Master's Thesis

Title

Prediction-based Control Theoretic Approach for Robust Traffic Engineering

Supervisor

Professor Masayuki Murata

Author

Tatsuya Otoshi

February 10th, 2014

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

Prediction-based Control Theoretic Approach for
Robust Traffic Engineering

Tatsuya Otoshi

Abstract

In recent years, the time variation of the Internet traffic has become large due to the growth of Internet services such as streaming and clouds. A backbone network has to accommodate such traffic without congestion. So far, backbone networks have addressed this problem by preparing the redundant link capacity so as to accommodate not only the average traffic but also traffic surge. However, this approach requires higher cost according as the average and variance of traffic increase. Moreover, this approach causes the waste of energy consumption due to the poor utility of network resources. Hence, a method to accommodate traffic without congestion on the network with limited resources is required to reduce such costs and power consumption caused by the over provisioning. Traffic Engineering (TE) is one approach to accommodating the time-varying traffic with limited resources. In the TE, a control server periodically observes traffic in a network and dynamically changes the routes so as to minimize the network congestion. However, TE using only the observed traffic mitigates only the observed congestion and cannot avoid the future congestion until the next control cycle. TE combined with the traffic prediction is one approach to solving such problem. In this approach, the control server periodically predicts the time variation in traffic, and then calculates the routes based on the predicted traffic. Naturally, the predicted traffic includes the prediction errors, which may cause the congestion. In this thesis, we propose a prediction-based TE which is robust to prediction errors. To achieve the robust control, our method uses the idea of Model Predictive Control (MPC), which is a method of process control based on the prediction of the dynamics of the system. In our method, the routes are calculated so that the congestion in the future time slots is avoided without sudden route changes based on the predicted traffic. Then, we apply the calculated routes for the next time slot, and observe traffic. By using the newly observed traffic, we predict the future traffic and calculate the routes again.

By continuing these steps, the impact of the prediction errors are mitigated because the traffic prediction is corrected in each time slot. Through the simulation using the actual traffic trace of a backbone network, we demonstrate that our method can accommodate all traffic variation under a certain target link capacity.

Keywords

Model Predictive Control
Traffic Engineering
Traffic Prediction
Multi-path Routing

Contents

List of Figures

1 Introduction

In recent years, the time variation of the Internet traffic has become large due to the growth of Internet services such as streaming and clouds. A backbone network has to accommodate such traffic without congestion.

So far, backbone network has addressed this problem by preparing the redundant link capacity so as to accommodate not only the average traffic but also traffic surge [1, 2]. However, this approach requires higher cost according as the average and variance of traffic increase. Moreover, this approach causes waste of energy consumption due to the poor utility of network resources; this approach prepares more than double the capacity required to accommodate the actual traffic. Hence, a method to accommodate traffic without congestion on the network with limited resources is required to reduce such costs and power consumption caused by the over provisioning.

Many TE methods have addressed the problem of accommodating time-varying traffic by using the limited resources effectively [3–6]. In the TE methods, a control server periodically observes the traffic in a network and dynamically changes the routes so as to accommodate the observed traffic. These methods, however, set the routes only for the observed traffic. Therefore, the configured routes does not suit the actual traffic when significant traffic change occurs, but the routes are not changed until the next control cycle.

Distributed TE methods [3, 4] control the routes with a short interval by using only locally observed traffic information. However the frequent route change causes the routing oscillations. The routing oscillations degrade the throughput of TCP sessions; the routing oscillations cause the packet reordering by delivering the packets of one TCP session via different paths, which reduces the window size of the TCP session. The routing oscillation also causes too frequent changes in RTT, which decrease the throughput of delay-based TCP [7]. Hence, the method to avoid congestions without significant route changes is required.

The TE with traffic prediction is one approach to solving such problem. In this method, the routes are calculated based on the predicted future traffic. The prediction methods of the network traffic have been studied for various time scales, the short-term variation such as milliseconds or seconds order [8–11], the daily variation [12, 13], and the long-term variation such as monthly or yearly variation [14, 15]. The traffic prediction for the TE, which considers both of the daily variation and short-term variation, has also been proposed [16]. However, any prediction method makes an error. If the routes are calculated with such incorrect traffic information, the routes are no longer appropriate for the actual traffic, and congestion may occur. Therefore, the TE with traffic prediction should be robust to such prediction errors.

In this thesis, we propose a TE method which uses the traffic prediction without impact of prediction errors. In our method, we apply the Model Predictive Control (MPC) [17] to the TE, which has been recently studied as a method of system control based on the prediction of the dynamics of the system. In the MPC, a controller inputs the parameters to the system so as to hold the output of the system close to a target value. The MPC controller predicts the output of the system, which reflects the changes in the input values, and calculates the optimal input values for future time slots. The input values only for the next time slot is implemented. Then, the MPC controller observes the output and corrects the prediction using the output value as a feedback. After the correction of the prediction, the MPC controller recalculates the input value for the next time slot with corrected prediction. By continuing the above steps, the MPC controller can calculate the accurate input for the future time slots even if the prediction errors occur. Moreover, the MPC controller avoids the overreaction to the temporal prediction error by avoiding the drastic changes in the input value. In this thesis, applying the MPC to the TE, we propose the TE method which follows the predicted traffic variation and is robust to the prediction errors.

The rest of this thesis is organized as follows. Section 2 surveys the TE and traffic prediction. Section 3 introduces the TE method using the predicted traffic. Section 4 describes our TE method to which we apply the MPC. Section 5 presents an evaluation of our TE method. Section 6 mentions the conclusion.

2 Related Work

2.1 Traffic Engineering

The traffic engineering (TE) have been studied as an approach to accommodating changing traffic by dynamically changing routes. The process of TE is composed of following three steps; (1) the traffic rates are observed at the network devices, (2) the routes are calculated so as to accommodate the current traffic, and (3) the calculated routes are implemented to the actual network. These steps are periodically repeated to follow the traffic changes. The details of the above steps are discussed below.

The traffic rates are observed in fixed intervals (e.g. one second, one minute, or one hour) called *time slot*. Generally, the traffic rates of aggregated flows are observed instead of observing the traffic rate of each flow, because there are a huge number of flows; In [3, 5], a number of flows are aggregated as Origin-Destination (OD) flow which traverses from the ingress Point-of-Presence (PoP) router to the egress PoP router. Similar to these existing work, we also aggregate the flows as OD flows in this thesis. Hereafter, we denote the traffic rate of OD flow i at the k-th time slot as $x_i(k)$ and the vector $\boldsymbol{x}(k) = {}^t(x_1(k), \cdots, x_q(k))$ represents the traffic rates of all OD flows at the k-th time slot where q is the number of OD flows. The traffic rates of the OD flows are monitored by the routers or traffic monitors attached to the routers. Then, the information can be collected using the Netflow protocol and so on.

After the traffic information is collected, the routes are calculated based on the observed traffic rates. The routes are defined by the fraction of traffic of each OD flow sent to each path. We denote the fraction as a matrix $R(k)$ whose (i, j)-element $R_{i,j}(k)$ indicates the fraction of traffic on the OD flow j which traverses the available path i. Assuming the traffic pattern does not change between current and next time slots, the expected traffic rates on links are calculated as

$$\hat{\boldsymbol{y}}(t+1) = G \cdot R(t+1) \cdot \boldsymbol{x}(t) \tag{1}$$

where $\hat{\boldsymbol{y}}(t+1) = {}^t(\hat{y}_1(t+1), \ldots, \hat{y}_l(t+1))$ is the vector whose component $\hat{y}_i(t+1)$ indicates the expected traffic rate of link i at the next time slot, l is the number of links, and G is a matrix whose (i, j)-element $G_{i,j}$ is 1 if the available path j traverses the link i, otherwise $G_{i,j}$ is 0. The TE is a process to calculate routes $R(t+1)$ so as to minimize the *cost function* $f(\hat{\boldsymbol{y}}(t+1))$ of the traffic rates on the links such as congestion, delay, or packet loss rate. Therefore the TE is formalized as the following optimization problem.

$$minimize \quad : \quad f(\hat{\boldsymbol{y}}(t+1)) \tag{2}$$

$$subject\ to \quad : \quad \hat{\boldsymbol{y}}(t+1) = G \cdot R(t+1) \cdot \boldsymbol{x}(t) \tag{3}$$

The most used cost function is the maximum link utilization [3,5] to accommodate the unexpected traffic surge.

Finally the calculated routes are implemented. One of the approach to implementing the routes is setting the MPLS Label Switched Paths (LSPs) between the OD pair along the calculated routes [4, 18]. In this approach, the control server calculate the set of the links used by each LSP and split ratio of OD flow among LSPs based on $R(t+1)$. Then, the calculated routes are implemented by establishing the LSPs.

In the existing TE method, the control server calculates the next routes $R(t+1)$ based on the latest observed traffic rates $\boldsymbol{x}(t)$. These routes $R(t+1)$, however, are not exactly suitable to the actual traffic rates at the time slot $t+1$ because the traffic rates of the time slot $t+1$ differs from the that of time slot t. Under the drastically changing traffic, the difference between the $\boldsymbol{x}(t+1)$ and $\boldsymbol{x}(t)$ becomes large and the calculated routes based on $\boldsymbol{x}(t)$ may no longer accommodate the actual traffic at the $(t+1)$-th time slot. To quickly respond to such traffic, distributed TE methods [3, 4] are proposed. In such methods, the routes are frequently calculated using only the local traffic information. However, the frequent and significant route change causes the degradation of the throughput of TCP sessions because of the packet reordering or the frequent changes in RTT. Therefore, we propose a TE method with traffic prediction which directly sets routes fitting to the traffic at the future time slots without the significant route changes.

2.2 Traffic Prediction

The predictability of the Internet traffic has been received a significant interest from many domain such as capacity planning, anomaly detection, admission control, and traffic engineering. Traffic prediction is a process that analyze the dynamics of the observed traffic and predicts the volume of traffic to arrive in the future. First, a model of traffic dynamics is constructed from the observed traffic rates. The model represents the time evolution such as $x(k+1) = F(x(1), \cdots, x(k))$ where F is a model of traffic dynamics. Then, the future traffic rates are predicted in accordance with the model. If we observe the traffic rate until the time slot t, the traffic rate at the time slot

$t+1$ is calculated as

$$\hat{x}(t+1) = F(x(1), \cdots, x(t)) \tag{4}$$

where $\hat{x}(t+1)$ is the predicted traffic at the time slot $t+1$. The traffic rates after the time slot $t+2$ is iteratively calculated by using the former predicted values instead of the observation values as $\hat{x}(t+k) = F(x(1), \cdots, x(t), \hat{x}(t+1), \cdots, \hat{x}(t+k-1))$.

To predict the network traffic, many prediction models have been proposed such as ARMA, ARIMA [8, 14], ARCH [11], GARCH [9], Neural Network [12, 13] and so on. Using these prediction models, traffic prediction has been studied targeting various time scales; small time scale such as milliseconds, seconds or minutes order [8–11], daily scale [12, 13], and larger time scale such as monthly or yearly variation [14, 15].

Among the various time scales, the daily traffic variation is important for TE, which changes the routes so as to follow hourly traffic change in a day. It is well known that network traffic has periodical pattern with 24-hour cycle, and the daily traffic variation is estimated by considering the periodicity. In [16], a traffic prediction method for the TE was proposed. This method aimed at avoiding the underprediction, which causes the lack of resources and congestion when the predicted traffic rates are used as an input of the TE. In this method, the observed traffic variation is separated into daily variation and short-term variation. The future daily variation is predicted from the observed daily variation. Additionally, the range of the short-term variation is calculated from the observed short-term variation. Then, by summing the predicted daily variation and the range of the short-term variation, the predicted traffic rates are obtained.

The prediction targeting the smaller time scale is also useful for the TE which tries to follow the fine-grained traffic change. Zhou *et al.* proposed a prediction method that can predict the traffic rate of the next time slot accurately, even when the length of each slot is from 100ms to 10s [9]. Balaji *et al.* proposed a one-step ahead prediction targeting the minutes order interval such as fifteen minutes [11] for being applied to dynamic bandwidth provisioning.

All prediction methods including mentioned above, however, cannot avoid prediction errors. For example, the predicted value cannot follow the sudden traffic change which has unknown variation pattern. Additionally, the prediction error becomes large as the target of prediction is far ahead. Therefore, the TE with traffic prediction should be able to absorb the influence of prediction error.

3 Traffic Engineering with Traffic Prediction

The traffic prediction is useful for the TE to solve the delay of route change due to the gap between the actual traffic and observed traffic. Figure 1 shows an overview of the TE with traffic prediction. Opposed to the existing observation-based TE, the observed traffic rates is not directly used to calculate the routes. Using the observed traffic, the future traffic rates are calculated by the traffic prediction process. After the traffic prediction, the routes are calculated based on the results of prediction instead of the observed traffic. This process is periodically repeated to follow the change in tendency of traffic.

Using the traffic prediction, the traffic rates on the links can also be predicted; the predicted traffic rates on links in the case of routes $R(t+1)$ is calculated as

$$\hat{\boldsymbol{y}}(t+1) = R(t+1)\hat{\boldsymbol{x}}(t+1). \tag{5}$$

In the TE with traffic prediction, the routes are calculated by considering the cost function of $\hat{\boldsymbol{y}}(t+1)$.

The TE with traffic prediction configures the routes so as to avoid future congestion without frequent route changes. One approach of the TE with traffic prediction is to configure the fixed routes R that minimizes the cost function during the future time slots from $t+1$ to $t+h$. The optimal fixed routes R is obtained by solving the following optimization problem.

$$minimize \quad : \quad f(\hat{\boldsymbol{y}}(t+1), \cdots, \hat{\boldsymbol{y}}(t+h)) \tag{6}$$

$$subject\ to \quad : \quad \hat{\boldsymbol{y}}(k) = R\hat{\boldsymbol{x}}(k), k = t+1, \cdots, t+h \tag{7}$$

The predicted traffic, however, includes the prediction errors. The prediction errors become large as the target time slot of prediction is far ahead from the lastly observed time slot. The routes calculated based on the predicted traffic may not be suitable to the actual traffic because of the prediction errors; congestion may occur if the actual traffic rates on the link is much larger than the predicted traffic rates. Thus, the TE method that is robust to prediction errors is required.

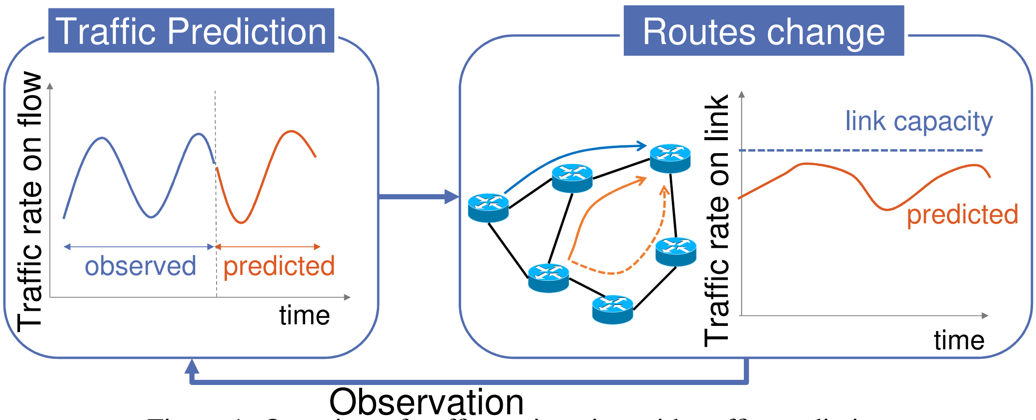


Figure 1: Overview of traffic engineering with traffic prediction

4 Traffic Engineering using Model Predictive Control

4.1 Model Predictive Control

The MPC is a method of system control based on the prediction of the dynamics of the system, which has been studied in recent years. In the MPC, a controller set the parameter (input) so as to keep the performance (output) of the system close to the target designed by an operator. Opposed to the traditional system control, the MPC controller predicts the changes in the output value to calculate the inputs for the future time slots $[t+1, t+h]$ called *predictive horizon* where h is the length of the predictive horizon. We describe the input and output at the k-th time slot as $u(k)$ and $y(k)$, respectively. The MPC controller calculates the inputs for the predictive horizon $[t+1, t+h]$ so as to keep $y(k)$ close to $r_y(k)$ which is the target value. The inputs $u(t+1), \cdots, u(t+h)$ that keep $y(k)$ close to $r_y(k)$ are obtained by using the objective function $J_1 = \sum_{k=t+1}^{t+h} \|y(k) - r_y(k)\|^2$ where $\|\cdot\|$ represents the Euclidean norm;

$$(u(t+1), \cdots, u(t+h)) = \mathop{\arg\min}_{(u(t+1),\cdots,u(t+h))} J_1. \tag{8}$$

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

$$z(k+1) = \phi(k, z(k), u(k)) \tag{9}$$

$$y(k) = \psi(k, z(k), u(k)) \tag{10}$$

where $z(k)$ is the state of the system at k-th time slot, ϕ and ψ are the function which maps the current state and input onto the next state and output, respectively.

Modeling the system by mathematical formula, however, may cause the modeling error such as the function ϕ or ψ is different from the actual system dynamics; if the output is predicted using the incorrect model, the result of prediction is different from the actual system output. This prediction error becomes large especially when the prediction target becomes far ahead from the last observation. Therefore, the MPC controller implement only the first one $u(t+1)$ of the calculated inputs $u(t+1), \cdots, u(t+h)$ for the predictive horizon. Then, the MPC controller observes the output and corrects the prediction using the output value as a feedback. After the correction of the prediction, the MPC controller recalculates the input value for the next time slot with corrected prediction.

In addition, the prediction errors may cause the significant change in the input value, which makes the system unstable. Hence, the controller restricts the amount of change in the input in order to absorb influence of the prediction error. We denote the amount of change in the input at the time slot k as $\Delta u(k) = u(k) - u(k-1)$ and the aggregated amount of change during the predictive horizon as $J_2 = \sum_{k=t+1}^{t+h} \|\Delta u(k)\|$. Instead of the input values determined by Eq.(8), the controller calculates the input values by the following optimization problem.

$$(u(t+1), \cdots, u(t+h)) = \underset{(u(t+1),\cdots,u(t+h))}{\arg\min} \quad J_1 + wJ_2 \tag{11}$$

where w is a parameter to balance the importance of the two objective functions J_1 and J_2.

4.2 Applying Model Predictive Control to Traffic Engineering

4.2.1 Traffic Engineering Model for Model Predictive Control

To achieve a prediction-based TE which is robust to prediction errors, we apply the MPC to TE. Figure 2 shows an overview of our TE method to which the MPC is applied. We assume that a control server collects all information of traffic and sets the routes. In the TE, the central control server plays a role as the MPC controller, which inputs the routes $R(k)$ and measures the outputs of the network, the traffic rates on the links $\boldsymbol{y}(k)$. The control server periodically changes the routes by repeating the following two steps. 1) the control server predicts the traffic rates of OD flows for the target time slots based on the previously observed traffic rates. 2) the control server calculates the routes based on the prediction so as to avoid congestion.

To avoid congestion, the central control server calculates the routes so as to hold the traffic rates on links under *target capacities* denoted by c_i. To achieve this, we introduce a cost function called *congestion level* of path j. The congestion level is determined by the amount of traffic which overshoots the target link capacities $\boldsymbol{c} = (c_1, \cdots, c_l)$. We assume that (1) the congestion on a link equally affects the paths which traverse the link, and (2) the congestion level on a path is determined by the bottle neck link which is the most congested intermediate link on the path. From the assumption (1), the overshooting traffic per path at link i is calculated by dividing the overshooting link traffic $[y_i(k)-c_i]^+$ by the number of traversing paths n_l. By the assumption (2), the congestion level of the path j, $\zeta'_j(k)$ is determined by the maximum overshooting link traffic over the path j. That is,

$$\zeta'_j(k) = \max_{i \in j}[y_i(k) - c_i]^+/n_i \tag{12}$$

where $[x]^+$ equals to x if the value of x is positive, otherwise $[x]^+$ equals to 0. We define the congestion level of the path j by scaling the value of $\zeta'_j(k)$ with the maximum link capacity as

$$\zeta_j(k) = \zeta'_j(k)/\max_l c_l. \tag{13}$$

4.2.2 Formulation of Optimization Problem

The control server computes the routes by considering the following two objective functions; $J_1 = \sum_{k=t+1}^{t+h} \|\boldsymbol{\zeta}(k)\|^2$ which indicates the summation of squares of the congestion level, and $J_2 = \sum_{k=t+1}^{t+h} \|\Delta R(k)\|^2$ which indicates the summation of squares of the amount of route changes. This multi-objective optimization is conducted by minimizing the weighted sum $(1-w)J_1 + wJ_2$ where $0 \le w \le 1$ indicates the importance of the restriction on the route changes.

In our TE method, the control server solves the following optimization problem at each time slot t:

$$\begin{aligned}
minimize \quad &: \quad \sum_{k=t+1}^{t+h} \left((1-w)\|\boldsymbol{\zeta}(k)\|^2 + w\|\Delta R(k)\|^2\right) && (14)\\
subject\ to \quad &: \quad \forall k, \forall p, n_l\zeta_p(k) = \max_{l \in p}\left[\hat{y}_l(k) - c_l\right]^+ / \max c_l && (15)\\
& \quad\quad \forall k, \hat{\boldsymbol{y}}(k) = G \cdot R(k) \cdot \hat{\boldsymbol{x}}(k) && (16)\\
& \quad\quad \forall k, \forall i, \forall j, R_{i,j}(k) \in [0,1] && (17)\\
& \quad\quad \forall k, \sum_{i \in \wp(j)} R_{i,j}(k) = 1 && (18)
\end{aligned}$$

where the $\boldsymbol{c}, \hat{\boldsymbol{x}}(k), G, n_l$ are given variables and $\boldsymbol{\zeta}(k), R(k), \hat{\boldsymbol{y}}(k)$ are variables to be optimized. The Eq.(15) corresponds to the definition of the congestion level $\boldsymbol{\zeta}(k)$. The Eq.(16) represents the relation between the traffic rates of the OD flows and links. The Eqs.(17) and (18) mean that all traffic on each OD flow is allocated to one of available paths.

Although all of the routes $R(t+1), \cdots, R(t+h)$ during the predictive horizon are obtained by solving the above optimization problem, the control server implements only the next routes $R(t+$

1). After the implementation of route change, the control server corrects the traffic prediction $\hat{\boldsymbol{x}}(k)$ using the newly observed traffic rate and recalculates the next routes by solving the optimization problem again.

Though the above optimization problem Eqs.(14)–(18) includes the non-linear constraint Eq.(15), it can be rewritten as a convex optimization problem introducing slack variables. The calculation of $[\hat{y}_l(k) - c_l]^+$ can be replaced by a linear constraint $[\hat{y}_l(k) - c_l]^+ = \hat{y}_l(k) - c_l + S_l(k)$ where $S_l(k) \geq 0$ is a slack variable. In addition, the operation $max_{l \in p}$ is translated by inequality constraints $n_l \zeta_p(k) \geq \max_{l \in p} [\hat{y}_l(k) - c_l]^+ / \max c_l$ for all the link l in the path p. As a result, the original optimization problem Eqs.(14)–(18) is rewritten as

$$
\begin{aligned}
minimize \quad &: \quad \sum_{k=t+1}^{t+h} (\|\boldsymbol{\zeta}(k)\| + w\|\Delta R(k)\|) && (19)\\
subject\ to \quad &: \quad \forall k, \forall p, \forall l \in p, n_l \zeta_p(k) \geq \alpha_l(k) / \max c_l && (20)\\
& \quad \forall k, \forall l, \alpha_l(k) = \hat{y}_l(k) - c_l(k) + S_l(k) && (21)\\
& \quad \forall k, \forall l, \alpha_l(k) \geq 0 && (22)\\
& \quad \forall k, \forall l, S_l(k) \geq 0 && (23)\\
& \quad \forall k, \hat{\boldsymbol{y}}(k) = G \cdot R(k) \cdot \hat{\boldsymbol{x}}(k) && (24)\\
& \quad \forall k, \forall i, \forall j, R_{i,j}(k) \in [0,1] && (25)\\
& \quad \forall k, \sum_{i \in \wp(j)} R_{i,j}(k) = 1 && (26)
\end{aligned}
$$

where $\alpha_l(k) \geq 0$ represents the value of $[\hat{y}_l(k) - c_l]^+$. The solution of this optimization problem satisfies the original constraint Eq.(15) because the variables satisfy the inequality formulation $n_l \zeta_p(k) \geq max_{l \in p} \alpha_l(k) / \max c_l \geq max_{l \in p} [\hat{y}_l(k) - c_l]^+ / \max c_l$ and the equality is attained if the $\zeta_p(k)$ is minimized.

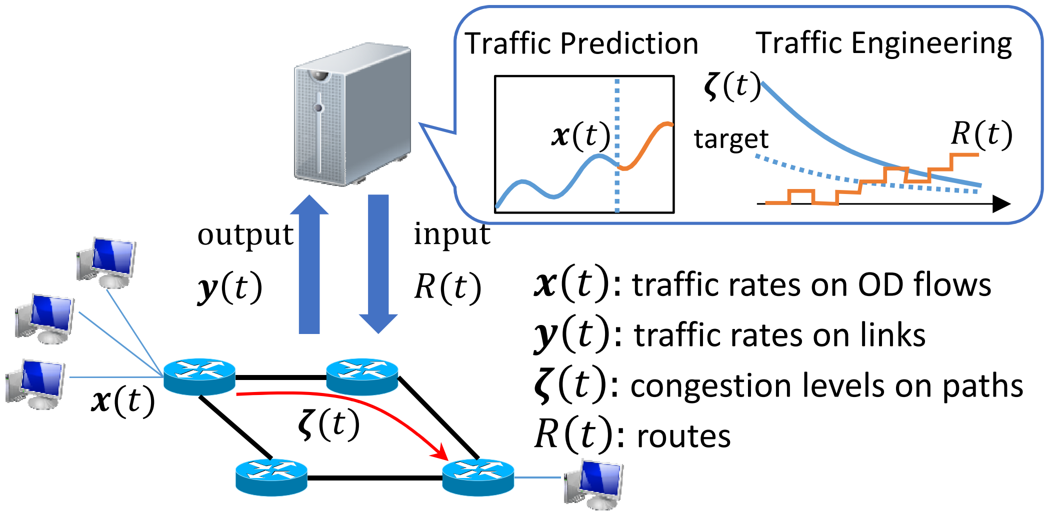


Figure 2: Overview of traffic engineering based on MPC

5 Evaluation

5.1 Evaluation of Basic Behavior of MPC-based TE

In this subsection, we investigate the behavior of the MPC-based TE under the basic situation.

5.1.1 Simulation Environment

Network Topology We use the simple network topology shown in Figure 3. In this simple network, there are only two OD flows from node 0 to node 1 and from node 4 to node 5. Each OD flow has two available paths shown by the arrows in Fig. 3, the paths 0-1 and 0-2-3-1 for the OD flow between node 1 and node 2 and the paths 4-5 and 4-2-3-5 for another OD flow. Due to the overlap of a link between paths 0-2-3-1 and 4-2-3-5 the control server has to adjust the split ratio of traffic among the paths. For example, if the traffic rates increase at the OD flow 0-1, more traffic should be bypassed on the path 0-2-3-1 and traffic at OD flow 4-5 should not traverse the path 4-2-3-5 so much to avoid the congestion.

Network Traffic We use artificial traffic shown in Figure 4. This artificial traffic includes traffic increase and decrease, which causes the congestion unless the routes are appropriately changed.

Prediction Method In this evaluation, we use a simple prediction method detailed as follows. First, we find a best-fit straight line $l_k = ak + b$ which minimizes the sum of squared distance from the previous observed traffic rates $x_{t-s}, x_{t-s+1}, \cdots, x_t (x \geq 1)$ denoted as $\sum_{k=0}^{s}(x_{t-s+k} - l_{t-s+k})^2$. Then, we obtain the future traffic rate as $\hat{x}_{t+k} = l_{t+k}$. Though there are many more sophisticated prediction methods, we use the above simple prediction with $s = 1$ to verify the effect of correcting the prediction by the feedback from new observation, which is one of the main effects of MPC.

Calculation of Routes To solve the optimization problem Eqs.(19)–(26), we use the CPLEX [19] which is a solver of optimization problems. The optimization problem is a convex quadratic programming problem which can be directly solved by using CPLEX. We run the CPLEX on a computing machinery with four Intel Xeon Processors each of which has 10 Cores, and 30MB Cache.

Compared Methods

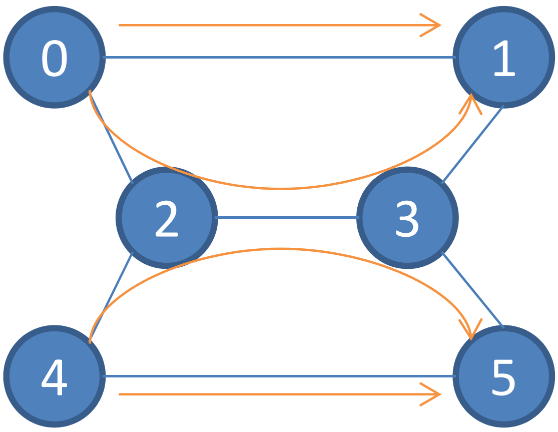


Figure 3: Simple network topology

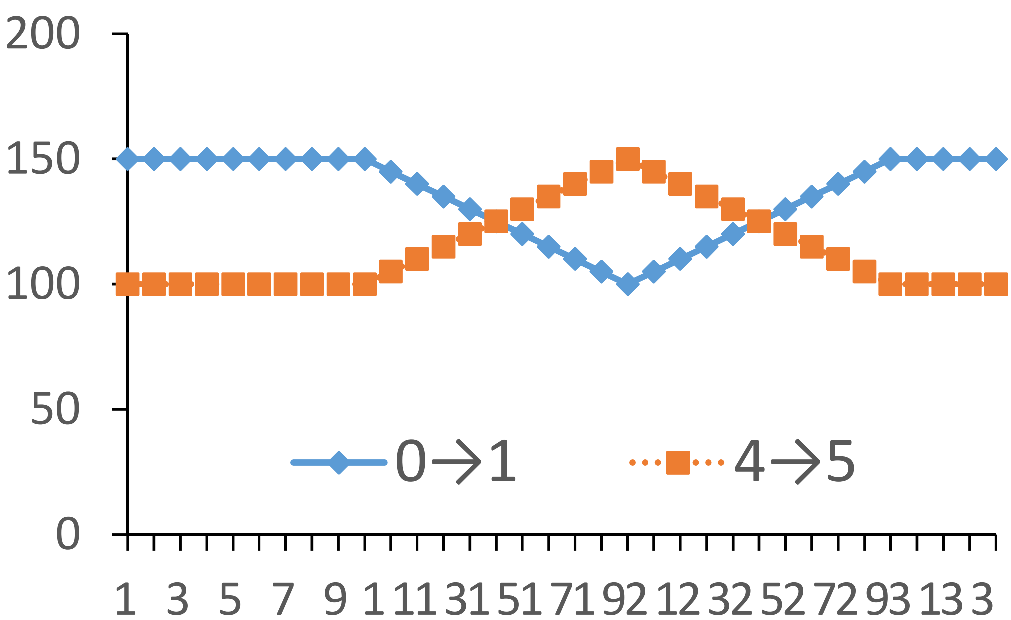


Figure 4: Network traffic for simple network topology

Observation-based TE In the observation-based TE, the control server only uses the observed traffic rates instead of the predicted traffic rates. By comparing the MPC-based TE with this observation-based TE, we demonstrate the effect of considering the future traffic variation.

Zero-Buffer-Path-Flow (ZBPF) Model Retvari and Nemeth also applied the MPC to TE based on Zero-Buffer-Path-Flow (ZBPF) model [20]. The ZBPF model, however, uses only the observed traffic rate, and it does not use the predicted rate. In the ZBPF model, they assume that no further traffic arrives within the predictive horizon. That is, the future traffic $\hat{x}_i(k)$ is regarded as zero. Hence, the dynamics of the amount of traffic to be delivered on a flow is described as follows

$$x_i(k) \quad = \quad x_i(t) - \tau \sum_{j=t}^{k-1} u_i(j) \tag{27}$$

where $u_i(j)$ is the amount of traffic rates to send on flow i at the j-th time slot.

The original TE method with ZBPF model described in [20] determines the traffic rates $u_i(j)$ to send at each time slot so as to complete the transmission of traffic within the predictive horizon. For comparison with our TE method described in Section 4, we implement the TE with ZBPF model as adjusting the fraction of traffic $R(k)$ so as to minimize the congestion level $\boldsymbol{\zeta}(k)$.

5.1.2 Congestion Level

Figure 5 shows the sum of $\zeta_i'(k)$ for all paths which are the amounts of traffic exceeding the target link capacity at each time slot. The label "MPC" represents the result of MPC-based TE with length of predictive horizon set to 3. We use the label "prediction base" to represent the result of MPC-based TE with $w = 0$, which performs as the simple prediction-based TE where the routes are calculated simply based on the predicted traffic rates without restricting the route changes. The label "observation base" and "ZBPF" means the result of the observation-based TE and the TE with ZBPF model. Although the ZBPF model also has a parameter h to determine the length of prediction horizon, we show only the result of $h = 1$ which was the best parameter for ZBPF model in the simulation. The ZBPF model with $h = 1$ is eventually same as the observation-based TE because the routes are calculated without considering the future traffic rates.

To clarify the effect of MPC, we compare the two cases of the weight for route change ($w = 0$ and $w = 0.5$). When the w is 0, the control server calculates the routes so as to simply minimize the congestion level for given traffic rate without restricting the amount of route change. Therefore, the routes may be wrong when the predicted traffic has prediction errors. On the other hand, when the w is 0.5, the control server determine the routes so as to minimize not only the congestion level but also the amount of route change. In this case, the control server can change the routes avoiding the effect of temporal prediction error.

In Fig.5(a), the congestion occurs at some time slots for all TE method when the w is 0. However, the reasons why the congestion occur are different between the prediction-based TE and the observation-based TE (or ZBPF). At time slots 11, 21 and 31, linear prediction makes an error because the increasing or decreasing slope of traffic rates is changed at those points. Due to these prediction errors, the prediction-based TE configures wrong routes and cause the congestion. On the other hand, the observation-based TE and ZBPF set wrong routes when the traffic rates increase or decrease because the routes based on previous traffic rates are no longer suitable to the next traffic pattern.

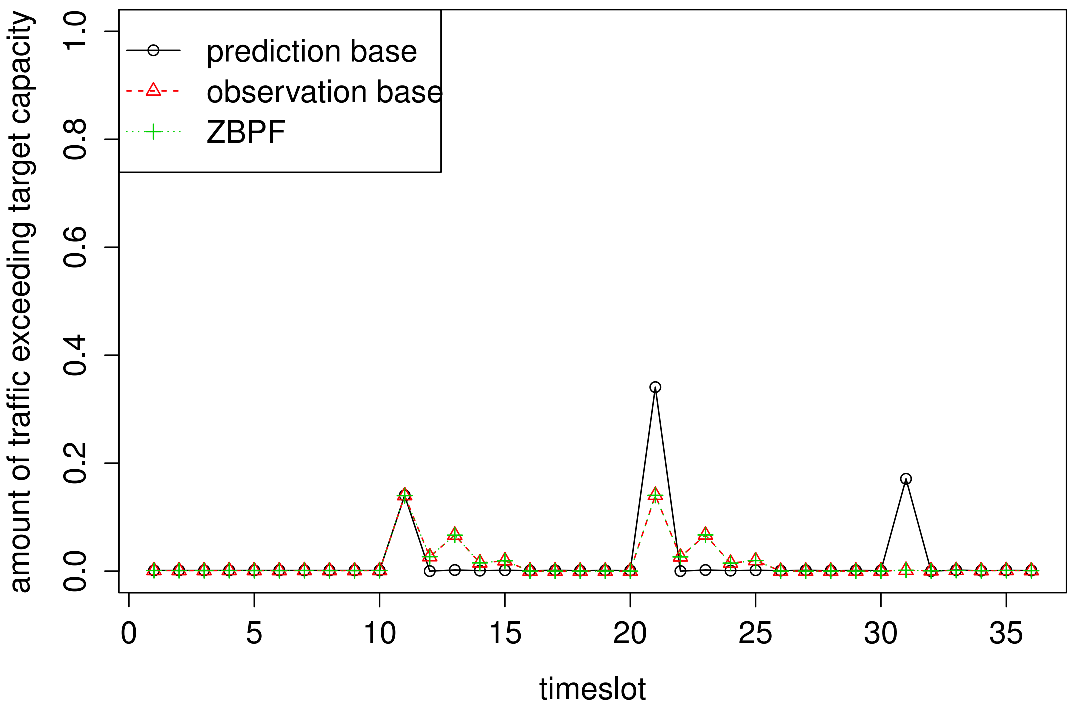
By restricting the amount of route change, as shown in Fig.5(b), the MPC-based TE avoids the congestion even when the prediction errors occur. This is because the MPC-based TE can absorb the impact of the prediction errors by avoiding the large route change caused by wrong traffic information. By contrast, the observation-based TE and ZBPF cause the heavier congestion than the case of $w = 0$ because the large w slows the response to the traffic changes.

The above results indicates that the idea of MPC, which controls the input based on prediction with absorbing the influence of prediction error, is effective for TE; the MPC-based TE avoids future congestions, while the simple prediction based TE or observation-based TE cannot avoid congestion due to prediction errors or traffic changes.

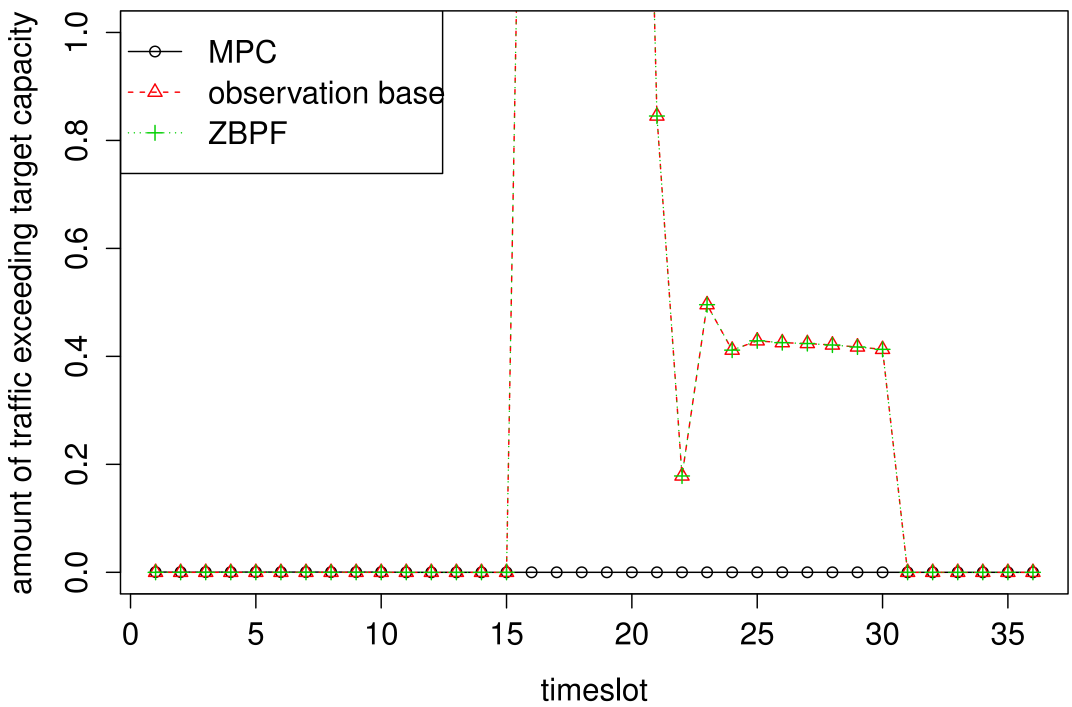
5.1.3 End-to-End Delay

By reducing the congestion level, the MPC-based TE provides lower-delay communication even when the traffic rates are changing. To verify this effect, we also evaluate the End-to-End delay when the MPC-based TE is conducted.

We calculates the link delay from the link utilization with approximating the packet processing in the Internet by M/M/1 queuing model. According to the queuing theory, the link delay is calculated as $\frac{\bar{L}}{C_l - y_l} + p_l$ where $\bar{L}$ is an average packet length, p_l is the propagation delay, and C_l is the actual capacity of the link l. The delay of OD flow is weighted sum of the delays of all



(a) without restricting the amount of route change $(w = 0)$



(b) with restricting the amount of route change $(w = 0.5)$

Figure 5: Amount of traffic exceeding the target link capacity in the case of simple network

available paths $\sum_p r_p d_p$ where r_p is the fraction ratio of traffic over path p and d_p is the delay of the path which is the summation of delays on all links on the path. A large delay is caused by not only the congestion but also the path length. Therefore, if most traffic traverse the long path, the delay of OD flow becomes large even at the low congestion level.

Figures 6 and 7 show the average delay and maximum delay of all OD flows, respectively. From these figures, the MPC-based TE reduces both average and maximum delay. This is because the MPC-based TE keeps lower congestion level and similar path length to the observation-based TE.

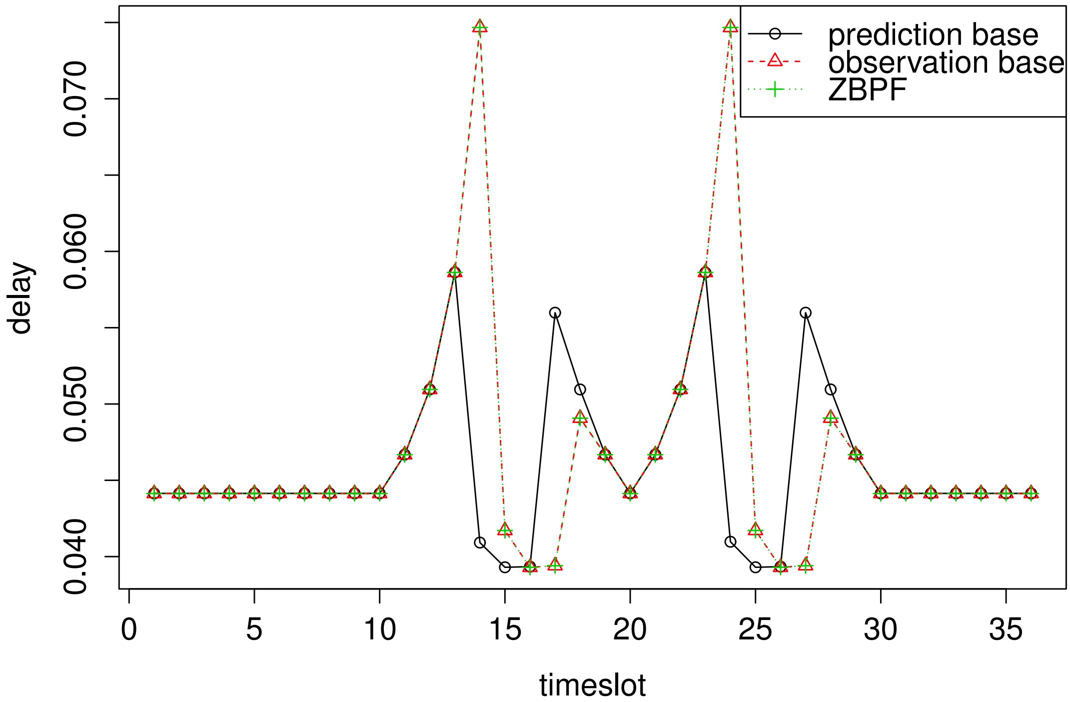
5.2 Evaluation of Congestion Level in Actual Network

From the above simulation result, we clarify that the MPC-based TE can reduce the congestion level and End-to-End delay for simple situation in which only one link is shared by two OD flows. In the actual network, however, the situation is more complex; some links are shared with some OD flows. To clarify that the MPC-based TE is also effective for actual network, we evaluate the performance on the topology of Internet2 using the actual traffic trace.

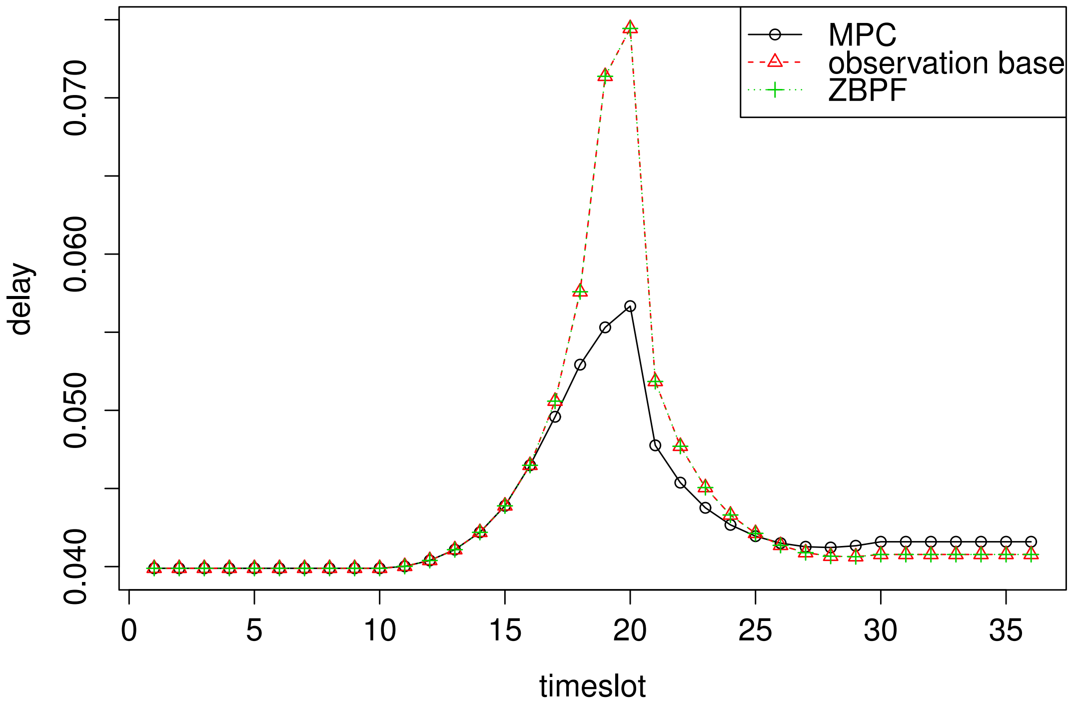
5.2.1 Simulation Environment

Network Topology In this subsection, we use an actual backbone network of Internet2 shown in Figure 8. The link capacities of Internet2 are over provisioned so that the maximum link utilization are lower than 20%. Hence, we set the target capacity of link to 15% of the actual link capacity in our simulation.

Network Traffic We use the actual traffic trace [21]. These traffic data are collected by Netflow protocol at each of the PoP routers. The sampling rate is one packet in every 100 packets, and aggregated data are exported every five minutes. Sampling method has mainly two problems that it causes sampling error and there may be unsampled flows. However, it is not critical problem for our evaluation because we only needs the traffic rate of aggregated OD flow, which has a large number of samples. We use four minutes' worth of data by avoiding the file boundary by excluding the start and end of thirty seconds of the Netflow data during 11/01/2011, 12:00 – 12:05 p.m.. The traffic data is aggregated into the OD flows between PoP routers using the BGP information. Using the start and end times and the total amount of traffic of each flow in the Netflow data, we

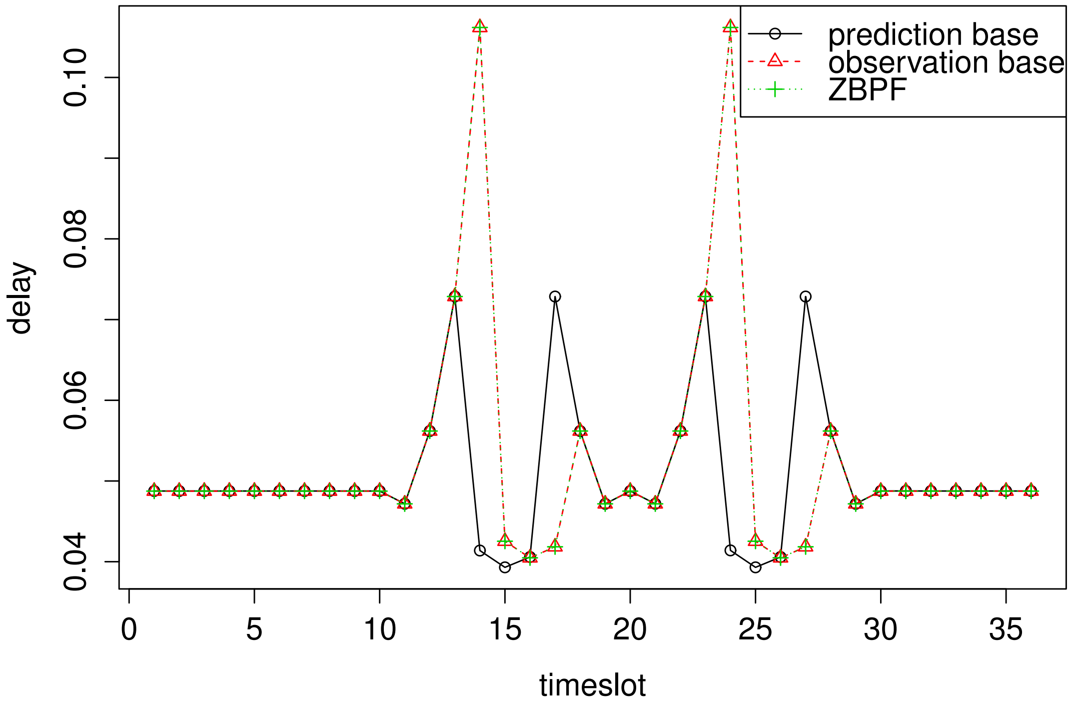


(a) average delay of all OD flows ($w = 0$)

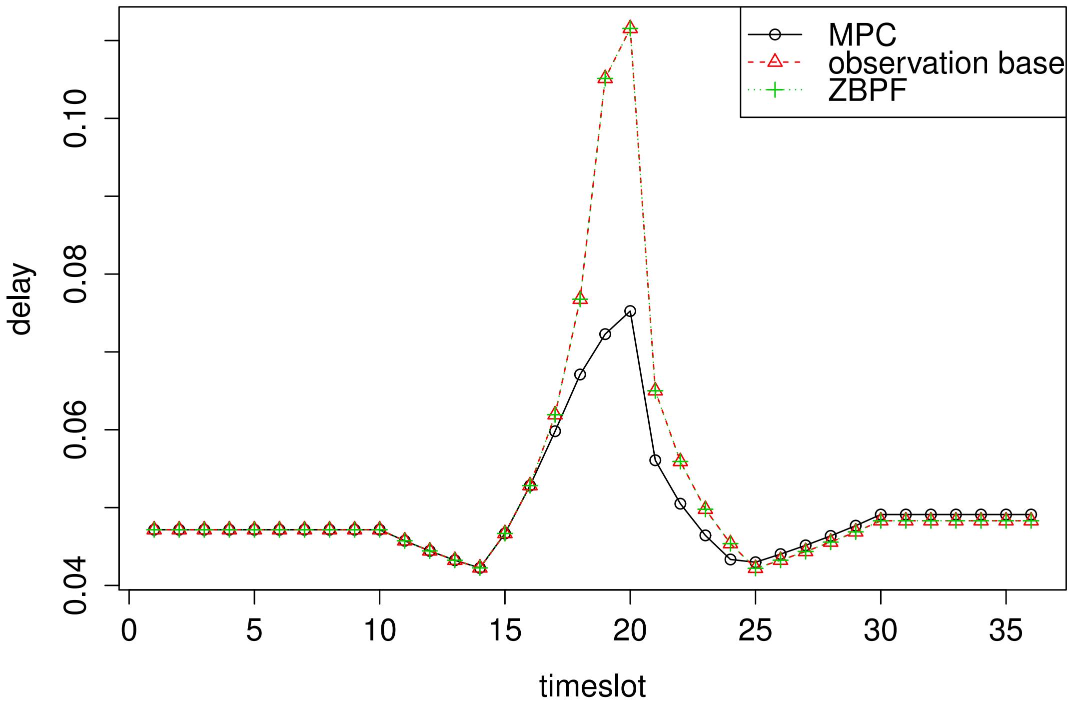


(b) average delay of all OD flows ($w = 0.5$)

Figure 6: Average End-to-End delay of all OD flows in the case of simple network



(a) maximum delay of all OD flows ($w = 0$)



(b) maximum delay of all OD flows ($w = 0.5$)

Figure 7: Maximum End-to-End delay of all OD flows in the case of simple network

obtain the traffic rate every second. The start and end times are recorded with the granularity of a millisecond. If the start and end times of a flow are t_s and t_e, the amount of traffic during a certain period τ is calculated as

$$x = \frac{\theta}{t_e - t_s}\tau \tag{28}$$

by assuming that traffic arrives at a constant bit rate, where θ is the total amount of traffic of the flow. The traffic amount at the time slot k corresponding to the actual time interval $[t_k, t_{k+1}]$ depends on the active time of the flow in the time slot, hence the τ is set to the active time as

$$\tau = \begin{cases} t_{k+1} - t_s & (t_k < t_s \wedge t_{k+1} < t_e) \\ t_e - t_s & (t_k < t_s \wedge t_{k+1} \geq t_e) \\ t_{k+1} - t_k & (t_k > t_s \wedge t_{k+1} < t_e) \\ t_e - t_k & (t_k > t_s \wedge t_{k+1} \geq t_e) \\ 0 & (otherwise). \end{cases} \tag{29}$$

Finally, the traffic rate of an OD flow is obtained by summing the traffic amount for all flows in the OD flow.

The calculated traffic rates are shown in Figure 9.

Prediction Method We use the same prediction method used in Section 5.1.

Calculation of Routes Similar to Section 5.1, we use the CPLEX [19] to calculate the routes. In this evaluation, the optimization is finished within one second when $h = 3$ in the case of Internet2.

Compared method In addition to the observation-based TE, we also compare the MPC-based TE with the following smoothed observation-based TE. The smoothed observation-based TE calculates the next routes $R(t+1)$ using the smoothed value $\bar{\boldsymbol{x}}(t)$ which reduces the noise of observation value $\boldsymbol{x}(t)$. We use an exponential moving average (EMA) for smoothing; if $\bar{x}_i(t-1)$ is a previous smoothed value of the flow i, and we observe the current traffic rate $x_i(t)$, then we update the smoothed value as $\bar{x}_i(t) = \eta x_i(t) + (1-\eta)\bar{x}_i(t-1)$ where η represents the degree of weighting decrease of historical data. By comparing the MPC-based TE with the smoothed observation-based TE, we demonstrate that the advantages of the MPC-based TE is not due to

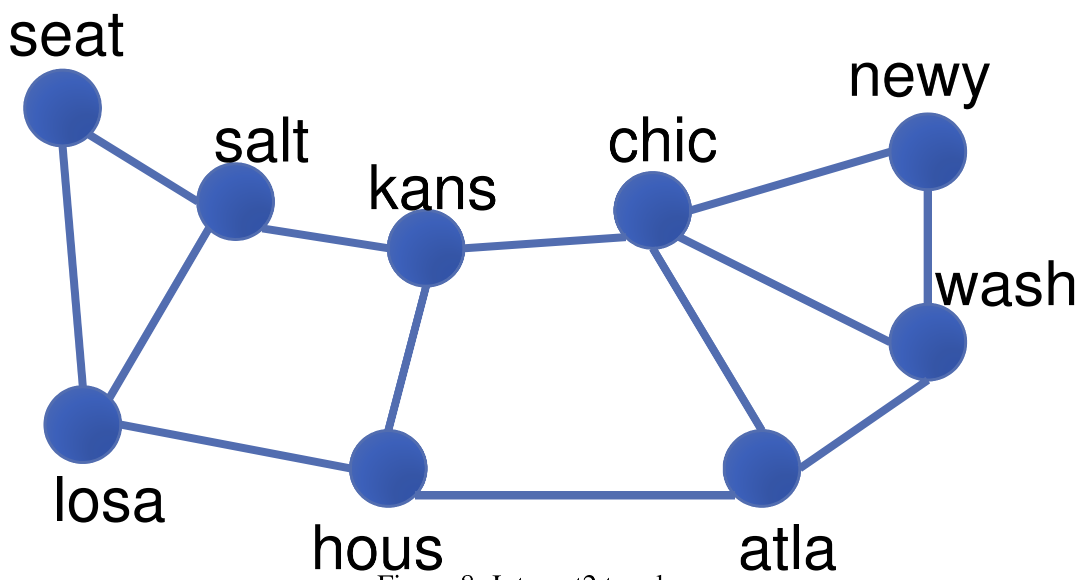


Figure 8: Internet2 topology

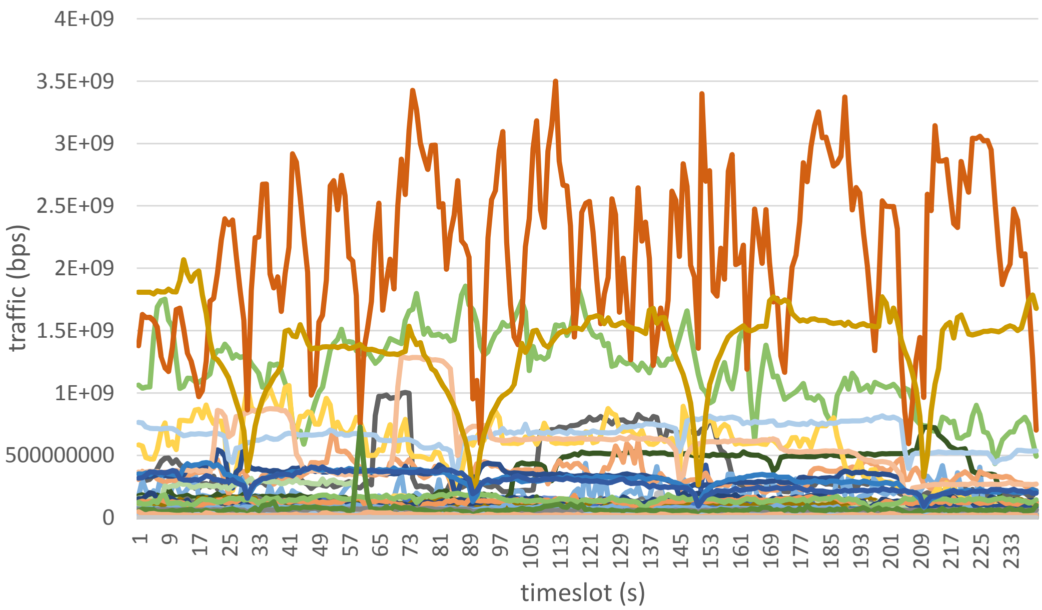


Figure 9: Network traffic in Ineternet2

smoothing the observed traffic rates though the traffic prediction obtains the average dynamics of traffic and eliminates the short-term variation of traffic.

5.2.2 Results

Figure 10 shows the amount of traffic exceeding the target link capacity when the MPC-based TE is conducted on the Internet2 topology with actual traffic trace. For comparison, we show the results of the observation-based TE and smoothed observation-based TE. The label "TE with smoothing" represents the result of smoothed observation-based TE.

In Fig.10, the same behavior of the MPC-based TE appears as the simple network. When the weight of route changes w equals to 0, not only the observation-based TE but also the prediction-based TE causes the congestion at some time slots. This is because the prediction errors sometimes occur in respond to the change in slope of traffic rates. When w equals to 0.5, the MPC-based TE keeps the traffic on the links under the given link capacities. Therefore, the MPC-based TE is also effective for actual network situation.

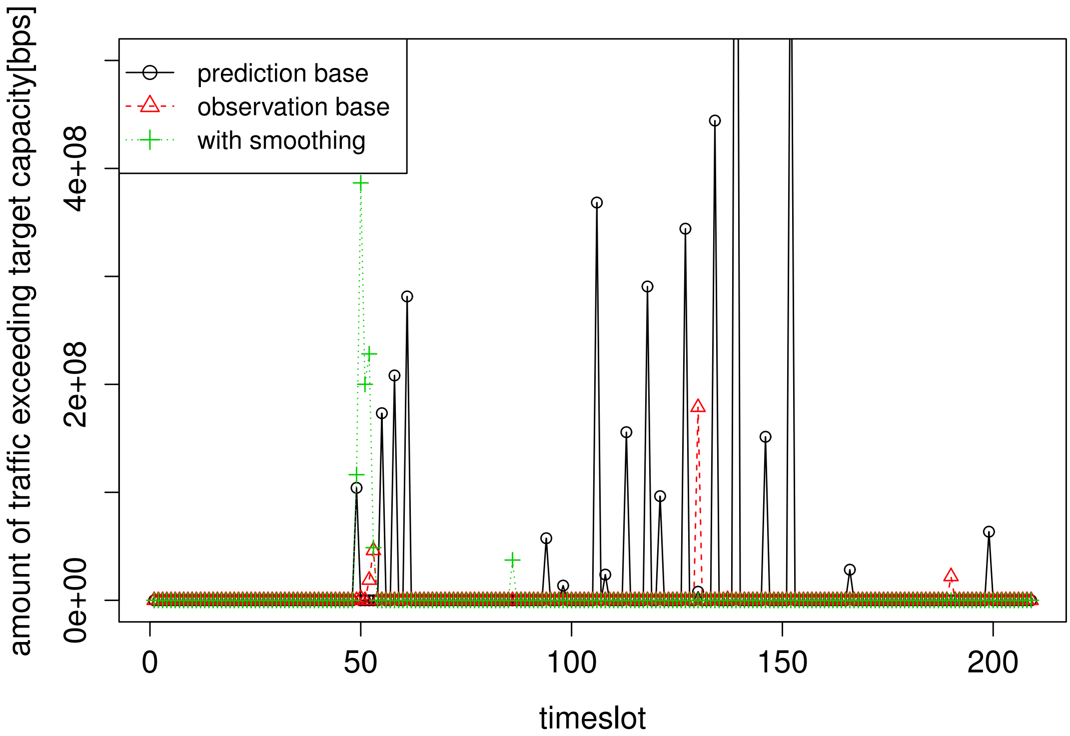
By comparing the result of MPC-based TE with smoothed observation-based TE, we can distinguish the effect of smoothing and prediction. From Fig. 10, the TE simply using the smoothing cannot avoid the congestion. This is because the smoothing amplifies the difference of traffic rates between current time slot and next time slot, which slows the response to the traffic change.

5.3 Discussion on Parameter Setting

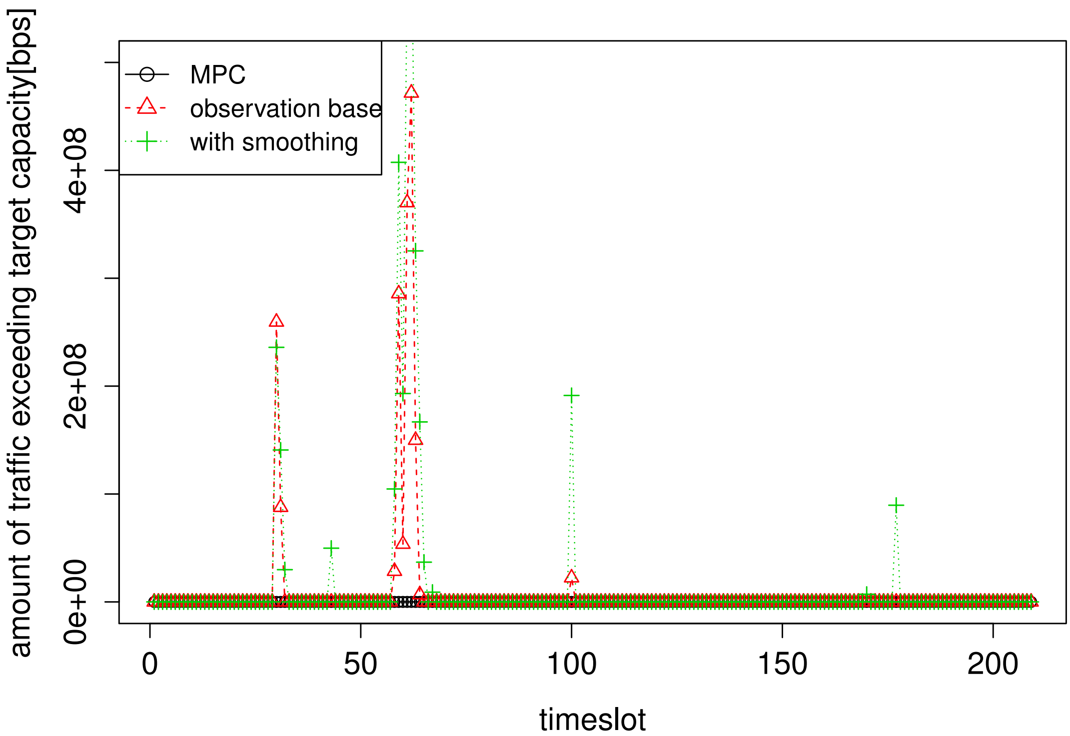
The MPC-based TE has some parameters such as weight for route change, length of predictive horizon, and cycle length of control and prediction. We investigate effect of these parameters in detail using the Internet2 topology with actual traffic trace.

5.3.1 Weight for Route Change

First, we examine the impact of w which is the weight of route change. In the above evaluation, we show that w have an important role in changing the routes with predicted traffic; the TE is sensitive to prediction error when $w = 0$ and robust to prediction error when $w = 0.5$. The value of w, however, represents the sensitivity to not only the prediction error but also the changing traffic. Hence, we may have to consider the trade-off between the robustness and sensitivity to set



(a) without restricting the amount of route change ($w = 0$)



(b) with restricting the amount of route change ($w = 0.5$)

Figure 10: Amount of traffic exceeding the target link capacity in the case of Internet2 with actual traffic trace

an appropriate value of w.

Figure11 shows the maximum amount of traffic exceeding the target link capacity for all time slots when the MPC-based TE is conducted with various values of w. The y-axis is the amount of exceeding traffic, and the x-axis is the value of w. The label h means that the MPC-based TE is conducted with the predictive horizon length of h.

In Fig.11, the medium value of w such as w=0.1–0.6 is appropriate for avoiding the congestion, which achieves to balance the robustness and sensitivity. In addition, the achieved performance of the MPC-based TE is not sensitive to w within the range of w=0.1–0.6.

5.3.2 Length of Predictive Horizon

Second, we investigate the impact of length of predictive horizon h. This parameter indicates how long future the control server considers to calculate the routes. Using the large value of h, the control server can take into account not only the next time slot but also further time slot to change the routes gradually in advance of traffic changes. However, setting too large h may cause wrong route changes because the prediction errors generally become large as the prediction target is far ahead. In addition, the larger h becomes, the longer time the calculation of routes takes.

Figure 12 shows the maximum amount of traffic exceeding the target link capacity when the MPC-based TE is conducted with various values of h, setting the value of w to 0.5. When the h is larger than 27, the congestion level increases as h becomes large. This is because the influence of prediction error becomes large as the predictive horizon becomes long. Too small values of $h = 1, 2$ also cause the congestion because the control server does not consider the traffic change in further future. The appropriate values of h to avoid the congestion are within the range of 3–26. Hence, it is sufficient for the MPC-based TE to set the h to 3 or a bit large values.

5.3.3 Cycle Length of Control and Prediction

Finally, we discuss the cycle length of control and prediction. In the above simulation, we set the control and prediction cycle length so that they equal observation cycle length (one second). However, the frequent control makes routes unstable, and it may degrade the throughput of the TCP sessions. Additionally, the frequent control imposes a limitation of calculation time on the control server. On the other hand, the control server cannot follow the traffic change, when the

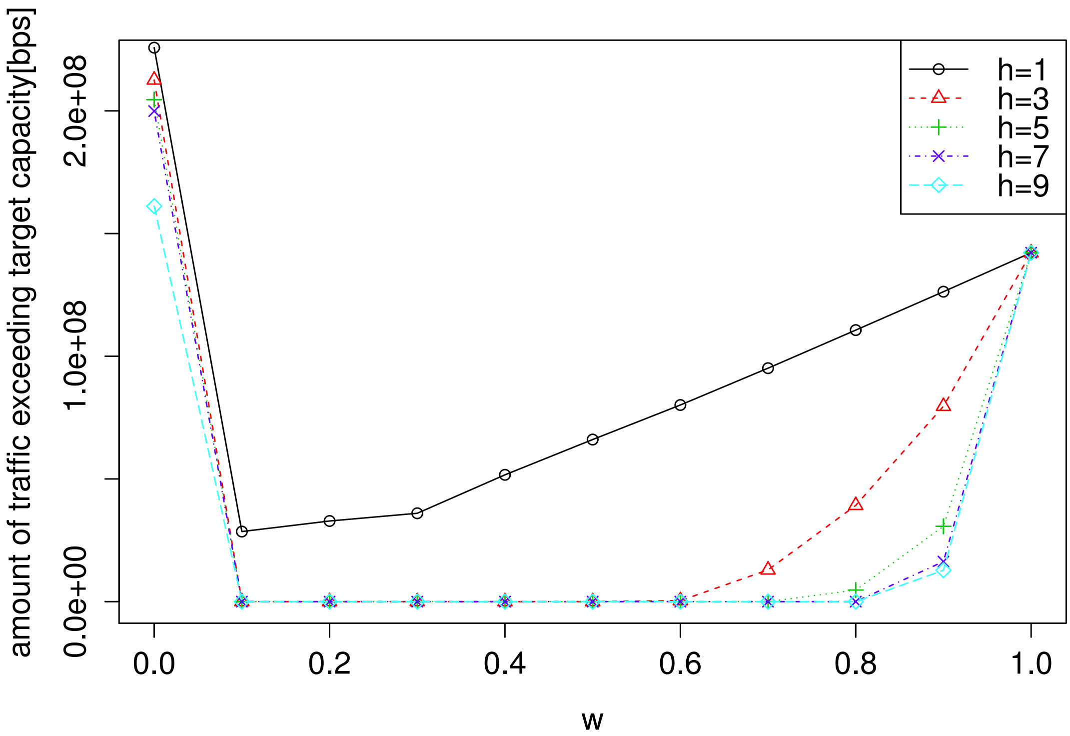


Figure 11: Maximum amount of traffic exceeding the target link capacity for all time slots when the MPC-based TE is conducted with various values of w

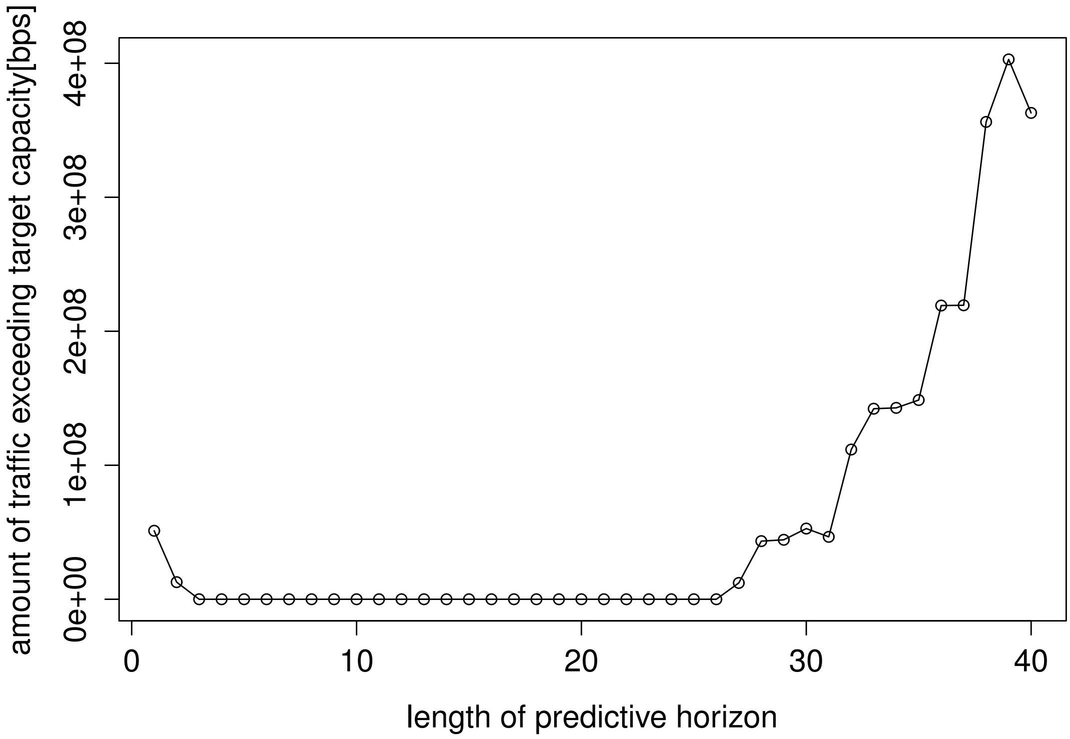
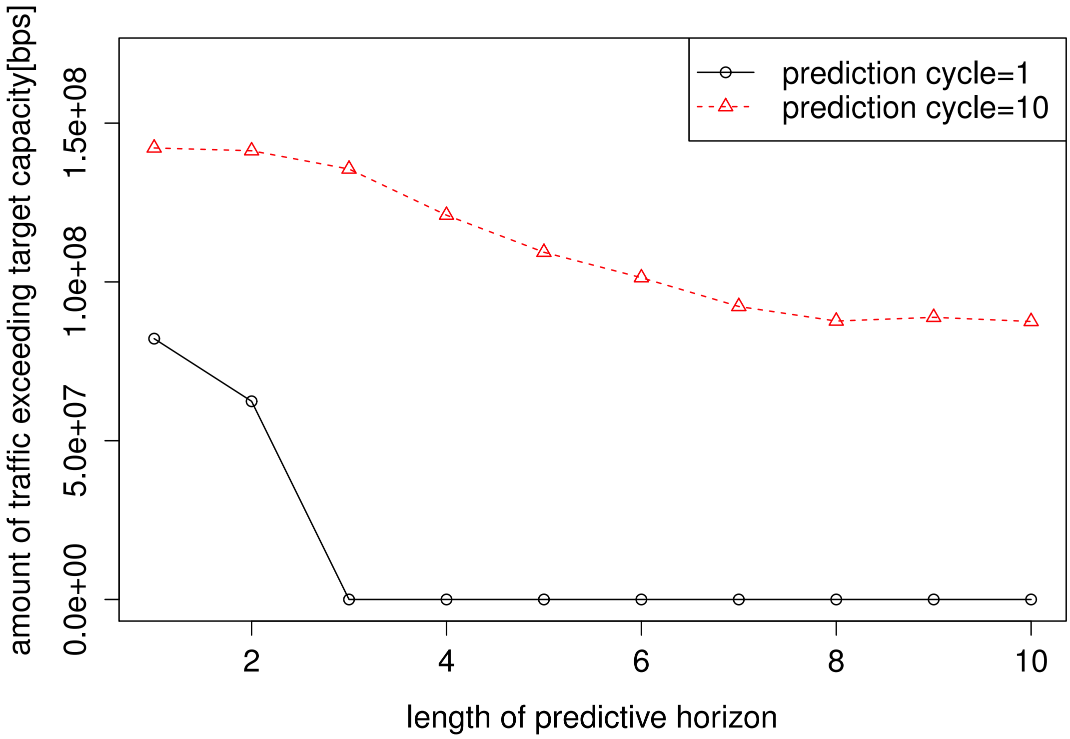


Figure 12: Maximum amount of traffic exceeding the target link capacity for all time slots when the MPC-based TE is conducted with various values of h ($w = 0.5$)

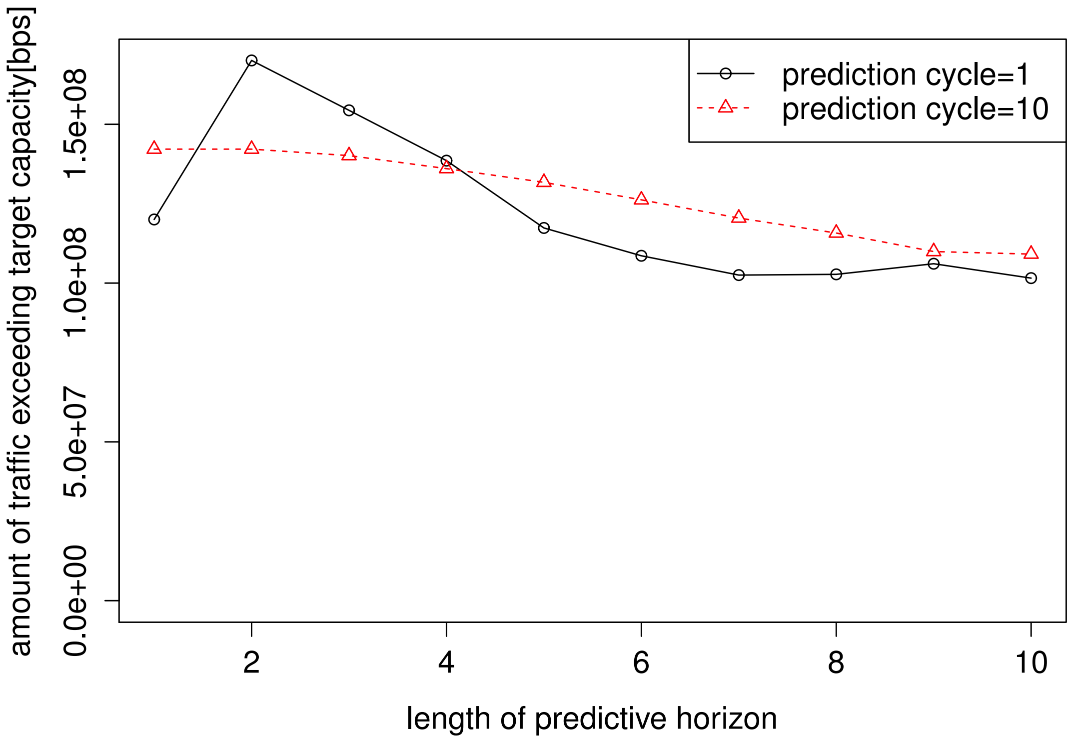
control and prediction cycle is large. Therefore, it is important to clarify which length of cycle is appropriate to avoid the congestion and a large calculation time.

Figure 13 shows the maximum amount of traffic exceeding the target link capacity for all time slots when the MPC-based TE is conducted with various lengths of control and prediction. We set the x-axis to the length of predictive horizon as similar to Fig.12 because the effect of predictive horizon will change with the change of cycle length, The label "prediction cycle = i" means that the prediction cycle length is set to i seconds. To change the cycle length, we change the length of the time slot of control and prediction cycle. If the control cycle is m seconds, the control server calculates the routes using the average rate of predicted traffic in each time slot, $\hat{x_i}'(k) = \frac{1}{m}\sum_{j=(k-1)m}^{km-1}\hat{x}_i(j)$. Similarly, traffic prediction is conducted with the aggregated traffic rates for the length of time slots. Though the period of control and prediction is changed, the time grain of traffic change is not changed. That is, traffic rates change in every one second.

From Fig.13, frequent control and prediction are better for avoiding the congestion. This is simply because the routes are quickly changed corresponding to the traffic change by the frequent control and prediction. However, there is a difference between the impact of control cycle and prediction cycle. In Fig.13(a), the congestion can be avoided even when the control cycle is 10 seconds. On the other hand, the congestion cannot be avoided when the prediction cycle is 10 seconds. This is because predicting with fine granularity can follow the changing traffic and the control server can accommodate traffic even with fixed routes considering the fluctuation of traffic. Therefore, we can set the length of control cycle to bit large while the prediction have to be frequently conducted.



(a) control cycle 10 seconds



(b) control cycle 60 seconds

Figure 13: Maximum amount of traffic exceeding the target link capacity for all time slots when the MPC-based TE is conducted with various lengths of control and prediction ($w = 0.5$)

6 Conclusion

In this thesis, we proposed a TE method which uses the predicted traffic rates instead of the observed value. According to the prediction-based control theory, our TE method calculates the routes with correcting the prediction and avoiding the large route change to absorb the impact of prediction errors. Through the simulation with the actual traffic trace of a backbone network, we demonstrated that our TE method can avoid the congestion while the observation-based TE cannot avoid the congestion. In addition, we discussed the parameter setting such as the weight for route change w, the length of predictive horizon h, and the cycle length of control and prediction. Then, we clarify the following characteristics about the parameter setting. First, the weight of route change has the role to absorb the effect of prediction errors by balancing the sensitivity and robustness to traffic change. We find that the performance is not sensitive to w in a certain range, and we can select a safe value of w from the range. Second, our TE method works well when h is 3 or bit more. Finally, changing routes in even 10 seconds intervals is sufficient to respond to the change in traffic rate at every one second while the prediction has to be conducted in one second.

Our future work includes the clarification of the robustness of the MPC-based TE through theoretical analyses of the MPC-based TE.

Acknowledgment

Foremost, I would like to express my deepest gratitude to Professor Masayuki Murata of Osaka University for his exact guidance, encouragement, and insightful comments. Furthermore, I would like show my sincere appreciation to Assistant Professor Yuichi Ohsita of Osaka University for continuous support, helpful discussions, and insightful advices.

My sincere appreciation also goes to Dr. Kohei Shiomoto, Dr. Keisuke Ishibashi, Dr. Noriaki Kamiyama, and Mr. Yousuke Takahashi of NTT Network Technology Laboratories for their helpful comments and fruitful discussions.

Moreover, I would like to show my appreciation to Assistant Professor Tomoaki Hashimoto of Osaka University and Associate Professor Kenji Kashima of Kyoto University for support in the theoretic aspect.

I am also grateful to the helpful advices from Professor Naoki Wakamiya, Associate Professor Shin'ichi Arakawa, and Assistant Professor Daichi Kominami of Osaka University.

Finally, I would like to thank all the members of the Advanced Network Architecture Research Group of Osaka University for their support, encouragement, and advices.

References

[1] C. Graleigh, F. Tobagi, and C. Diot, "Provisioning IP backbone networks to support latency sensitive traffic," in *Proceedings of IEEE INFOCOM 2003*, pp. 375–385, Apr. 2003.

[2] M. Roughan, "Robust network planning," *Guide to Reliable Internet Services and Applications Conputer Communications and Networks*, pp. 137–177, 2010.

[3] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: responsive yet stable traffic engineering," in *Proceedings of ACM SIGCOMM 2005*, pp. 253–264, Aug. 2005.

[4] E. Anwar, C. Jin, L. Steven, and W. Indra, "MATE: MPLS adaptive traffic engineering," in *Proceedings of IEEE INFOCOM 2001*, pp. 1300–1309, Apr. 2001.

[5] 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, (New York, NY, USA), pp. 99–110, ACM, Aug. 2006.

[6] 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, pp. 36–56, first quarter 2008.

[7] K. Srijith, L. Jacob, and A. Ananda, "TCP Vegas-A: Improving the performance of TCP Vegas," *Computer Communications*, vol. 28, pp. 429–440, Mar. 2005.

[8] M. F. Zhani, H. Elbiaze, and F. Kamoun, "Analysis and prediction of real network traffic," *Journal of Networks*, vol. 4, pp. 855–865, Nov. 2009.

[9] 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*, pp. 1–10, Sept. 2006.

[10] L. Xiang, "A new hybrid network traffic prediction method," in *Proceedings of IEEE GLOBECOM 2010*, pp. 1–5, Dec. 2010.

[11] B. Krithikaivasan, T. Zenf, K. Deka, and D. Medhi, “ARCH-based traffic forecasting and dynamic bandwidth provisioning for periodically measured nonstationary traffic,” *IEEE/ACM Transactions on Networking*, vol. 15, pp. 683–696, June 2007.

[12] 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, pp. 683–687, June 2004.

[13] 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*, pp. 1082–1088, Apr. 2012.

[14] 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, pp. 1178–1188, Mar. 2003.

[15] N. K. Groshwitz and G. C. Polyzos, “A time series model of long-term NSFNET backbone traffic,” in *Proceedings of IEEE ICC 1994*, pp. 1400–1404, May 1994.

[16] T. Otoshi, Y. Ohsita, M. Murata, Y. Takahashi, K. Ishibashi, and K. Shiomoto, “Traffic prediction for dyanamic traffic engineering considering traffic variation,” in *Proceedings of IEEE GLOBECOM 2013*, pp. 1592–1598, Dec. 2013.

[17] 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.

[18] D. O. Awduche and U. (mci Worldcom), “MPLS and traffic engineering in IP networks,” *IEEE Communications Magazine*, vol. 37, pp. 42–47, Dec. 1999.

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

[20] G. Retvari and G. Nemeth, “On optimal multipath rate-adaptive routing,” in *Proceedings of the Fifteenth IEEE Symposium on Computers and Communications*, pp. 605–610, June 2010.

[21] "Internet2 data." Available from http://internet2.edu/observatory/archive/data-collections.html.