Traffic Prediction for Dynamic Traffic Engineering

Tatsuya Otoshi, Yuichi Ohsita, Masayuki Murata,
Yousuke Takahashi, Keisuke Ishibashi, and Kohei Shiomoto

***Abstract*—Traffic engineering with traffic prediction is a promising approach to accommodate time-varying traffic without frequent route changes. In this approach, the routes are decided so as to avoid congestion on the basis of the predicted traffic. However, if the range of variation including temporal traffic changes within the next control interval is not appropriately decided, the route cannot accommodate the shorter-term variation and congestion still occurs. To solve this problem, we propose a prediction procedure to consider the short-term and longer-term future traffic demands. Our method predicts the longer-term traffic variation from the monitored traffic data. We then take account of the short-term traffic variation in order to accommodate prediction uncertainty incurred by temporal traffic changes and prediction errors. We use the standard deviation to estimate the range of short-term fluctuation. Through the simulation using actual traffic traces on a backbone network of Internet2, we show that traffic engineering using the traffic information predicted by our method can set up routes that accommodate traffic variation for several or more hours with efficient load balancing. As a result, we can reduce the required bandwidth by 18.9% using SARIMA with trend component compared with that of the existing traffic engineering methods.**

***Index Terms*—Traffic Engineering, Traffic Prediction, Data Mining, Trend Component, SARIMA Model, ARIMA Model**

I. INTRODUCTION

In recent years, time variation of Internet traffic has increased due to wide deployments of streaming and/or cloud services. Backbone networks are expected to accommodate such time-varying traffic without congestion. So far, backbone networks have addressed this problem by preparing redundant link capacity by considering not only average traffic but also traffic surges [1,2]. However, such an approach requires overly large capacity in accordance with the level of traffic change increases and causes low bandwidth utilization. For the last dozen years, the literature has reported that average link utilization of backbone networks has been very low, such as 17–29% in Google backbone [3], less than 50% in Sprint backbone [4], and 20% utilization is targeted in Internet2 [5]. This not only causes the waste of the bandwidth due to poor utilization of the network resource but also incurs unnecessary energy consumption. Henceforth, the traffic congestion must be avoided with limited resources, which will definitely reduce the over-provisioning cost and power consumption.

Adaptive traffic engineering is a promising approach for accommodating time-varying traffic by appropriately setting up the Origin–Destination (OD) routes [6–10]. In such traffic engineering methods, a control server periodically measures the traffic load in the network (typically every hour) and dynamically changes the routes so as to minimize the network congestion. However, traffic engineering using the measured traffic only mitigates the observed congestion and never avoids the future congestion. The currently congested links are resolved by changing routes at the next control epoch. By making the control interval shorter (say, in a unit of minutes), the control server may respond quickly to such traffic changes. However, it obviously causes the heavy load at the control server and affects the performance of the upper-layer protocol TCP due to frequent route changes. Such routing oscillation degrades the throughput of TCP sessions because of packet reordering and changes of RTT [11]. Our solution here is to execute traffic engineering by predicting the future traffic changes. That is, the control server should set up routes by considering the future traffic demands, not past ones. More exactly, the control server predicts the traffic variation in the next control cycle and then determines routes that can accommodate the predicted traffic without causing congestion in the next control cycle. For deciding the traffic variation, we again have the "time-scale" problem: if we want to have stable operation, we need set up a larger control cycle, but in that case, we cannot react to the temporal changes of traffic variation within the control interval. The shorter control cycle has exactly the same problem described above.

So far, various prediction methods have been studied on the basis of traffic predictive models such as ARMA, ARIMA [12, 13], ARCH [14], GARCH [15], and Neural Network [16–18]. However, to the best of our knowledge, existing prediction methods do not solve the above problem because they can predict the traffic variation accurately only for its target time scale. For example, the method proposed by Guang et al. [17] targets prediction in time scale of several hours. Therefore, it cannot obtain information about shorter-term variations because they are removed as noise before the prediction. On the other hand, a prediction method targeting a small time scale such as milliseconds or minutes [12, 14, 15, 19] is only effective for very near future prediction because of the significant degradation of prediction accuracy in the far future.

In this paper, we propose a traffic prediction procedure intended for application to traffic engineering by separating the short-term (non-periodical or temporal) and longer-term (hour or day) variations. We directly predict the longer-term variation as existing methods and estimate the short-term variation instead of predicting it. We then obtain the range of traffic variation including short-term variation during the next several hours, which is used as a basis for calculating the necessary capacities of each route in the next control interval. That is, our

Tatsuya Otoshi, Yuichi Ohsita and Masayuki Murata are with the Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565–0871, Japan(e-mail: t-otoshi, y-ohsita, murata@ist.osaka-u.ac.jp).

Yousuke Takahashi, Keisuke Ishibashi and Kohei Shiomoto are with NTT Network Technology Laboratories, NTT Corporation, Musashino-Shi, 180–8585, Japan(e-mail: takahashi.yousuke, ishibashi.keisuke, shiomoto.kohei@lab.ntt.co.jp).

key contribution here is that we investigate how to handle the prediction uncertainty in order to apply our method to traffic engineering. As described before, the prediction uncertainty stems from two factors (prediction error for periodical pattern and noisy short-term variation), and we take account of such prediction uncertainty in determining the necessary resources for each route. In this paper, we focus on the results of traffic engineering instead of the accuracy of prediction, because prediction methods with small error are not always suited to traffic engineering. Even when mean prediction error is low, congestion cannot be avoided by traffic engineering using the predicted traffic if the temporal increase of the traffic causing congestion is not predicted. On the other hand, a prediction method responsive to the traffic increase that may cause the congestion can avoid congestion even if the method's mean prediction error is large. Therefore, we evaluate our prediction procedure by investigating the influence of prediction method on traffic engineering performance.

In our earlier work [20], we only compared the effectiveness of traffic engineering using predicted traffic with observation-based traffic engineering. This paper also investigates details of the impact of traffic prediction on traffic engineering. We first investigate the impact of two parameters in our prediction procedure having a large impact on traffic engineering, the confidence level of prediction errors and the confidence level of short-term variation. We find that the confidence level of the short-term variation should be set to a large value, while a small confidence level for prediction errors is generally sufficient.

We then investigate the impact of considering periodicity, and find that even prediction without considering periodicity is sufficient if the control period is a few hours, while traffic prediction considering periodicity improves the worst-link utilization achieved by traffic engineering if the control period is larger than 24 hours.

The rest of this paper is organized as follows. Section II surveys related work of traffic prediction and traffic engineering. Section III introduces the traffic engineering method using the predicted traffic. Section IV describes the prediction procedure. Section V presents an evaluation of our prediction procedure. Section VI mentions the conclusion and future work.

II. Related Work

A. *traffic engineering*

There is a large body of literature regarding TE [6–10]. The most of existing traffic engineering methods are observation-based approach in which the control server collects the current traffic information and then sets the routes so as to accommodate the observed traffic. However, such observation-based method may not be able to accommodate the future traffic because the traffic pattern will change from the observed pattern.

One approach to handling such uncertainty of the future traffic is to allocate sufficient resources to accommodate worst-case traffic patterns. For example, a static routing method called *oblivious routing* [21–23] sets a fixed route to accommodate worst-case traffic. Instead of observing current traffic, this method tries to accommodate all possible traffic patterns by minimizing the maximum link load. Wang *et al.* proposed a robust traffic engineering method by introducing the oblivious routing concept [6]. Their method considers the *convex hull* of a set of historical traffic patterns, namely the set of arbitrary weighted average of observed traffic. It handles uncertain future traffic dynamics by optimizing routes for this convex hull under constraints where the worst-case performance is not degraded. However, the approach requires large resources to accommodate worst-case traffic.

To accommodate the future traffic variation with a small resources, it is important to know the future traffic. Thus, our traffic engineering approach uses the prediction of time series of traffic to decide the routes.

B. *traffic prediction*

The predictability of Internet traffic has received significant interest in various domains, such as capacity planning, anomaly detection, admission control, and traffic engineering. The prediction methods of the network traffic have been studied for various time scales such as milliseconds, seconds or minutes order [12,14,15,19], daily [16–18], and even monthly variation [13].

The prediction based traffic engineering requires the traffic prediction for the control period. The control period may be a few hours or more. Thus, the traffic prediction should follow the daily variation. On the other hand, the traffic variation during the control period includes the temporal changes, which should also be considered by the traffic engineering so as to avoid the congestion.

Some of existing prediction methods focus on the daily traffic variation [16–18]. However, they exclude the short-term variation, which is also important for the traffic engineering. For instance, the method in [17] eliminates the values which is too far from average traffic value, and then removes the white noise from the data by Fourier analysis before inputting the data to the prediction process. If these eliminated data is not considered in traffic engineering, the calculated routes cannot accommodate the temporal traffic change and may cause the congestion.

One simple approach to consider these removed variation is to use the short-term prediction method [12, 14, 15, 18, 19]. However, the short-term prediction method causes a large prediction error when it is used to predict the traffic during the control period, which may be a few hours or more. To predict the daily variation with a small time granularity, a number of iterations of one-step ahead prediction is required, which causes inaccurate prediction for the distant future due to the accumulation of errors. For instance, in [18], the error of the iterative prediction with 5 minutes of granularity monotonically increases as the prediction target becomes long.

Therefore, in this paper, we clarify how to handle the long-term and short-term variation for the prediction based traffic engineering. In our approach, we decompose the traffic variation into long-term and short-term variation. Then, in addition to the prediction of the long-term variation, we also estimate the range of the short-term variation. Finally, we obtain the predicted upper bound of traffic variation by summing the predicted long-term variation and estimated range of the short-term variation.

In addition, we evaluate the prediction method combined with the traffic engineering. Though most of the existing work on the traffic prediction discuss their prediction accuracy by comparing the predicted values with the actual values. However, prediction errors of some flows may have only a small impact on the performance on the traffic engineering, while other flows may have a large impact; the large flows affect the link utilization significantly than the small flows and may be required to be predicted accurately. Therefore, in this paper, we discuss the suitable prediction method considering the results of the traffic engineering using the predicted traffic.

III. Traffic Engineering with Traffic Prediction

In this paper, we deploy a central control server that controls the network. The central control server observes and predicts the traffic rate and calculates routes on the basis of the predicted traffic.

The control server observes the traffic rate at each flow in fixed intervals (e.g. 10 minutes, 30 minutes, or one hour) called *time slots*. The observed traffic rates of all flows in the t-th time slot are represented as a vector. We denote this vector as $\boldsymbol{x}_t$. The aggregation of a number of flows is useful for reducing the observing cost and prediction time. In this paper, we aggregate the flow as OD flow that traverses from the ingress Point-of-Presence (PoP) router to the egress PoP router. This flow grain is sufficient to decide the routing in a backbone network, and the existing observation based traffic engineering methods often use OD flow [6, 7].

Using the observed traffic rates until the t-th time slot, the control server predicts the future traffic rates in the next f time slots. The prediction of future traffic is formulated as

$$\hat{\boldsymbol{x}}_{t+1..t+f} = F\left(\boldsymbol{x}_{t-h+1..t}\right), \tag{1}$$

where $\boldsymbol{x}_{a..b} = (\boldsymbol{x}_a, \boldsymbol{x}_{a+1}, \cdots, \boldsymbol{x}_b)$ is a matrix in which each column corresponds to each vector, $\hat{\boldsymbol{x}}_k$ is the predicted traffic in the k-th time slot, f is the number of time slots where the traffic rate is predicted, h is the length of observed time slots used in the prediction, and F is a prediction function defined by a prediction method.

In traffic engineering, the control server calculates the routes so as to avoid congestion for f time slots. We define these f time slots as the *control period*. In this paper, we consider the case in which the control period is 3–24 hours. The calculated routes are represented as a matrix A called *routing matrix*. The (i, j)-element $a_{i,j}$ in the routing matrix A represents the ratio of the traffic over the OD flow j mapped onto the link i. Corresponding to the routing matrix, the predicted traffic mapped onto each link in the control period is represented as

$$\hat{\boldsymbol{y}}_{t+1..t+f} = A\hat{\boldsymbol{x}}_{t+1..t+f}, \tag{2}$$

where $\hat{\boldsymbol{y}}_k$ is the vector indicating the predicted traffic on all links in the k-th time slot. Traffic engineering is the process to adjust A so as to control $\hat{\boldsymbol{y}}_{t+1..t+f}$ in some desirable way.

In traffic engineering, the most widely used metric of congestion is maximum link utilization [6, 7], i.e. the utilization of the most congested link. In this paper, we use a simple optimization approach that minimizes the maximum utilization among all links for all time slots within the control period, though there may be a more sophisticated approach using the predicted traffic. Using this simple approach, we can clarify the impact of the prediction on the traffic engineering performance by simply observing the achieved maximum link utilization. If the traffic engineering method using the traffic information predicted by a method keeps the small link utilization for a long time, the prediction method is suitable for the traffic engineering intended to stabilize traffic accommodation.

The optimization problem is formulated as the following linear programming problem:

$$minimize{:}U \tag{3}$$

$$subject\ to{:}\forall s, d, \sum_{p(l)=s} A^{s,d}(l) = 1 \tag{4}$$

$$\forall s, d, \sum_{f(l)=d} A^{s,d}(l) = 1 \tag{5}$$

$$\forall s, d, n, \sum_{p(l)=n} A^{s,d}(l) = \sum_{f(l)=n} A^{s,d}(l) \tag{6}$$

$$\forall l, k \in [t+1, t+f], \sum_{s,d} \frac{A^{s,d}(l)\hat{x}_k^{s,d}}{C(l)} < U, \tag{7}$$

where U is the maximum link utilization, $A^{s,d}(l)$ is the ratio of traffic from s to d routed over the link l, and $p(l)$ and $f(l)$ are the start and end nodes of the link l, respectively, $\hat{x}_k^{s,d}$ is the predicted traffic rate of the flow from s to d at the k-th time slot and $C(l)$ is the capacity of the link l. $\hat{x}_k^{s,d}$ and $C(l)$ are given in this problem, and $A^{s,d}(l)$ and U are the variables to be obtained. Eqs. (4–6) are the constraints for flow conservation. Eq. (7) ensures that U is the maximum link utilization of all the links for all the time slots within the given control period. By solving the above problem, we obtain routing matrix A, which is used for the control period $[t+1, t+f]$, and is not changed before $t+f+1$. Setting f to a large value avoids frequent route changes, but to do so the traffic prediction should be response to traffic variation occurring in the control period $[t+1, t+f]$; if the predicted traffic cannot respond to the temporal traffic variation that occurs in some time slots, congestion may occur. In Section IV we therefore discuss a traffic prediction procedure that considers traffic variation in each time slot.

To map the routing matrix A to actual network, we assume the paths between an OD pair of routers are determined by MPLS Label Switched Paths (LSPs). According to the link-based routing determined by A, the control server can calculate the path-based routing, i.e. defining the link set used by each LSP and split ratio among the LSPs. Each PoP router splits traffic among the LSPs corresponding to an OD flow using the hashing method described by Anwar et al. [8]. In this method, each fine-grain flow (e.g TCP flow) is routed to only one LSP to avoid the packet reordering that degrades TCP throughput.

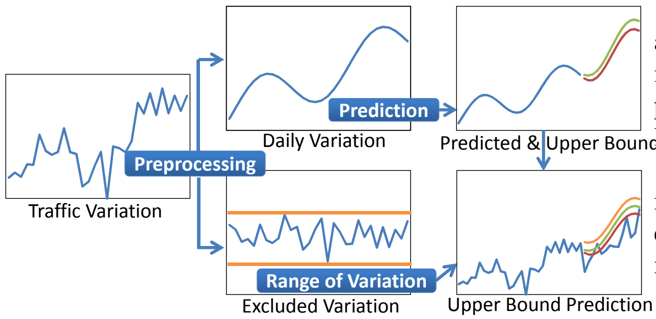


Fig. 1. Prediction process

IV. Traffic Prediction Process

A. Overview

In the network, traffic variation has a daily pattern in longer-term (hour or day) variation, and the traffic changes every few hours. The traffic prediction needs to follow longer-term variation so that the traffic engineering calculates the routes suitable for the next few hours. However, the actual traffic variation includes noisy variation (short-term variation), and the longer-term tendency is polluted. Such polluted data cause a large prediction error. Therefore, we use preprocessing that extracts the daily periodical variation excluding the noisy variation to improve the prediction accuracy.

On the other hand, the short-term traffic variation excluded by the preprocessing may cause the congestion. The short-term traffic variation is hard to predict, but it can be considered as a noisy fluctuation whose mean and variance are stable if the preprocessing extracts the longer-term traffic variation accurately. Thus, we consider the short-term traffic variation by calculating the variance of the traffic variation excluded by the preprocessing. Then, by adding the confidence interval of the calculated variance to the predicted longer-term traffic variation, we avoid the underprediction caused by the short-term traffic variation.

Moreover, we also consider the confidence interval of the prediction error to avoid the impact of the prediction error on the traffic engineering. The confidence interval causes the overprediction. However, the overprediction has a smaller impact than the underprediction. This is because the underprediction causes the lack of allocated resources and congestion while the overprediction does not affect the communication performance until the overpredicted flow blocks resources to be allocated to other flows.

Our approach is summarized in Fig. 1. First, we extract the longer-term variation from the actual traffic variation by the preprocessing. Second, we predict the future traffic variation using the extracted variation and estimate the variance of excluded variation. Finally, we obtain the upper bound of traffic variation summing up the predicted upper bound of longer-term variation and the confidence interval of the excluded variation. The obtained upper bound is used as input of the traffic engineering.

B. Prediction Preprocessing

In the preprocessing, we extract the daily periodical variation from the observed traffic. The object of preprocessing is to filter out the short-term traffic variation that is hard to predict. This increases the accuracy of the prediction of the longer-term traffic variation.

In this paper, we investigate the following preprocessing methods: the lowpass filter, the trend component, and the envelope. The rest of this subsection details the preprocessing methods.

1) Lowpass Filter: One approach to extract the longer-term variation of the traffic variation is to use the lowpass filter, which extracts the longer-term variation using the Fourier transform.

By using the Fourier transform, the time series of the traffic data can be represented as

$$x_k = \sum_{n=0}^{h-1} f_n \exp\left(2\pi i \frac{nk}{h}\right), \tag{8}$$

where f_n is Fourier coefficient corresponding the frequency n/h and i is the imaginary unit. Eq. (8) also includes high frequency variations such as noise. To reduce these noisy variations, the lowpass filter removes the terms with large n and extracts the longer-term variation as

$$l_k = \sum_{n=0}^{L} f_n \exp\left(2\pi i \frac{nk}{h}\right), \tag{9}$$

where L is the threshold to remove the high frequency variations. In this paper, we set L so as to remove the variation of frequency higher than the daily variation, i.e. the lowpass filter extracts the daily pattern of traffic variation.

2) Trend Component: In the second approach, we extract the longer-term variation by using a time series model. One approach to model the longer-term variation is the trend model [24]. We call the traffic variation extracted by using the trend model *trend component*. The trend component includes the daily traffic variation and longer-term traffic variation. The trend model is denoted as

$$x_k = t_k + \epsilon_k \tag{10}$$

$$\Delta t_k = \Delta t_{k-1} + w_k, \tag{11}$$

where x_k is the traffic rate of a flow in the k-th time slot, t_k is the trend component, $\Delta t_k = t_k - t_{k-1}$, $\epsilon_k \overset{\text{i.i.d.}}{\sim} N(0, \theta^2)$ is the noise of observation, and $w_k \overset{\text{i.i.d.}}{\sim} N(0, \lambda^2)$ is the noise in the trend component.

Eq. (10) indicates that the original data are composed of the trend component and the noise, and Eq. (11) indicates that the trend component is perturbed by Gaussian noise.

At the first step to calculate the trend component, the variances θ^2 and λ^2 are found by the Maximum Likelihood Estimation (MLE). Then, the trend component $t_i (i = t - h + 1, \cdots, t)$ is determined by the conditional expectation $E\left[t_i | x_{t-h+1..t}\right]$ with the probability of transition in Eqs. (10) and (11).

In terms of extracting the daily variation, the trend component approach is the same as the lowpass filter. However, the trend component extracts the main tendency of the traffic variation, while the lowpass filter extracts the targeted frequency component. Therefore, the trend component also extracts the variation mismatched to the frequency component when the variation can be taken as the main tendency.

3) Envelope: Extracting the variation of traffic upper bounds may be useful to predict the bandwidth required to accommodate the short-term traffic variation. In the third approach, we extract the upper bound variation by tracing the peak value in the fixed time interval. We divide the observed values $x_{t-h+1}, \cdots, x_t$ into $l = \frac{h}{\tau}$ intervals, where τ denotes the length of the intervals. The set of the time slots in the k-th interval is denoted as

$$I_k = \{(k-1)\tau + t - h + 1, \cdots, k\tau + t - h\}. \quad (12)$$

We set the interval length τ to 12 hours considering the daily variation.

The peak value in I_k is represented by x_{p_k}, where p_k represents the peak time slot denoted as

$$p_k = \arg\max_{j \in I_k} x_j. \quad (13)$$

In this paper, we extract the envelope by connecting the peak values $x_{p_1}, \cdots, x_{p_l}$ and the latest value $x_{p_{l+1}} = x_t$ with lines. By including the latest value x_t, the prediction can reflect the latest data. We simply perform the linear interpretation for points between $x_{p_{k-1}}$ and x_{p_k}, and each point is interpreted as

$$\begin{aligned} x_j &= x_{p_k} + \frac{x_{p_{k+1}} - x_{p_k}}{p_{k+1} - p_k}(j - p_k) \\ j &= p_k, p_k + 1, \cdots, p_{k+1}, \quad k = 1, \cdots, l. \end{aligned} \quad (14)$$

C. Prediction

The traffic prediction is performed on the basis of the prediction model after each preprocessing. To predict the traffic, many prediction models have been proposed. Though our prediction process is not limited by a certain prediction model, we use two traffic prediction models (ARIMA and SARIMA) as examples to discuss the effect of considering the periodicity of traffic variation. The model-based prediction learns the model parameters from inputted data and then predicts the future values on the basis of the obtained model. The rest of this section gives an overview of prediction with the ARIMA and SARIMA models.

1) Prediction models:

a) ARMA model: Before describing the ARIMA and SARIMA models, we briefly explain the ARMA model, which is the base model for the ARIMA and SARIMA models.

The ARMA model represents data at each time slot using the previous data and errors as

$$\begin{aligned} x_n &= \sum_{i=1}^{p} a_i x_{n-i} + \sum_{i=0}^{q} b_i \epsilon_{n-i} + c \\ b_0 &= 1, \end{aligned} \quad (15)$$

where p and q respectively denote the numbers of past data and errors on which the current data depends. a_i and b_i are the coefficients, ϵ_i is the error at the i-th time slot and c is a constant.

b) ARIMA model: The ARIMA model is an extension of the ARMA model so as to model the non-stationary data, such as the data whose mean value fluctuates over time. To apply the ARMA model to such data, the non-stationarity is removed. When the variation of the mean has a linear characteristic, the differenced data $\Delta x_n = x_n - x_{n-1}$ exclude the variation of the mean. In this manner, d times differencing operation Δ^d can remove the mean variation following a polynomial of degree d. In the ARIMA model, ARMA model in Eq. (15) is applied to the differenced data $\Delta^d x_n$.

c) SARIMA model: The SARIMA model is a generalization of the ARIMA model. Considering the periodicity, the SARIMA model applies a periodical differencing to the data as $\Delta_s x_n = x_n - x_{n-s}$, where s is a period length. After the D times of the periodical differencing $\Delta_s^D x_n$ are applied, the differencing method in the ARIMA model is also applied. Therefore, differenced data are finally denoted as $\Delta^d \Delta_s^D x_n$. Considering the daily periodicity and the weekday/weekend difference, we set s to the weekly length.

The differenced data are fitted to the following model, which expands the ARMA model by adding the data and errors in previous periods as

$$\begin{aligned} x_n =& \sum_{i=1}^{p} a_i x_{n-i} + \sum_{i=0}^{q} b_i \epsilon_{n-i} + c \\ &+ \sum_{j=1}^{P} A_j \sum_{i=1}^{p} a_i x_{n-sj-i} + \sum_{j=1}^{Q} B_j \sum_{i=0}^{q} b_i \epsilon_{n-sj-i} \\ b_0 =& 1, \end{aligned} \quad (16)$$

where P and Q denote the numbers of previous periods for depended data and errors, respectively. A_i and B_i are the coefficients that indicate how the previous i-th period affects the current time slot.

2) Model Fitting: An ARIMA or SARIMA model is fitted to the data by the following steps.

First, the differencing parameter is determined by differencing the data until the data become stationary. A stationarity test is performed by examining whether the data follow a non-stationary process $x_t = x_{t-1} + \epsilon$ called *unit root process*. We use the KPSS test [25] for determining d. The KPSS test examines the null hypothesis $\epsilon = 0$, which means the data are stationary. For determining D in the SARIMA model, we use the Canova-Hansen test [26]. The Canova-Hansen test applies the null hypothesis test to the Fourier coefficients variation of each period.

Second, the coefficients and the number of the terms in a model are determined. To determine the number of the terms in a model, we determine the coefficients by the MLE for each case of the number of terms. Then, we determine the model by selecting the model with the highest goodness among the models calculated by the MLE. The goodness of a model is defined by the Akaike Information Criterion (AIC) [27], which is defined by

$$\mathrm{AIC} = -2\log L + 2k, \quad (17)$$

where L is the maximized likelihood with the MLE and k is the number of parameters. $k = p+q+P+Q$ in the SARIMA model, and $k = p + q$ in the ARIMA model. A model with a large number of parameters can fit the data well but may fit the incidental variation such as noise. By penalizing k, AIC can select the best model while avoiding overfitting the data. To search for the model with the highest goodness, we use the method proposed by Hyndman *et al.* [28]. In this method, the model with the highest goodness is searched for by changing p, q, P and Q by one until no new model can improve AIC.

3) Prediction with Fitted Model: After the fitting of a model, the future traffic is predicted in accordance with the obtained model. The predicted traffic in the next k-th time slot is calculated as following conditional expectation of x_{t+k} given the previous observation values:

$$\bar{x}_{t+k} = E[x_{t+k}|x_{t-h+1..t}]. \tag{18}$$

According to the prediction model (15 or 16), the traffic rate of the next one time slot is directly calculated with observation values. The next two or more time slots are iteratively predicted by using the former predicted value instead of the observation value.

4) Confidence Interval: The model-based prediction can calculate the confidence interval for the prediction error. The upper confidence bound for the prediction can be calculated by $\bar{x}_{t+k} + \alpha\hat{\sigma}_{t+k}$, where $\bar{x}_{t+k}$ is the predicted traffic rate at the next k-th time slot, α is a parameter indicating the considered confidence level, and $\hat{\sigma}_{t+k} = \sqrt{V[x_{t+k}|x_{t-h+1..t}]}$ is the estimated standard deviation of prediction error where $V[x_{t+k}|x_{t-h+1..t}]$ is the conditional variance of predicted value given the observed values.

D. Range of Excluded Variation

The traffic variation excluded by the preprocessing should also be considered because it may cause the congestion. In this paper, we consider the excluded traffic variation by using the standard deviation of the excluded traffic variation. The standard deviation is calculated as

$$\sigma = \sqrt{\frac{1}{h}\sum_{k=t-h+1}^{t}(x_k - x'_k)^2}, \tag{19}$$

where x_k is the original traffic rate on a flow at k-th time slot and x'_k is the extracted variation by preprocessing. Using σ, we compensate for the excluded variation in the predicted traffic with $\bar{x}_t + \beta\sigma$ where β is a parameter indicating the confidence level for the upper bound prediction of the excluded variations.

Finally, the upper bound prediction including both the prediction error and the excluded variation in the preprocessing can be calculated as $\hat{x}_i = \bar{x}_i + \alpha\hat{\sigma}_i + \beta\sigma$.

V. Evaluation

A. Datasets

We use actual traffic traces from the backbone network of Internet2 [29], a research and education network in the United States. Figure 2 shows its topology, and the capacity of each link is described in [30]. The traffic data are collected by a Netflow protocol at each of the nine PoP routers. The sampling rate is one packet in every 100 packets, and aggregated data

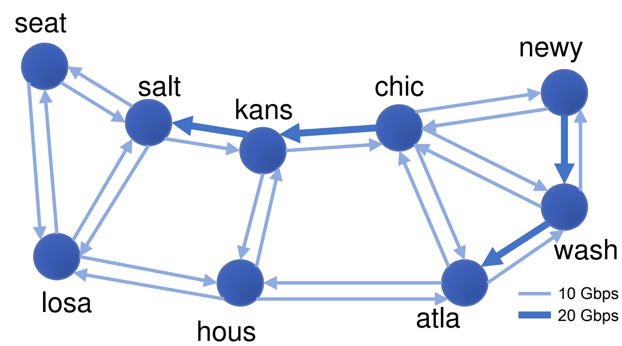


Fig. 2. Internet2 topology

TABLE I
Number of Data Used in Training and Test series

control period [hours]	training series [hours]	test series [hours]
3	336	3
12	336	12
24	336	24

are exported every five minutes. The sampling method has two main problems: it causes sampling errors, and there may be unsampled flows. However, it is not a critical problem for our evaluation because we only need the traffic rate of aggregated OD flow, which has a large number of samples. The large daily variation between day and night is mainly observed in the traffic variation. Focusing on such traffic variation over several hours, we set the length of the observation time slot to one hour and aggregate the observed data into the time slots.

We use four week's worth of data (11/28/2011 to 12/25/2011) aggregated into the flows between PoP routers using the BGP information. Table I summarizes the number of time slots used to train the traffic model, and number of time slots used to test the prediction accuracy or traffic engineering performance. We use the data from the previous two weeks as the observed data. We perform the preprocessing and prediction processes using these data, then compute optimal routes for the targeted control period using the predicted traffic. Finally, we evaluate these routes with actual traffic traces during the control period. We perform the above process 24 times, changing the start time of the prediction because traffic variation at the start of the prediction greatly affects its accuracy. Due to an over-provisioning policy [5], link utilization on the Internet2 network is less than 20%. Congestion rarely occurs in such situations, but this means that most of equipped capacity is redundant and unnecessary energy consumption is incurred. Our interest here is how to deal with congestion under limited resources in a way that reduces over-provisioning and power consumption costs, so we multiplied actual traffic amounts by 5 in the following evaluation.

B. Characteristics of the Traffic Prediction

1) Prediction Error: Before the evaluation of prediction based traffic engineering, we investigate the characteristics of the prediction method. First, to investigate accuracies of the prediction methods, we compare the *mean absolute percentage error (MAPE)*, defined as $\mathrm{MAPE}_k = \frac{1}{k}\sum_{i=t+1}^{t+k}\frac{|\bar{x}_i - x_i|}{x_i}$, where $\bar{x}_i$ is the predicted traffic rate, x_i is the actual traffic rate, and k is the length of the test series. This is one of the most frequently used metrics of prediction performance in previous work (e.g. [16, 18]).

Fig. 3 compares the MAPE corresponding to the length of the prediction target. In Fig. 3, "non-preprocess" means prediction using original data without preprocessing; "trend," "envelope," and "lowpass" mean prediction with each corresponding preprocessing; and "arima" and "sarima" mean prediction by the ARIMA and SARIMA models, respectively. Fig. 3 indicates that any traffic prediction includes prediction errors (e.g. at least around 40% in the case of "lowpass"). Fig. 3 also indicates that the MAPE generally increases as the prediction target becomes far from the current time slot except the case of "envelope". This increase is caused by the accumulation of one step prediction errors. In the SARIMA and ARIMA models, the future traffic value is predicted by continuing the one step prediction. As a result, even if the prediction errors included in each step is small, the prediction errors in the far future become large by accumulating the prediction errors included in each step.

From Fig. 3, prediction with the envelope has the largest prediction error. This is because the "envelope" includes the large short-term fluctuation, since the upper bound of traffic is frequently changed by temporal traffic changes. It is difficult for SARIMA or ARIMA model to fit to the traffic pattern which includes such a large fluctuation. As a result, the prediction result with the envelope has large error even in one-step prediction. This large prediction errors also makes the MAPE of envelope independent from the time slot, while the MAPE of the other prediction methods increases as the time slot becomes far from the current time slot.

In Fig. 3, prediction with the lowpass filter achieves the lowest prediction error, because the lowpass filter effectively improves the prediction accuracy by excluding noisy variation. However, this result does not necessarily mean that prediction methods using the lowpass filter are best suited to traffic engineering. The MAPE indicates the overall accuracy of the prediction of all flows. However, for the traffic engineering, the importance of the prediction may depend on the flows; the prediction of the large flows may be important since the large flows have a large impact on the link utilization. We demonstrate the impact of the prediction on traffic engineering in Subsection V-C.

2) Predicted Traffic Variation in Case of Daily Traffic Pattern: To investigate the detailed characteristic of each prediction method, we show the predicted traffic time series. As an example, Fig. 4 shows the prediction results of a flow using each preprocessing method without a confidence interval. In Fig. 4, "SARIMA" and "ARIMA" mean the prediction methods using the SARIMA model and the ARIMA model, respectively. Additionally, "real" means the actual traffic rate. Figs. 4(a)–(c) show the prediction results using the trend component, the lowpass filter, and the envelope. Fig. 4(d) shows the prediction results using original data without preprocessing.

Fig. 4 indicates that the preprocessing methods "trend" and "lowpass" improve the accuracy of the prediction of the daily variation. This is because the preprocessing excludes the noisy

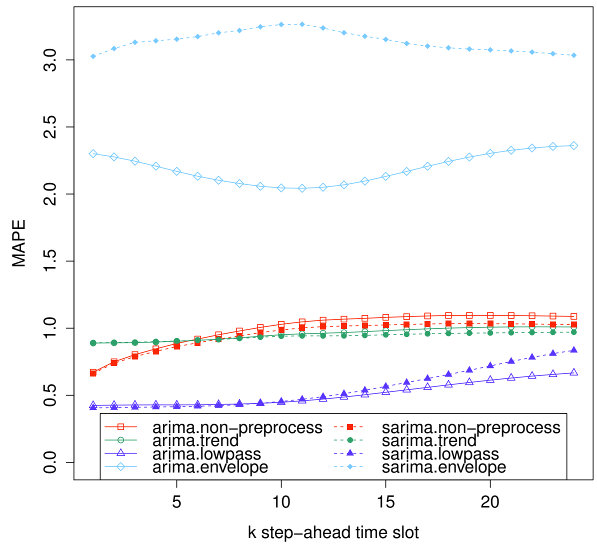
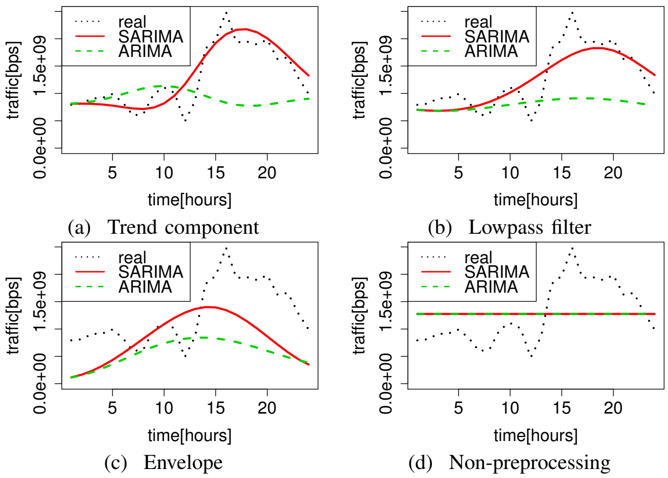


Fig. 3. MAPE of each prediction method

variation and clarifies the longer-term traffic variation, which enables accurate modeling of the daily traffic variation. Fig. 4 also indicates that the SARIMA model predicts the daily variation more accurately than the ARIMA model. This is because considering the periodicity in the prediction model is effective for predicting the daily variation. The results shown in Fig. 4 is different from those in Fig. 3. This is caused by that the prediction errors in the small flows; the MAPE is average of the prediction errors normalized by their actual values, and the prediction errors in the flows whose actual traffic amounts are small have significantly large impacts on the MAPE. In addition, the large prediction errors occur in the small flows, especially in the flows whose average traffic amounts are small but that have some spikes. Figure 5 shows an example of such small flows with spikes. In this figure, the vertical dotted line indicates the start point of the prediction. In this figure, there is a spike before the start point of the prediction. Such spikes cannot completely extracted by the trend or lowpass filter, and have a impact on the extracted long-term tendency. In the case of Fig. 3, the spike causes the sudden increase and decrease in the extracted tendency just before the start point of the prediction. As a result, the SARIMA model whose parameters are set to fit such sudden changes becomes different from the long-term trend of the traffic, and causes a large prediction error.

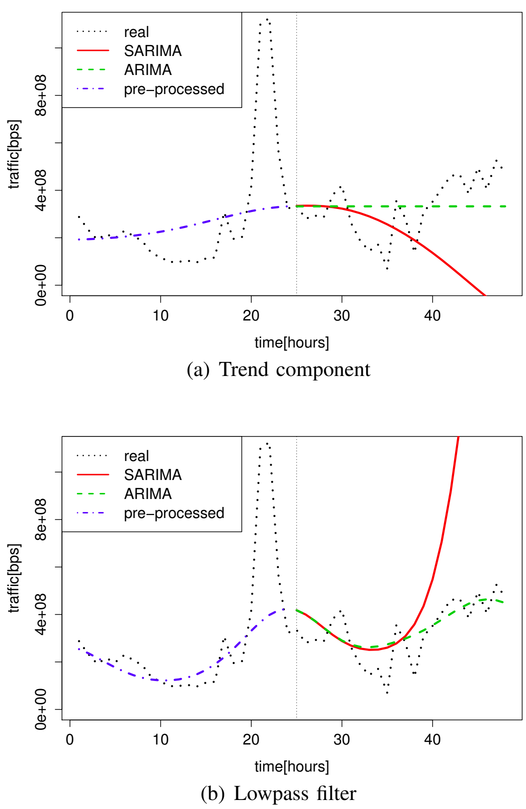
However, such spikes causing the large prediction errors are not found in the large flows. This is because the large flow includes a numerous number of user flows. The spikes in the flows are caused by the spikey behavior of the user flows. However, even if the flow includes the user flows whose behaviors are spikey, the spikey flows have only small impacts on the total traffic amounts of the flow, when the flow includes a large number of user flows.

Considering the traffic engineering, the prediction of the large flows such as a flow shown in Fig. 4 are important, compared with the small flows, since the large flows have a large impact on the link utilizations. Thus, the evaluation of the accuracy of the prediction is not sufficient, and we need



(a) Trend component (b) Lowpass filter (c) Envelope (d) Non-preprocessing

Fig. 4. Example of the predicted traffic time series using each preprocessing method



(a) Trend component

(b) Lowpass filter

Fig. 5. Example of failure prediction with SARIMA

the evaluation of the performance of the traffic engineering using the predicted traffic, which is discussed in Section V-C.

3) Predicted Traffic Variation in Case of Sudden Traffic Change: Though we do not need the accurate prediction on the spikey flows with small average traffic rates, the large flow may have the traffic variation which suddenly deviate from the longer-term pattern. Since the large flow affects the performance of the traffic engineering, the prediction should follow the main pattern of variation even in this case.

In this subsection, we investigate the accuracy of the prediction with a lowpass filter and trend component when such sudden change occurs in the large flows. Fig. 6 shows the prediction results of the SARIMA and ARIMA of a flow when the sudden traffic change is included. Fig. 6 plots the actual traffic variation and the predicted variation. In this figure, we plot the prediction results of two prediction methods (lowpass and trend) that can accurately predict the daily traffic variation as discussed in the previous subsection. The vertical dotted line indicates the start point of the prediction. "upper lowpass" and "upper trend" indicates the upper bound calculated by setting α and β to 0.84, which correspond to the confidence level of 80% for prediction error and short-term variation, respectively.

Unlike the spikey flows with small average traffic rates, the large flow, whose traffic rates suddenly increase, increases over multiple time slots as shown in Fig. 6. Thus, we can obtain the information used for the prediction of the sudden increase.

In Fig. 6, the method using the trend component follows the main variation under sudden traffic change more accurately than the method using the lowpass filter. This is because the trend component extracts the tendency to increase, while the lowpass filter removes all the variation shorter-term than daily variation. As a result, the lowpass filter removes the increasing tendency and underpredicts the sudden increase in traffic.

Fig. 6 also indicates that the predicted traffic of the ARIMA and SARIMA are almost the same. This is because the periodicity of the traffic variation is not effective for such sudden variation.

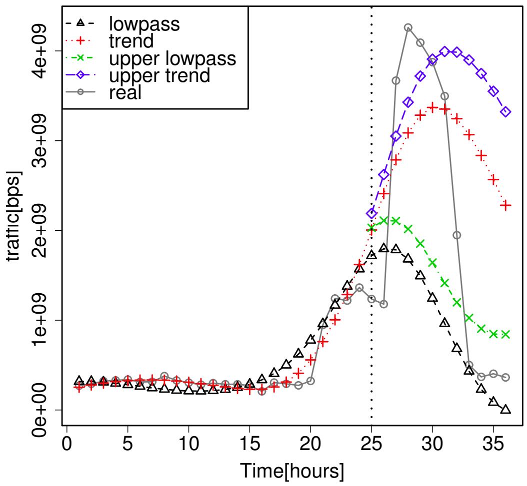
C. Performance of the Traffic Engineering

In this subsection, we investigate the performance of the traffic engineering using the predicted traffic. In this evaluation, we compute the optimum routes by solving the linear programming problem in Eqs. (3–7) using the predicted traffic. The linear programming problem is solved by CPLEX [31]. After the calculated routes are set, we investigate the performance of traffic engineering using actual traffic with the calculated routes.

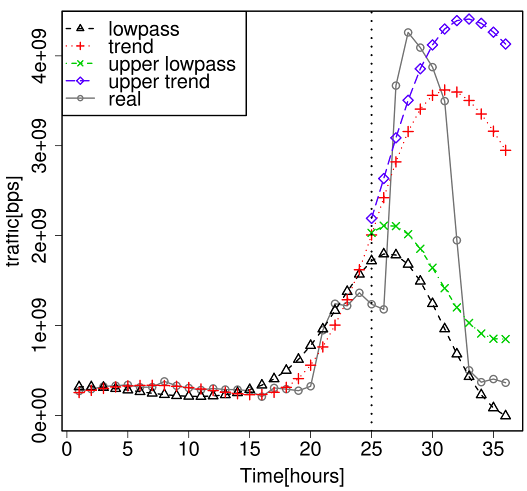
To evaluate the performance of traffic engineering, we investigate the link load of each link at each slot, which are the sum of traffic passing the link. In this evaluation, we focus on the peak link loads during the control period, because the network operator should set the bandwidth of each link so as to accommodate peak traffic without congestion. Among all links, we also focus on the most congested link, because the reduction of the link load on the most congested link is one of important objectives in the traffic engineering. Since the most congested link is passed by a large number of flows, the mitigation of the congestion of such a link improves the performance of a large number of flows. In addiction, the reduction of the link load on the most congested link avoids the concentration of traffic on a certain link, which may cause the necessity of enhancement of the link capacities. Thus, we use the maximum peak link loads defined by

$$r = \max_{l,k \in [t+1, t+f]} y_k(l) \qquad (20)$$

where f is the length of control period, and $y_k(l)$ is the traffic rate on the link l at the time slot k. The small r indicates that we do not require a large bandwidth to accommodate traffic without congestion.



(a) SARIMA prediction



(b) ARIMA prediction

Fig. 6. Example of the prediction for the sudden traffic change

In the evaluation, we normalize the value of r by that of *InvCap routing*, which is the most commonly used method for load balance routing. InvCap routing calculates the shortest path using the inverse of link capacities as weights, splitting the traffic equally among equal weighted paths. The normalized maximum peak link load r' is defined as

$$r' = \frac{r}{r_{\mathrm{InvCap}}} \tag{21}$$

where $r_{\mathrm{InvCap}} = \max_{l,k} y_k^{\mathrm{InvCap}}(l)$ is the maximum peak link load under the InvCap routing, and $y_k^{\mathrm{InvCap}}(l)$ is the traffic rate on the link l at the time slot k under the routes determined by InvCap routing. In our evaluation, we focus on the largest value of r' to clarify the reduction in the required bandwidths to avoid congestion.

1) Impact of Considering the Short-Term Variation and Prediction Errors: We compare the maximum peak link load by the traffic engineering using the predicted traffic with various α and β. Figures 7–8 show the complement cumulative distribution function (CCDF) of the normalized maximum peak link load. Figures 7–8 show the cases of SARIMA and ARIMA with the trend component for various control periods.

TABLE II
VALUES OF PARAMETERS FOR CONFIDENCE LEVELS (α,β)

control period [slots]	3	12	24
trend	(0.5,0.6)	(0.5,0.8)	(0.9,0.8)
lowpass	(0.8,0.5)	(0.7,0.8)	(0.8,0.9)
envelope	(0.8,-)	(0.9,-)	(0.8,-)
non-preprocess	(0.7,-)	(0.7,-)	(0.8,-)

In this comparison, when changing α, β is set to 0. On the other hand, when changing β, α is set to 0. Here, "mean" indicates the result using mean prediction without confidence interval, and "k %" means that confidence level corresponds to k%. The confidence interval corresponding to a confidence level is calculated under the assumption that predictive error follows a Gaussian distribution. This assumption can be examined by a Kolmogorv-Smirov (KS) test, and the null hypothesis of Gaussian distribution cannot be rejected at a significance level of 5% for more than 85% of OD flows.

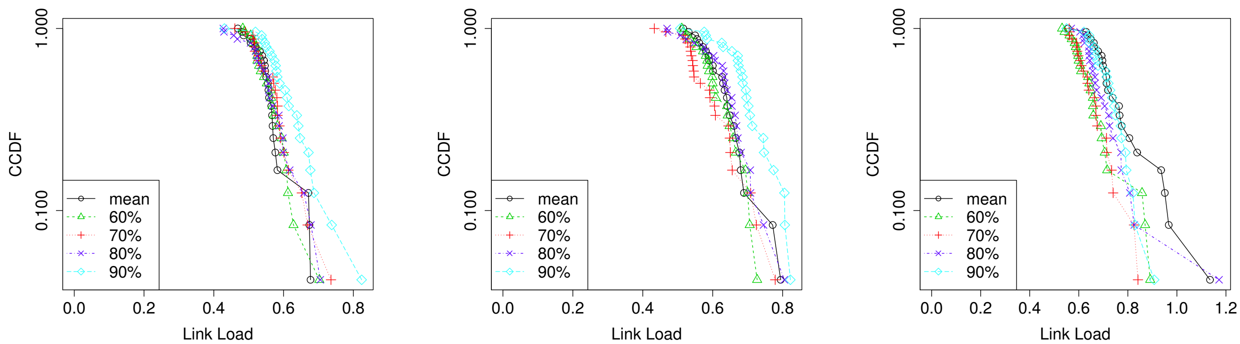
In most cases in Figs. 7–8, the largest link load of prediction-based traffic engineering is improved by considering the confidence level. This is because by considering the range of the short-term variation and prediction errors, the congestion occurred by temporal traffic variation can be avoided. When these ranges are not considered, temporal traffic variation sometimes causes congestion. Moreover, the difference between considering confidence intervals or not becomes large when the control period is large. This is because a large control period has a higher possibility of temporal traffic changes which may causes the congestion.

From Figs. 7–8, an overly large α sometimes requires large capacity, while the maximum peak link load is kept small even when β is set to a large value. When α is set to a large value, the predicted traffic rate of the distant future time slots becomes large because the traffic of the distant future time slot is difficult to predict and the variance of the prediction becomes large. As a result, too many resources are allocated to the traffic whose variance of the prediction is large. On the other hand, the variance of the short-term traffic variation is constant for all time slots in our prediction procedure. Thus, even when β is set to a large value, no traffic is predicted as a too large value. Therefore, setting β to a large value and α to 0 is sufficient to avoid future congestion caused by short-term variation.

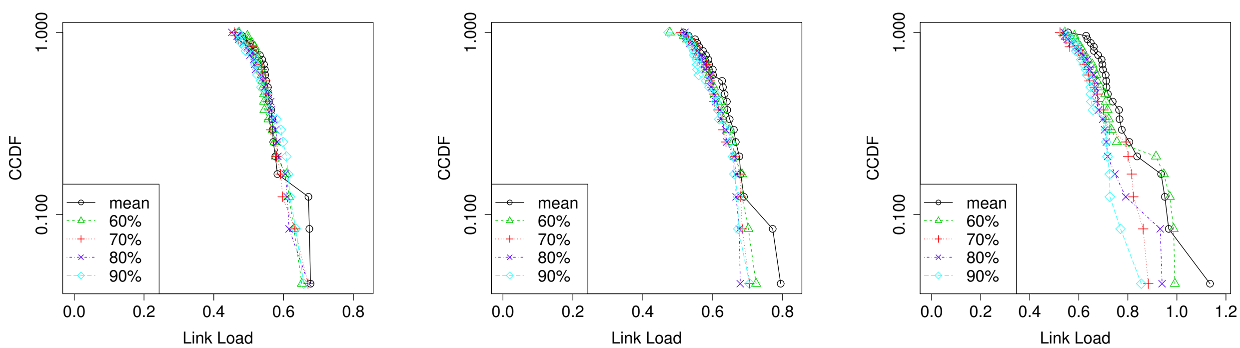
2) Comparison of the Preprocessing Methods: We compare the impacts of the preprocessing methods on the traffic engineering using the predicted traffic. Hereafter, we configure the confidence levels (α and β) of each prediction method in traffic engineering so that the maximum link load at the peak time slot is minimized. Table II shows the configured values of (α,β). For "envelope" and "non-preprocess", the value of β is not valid because it makes no sense to consider the removed variance in preprocessing in these methods.

Figs. 9–10 show the CCDF of normalized maximum peak link load at each control period when the traffic is predicted by the SARIMA or ARIMA with each preprocessing. Here, "observation-based TE" means calculating routes using the previous one hour's worth of data instead of the predicted traffic.

Figs. 9–10 show that the traffic engineering with the predic-

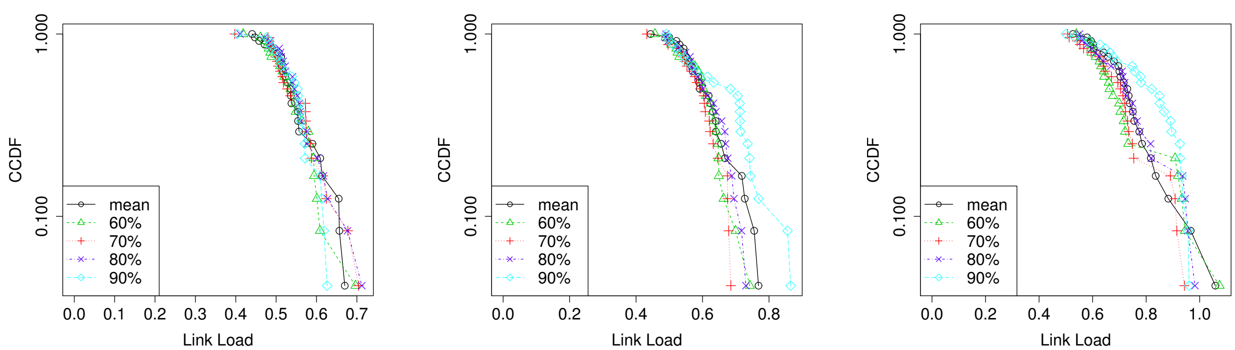


(a) Different α levels (control period: 3 slots) (b) Different α levels (control period: 12 slots) (c) Different α levels (control period: 24 slots)

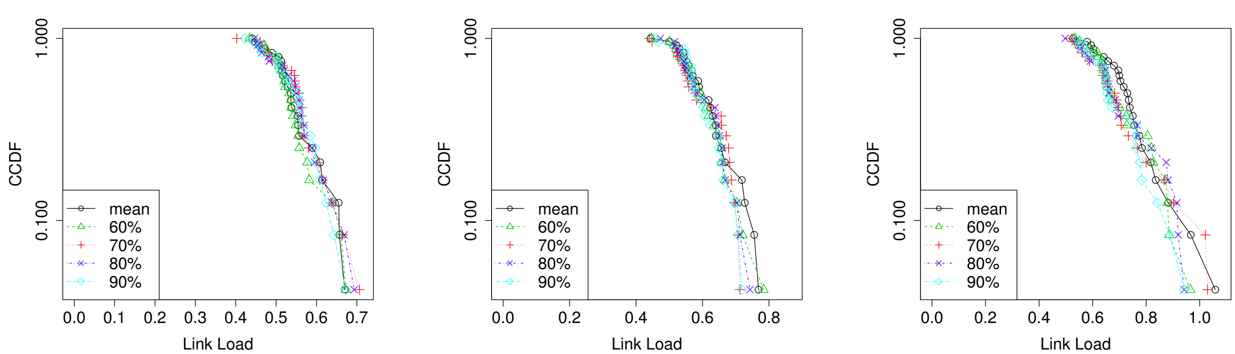


(d) Different β levels (control period: 3 slots) (e) Different β levels (control period: 12 slots) (f) Different β levels (control period: 24 slots)

Fig. 7. Complement Cumulative distribution of maximum peak link load normalized by InvCap routing with different confidence levels in the SARIMA model prediction with the trend component



(a) Different α levels (control period: 3 slots) (b) Different α levels (control period: 12 slots) (c) Different α levels (control period: 24 slots)



(d) Different β levels (control period: 3 slots) (e) Different β levels (control period: 12 slots) (f) Different β levels (control period: 24 slots)

Fig. 8. Complement Cumulative distribution of maximum peak link load normalized by InvCap routing with different confidence levels in the ARIMA model prediction with the trend component

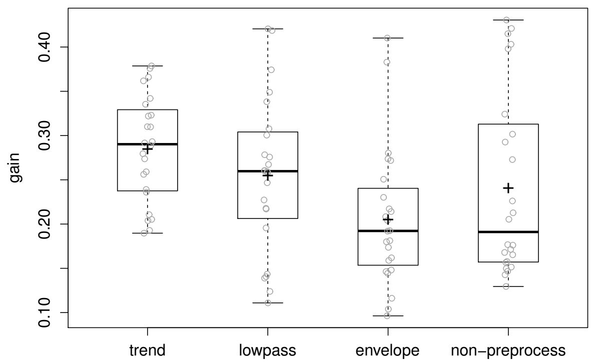


Fig. 11. Box plot of gain of prediction-based traffic engineering in reduction of maximum peak link load compared with the observation-based traffic engineering (Control period: 12slots)

tion keeps maximum peak link load low for the worst or almost worst case than "observation-based TE". This is because the traffic engineering using the predicted traffic variation sets the routes so as to avoid the future congestion by considering the future traffic variation. On the other hand, the "observation-based TE" sets the routes on the basis of the observed traffic, which sometimes differs from the future traffic significantly. As a result, the "observation-based TE" causes the congestion on a certain link.

Comparing the results of the different control periods, the maximum peak link loads increases as the control period becomes large. This is because the traffic changes included in the control period increases as the control period becomes large. As a result, more bandwidth is required to accommodate the traffic fluctuation during the control period. However, the short control period causes frequent route changes. In addition, the routing optimization may take a long time, the control period cannot be set to a small value especially in a large network. Even in the case of the long control period, the prediction based TE does not required a large bandwidth, compared with the observation based TE, which is one of the important advantage of the prediction based TE.

In Fig. 9, the SARIMA method with the trend keeps the worst value of link load small compared with the other methods. This is caused by that the SARIMA with the trend follows both of the long-term variation and sudden changes. As a result, traffic engineering using SARIMA with the trends allocates sufficient resources considering the long-term variation and sudden changes.

We also investigate the gain of the prediction based TE compared with "observation-based TE " , defined by

$$1 - \frac{r}{r_{\text{observation}}}$$

where r is the maximum peak link load of the prediction based TE, $r_{\text{observation}} = \max_{l,k} y_k^{\text{observation}}(l)$ is that of the observation based TE, and $y_k^{\text{observation}}(l)$ is the traffic rate on the link l at the time slot k under the routes determined by observation based TE. Figure 11 shows the performance gain of each control period when the control period is set to 12 slots. In Fig. 11, the maximum, third quartile, median, first quartile, and minimum values are plotted as horizontal line from top to bottom, and the average value is plotted as a crossed point. Similar to the previous results, we focus on the worst case to evaluate the reduction of capacity which must be prepared. Although there is difference among the prediction method, the worst case of gain is positive in all methods. Especially, the gain of the prediction based traffic engineering using SARIMA with the trend component is at least 18.9%. That is, the prediction based traffic engineering using SARIMA with trend component reduces the required bandwidth by 18.9% compared with the observation based traffic engineering.

3) Comparison of the ARIMA and SARIMA Models: We also compare the performance of the traffic engineering using the traffic predicted by the ARIMA and SARIMA models. Fig. 12 compares the CCDF of maximum peak link load normalized by InvCap routing. In Fig. 12, we present the results for the traffic engineering using the traffic predicted by the ARIMA/SARIMA with the trend component or lowpass filter, and the observation-based traffic engineering.

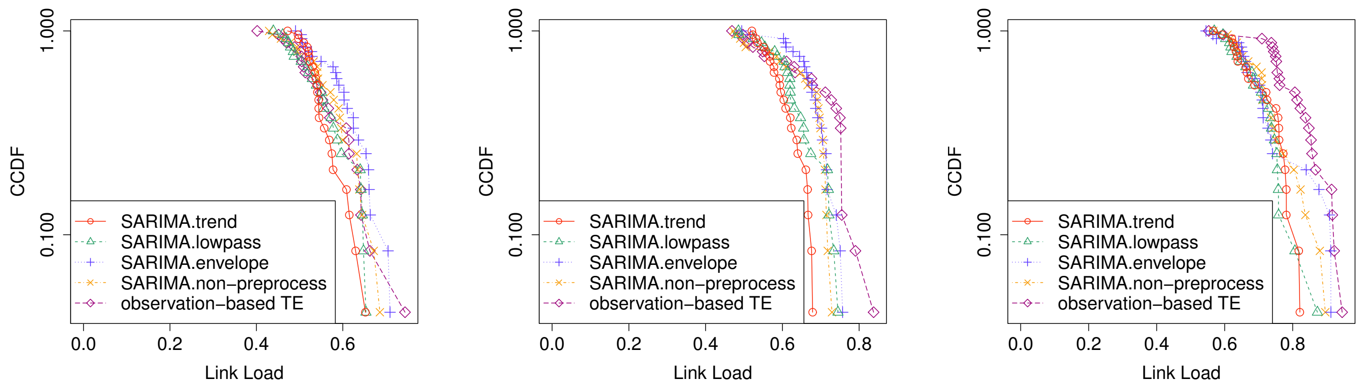
Fig. 12 indicates that the traffic engineering using the traffic predicted by the ARIMA keeps maximum peak link load similar in size to that of the traffic engineering using the traffic predicted by the SARIMA when the control period is small. This is because the traffic variation of the short control period can be predicted even without considering the periodicity of the traffic variation.

On the other hand, the SARIMA method achieves lower maximum peak link load than the ARIMA when the control period is 24 slots. Because the longer-term traffic variation cannot be predicted without considering the periodicity, the prediction errors of the ARIMA become large. On the other hand, the SARIMA predicts the longer-term variation accurately by considering the periodicity. As a result, the traffic engineering using the traffic predicted by the SARIMA allocates the resources to the traffic properly.

We also compare the computational complexity of the ARIMA and SARIMA. The computational complexity of the prediction with ARIMA and SARIMA is $O(m^3 t)$ where m is the oldest time slot in the model and t is the length of the data used to learn the parameters of the model. $m = max(sP, sQ) + max(p, q)$ in the SARIMA model and $m = max(p, q)$ in the ARIMA model. In the case of datasets used in our evaluation, the period length s equals 168, the number of period terms P or Q usually equals 1, and p or q usually equals 3–5. Therefore, the value of m in the SARIMA model is about 30–60 times larger than the ARIMA model. Thus, the ARIMA predicts the traffic about 38,000–180,000 times faster than the SARIMA.

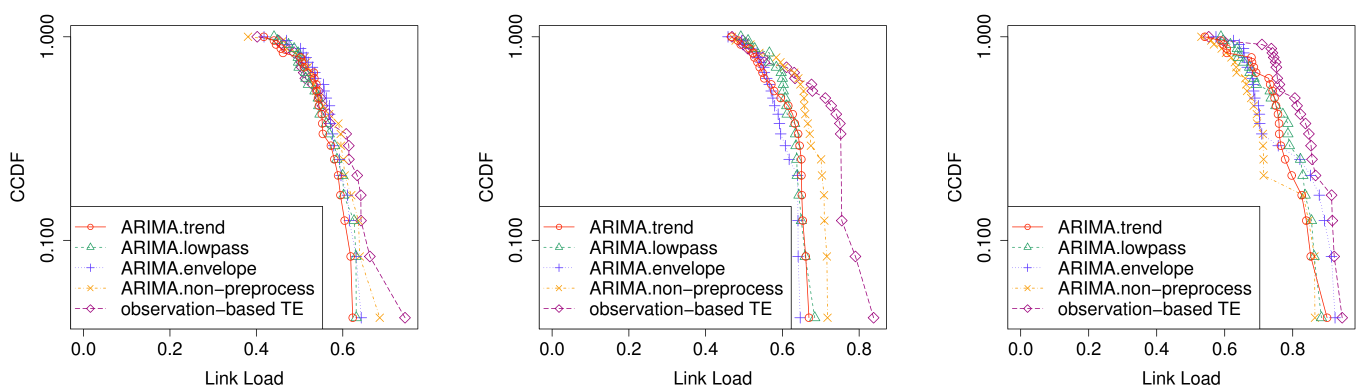
Therefore, the ARIMA prediction is useful for the traffic engineering targeting the short time scale. This kind of the traffic engineering may require the future traffic variation to be frequently recalculated, and the ARIMA predicts the traffic quickly. In addition, the ARIMA predicts the traffic variation with sufficient accuracy to avoid the congestion during the short control periods.

On the other hand, the SARIMA is required when we aim to calculate the stable routes for a long control period. By



(a) Control period: 3 slots (b) Control period: 12 slots (c) Control period: 24 slots

Fig. 9. Complement Cumulative distribution of maximum peak link load normalized by InvCap routing when the SARIMA model prediction with the different preprocessing is used in traffic engineering



(a) Control period: 3 slots (b) Control period: 12 slots (c) Control period: 24 slots

Fig. 10. Complement Cumulative distribution of required maximum peak link load normalized by InvCap routing when the ARIMA model prediction with the different preprocessing is used in traffic engineering

considering the longer-term traffic variation, we may handle even unexpected traffic changes by changing routes of only a small amount of traffic corresponding to the unexpected changes. As a result, we keep the network stable.

To achieve this, we must predict both the longer- and short-term traffic variations accurately. The SARIMA predicts the longer-term variation accurately, while the ARIMA cannot. Though the SARIMA takes a long time to predict the traffic, the future traffic does not need to be frequently recalculated when the target control period is large.

VI. Conclusion

In this paper, we proposed a traffic prediction procedure that obtains all the information required for traffic engineering. In our prediction procedure, we extract the longer-term variation before the prediction so as to improve the prediction accuracy of the daily traffic variation. The short-term traffic variation is also handled by calculating the variance of the traffic variation excluded by the preprocessing. Through the simulation, we clarified that the results of traffic engineering using the predicted traffic show that considering the short-term variation and prediction errors avoids the congestion caused by the prediction uncertainty. The results also indicate that the ARIMA model is suitable for the traffic engineering method targeting the short-term control period and the SARIMA model is suitable for the longer-term control period.

Our future work will include further investigation of more sophisticated prediction models such as neural networks, and developing traffic engineering methods suitable for use with predicted traffic.

Acknowledgements

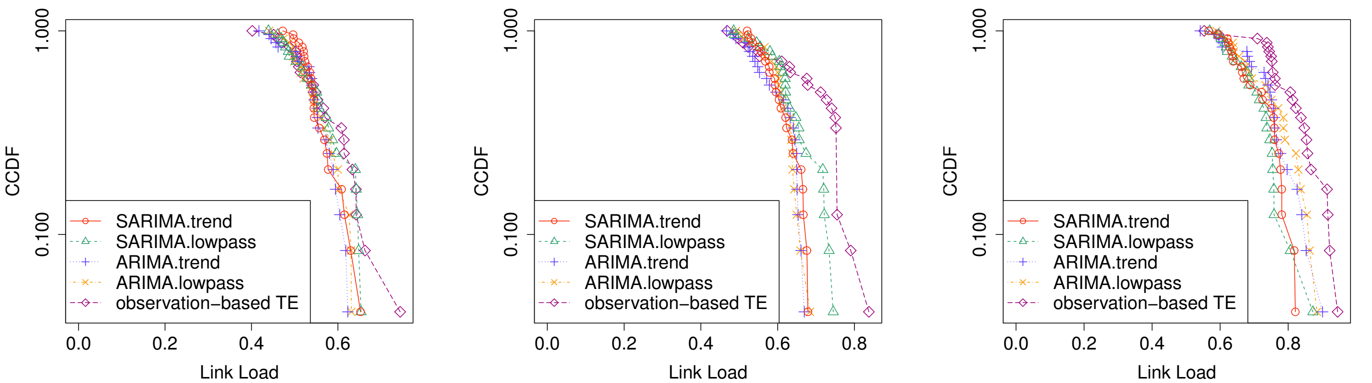
This work was supported in part by the Strategic Information and Communications R&D Promotion Programme of the Ministry of Internal Affairs and Communications, Japan.

References

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

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

[3] A. Hassidim, D. Raz, M. Segalov, and A. Shaqed, "Network utilization: The flow view," in *Proceedings of IEEE INFOCOM 2013*, Apr. 2013, pp. 1429–1437.



(a) Control period: 3 slots (b) Control period: 12 slots (c) Control period: 24 slots

Fig. 12. Complement Cumulative distribution of maximum peak link load normalized by InvCap routing when the ARIMA and SARIMA model predictions with the different preprocessing are used in traffic engineering

[4] S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot, "An approach to alleviate link overload as observed on an IP backbone," in *Proceedings of IEEE INFOCOM 2003*, Apr. 2003, pp. 406–416.

[5] "Internet2 headroom practice," available from https://wiki.internet2.edu/confluence/download/attachments/17383/Internet2+Headroom+Practice+8-14-08.pdf?version=1&modificationDate=1239396646935.

[6] H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg, "COPE: traffic engineering in dynamic networks," in *Proceedings of ACM SIGCOMM 2006*, vol. 36, no. 4, Aug. 2006, pp. 99–110.

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

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

[9] 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.

[10] D. O. Awduche, "MPLS and traffic engineering in IP networks," *IEEE Communications Magazine*, vol. 37, pp. 42–47, Dec. 1999.

[11] K. Srijith, L. Jacob, and A. Ananda, "TCP Vegas-A: Improving the performance of TCP Vegas," *Computer Communications*, vol. 28, no. 4, pp. 429–440, Mar. 2005.

[12] M. F. Zhani, H. Elbiaze, and F. Kamoun, "Analysis and prediction of real network traffic," *Journal of Networks*, vol. 4, no. 9, pp. 855–865, Nov. 2009.

[13] K. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot, "Long-term forecasting of Internet backbone traffic: Observations and initial models," in *Proceedings of IEEE INFOCOM 2003*, vol. 2, Mar. 2003, pp. 1178–1188.

[14] 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, no. 3, pp. 683–696, Jun. 2007.

[15] B. Zhou, D. He, Z. Sun, and W. H. Ng, "Network traffic modeling and prediction with ARIMA/GARCH," in *Proceedings of the Fourth International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks*, Sep. 2006, pp. 1–10.

[16] M. L. F. Miguel, M. C. Penna, J. C. Nievola, and M. E. Pellenz, "New models for long-term Internet traffic forecasting using artificial neural networks and flow based information," in *Proceedings of IEEE Network Operations and Management Symposium 2012*, Apr. 2012, pp. 1082–1088.

[17] C. Guang, G. Jian, and D. Wei, "Nonlinear-periodical network traffic behavioral forecast based on seasonal neural network model," in *Proceedings of the Second International Conference on Communications, Circuits and Systems*, vol. 1, Jun. 2004, pp. 683–687.

[18] P. Cortez, M. Rio, and P. Sousa, "Multi-scale Internet traffic forecasting using neural networks and time series methods," *Expert Systems*, vol. 29, no. 2, p. 143?155, May 2012.

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

[20] 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*, Dec. 2013, pp. 1592–1598.

[21] M. Hajiaghayi, J. H. Kim, T. Leighton, and H. Räcke, "Oblivious routing in directed graphs with random demands," in *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, May 2005, pp. 193–201.

[22] H. Räcke, "Optimal hierarchical decompositions for congestion minimization in networks," in *Proceedings of the fortieth annual ACM symposium on Theory of computing*, May 2008, pp. 255–264.

[23] G. Németh, "On a new competitve measure for oblibious routing," *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 1, pp. 117–123, Feb. 2014.

[24] G. Kitagawa and W. Gersch, "A smoothness priors-state space modeling of time series with trend and seasonality," *Journal of the American Statistical Association*, vol. 79, no. 386, pp. 378–389, Jun. 1984.

[25] D. Kwiatkowski, P. C. Philips, and P. Schmidt, "Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?" *Journal of Econometrics*, vol. 54, no. 1–3, pp. 159–178, Oct–Dec 1992.

[26] F. Canova and B. E. Hansen, "Are seasonal patterns constant over time? A test for seasonal stability," *Journal of Business & Economic Statistics*, vol. 13, no. 3, Jul. 1995.

[27] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Proceedings of the Second International Symposium on Information Theory*, 1973, pp. 267–281.

[28] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecast package for R," *Journal of Statistical Software*, vol. 27, no. 3, pp. 1–22, Jul. 2008.

[29] "Internet2 data," available from http://internet2.edu/observatory/archive/data-collections.html.

[30] S. Stoev, G. Michailidis, and J. Vaughan, "On global modeling of backbone network traffic," in *Proceedings of IEEE INFOCOM 2010*, Mar. 2010, p. 1?5.

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

Tatsuya Otoshi received M.E. degree in information science and technology in 2014 from Osaka University, where he is currently a postgraduate studying for a Ph.D. degree. His research interests include traf?c engineering and traffic prediction. He is a student member of IEEE.

Yuichi Ohsita received M.E. and Ph.D. degrees in information science and technology in 2005 and 2008 from Osaka University, where he is currently an assistant professor in the Graduate School of Information Science and Technology. His research interests include traffic matrix estimation and countermeasures against DDoS attacks. He is a member of IEICE, IEEE, and the Association for Computing Machinery (ACM).

Masayuki Murata received M.E. and Ph.D. degrees in information science and technology from Osaka University in 1984 and 1988. In April 1984, he joined the Tokyo Research Laboratory IBM Japan as a researcher. From September 1987 to January 1989, he was an assistant professor with the Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. From 1992 to 1999, he was an associate professor with the Graduate School of Engineering Science, Osaka University, and since April 1999, he has been a professor. He moved to the Graduate School of Information Science and Technology, Osaka University in April 2004. He has published more than 300 papers in international and domestic journals and conferences. His research interests include computer communication networks, performance modeling, and evaluation. He is a fellow of IEICE and a member of IEEE, the Association for Computing Machinery (ACM), The Internet Society, and IPSJ.

Yousuke Takahashi received an M.E. degree in information science and technology from Osaka University in 2009. He is currently a member of the Traffic Engineering Group, Communication Traffic & Service Quality Project at NTT Network Technology Laboratories. His research interests include network economy and traffic analysis. He is a member of IEICE.

Keisuke Ishibashi received an M.S. degree in mathematics from Tohoku University in 1995 and a Ph.D. degree in information science and technology from the University of Tokyo in 2005. He is currently a senior research engineer, Traffic Engineering Group, Communication Traffic & Service Quality Project at NTT Network Technology Laboratories. His research interests include Internet measurement and traffic analysis. He is a member of IEICE, IEEE, and the Operation Research Society of Japan.

Kohei Shiomoto received M.E and Ph.D. degrees in information and computer science from Osaka University in 1989 and 1998. He is currently a Senior Manager, Communication Traffic & Service Quality Technology at NTT Network Technology Laboratories. His research interests include traffic engineering, routing, network algorithms, optimization, and protocols. He is a fellow of IEICE and a member of IEEE and the Association for Computing Machinery (ACM).