IEICE Communications Express, Vol.13, No.1, 5–8

ComEX

LETTER

Link delay estimation under undeterministic routing using neural network

Yuta Ushizuka[1] **and Ryoichi Kawahara**[1, a)]

Abstract Network virtualization allows the provision of various network services and enables flexible control by dynamically changing the path according to the service etc. While conventional network tomography uses path information to estimate the internal network status, such as each link delay, dynamic path changes make it difficult to determine the path that a packet will take. For networks with undeterministic routing, this study proposes a method for estimating the status of each link using a neural network that does not require path information as an input. Instead, it estimates the status of each link using only end-to-end measurements. The neural network is trained using various patterns of individual link statuses as teaching signals on a simulated network where the path changes dynamically. We evaluated the effectiveness of our method through simulations. The results show that the proposed method can identify degraded links with a true positive rate of 98% and false positive rate of 8%.

Keywords: delay, neural networks, network tomography, virtual networks
Classification: Network management/operation

1. Introduction

Virtual network technologies such as network functions virtualization (NFV) and software-defined networking (SDN) have attracted significant attention. Network virtualization allows the provision of various network services and enables flexible control by dynamically changing the path according to the service etc.

Network providers must correctly understand the internal status of a network to manage dynamically changing traffic and maintain the quality of service (QoS). However, the factors affecting the QoS may be complex because of the diversity in services [1]. Therefore, estimating and controlling the internal status of a network are significant technical challenges. Network tomography can be used to estimate the internal status of a network based on end-to-end measurements [2]. Most network tomography techniques combine end-to-end measurements with path information, which describes the links/network components through which end-to-end measurements pass (e.g., [3, 4, 5]).

However, in virtual networks, such as those using SDN, the path varies according to the service and traffic conditions; hence, flows can take different routes, even if the origin–destination node pairs are the same. It is difficult or costly to identify the path through which each flow passes by using a routing table etc. Therefore, it is difficult to apply conventional network tomography techniques that require path information to networks with undeterministic routing, such as virtual networks.

This study presents a network tomography technique that can be used to estimate the internal status of a network without path information. Instead, the proposed method uses a neural network to estimate the status of each link, such as link delay from end-to-end measurements, by simulating various patterns of individual link statuses on a simulated network in which the path changes dynamically. The neural network was trained using data consisting of multiple patterns of individual link statuses and corresponding end-to-end measurement data on the simulated network. In other words, our method uses the simulated link statuses as teaching signals. It was tested by inputting end-to-end measurements and estimating the status of each link. This method does not require path information; therefore, it can be applied to virtual networks in which the path changes dynamically.

In a related work, Tagyo et al. [6] proposed network tomography for undeterministic routing that uses the estimated path information by defining the routing probability based on the measured number of flows recorded at each link. In contrast, our method does not require any path information as an input when estimating the internal network status. Ma et al. [7] proposed neural-network-based tomography. In contrast to our goal, their objective was to infer end-to-end path performance metrics for all unmeasured node pairs when the measured end-to-end path performance metrics are for a subset S of T where T is the complete node pair set.

2. Proposed Method

2.1 Prerequisites

There are several perquisites for the proposed method. First, a simulated network is prepared to simulate the behavior of each link in the network to be estimated, such as link delay. We simulate various patterns of individual link statuses in a simulated network to collect end-to-end measurements resulting from undeterministic routing. (The procedure is explained in detail in the following subsection.) That is, as training data, we collect multiple patterns of individual link statuses as teaching signals and the resulting end-to-end measurements (end-to-end delays) as the input signals for a neural network. Subsequently, the neural network is trained using the training data.

Next, end-to-end measurements are collected from the target network for analysis, and the status of each link is estimated using the trained neural network. The proposed method is intended for use by network administrators who

[1] Graduate School of Information Networking for Innovation and Design, Toyo University, Tokyo, 115-8650, Japan. (This study was conducted when the first author was at Toyo University.)
[a)] ryoichi.kawahara@iniad.org

DOI: 10.23919/comex.2023XBL0132
Received October 12, 2023
Accepted October 26, 2023
Publicized November 10, 2023
Copyedited January 1, 2024

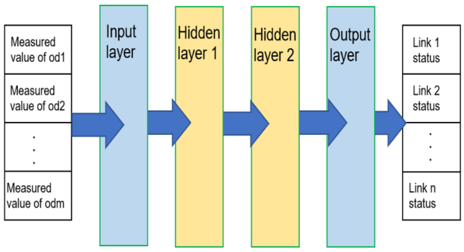

Fig. 1 Schematic diagram of the proposed neural network

need to estimate the internal status of a network; however, owing to its versatility, it may be used by other parties, such as application service providers, who have relatively limited access to network information.

2.2 Proposed neural network

The proposed neural network is illustrated in Fig. 1. It is an all-coupled model, where the input layer has dimension m; there are two hidden layers of dimension s where $s \approx m \times 1.3$ and the activation function is assumed to be sigmoid; and the output layer has dimension n. The input vector consists of the end-to-end measurement of each source-destination node pair, and the output vector consists of each link delay. In this study, we set the number of epochs to 3000 and the batch size to 4 during training. Pair i from source node o to destination node d is labeled od_i. The number of node pairs that perform end-to-end measurements m is the number of node pairs in the network, excluding adjacent node pairs n where n is the number of links. Therefore, if the number of nodes is N_n, $m = {}_{N_n}C_2 - n$. We assume that the network is treated as an undirected graph.

In our method, multipath routing is assumed because we wish to consider a virtual network with undeterministic routing in which the path changes dynamically and flexibly. Specifically, we assume that the probability of passing through the shortest path (in terms of the hop count) in od_i is $p_{1,i}$. Similarly, the probability of passing through the second-shortest path is $p_{2,i}$, where $p_{1,i} + p_{2,i} = 1$. Let $t_{1,i}$ and $t_{2,i}$ denote the end-to-end measurement values obtained for the shortest and second-shortest paths, respectively. Subsequently, the expected value of the end-to-end measurement e_i for od_i is expressed as

$$e_i = t_{1,i} \times p_{1,i} + t_{2,i} \times p_{2,i}. \tag{1}$$

In this study, we estimated link delay. Let x_j denote the delay in link j. The end-to-end measurement on the shortest path $t_{1,i}$ of od_i is the sum of the delays of all the links on the path and is expressed as

$$t_{1,i} = \sum_{j \in S_{1,i}} x_j, \tag{2}$$

where $S_{1,i}$ denotes the set of links along the shortest path of od_i. Moreover, $t_{2,i}$ can be defined similarly.

The input vector for the neural network is defined as $E = [e_1, e_2, \ldots, e_m]^T$. The output vector is defined as $L = [x_1, x_2, \ldots, x_n]^T$. The neural network is trained using end-to-end measurements and the status of each link. Specifically, the status of each link x_j is given and the end-to-end measurements are calculated using Eqs. (1) and (2), respectively. The model is trained sufficiently to predict the status of each link from the given end-to-end measurements during the testing phase.

3. Simulation experiment

The network topology used in this study was created using Python and was based on the topology in [8], as shown in Fig. 2. In the experiment, random numbers were used as the status of each link, that is, link delay.

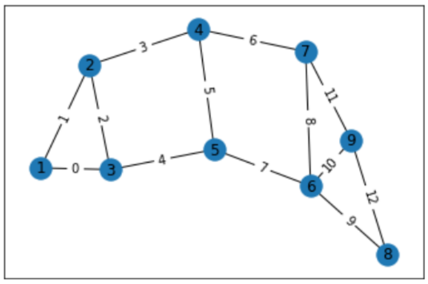

Fig. 2 Topology of the target network

3.1 Experimental setup

The experimental data consisted of end-to-end measurements and link status data. The procedure used to generate the data is as follows:

Assign a value to each link as its status: First, a delay value is assigned to each link. The delay value was randomly generated with an exponential distribution of the average value of 10 ms and rounded to two decimal places.

End-to-end measurement: Once the status of each link was assigned, end-to-end measurements were calculated using Eqs. (1) and (2), respectively. First, routes with the shortest and second-shortest paths (in terms of the hop count) were selected. If multiple routes have the same number of hops, the node with the smallest number is used to determine the shortest path. We set $p_{1,i} = 2/3$ and $p_{2,i} = 1/3$ in Eq. (1).

The number of node pairs that perform the end-to-end measurements m is given by $m = {}_9C_2 - 13 = 23$. The dimension s of the middle layer of the neural network was set to be 30 for both the layers.

Generating multiple patterns of data: The set of data containing the status of each link and the end-to-end measurements generated by the above procedure was considered as one pattern, and 1100 patterns were prepared. Of the 1100 patterns, 900 were used as training data, 100 as evaluation data, and 100 as test data.

During the training phase, the status of each link was used as the teaching signal, and the neural network was trained to estimate the status of each link from end-to-end measurements. During the test phase, we evaluated the ability of the neural network to estimate the status of each link using unknown end-to-end measurements.

3.2 Experimental items

Two experiments were conducted. These results are presented in the following subsections.

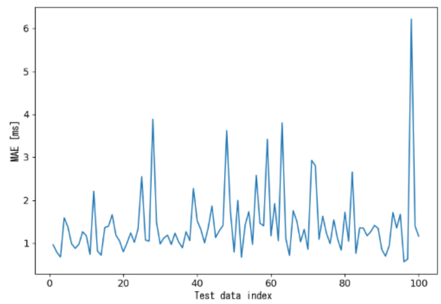

Fig. 3 Estimation accuracy for Exp. 1

Table I Exp. 1: MAE (in ms) of the estimated and true values

	Mean	Median	Max.	Min.
MAE	1.44	1.24	6.22	0.56

Exp. 1: Delay estimation: The delay is estimated, and the accuracy of the estimation is evaluated.

Exp. 2: Degraded-link identification: The degraded links are identified experimentally. The test data consisted of end-to-end measurement data including a link delay of 100 ms for one of the 13 links, representing a performance-degraded link. Ten patterns were prepared for each link, which resulted in 130 sets of test datasets. The trained model was the same as that used for the delay estimation. In other words, no degraded links were included in the training dataset. A threshold was defined, and if the estimation result of a link was equal to or larger than the threshold, the link was considered degraded. The results are evaluated to determine whether the location of the degraded link can be identified.

3.3 Evaluation Results

3.3.1 Exp. 1: Delay estimation

Using 100 test data patterns, we evaluated the mean absolute error (MAE) of the estimated and true values of 13 links for each pattern. The estimation results of the test data are shown in Fig. 3, where the y-axis represents the MAE and the x-axis represents the index of the test data. The mean, median, maximum, and minimum values of the MAE for 100 patterns of the test data are listed in Table I. Figure 4 shows the scatter plots of the test data when the MAE was approximately equal to the median value (upper graph) and when it was at its maximum value (lower graph).

At the maximum MAE, the standard deviation of the delay on each link in the test data was 22.86 ms. In contrast, at the minimum MAE, the standard deviation of the delay on each link was 3.51 ms. This indicates that the accuracy decreased when there was a variation among the links. However, from Fig. 3 and the fact that the average MAE was 1.44 ms compared to the average link delay of 10 ms, we can conclude that the proposed method can be used to estimate the delay.

3.3.2 Exp. 2: Degraded link identification

We conducted an experiment to identify the degraded links. A threshold was defined, and links were considered degraded if the estimated delay of the link was greater than or equal to the threshold. The *average* of the estimated results for the 13 links was used as the initial threshold value. We then evaluated the accuracy of the degraded link identification when the threshold was increased by $average \times 0.1$ from $average \times 1.0$ to $average \times 4.0$. If the estimated value of a link was greater than or equal to the threshold, and the true value of the link was less than 100 ms, it was considered a false positive. If the estimated value was below the threshold and the true value was greater than or equal to 100 ms, it was considered a false negative. If the estimated value was greater than or equal to the threshold, and the true value was greater than or equal to 100 ms, it was considered a true positive. Finally, if the estimated value was below the threshold and the true value was less than 100 ms, it was considered a true negative. The false positive rate and true positive rates (FPR and TPR, respectively) were calculated. The receiver operating characteristic (ROC) curve was then plotted for thresholds up to $average \times 4.0$, as shown in Fig. 5 (where coordinates (1,1) and the graph are connected). The x- and y-axes indicate FPR and TPR, respectively.

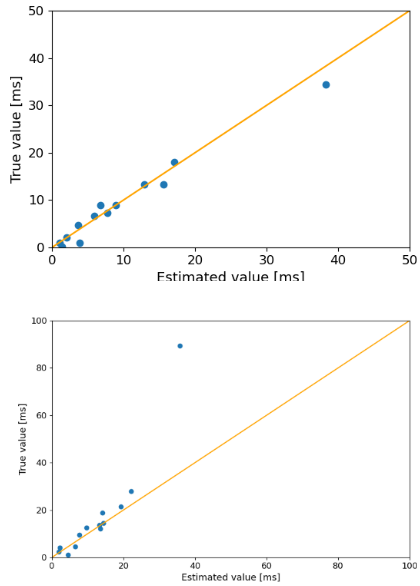

Fig. 4 Scatterplot in Exp. 1: upper: median, lower: max

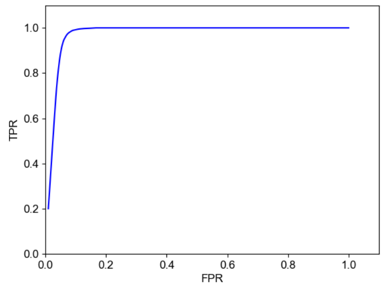

Fig. 5 ROC curve when identifying degraded links

When the threshold was $average \times 1.0$, the FPR and TPR were 0.263 and 1.0, respectively. When the threshold was $average \times 3.0$, the FPR and TPR were 0.079 and 0.980, respectively. The area under the curve (AUC) was calculated as 0.97. Therefore, we conclude that the proposed method can successfully identify degraded links.

4. Conclusion

It is difficult to estimate the internal status of a network using conventional network tomography because it is difficult to determine the path information of end-to-end measurements in a virtual network environment. Therefore, we used a neural network to address this problem, and the proposed network tomography method estimated the status of each link from end-to-end measurements without knowing the path information. This was achieved by collecting end-to-end measurements and the status of each link using a simulation, and using these data to train the model. We evaluated whether the status of each link could be estimated from the end-to-end measurements at the time of estimation. The results show that the proposed method is an effective means of estimating each link delay and identifying degraded links.

Acknowledgments

We would like to thank Editage (www.editage.jp) for English language editing.

References

[1] R. Kawahara, K. Watanabe, S. Harada, and T. Kawata, "Application of AI to network operation," *IEICE Commun. Soc. Global Newsletter*, vol. 44, no. 2, June 2020.

[2] 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. DOI: 10.1109/79.998081

[3] T. Matsuda, M. Nagahara, and K. Hayashi, "Link quality classifier with compressed sensing based on $l_1 - l_2$ optimization," *IEEE Commun. Lett.*, vol. 15, no. 10, pp. 1117–1119, Oct. 2011. DOI: 10.1109/lcomm.2011.082911.111611

[4] V.W. Bandara, A.P. Jayasumana, and R. Whitner, "An adaptive compressive sensing scheme for network tomography based fault localization," IEEE ICC 2014, pp. 1290–1295, June 2014. DOI: 10.1109/icc.2014.6883499

[5] R. Kawahara and Y. Tonomura, "Mobile QoS tomography using compressed sensing," 2014 26th International Teletraffic Congress (ITC), Sept. 2014. DOI: 10.1109/itc.2014.6932955

[6] R. Tagyo, D. Ikegami, and R. Kawahara, "Network tomography using routing probability for undeterministic routing," *IEICE Trans. Commun.*, vol. E104-B, no. 7, pp. 837–848, July 2021. DOI: 10.1587/transcom.2020ebp3149

[7] L. Ma, Z. Zhang, and M. Srivatsa, "Neural network tomography," arXiv preprint, arXiv:2001.02942v1 [cs.NI], Jan. 2020.

[8] A. Suzuki, R. Kawahara, M. Kobayashi, S. Harada, Y. Takahashi, and K. Ishibashi, "Extendable NFV-integrated control method using reinforcement learning," *IEICE Trans. Commun.*, vol. E103-B, no. 8, pp. 826–841, Aug. 2020. DOI: 10.1587/transcom.2019ebp3114