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
Email: {t-otoshi, y-ohsita, murata}@ist.osaka-u.ac.jp
††NTT Network Technology Laboratories, NTT Corporation
Email: {takahashi.yousuke, ishibashi.keisuke, shiomoto.kohei}@lab.ntt.co.jp
‡Department of Mechanical Engineering, Osaka Institute of Technology
Email: tomoaki.hashimoto@oit.ac.jp

***Abstract*—Traffic engineering (TE) plays an essential role in deciding routes that effectively use network resources. Since managing the routes of a large network takes a large overhead, multiple controllers are introduced in the network which hierarchically decide the routes. We call this approach *hierarchical TE*. In hierarchical TE, avoiding the route oscillation is a main problem since routes change at a layer causes the additional routes changes at other layers. The existing hierarchical TE avoids these route oscillation by setting the longer control interval on the upper layer. This approach, however, causes another problem that the routes change of upper layer delays to traffic changes. In this paper, we propose a hierarchical TE method called *hierarchical model predictive traffic engineering (hierarchical MP-TE)* which avoids routing oscillation without setting the long control interval. In hierarchical MP-TE, each control server gradually changes the routes based on the traffic prediction to stabilize the routing instead of setting the long control interval. Through the simulation, we show that the hierarchical MP-TE achieves the routing convergence with the short control interval.**

***Index Terms*—Model Predictive Control, Traffic Engineering, Topology Aggregation, Traffic Prediction**

I. Introduction

Traffic engineering (TE) is a promising solution for handling time variation of traffic [1, 2]. In the TE, a control server periodically collects the traffic information, and changes the routes of the flows within the network based on the collected traffic information. By dynamically reconfiguring the routes, these methods avoid the congestion even when traffic change occurs.

Because handling a large number of flows takes a large overhead of the control server, hierarchical multiple controllers are deployed in a large network [3, 4]. We call these approach *hierarchical TE*. In the hierarchical TE, the network is hierarchically divided into multiple areas; the area with the lowest layer is constructed from a small number of nodes, and the area with the upper layer is constructed from multiple areas of the lower layer. A control server is deployed at each area of each layer. Each control server collects the traffic information and calculates the routes within the corresponding area. In addition, the control servers of the upper layer use the aggregated information on the network topology and traffic. As a result, a large overhead of the control server is avoided.

In the hierarchical TE, the route oscillation is an important problem. The route changes of the upper layer causes the traffic passing through areas of the lower layer, which may cause the necessity of the route change of the lower layer. On the other hand, the route changes of an area of the lower layer changes the available capacity of the area, which stimulates the route changes of the upper layer. Such interaction between layers causes the route oscillation.

The commonly used way to handle the route oscillation in the hierarchical TE is to set the control interval of the upper layer to a large value [5,6]. By doing so, the control servers of the lower layer change the routes with sufficient times before the other layers changes the routes. Moreover, the impact of the route changes of the lower layer on the upper layer is also avoided by using the averaged traffic information. However, a large control interval may increase the time required to mitigate the congestion; if all links of an area are congested, the congestion cannot be mitigated until the control server of the upper layer changes the routes.

In this paper, we introduce a new mechanism to avoid the route oscillation without setting a large control interval. Our method is based on the *model predictive control (MPC)*, which changes the input gradually based on the prediction so as to maintain system output at close to a target value. We call the hierarchical TE based on the MPC *hierarchical MP-TE*. In the hierarchical MP-TE, each control server gradually changes the routes based on the traffic prediction. As a result, each control server reduces the impact on the routing of other layers, and avoids the unnecessary routes changes induced by other layers. Thus, the route oscillation is avoided without setting a long control interval. Even when traffic change causes the congestion that cannot be mitigated by the route change within an area, the hierarchical MP-TE can mitigate the congestion immediately, because the control server at upper layer can change the inter-area routes to avoid the congested area quickly with the short control interval.

Though we have already applied the MPC to the TE [7, 8], the target of our previous work is to avoid congestion in a small network, and the impact of the MPC on the hierarchical TE was not investigated. In this paper, we evaluate the hierarchical MP-TE by simulation, and demonstrate that the hierarchical MP-TE can avoid route oscillation even when control intervals of all layers are set to small values.

The rest of this paper is organized as follows. Section II explains the overview of hierarchical TE. Section III describes hierarchical MP-TE which we propose in this paper. Section IV presents an simulation result of hierarchical MP-TE. Section V presents our concluding remarks.

II. Hierarchical Traffic Engineering

In the hierarchical TE, the network is hierarchically divided into areas; the areas of the lowest layer are constructed of a small number of nodes, and the areas of the upper layer are constructed of multiple lower layers. Hereafter, we call the set of the hierarchically divided networks the *hierarchical network*. The control server is deployed at each area of each layer. In the upper layer, the control server calculates the routes of the flows between areas of the lower layer over the aggregated network topology including only the nodes at the border of the areas of the lower layer. The routes of the flows within the areas of the lower layer are calculated by the control servers at the lower layer. The rest of this section explains the construction of the hierarchical network, and the overview of the TE on the hierarchical network.

A. Construction of the Hierarchical Network

The hierarchical network is constructed by *area partitioning* and *topology aggregation*.

1) Area Partitioning: The area partitioning divides the network into multiple areas so that each area includes the connected subnetwork of the original network. In this paper, we use the area partitioning, which divides the network so that any nodes are included in one of the areas, and no nodes are included in multiple areas. We denote the set of nodes included in the area a by V_a. The set of the links E_a included in the area a is the set of the links $\{(i, j) \in E | i; j \in V_a\}$ where E is the set of all links of the original network. In the above area partitioning, the links connecting the nodes within different areas are not included in any areas, and included in the upper layer.

We can use any area partitioning strategy. For instance, a partitioning method which minimizes the control overhead such as the total bits of information exchanged for intra-area and inter-area routing is proposed in [9]. In the evaluation described in Section IV, we manually divide network into areas so that each area includes the same number of nodes.

2) Topology aggregation: In the hierarchical TE, the control server of the upper layer maintains the aggregated network topology instead of the original network topology so as to avoid a large calculation time. The topology aggregation replaces each area of the lower layer by the set of a small number of nodes and links connecting them. There are a lot of methods to aggregate the topology information [4, 10]. For example, Lui et al. proposed an aggregation method that aggregates the topology as a star topology with some bypass links [4]. This method can provide the information about delay and bandwidth within inner-area with the topology complexity of $O(B)$ where B is the number of border nodes.

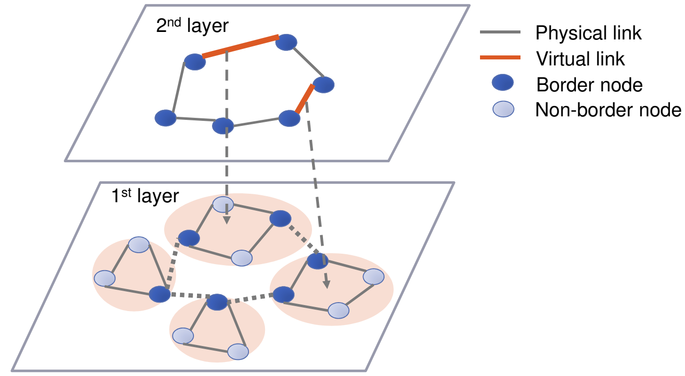


Fig. 1. Hierarchical Network Model

In this paper, we use the full-mesh topology to aggregate topology so as to keep the accurate information between the nodes at the borders of the areas. By using the full mesh topology, the abstracted topology of an area includes the set of the nodes at the border and the set of the links between all pairs of the nodes at the border. Hereafter we call the links generated by the topology aggregation the *virtual links*.

Figure 1 shows an example of the hierarchical network. In this network, the upper layer includes the virtual links and the physical links between different areas.

B. Traffic Engineering

Each control server periodically 1) collects the information on the traffic rates and the link capacities within the area, 2) calculates the routes of the flows based on the collected traffic information, and 3) configures the network devices based on the calculated routes.

1) Collection of Information: The control server collects the traffic information from the nodes within the area at a fixed interval (e.g. one second, one minute, and one hour), with the times observations called *time slots*. Each node monitors traffic rates per source and destination address pair. The control server collects the traffic rates monitored by each nodes, and calculates the sums of the traffic rates of the flows that are from the same node to another same node within the area. We represent the observed traffic rates by each control server at area a of layer m as a vector $\boldsymbol{x}^{m;a}(k)$ whose element $x^{m;a}_{i;j}(k)$ is the traffic rate from nodes i to j.

The control server also collects the information on the link capacities available for the flows whose routes are controlled by the server. At the lowest layer, the routes of all flows passing through the area are controlled by the control server, and the capacity of the physical link is available for the flows. However, the control server at the upper layer cannot change the routes of the flows whose source and destination nodes belong to the same area at the lower layer, and the flows maintained by the control server can use only the residual capacity. We represent the link capacities by $\boldsymbol{C}^{m;a}$ whose element $C_l^{m;a}$ is the link capacity of l that can be used by the flows whose routes can be changed by the control server.

2) Route Calculation: The control server calculates the routes within the area based on the collected information. The routes are defined by the fraction of traffic of each flow sent to each path. We denote the fractions by a matrix $R^{m;a}(k)$ whose element $R_{i;j}^{m;a}(k)$ indicates the fraction of traffic on the flow j that traverses the available path i. When the routes are decided by $R^{m;a}(k)$, the traffic rate on each link is calculated by

$$\boldsymbol{y}^{m;a}(k) = G^{m;a} \cdot R^{m;a}(k) \cdot \boldsymbol{x}^{m;a}(k) \tag{1}$$

where $\boldsymbol{y}^{m;a}(k)$ is a vector whose element $y_l^{m;a}(k)$ is a traffic rate on the link l, $G^{m;a}$ is a matrix whose element $G_{i;j}^{m;a}$ is 1 if the available path j traverses the link i and 0 otherwise. The control server calculates routes so that

$$\forall l, y_l^{m;a} \leq C_l^{m;a}. \tag{2}$$

3) Configuration of Network: Finally, the control server configures the network based on the calculated routes. The calculated routes can be set by the technologies such as the OpenFlow [11]. By using the Openflow, the control server configures the routing tables of the nodes within the area.

C. Problem of Hierarchical TE

In the hierarchical TE, the route oscillation may become a serious problem. The route changes of the upper layer causes the traffic passing through the areas of the lower layer, which may cause the necessity of the route change of the lower layer. On the other hand, the route changes of an area of the lower layer changes the available capacity of the area, which stimulates the route changes of the upper layer. Such interaction between layers causes the route oscillation.

The typical approach to handling the route oscillation is setting the long control interval at upper layer. This method however requires long time to set appropriate routes because the long control interval delays the route changes of the upper layer.

III. Hierarchical MP-TE

A. Model Predictive Control

First, we briefly explain the concept of MPC. MPC is a method of system control based on predictions of system dynamics that has been studied in recent years. Figure 2 shows an overview of MPC. A MPC controller sets an input so as to maintain system performance at close to an operator-specified target. Unlike traditional system control, the MPC controller predicts how the output changes to calculate inputs for the *predictive horizon*, time slots $[t+1, t+h]$ where h is the distance to the predictive horizon. We denote the input and output at the time slot k by $u(k)$ and $y(k)$, respectively. The MPC controller calculates $u(k)(k \in [t+1, t+h])$ so as to keep $y(k)$ close to the target value $r_y(k)$. In other words,

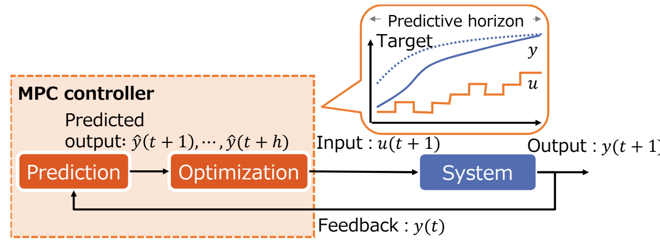


Fig. 2. Overview of MPC

the MPC controller minimizes an objective function $J_1 = \sum_{k=t+1}^{t+h} \|y(k) - r_y(k)\|^2$, where $\|\cdot\|$ represents the Euclidean norm. Moreover, the MPC controller restricts the amount of allowed changes in input at each time slot to avoid instability of system due to drastic shifts of system state. We denote the amount of change in the input at the time slot k by $\Delta u(k) = u(k) - u(k-1)$, and the aggregated amount of change during the predictive horizon by $J_2 = \sum_{k=t+1}^{t+h} \|\Delta u(k)\|^2$. Thus, the MPC controller determines $u(k)$ as:

$$(u(t+1), \cdots, u(t+h)) = \underset{(u(t+1),\cdots,u(t+h))}{\arg\min} (1-w)J_1 + wJ_2 \tag{3}$$

where $0 \leq w \leq 1$ is a parameter for weighting the two objective functions J_1 and J_2.

To solve the above optimization problem, future outputs $y(t+1). \cdots, y(t+h)$ must be predicted from inputs $u(t+1), \cdots, u(t+h)$. The future output under a given input is calculated by a system model that represents the system dynamics. In system control, a system model is often represented by a mathematical formula, the *state space representation*, described as

$$z(k+1) = \phi(k, z(k), u(k)) \tag{4}$$
$$y(k) = \psi(k, z(k), u(k)), \tag{5}$$

where $z(k)$ is the state of the system at the time slot k, and ϕ, ψ are functions that respectively map the current state and input onto the next state and ouput.

Modeling the system by a mathematical formula, however, may entail modeling errors, such as the use of ϕ, ψ that do not well represent actual system dynamics. Predictions of system output will be inaccurate under an incorrect model, and prediction error becomes increasingly large with more distant predictive horizons. The MPC controller therefore implements only the first of the calculated inputs $u(t+1)$. Then, the MPC controller observes the output and corrects the prediction by using the output value as feedback. After prediction correction, the MPC controller recalculates the input value for the next time slot with the corrected prediction.

B. Hierarchical MP-TE

In this subsection, we propose the hierarchical MP-TE. In the hierarchical MP-TE, each control server performs the following steps based on the MPC; it 1) collects the information on traffic rates and available capacities within the corresponding area, 2) predicts future traffic rates and available capacities, 3) calculates the routes based on the prediction, and 4) configures the network devices within the range. By continuing these steps, each control server controls the routes within the range so as to follow the traffic changes. Because each control server gradually changes the routes, route changes at each time slot are not drastic, and have only a small impact on the other layers.

The rest of this subsection explains the details of the above steps.

1) Collection of Information: Each control server collects the information on the traffic rates and available capacities within the range in a similar way described in Subsection II-B1.

In this paper, the capacity of the virtual link is set to the total capacity of all the available paths between both ends of the virtual link. That is, the capacity of the link l is set by

$$C_l(k) = \sum_{p \in P(i)} \min_{i \in L(p)} [C_i^{m-1} - y_i^{m-1}(k)]^+ + y_l^m(k) \tag{6}$$

where $y_i^m(k)$ is the traffic rate on link i, $P(i)$ is the set of paths on the inner-area whose starting and ending nodes are same as that of virtual link i, and $L(p)$ is the set of physical links which are included in a path p. In this equation, $\min_{l \in L(p)}[C_l - y_l^{m-1}(k)]^+$ denotes the capacity of the path p which equals the capacity of bottleneck link on the path, and the virtual link capacity sums up the path capacity for all available paths. Note that $C_i^m(k)$ contains the congestion information of only the lower layer by adding the last term $y_i^m(k)$.

The paths between the both ends of a virtual link may pass through the same physical links as the paths between the both ends of the other virtual links, and the unpredictable congestion may occur when the route calculation of the upper layer increases the traffic on multiple virtual links at the same time. However, because the control server does not change the routes drastically at each time slot, this problem does not occur in the hierarchical MP-TE.

C. Prediction

Each control server predicts future traffic rates and available capacities. We denote the predicted traffic rates as $\hat{\boldsymbol{x}}^{m;a}(k)$ and link capacities as $\hat{\boldsymbol{C}}^{m;a}(k)$. In the hierarchical MP-TE, any prediction method can be used. Though the prediction errors depend on the prediction methods, the suitable prediction method is out of scope of this paper. In the evaluation described in Section IV, we use one of the simplest prediction method, and demonstrate that the hierarchical TE works properly even in the case of inaccurate prediction.

D. Route Calculation

The control server calculates the routes so that all flows are accommodated without congestion. To achieve this objective, we define a metric called *exceeding traffic*. The exceeding traffic $\zeta_l^{m;a}(k)$ on the link l is defined by

$$\zeta_l^{m;a}(k) = [y_l^{m;a}(k) - C_l^{m;a}(k)]^+ \tag{7}$$

where $[x]^+$ equals x when $x \geq 0$ and equals 0 otherwise. By making $\zeta_l^{m;a}(k)$ zero, we can avoid the congestion. We define $\boldsymbol{\zeta}^{m;a}(k)$ as a vector whose element is $\zeta_l^{m;a}(k)$. By minimizing $\sum_{k=t+1}^{t+h} \|\boldsymbol{\zeta}^{m;a}(k)\|^2$ where t is the current time slot and h is the length of predictive horizon, we can avoid future congestion.

In the hierarchical MP-TE, we use another objective that the drastic route change should be avoided. For this objective, $\Delta R^{m;a}(k) = R^{m;a}(k) - R^{m;a}(k-1)$ should be small.

Considering both of the above objectives, each control server solves the following optimization problem.

$$minimize: \sum_{k=t+1}^{t+h} \left(\frac{1-w}{N_L^{m;a}} \left\| \frac{\hat{\boldsymbol{\zeta}}^{m;a}(k)}{Z^{m;a}} \right\|^2 + \frac{w}{N_P^{m;a}} \|\Delta R^{m;a}(k)\|^2 \right) \tag{8}$$

$$subject\ to: \forall k, \hat{\boldsymbol{y}}^{m;a}(k) = G^{m;a} \cdot R^{m;a}(k) \cdot \hat{\boldsymbol{x}}^{m;a}(k) \tag{9}$$

$$\forall k, l, \hat{\zeta}_l^{m;a}(k) = [\hat{y}_l^{m;a}(k) - C_l^{m;a}]^+ \tag{10}$$

$$\forall k, f, p, R_{p;f}^{m;a}(l) \in [0,1] \tag{11}$$

$$\forall k, f, \sum_{p \in \wp^{m;a}(f)} R_{p;f}^{m;a}(k) = 1 \tag{12}$$

where $\wp^{m;a}(f)$ is the set of the available paths of flow f, $Z^{m;a} = \max_{l,k}[\{G^{m;a} \cdot R^{m;a}(t) \cdot \hat{\boldsymbol{x}}^{m;a}(k)\}_l - C_l^{m;a}]^+$ is the maximum exceeding traffic if the current routes $R^{m;a}(t)$ is used during the predictive horizon, and $N_L^{m;a}, N_P^{m;a}$ are the numbers of links and paths respectively. Here, $\hat{\boldsymbol{x}}^{m;a}(k), G^{m;a}$ are given variables and $R^{m;a}(k), \hat{\boldsymbol{y}}^{m;a}(k), \hat{\boldsymbol{\zeta}}^{m;a}(k)$ are the variables to be optimized. Eq. (8) is the objective function which is the weighted summation of exceeding traffic $\boldsymbol{\zeta}^{m;a}(k)$ and the amount of routes change $\Delta R^{m;a}(k)$. To clarify the effect of weighting parameter w, we normalize the objective function with dividing $\hat{\boldsymbol{\zeta}}^{m;a}(k)$ by $Z^{m;a}$, and dividing the exceeding traffic on links and routes changes on paths by $N_L^{m;a}$ and $N_P^{m;a}$ respectively. Eq. (9) represents the relation between the traffic rates of the flows and links. Eq. (10) is the definition of ζ. Eqs. (11) and (12) mean that all traffic on each flow are allocated to some available paths.

Although the above optimization problem is not defined when $Z^{m;a} = 0$, this case is not critical for the TE because the current routes $R^{m;a}(t)$ minimizes both $\boldsymbol{\zeta}^{m;a}(k)$ and $\Delta R^{m;a}(k)$ when $Z^{m;a} = 0$. Therefore, in this paper, we calculate the routes by the above optimization problem only when $Z^{m;a} \neq 0$.

IV. Evaluation

In this section, we demonstrate that the hierarchical MP-TE absorbs the interactions between layers. In this evaluation, we change w to show the impact of restricting the route changes, and change h to show the impact of using the predicted traffic. In addition, we compare the hierarchical MP-TE with the typical hierarchical TE.

A. Simulation Environment

1) Network Topology: In the following evaluation, we use a lattice topology shown in Figure 3. The network contains 64 nodes, and the all links have a same link capacity of 2 Gbps. We divide the network into four areas as shown in the figure.

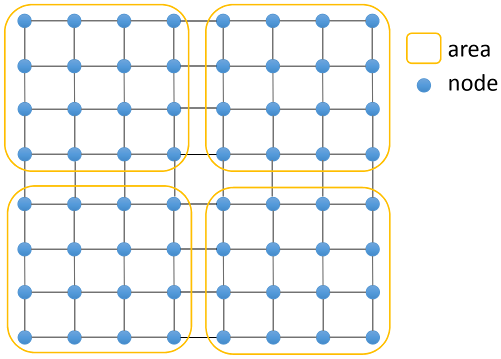


Fig. 3. Lattice Topology with 64 nodes

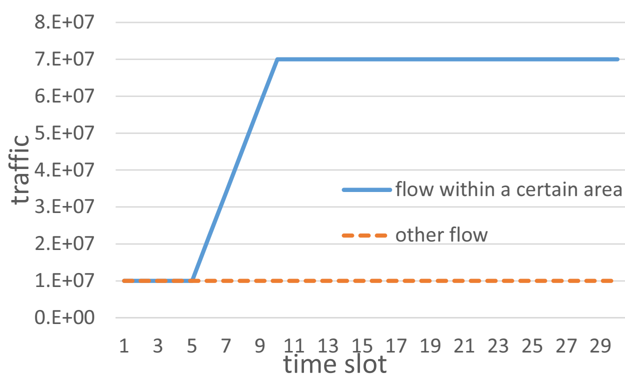


Fig. 4. Time Series of Traffic Used in the Evaluation

2) Traffic: To investigate the interaction between layers, we generate the traffic so that the congestion that cannot be solved by the route change at the lowest layer. We generate the traffic shown in Figure 4. In this traffic pattern, the traffic in an area linearly increases from the time slot 6 to 10, while the traffic in the other areas does not change. In this situation, an area becomes congested without the control of the upper layer.

Though we also conduct a simulation with increasing inter-area traffic, the general behavior of the MP-TE is same with the case of increasing inner-area traffic. In this paper, we only show the result of the case of increasing inner-area traffic due to the limitations of space.

3) Prediction Method: In this evaluation, we use a simple prediction method. First, we find a best-fit straight line $l(k) = ak + b$ that minimizes the sum of squared distances from the previous observed values $x(t-s), x(t-s+1), \cdots, x(t) (s \leq 1)$, denoted as $\sum_{k=0}^{s}(x(t-s+k)-l(t-s+k))^2$. We then obtain the future traffic rate or link capacity as $\hat{x}(t+k) = l(t+k)$. Though the generated traffic changes linearly, this prediction method cannot predict future traffic and capacity accurately, because the traffic rates and capacities maintained by each controller are affected by the route changes at the other layer. Though there are many more sophisticated prediction methods, the prediction method suitable to the hierarchical MP-TE is out of the scope of this paper.

4) Routing Calculation: In a similar way to [7], the optimization problem (8)–(12) is transformed as a convex quadratic programming problem, which can be solved by common solvers. To solve the optimization problem at each area of each layer, we use the CPLEX [12], which is an optimization problem solver. We run CPLEX on computers equipped with four Intel Xeon Processors, each having 10 cores and 30 MB of cache memory. The calculation time of each time slot is averagely lower than 10 seconds even with the 100 nodes.

5) Compared Methods: We use a simple hierarchical prediction-based TE in our comparison. In this method, the controllers at lower and upper layer simply calculate the routes without restricting the route changes. This TE method is a special case for our method when parameters are set as $h = 1$ and $w = 0$.

According to the traditional approach to stabilizing the routes, the control server of upper layer changes the routes with longer control interval than that of lower layer. In this evaluation, we introduce a parameter s, which is the ratio of the control interval of the upper layer to the lower layer. If s is set to a value larger than 1, the controller of the upper layer predict the future traffic and link capacity every s time slots using the previous average rates $\bar{x}(k) = \frac{1}{s}\sum_{i=(k-1)s}^{ks-1} x(i)$. Then, the control server calculates the routes using this predicted traffic and link capacity for next s time slots.

6) Metrics: We use $\zeta_l^{m;a}(k)$ and $\Delta R_{i;j}^{m;a}(k)$ as the metrics to evaluate the hierarchical TE. $\zeta_l^{m;a}(k)$ is used to check whether the calculated routes are appropriate or not to accommodate traffic. On the other hand, $\Delta R_{i;j}^{m;a}$ is used to check whether the routing is converged or not. We also use the following metrics:

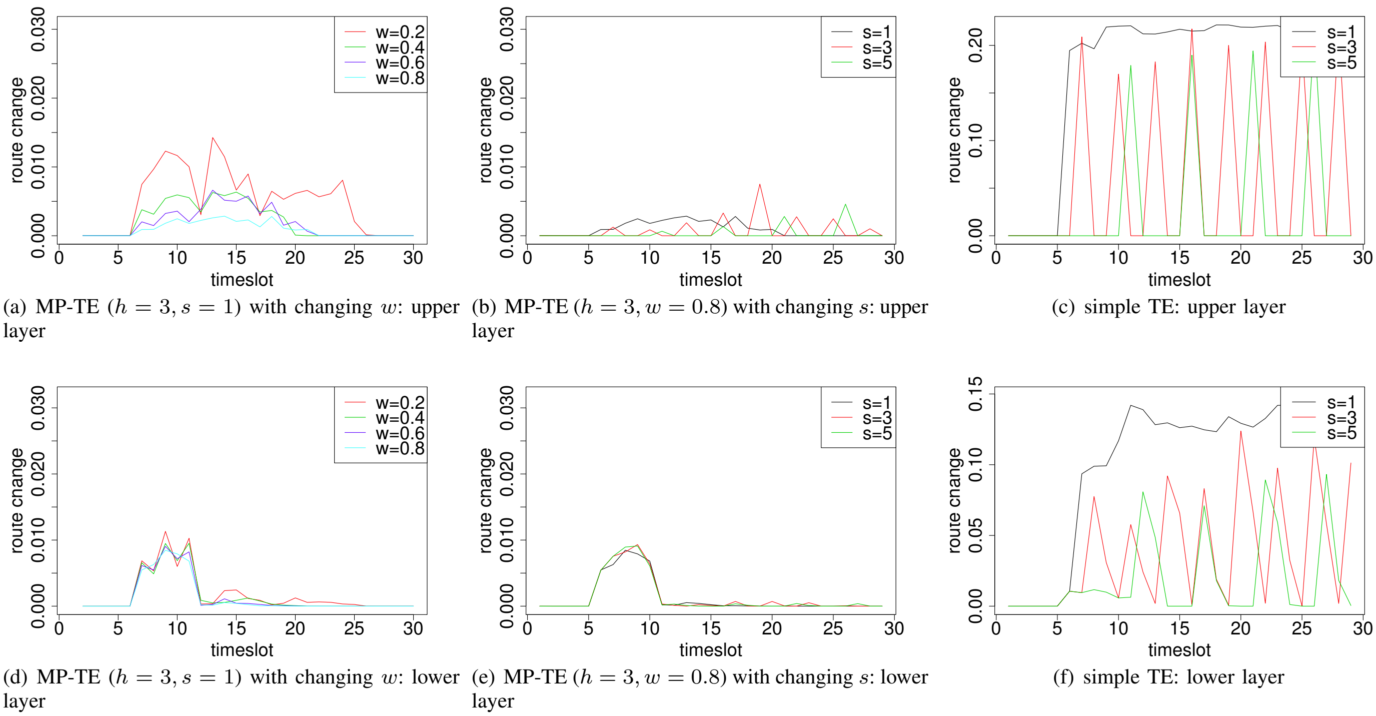
- *Routing convergence time*: the number of time slots from the first route change to the last route change.
- *Congestion time*: the amount of time slots where $\zeta_l^{m;a}(k)$ is not 0.
- *Total exceeding traffic*: the total amount of exceeding traffic during the simulation, which is defined as $\sum_{k,m,a,l} \zeta_l^{m;a}(k)$.

B. Results

Figure 5 shows the average value of routes change $|\Delta R^{m;a}(t)|$ at each time slot caused by MP-TE and simple TE. Figs. 5(a) and 5(d) show the result for the MP-TE and Figs. 5(c) and 5(f) show the result for the simple TE. We also show the result of MP-TE ($w = 0.8$) with changing the control interval at the upper layer in a similar way to simple TE method in Figs. 5(b) and 5(e). In this figure, we set h to 3 for the MP-TE. Figure 6 shows the maximum exceeding traffic for each case shown in Fig. 5.

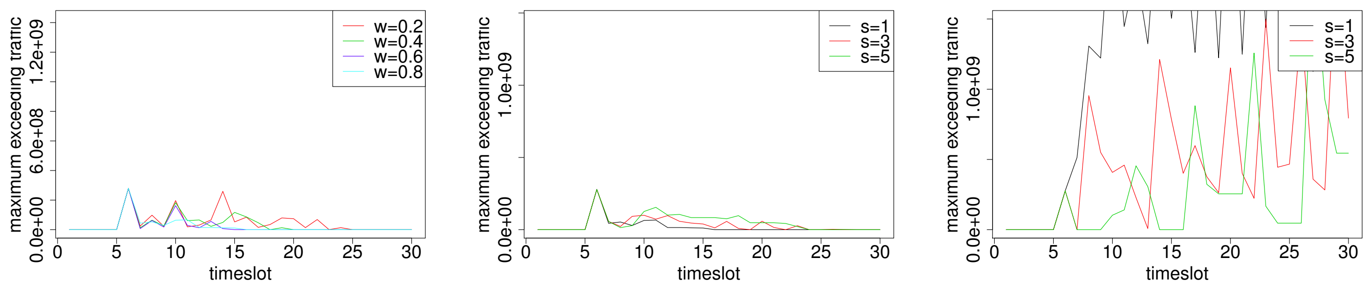
The rest of this subsection, we discuss the results shown in Figs. 5 and 6.

1) Impact of Restricting Routing Changes: All methods start the route changes at the time slot 7 after the increase of the traffic is detected. Though Figs. 5(c) and 5(f) show that the simple TE with $s = 1$ continues significant route



(a) MP-TE ($h = 3, s = 1$) with changing w: upper layer
(b) MP-TE ($h = 3, w = 0.8$) with changing s: upper layer
(c) simple TE: upper layer
(d) MP-TE ($h = 3, s = 1$) with changing w: lower layer
(e) MP-TE ($h = 3, w = 0.8$) with changing s: lower layer
(f) simple TE: lower layer

Fig. 5. Time Series of Average Routes Changes for the Cases of MP-TE and Simple TE



(a) MP-TE ($h = 3, s = 1$) with changing w
(b) MP-TE ($h = 3, w = 0.8$) with changing s
(c) simple TE

Fig. 6. Time Series of Traffic Rate Exceeding Targeted Capacity for the Cases of MP-TE and Simple TE

changes at all time slots after the time slot 7, Figs 5(a) and 5(d) show that the MP-TE completes the route changes by the time slot 25 even when $s = 1$. In addition, Fig. 6 indicates that the hierarchical MP-TE mitigates congestion after the route changes are completed. That is, restricting the routes change achieves the convergence of the routes without setting the long control interval.

We also compare the MP-TE with the different w. Fig. 5(a) shows that the large w reduces the route changes of the upper layer. But, w does not have the large impact on the route changes of the lower layer as shown in Fig. 5(d). In this simulation, traffic on all flows within the congested area of lower layer increases, and causes a large congestion. Thus, the control server at the congested area cannot reduce the routes changes not to cause the large exceeding traffic. On the other hand, the congestion is relatively small at the upper layer since the congestion occurs at only one area. Thus, the control server can reduce the routes changes without significant increase in the exceeding traffic.

2) Impact of the Control Interval of the Upper layer: Figs. 5(c) and 5(f) show that the route changes become slightly small when setting s to a large value. In addition, Fig. 6(c) shows that the maximum exceeding traffic is also reduced by setting s to a large value. However, setting s to 5 is insufficient to avoid the interaction between two layers, and the maximum exceeding traffic becomes large even when s is set to 5.

On the other hand, Figs. 5(b) and 5(e) show that setting s to a large value makes the convergence time larger in the hierarchical MP-TE. this is caused by that the large s delays the route changes of the upper layer. In addition, as discussed in the previous subsection, the hierarchical MP-TE achieves the convergence of the routes without setting s to a large value.

TABLE I
CONVERGENCE TIME AND EXCEEDING TRAFFIC FOR THE CASES OF MP-TE WITH VARYING THE PREDICTIVE HORIZON LENGTH

	$h = 1$	$h = 3$	$h = 5$
routing convergence time	31	16	18
congestion time	29	11	11
total exceeding traffic	2.85E+09	1.21E+09	1.63E+09

That is, s should be set to a small value in the hierarchical MP-TE.

3) Impact of the Length of the Predictive Horizon: We evaluate the MP-TE with the various lengths of predictive horizon. Table I summarizes the results. In this evaluation, w is set to 0.8.

Tab. I shows that the MP-TE with $h = 3, 5$ achieves the shorter convergence time and congestion time than the case of $h = 1$. This is because the prediction of the future time slots enables the controllers to change the routes in advance at the early time slots. Thus, the slight route changes at each time slot are sufficient to mitigate the congestion. As a result, the interaction between two layers are mitigated by predicting the future traffic.

The routing convergence time of the MP-TE with $h = 5$ is larger than the MP-TE with $h = 3$. This is caused by the prediction errors. The prediction errors become large as the target time slots become large with more distant predictive horizons. Thus, the prediction horizon should be chosen carefully. How to set the prediction horizon is one of our future work.

V. CONCLUSION

Setting the long control interval at upper layer is a common approach to avoid the route oscillation in hierarchical TE methods. However, it requires a long time to mitigate the congestion which cannot be solved by the routing in lower layer. To solve this problem, we have proposed hierarchical MP-TE to achieve the routing convergence with setting the short control period. In the hierarchical MP-TE, the network is hierarchically divided into multiple areas, and multiple controllers are deployed to calculate routes in a similar way to other hierarchical TE methods. To avoid the route oscillation, in the hierarchical MP-TE, each controller gradually changes the routes based on the predicted traffic instead of setting the long control interval. Through simulation, we demonstrated that the hierarchical MP-TE achieves the routing convergence by restricting the routes changes even with setting the short control interval. In addition, we showed that setting short control interval and using the multiple ahead future information improves the convergence time of hierarchical routing.

Future work will include the method determining an appropriate partitioning of a given network. Furthermore, we will conduct further verification of the hierarchical MP-TE using more realistic traffic.

ACKNOWLEDGEMENT

This work was supported in part by the Strategic Information and Communications R&D Promotion Programme of the Ministry of Internal Affairs and Communications, Japan.

REFERENCES

[1] S. Agarwal, M. Kodialam, and T. V. Lakshman, “Traffic engineering in software defined networks,” in *Proceedings of IEEE INFOCOM 2013*, Apr. 2013, pp. 2211–2219.

[2] 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.

[3] Y. Ohsita, T. Miyamura, S. Arakawa, S. Kamamura, D. Shimazaki, K. Shiomoto, A. Hiramatsu, and M. Murata, “Aggregation of traffic information for hierarchical routing reconfiguration,” *Computer Networks*, vol. 76, pp. 242–258, Jan. 2015.

[4] 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.

[5] B.-J. Chang and R.-H. Hwang, “Distributed cost-based update policies for QoS routing on hierarchical networks,” *Information Sciences*, vol. 159, no. 1–2, pp. 87–108, Jan. 2004.

[6] M. Chamania, X. Chen, A. jukan, F. Rembach, and M. Hoffmann, “An adaptive inter-domain PCE framework to improve resource utilization and reduce inter-domain signaling,” *Optical Switching and Networking*, vol. 6, no. 4, pp. 259–267, Dec. 2009.

[7] T. Otoshi, Y. Ohsita, M. Murata, Y. Takahashi, N. Kamiyama, K. Ishibashi, K. Shiomoto, and T. Hashimoto, “Traffic engineering based on model predictive control,” *IEICE Transactions on Communications*, vol. E09-B, no. 6, pp. 996–1007, Jun. 2015.

[8] T. Otoshi, Y. Ohsita, M. Murata, Y. Takahashi, K. Ishibashi, K. Shiomoto, and T. Hashimoto, “Traffic engineering based on stochastic model predictive control for uncertain traffic change,” in *Proceedings of The Seventh IFIP/IEEE International Workshop on Management of the Future Internet*, Ottawa, May 2015, pp. 1165–1170.

[9] X. Li, P. Djukie, and H. Zhang, “Zoning for hierarchical network optimization in software defined networks,” in *Proceedings of IEEE Network Operations and Management Symposium 2014*, May 2014, pp. 1–8.

[10] S. Uludag, K. shan Lui, K. Nahrstedt, and G. Brewster, “Analysis of topology aggregation techniques for QoS routing,” *ACM Computing Surveys*, vol. 39, no. 3, Sep. 2007.

[11] lan F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, “A roadmap for traffic engineering in SDN-OpenFlow netwoks,” *Computer Networks*, vol. 71, pp. 1–30, Oct. 2014.

[12] “IBM ILOG CPLEX Optimizer,” optimization software : http://www-01.ibm.com/software/integration/optimization/cplex-optimizer.