

モデル予測制御にもとづく仮想ネットワーク間資源調停

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

上山 憲昭^{††} 塩本 公平^{††}

[†] 大阪大学 大学院情報科学研究科

^{††} 日本電信電話株式会社 NTT ネットワーク基盤技術研究所

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

^{††}{takahashi.yousuke,ishibashi.keisuke,kamiyama.noriaki,shiomoto.kohei}@lab.ntt.co.jp

あらまし 仮想ネットワーク環境下においては、各仮想ネットワークに必要な資源を割り当てることが重要となる。ユーザが増加したサービスがより多くの帯域を必要とするなど、各仮想ネットワークが必要とする資源は時間により変化する。そのため、仮想ネットワークが必要とする資源量の変動に合わせて、各仮想ネットワークに割り当てる資源の再設定を定期的に行うことが必要となる。しかしながら、この資源割り当ての再設定周期が長いと、資源不足が発生後、資源不足の解消までに時間がかかってしまうという問題がある。その一方、頻繁に資源の再割り当てを行うと、サービスの不安定化を招く恐れがある。この問題を解決する方法として、必要資源量の予測と連携した資源割り当てが考えられる。この方法では、必要資源量を予測し、それに合わせた資源割り当てを事前に行うことで資源不足の発生を防ぐ。しかしながら、予測された必要資源量には予測誤差が含まれる。そのため、予測された必要資源量をもとにした資源割り当て制御では予測誤差を考慮することが必須となる。本稿では、システム制御の分野において、予測に基づいた制御として検討が進められているモデル予測制御を仮想ネットワーク間の資源調停に適用し、予測を用いながら、全仮想ネットワークに十分な資源の割り当てを試みる手法の検討を行う。

キーワード 仮想ネットワーク、ネットワークスライス、モデル予測制御

Resource Optimization of Virtual Networks Based on Model Predictive Control

Yuichi OHSITA[†], Tatsuya OTOSHI[†], Masayuki MURATA[†], Yousuke TAKAHASHI^{††},

Keisuke ISHIBASHI^{††}, Noriaki KAMIYAMA^{††}, and Kohei SHIOMOTO^{††}

[†] Graduate School of Information Science and Technology, Osaka University

^{††} NTT Network Technology Laboratories, NTT Corporation

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

^{††}{takahashi.yousuke,ishibashi.keisuke,kamiyama.noriaki,shiomoto.kohei}@lab.ntt.co.jp

Abstract In this paper, we consider the resource optimization of virtual networks which have the subset of the resources of the physical network. The physical network manager should provide his/her resources to virtual networks so that all virtual networks satisfies their requirements. However, the demands in virtual networks may change in time, and the virtual network whose demands increase requires more resources. Thus, re-optimization of resources is required. Though frequent resource re-optimization makes the network unstable, resource re-optimization should be completed immediately after the lack of resources occurs. To solve this problem, we consider the resource optimization method based on the predicted demands. Our method uses the idea of the Model Predictive Control(MPC), which is a method of process control with predicting the system dynamics. Through numerical evaluation, we demonstrate that our method avoids lack of resources.

Key words Virtual Network, Network Slice, Model Predictive Control

1. はじめに

多種多様なネットワークサービスを一つのネットワーク上に収容する方法として、物理的なネットワークをネットワークスライスとして切り出すネットワーク仮想化技術に関する検討が進められている [1]。ネットワーク仮想化技術を用いることにより、一つの物理ネットワークの資源の一部を切り出した仮想ネットワークを各サービスに対して構築し、各仮想ネットワークを用いてネットワークサービスを提供することが可能となる。

仮想ネットワーク環境下においては、各仮想ネットワークに必要な資源を割り当てることが重要となる。ユーザが増加したサービスがより多くの帯域を必要とするなど、各仮想ネットワークが必要とする資源は時間により変化する。そのため、仮想ネットワークが必要とする資源量の変動に合わせて、各仮想ネットワークに割り当てる資源の再設定を定期的に行うことが必要となる。しかしながら、この資源割り当ての再設定周期が長いと、資源不足が発生後、資源不足の解消までに時間がかかってしまうという問題がある。その一方、頻繁に資源の再割り当てを行うと、サービスの不安定化を招く恐れがある。

この問題を解決する方法として、必要資源量の予測と連携した資源割り当てが考えられる。この方法では、必要資源量を予測し、それに合わせた資源割り当てを事前に行うことで資源不足の発生を防ぐ。しかしながら、予測された必要資源量には予測誤差が含まれる。そのため、予測された必要資源量をもとにした資源割り当て制御では、予測誤差を考慮することが必須となる。

予測に基づいた制御として、システム制御の分野において、モデル予測制御 (MPC; Model Predictive Control) [2] の検討が進められている。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。我々は、これまで、MPC をネットワーク内の経路の動的制御に適用し、その有効性の評価を行ってきた [3,4]。

本稿では、MPC を仮想ネットワーク間の資源割り当てに適用し、その有効性について検討を行う。

以降、本稿では、2. で、本稿で扱う仮想ネットワーク間の資源調停問題について説明する。その後、3. で、MPC と我々がこれまでに検討を行ってきた MPC にもとづく経路制御手法について述べる。そして、4. で、MPC にもとづく仮想ネットワーク間の資源調停手法について述べる。5. で評価を行い、6. でまとめと今後の課題について述べる。

2. 仮想ネットワーク間資源調停

本稿では、物理ネットワーク管理者が物理ネットワークの管

理を行い、ネットワークサービス提供者に対して物理ネットワークの資源を切り出した仮想ネットワークを提供する。ネットワークサービス提供者は提供された仮想ネットワークを用い、自身のサービスを提供する。本稿では、一つの物理ネットワークのみが存在し、当該物理ネットワークの資源を複数のサービス提供者に分配する状況を考える。

以降、サービス提供者の集合を S とし、サービス s に対応する仮想ネットワークを V_s とする。物理ネットワーク上のリンクの集合を L とし、 $l \in L$ に対して、 l の総帯域を B_l^{phy} とし、そのうち、仮想ネットワークを V_s に割り当てられた帯域を $B_{V_s,l}$ とする。

2.1 仮想ネットワークの制御

サービス s では、サービス s に関する全ノード間のトラヒック量を把握し、それを元に、仮想ネットワーク V_s 内の経路を制御する。サービス s に関する時刻 t におけるノード間トラヒック量を $X^s(t) = (x_1^s(t), x_2^s(t), \dots, x_f^s(t))^T$ とする。ただし、 f はノード間トラヒックの本数、 $x_i^s(t)$ は時刻 t におけるサービス s に関する i 本目のノード間トラヒックのトラヒック量である。また、サービス s 上の経路制御により定められた経路を $R^s(t)$ とあらわす。ただし、 $R^s(t)$ は、 i 本目のノード間トラヒックを j 本目の経路に転送する割合を示す $r_{i,j}^s(t)$ を要素とする行列である。全リンク上の時刻 t におけるサービス s 上のトラヒック量を表すベクトル $Y^s(t) = (y_1^s(t), y_2^s(t), \dots, y_{|L|}^s(t))^T$ は、以下のように表すことができる。

$$Y^s(t) = R^s(t)GX^s(t)$$

ただし、 G は各候補経路が経由するリンクを表す行列で、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。

サービス s の提供者は、自身のサービスに関するトラヒック量 $T^s(t)$ を定期的に観測し、輻輳なく全トラヒックを収容できるように経路 $R^s(t)$ を定める。つまり、

$$\forall l \in L: y_l^s(t) \leq B_{V_s,l}$$

を満たすように、 $R^s(t)$ を決め、動的に経路の変更を行う。

ただし、 $R^s(t)$ は時刻 t よりも前に設定する必要があるため、 $R^s(t)$ を定める際には $T^s(t)$ を把握することはできない。そこで、 $T^s(t-1)$ までに観測されたトラヒック量から $T^s(t)$ を予測し、予測した値をもとに経路を決めることが考えられる。このようなトラヒック量の予測と連携した動的な経路制御手法として、我々は、モデル予測制御にもとづく動的経路制御手法を提案 [3,4] しており、本稿の 6. においても、モデル予測制御にもとづく動的経路制御手法を用いる。モデル予測制御にもとづく動的経路制御手法については、3. で述べる。

2.2 仮想ネットワーク間資源調停

物理ネットワークの管理者は、各仮想ネットワークが必要としている資源量に合わせて、各仮想ネットワークに配分する資源を定期的に調整を行う。物理ネットワークの管理者は、各仮想ネットワーク上で流れるトラヒック量の詳細は把握できないものの、各仮想ネットワークのリンク上に流れるトラヒック量

$Y^s(t)$ は観測できるものとする。

そして、以下の式を満たすように、時刻 t における各仮想ネットワークに割り当てられるリンクの帯域 $B_{V_s,l}(t)$ を決める。

$$\forall l \in L: y_l^s(t) \leq B_{V_s,l}(t)$$

$$\text{s.t. } \sum_{s \in S} B_{V_s,l}(t) \leq B_l$$

ただし、 $y_l^s(t)$ は、 $B_{V_s,l}(t)$ を決める際には、未知であるため、予測した値を用いる必要がある。

3. モデル予測制御とモデル予測制御にもとづく動的経路制御

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

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

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。

$$z(k+1) = \phi(k, z(k), u(k)) \quad (2)$$

$$y(k) = \psi(k, z(k), u(k)) \quad (3)$$

ここで、 ϕ, ψ 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 (1) 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 (1) の代わりに

$$(u(t+1), \dots, u(t+h)) = \arg \min_{u(t+1), \dots, u(t+h)} J_1 + wJ_2 \quad (4)$$

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 モデル予測制御にもとづく経路制御

まず、これまでに我々が検討を行ってきたモデル予測制御にもとづく経路制御手法を仮想ネットワーク内の経路制御に用いた場合の動作について説明する。

動的な経路制御では、各タイムスロットの経路 $R^s(k)$ 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 $Y^s(k)$ が出力に該当する。ここでは、制御の目的は、(1) $Y^s(k)$ を目標帯域内に抑え、輻輳を回避すること、(2) 輻輳が回避できる環境下においては、ホップ数が短い経路にトラフィックを収容することで、伝搬遅延を小さく抑えることとする。

経路制御の際には、 H タイムスロット先までの予測されたトラフィック量 $\hat{X}^s(t)$ 、経路候補の集合と、各候補経路が経由するリンクを表す行列 G 、各経路候補のホップ数を表すベクトル $D = (d_1, d_2, \dots)^T$ を入力として得る。ただし、 d_k は k 本目の経路候補のホップ数であるとする。そして、これらの入力値をもとに、経路 $R^s(t)$ を定める。

時刻 $k+1$ における各リンク上を流れるトラフィック量は

$$\hat{Y}^s(k+1) = G \cdot R^s(k+1) \cdot \hat{X}^s(k+1) \quad (5)$$

として表される。そのため、輻輳を解消する目的に対して、我々は、全リンクにおいて与えられた帯域を超えて送信されたトラフィック量を示す変数 $\zeta^s(k)$ を導入する。 $\zeta^s(k)$ は以下の式で定義される。

$$\zeta^s(k) = \sum_{l \in L} \frac{[y_l^s(k) - B_{V_s,l}]^+}{|L|B_l} \quad (6)$$

ただし、 $[x]^+$ は $x \geq 0$ の時 x を取り、それ以外は 0 を取る。また、式 (15) では、 $|L|B_l$ で割ることにより、 $\zeta^s(k)$ の値が帯域やリンク数に依存することを避けている。

また、時刻 t における i 本目のノード間トラフィックの平均ホップ数 $h_i(t)$ は、以下の式であらわすことができる。

$$h_i(t) = \sum_k d_k r_{i,k}^s(t) \quad (7)$$

本稿では、この 2 点について重み付きの和を最小化することを目標とする。つまり、制御の目的関数は、 $J_1 = \sum_{k=t+1}^{t+H} \left((1-w_1)\zeta^s(k) + w_1 \sum_{i=0}^f h_i(t)/f \right)$ となる。モデル予測制御では、 J_1 と合わせて、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。

各タイムスロット t でサービス s の管理者が解く最適化問題を以下に示す。

$$\begin{aligned} \text{minimize : } & \sum_{k=t+1}^{t+H} \left((1-w_1-w_2)\zeta^s(k) \right. \\ & \left. + w_1 \sum_{i=0}^f h_i(t)/f \right) \end{aligned}$$

$$+w_2\|\Delta R(k)\| \quad (8)$$

$$\text{subject to : } \forall k, \zeta^s(k) = \sum_{l \in L} \frac{[y_l^s(k) - B_{V_s, l}]^+}{|L|B_l} \quad (9)$$

$$\forall i, h_i(t) = \sum_k d_k r_{i,k}^s(t) \quad (10)$$

$$\forall k, \hat{Y}^s(k) = G \cdot R^s(k) \cdot \hat{X}^s(k) \quad (11)$$

$$\forall k, \forall i, \forall j, r_{i,j}^s(k) \in [0, 1] \quad (12)$$

$$\forall k, \sum_{i \in \varphi(j)} r_{i,j}^s(k) = 1 \quad (13)$$

ここで、目的関数における w_1 はホップ数に対する重み、 w_2 は経路変更に対する重みである。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 $R^s(t+1), \dots, R^s(t+H)$ が得られるが、実際に投入するのは、直近の設定 $R^s(t+1)$ のみで、以降の設定は、新たな観測値を基に予測値 $\hat{x}(k)$ 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。

4. モデル予測制御にもとづく仮想ネットワーク間資源調停

本稿では、モデル予測制御を仮想ネットワーク間の資源調停にも適用する。物理ネットワーク管理者は、各仮想ネットワークのリンク上に流れるトラフィック量の予測値 $\hat{Y}^s(t)$ を入力とし、時刻 k における各仮想ネットワークに割り当てる各リンクの帯域 $B_{V_s, l}(k)$ を定める。その際、制御の目的は、すべての仮想ネットワークに十分な帯域を割り当てることである。ただし、各仮想ネットワークでは、動的に経路を変更するため、仮想ネットワーク上の特定のリンクを流れるトラフィック量が増加すると予測されたとしても、増加分のトラフィックは他のリンクに迂回させることにより、輻輳が回避される可能性がある。そのため、仮想ネットワーク間資源調停では、リンク上に流れるトラフィックが迂回される可能性についても、考慮する必要がある。

そこで、本稿では、各仮想ネットワークに割り当てる各リンクの帯域 $B_{V_s, l}(k)$ 以外にも、各リンクを流れているトラフィックを仮想ネットワーク内の経路制御により迂回させる可能性を考慮し、サービス s について、 i 番目のリンクを流れるトラフィックを j 番目の迂回経路に迂回させる割合 $r_{i,j}^s(k)$ も変数として導入する。なお、 $r_{i,0}^s$ は i 番目のリンクをそのまま利用して送る場合に対応させる。

これらの入力値と変数を用いることにより、時刻 $k+1$ に各リンクを流れる各サービスのトラフィックの予測値 $\hat{Y}^s(k+1)$ は、以下のように表すことができる。

$$\hat{Y}^s(k+1) = G \cdot R^s(k+1) \cdot \hat{Y}^s(k+1) \quad (14)$$

3.と同様に、 $\hat{Y}^s(k+1)$ を用いて、全リンクにおいて与えられた帯域を超えて送信されたトラフィック量を示す変数 $\zeta^s(k)$ を以下のように定義する。

$$\zeta^s(k) = \sum_{l \in L} \frac{[y_l^s(k) - B_{V_s, l}]^+}{|L|B_l} \quad (15)$$

輻輳を回避するために、 $\zeta^s(k)$ を最小化することが制御の目標

となる。

ただし、この問題を解くにあたり $r_{i,j}^s(k)$ を変数として導入しているものの、実際の経路変更は仮想ネットワークを管理しているサービス提供者が行うため、実際の仮想ネットワーク内の経路は、資源割り当ての際に計算された $r_{i,j}^s(k)$ とは異なる。また、 $r_{i,j}^s(k)$ を急激に変更することにより、輻輳を解消できる場合であっても、サービスの安定的なトラフィック収容のためには、急激な経路変更は行うことができない場合もあり、可能であれば、 $B_{V_s, l}$ の変更により、輻輳を解消できることが望ましい。そこで、本稿では、可能な限り迂回経路を取らずに輻輳の解消を試みるよう、3.と同様、トラフィック収容先の経路のホップ数も目的関数に含める。

なお、 i 番目のリンクを流れるトラフィックの k 番目の迂回経路のホップ数を d_k と置くと、 i 番目のリンクを流れるトラフィックの迂回後の平均ホップ数は以下の式であらわすことができる。

$$h_i^s(t) = \sum_k d_k r_{i,k}^s(t) \quad (16)$$

そして、3.と同様、この2点について重み付きの和を最小化することを目標とする。つまり、制御の目的関数は、 $J_1 = \sum_{k=t+1}^{t+H} ((1-w_1)\zeta^s(k) + w_1 \sum_{i=0}^f h_i(t)/|L|)$ となる。モデル予測制御では、 J_1 と合わせて、予測ホライズン全体での経路割当の変更量 $J_2 = \sum_{k=t+1}^{t+h} \sum_s \sum_l \|\Delta B_{V_s, l}\|$ も同時に最小化することで、急激な割り当て資源量の変更を防止する。

各タイムスロット t で物理ネットワークの管理者が解く最適化問題を以下に示す。

$$\begin{aligned} \text{minimize : } & \sum_{k=t+1}^{t+H} ((1-w_1-w_2) \sum_s \zeta^s(k) \\ & + w_1 \sum_s \sum_{i=0}^f L|h_i^s(t)/|L| \\ & + w_2 \sum_s \sum_l \|\Delta B_{V_s, l}\|) \end{aligned} \quad (17)$$

$$\text{subject to : } \forall s \forall k, \zeta^s(k) = \sum_{l \in L} \frac{[y_l^s(k) - B_{V_s, l}]^+}{|L|B_l} \quad (18)$$

$$\forall s \forall i, h_i(t) = \sum_k d_k r_{i,k}^s(t) \quad (19)$$

$$\forall l \sum_s B_{V_s, l} = B_l \quad (20)$$

$$\forall s \forall k, \hat{Y}^s(k) = G \cdot R^s(k) \cdot \hat{X}^s(k) \quad (21)$$

$$\forall s \forall k, \forall i, \forall j, r_{i,j}^s(k) \in [0, 1] \quad (22)$$

$$\forall s \forall k, \sum_{i \in \varphi(j)} r_{i,j}^s(k) = 1 \quad (23)$$

5. 評価

5.1 評価環境

本稿では、提案する資源再割り当て手法の動作の確認のため、以下の環境において動作確認を行った。

5.1.1 ネットワークトポロジ

図1に示す、9ノードの格子型ネットワークを用いる。図中のリンクは双方向リンクであり、全リンクともに物理帯域を20

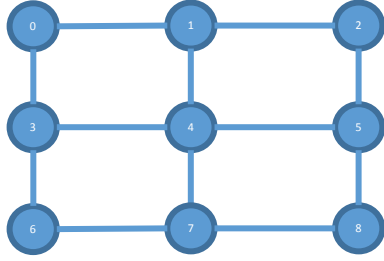


図1 評価に用いたネットワークポロジ

と設定する。

5.1.2 仮想ネットワークに流れるトラヒック量

本評価では、2つの仮想ネットワークを収容した環境で評価する。本評価では、各仮想ネットワークが必要とする資源が、時間によって大きく変動する環境を評価する。そのような時間によって大きく変動する環境を生成するため、トラヒック量は、以下のような sin 関数を元にした関数で生成した。

$$x_i^s(t) = \left[A \sin \left(\frac{t + \theta}{N} \right) + \eta + B \right]^+ \quad (24)$$

ただし、 $[x]^+$ は $x > 0$ であれば x 、それ以外は 0 を出力し、 A は変動の振幅、 B は平均、 N は周期、 θ は位相を表し、 η は平均 0、分散 σ の正規乱数である。

以下に、本評価で用いた各仮想ネットワークに流すトラヒックについて述べる。

a) 仮想ネットワーク 1

仮想ネットワーク 1 では、ノード 0 から 2 と、ノード 3 から 5 の 2 本のトラヒックを流す。いずれのトラヒックも振幅は 6、平均は 8、周期は 24 時間、正規乱数の分散の大きさは 0.5 と定めた。位相は、ノード 0 から 2 のトラヒックは 0 とし、ノード 3 から 5 のトラヒックは 6 時間ずらしたものとした。

b) 仮想ネットワーク 2

仮想ネットワーク 2 では、ノード 3 から 5 と、ノード 6 から 8 の 2 本のトラヒックを流す。仮想ネットワーク 1 と同様、いずれのトラヒックも振幅は 6、平均は 8、周期は 24 時間、正規乱数の分散の大きさは 0.5 と定めた。位相は、ノード 3 から 5 のトラヒックは 12 時間とし、ノード 3 から 5 のトラヒックは 18 時間ずらすことにより、要求帯域が最も大きくなる時間帯を仮想ネットワーク 1 とずらした。

5.1.3 トラヒック予測方法

各仮想ネットワークにおける経路制御、仮想ネットワーク間の資源調停のいずれも、トラヒック量の予測を必要とする。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。

この予測方法は単純であり、大きな予測誤差を含むものの、トラヒックの増加傾向・減少傾向をとらえたものとなる。そのため、この予測方法を用いて評価を行うことにより、増加現象の傾向をとらえつつも、正確な予測ができない場合の資源調停

手法の評価を行うことができる。

5.1.4 仮想ネットワーク上の経路制御

各仮想ネットワーク上の経路制御は、1分に1回の間隔で行う。その際、ホライゾン $h = 10$ とし、予測には、過去 10 回の観測結果を用いる。また、経路制御の際には、 $w_1 = 0.02$ 、 $w_2 = 0.4$ と設定した。

5.1.5 資源割り当ての間隔

本評価では、30分に一度資源割り当てを変更するものとした。その際、各リンク上のトラヒック量は 30 分間の平均値を観測し、観測した結果を元に予測を行って用いる。

5.2 比較対象

5.2.1 提案手法

4. で述べた最適化問題を CPLEX [5] で解くことにより、資源割り当てを行った。本評価では、ホライゾン $h = 2$ とし、予測の際には過去 5 回の観測結果を用いた。最適化問題における目的関数の重みは、 $w_1 = 0.99$ 、 $w_2 = 0.002$ とし、可能な限りトラヒックの迂回がなくても帯域を確保できる経路を割り当てを行うようにした。

5.2.2 観測結果ベースの手法

直前 30 分のリンク上のトラヒック量の観測結果を用い、次の 30 分間の資源割り当てを行った。この手法では、直前 30 分間の各リンク上のトラヒック量を入力とし、 $h = 1$ として 4. で述べた最適化問題を CPLEX [5] で解くことにより、資源割り当てを行った。その際、経路変更の重みは、提案手法と同じく、 $w_1 = 0.99$ 、 $w_2 = 0.000$ と設定することにより、観測されたトラヒック状況に合わせた資源割り当てを行う。

5.2.3 資源再割り当てなし

評価開始時に、最初の時刻に各リンクに流すトラヒック量を入力とし、 $w_1 = 0.99$ 、 $w_2 = 0.000$ とし 4. で述べた最適化問題を CPLEX [5] で解くことにより、当該時刻に適した資源割り当てを行った。その後は、資源の再割り当ては行わない。提案手法を、このような資源再割り当てなしの場合と比較することにより、資源再割り当ての必要性を確認する。

5.3 結果

図 2 に結果を示す。図では、各仮想ネットワークについて、以下で定義される最大リンク使用率の変化を示す。

$$\max_{l \in L} \frac{y_l^s(t)}{B_{V_s, l}}$$

図より、資源再割り当てをしない場合であれば、本評価開始時においてトラヒック量が少なかった仮想ネットワーク 2 には十分な帯域を割り当てられず、仮想ネットワーク 2 のトラヒック量が増加する時間帯においては、著しくリンク使用率が高くなっていることがわかる。その一方、仮想ネットワーク 1 においても、十分な資源を確保できておらず、リンク使用率が 1 を上回る時間帯も生じている。このため、各仮想ネットワーク内の経路制御のみでは解消できない輻輳が生じており、資源の動的な再割り当てが必要であることが分かる。

また、観測ベースの手法の場合では、著しくリンク使用率が高い時間帯が生じていることが分かる、これは、一方の仮想ネットワークに割り当てられる資源が著しく少ない場合に発生して

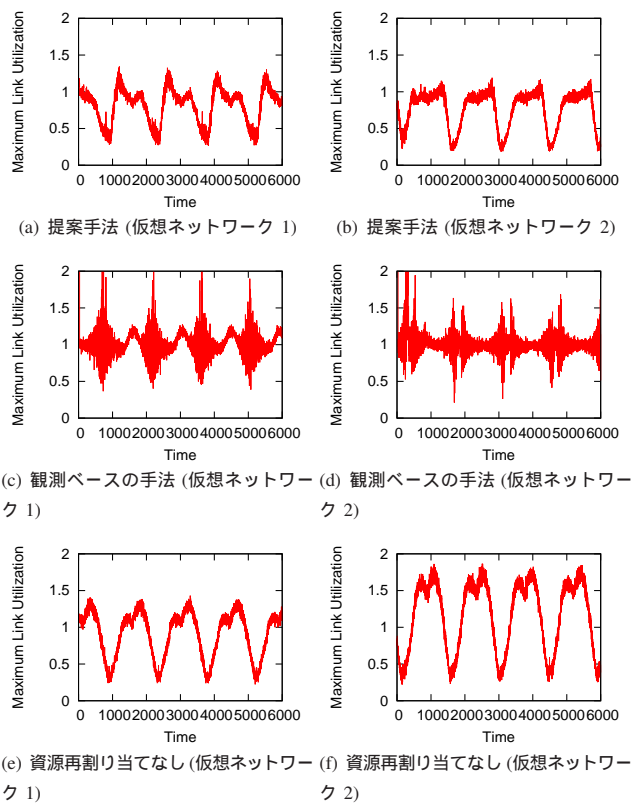


図2 各仮想ネットワークにおける最大リンク使用率の変化

おり、直前のタイムスロットで観測されたトラフィック量をもちいて資源割り当てを行った結果、必要な帯域は著しく少ないと考えられた仮想ネットワークへ割り当てる資源を激減させたものの、その次の時刻では、当該仮想ネットワークのトラフィックが増加し、トラフィック量に対して割り当て帯域が極端に小さくなるのが原因である。

それに対して、提案するモデル予測制御にもとづく資源割り当て手法では、いずれの仮想ネットワークの最大リンク使用率も1前後以下に抑えることができている。これは、予測した値をもとに資源割り当てを行うことにより、資源が不足しそうな仮想ネットワークに前もって資源を割り当てることができていることに加え、急激な資源割り当ての変化を抑えることにより、予測の誤差が生じた場合であっても、誤差の影響を受けた割り当て帯域の急減を防いでいるためである。

6. まとめ

本稿では、モデル予測制御にもとづく仮想ネットワーク間資源調停手法を提案した。そして、単純なトポロジにおける評価を行い、提案手法が、全仮想ネットワークで輻輳を防ぐことができるような資源割り当てを行うことができていることを示した。

今後は、より大規模なネットワークにおいて、実ネットワークでのトラフィック観測結果にもとづいた評価を行い、手法の有効性を確認するとともに、手法におけるパラメータの影響について考察を行う予定である。

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] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "Flowvisor: A network virtualization layer." Technical Report, Stanford University, 2009.
- [2] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, pp. 733–764, July 2003.
- [3] 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, pp. 7–12, October 2013.
- [4] 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, "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," 電子情報通信学会 技術研究報告 (IN2013-194), March 2013.
- [5] "IBM ILOG CPLEX Optimizer." optimization software : <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer>.