

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

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^{†††}

[†] 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,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ヒックを収容するには、トラヒ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ヒックエンジニアリングと呼ばれる手法が有効である。しかし、各時刻のトラヒック量のみを考慮したトラヒックエンジニアリングでは、変動への応答の遅れや、大規模な経路変更の頻発を生じ、トラヒ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、将来にわたるトラヒック変動を予測し、それを踏まえた制御が必要となる。しかしながら、予測されたトラヒックには誤差が含まれる。本稿では、モデル予測制御に基づき、将来のトラヒ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トラヒックエンジニアリング手法を提案する。シミュレーション評価により、提案手法を用いることにより、予測誤差が存在する場合でも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, トラヒックエンジニアリング, トラヒック予測, マルチパスルーティング

Evaluation of Traffic Engineering Considering Traffic Prediction

Tatsuya OTOSHI[†], Yuichi OHSITA[†], Masayuki MURATA[†], Yousuke TAKAHASHI^{††}, Noriaki KAMIYAMA^{††}, 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
9-11, Midori-Cho 3-Chome 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,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. In this paper, we propose 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. Through simulation, 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ヒ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、大きなトラヒ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ヒックを収容する必要がある。

このような時間変動の大きなトラヒックを輻輳を発生させずに収容する手法として、トラヒックの変動に合わせて経路を動的に変更するトラヒ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 [1-4]。これらの研究では、トラヒック変動は定期的なトラヒック観測により取得し、各時刻で観測されたトラヒ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、これらの手法も、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ヒ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 [1, 2] も検討されているものの、経路制御後に発生したトラヒック変動への対応は次の経路制御時となり、そのようなトラヒック変動による輻輳を回避することができない。

このような問題を解消する方法として、トラヒック予測と連携し、予測された将来のトラヒ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ヒ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 [5, 6] から、日周期変動を対象としたもの [7, 8] や数ヶ月から年に渡る長期変動を対象としたもの [9] 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 [10, 11] 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、突発的なトラヒック変動も発生するため、予測されたトラヒ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。

そこで、本稿では、予測誤差も考慮に入れ、トラヒック予測と連携して動作する TE について検討を行う。本稿では、予測に基づいた制御として、システム制御の分野で検討が進められているモデル予測制御 (MPC; Model Predictive Control) [12] 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、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。本稿では、MPC を TE 制御に応用することで、予測したトラヒ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

手法の検討を行う。

以降の本稿の構成は次の通りである。2章では、TE とトラヒック予測のそれぞれについての概要を述べる。3章では、提案手法である MPC を適用した TE について述べ、4章では、その評価を述べる。5章で、まとめと今後の課題について述べる。

2. トラヒックエンジニアリングとトラヒック予測

2.1 トラヒ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ヒックに合わせて動的に変更することにより、輻輳なくトラヒ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ヒ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ヒック量を把握し、そのタイムスロットのトラヒ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 間のトラヒック量のうち、各候補経路に流す割合を、OD 間のトラヒ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 フローのトラヒ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 のトラヒックのうち、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 フローのトラヒック需要 $\mathbf{x}(k)$ を転送する時、全 l 本のリンク上を流れるトラヒック $\mathbf{y}(k) = {}^t(y_1, \dots, y_l)$ 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 を経由する時に 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

$t+h$ までのトラフィ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、トラフィック予測には、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フィ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 $y(k)$ 、ターゲットを $r_y(k)$ 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 $J_1 = \sum_{k=t+1}^{t+h} \|y(k) - r_y(k)\|^2$ を最小化する入力列を

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

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 $u(k)$ が与えられた際の出力 $y(k)$ 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 $z(k)$ も考慮した状態空間モデルが主に用いられる。

$$z(k+1) = f(k, z(k), u(k)) \quad (3)$$

$$y(k) = h(k, z(k), u(k)) \quad (4)$$

ここで、 f, h 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 $\hat{y}(k)$ 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 $u(t+1), \dots, u(t+h)$ の内、実際にシステムに投入するのは直近の $u(t+1)$ 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 $\Delta u(k) = u(k) - u(k-1)$ 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 $[t+1, t+h]$ 全体においての入力の変更量 $J_2 = \sum_{k=t+1}^{t+h} \|\Delta u(k)\|$ 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

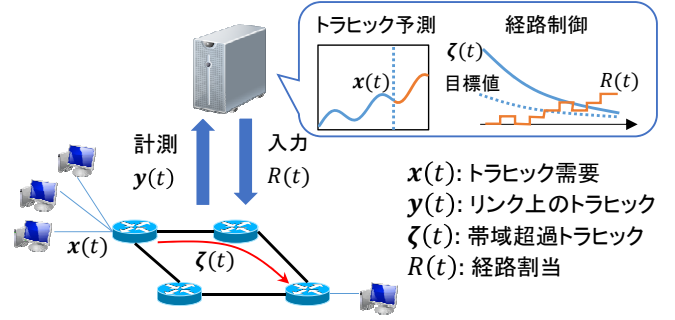


図1 MPCを適用したTEの概要図

$$(u(t+1), \dots, u(t+h)) = \arg \min_{u(t+1), \dots, u(t+h)} J_1 + wJ_2 \quad (5)$$

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フィ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フィック $y(k)$ が出力に該当する。制御の目的は、 $y(k)$ 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フィック量 $\zeta(k) = {}^t(\zeta_1(k), \dots, \zeta_p(k))$ を 0 に近づけることが制御目標となる。リンクの目標帯域を超えて送出されたトラフィックは、バッファ等に蓄えられ、当該タイムスロット内に宛先まで転送できない可能性がある。そのため、本稿では、リンクの目標帯域を超えて送出されたトラフィックは、次のタイムスロットにおいても当該経路に残っているものとして、ネットワークをモデル化する。つまり、リンク上に流れるタイムスロット $k+1$ のトラフィックの予測値 $\hat{y}(k+1) = {}^t(\hat{y}_1(k+1), \dots, \hat{y}_p(k+1))$ は、式 (1) を基にトラフィック需要の予測値 $\hat{x}(k)$ を用いて

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

として表される。

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フィック量 $\zeta(k)$ は、各リンクでの帯域超過 $[y(k) - c]^+$ 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フィ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。

3.2.2 最適化問題としての定式化

本稿では、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。

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

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

$$\forall k, \hat{\mathbf{y}}(k) = G \cdot (R(k) \cdot \hat{\mathbf{x}}(k) + \zeta(k-1)) \quad (10)$$

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

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

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。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。

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

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

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

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

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

$$\forall k, \hat{\mathbf{y}}(k) = G \cdot (R(k) \cdot \hat{\mathbf{x}}(k) + \zeta(k-1)) \quad (18)$$

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

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

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、 $\zeta_p(k) \geq \max_{l \in p} \alpha_l(k) / n_l \geq \max_{l \in p} [y_l(k) - c_l]^+ / n_l$ 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 を図 2 に示すネットワークにおいてシミュレーションを行い評価する。ノード 0,1 間と、ノード 4,5

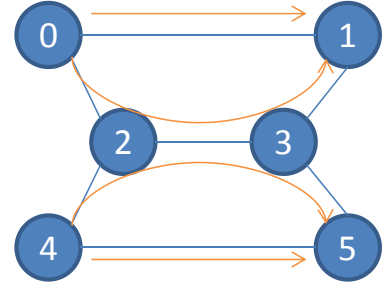


図 2 評価に用いたネットワークポロジ

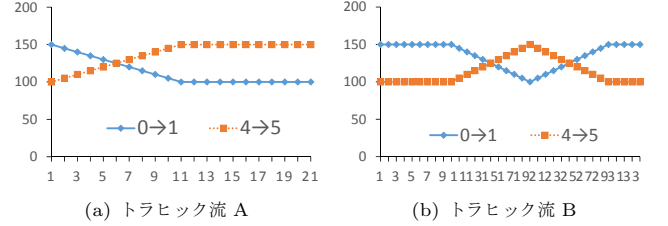


図 3 評価に用いたトラフィック流

間の二つの OD フローが存在し、それぞれが図の矢印で示されるような二つの候補経路を持っている状況を想定する。OD フローのトラフィック需要は、図 3 に示すように、ある区間で需要の増減が生じるようなトラフィックを与えた。各タイムスロットにおける経路設定は最適化問題 (13)–(20) を CPLEX [13] を用いて解くことにより行う。

4.1.2 比較対象

a) 観測ベースのトラフィックエンジニアリング

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) Zero-Buffer-Path-Flow モデル

文献 [14] でも、MPC を適用したトラフィックエンジニアリングが提案されている。この手法では、将来のトラフィックの予測は行わず、新たに到着するトラフィックをゼロ ($\hat{\mathbf{x}}(k) = 0$) と仮定して、観測時に到着したトラフィックのみを考慮する Zero-Buffer-Path-Flow (ZBPF) モデルを用いて TE を行う。ZBPF モデルでは、タイムスロット t のトラフィック観測値 $\mathbf{x}(t)$ のみを用いて、未送信のトラフィック量 $\mathbf{z}(k) = \mathbf{x}(t) - \sum_{i=t}^{k-1} \mathbf{u}(i)$ を予め定められた時間内に解消するように、各タイムスロットにおける各経路候補への転送レート $\mathbf{u}(k)$ を決定する。文献 [14] では、直接各経路候補へ転送されるトラフィック量を定めることで制御を行っているが、3 章の MPC を適用した TE との比較の条件を揃えるため、本稿の評価では、各経路候補に転送されるトラフィックの割合 $R(k)$ を設定した上で、転送しきれなかったトラフィック量 $\zeta(k)$ を解消する制御として ZBPF を動作させ、比較を行う。

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。この予測手法では、 $s = 1$ とすると、新たな観測値とその直前の値との差に応じて傾きが即座に修正される。一方、 s を大きく取ることで、過去の観測値を重視し、新たな観測値によるばたつきを抑えることも可能である。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、予測の修正が早い $s = 1$ の場合を用いる。

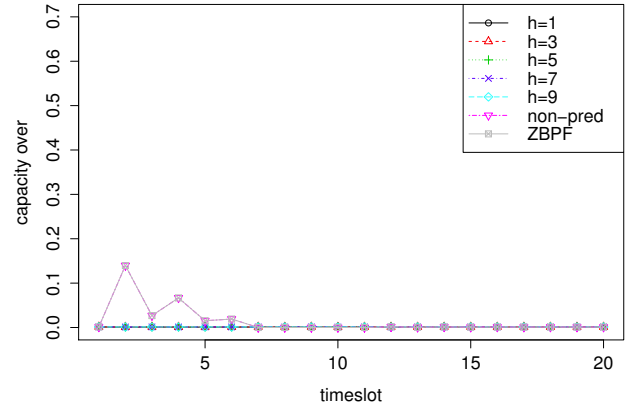
4.2 評価結果

4.2.1 将来のトラヒック変動が既知の場合

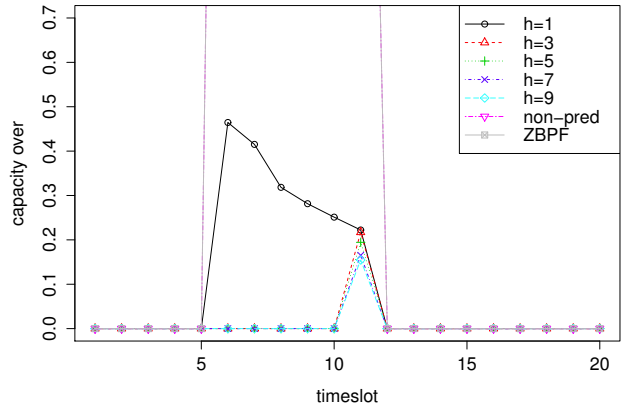
将来の変動を考慮した経路変更を行うことにより、トラヒック変動に追従した経路変更が行えることを確認するため、予測誤差のない理想的なトラヒック予測ができ、将来のトラヒック需要の変動が既知であるとした場合の動作を評価する。トラヒック流は図 3(a) のものを用いた。

図 4 に各制御についてシミュレーションを行った際に、各タイムスロットで生じた帯域超過量の総和を示す。提案手法では、予測ホライズンの長さ h は管理者の設定するパラメータとなるため、 h の値を変更してシミュレーションを行っている。図中で、“ $h=i$ ”は提案手法において、予測ホライズンの長さを i に設定した上で行ったシミュレーションの結果である。また、“non-pred”は観測ベースの TE の結果で、“ZBPF”は ZBPF モデルを用いた場合の結果を表す。ZBPF モデルでも、予測ホライズンの長さがパラメータとなるが、最も帯域超過を抑えられていた $h = 1$ の結果のみを示している。各 TE 手法において、経路変更量の重み w を変更し、各タイムスロットにおける経路変更量を抑える効果を確認した。経路変更の制限をおかない場合を図 4(a) に示し、 w により経路変更が抑えられている場合を図 4(b) に示す。

図 4(a) より、提案手法では、帯域超過は発生していない一方で、トラヒック予測を用いていない“non-pred”及び“ZBPF”では、帯域超過が生じていることが分かる。これは、提案手法がトラヒック予測を元に、変動に合わせて正しい経路を設定できているのに対して、“non-pred”及び“ZBPF”では観測ベースであるために、トラヒック変動への対応が遅れるためである。また、図 4(b) では、“non-pred”及び“ZBPF”において非常に大きな帯域超過が生じている。これは、経路変更を抑えることで、現状の経路設定を維持しようとし、トラヒック変動への対応が遅れが生じるためである。提案手法においても、この影響が見られ、特に $h = 1$ の場合において大きく帯域超過が生じている。 $h = 1$ で顕著に経路変更を制限する影響が現れている理由は、1 ステップ先までの予測ではトラヒックの増減量は顕著でなく、目的関数に占める経路変更の抑制の項の影響が著しく大きくなるためだと考えられる。



(a) 経路変更制限がない場合 ($w = 0$)



(b) 経路変更を抑えた場合 ($w = 10^8$)

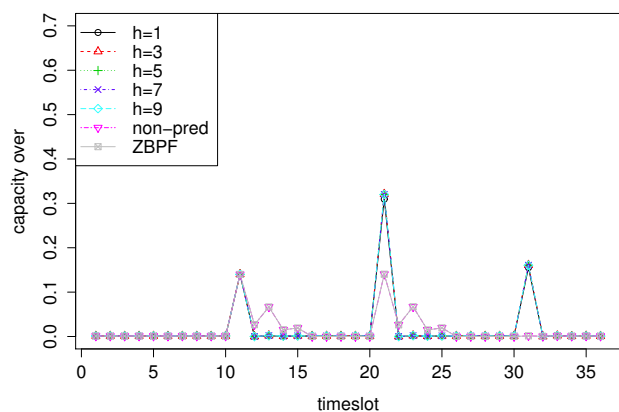
図 4 各タイムスロットで生じた帯域超過量の総和 (トラヒック変動が既知)

4.2.2 予測を逐次修正して与えた場合

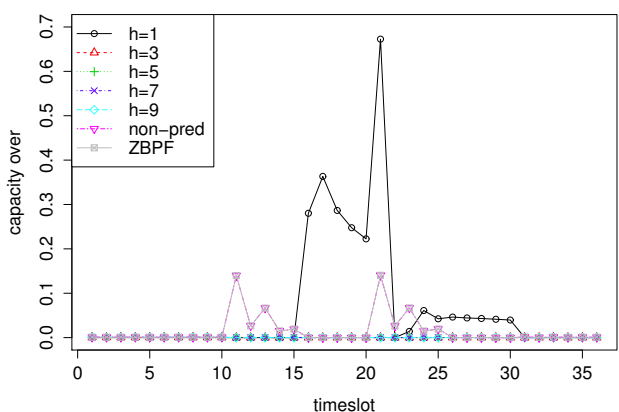
次に、予測が正確でない場合でも、安定的に経路変更を行うことで、予測誤差の影響を緩和できることを確認するため、実際にトラヒック予測を行い、最新の観測値を元に予測を逐次修正して、提案手法による制御を行う場合の動作を評価する。トラヒック流は図 3(b) のものを用いた。

図 5 に各タイムスロットで生じた帯域超過量の総和を示す。提案手法において、経路変更の重み w を変更して、各タイムスロットにおける経路変更量を抑える効果を確認した。経路変更の制限をおかない場合を図 5(a) に示し、 w により経路変更が抑えられている場合を図 5(b) に示す。予測を用いない“non-pred”及び、“ZBPF”では経路変更を制限すると、単純にトラヒック変動への対応が遅れてしまうため、 $w = 0$ に固定している。

図 5(a) では、トラヒックの増減が切り替わるタイムスロット 11, 21, 31 において予測誤差が生じるため、提案手法はその影響を受けて誤った経路変更を行い、観測ベースよりも大きな帯域超過を生じている。一方で、図 5(b) では、各ステップにおける経路変更を抑えることにより、予測誤差の影響を緩和することができ、提案手法は適切な経路を設定することができ、帯域超過を回避できている。ただし、 w を過剰に大きくすると、現状の経路設定を維持しようとし、トラヒック変動への対応が遅れが生じる。このため、経路変更を制限する影響の大きい $h = 1$ の場合において、大きな帯域超過が発生している。従って、予測ホライズンの長さ h はある程度大きく設定した上で、経路変更の重み w を設定することが望ましい。



(a) 経路変更制限がない場合 ($w = 0$)



(b) 経路変更を抑えた場合 ($w = 10^8$)

図5 各タイムスロットで生じた帯域超過量の総和 (予測を逐次修正)

5. まとめ

本稿では、トラフィック予測とTEを連携させるために、トラフィック予測を考慮したTE手法を提案・評価した。提案手法は、正確な予測が可能であるような理想的な場合には、予測されるトラフィック変動に対して前もって経路変更を行うことにより、観測値のみを考慮したTEと比べて、トラフィック変動時の輻輳を回避できることを示した。また、予測誤差が生じる場合でも、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を示した。

今回のシミュレーション評価では、単純なネットワーク環境を想定しており、経路設定におけるODフロー間の競合も単純なものとなっていた。今後の課題としては、より現実的なネットワーク環境でのシミュレーションで提案手法の有効性を確認することと、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。

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

文献

- [1] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: responsive yet stable traffic engineering," in *Proceedings of SIGCOMM*, Aug. 2005, pp. 253–264.
- [2] E. Anwar, C. Jin, L. Steven, and W. Indra, "MATE: MPLS adaptive traffic engineering," in *Proceedings of INFOCOM*, Apr. 2001, pp. 1300–1309.
- [3] H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg, "COPE: traffic engineering in dynamic net-

- works," in *Proceedings of SIGCOMM*, vol. 36, no. 4. New York, NY, USA: ACM, Aug. 2006, pp. 99–110.
- [4] 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.
- [5] B. Zhou, D. He, Z. Sun, and W. H. Ng, "Network traffic modeling and prediction with ARIMA/GARCH," in *Proceedings of HET-NETs*, Sep. 2006, pp. 1–10.
- [6] 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.
- [7] C. Guang, G. Jian, and D. Wei, "Nonlinear-periodical network traffic behavioral forecast based on seasonal neural network model," in *Proceedings of ICCAS*, vol. 1, Jun. 2004, pp. 683–687.
- [8] M. L. F. Miguel, M. C. Penna, J. C. Nievola, and M. E. Peltenz, "New models for long-term Internet traffic forecasting using artificial neural networks and flow based information," in *Proceedings of NOMS*. IEEE, Apr. 2012, pp. 1082–1088.
- [9] K. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot, "Long-term forecasting of Internet backbone traffic: Observations and initial models," in *Proceedings of INFOCOM*, vol. 2, Mar. 2003, pp. 1178–1188.
- [10] 大歳 達也, 大下 裕一, 村田 正幸, 高橋 洋介, 石橋 圭介, 塩本 公平, "トラフィックの時間変動を考慮した動的トラフィックエンジニアリングのためのトラフィック予測," *電子情報通信学会 技術研究報告 (NS2012-115)*, vol. 112, no. 287, pp. 65–70, Nov. 2012.
- [11] 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 GLOBECOM*, Dec. 2013, (In press).
- [12] 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.
- [13] "IBM ILOG CPLEX Optimizer," optimization software : <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer>.
- [14] G. Retvari and G. Nemeth, "On optimal multipath rate-adaptive routing," in *Proceedings of ISCC*, Jun. 2010, pp. 605–610.