IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 22, 2020, accepted October 2, 2020, date of publication October 23, 2020, date of current version November 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3033331

Network Topology Inference Based on Subset Structure Fusion

JIAN YE, GAOLEI FEI, XUEMENG ZHAI, AND GUANGMIN HU
School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Corresponding author: Gaolei Fei (fgl@uestc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62071095, and in part by the Sichuan Science and Technology Program under Grant 2019YFG0456.

ABSTRACT Network topology measurement is an important component in network research. Network tomography is able to accurately infer network topology by using end-to-end measurement without cooperation of internal routers. Unfortunately, traditional network tomography methods can not accurately estimate topology in the non-stationary network due to the variability of traffic distribution. In this paper, we present a novel network topology inference method based on subset structure fusion for accurate topology inference in the non-stationary network. First, we propose an end-to-end measurement method named three-packet to accurately probe the three-leaf-nodes subset structures of the network without the assumption that the packet delay or loss follows a stable distribution. Second, we propose a metric for the shared path length based on the structural characteristics of the subset structures to fuse these subset structures into a correct complete topology. The analytical and simulation results show that our method is more applicable for topology inference in the non-stationary network compared with the existing methods.

INDEX TERMS End-to-end measurement, network tomography, non-stationary network, subset structure fusion, topology inference.

I. INTRODUCTION

The rapid development of the network makes it increasingly difficult to manage the network. The network topology, which is the foundation of network management, can precisely depict the connection relationship between network device nodes [1]. Grasping the accurate network topology helps us to manage the network more effectively. For example, by identifying and analyzing the weak parts of the network topology, we can efficiently optimize the network structure, reduce network congestion, and prevent hackers from attacking the vulnerable parts of the network.

There are two primary network topology measurement methods, one based on internal node cooperation and another based on network tomography. The method based on internal node cooperation can quickly and accurately estimate the topology by using the feedback routing information returned from the internal routers [2]–[4]. However, this method may fail to estimate the topology when the internal routers refuse to reply to the topology information because it probably causes security issues and the probe packets may be filtered by the firewall. The method based on network tomography [5], [6] (also known as network topology inference method) infers topology by using the path performance parameters obtained from end-to-end measurement. Compared with the method based on internal node cooperation, network topology inference method does not require extra cooperation from the internal nodes and is more feasible. In addition, network tomography can also be used to study more network internal performance parameters [7]–[9], such as packet delay [10], packet loss [11], and bandwidth [12].

Existing network topology inference methods generally use well-designed end-to-end measurements such as back-to-back packet [13] and "sandwich" packet [14] to obtain the path performance parameters of the network. These methods use the metrics obtained by calculating the path performance parameters to measure the length of the shared path from a single source to all destination nodes pairs, and construct the network topology based on the relative size between the metrics. However, in the non-stationary network, existing network topology inference methods can not obtain accurate metrics for shared path length to recover the correct topology. The reason why existing methods are not applicable is that the path performance parameters such as packet delay or loss in

The associate editor coordinating the review of this manuscript and approving it for publication was Haipeng Yao.

the non-stationary network do not follow a stable distribution, whereas existing methods assume that the packet delays or loss of different probes on the same link follow the identical distribution.

We propose a network topology inference method based on subset structure fusion to estimate the network topology from a source node to a set of destination nodes in the non-stationary network. We first measure and identify the three-leaf-nodes subset structures of the topology via three-packet end-to-end measurement. Then we fuse these subset structures to build a complete network topology. The main contributions of this paper are summarized in the following three aspects:

- First, we propose an end-to-end measurement named three-packet to probe the three-leaf-nodes subset structures in the whole network topology. In this way, the topology inference problem is decomposed into multiple simple sub-problems that are the inference of subset structures containing three leaf nodes. Three small probe packets with short time intervals are sent to obtain such three-leaf-nodes subset structures. The three-packet sends less probe packet than back-to-back packet and can accurately measure the shared path length of subset structures in the non-stationary network. Therefore, the three-packet can effectively reduce the number of probe packets and also accurately probe the subset structures, which is benefit to recover the complete network topology.
- Second, we propose a binary tree topology inference method based on subset structure fusion to aggregate the separated subset structures into a complete binary tree network topology. A metric for shared path length based on the structural characteristics of the subset structures is proposed to fuse the subset structures into a binary tree topology. Based on the accurate measurement of subset structures via three-packet, this metric can precisely measure the shared path length of the topology in the non-stationary network and is conductive to estimate a correct binary topology.
- Third, based on the metric above, we propose a general tree topology inference method that deletes false links by setting different thresholds for different subset structures. The subset structures are determined whether they are general tree-like or binary tree-like with the given thresholds of themselves. Then the false links (the links that do not exist in the real network topology) are deleted if there are more general tree-like subset structures contain these links than binary tree-like subset structures. Therefore, different thresholds are more feasible for the complex traffic distribution (such as traffic unbalance and non-stationary) in practical application, which greatly improve the accuracy of topology inference.

Our approach is more applicable to topology inference in the non-stationary network because we can accurately measure the shared path length of the subset structures via three-packet end-to-end measurement. The remainder of the paper is organized as follows. In Section II, we review the related works. In Section III, we introduce related models and concepts. In Section IV, we give the three-packet end-to-end measurement and the topology inference method. In Section V, we describe the evaluations of the method under NS2 simulation. We finally conclude this paper in Section VI.

II. RELATED WORKS

There are already lots of network topology inference methods. Ratnasamy and McCanne [15] took the lead to study network topology using multicast network tomography. They obtained the path-level packet loss rate via end-to-end measurement and calculated the link-level packet loss rate of the shared path. Based on the work of Ratnasamy *et al.*, Duffield *et al.* [16], [17] mapped the packet loss rate of the shared path to a metric that could measure the length of the shared path. In addition, they [18] also applied the path delay covariance as a new metric to infer the topology. Network tomography methods are often based on some assumptions, but when some of the assumptions fail, the accuracy of topology inference will be greatly reduced. In order to solve this problem, Nguyen and Zheng [19] proposed the sequential binary independent component analysis algorithm to precisely estimate the network topology under the assumption of independent failure of intermediate nodes or links. Recently, Bowden and Veitch [20] proposed the shared loss topology discovery algorithm capable of returning the correct network topology under the assumption that the link processes were not completely independent.

Due to network security and other reasons, most of the existing networks do not support multicast packet routing now, so the main research focuses on the topology inference method based on unicast probe. Castro and Nowak [13] proposed a method for unicast end-to-end measurement based on ‘‘sandwich’’ packet and used the agglomerative likelihood tree algorithm to infer binary trees. In order to infer the general tree topology, Castro *et al.* [21] also proposed a Markov Chain Monte Carlo algorithm, which used the maximum likelihood method to select the maximum likelihood tree from the candidate trees. Based on the works of Castro *et al.*, Shih and Hero [22] proposed a hierarchical clustering topology inference algorithm, which used graph clustering and unsupervised learning of a finite mixture model to recursively partition the destination node from top to bottom. In order to reduce the number of probe packets, Eriksson *et al.* [23], [24] proposed a probe method based on Deep-First Search (DFS) and used Ordered Logical Topology Discovery (OLTD) algorithm to infer network topology. Pepe and Puleri [25] also proposed a method to find the smallest end-to-end measurement set to automatically determine the smallest set of paths to probe, reducing the number of probe packets.

In addition, the fusion of multi-source information can also improve the accuracy of topology inference, Ni *et al.* [26] proposed a framework that integrated multiple metrics, which could integrate the metrics of shared path length obtained by measurement methods such as multicast, unicast, or traceroute. Malekzadeh and MacGregor [27] proposed a new probe scheme named traceroute with sandwich probe based on end-to-end unicast delay measurement, which combined the delay-based "sandwich" packet unicast probe model and traceroute. Fei *et al.* [28] use the delay cumulants of second-order and above to infer the topology, so the statistical information of the path delays can be more fully utilized. Although existing network topology tomography methods is relatively mature, it is suitable for small networks. For this reason, Santos *et al.* [29] proposed a tomography method based on the divide-and-conquer to effectively and accurately probe large networks. To infer more general network topology, Rai and Modiano [30] proposed two statistical generic methods expectation-maximization and evolutionary sampling for inference of additive metrics using unicast probing. Rahali *et al.* [31] used path interference to identify general topology which outperformed the algorithm that used distance measurements.

Related works often assume that that the delays or loss of different probes on the same link are identically distributed. Therefore, these methods are not applicable in the non-stationary network because the path performance parameters in the non-stationary network does not follow a stable distribution. The method we proposed is able to accurately measure the shared path length of the subset structures and recover the complete topology by fusing the subset structures in the non-stationary network.

III. PROBLEM STATEMENT

A. NETWORK MODEL

In this paper, we aim at the problem of how to infer the logical routing topology from a source node to a set of destination nodes in a non-stationary network, where all internal router nodes do not cooperate with each other and refuse to reply to any information about the topology. As in most literature [16]–[21], the network topology mentioned in this paper refers to a logical tree. We use $T = (V, E)$ to represent a tree-like logical topology, where $V = \{v_1, v_2, \cdots, v_n\}$ represents the set of physical nodes and E represents the set of links between nodes. V consists of a root node s, a series of internal nodes W (path branching nodes), and the leaf nodes D. The root node s is the source of our measurement and the leaf nodes D are the destinations. Except for the source node, each node $v \in W \cup D$ has a unique parent node $f(v)$. For a pair of destination nodes $\{i, j\}$, we use $f(i, j)$ to represent their nearest parent node. The path from the source node s to the parent node $f(i, j)$ is called the shared path of the destination node pair $\{i, j\}$. The path from node s to a destination node i is represented as $p(s, i)$, which is composed of a series of links e. We use $T_{sub} = (V_{sub}, E_{sub})$ to represent a subset structure of T, where $V_{sub} \in V$ and $E_{sub} \in E$. In addition, the root node of T_{sub} is s and the leaf node set $D_{sub} \in D$.

B. TOPOLOGY INFERENCE PRINCIPLE

Topology inference methods use end-to-end measurement to probe the network since it is not cooperative. Traditional end-to-end measurement methods probe the network via "sandwich" packet or back-to-back packet. These methods probe a pair of destination nodes at one time. For the convenience of description, we abstract a topology include a source node and a pair of destination nodes as Fig. 1. $link_1$ represents the shared path of the destination node pair. $link_2$ and $link_3$ represent the link between the leaf node and the nearest common ancestor node.

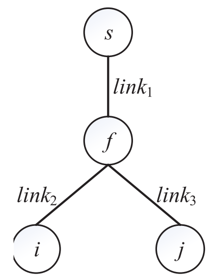

FIGURE 1. Simplified binary tree network model.

The end-to-end measurement based on the "sandwich" packet [14] selects a leaf node pair $\{i, j\}$ as the destinations to send three probe packets $< i_1, j_1, i_2 >$ with a small interval from the source node. The more routers the probe packet passes through, the greater the queuing delay. Therefore, the greater the difference between the delay values of the two probe packets received at the leaf node i, the longer the shared path length of the leaf node pair $\{i, j\}$. Although the back-to-back packet [13] also selects a leaf node pair $\{i, j\}$ as the destinations, it sends only two packets from the source node in a short interval. The transmission interval between the back-to-back packet is small enough, so it can be considered that the behavior (such as transmission, alignment, and discarding) of the probe packets on the shared path $link_1$ is almost the same. Thus, we can obtain the packet loss rate and delay variance on the shared path by calculating the end-to-end packet loss rate and delay variance. The higher the probability of packet loss or the greater the variance of delay, the longer the shared path length of $\{i, j\}$.

The end-to-end information collected by the "sandwich" packet or the back-to-back packet cannot be directly used to infer the network topology, so further calculations are needed to obtain the metric for the shared path length. To ensure that the metric for shared path length obtained by end-to-end measurement can accurately infer the topology, we make the following assumptions:

Assumption 1 (Structure stability): The network topology remains unchanged during the end-to-end measurement.

Assumption 2 (Spatial independence): The packet delays or loss on different links are statistically independent.

Assumption 3 (Temporal independence): The packet delays or loss of different probes on the same link are statistically independent.

Based on the above assumptions, the topology can be recovered by comparing the metric for the shared path length. Fig. 2 shows a shared path length model. f_1 represents the nearest common ancestor node of the leaf node pair $\{i, k\}$, and $p(s, f_1)$ is the shared path from the source node s to the leaf node pair $\{i, k\}$. f_2 represents the nearest common ancestor node of the leaf node pair $\{i, j\}$, and $p(s, f_2)$ is the shared path from the source node s to the leaf node pair $\{i, j\}$. From Fig. 2, we can see that the shared path $p(s, f_2)$ has more links than the shared path $p(s, f_1)$, so the metric for $p(s, f_2)$ is larger than the metric for $p(s, f_1)$ and f_2 is far away from s comparing with f_1.

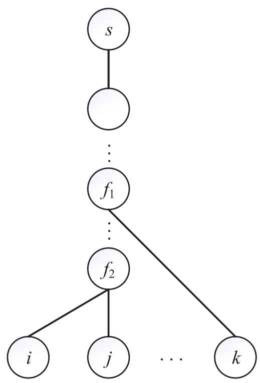

FIGURE 2. **Description of the shared path length.**

Therefore, the topology inference principle is: the greater the metric for the leaf node pair's shared path is, the farther the parent node is from the source node. We can insert the leaf nodes one-by-one based on the shared path length metric to recover the tree topology.

C. METRIC FOR SHARED PATH LENGTH

The key for topology inference is to obtain accurate metrics for the shared path length. One of the most common metrics is the delay covariance. Taking probing topology in Fig. 1 as an example, we send back-to-back packet to probe node pair $\{i, j\}$. Let Y_i and Y_j be the packet delay on the paths from s to i and j. Let X_1 and X_2 be the delay of the packet send to i on $link_1$ and $link_2$, and let X_3 and X_4 be the delay of the packet send to j on $link_3$ and $link_1$. According to the additive of packet delay, we have

$$\begin{aligned} Y_i &= X_1 + X_2 \\ Y_j &= X_3 + X_4. \end{aligned} \tag{1}$$

According to the characteristics of the back-to-back packet, we know that the two packets of a back-to-back packet experience the same delay on the shared path except for the queuing delay of the last packet caused by the previous packet. The queuing delay is the same for each probe. Therefore, we rewrite (1) as:

$$\begin{aligned} Y_i &= X_1 + X_2 \\ Y_j &= X_1 + X_3 + Z, \end{aligned} \tag{2}$$

where Z is the queuing delay.

Calculating the variance of Y_i and Y_j, we have

$$\begin{aligned} Var(Y_i) &= Var(X_1) + Var(X_2) \\ Var(Y_j) &= Var(X_1) + Var(X_3). \end{aligned} \tag{3}$$

By adding the two equations in (2) and then calculating the variance, we obtain

$$Var(S_{i,j}) = 4Var(X_1) + Var(X_2) + Var(X_3), \tag{4}$$

where $S_{i,j}$ represents the sum of path delay Y_i and Y_j.

The covariance of Y_i and Y_j is

$$Cov(Y_i, Y_j) = \frac{Var(S_{i,j}) - Var(Y_i) - Var(Y_j)}{2}. \tag{5}$$

By combining (3), (4), and (5), we obtain the variances of $link_1$, $link_2$, and $link_3$ as following:

$$\begin{aligned} Var(X_1) &= Cov(Y_i, Y_j) \\ Var(X_2) &= Var(Y_i) - Cov(Y_i, Y_j) \\ Var(X_3) &= Var(Y_j) - Cov(Y_i, Y_j). \end{aligned} \tag{6}$$

From (6), we can see that the variance of the shared path $link_1$ is equal to the path delay covariance $Cov(Y_i, Y_j)$, so we can get the variance of the shared path by calculating the covariance of the end-to-end delay.

IV. TOPOLOGY MEASUREMENT AND INFERENCE

In order to overcome the inapplicability of existing methods for topology inference in the non-stationary network, we decompose the topology inference into a series of smallest sub-problems on the basis of divide-and-conquer. The smallest sub-problem of topology inference is the measurement and inference of the subset structure T_{sub} containing only three leaf nodes, this is because the subset structure containing only two leaf nodes is fixed, and the entire topology cannot be recovered by fusing these subset structures. For the convenience of description, if there is no special description in this paper, the subset structures we mentioned all refer to the subset structures T_{sub} containing only three leaf nodes.

In this section, we will introduce our work from five aspects. Firstly, we propose an end-to-end measurement named three-packet to probe the subset structures. Secondly, we infer the subset structures by comparing the shared path length metrics based on the delay covariance. Thirdly, we propose a metric for the shared path length based on the structural characteristics of the subset structure and merge all the subset structures to recover the entire binary tree topology. Fourthly, we modify the binary tree topology to obtain the estimation of general tree topology by setting different thresholds for different subset structures. Finally, we demonstrate why the back-to-back packet is not applicable for topology inference in the non-stationary network.

A. THREE-PACKET END-TO-END MEASUREMENT

We first propose an end-to-end measurement named three-packet. The three-packet is applied to the tree topology with only three leaf nodes, as shown in Fig. 3. The three-packet

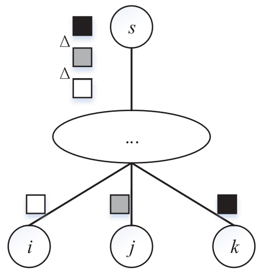

FIGURE 3. Three-packet end-to-end measurement.

respectively sends a probe packet of the same size from the source node to the three destination nodes. The sending interval between adjacent probe packets is very small. In order to reduce the network burden, the size of the probe packet is set to very small. We probe all the destination nodes at once instead of three times like back-to-back packet. Although the three-packet sends one more packet each time than the back-to-back packet, the time interval between the three packets is still very small, so it can also be considered that the three probe packets are in the same network environment on the shared path and have the same behavior (discarded, transmitted, or queued). Therefore, any two of the probe packets form a back-to-back packet, so we can use the method in Section III-C to obtain the delay covariance of the shared path and the delay covariance of different probe on the same shared path are the same.

Three-packet end-to-end measurement has many advantages compared with the back-to-back packet. On the one hand, the three-packet reduces the number of probe packets. For a subset structure with only three leaf nodes, the back-to-back packet needs to send 6 packets in a probe cycle, whereas the three-packet only needs to send 3. So three-packet reduces the number of probe packets by half. On the other hand, for the three-packet, the delay covariance of different probe on the same shared path are the same, whereas, for the back-to-back packet, the delay covariance of different probe on the same shared path are different in the non-stationary network. Therefore, the three-packet obtains accurate metrics for shared path length and is more suitable for end-to-end measurement in the non-stationary network.

B. SUBSET STRUCTURE INFERENCE

In this section, we aim at the problem of how to infer the subset structure based on the precise end-to-end measurement via three-packet. We first calculate the packet delay covariance and then we get the unique binary subset structure by comparing the packet delay covariance.

There are only four kinds of subset structures, as shown in Fig. 4. We can see that different subset structures correspond to the only longest shared path except Fig. 4(d). For example, if the delay covariance of the node pair $\{i, j\}$ is the largest, their shared path is the longest, so the corresponding subset structure is Fig. 4(a). Similarly, the subset structure Fig. 4(b) corresponds to the node pair $\{i, k\}$ has the longest shared path and the subset structure Fig. 4(c) corresponds to the node pair $\{j, k\}$ has the longest shared path. Although the subset structure Fig. 4(d) is also a subset structure with only three leaf nodes, we first recognize it as a binary subset structure like Fig. 4(a), Fig. 4(b), or Fig. 4(c). Then we delete the link which smaller than a given threshold to convert the binary subset structure into the subset structure shown in Fig. 4(d). We will discuss it in detail in the general tree topology inference in Section IV-D.

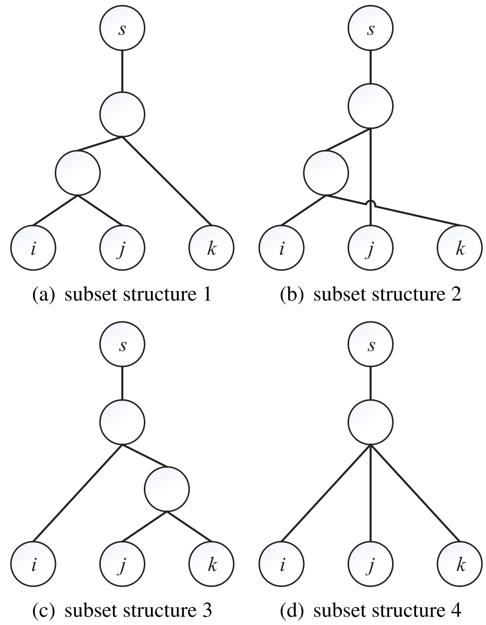

FIGURE 4. The subset structures with three leaf nodes.

Therefore, the subset structure inference principle is: different subset structures correspond to the only longest shared path.

Based on the subset structure inference principle, we infer all the subset structures of the topology by comparing the packet delay covariance. Then we can obtain a binary subset structure set. This subset structure set will be the input for subset structure fusion.

C. SUBSET STRUCTURE FUSION

In this section, we first propose a new metric for shared path length by analyzing the structural characteristics of the subset structures. Then we use this metric to fuse the subset structures to recover the complete binary tree topology.

Observing Fig. 4, we can find that the shared path length of the leaf node pair $\{i, j\}$ is the longest in Fig. 4(a), the shared path length of the leaf node pair $\{i, k\}$ is the longest in Fig. 4(b), and the shared path length of the leaf node pair $\{j, k\}$ is the longest in Fig. 4(c). We use this structural characteristic of the binary subset structures to construct the metric for the shared path length of the binary tree topology.

A binary tree network T with $M(M \geq 3)$ leaf nodes has $\binom{M}{3}$ different subset structures containing only three leaf nodes. Each subset structure is one of the first three structures shown in Fig. 4. The binary tree T has $\binom{M}{2}$ different leaf node pairs, and for each leaf node pair $\{i, j\}$, there are $M - 2$ different subset structures $\hat{T}_{sub}$ containing this leaf node pair. The leaf nodes of these $M-2$ subset structures are i, j, and k. The parent node of $\{i, j\}$ is f and f has N descendant leaf nodes. For a subset structure $\hat{T}_{sub}$, if k is the descendant node of f, we can find that the leaf node pair $\{i, j\}$ does not have the longest shared path. If k is not the descendant node of f, the leaf node pair $\{i, j\}$ has the longest shared path. Therefore, we set the metric for the shared path length of the leaf node pair $\{i, j\}$ to $M - N$.

More generally, for a subpath $p(w_0, w_n)$ in the shared path of the leaf node pair $\{i, j\}$. $w_0, w_1, \cdots$, and w_n are the nodes in this path, where $n > 0$. Every w_i has N_i descendant leaf nodes, $i \in \{0, 1, \cdots, n\}$. Let T_s is the subtree of T and the root node of T_s is node w_0. So the path $p(w_0, w_n)$ is the shared path of $\{i, j\}$ in the subtree T_s. Therefore, the metric for path $p(w_0, w_n)$ is $N_0 - N_n$.

The shared path length metric ρ can be used to recover the topology if it is an addictive metric [31]. So it needs to meet the following two conditions:

Condition 1: $0 < \rho < \infty$;

Condition 2: $\rho(i, j) = \sum \rho(e), \forall i, j \in V$, where $\rho(e)$ represent the length of link e in path $p(i, j)$ and $\rho(i, j)$ represent the distance between node i and node j.

The metric for the shared path length based on the structural characteristics of the subset structures is a positive number that satisfies *Condition 1*. This metric also satisfies the *Condition 2* and we will prove this by mathematical induction.

Lemma 1: For a shared path $p(w_0, w_n)$. $w_0, w_1, \cdots$, and w_n are the nodes in this path, where $n > 0$. Every w_i has N_i descendant leaf nodes, $i \in \{0, 1, \cdots, n\}$. The metric for the shared path length based on the structural characteristics of the subset structures satisfies $\rho(w_0, w_n) = \rho(w_0, w_1) + \rho(w_1, w_2) + \cdots + \rho(w_{n-1}, w_n)$.

Proof:

1) For $n = 1$, $\rho(w_0, w_1) = \rho(w_0, w_1)$.
2) For $n = k$, we have $\rho(w_0, w_k) = \rho(w_0, w_1) + \rho(w_1, w_2) + \cdots + \rho(w_{k-1}, w_k) = N_0 - N_k$. So when $n = k + 1$, we have $\rho(w_k, w_{k+1}) = N_k - N_{k+1}$ and $\rho(w_0, w_{k+1}) = N_0 - N_{k+1}$, so we can get $\rho(s, w_{k+1}) = \rho(w_0, w_k) + \rho(w_k, w_{k+1}) = N_0 - N_{k+1}$.
3) In sumarry, $\rho(w_0, w_n) = \rho(w_0, w_1) + \rho(w_1, w_2) + \cdots + \rho(w_{n-1}, w_n)$.

The metric for the shared path length based on the structural characteristics of the subset structures satisfies *Condition 2* through the above proof. Therefore, this metric can accurately measure the length of the shared path and can be used as the input of the topology inference algorithm.

We use a bottom-up topology inference algorithm described in **Algorithm 1** to fuse the subset structures into a binary tree topology.

Using the BTI algorithm, we insert the leaf nodes one-by-one to construct a binary tree topology by constantly comparing and updating the shared path length metrics.

Algorithm 1 Binary Tree Inference (BTI) Algorithm

Input: Root node s, the leaf node set D, and the estimated length of shared path from the root node to any two leaf nodes $\hat{\rho} = \{\hat{\rho}(\hat{i}, \hat{j}) : \hat{i}, \hat{j} \in D, \hat{i} \neq \hat{j}\}$;
Output: The estimated binary tree topology $\hat{T} = (\hat{V}, \hat{E})$.
Initialize: $\hat{V} = \{s\} \cup D$, $\hat{E} = \emptyset$;
while $|D| \geq 2$ **do**
Find two leaf nodes $\hat{i}$ and $\hat{j}$, s.t. $\{\hat{i}, \hat{j}\} = \arg\max_{\hat{i},\hat{j} \in D} \hat{\rho}(\hat{i}, \hat{j})$;
Create a node $\hat{f}$ as the parent of $\hat{i}$ and $\hat{j}$;
$\hat{E} = \hat{E} \cup \{(\hat{f}, \hat{i}), (\hat{f}, \hat{j})\}$, $D = D \backslash \{\hat{i}, \hat{j}\}$, $\hat{V} = \hat{V} \cup \{\hat{i}, \hat{j}\}$;
for $k \in D$ **do**
$\hat{\rho}(k, \hat{f}) = 0.5 * (\hat{\rho}(k, \hat{i}) + \hat{\rho}(k, \hat{j}))$;
$D = D \cup \hat{\rho}(\hat{i}, \hat{j})$.
end for
end while
$\hat{E} = \hat{E} \cup \{(s, k)\}, k \in D$.

D. GENERAL TREE TOPOLOGY INFERENCE

In the last section, we estimated a binary tree topology by fusing the subset structures. However, in an actual network, the topology is usually a more general tree instead of a binary tree, therefore we need to further optimize the estimated binary tree.

Existing methods usually get a general tree topology by setting a fixed threshold. Literature [14] infers the tree topology by comparing the shared path length between the leaf node pairs and merges the parent nodes of the leaf node pairs into one if the difference of the shared path length is smaller than a given threshold. Literature [21] first gives a penalty factor. Then literature [21] adds a punishment term in the likelihood function in accordance with the nodes number and controls the number of nodes to get the general tree topology. Literature [32] believes that the true links usually longer than the false links (the links that don't exist in the actual network), and deletes the links less than the threshold.

We also set thresholds to infer the general tree. But one fixed threshold does not apply to the general tree topology inference due to the complexity of the network environment. Therefore, different from the existing methods to set a fixed threshold, we set different thresholds for different subset structures. Since the subset structures we infer in Section IV-B are all binary tree-like, their structure is very simple and only contains one intermediate link (the links that do not contain the root node or the leaf nodes). We only need to consider whether this intermediate link of the subset structure should be deleted because the links that contain the source node or the leaf nodes exist in the actual network. If the intermediate link shorter than the threshold, we can consider that this link should be deleted and this subset structure is a general tree-like subset structure, vice versa. We use $g(T_{sub}(i, j, k))$ to represent whether the $T_{sub}(i, j, k)$ is general tree-like.

$$g(T_{sub}(i,j,k)) = \begin{cases} 0, & \text{if } T_{sub}(i,j,k) \text{ is binary,} \\ 1, & \text{if } T_{sub}(i,j,k) \text{ is general.} \end{cases} \quad (7)$$

Our link deletion principle is: an intermediate link is a false link when there are more general tree-like subset structures that contain this link than binary tree-like subset structures. This is because this link is more like a false link and should be deleted if there are more general tree-like subset structures. On the contrary, we do not delete this link when there are more binary tree-like subset structures. Based on this link deletion principle and **Algorithm 1**, we propose a general tree topology inference algorithm described in **Algorithm 2**. Before describing the algorithm, we first introduce some necessary notations. For a tree $T = (V, E)$, let $l(v_i)$ represent a set that contains some leaf nodes, if v_i is a leaf node, $l(v_i) = \{v_i\}$, and if not, $l(v_i)$ is the set of the descendant leaf nodes of v_i.

Algorithm 2 Subset Structure Fusion Topology Inference (SSFTI) Algorithm

Input: Root node s, the leaf node set D, the estimated length of shared path from the root node to any two leaf nodes $\hat{\rho} = \{\hat{\rho}(\hat{i},\hat{j}) : \hat{i},\hat{j} \in D, \hat{i} \neq \hat{j}\}$, and the subset structure set $\hat{g} = \{\hat{g}(T_{sub}(i,j,k)) : i,j,k \in D, i \neq j \neq k\}$;
Output: The estimated general tree topology $\hat{T} = (\hat{V}, \hat{E})$.
Initialize: $\hat{V} = \{s\} \cup D$, $\hat{E} = \emptyset$, $\hat{l} = \{\hat{l}(v_i), 1 \leq i \leq |\hat{V}|\}$;
while $|D| \geq 2$ **do**
  Find two leaf nodes $\hat{i}$ and $\hat{j}$, s.t. $\{\hat{i},\hat{j}\} = \arg\max_{\hat{i},\hat{j}\in D}\hat{\rho}(\hat{i},\hat{j})$;
  Create a node $\hat{f}$ as the parent of $\hat{i}$ and $\hat{j}$;
  $D = D\backslash\{\hat{i},\hat{j}\}$, $\hat{V} = \hat{V} \cup \{\hat{i},\hat{j}\}$, $\hat{l}(\hat{f}) = \hat{l}(\hat{i}) \cup \hat{l}(\hat{j})$, $\hat{E} = \hat{E} \cup \{(\hat{f},\hat{i}),(\hat{f},\hat{j})\}$;
  for $k \in D$ **do**
    $num0 = 0$, $num1 = 0$;
    for $v_w \in \hat{l}(k)$, $v_x \in \hat{l}(i)$, $v_y \in \hat{l}(j)$ **do**
      if $\hat{g}(T_{sub}(v_w, v_x, v_y)) = 0$ **then**
        $num0 = num0 + 1$;
      else
        $num1 = num1 + 1$;
      end if
    end for
    if $num1 \geq num0$ **then**
      $D = D\backslash\{k\}$;
      $\hat{E} = \hat{E} \cup (\hat{f}, k)$;
      $\hat{l}(\hat{f}) = \hat{l}(\hat{f}) \cup \hat{l}(k)$;
    end if
  end for
  for $k \in D$ **do**
    $\hat{\rho}(k,\hat{f}) = 0.5 * (\hat{\rho}(k,\hat{i}) + \hat{\rho}(k,\hat{j}))$;
    $D = D \cup \hat{\rho}(\hat{i},\hat{j})$;
  end for
end while
$\hat{E} = \hat{E} \cup \{(s,k)\}, k \in D$.

E. PROBLEM ARGUMENTATION

In this section, we will demonstrate that the back-to-back packet cannot accurately measure the shared path length to recover the correct topology in the non-stationary network.

The back-to-back packet sends a probe packet of the same size to the destination node pair respectively. There is a very small sending time interval between the two probe packets, so we can consider that the two probe packets passed through the shared path are in a very similar network environment and have the same behaviors, such as packet loss and queuing.

The back-to-back packet currently has two main ways to probe the topology, which are random probe and periodic probe. Random probe stochastically selects two nodes from the set of destination nodes as the destination to send the back-to-back packet, whereas periodic probe detects all destination node pairs in each probe cycle. Periodic probe selects two nodes as the destination in sequence according to the order of the destination node and the sending time interval between the back-to-back packets is very small.

Traditional network topology inference methods assume that the network is stable. The packet delay or packet loss obtained through end-to-end measurement such as back-to-back packet conform to a given distribution, so the metric for the shared path length obtained through statistical calculation can describe the length of the shared path accurately. However, since the real network environment is often very complicated, the performance parameters of the network cannot be described by a definite distribution, so existing methods are not applicable.

We use back-to-back packet to probe the topology in Fig. 2 in a non-stationary network. From Fig. 2, we can see that $\{i,k\}$ and $\{j,k\}$ both have the shared path $p(s,f_1)$. Let Y_1 and Y_2 denote the packet delay on the path $p(s,i)$ and $p(s,k)$ respectively obtained by probing the node pair $\{i,k\}$. Without loss of generality, let Y_3 and Y_4 denote the packet delay on the path $p(s,j)$ and $p(s,k)$ respectively obtained by probing the node pair $\{j,k\}$. Let X_1 and X_2 denote the packet delay on the shared path $p(s,f_1)$ respectively when probing the destination node pair $\{i,k\}$, and let X_3 and X_4 denote the packet delay on the shared path $p(s,f_1)$ respectively when probing the destination node pair $\{j,k\}$. Same as the back-to-back packet measurement mentioned in Section III-C, we also set the queuing delay of the last packet in the back-to-back packet caused by the previous probe packet on the shared path to Z. According to (6), we have

$$\begin{aligned} Var(X_1) &= Var(X_2) = Cov(Y_1, Y_2) \\ Var(X_3) &= Var(X_4) = Cov(Y_3, Y_4). \end{aligned} \quad (8)$$

We can see that $Var(X_1)$ and $Var(X_3)$ are not equal when using the random probe method because the network environment is inconsistent when probing the node pair $\{i,k\}$ and the node pair $\{j,k\}$ in the non-stationary network. So we cannot obtain an accurate metric for shared path length via the random probe.

If we use the periodic probe to detect the destination node pair in turn, assuming that probe destination node pair $\{i, k\}$ before $\{j, k\}$, we have the following analysis:

If at the nth probe, there are b_n background traffic packet between the probe packet sent to j when probing node pair $\{j, k\}$ and the probe packet sent to i when probing node pair $\{i, k\}$. Then the relationship between X_1 and X_3 is

$$X_3 = X_1 + mZ + X_b, \tag{9}$$

where m is the number of probe packets between the probe packets sent to j when probing the node pair $\{j, k\}$ and the probe packets sent to i when probing the node pair $\{i, k\}$. X_b is the packet delay of the background traffic between the probe packets.

Since X_1 and X_b are independent of each other, we get the variance of (9):

$$Var(X_3) = Var(X_1) + Var(X_b). \tag{10}$$

The network is non-stationary, so we have

$$Var(X_b) > 0. \tag{11}$$

According to (6), (10), and (11), we have

$$Cov(Y_3, Y_4) > Cov(Y_1, Y_2). \tag{12}$$

We can find that in a non-stationary network, the later the packets sent, the greater the delay covariance. In the case of a larger network, the more back-to-back packets sent in a probe cycle, the higher the possibility of mixing background traffic in the probe packets and the lower the accuracy of topology inference. Therefore, we decompose the topology inference problem into the smallest sub-problems based on the divide-and-conquer idea and only probe the subset structures with three leaf nodes each time. We send three packets to probe all the three leaf nodes of the subset structure at once, so our end-to-end measurement is not affected by network traffic distribution. Therefore, we can obtain accurate subset structures of the topology and recover the correct complete topology by fusing the subset structures.

V. SIMULATION AND RESULTS

In this section, we evaluate the performance of our method on the network simulator version 2 (NS2) [33] and compare its topology inference results with the methods based on the back-to-back packet. We first introduce the settings of three different network environments in the simulation, and then design multiple simulations to demonstrate the effectiveness of our method.

A. SIMULATION SCENARIOS SETUP

In this section, we construct multiple non-stationary network simulation environments in NS2. In practice, it is difficult to construct a real network environment, so NS2 is an effective experimental tool for network researchers and has been applied to many network studies. Using NS2 for computer network simulation provides a solid foundation for computer network knowledge and skills, covering everything from simple operating system commands to complex network performance indicator analysis [34]. We performed a lot of simulations under different traffic scenarios in NS2 to fully evaluate the performance of our method.

Fig. 5 and Fig. 6 are the two different tree topologies we construct. Fig. 5 is a small binary topology containing 8 leaf nodes and Fig. 6 is a large general tree topology containing 30 leaf nodes. We use Fig. 5 to evaluate the performance of the subset structure inference method and the binary tree inference method. We also use Fig. 6 to evaluate the performance of the general tree topology inference method. We set the parameters of the simulation by referring to the network environment settings of the classic literature [22] and the computing ability of our simulation equipment comprehensively. The parameter setting of the two simulation topologies is very similar. The bandwidth of each link is set to 10 Mbps and the propagation delay of each link is set to 2 ms. Each link has a first in first out (FIFO) queue with the 50-packets buffer size. NS2 is a discrete event simulator, it will compute and record all transfer status (e.g. enqueue, dequeue, receive, drop, et. al) for every packet. So a larger bandwidth, traffic rate, and other parameters setting in the simulation indicate that the simulation cannot be finished in a reasonable time.

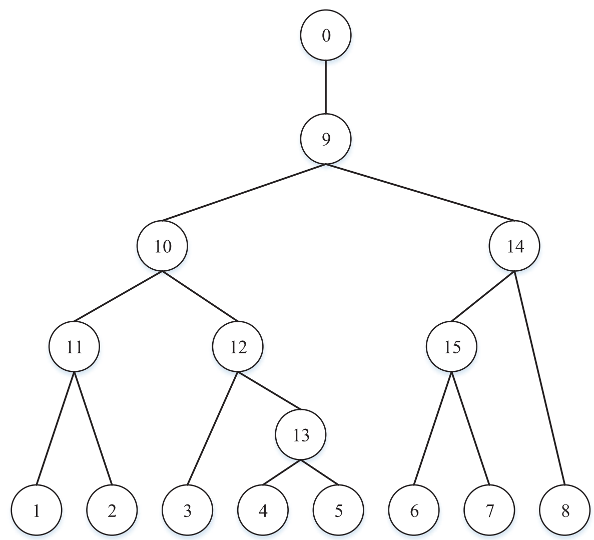

FIGURE 5. Binary tree topology.

We send three-packet to get the end-to-end delay of the subset structure. In each probe cycle, we respectively send one packet from the source node with a size of 20 Bytes to the three leaf nodes of the subset structure. The time interval between adjacent probe packets is 100 ns and every 1ms is a probe cycle. The number of three-packets sent in a subset structure is about 700.

We add two kinds of background traffic in our simulation topologies to simulate the actual non-stationary network. The two kinds of background traffic are called stationary and bursty traffic. The stationary traffic comprises 30 long

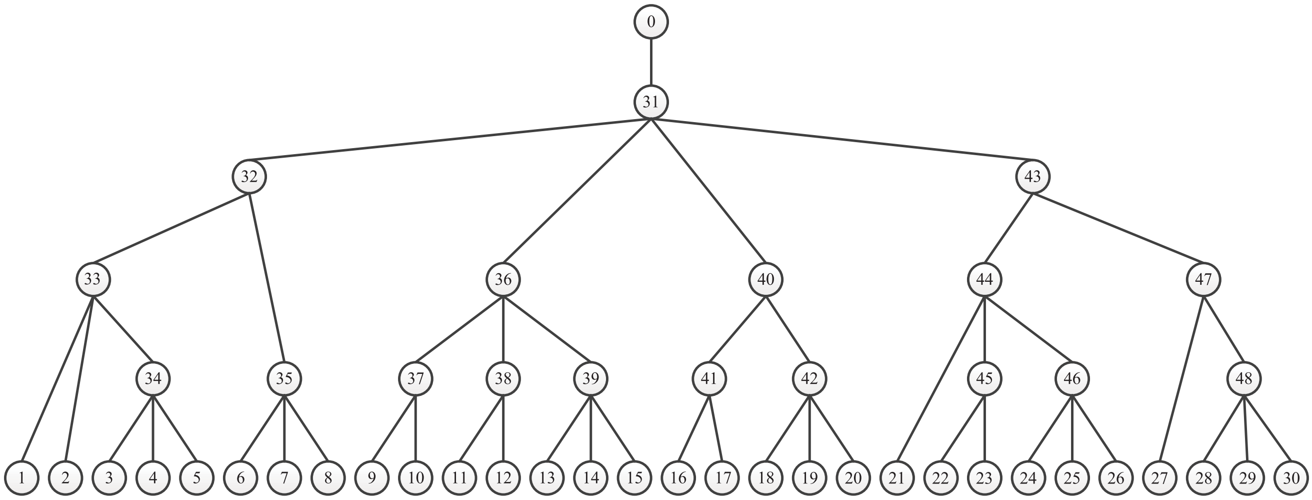

FIGURE 6. General tree topology.

times UDP streams which are Pareto distributed and 30 TCP streams with a constant rate. Both the burst time and idle time of the UDP streams are 2 s. The burst rate of the UDP streams is 0.2 Mbps. The rate of each TCP stream is 0.2 Mbps. Each stationary stream starts randomly during the first 10% of the simulation time and ends randomly during the last 10% of the simulation time. Each bursty stream is randomly generated and last for 2 s during different periods of the simulation. The packet size of each background traffic stream is 500 Bytes.

The bursty traffic is the short-term and high-speed UDP or TCP stream. We build three different bursty traffic scenarios by controlling bursty traffic, which is the slight bursty, middle bursty, and heavy bursty. The slight bursty scenario has two 0.5 Mbps UDP streams and two 0.5 Mbps TCP streams on each link. The middle bursty scenario has three 1 Mbps UDP streams and three 1 Mbps TCP streams. The heavy bursty scenario has three 2 Mbps UDP streams and three 2 Mbps TCP streams. For each traffic scenario, we run 1000 independent simulations to get more accurate results and avoid accidental errors.

To verify the effectiveness of the topology inference algorithm proposed in this paper, we also use the Rooted Neighbor-Joining (RNJ) algorithm [30] and the OLTD algorithm [24] based on the back-to-back packet as a comparison. Both the RNJ algorithm and the OLTD algorithm can be used to infer the general tree network topology. We first set the thresholds in both the RNJ algorithm and the OLTD algorithm to 0 to infer the binary tree topology Fig.5 and then set the thresholds to infer the general tree topology Fig. 6. Both the RNJ algorithm and the OLTD algorithm start from a binary tree. Then by comparing the length of the shared path, the destination nodes are added to the topology one by one and the link which is less than the threshold will not be added to the topology. The difference between the two algorithms is that the RNJ algorithm selects the destination node corresponding to the maximum metric for shared path length to insert to the topology each time, whereas the OLTD algorithm inserts nodes based on the order obtained by DFS ordering. In this paper, the DFS ordering of the OLTD algorithm is from our estimated binary tree.

The RNJ algorithm sends back-to-back packets from the source node to two random leaf nodes, whereas the OLTD algorithm sends back-to-back packets in the DFS ordering. The size of the back-to-back packet is set to 20 Bytes. To make the total number of probe packets the same as our method, the number of back-to-back packets sent to each pair of leaf nodes is about 2,100.

B. TOPOLOGY INFERENCE WITH INCREASING NETWORK SIZE

First, we design a topology inference simulation based on the RNJ algorithm in different network scales to prove the necessity of decomposing the topology inference to sub-problem. The network environment of this simulation is middle bursty and the simulation time is 200 s. We measure the network scale in this simulation by the number of leaf nodes of the topology. The more leaf nodes, the larger the topology. In this simulation, we select 13 binary tree topologies with 3 to 15 leaf nodes. Each topology with a number of n leaf nodes has multiple structures. We run 1000 simulations and select one structure for each simulation randomly.

We use two methods to evaluate the performance of topology inference, which are the tree accuracy and the tree edit distance between the estimated topology and the real topology. These two evaluation methods assess the accuracy of topology inference from two different dimensions.

We define the tree accuracy as the ratio of the simulation number that the tree is correctly estimated to the total simulation number same as we did in literature [28]. The tree accuracy can intuitively reflect the precision of topology inference. However, the tree accuracy is only a macroscopic assessment of the performance of topology inference, whereas the tree edit distance can assess the performance of topology inference from a microscopic point of view.

The tree edit distance between two trees is the total minimum cost of transforming one tree into another tree through editing operations such as deleting, inserting, and relabeling. According to the definition of tree edit distance, we can know that the higher the tree edit distance, the greater the difference between the two trees. Therefore, when the topology cannot be estimated completely, the tree edit distance between the estimated tree and the real tree can also evaluate the performance of the topology inference. The smaller the tree edit distance, the higher the accuracy of the topology inference.

We calculate the tree accuracy and tree edit distance of topology inference. The results of this simulation are shown in Fig. 7 and Fig. 8. From these figures, we can see that as the number of leaf nodes increases which means the scale of the binary tree network topology increases, the tree accuracy of the RNJ algorithm shows a gradual decrease overall, and tree edit distance also shows a gradual increase overall at the same time, indicating that the performance of topology inference has a decreasing trend overall. So this verifies that as the network scale increases, the performance

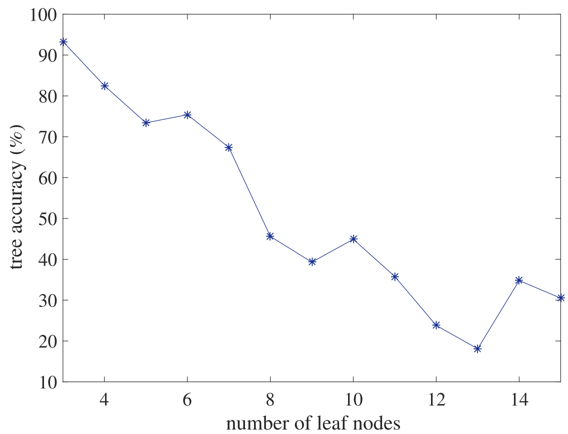

FIGURE 7. **Tree accuracy of binary tree inference in networks of different sizes.**

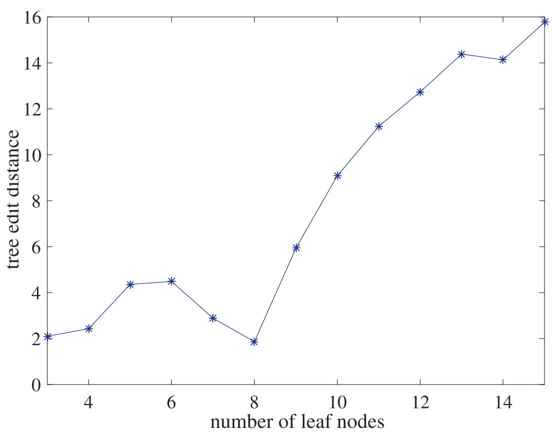

FIGURE 8. **Tree edit distance of binary tree inference in networks of different sizes.**

of RNJ algorithm becomes lower and lower. This is because in the non-stationary network, as the network scale increases, the accuracy of the back-to-back packet becomes lower. The delay covariance metric cannot accurately measure the length of shared path. In order to solve the problem that the performance of the topology inference algorithm decreases sharply with the increase of the network size, we decompose the topology inference problem into the subset structure inference based on the idea of divide-and-conquer, which is more suitable for topology inference in the non-stationary network.

C. SUBSET STRUCTURE INFERENCE RESULTS

Obtaining accurate subset structures is the basis of our approach. Hence, we design a simulation to verify whether our method can get accurate subset structures in different network environments. We run simulations under the three bursty traffic scenarios and calculate the tree accuracy of subset structure inference. The results are shown in Fig. 9.

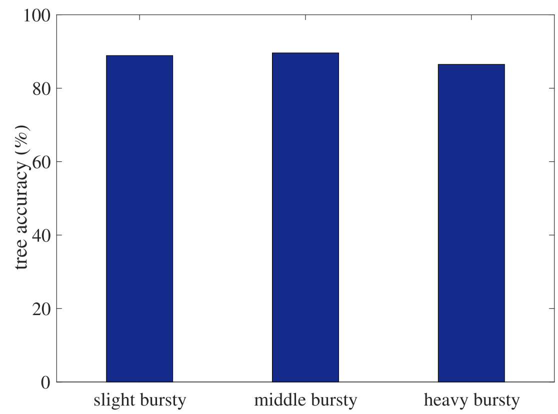

FIGURE 9. **Tree accuracy of subset structure inference.**

From Fig. 9, we can see no matter in what kind of bursty traffic scenario, the overall tree accuracy of the subset structures is very high, reaching over 86%. Therefore, subset structure inference not only has relatively high accuracy but also maintains good stability, which provides a good foundation for topology inference based on the subset structure fusion.

D. SHARED PATH LENGTH MEASUREMENT RESULTS

In this section, we design a simulation to verify the accuracy of the metric for shared path length based on the structural characteristics of the subset structures obtained under different network environments. We calculated the length of the shared path $p(0, 13)$ in Fig.5. We select $p(0, 13)$ because it contains the maximum number of links in all shared paths. We average the theoretical and estimated metrics from 1000 simulations and depict the comparison between theoretical and estimated metrics in Fig. 10. We did not calculate the length of the shared path $p(0, 9)$ because the BTI algorithm did not need the length of $p(0, 9)$.

It can be seen from Fig. 10 that the metrics for shared path length under the three network environments are slightly

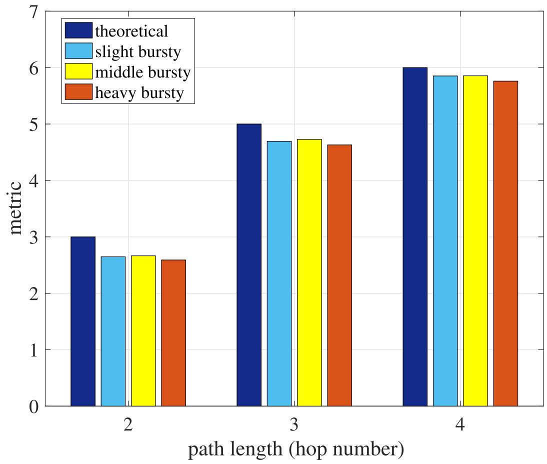

FIGURE 10. Estimated metrics versus theoretical metrics in the three bursty traffic scenarios.

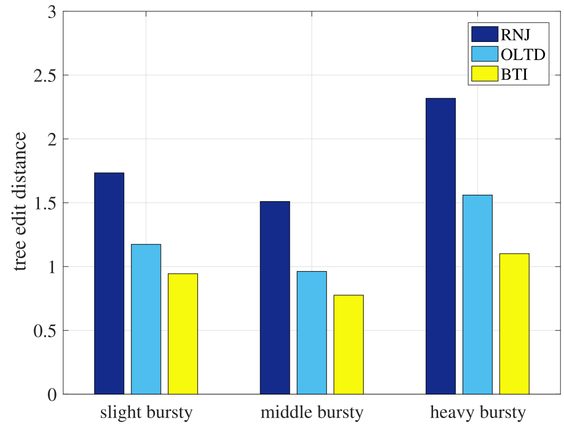

FIGURE 12. Tree edit distance of binary tree inference in the three bursty traffic scenarios.

deviate from the theoretical values. It is because the estimation errors are mainly introduced by *Assumption 2*, which may not be entirely satisfied in real network. In general, these metrics are all very close to the theoretical value, indicating the metric for shared path length based on the subset structure characteristics can be accurately estimated from measured subset structures and it can also be used to measure the length of shared path. This is also the key to accurately infer the topology based on subset structure fusion.

E. BINARY TREE TOPOLOGY INFERENCE RESULTS

Herein, we evaluate the performance of the binary tree topology inference in the three bursty traffic scenarios in the topology in Fig. 5. We first probe and infer all the subset structures using three-packet, and then use the proposed binary tree inference method to infer the tree topology. For comparison, we also use RNJ and OLTD algorithm to infer the topology. We respectively calculated the tree accuracy and the average tree edit distance of each method in the three traffic scenarios. The results are shown in Fig. 11 and Fig. 12.

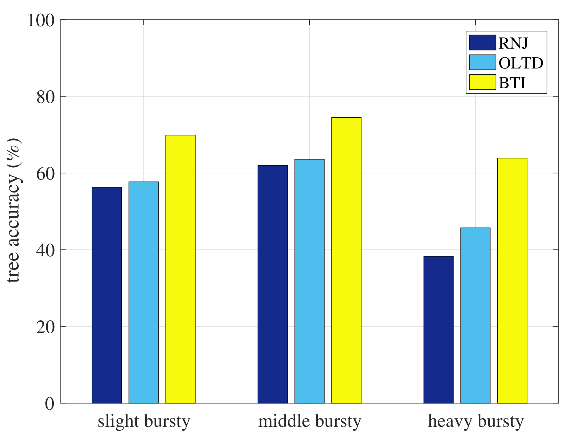

FIGURE 11. Tree accuracy of binary tree inference in the three bursty traffic scenarios.

Comparing the tree accuracy of the three methods in the same network scenario in Fig. 11, we observe that the BTI algorithm has the highest tree accuracy, followed by the OLTD algorithm, and the RNJ algorithm has the lowest tree accuracy. At the same time, comparing the tree edit distance of the three methods in the same network scenario in Fig. 12, we observe that the BTI algorithm has the smallest tree edit distance, followed by the OLTD algorithm, and the RNJ algorithm has the highest tree edit distance. Since the higher the accuracy and the smaller the tree edit distance, the better the topology inference method, so this simulation demonstrates that the BTI algorithm is the best, the OLTD algorithm is the second, and the RNJ algorithm is the worst. Both the RNJ algorithm and OLTD algorithm use back-to-back packet to measure the topology, but the back-to-back packet cannot measure precise shared path length in the non-stationary network. So RNJ algorithm and OLTD algorithm cannot estimate correct topology. Different from RNJ and OLTD algorithm, BTI algorithm decomposes the topology inference to the inference of subset structures that can accurately measure the shared path length via three-packet. Then BTI algorithm can accurately recover the topology by fusing these subset structures. So BTI algorithm is more applicable for topology inference in the non-stationary network.

Fig. 11 and Fig. 12 also illustrate the topology inference results under different network environment. We observe that the BTI algorithm has the highest tree accuracy under all three scenarios and achieves the best performance under the middle bursty scenario. The different performances of the BTI algorithm in different scenarios can be explained as the following two folds. First, in the slight bursty, the change of the packet delay is relatively small, which leads to an error of delay covariance. So the metric for the shared path length cannot be measured very accurately, leading to low accuracy of the subset structure inference and the final binary tree topology inference. Second, in the heavy bursty scenario, the correlation of the three packets in a three-packet may have inconsistent delays in the same shared path, which may also cause inaccurate delay covariance estimation and make it difficult to infer the network topology correctly.

F. GENERAL TREE TOPOLOGY INFERENCE RESULTS

Finally, we evaluate the performance of the general tree topology inference method in the topology in Fig. 6. We run 1000 simulations for each traffic environment. We use tree edit distance to evaluate the performance of the general tree topology inference method, because the large general tree topology cannot be correctly estimated in most simulations.

For each simulation, we first probe the subset structures by using the three-packet end-to-end measurement. Then we obtain the binary subset structures by comparing the shared path length and the general subset structures by setting thresholds. Finally, we estimate the general tree topology by fusing the subset structures.

In order to exhibit the performance of our method, we compare the performance of SSFTI with two classical methods RNJ and OLTD. RNJ and OLTD algorithms both have a tunable parameter, Δ, that must be chosen, whereas SSFTI algorithm needs to set different parameters for different subset structures. We use the same strategy to set Δ regardless of the algorithm. We first compute the maximum shared path length ρ_{max} and divide it into twenty equal parts. Then each threshold is set as follows:

$$\Delta_i = \frac{\rho_{max} \cdot i}{20}, \quad i = 1, 2, \cdots, 20. \tag{13}$$

Fig. 13 to Fig. 15 plot the variation of tree edit distance with the number of probe packets. Note that we choose the result with the highest accuracy in the conditions of setting different thresholds. In our simulation, we find that the best threshold for RNJ is Δ_2 and for OLTD and SSFTI are Δ_4.

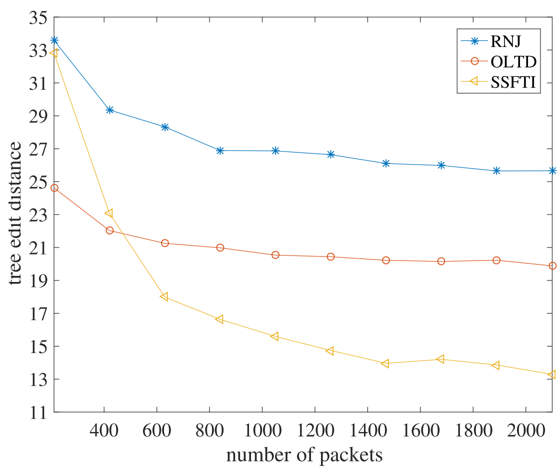

FIGURE 13. **Tree edit distance comparison between SSFTI, RNJ, and OLTD (in the slight bursty traffic scenario).**

From Fig. 13 to Fig. 15 we can see that the accuracy of topology inference is higher if more probe packets are sent. This is because that the end-to-end measurements are the only information that can be used to infer topology and we can obtain more accurate end-to-end delay when sending more

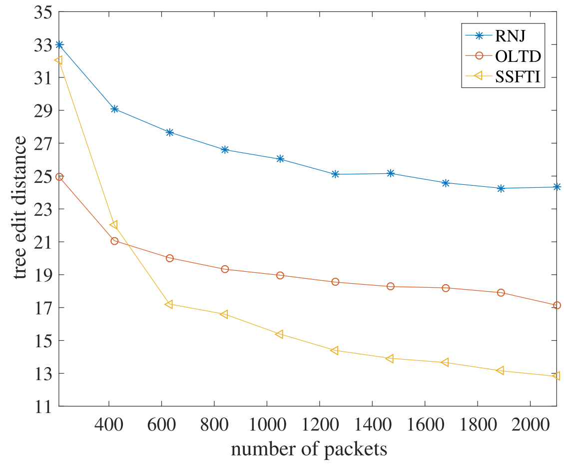

FIGURE 14. **Tree edit distance comparison between SSFTI, RNJ, and OLTD (in the middle bursty traffic scenario).**

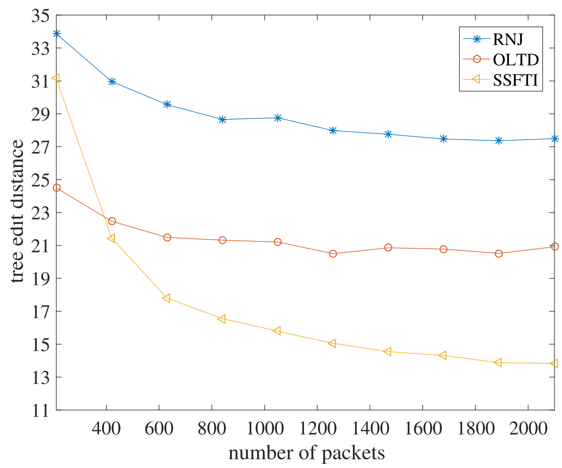

FIGURE 15. **Tree edit distance comparison between SSFTI, RNJ, and OLTD (in the heavy bursty traffic scenario).**

probe packets. More importantly, Fig. 13 to Fig. 15 show that our method outperforms the RNJ algorithm and OLTD algorithm. Both the RNJ and OLTD are able to recover the correct tree topology if the estimated link length errors are smaller than a quarter of the minimum link length, but many factors such as network burst and imbalance may deeply impact the link length and the corresponding error. Therefore, it is difficult to choose only one appropriate threshold to recover accurate topology. Different from most existing methods, our method selects different thresholds for different subset structures adapting to different measurement environments. Therefore, we can get accurate subset structures by unique thresholds, resulting in an accurate estimation of general tree topology.

VI. CONCLUSION

In this paper, we presented a method to infer topology in the non-stationary network by probing and fusing the subset structures. Firstly, we probed and estimated the subset structure containing three leaf nodes via three-packet end-to-end measurement. Secondly, we fused the subset structures to get a binary tree topology by constructing a metric that could effectively describe the length of the shared path from a source node to pairs of destinations. Thirdly, we set different thresholds for different subset structures and deleted the false links in the binary tree to obtain a correct general tree topology. We evaluated the performance of our method by using NS2 simulation and found that our method outperformed existing methods that probed the tree topology using the back-to-back packet in the non-stationary network. We only use packet delay covariance to infer the subset structure, so in the future, we will focus on the subset structure inference based on the packet loss to evaluate the performance of our topology inference method more comprehensively.

REFERENCES

[1] R. Motamedi, R. Rejaie, and W. Willinger, "A survey of techniques for Internet topology discovery," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 1044–1065, 2nd Quart., 2015.

[2] B. Donnet, P. Raoult, T. Friedman, and M. Crovella, "Deployment of an algorithm for large-scale topology discovery," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 12, pp. 2210–2220, Dec. 2006.

[3] H. Gobjuka and Y. J. Breitbart, "Ethernet topology discovery for networks with incomplete information," *IEEE/ACM Trans. Netw.*, vol. 18, no. 4, pp. 1220–1223, Aug. 2010.

[4] P. Marchetta, P. Merindol, B. Donnet, A. Pescape, and J.-J. Pansiot, "Topology discovery at the router level: A new hybrid tool targeting ISP networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 9, pp. 1776–1787, Oct. 2011.

[5] 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.

[6] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: Recent developments," *Stat. Sci.*, vol. 19, no. 3, pp. 499–517, Aug. 2004.

[7] T. He, A. Gkelias, L. Ma, K. K. Leung, A. Swami, and D. Towsley, "Robust and efficient monitor placement for network tomography in dynamic networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1732–1745, Jun. 2017.

[8] H. Li, Y. Gao, W. Dong, and C. Chen, "Preferential link tomography in dynamic networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 5, pp. 1801–1814, Oct. 2019.

[9] C. Feng, L. Wang, K. Wu, and J. Wang, "Bound-based network tomography with additive metrics," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Paris, France, Apr. 2019, pp. 316–324.

[10] 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.

[11] Y. Qiao, "Robust loss inference in the presence of noisy measurements," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Honolulu, HI, USA, Apr. 2018, pp. 738–746.

[12] F. Thouin, M. Coates, and M. Rabbat, "Large scale probabilistic available bandwidth estimation," *Comput. Netw.*, vol. 55, no. 9, pp. 2065–2078, Jun. 2011.

[13] M. J. Coates and R. Nowak, "Network loss inference using unicast end-to-end measurement," in *Proc. ITC Conf. IP Traffic, Modeling Manage.*, Monterey, CA, USA, 2000, pp. 1–9.

[14] M. Coates, R. Castro, R. Nowak, M. Gadhiok, R. King, and Y. Tsang, "Maximum likelihood network topology identification from edge-based unicast measurements," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 30, no. 1, pp. 11–20, Jun. 2002.

[15] S. Ratnasamy and S. McCanne, "Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements," in *Proc. IEEE INFOCOM*, New York, NY, USA, Mar. 1999, pp. 353–360.

[16] N. G. Duffield, J. Horowitz, and F. L. Presti, "Adaptive multicast topology inference," in *Proc. IEEE INFOCOM*, Anchorage, AK, USA, Apr. 2001, pp. 1636–1645.

[17] N. G. Duffield, J. Horowitz, F. L. Presti, and D. Towsley, "Multicast topology inference from measured end-to-end loss," *IEEE Trans. Inf. Theory*, vol. 48, no. 1, pp. 26–45, Jan. 2002.

[18] N. G. Duffield and F. LoPresti, "Network tomography from measured end-to-end delay covariance," *IEEE/ACM Trans. Netw.*, vol. 12, no. 6, pp. 978–992, Dec. 2004.

[19] H. Nguyen and R. Zheng, "A binary independent component analysis approach to tree topology inference," *IEEE Trans. Signal Process.*, vol. 61, no. 12, pp. 3071–3080, Jun. 2013.

[20] R. Bowden and D. Veitch, "Finding the right tree: Topology inference despite spatial dependences," *IEEE Trans. Inf. Theory*, vol. 64, no. 6, pp. 4594–4609, Jun. 2018.

[21] R. M. Castro, M. J. Coates, and R. D. Nowak, "Likelihood based hierarchical clustering," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2308–2321, Aug. 2004.

[22] M.-F. Shih and A. O. Hero, III, "Hierarchical inference of unicast network topologies based on end-to-end measurements," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1708–1718, May 2007.

[23] B. Eriksson, G. Dasarathy, P. Barford, and R. Nowak, "Toward the practical use of network tomography for Internet topology discovery," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 1–9.

[24] B. Eriksson, G. Dasarathy, P. Barford, and R. Nowak, "Efficient network tomography for Internet topology discovery," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 931–943, Jun. 2012.

[25] T. Pepe and M. Puleri, "Network tomography: A novel algorithm for probing path selection," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 5337–5341.

[26] J. Ni, H. Xie, S. Tatikonda, and Y. R. Yang, "Network routing topology inference from end-to-end measurements," in *Proc. IEEE 27th Conf. Comput. Commun. (INFOCOM)*, Phoenix, AZ, USA, Apr. 2008, pp. 36–40.

[27] A. Malekzadeh and M. H. MacGregor, "Network topology inference from end-to-end unicast measurements," in *Proc. 27th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Barcelona, Spain, Mar. 2013, pp. 1101–1106.

[28] G. Fei, J. Ye, S. Wen, and G. Hu, "Network topology inference using higher-order statistical characteristics of end-to-end measured delays," *IEEE Access*, vol. 8, pp. 59960–59975, 2020.

[29] A. Santos, V. Matta, and A. H. Sayed, "Divide-and-conquer tomography for large-scale networks," in *Proc. IEEE Data Sci. Workshop (DSW)*, Lausanne, Switzerland, Jun. 2018, pp. 170–174.

[30] A. Rai and E. Modiano, "Topology discovery using path interference," in *Proc. IFIP Netw.*, Warsaw, Poland, 2019, pp. 1–2.

[31] M. Rahali, J.-M. Sanner, and G. Rubino, "Unicast inference of additive metrics in general network topologies," in *Proc. MASCOTS*, Rennes, France, 2019, pp. 107–115.

[32] J. Ni, H. Xie, S. Tatikonda, and Y. R. Yang, "Efficient and dynamic routing topology inference from end-to-end measurements," *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 123–135, Feb. 2010.

[33] T. Issariyakul and E. Hossain, "Simulation of computer networks," in *Introduction to Network Simulator 2 (NS2)*. Bostan, MA, USA: Springer, 2009, pp. 1–18, doi: 10.1007/978-0-387-71760-9_2.

[34] A. K. Nayak, S. C. Rai, and R. Mall, *Computer Network Simulation Using NS2*. Boca Raton, FL, USA: CRC Press, 2016.

JIAN YE was born in Sichuan, China, in 1994. He received the B.S. and M.S. degrees from the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2017 and 2020, respectively.

He is currently a Research Assistant with the School of Information and Communication Engineering, UESTC. His main interest includes network topology measurement.

GAOLEI FEI was born in Zhejiang, China, in 1982. He received the B.S. and Ph.D. degrees from the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2006 and 2012, respectively.

From 2010 to 2011, he was a Graduate Research Trainee at McGill University, Montreal, QC, Canada. He is currently an Associate Professor with the School of Information and Communication Engineering, UESTC. His main research interests include network topology measurement, social network analysis, and social media text mining.

XUEMENG ZHAI was born in 1991. He received the B.S. and Ph.D. degrees from the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2014 and 2020, respectively.

He is currently an Assistant Professor with the School of Information and Communication Engineering, UESTC. His research interests include network science, network sparse representation, and stochastic network models.

GUANGMIN HU received the B.S. degree from the Department of Computer Science, Nanjing University, China, in 1986, and the M.S. and Ph.D. degrees from the Chengdu University of Technology, Chengdu, China, in 1992 and 2000, respectively.

From 2000 to 2003, he was a Postdoctoral Researcher with the School of Communication and Information Engineering, University of Electronic Science and Technology of China (UESTC). From 2002 to 2003, he was a Visiting Scholar at The Hong Kong Polytechnic University. He is currently a Professor with the School of Information and Communication Engineering, UESTC. His current research interests include computer networks and signal processing.