PAPER

Network Tomography Using Routing Probability for Undeterministic Routing*

Rie TAGYO[†a)], **Daisuke IKEGAMI**[†], ***and*** **Ryoichi KAWAHARA**[††], ***Members***

SUMMARY The increased performance of mobile terminals has made it feasible to collect data using users' terminals. By making the best use of the network performance data widely collected in this way, network operators should deeply understand the current network conditions, identify the performance-degraded components in the network, and estimate the degree of their performance degradation. For their demands, one powerful solution with such end-to-end data measured by users' terminals is network tomography. Meanwhile, with the advance of network virtualization by software-defined networking, routing is dynamically changed due to congestion or other factors, and each end-to-end measurement flow collected from users may pass through different paths between even the same origin-destination node pair. Therefore, it is difficult and costly to identify through which path each measurement flow has passed, so it is also difficult to naively apply conventional network tomography to such networks where the measurement paths cannot be uniquely determined. We propose a novel network tomography for the networks with undeterministic routing where the measurement flows pass through multiple paths in spite of the origin-destination node pair being the same. The basic idea of our method is to introduce routing probability in accordance with the aggregated information of measurement flows. We present two algorithms and evaluate their performances by comparing them with algorithms of conventional tomography using determined routing information. Moreover, we verify that the proposed algorithms are applicable to a more practical network.

key words: *network tomography, routing probability, mobile crowd sensing (MCS)*

1. Introduction

The advent of high-performance terminals such as smartphones has made it possible to efficiently and extensively collect data. The data are cooperatively collected from mobile terminals to solve various specific problems. This data collection scheme, called mobile crowd sensing (MCS), has been attracting much attention because of its efficiency and convenience, and the data collected with MCS are used in various fields [3], [4]. Particularly in communication networking, MCS is commonly used to measure end-to-end performance [5]. Network operators take a considerable interest in using such end-to-end measured data for efficient network operation, taking into account actual user-communication quality. Specifically, network operators hope that data widely collected from users will enable them to better determine whether the current network conditions satisfy the user-communication quality or not. When the communication quality is degraded due to network congestion or failure, they want to identify the performance-degraded components in the network and estimate how much these components degrade communication quality.

To identify degraded components and estimate their degree of degradation in a network from end-to-end measurements, we can use a network monitoring method called network tomography [6]. Network tomography estimates the internal network status of individual components, such as the delay and packet loss ratio of each node or link, from end-to-end measurements. Several methods of network tomography using the data collected from MCS have been proposed. Dinc et al. [7] proposed an MCS-based data collection scheme for network tomography. Kawahara and Tonomura [8] proposed a network tomography to identify the degraded components in end-to-end communications, such as user terminals, access networks, or servers, by using the MCS data. To estimate the internal network status with network tomography, we generally use routing information of individual end-to-end measurement flows or information of which measurement flow passes through each network component, in addition to the end-to-end performance measurements of the flow, such as the delay or packet loss ratio.

Routing can be flexibly and dynamically controlled by using network virtualization by software-defined networking and network functions virtualization [9] to achieve fine-grained traffic engineering and/or service function chaining with various types of virtual network functions such as intrusion detection systems or deep packet inspection. In addition, schemes for stochastically routing flows to multi-paths are now used in real networks [10], [11]. As a result, individual measurement flows may pass through different paths even between the same origin-destination (O-D) node pair. The measurement flows collected using the MCS make it difficult and costly to identify through which path each flow passed by using a routing table or other observations. Therefore, conventional network tomography is difficult to naively apply to these networks.

To tackle this problem, we propose a novel network tomography for the networks with undeterministic routing in which the routing paths cannot be uniquely identified. The basic idea of our method is to introduce routing probability in accordance with the proportion of the traffic volume (e.g., number of flows) passing through each link among the traffic volume between the same O-D pair, where the traffic volume is measured within a certain measurement period.

Manuscript received October 1, 2020.
Manuscript publicized January 14, 2021.
[†]The authors are with Network Technology Laboratories, NTT Corporation, Tokyo, 180-8585 Japan.
[††]The author is with Faculty of Information Networking for Innovation and Design, Toyo University, Tokyo, 115-0053 Japan.
*Earlier versions of this paper were presented in [1], [2].
a) E-mail: rie.tagyo.rz@hco.ntt.co.jp
DOI: 10.1587/transcom.2020EBP3149

The proposed method makes it possible to identify the degraded components in a network and to estimate their degree of degradation even if the routing path of each single flow cannot be identified. We present two algorithms and evaluate their performances by comparing them with algorithms of conventional tomography using determined routing information. Moreover, in a practical network, we have to consider that the link delay has a continuous value and depends on delays of the neighboring links and that the measurements are affected by noise. Hence, we verify that the proposed algorithms are applicable to such practical networks.

This paper is organized as follows. In Sect. 2, we review related work on network tomography. We explain the assumed data collection system in Sect. 3.1, the model of the proposed network tomography method in Sect. 3.2 and the algorithms of our method in Sect. 3.3. In Sect. 4, we evaluate the performance of the algorithms of the proposed method by comparing them with algorithms of the conventional tomography method with the determined route information. We verify the applicability of the proposed algorithms to a practical network in Sect. 5. Finally, we conclude the paper in Sect. 6.

2. Related Work

Many network tomography methods have been developed for estimating the internal network status. Caceres et al. [12] proposed a multicast-based method to estimate loss rates by maximum likelihood estimation. Methods have been proposed for estimating the internal parameters of a network, i.e., delay distribution [13] and packet loss ratio [14], for unicast measurements. Firooz and Roy [15] derived a condition on a network routing matrix under which individual link delays can be estimated by assuming at most k performance-degraded links. Fan and Li [16] proposed a method of estimating link delay by using sparse Bayesian learning. Li et al. [17] proposed a method of determining the minimum monitor arrangement under which the performance metric of all links can be estimated even when the topology is changed, based on the assumption that the monitor node can freely select the measurement paths. More recently, Tootaghaj et al. [18] showed the optimal selection of monitoring paths considering the trade-off of identifiability and probing cost. Feng et al. [19] proposed bound-based network tomography that derives the tightest total error bound for unidentifiable links and a method to add a new monitor to maximally reduce the total error bound. Ma et al. [20] proposed a deep neural network based tomography scheme to predict the unmeasured performance metrics.

The same as our proposal, many network tomography methods using a compressed sensing (CS) approach have been proposed [21]. Bandara et al. [22] and Chen et al. [23] proposed algorithms to estimate congested or defective links by using CS. Fan et al. [24] estimate link delays considering clock synchronization error between origin and destination nodes using a CS approach. CS is a signal processing technique to reconstruct a signal from a series of sampled measurements, assuming that the signal vector is sparse. In network tomography, we can assume that the number of degraded links is small, i.e., the degraded links are sparse in the network, so we use CS to estimate the delay. Moreover, these studies assume that the measurement path can be identified.

We explain the existing work for network tomography to solve the problem that the measurement path cannot be uniquely identified. Pan et al. [25] proposed network tomography for multipath routing, where they identified the measurement path by using their previous method [26] in advance, and detected congested links by modeling the constraints between end-to-end paths and the links by using link delay variances and link loss rates. In their previous method [26], the measurement path is determined by a clustering method on the basis of inter-flow correlations using the delay measurements when multipath routing is used. In their scheme, the measurement path can be determined when the lengths of the shared paths between the target multipath and an identifiable path differ. In our study, we can use our proposal no matter whether multiple routing paths of measurements share the links or not.

Related to the introduction of the routing probability in our proposed method, Saito [27] theoretically analyzed the observability and independence of a system monitored with distributed sensing and showed its application for monitoring end-to-end network quality. He presented an example of determining the observability of the quality state (good/bad) of each link in the network called a binary problem. In his paper, when the quality between the O-D nodes is observable and the routing information between them is not observed, the path between them is considered to be stochastically chosen from among the multiple paths. We need to not just solve the binary problem but also determine how much the network component degrades. Rad et al. [28] assumed a network with random routing according to a certain distribution and modeled traffic over the network as a continuous-time bivariate Markov chain. They estimated the parameters of the model from end-to-end measurements and used these parameters to estimate the density of each link delay. Unlike them [28], we do not assume a specific traffic model. Herodotou et al. [29] computed the failure score of each link/device by a probabilistic routing modeling when the measurement path cannot be specified on the basis of the principle that only the shortest paths between the origin and destination nodes are valid routes and that all next hops from a node are equally probable. In contrast, we deal with multiple paths with no limitation on the path selection. We propose a novel network tomography that can be used without depending on assumptions such as traffic models or routing schemes.

3. Network Tomography with Routing Probability

We explain our proposed method that makes it possible to estimate the degree of degradation at each link in the network by introducing routing probability even if the path of

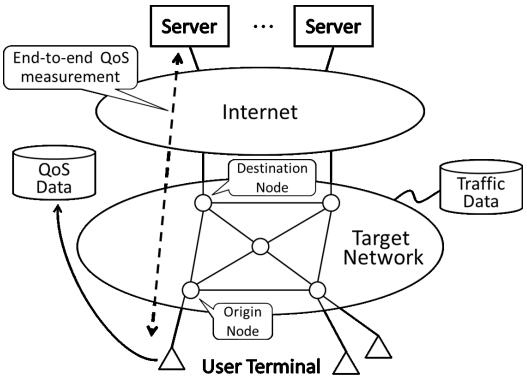

Fig. 1 Data collection system.

Table 1 Notation list.

L	total number of links in network $\mathcal{G}$
$z_{m,n}$	measured QoS of flow n between O-D pair m
M	total number of O-D pairs for measurements
N_m	total number of measurement flows between O-D pair m
y_m, $\boldsymbol{y}$	average of QoS between O-D pair m and QoS vector
$a_{m,l}$	traffic ratio between O-D pair m of traffic through link l
$\boldsymbol{A}$	routing probability matrix
$F_{m,l}$	number of flows between O-D pair m passing through link l
Ω_O	set of links connected to O node
x_l, $\boldsymbol{x}$	degradation degree of link l and degradation vector
λ	regularization parameter for estimation
Φ	set of non-degraded O-D pairs
Θ_m	set of links included in flows between O-D pair m
Θ	set of non-degraded links in all L links
$\boldsymbol{A}'$, $\boldsymbol{y}'$	partially deleted matrix $\boldsymbol{A}$ and vector $\boldsymbol{y}$
L_{deg}	number of degraded links
p	routing probability of the shortest path
$\mathcal{S}_{m,1}$, $\mathcal{S}_{m,2}$	set of links on the shortest and quasi-shortest paths of O-D pair m, respectively
$\boldsymbol{B}$	routing matrix of conventional tomography
$\boldsymbol{y}^*$	QoS vector of flows on the shortest path
d_{pl}, d_{ql}	propagation and queueing delays of link l
T_m, T'_m	traffic volume between O-D pair m in target and normal network
ρ_l	l-th link utilization in target network
y_m^{normal}, $\boldsymbol{y}^{\mathrm{normal}}$	normal delay between O-D pair m and normal delay vector
N	number of flows between O-D pair in target network
N'	number of flows between O-D pair in normal network
N_A	number of flows between O-D pair for estimating $\boldsymbol{A}$

each flow cannot be uniquely identified. For a routing probability, we use the proportion of the traffic volume (e.g., number of flows) passing through each link among the traffic volume between the same O-D pair, where the traffic volume is measured within a certain measurement period. That is, we utilize not only active measurement like general network tomography, but also passive measurement to estimate routing probability. In this section, we explain a data collection system and a network model for our network tomography. Then, we show one example of methods of obtaining routing probability, and propose our method including two algorithms.

3.1 Data Collection System

The data collection system used in our study is shown in Fig. 1. We assume that two types of data are collected. One is end-to-end quality of service (QoS) measurement data, for example, between an end user terminal and content server. The measured QoS is, for example, the end-to-end delay or packet loss ratio. It is also assumed that we can identify the O-D nodes for the flow in the network by using the addresses of the flows that are recorded. The other type of data is traffic data (based on network topology information), which is the number of flows between the O-D nodes monitored for each link in a certain period. One example of monitoring the number of flows is Netflow. The Netflow data include the O-D IP addresses, so we can identify the corresponding O-D nodes for each flow. These data are used to construct routing probability in our method and are discussed in detail in the next section.

3.2 Model

3.2.1 Network Model

We explain the network model of our proposed method. (The notations in this paper are summarized in Table 1.) Consider a network represented by a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ denotes the set of links, and the number of nodes and links are $|\mathcal{V}|$ and $|\mathcal{E}|(= L)$, respectively. The end nodes of the QoS measurement flows are defined as an origin node "O" and destination node "D" through which the measurement flows enter and exit the network we focus on, respectively. Both O and D are selected from $|\mathcal{V}|$ nodes in the network. Let $z_{m,n}$ be the measured QoS (hereafter QoS) of flow n between O-D pair m (the number of O-D pairs is M and $m = 1, \ldots, M$). The number of flows between O-D pair m is defined as N_m. Let y_m be the average of QoS between the O-D pair m, where y_m is given by $y_m = \frac{1}{N_m} \sum_{n=1}^{N_m} z_{m,n}$. We construct a vector $\boldsymbol{y} \in R^{M\times 1}$ with the m-th element as y_m and call it the *QoS vector*. Regarding QoS, when round trip time (RTT) is used, we treat the network as an undirected graph. When one-way quality is used, we treat it as a directed graph.

We denote the traffic ratio of the traffic passing through link l to all traffic between O-D pair m as $a_{m,l}$, where $a_{m,l}$ is from 0 to 1, and we estimate and determine it from the traffic data described in Sect. 3.1. An example of a method of estimating $a_{m,l}$ is described in Sect. 3.2.2. We then construct a routing probability matrix $\boldsymbol{A} \in R^{M\times L}$ with the (m, l)-th element as $a_{m,l}$. This probabilistically expressed routing matrix is the fundamental idea of our proposed method, and we model the probabilistic routing of multiple paths by focusing on the O-D pair and aggregating $z_{m,n}$s into y_m instead of identifying individual QoS measurement flows. By using the above model, we estimate the degradation degree of each link.

3.2.2 Estimation of Routing Probability

We introduce an example method of estimating $a_{m,l}$, which is an element of the routing probability matrix $\boldsymbol{A}$, on the basis of the collected data introduced in Sect. 3.1. From the traffic data, we can determine the number of flows that pass through each link and identify the O-D pair of each flow. Let $F_{m,l}$ be the number of flows between O-D pair m passing through link l. By using $F_{m,l}$, we calculate $a_{m,l}$ as follows.

$$a_{m,l} = \frac{F_{m,l}}{\sum_{l' \in \Omega_O} F_{m,l'}}, \tag{1}$$

where Ω_O is a set of links connected to node O of O-D pair m. The time complexity of estimating of $\boldsymbol{A}$ depends on the number of O-D pairs and number of flows. Instead of the above, $a_{m,l}$ may also be determined by using information on configuration such as a routing table if it is available.

3.3 Proposed Method

We explain our proposed method including two algorithms: a simple algorithm, and its extended version. Hereafter, an end-to-end connection or communication flow corresponding to the QoS measurement is called *a flow*. The degradation degree of link l ($l = 1, \ldots, L$) is denoted as x_l. The degradation degree of a link might be defined as an additional increase in the delay of the link from the normal state, for example. Let the degradation vector $\boldsymbol{x}$ having an element x_l represent the degradation degree of all links. It is defined as

$$\boldsymbol{x} = (x_1, x_2, \cdots, x_L)^\top, \tag{2}$$

where $\top$ means the transpose of the matrix. In Sect. 3.3.1, we discuss the basic estimation of $\boldsymbol{x}$ based on CS with a non-negative constraint, and in Sect. 3.3.2, we more precisely calculate $\boldsymbol{x}$ by using non-negative least squares (NNLS) when it is not an ill-posed problem in addition to the estimation by CS.

3.3.1 Proposed Algorithm 1 (PA1)

We estimate $\boldsymbol{x}$ by using $\boldsymbol{y}$ and $\boldsymbol{A}$. We do not set QoS as $\boldsymbol{y}$ as it is. Instead, $\boldsymbol{y}$ is set to the indicator having the value of 0 if QoS is less than a threshold; otherwise, it is set to a positive real number by substituting the baseline, for example, delays that occur when the network is in the normal state if we use delay as QoS. This idea has already been proposed [8]. After this preprocessing, a value of 0 means that the quality of the measured path is good. When we use the packet loss ratio as QoS, we do not need this preprocessing because we assume that the packet loss ratio of the normal state is negligibly small, and we transform y_m into $-\log(1 - y_m)$ [8]. By this preprocessing of $\boldsymbol{y}$, $\boldsymbol{x}$ can be regarded as sparse. We estimate $\boldsymbol{x}$ by solving the following problem on the basis of CS with a non-negative constraint.

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left\{ \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 \right\} \quad \text{subject to} \quad \boldsymbol{x} \geq 0, \tag{3}$$

where $\| * \|_2$ and $\| * \|_1$ represent the calculation of ℓ_2-norm and ℓ_1-norm, respectively, and λ represents the regularization parameter. Since we use a non-negative constraint, $\boldsymbol{x}$ has a non-negative value due to QoS. We can obtain the estimated vector $\hat{\boldsymbol{x}}$ from the above optimization problem.

3.3.2 Proposed Algorithm 2 (PA2)

Our proposed algorithm, PA2, consists of two parts. The first part is for excluding the non-degraded links from the estimation target links. We exclude links known to be non-degraded from the estimation target links in advance because non-degraded links do not need to be estimated. The operations related to this first part were introduced by Bandara et al. [22] and Chen et al. [23]. The second part is for estimating $\boldsymbol{x}$. In addition to using CS, depending on the condition of $\boldsymbol{A}$, we estimate $\boldsymbol{x}$ by NNLS if it is not an ill-posed problem.

1. Exclusion of non-degraded links and O-D pairs: To construct $\boldsymbol{y}$, the same preprocessing is conducted as PA1 introduced in Sect. 3.3.1. As described above, element y_m of the preprocessed $\boldsymbol{y}$ taking 0 means links through which the flows between O-D pair m pass are not degraded. A set of O-D pairs corresponding to the flow that is not degraded is defined as $\Phi = \{m | y_m = 0, m = 1, \ldots, M\}$. A set of links through which the flows between the O-D pair m pass is denoted as $\Theta_m = \{l | a_{m,l} \neq 0, l = 1, \ldots, L\}$. Hence, a set of non-degraded links in all L links is represented as $\Theta = \bigcup_{m \in \Phi} \Theta_m$. On the basis of the above, the m-th row and l-th column of $\boldsymbol{A}$ and the m-th element of $\boldsymbol{y}$ where $\forall m \in \Phi$, $\forall l \in \Theta$ are deleted. The m-th row or element and the l-th column correspond to the non-degradation O-D pair and link, respectively. Let the partially deleted matrix and vector be $\boldsymbol{A}'$ and $\boldsymbol{y}'$. We can identify $x_l = 0$ without estimating when $l \in \Theta$. On the other hand, when $l \notin \Theta$, we estimate x_l by using the method introduced in the next part.

2. Estimation of degradation vector: By using $\boldsymbol{A}'$ and $\boldsymbol{y}'$, we estimate $\boldsymbol{x}$. If $\boldsymbol{A}'$ is full rank or full column rank, we estimate $\boldsymbol{x}$ by NNLS as follows.

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}'\boldsymbol{x} - \boldsymbol{y}'\|_2^2, \quad \text{subject to} \quad \boldsymbol{x} \geq 0, \tag{4}$$

If neither, we estimate $\boldsymbol{x}$ using Eq. (3) after changing $\boldsymbol{A}, \boldsymbol{y}$ to $\boldsymbol{A}', \boldsymbol{y}'$, respectively.

4. Performance Evaluation of Proposed Method in Basic Network

4.1 Network Model for Simulation

We evaluate the performance of the proposed method on the network in which the degradation degree of each link is preset to a fixed value. In this section, we introduce the network

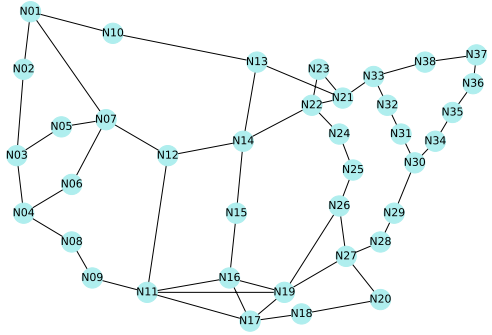

Fig. 2 Network topology.

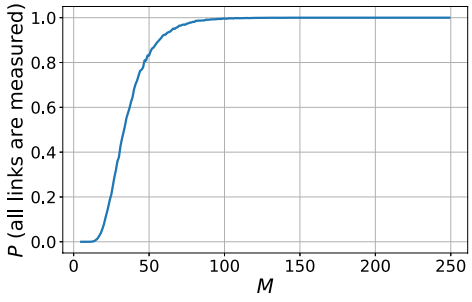

Fig. 3 Characteristics of topology.

topology and the parameters in our evaluation.

The network topology for our evaluation is shown in Fig. 2. It was made on the basis of Internet 2 layer 2 service topology [30] and has 38 nodes and 52 links, that is, $|\mathcal{V}| = 38$, $L = 52$, respectively. In this network, it is supposed that L_{deg} links are degraded and all other links are normal. The degradation degree of each degraded link is 100 and that of each normal link is 0. The M O-D pairs are randomly selected from all nodes of the network. Each flow between an O-D pair may pass through multiple paths. For simplicity, the number of paths is set to two. One is the shortest path between the O-D pair with probability p, and the other is a quasi-shortest path with probability $1 - p$. Instead of generating individual flows, we calculate $p \sum_{l \in \mathcal{S}_{m,1}} x_l + (1 - p) \sum_{l \in \mathcal{S}_{m,2}} x_l$ as y_m, where $\mathcal{S}_{m,1}$ is a set of links on the shortest path of O-D pair m and $\mathcal{S}_{m,2}$ is one of the links on the quasi-shortest path of O-D pair m. Moreover, the (m, l)-th element of $\boldsymbol{A}$, $a_{m,l}$ is calculated from $p\, \mathbf{1}_{\mathcal{S}_{m,1}}(l) + (1 - p)\, \mathbf{1}_{\mathcal{S}_{m,2}}(l)$, where $\mathbf{1}_{\mathcal{S}}(i)$ represents the indicator function, which means that $\mathbf{1}_{\mathcal{S}}(i) = 1$ if link i is in a set $\mathcal{S}$; otherwise 0. Note that, in this evaluation, we treat the network as an undirected graph.

The characteristics of the topology are shown in Fig. 3. The horizontal axis represents M. The vertical axis shows the average ratio of links that are included in the flows to all links of the network (referred to as P(all links are measured) in the graph) for 10,000 trials of an O-D pair selection. From this figure, the flows can almost cover the entire network when we choose about 100 or more O-D pairs.

4.2 Comparison with Conventional Method

We evaluate the proposed method using routing probability by comparing it with conventional tomography. As introduced in Sect. 2, since existing methods that use undetermined route information are difficult to compare with our proposed method due to the difference in the assumptions, we use a conventional method with determined route information. Specifically, we use a CS approach [15], [22], [23] for comparison, which estimates $\boldsymbol{x}$ with CS by using a determined routing matrix, which represents through which link the flow passed, and the element of the matrix corresponding to link through which the flow passes is 1, otherwise the element is 0. Two algorithms are used as the conventional method.

Conventional Algorithm 1 (CA1): The same $\boldsymbol{y}$ as that for the proposed method is used. Since we do not have the information of which path is selected for each flow, we simply suppose that all the flows pass through the shortest path. If the flow between O-D pair m passes through link l (i.e., $l \in \mathcal{S}_{m,1}$), the (m, l)-th element of routing matrix $b_{m,l} = 1$; otherwise, $b_{m,l} = 0$. Using the routing matrix $\boldsymbol{B}$ and $\boldsymbol{y}$, we estimate $\boldsymbol{x}$ by using Eq. (3) in which $\boldsymbol{A}$ is replaced with $\boldsymbol{B}$.

Conventional Algorithm 2 (CA2): We also use the same $\boldsymbol{B}$ as in CA1. In this method, we suppose that we know the QoS of the shortest path corresponding to $\boldsymbol{B}$. Specifically, we denote $\boldsymbol{y}^*$ as the QoS measured when the flows pass through the shortest paths. We estimate $\boldsymbol{x}$ by using Eq. (3) in which $\boldsymbol{A}$ and $\boldsymbol{y}$ are replaced with $\boldsymbol{B}$ and $\boldsymbol{y}^*$, respectively.

We compare the estimation accuracies of PA1 and PA2 of our method with those of CA1 and CA2 of the conventional method. The parameters are set to $L_{\text{deg}} = 3$ and $p = 2/3$. For each trial, the three degraded links are randomly selected from all 52 links of the network. The M O-D pairs are also randomly selected. Then, we calculate the root mean square error (RMSE) between the true and estimated values of $\boldsymbol{x}$ for each trial. We conducted 5000 trials. Figure 4 shows the average RMSEs of 5000 trials when M varies from 5 to 250. Note that the trials may include cases in which the flows do not pass through any degraded links. This is because when our method is applied in an actual environment, it is difficult to assume all degraded links are included in the flows. Note that λ was set to 0.05 in this section.

First, to assess the effect of introducing $\boldsymbol{A}$, we compare PA1 with the algorithms of the conventional method, as shown in Fig. 4. The difference between the two proposed algorithms, PA1 and PA2, will be described later. PA1, CA1, and CA2 are represented by solid, dashed&dotted, and dotted lines, respectively. For the three algorithms, the same estimation algorithm based on CS is used. Our PA1 has a low average RMSE in all ranges of M, which means that the proposed method exhibits good estimation accuracy by using the routing probability matrix. For the three algorithms, as the number of O-D pairs increases, the average RMSE

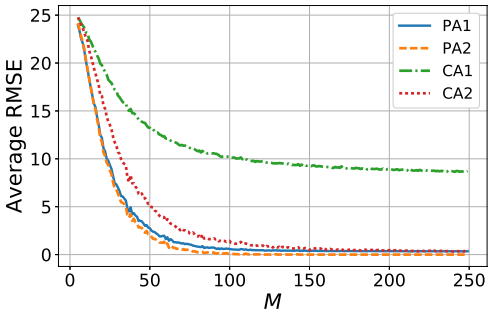

Fig. 4 Average RMSE vs. number of O-D pairs M.

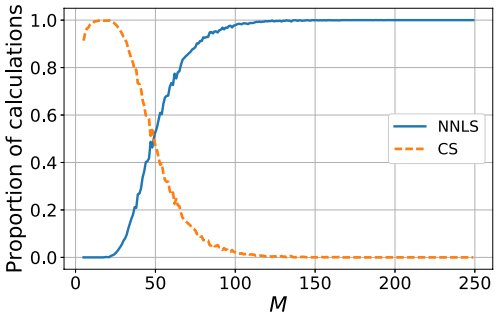

Fig. 6 Proportion between CS and NNLS of estimation in PA2.

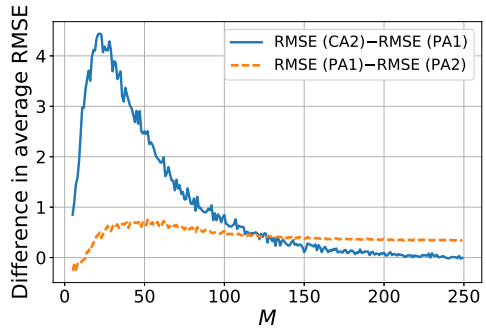

Fig. 5 Difference between algorithms in average RMSE.

converges to the lower limit. CA1 has the highest average RMSE, which does not become less than about 8, even when the number of O-D pairs increases. This result is due to the unavailability of the information about two paths for estimation. CA1 use the routing information of only the shortest path even though CA1 use $\boldsymbol{y}$, which includes the QoS of the shortest and quasi-shortest paths; therefore, PA1 performs better than CA1.

We focus on the difference between the accuracies of PA1 and CA2. The solid line in Fig. 5 shows the difference between the average RMSEs of these algorithms. Even CA2, in which the QoS of the shortest path can be known, has a higher average RMSE than PA1 in the area where the number of O-D pairs is small and has nearly the same average RMSE as PA1 in the area where the number of O-D pairs exceeds about 200. This indicates that introducing the routing probability matrix in PA1 makes it possible to use the information of two paths, but CA2 can use the information of only the shortest path, so the measurement coverage rate of the link in CA2 is lower than that of the link in PA1. Due to this difference, PA1 has better accuracy than CA2.

Next, we compare PA2 (dashed line) with PA1 (solid line), as shown in Fig. 4. The difference between the average RMSEs of PA2 and PA1 is shown by the dashed line in Fig. 5. PA2 has a lower RMSE than PA1 in most ranges of M. This shows that PA2 results in a more exact solution than PA1. This is because NNLS in PA2 gives the exact solution. Figure 6 shows the proportion between CS and NNLS in estimating by PA2, that is, the ratio between the solution of NNLS and CS in regards to the number of O-D pairs. In this figure, estimation by NNLS based on Eq. (4) is indicated by the solid line, and estimation by CS is indicated by the dashed line. In the area where M is around 50 in Fig. 5, the difference in average RMSE is large due to the effect of estimation by NNLS. When the number of O-D pairs increases, the average RMSE of PA1 becomes smaller, as shown in Fig. 4; therefore, the difference between the RMSEs of PA1 and PA2 becomes small and stabilizes at around 0.4. NNLS is supposed to give the exact solution, and this difference is the difference between the estimation value by CS and the exact solution.

Note that PA2 took only 10.33 seconds to calculate 5000 times where $M = 100$ by implementing *Python* on *Xeon E5-2667*. When our method is used for actual operation, we only have to calculate once for a measurement dataset, so PA2 takes a feasibly short amount of time.

5. Performance Evaluation for Practical Network

In this section, we verify the applicability of the proposed method to a more practical network in which practical delays occur. For practical delay, we consider that delays have continuous values and depend on delays of neighboring links in contrast to the previous section where delays have random and discrete values, 0 and 100, with no dependence among neighboring links. We assess the effect of the practical delays on the estimation accuracy of the proposed method. In addition, we consider measurement values including noise that occurs when we measure QoS and evaluate the effect of noise on the estimation accuracy.

5.1 Practical Delay Model

We assess the performance of the proposed method for practical delays with the same network topology used in the previous section, and the number of paths for each flow is also set to two. In this section, we do not give the fixed value to the degraded link, but simulate a traffic matrix to obtain practical delays. We consider propagation and queueing delays that are defined as $d_{\mathrm{p}l}$ and $D_{\mathrm{q}l}$ for link l, respectively. Note that the queueing delay is a random variable. Queueing delays are set in the network we use as follows.

First, we add traffic flows between all O-D pairs so that no link utilization exceeds 1.0. Specifically, we first add small amounts of traffic T'_m between all O-D pairs. Then, we

add a large amount of traffic T_m between a randomly chosen O-D pair m and repeat this procedure until no link utilization exceeds 1.0. T'_m and T_m are uniform random variables. The traffic flows between an O-D pair are divided into the shortest path with p and the quasi-shortest path with $1-p$. Note that we use symmetric traffic with respect to uplink and downlink. This means we suppose that traffic volume from node i to node j is the same as that from nodes j to i. We also assume that the routing probabilities of uplink and downlink traffic between all O-D pairs are identical.

Then, counting the traffic volume of each link, we calculate link utilization ρ_l ($l = 1, \ldots, 52$) on the basis of link capacity preset in advance. We regard all links as M/M/1 queue systems with a First-Come First-Served (FCFS) service, and the cumulative distribution of one-way queueing delay $D_{\mathrm{q}l}$ of link l is $P[D_{\mathrm{q}l} \leq x] = 1 - \exp(-\mu(1-\rho_l)x)$, where $1/\mu$ represents the mean service time.

Moreover, with respect to propagation delay, we set the delay of each link to fixed value. We consider delay of round-trip measurement, that is, a summation of round-trip propagation and queueing delays as $z_{m,n}$, which represents measured value of the flow n between O-D pair m. We assume that the links the flow from a certain node N1 as O node to a certain node N2 as D node pass through are the same as those from N2 to N1 when we measure round-trip delay in the following evaluation. Note that our method is expected to be able to deal with the case when the paths from a certain node N1 to a certain node N2 and those from N2 to N1 are independently selected unless routing probability of the flow from N1 to N2 is different from that of the flow from N2 to N1.

We consider preprocessing of $\boldsymbol{y}$ by using delays in a normal state introduced in Sect. 3.3.1 because we use delays having positive values even if the links are in the normal state. We regard the difference between delays of our estimation target (we call the network that is in the state we want to measure and has the delay we want to estimate the *target network*) and those in a normal state as the degradation vector $\boldsymbol{x}$. We introduce the delay of the *normal network*, which is the delay that occurs when the network is in the normal state. The delay of the normal network also includes propagation and queueing delays; the propagation delay does not change in response to the status of the network, while the queueing delay can be assumed to be negligible small, it is obtained in the same manner as making $\boldsymbol{y}$. The traffic matrix of the normal network is constructed by means of adding a small amount of traffic T''_m between all O-D pairs. The traffic flows between an O-D pair are divided into the shortest and quasi-shortest paths with p and $1-p$, respectively. We measure delays of round-trip measurements between M O-D pairs where the number of the measurements between each O-D pair is N'. In terms of N', we assume a sufficient number of measurement flows in the normal network can more likely be obtained than N in the target network. Let y_m^{normal} be the average of QoS between the m-th O-D pair in the normal network. We construct $\boldsymbol{y}^{\mathrm{normal}}$ and subtract $\boldsymbol{y}^{\mathrm{normal}}$ from $\boldsymbol{y}$.

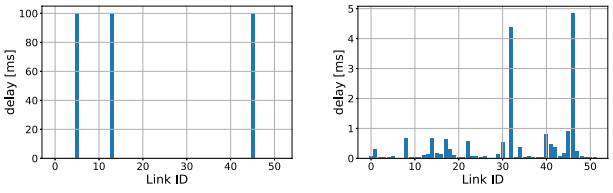

Fig. 7 Examples of link delays in the target network in Sects. 4 (left) and 5 (right).

In the following evaluation, we set $100 \leq T_m \leq 250$ [Mbps], $0.5 \leq T'_m \leq 1$ [Mbps], $0.5 \leq T''_m \leq 1$ [Mbps], and $1/\mu = 0.008$ [ms] supposing that we set the average packet size to 1000 [bytes] and link capacity to 1 [Gbps], and set the propagation delay of all links to 1 [ms] (one way), for simplicity. Examples of the delays used in Sect. 4 and the delays used in this section are shown in Fig. 7. For the estimation, λ was set to 0.01 in this section. We calculate RMSE between the true and estimated values of $\boldsymbol{x}$ for each trial. We conducted 1000 trials.

5.2 Applicability to Practical Delay without Noise

We evaluate whether the proposed method can be applied to the practical network model described in Sect. 5.1. To assess the applicability of PA1 and PA2, we make 10 practical network scenarios by randomly changing T_m and T'_m. We assume here that noise depending on the measurements can be ignored. Thus, in this section, we always obtain a summation of round-trip average delays of the links through which the flows pass as the end-to-end delay $z_{m,n}$. Here the average delay of link is a summation of propagation delay and average M/M/1 queueing delay calculated from ρ of the link l. This corresponds to ignoring sampling noise at the time of measurements. Then, $\boldsymbol{y}$ is ideally obtained as the probability-weighted average of delays of the shortest and quasi-shortest paths, that is,

$$y_m = 2p \sum_{l \in \mathcal{S}_{m,1}} (\mathbb{E}(D_{\mathrm{q}l}) + d_{\mathrm{p}l}) + 2(1-p) \sum_{l \in \mathcal{S}_{m,2}} (\mathbb{E}(D_{\mathrm{q}l}) + d_{\mathrm{p}l}), \quad (5)$$

where $\mathbb{E}(*)$ represents the expectation of $*$, and $\mathbb{E}(D_{\mathrm{q}l}) = 1/(\mu(1-\rho_l))$ from M/M/1 queueing delay. The routing probability matrix $\boldsymbol{A}$ is obtained in the same way in Sect. 4. We consider only the propagation delay in the normal network and regard the queueing delay as 0. For O-D pair m, we obtain y_m^{normal} as follows,

$$y_m^{\mathrm{normal}} = 2\,p \sum_{l \in \mathcal{S}_{m,1}} d_{\mathrm{p}l} + 2\,(1-p) \sum_{l \in \mathcal{S}_{m,2}} d_{\mathrm{p}l}. \quad (6)$$

We subtract $\boldsymbol{y}^{\mathrm{normal}}$ from $\boldsymbol{y}$ as preprocessing. In the following results, we use $p = 2/3$. In calculating RMSE, we use the average queueing delays $\mathbb{E}(D_{\mathrm{q}l})$ of the target network as true values of delays.

From the estimation results of PA2 in 10 scenarios

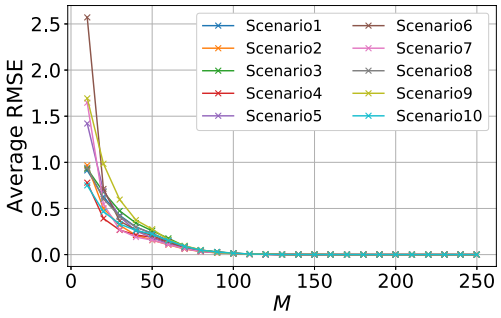

Fig. 8 Average RMSE of PA2 in 10 scenarios.

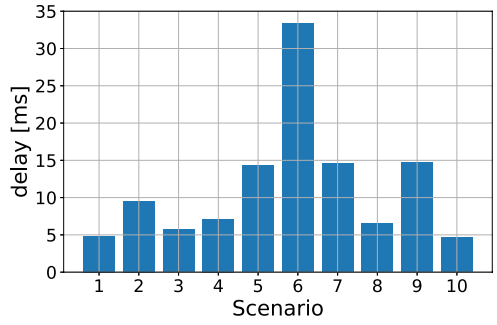

Fig. 9 Max delay of 10 scenarios.

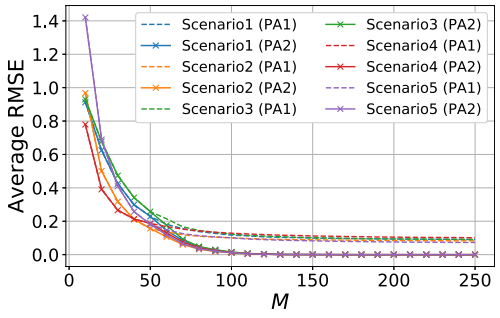

Fig. 10 Average RMSE of PA1 and PA2 in 5 scenarios.

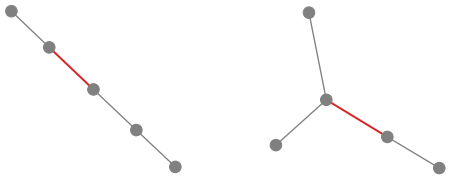

Fig. 11 Examples of topology; the maximum value of node-degrees of nodes on both sides of degraded link (red link) is two (left) and three (right).

Table 2 The maximum value of node-degrees of the nodes on both sides of the link having the maximum delay for each scenario.

Scenario	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Max degree	3	3	5	5	3	4	4	3	4	4

(Fig. 8), although the RMSE in each scenario varies in the range of $M < 50$, the RMSEs in all scenarios converge at similar values in the range in which M is more than about 100. This shows that we can obtain good estimation results if each link is measured by at least one flow where M is more than around 100. Therefore, the proposed method can be said to be applicable to various practical delays that have continuous values. On the other hand, when $M = 10$, the RMSE of each scenario is different from that of the other scenarios because not all links are measured. This difference is related to the difference in the maximum value of link delay in each scenario, which is shown in Fig. 9. From the results including the estimation of PA1 in 5 scenarios (Fig. 10), the RMSEs of PA1 and PA2 are comparable when M is less than about 50. This is because PA2 estimates link delays by CS in this range of M. In contrast, the RMSE of PA2 is lower than that of PA1 when M is more than about 60. This indicates that the increase of M is causing the exact solutions obtained by NNLS, and this behavior is similarly observed in Fig. 4.

We discuss why our method works well. In general, the estimation is presumed to be complicated more by a degraded link being included in successively connected links (shown in Fig. 11 (left)) than by a degraded link having branching (Fig. 11 (right)). It is difficult to identify which link is degraded among the successively connected links because the degraded link and its neighboring links are very likely to be simultaneously included in the path of the measurement flow in the left case. In a randomly superimposed traffic matrix, however, delay of a link connecting to a node having the higher degree (like Fig. 11 (right)) is likely to have a high value. In this evaluation, the maximum value of node-degrees of the nodes on both sides of the link having the maximum delay in any scenarios is actually more than two as shown in Table 2. Note that in Fig. 11 for example, if the red links have maximum delays, the maximum value of node-degrees of nodes on both sides of the maximum delay link is two in the left figure and three in the right one. In all scenarios, the maximum value of the node-degrees being more than two is the reason we can obtain the good results.

Hence, we investigate whether the reduction of the estimated accuracy is caused by a degraded link that has higher delay being included in successively connected links, that is, the maximum value of node-degrees of nodes on both sides of the degraded link is two. Here, we use the delays of Scenario 6 and replace the maximum delay with a delay of the link that connects to nodes whose degree is two. The replacement of link delays is shown in Fig. 12 and Table 3. In Scenario I, nothing is replaced, whereas in Scenarios II, III, and IV, the delay value of the link placed between two-degree nodes is replaced with the delay value of the link having the maximum delay. The results in Fig. 13 show that the RMSEs in Scenarios II, III, and IV, in which the maximum delay is placed between two-degree nodes, are higher than that in Scenario I in the range of smaller M. However, the RMSEs of all scenarios are comparable if we have a sufficient number of M, because the increase in M is considered to cause the increase in the probability that a two-degree node is chosen as an O or D node. Furthermore, six links are indeed successively connected in Scenario II and

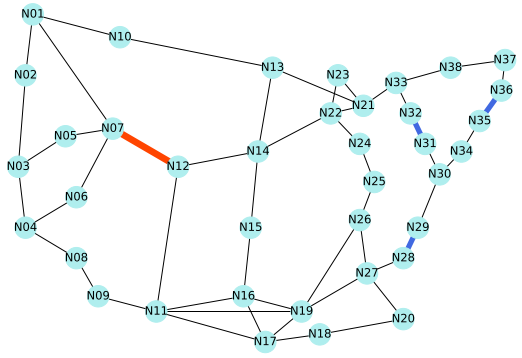

Fig. 12 Replacement links in network topology.

Table 3 The link whose delay is replaced with the maximum delay of the link (N07, N12) and the link having the maximum delay after replacement in each scenario ((Ni, Nj) means the link between nodes Ni and Nj in Fig. 12).

	Replacement links	Max delay link
Scenario I	no replacement	(N07, N12)
Scenario II	(N35, N36) with (N07, N12)	(N35, N36)
Scenario III	(N28, N29) with (N07, N12)	(N28, N29)
Scenario IV	(N31, N32) with (N07, N12)	(N31, N32)

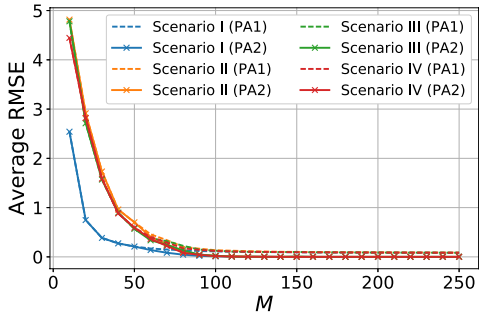

Fig. 13 Average RMSE in replacing max delay link.

three links are in Scenarios III and IV, but the RMSEs in Scenarios II, III, and IV are comparable. This indicates that the number of successively connected links does not affect the RMSE. This is because whether we can accurately estimate the delay of the successively connected links depends on whether only end nodes of the degraded link are chosen as the O or D node, since in these scenarios, neighboring links are not degraded.

From the fact that the maximum delay being placed between two-degree nodes causes low estimation accuracy in the range of $M < 100$, we found where the degraded link is in the topology affects the estimation accuracy. Generally, a link connecting multiple links is likely to become a link having the large delay because of traffic congestion, so we expect to obtain good estimation results. However if the middle link connected in series has the maximum delay like Scenarios II, III, and IV, we have to collect a sufficient number of M.

Table 4 Noise pattern.

Pattern	1	2	3	4
$\boldsymbol{y}$	fixed value	with noise	with noise	with noise
$\boldsymbol{A}$	fixed value	fixed value	with noise	with noise
$\boldsymbol{y}^{\text{normal}}$	propagation	propagation	propagation	propagation & queueing

5.3 Performance Evaluation for Practical Delay with Noise

We assess the noise effect on the estimation accuracy because we cannot practically ignore noise depending on the measurements. For comparison, we use four patterns of conditions shown in Table 4. In Pattern 1, we do not consider noise in any measurements, and it is the same as the conditions in Sect. 5.2. On the other hand, in Patterns 2, 3, and 4, we take noise with regards to measurement of $\boldsymbol{y}$ into account. We regard $N_m = N$, that is, the number of the flows is identical regardless of m. In Pattern 1, we use average M/M/1 queueing delay calculated from ρ as a link delay and obtain $\boldsymbol{y}$ calculated from the ideal probability-weighted average of delays of shortest and quasi-shortest paths. On the other hand, in Patterns 2, 3, and 4, we determine the number of the flows that pass through the shortest path among all N flows in accordance with a binomial distribution $B(N, p)$. If we obtain $K(m)$ as a sample of $B(N, p)$, y_m is described as

$$y_m = 2\,\frac{1}{N}\sum_{i=1}^{K(m)}\sum_{l\in\mathcal{S}_{m,1}}(d_{\mathrm{q}l,i}+d_{\mathrm{p}l}) + 2\,\frac{1}{N}\sum_{i=1}^{N-K(m)}\sum_{l\in\mathcal{S}_{m,2}}(d_{\mathrm{q}l,i}+d_{\mathrm{p}l}), \tag{7}$$

where $d_{\mathrm{q}l,i}$ is a i-th sample of M/M/1 queueing delay distribution for each flow. Thus, the noise of $\boldsymbol{y}$ is related to N.

Moreover, we consider noise with regards to measurement for estimating $\boldsymbol{A}$ in Patterns 3 and 4. Here, we estimate $\boldsymbol{A}$ by counting how many flows pass through the shortest path. We define the number of measurement flows for estimating $\boldsymbol{A}$ as N_A. We determine the number of the measurement flows that pass through the shortest path by means of taking a sample of binomial distribution $B(N_A, p)$. From this operation, if we obtain $K_A(m)$ as the number of the flows through the shortest path between O-D pair m, the $a_{m,l}$, the (m, l)-th element of $\boldsymbol{A}$, is calculated as follows,

$$a_{m,l} = \frac{K_A(m)}{N_A}\,\mathbf{1}_{\mathcal{S}_{m,1}}(l) + \frac{N_A - K_A(m)}{N_A}\,\mathbf{1}_{\mathcal{S}_{m,2}}(l), \tag{8}$$

so the noise of $\boldsymbol{A}$ is related to N_A.

Regarding normal delay, we use propagation and queueing delays in Pattern 4 and only propagation delay in the other patterns. In Patterns 1 and 2, we obtain $\boldsymbol{y}^{\text{normal}}$ in the same way described in Sect. 5.2. In Pattern 3, $\boldsymbol{y}^{\text{normal}}$ is constructed by the measurement results of $\boldsymbol{A}$, and we assume that the propagation delay is known in advance, so

normal delays are made as follows,

$$y_m^{\text{normal}} = 2\frac{K_A(m)}{N_A}\sum_{l\in\mathcal{S}_{m,1}} d_{\text{p}l} + 2\frac{N_A - K_A(m)}{N_A}\sum_{l\in\mathcal{S}_{m,2}} d_{\text{p}l}. \quad (9)$$

Hence, in Pattern 3, the noise of $\boldsymbol{A}$ is caused by N_A measurements, so the noise of normal delay is related to N_A. In Pattern 4, we measure normal delay in the manner described in Sect. 5.1. We obtain $K'(m)$ as the number of the flows on the shortest path by means of taking a sample of the binomial distribution $B(N', p)$, where we measure a normal network with N' flows and assume that the path selection condition that the shortest and quasi-shortest paths are chosen with p and $1-p$, respectively, in the normal network is the same as that in the target network. Under this assumption, we calculate y_m^{normal} from the following.

$$y_m^{\text{normal}} = 2\frac{1}{N'}\sum_{i=1}^{K'(m)}\sum_{l\in\mathcal{S}_{m,1}}(d_{\text{q}l,i}^{\text{normal}} + d_{\text{p}l}) + 2\frac{1}{N'}\sum_{i=1}^{N'-K'(m)}\sum_{l\in\mathcal{S}_{m,2}}(d_{\text{q}l,i}^{\text{normal}} + d_{\text{p}l}), \quad (10)$$

where $d_{\text{q}l,i}^{\text{normal}}$ represents a i-th sample of queueing delay of link l in the normal network, so the noise of $\boldsymbol{y}^{\text{normal}}$ is related to N'.

Note that when we estimate link delays, we subtract $\boldsymbol{y}^{\text{normal}}$ from $\boldsymbol{y}$ as preprocessing. After that, we replace a negative value with 0 if the element of preprocessed $\boldsymbol{y}$ is negative. In calculating RMSE, in contrast, we use the average queueing delay $\mathbb{E}(D_{\text{q}l})$ of target network as true values of delays in Patterns 1, 2, and 3, and we use the difference between the average queueing delay in normal network and that in the target network in Pattern 4.

The estimation results of four patterns are shown in Fig. 14 when $M = 150$, $N_A = 3000$, and $N' = 3000$. The increase of N reduces the RMSEs in Patterns 2, 3, and 4, that is, the effect of noise of $\boldsymbol{y}$ related to N. The results of Pattern 2 approach those of Pattern 1. We find that when $N = 10^5$, the difference between RMSEs of PA2 in Patterns 1 and 2 is around 0.032, and RMSEs of PA1 in Patterns 1 and 2 are comparable. When N is large, the RMSE of PA2 is smaller than that of PA1. In Pattern 2, we can see the smaller RMSE of PA2 when N is more than around 5000, which is a realistic number of measurements.

Next, we show the results in increasing N_A in Fig. 15. These are obtained when $M = 150$, $N = 3000$, and $N' = 3000$. The larger N_A is, the lower the RMSEs of Patterns 3 and 4 are. This is because the noise of $\boldsymbol{A}$ is reduced by the increase of N_A. From the results, if we obtain 10^4 measurements for estimating $\boldsymbol{A}$, RMSEs in Patterns 3 and 4 are almost the same as those in Pattern 2, and we can accurately estimate delays.

Similarly, the results when $M = 150$, $N = 3000$, and

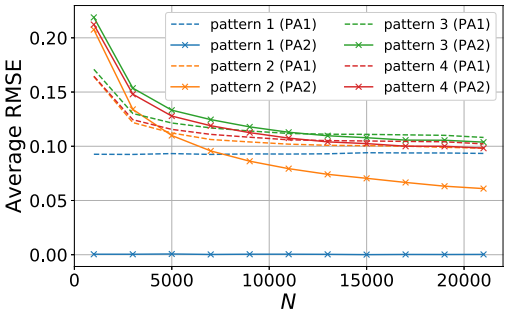

Fig. 14 Average RMSE vs. number of measurement flows between each O-D pair N.

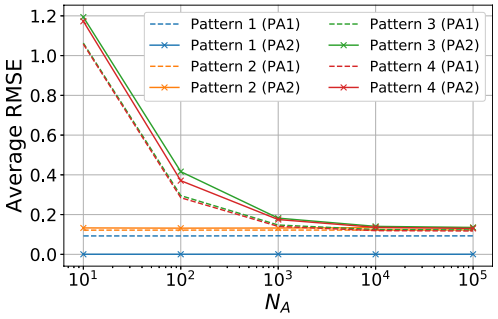

Fig. 15 Average RMSE vs. number of measurements for estimation of $\boldsymbol{A}$, N_A.

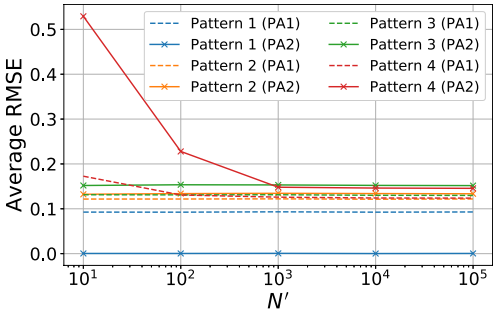

Fig. 16 Average RMSE vs. number of measurements of normal delay N'.

$N_A = 3000$ are shown in Fig. 16. When N' increases, the RMSEs in Pattern 4 become smaller. Owing to the increase of N', the noise effect we obtained in measuring normal delay in Pattern 4 decreases. According to these results, when $N' = 10^4$, the RMSEs of both PA1 and PA2 in Pattern 4 approach those in Pattern 2, respectively. If we obtain $N' = 10^4$, Pattern 4 performs the same as Pattern 2. We can assume normal delay is easily available because the network is in the normal state much of the time, so $N' = 10^4$ is a realistic condition.

6. Conclusion

We proposed a novel network tomography for1 networks with undeterministic routing in which the routing paths cannot be uniquely identified. We evaluated our idea for the

proposed method, the introduction of routing probability, by comparing it with conventional tomography with the determined route information. The results indicated that our method had better estimation accuracy than a conventional one. We found that our method made it possible to identify the performance-degraded components in a network and estimate their degree of degradation with good accuracy even when the routing path of each flow could not be identified. Moreover, we showed that our method is applicable to a practical network in which a link delay has a continuous value and depends on delays of the neighboring links and the measurements are affected by noise.

References

[1] R. Tagyo, D. Ikegami, and R. Kawahara, "Network tomography using routing probability for virtualized network," 2018 IEEE International Conference on Communications (ICC), May 2018.
[2] R. Tagyo, D. Ikegami, and R. Kawahara, "A study for practicality evaluation of network tomography using routing probability," Proc. 2019 IEICE Communications Society Conference, Sept. 2019 (in Japanese).
[3] R.K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," IEEE Commun. Mag., vol.49, no.11, pp.32–39, Nov. 2011.
[4] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," IEEE Commun. Mag., vol.52, no.8, pp.29–35, Aug. 2014.
[5] A. Faggiani, E. Gregori, L. Lenzini, V. Luconi, and A. Vecchio, "Smartphone-based crowdsourcing for network monitoring: Opportunities, challenges, and a case study," IEEE Commun. Mag., vol.52, no.1, pp.106–113, Jan. 2014.
[6] A. Coates, A.O. Hero, III, R. Nowak, and B. Yu, "Internet tomography," IEEE Signal Process. Mag., vol.19, no.3, pp.47–65, May 2002.
[7] E. Dinc, M. Ozger, A.F. Ates, I. Delibalta, and O.B. Akan, "Crowdsourcing-based mobile network tomography for xG wireless systems," 2016 IEEE Symposium on Computers and Communication (ISCC), pp.346–351, June 2016.
[8] R. Kawahara and Y. Tonomura, "Mobile QoS tomography using compressed sensing," 2014 26th International Teletraffic Congress (ITC), 2014.
[9] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," IEEE Commun. Mag., vol.53, no.2, pp.90–97, Feb. 2015.
[10] K. Yap, M. Motiwala, J. Rahe, S. Padgett, M. Holliman, G. Baldus, M. Hines, T. Kim, A. Narayanan, A. Jain, V. Lin, C. Rice, B. Rogan, A. Singh, B. Tanaka, M. Verma, P. Sood, M. Tariq, M. Tierney, D. Trumic, V. Valancius, C. Ying, M. Kallahalla, B. Koley, and A. Vahdat, "Taking the edge off with espresso: Scale, reliability and programmability for global internet peering," Proc. Conference of the ACM Special Interest Group on Data Communication (SIGCOMM), pp.432–445, 2017.
[11] B. Schlinker, H. Kim, T. Cui, E. Katz-Bassett, H. Madhyastha, I. Cunha, J. Quinn, S. Hasan, P. Lapukhov, and H. Zeng, "Engineering egress with edge fabric: Steering oceans of content to the world," Proc. Conference of the ACM Special Interest Group on Data Communication (SIGCOMM), pp.418–431, 2017.
[12] R. Caceres, N.G. Duffield, J. Horowitz, and D.F. Towsley, "Multicast-based inference of network-internal loss characteristics," IEEE Trans. Inf. Theory, vol.45, no.7, pp.2462–2480, Nov. 1999.
[13] M.J. Coates and R.D. Nowak, "Network tomography for internal delay estimation," 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings, vol.6, pp.3409–3412, 2001.
[14] N.G. Duffield, F.L. Presti, V. Paxson, and D. Towsley, "Inferring link loss using striped unicast probes," Proc. IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society, vol.2, pp.915–923, 2001.
[15] M.H. Firooz and S. Roy, "Network tomography via compressed sensing," 2010 IEEE Global Telecommunications Conference (GLOBECOM), 2010.
[16] X. Fan and X. Li, "Network tomography via sparse Bayesian learning," IEEE Commun. Lett., vol.21, no.4, pp.781–784, April 2017.
[17] H. Li, Y. Gao, W. Dong, and C. Chen, "Taming both predictable and unpredictable link failures for network tomography," Proc. ACM Turing 50th Celebration Conference. ACM, pp.35:1–35:10, 2017.
[18] D.Z. Tootaghaj, T. He, and T. La Porta, "Parsimonious tomography: Optimizing cost-identifiability trade-off for probing-based network monitoring," SIGMETRICS Performance Evaluation Review, vol.45, no.3, pp.43–55, 2018.
[19] C. Feng, L. Wang, K. Wu, and J. Wang, "Bound-based network tomography with additive metrics," IEEE INFOCOM, pp.316–324, 2019.
[20] L. Ma, Z. Zhang, and M. Srivatsa, "Neural network tomography," arXiv:2001.02942, 2020.
[21] G. Kakkavas, D. Gkatzioura, V. Karyotis, and S. Papavassiliou, "A review of advanced algebraic approaches enabling network tomography for future network infrastructures," Future Internet, vol.12, no.2, 2020.
[22] V.W. Bandara, A.P. Jayasumana, and R. Whitner, "An adaptive compressive sensing scheme for network tomography based fault localization," 2014 IEEE International Conference on Communications (ICC), pp.1290–1295, June 2014.
[23] J. Chen, X. Qi, and Y. Wang, "An efficient solution to locate sparsely congested links by network tomography," 2014 IEEE International Conference on Communications (ICC), pp.1278–1283, June 2014.
[24] X. Fan and X. Li, "Delay tomography via compressed sensing under exponential distribution," 2017 3rd IEEE International Conference on Computer and Communications (ICCC), pp.2080–2084, 2017.
[25] S. Pan, Y. Zhou, Z. Zhang, S. Yang, F. Qian, and G. Hu, "Identify congested links with network tomography under multipath routing," J. Netw. Syst. Manege., vol.27, no.2, pp.409–429, 2019.
[26] S. Pan, Z. Zhang, F. Yu, and G. Hu, "End-to-end measurement for network tomography under multipath routing," IEEE Commun. Lett., vol.18, no.5, pp.881–884, May 2014.
[27] H. Saito, "Observability and independence in distributed sensing and its application," Proc. 2013 25th International Teletraffic Congress (ITC), 2013.
[28] N.E. Rad, Y. Ephraim, and B.L. Mark, "Delay network tomography using a partially observable bivariate markov chain," IEEE/ACM Trans. Netw., vol.25, no.1, pp.126–138, Feb. 2017.
[29] H. Herodotou, B. Ding, S. Balakrishnan, G. Outhred, and P. Fitter, "Scalable near real-time failure localization of data center networks," Proc. 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp.1689–1698, 2014.
[30] Internet2. [Online]. Available: https://www.internet2.edu/products-services/advanced-networking/layer-2-services/

Rie Tagyo received B.S. and M.S. degrees in physics from Ochanomizu University, Tokyo in 2012 and 2014. Since joining NTT Laboratories in 2014, she has been engaged in research on performance analysis for mobile communication networks at NTT Network Technology Laboratories. She received the Young Engineer Award from IEICE in 2017.

Daisuke Ikegami received B.E., M.E. and Ph.D. degrees from Waseda University, Tokyo in 2002, 2004, and 2007, respectively. Since joining NTT Laboratories, he has been engaged in research on IP service network engineering including network planning of NGN, analysis of TCP performance and QoS estimation technologies. He received the Young Engineer Award of the IEICE in 2010.

Ryoichi Kawahara received his M.E. in automatic control and his Ph.D. in telecommunication engineering from Waseda University, Tokyo, Japan, in 1992 and 2001, respectively. He joined NTT Laboratories in 1992, and was engaged in research on traffic control for telecommunication networks, traffic measurement and analysis for IP networks, and network management, for 26 years. He is currently a professor at the Department of Information Networking for Innovation and Design, Faculty of Information Networking for Innovation and Design, Toyo University. He is a member of IEICE, IEEE, and ORSJ. He received Telecom System Technology Award from The Telecommunications Advancement Foundation in 2010, and Best Paper Awards from IEICE in 2003 and 2009.