y}_l(k) - c_l + S_l(k) \quad (15)$$

$$\forall k, \forall l, \alpha_l(k) \geq 0 \quad (16)$$

$$\forall k, \forall l, S_l(k) \geq 0 \quad (17)$$

$$\forall k, \hat{\mathbf{y}}(k) = G \cdot R(k) \cdot \hat{\mathbf{x}}(k) \quad (18)$$

$$\forall k, \forall i, \forall j, R_{i,j}(k) \in [0, 1] \quad (19)$$

$$\forall k, \sum_{i \in \varphi(j)} R_{i,j}(k) = 1 \quad (20)$$

ここで、 $\alpha_l(k) \geq 0$  は  $[y_l(k) - c_l]^+$  の値を表す変数である。この時、 $n_l \zeta_p(k) \geq \max_{l \in p} \alpha_l(k) / \max_l c_l \geq \max_{l \in p} [y_l(k) - c_l]^+ / \max_l c_l$  を満たし、 $\zeta_p(k)$  の最小化において等式が成り立つので、最適解は確かに元の制約条件 (9) を満たすことが分かる。

## 4. 評価

### 4.1 評価方法

#### 4.1.1 評価環境

MPC を適用した TE の実ネットワークにおける適用性を議論するため、Internet2 のバックボーンネットワークのトポロジー及びトラヒックトレース [16] を用いてシミュレーションを行う。図 2 に、Internet2 のトポロジーを示す。Internet2 には、9 つの PoP (Point-of-Presence) ルーターが存在し、それぞれの PoP 間を流れる計 72 本の OD フローが存在する。Internet2 のトラヒックトレースデータは、各 PoP ルーターが NetFlow を用いてフロー単位のトラヒックを計測したものである。各 PoP ルーターでは、100 パケットに 1 パケットのサンプリングレートで、パケットのサンプリングを行い、5 分毎に集計したフロー情報を出力する。パケットのサンプリングによるフロー計測は、収集される情報に欠落が生じ、トラヒックの統計情報の精度が劣化するという問題が発生する。しかし、TE で対象とするのは、OD フローに集約したフローであり、サンプル数が十分大きいものとなるため、サンプリングの誤差は本評価において大きな問題とはならない。

トラヒックトレースは、2011/1/11 12:00–12:05 における 5 分間に集計されたフロー情報のうち、ファイルの境界である前後 30 秒を避けた 4 分間のデータを用いた。Netflow によって計測されたフローの開始時刻  $t_s$ 、終了時刻  $t_e$ 、および合計のトラヒック量  $\theta$  から、1 秒間隔の各タイムスロットにおけるトラヒックレートを算出する。トラヒックレートが一定であると仮定すると、一定期間  $\tau$  の間に流れるトラヒック量は  $\frac{\theta}{t_e - t_s} \tau$  として算出される。各タイムスロットにおけるトラヒック量は、フローの生存期間によって決まるため、 $\tau$  の値はタイムスロットにおけるフローの生存期間に合わせる。最終的に、BGP の情報を用いて、各フローのトラヒック量を OD フロー単位に集約する。こうして得られた OD フローのトラヒックレートの時系列を図 3 に示す。

シミュレーションにおいて、OD フローは候補経路として、ネットワーク上に存在し得る全ての経路を持っているものとす

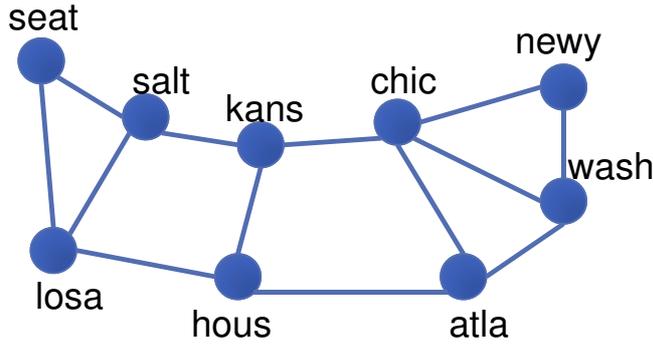


図 2 評価に用いたネットワークトポロジー

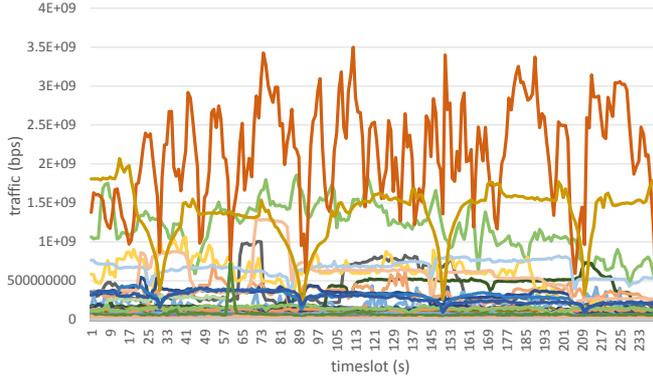


図 3 OD フロー単位で集約した Internet2 のトラフィックトレース (1 秒粒度)

る。従って、一つの候補経路中に含まれるリンクは、他の複数の候補経路によって共有されている可能性があり、これらのリンク上でトラフィックが集中しないように、各 OD フローのトラフィック量に応じて適切に経路を設定する必要がある。Internet2 では、リンク容量がトラフィック量に対して過剰に用意されているため、固定の経路を用いた場合でもリンク利用率が 20% に収まる。そこで、TE の制御目標帯域を実際の帯域の 15% としてシミュレーションを行った。経路計算は、式 (13)–(20) の最適化問題を CPLEX [17] を用いて解くことで求めた。

#### 4.1.2 比較対象

##### a) 観測ベースの TE

将来のトラフィック変動を考慮することによる効果を確認するため、トラフィック予測を用いず各タイムスロットにおいて観測したトラフィックに合わせて逐次経路を最適化する TE と MPC を適用した TE を比較する。この観測ベースの TE では、タイムスロット  $t$  のトラフィック観測値  $\mathbf{x}(t)$  をそのまま用いて、次のタイムスロットにおける経路割当  $R(t+1)$  を決定する。そのため、トラフィック変動への対応が 1 スロット分遅れる。

##### b) 観測トラフィックの平滑化を用いた TE

観測値をそのまま用いる TE だけでなく、平滑化した観測値を用いる TE との比較も行う。この手法では、トラフィックの観測値に平滑化を施し、現時点で平滑化されたトラフィックレート  $\bar{\mathbf{x}}(t)$  を用いて、次のタイムスロットにおける経路  $R(t+1)$  を決定する。トラフィックの平滑化には、指数移動平均 (EMA; Exponential Moving Average) を用いる。つまり、フロー  $i$  の平滑化されたトラフィックレートの直前の値が  $\bar{x}_i(t-1)$  である時、新たにトラフィックレートの観測値  $x_i(t)$  が得られると、 $\bar{x}_i(t) = \eta x_i(t) + (1 - \eta)\bar{x}_i(t-1)$  に従って、その時点の平

滑化されたトラフィックレートが算出される。ただし、 $\eta$  は古いデータに対する重みの減少率を表す。トラフィック予測では、トラフィックの平均的な変化を予測するため、短期的なトラフィック変動が除去され、平滑化と同様の効果が現れると考えられる。平滑化された観測値を用いた TE と MPC を用いた TE を比較することにより、MPC を用いた TE は単純にトラフィックの平滑化によってトラフィック変動に適した経路制御を行っているのではないことを示す。

#### 4.1.3 評価指標

本稿の TE の目的には、各リンクに流れるトラフィック量を目標帯域内に抑えることである。そのため、本評価において、各タイムスロットにおけるネットワーク全体の目標帯域超過量の総和  $\sum_p \zeta_p'(t)$  を評価指標として用いる。

#### 4.1.4 予測方法

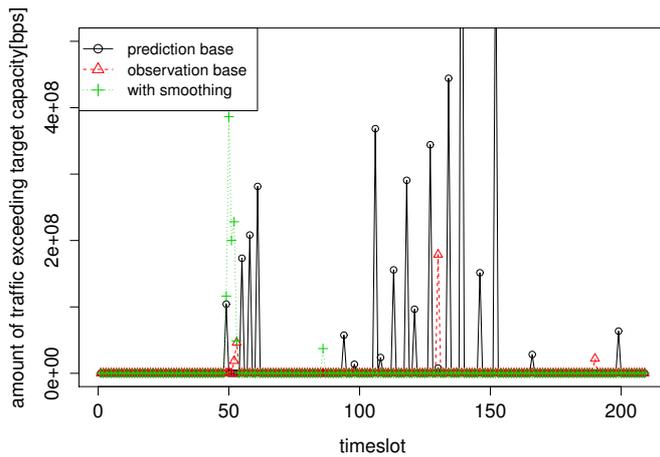
本評価では、以下のような単純な予測手法を用いる。まず、過去の観測時系列  $x_{t-s}, x_{t-s+1}, \dots, x_t (s \geq 1)$  をもとに、 $\sum_{k=0}^s (x_{t-s+k} - l_{t-s+k})^2$  が最小となるような直線  $l_k = ak + b$  を求める。そして、将来の時系列を  $\hat{x}_{t+k} = l_{t+k}$  として予測する。本評価では、MPC と予測の連携の効果のひとつである、フィードバックによる予測の修正の効果を確かめるため、上述の単純な予測方法において、 $s = 1$  としたものをを用いる。

## 4.2 評価結果

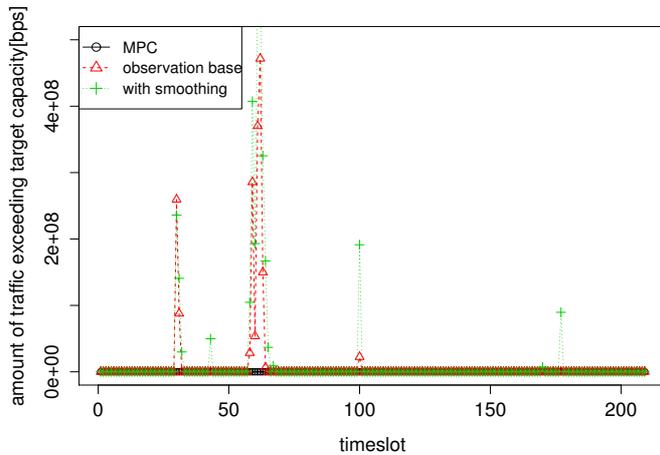
シミュレーションによる評価結果を図 4 に示す。図では、各手法での各タイムスロットにおけるネットワーク全体の目標帯域超過量を示しており、“MPC”はモデル予測を適用した TE の結果を示している。予測ホライズンの長さ  $h$  は 3 とした。経路変更に制限を課さない  $w = 0$  の場合については、MPC を用いた TE は単純に予測値に合わせた経路変更を行う TE と同様の振る舞いをするため、“prediction base”としてラベルを区別している。単純に予測値に合わせて経路変更を行った場合、予測誤差が生じた際に誤った経路変更を行う可能性がある。この単純な方法と比較することで、経路変更を徐々に進めることで予測誤差の影響を回避するという MPC の主な効果を確認する。“observation-base”と “with smoothing”はそれぞれ、観測値のみに基づく TE と平滑化した観測値に基づく TE の結果を示している。

経路変更に制限を加えない場合 (図 4(a)) は、観測ベースの TE や平滑化を用いた TE だけでなく、予測を用いた TE でも輻輳が生じているが、それぞれ原因が異なる。まず、予測を用いた TE では、トラフィックの増減率が変化する度に予測誤差が生じ、その誤った予測に従った経路変更を行った結果、輻輳が生じている。一方、観測ベースの TE では、トラフィック変動に対して、1 スロット遅れた経路変更を行うため、その時のトラフィックに対応できず輻輳を生じている。観測値を平滑化した TE では、観測トラフィックのノイズを抑えることで輻輳を回避する場合もあるものの、トラフィック変動に対する対応の遅れを増長するため、大きな輻輳を生じている。

図 4(b) では、経路変更を制限することで、MPC を用いた TE が予測誤差が生じた場合にも、輻輳を回避した経路を設定できていることが分かる。これは、経路変更を制限することで、



(a) 経路変更を制限しない場合 ( $w = 0$ )



(b) 経路変更を制限する場合 ( $w = 0.5$ )

図4 各タイムスロットにおけるネットワーク全体の目標帯域超過量の総和

MPC を用いた TE が予測誤差の影響を吸収しつつ経路変更を行えるためである。一方で、観測ベースの TE や平滑化を用いた TE では、経路変更に制限を加えない場合よりも、大きな輻輳を生じている。これは、予測を用いずに単純に経路変更を抑えるだけでは、トラフィック変動に対する対応の遅れを増長するためである。

## 5. まとめ

本稿では、実際のネットワーク環境に MPC を用いた TE を適用した際でも、適切な経路制御を行えることを確認するため、実際のトラフィックトレースデータを用いたシミュレーション評価を行った。評価の結果、単純に予測値に合わせた経路制御を行うだけでは、予測誤差の影響による輻輳が生じるような場合でも、MPC の考え方を導入することにより、各ステップでの経路変更量を抑えながら制御することで、輻輳を回避した経路を設定できることが明らかになった。

今後の課題としては、MPC を適用した TE に対して予測誤差の与える影響を制御理論に基づき理論的に解析することと、MPC を適用した TE を分散制御に拡張しスケラビリティを確保することが挙げられる。また、シミュレーションでは考慮し切れない、経路変更が TCP 等の他のネットワーク制御に与える影響も含めた評価を実機を用いて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. New York, NY, USA: ACM, 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] B. Zhou, D. He, Z. Sun, and W. H. Ng, "Network traffic modeling and prediction with ARIMA/GARCH," in *Proceedings of the Fourth International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks*, Sep. 2006, pp. 1–10.
- [9] M. F. Zhani, H. Elbiaze, and F. Kamoun, "Analysis and prediction of real network traffic," *Journal of Networks*, vol. 4, no. 9, pp. 855–865, Nov. 2009.
- [10] C. Guang, G. Jian, and D. Wei, "Nonlinear-periodical network traffic behavioral forecast based on seasonal neural network model," in *Proceedings of the Second International Conference on Communications, Circuits and Systems*, vol. 1, Jun. 2004, pp. 683–687.
- [11] M. L. F. Miguel, M. C. Penna, J. C. Nievola, and M. E. Pellenz, "New models for long-term Internet traffic forecasting using artificial neural networks and flow based information," in *Proceedings of IEEE Network Operations and Management Symposium 2012*, Apr. 2012, pp. 1082–1088.
- [12] K. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot, "Long-term forecasting of Internet backbone traffic: Observations and initial models," in *Proceedings of IEEE INFOCOM 2003*, vol. 2, Mar. 2003, pp. 1178–1188.
- [13] T. Otsoshi, Y. Ohsita, M. Murata, Y. Takahashi, K. Ishibashi, and K. Shiimoto, "Traffic prediction for dynamic traffic engineering considering traffic variation," in *Proceedings of IEEE GLOBECOM 2013*, Dec. 2013, pp. 1592–1598.
- [14] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, Jul. 2003.
- [15] 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, "トラフィック予測を考慮したトラフィ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.
- [16] "Internet2 data," available from <http://internet2.edu/observatory/archive/data-collections.html>.
- [17] "IBM ILOG CPLEX Optimizer," optimization software : <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer>.