A Method of Tree Network Topology Inference Based on Hierarchical Host Table

Hongbin Wang, Wei Ding, Haiting Zhu
School of Computer Science and Engineering
Southeast University
Nanjing, China

***Abstract*—**Network topology inference uses network measurement methods to find out the elements of internet in the target field and their connection relationship. In traditional network topology inference, traceroute-like methods which bases on the feedback information of routers are used, but anonymous routers seriously affect the performance. So tomography relying on end-to-end measurements has become a hot topic. This paper provides a method of topology inference based on hierarchical host table. On one hand, this method doesn't require any cooperation from the internal routers, which can avoid the problem of anonymous routers; on the other hand, unlike traditional tomography techniques, we limit the clustering problem on the same router level, which not only improves the topology inference accuracy, but also reduces the measurement cost. The simulation on NS-2 shows this method can get the logical network topology effectively.

Keywords-Network topology inference; tomography; anonymous router; hierarchical host table

I. INTRODUCTION

Network topology inference is a traditional research field. There are three different levels at which to describe the network topology [1]: the link layer topology, the internet topology and the overlay topology. Moreover, the internet topology can be further classified into four different levels: the IP interface level, the router level, the point of presence level and the AS level. In this paper, we focus on the router level of the internet topology to get the connectivity between routers and hosts. The router level topology information of the infrastructure not only help to locate network failure for network management system, but also contribute to the construction of an efficient overlay network[2] and the design of application-layer multicast protocols, etc.

In traditional methods of network topology inference, traceroute-like tools are often used to extract the router-level path [3,4]. Traceroute is implemented based on Internet control message protocol (ICMP). In traceroute, the source sends out a series of IP datagrams with increasing time-to-live (TTL) to the destination. From the returned ICMP error messages, it obtains intermediate router information. Those tools not only produce large amounts of ICMP packets, but also have the problems with anonymous routers [5]. Anonymous router means the router discards ICMP error messages and then the result of traceroute appears as "*" in this case. So, tomography relying on end-to-end measurements has become a hot topic. Without any cooperation from the internal routers, tomography can avoid the problem of anonymous router. Ratnasamy *et al*. [6] and Duffield *et al*. [7] pioneered this work by using multicast to infer the network topologies. By sending multicast probes from the root node of the tree to a pair of the leaf nodes, one can estimate the successful transmission rate on the shared portion of the probe paths, called shared path, based on end-to-end loss, and then use the deterministic binary tree classification algorithm to construct a binary logical tree in a bottom-up manner. The extension to a general tree is basically done by pruning the links with loss rates less than some heuristically selected threshold.

This paper focuses on the single-source multi-objective tree topology discovery. We formulate the topology estimation as hierarchical clustering of the leaf nodes based on pairwise correlations as similarity metric. On the condition that we have the hop information of all the objective nodes, which can be inferred by TTL values, we can build Hop-count-to-IP-set mapping table. We call this table Hierarchical Host Table (HHT). By sending a special probe named sandwich probe [8] based on HHT, we can assemble the objective nodes with the same topology into a same cluster and get the maximum shared path between clusters. Then the connection of internal routers is clear. This method can get the general tree directly and decrease the measurement cost effectively through divide and conquer principle.

This paper is organized as follows. In Section II, we introduce the tree network topology problem and some technical concepts. In section III, we introduce in detail the topology inference based on hierarchical host table. In section IV, we conduct comprehensive simulation in NS-2 to evaluate the performance of our algorithm. Section V provides the comparison and lists the limitations.

II. RELATED WORK

A. Formal description of network topology

Our work focuses on the tree structure network topology. In this topology structure, there is a source node and many target nodes. We call this kind of topology the single-source multi-objective tree topology. A logical tree T = (V, E) is defined by two sets of objects: V as the set of nodes, and E as the set of directed links. We let the root be defined as node v0, then V = {v0} ∪ Vi ∪ Vf, Vi means the set of internal nodes on behalf of routers, Vf means the set of leaf nodes on behalf of remote

hosts. The root is the only node having a single child node, while all internal nodes have at least two child nodes. Furthermore, for any a node v∈Vf, we define par (v) as the parent node of v. For any pair of nodes (vi, vj) from Vf set with the same height in the tree, if par (vi) equals to par (vj), then vi and vj are the sibling, otherwise cousins.

B. *Hierarchical host table*

The hierarchical host table is a tool to describe the tree network topology by levels of leaf nodes, for example remote hosts. Under the Internet Protocol, TTL (time-to-live) is an 8-bit field in the IP header. The TTL value can be taken as an upper bound on the time that an IP datagram can exist in an Internet system. The TTL field is set by the sender of the datagram, and reduced by all routers on the route to its destination. If the TTL field reaches zero before the datagram arrives at its destination, then the datagram is discarded and an ICMP error datagram is sent back to the sender. The purpose of the TTL field is to avoid a situation in which an undeliverable datagram keeps circulating on an Internet system, so such system could eventually be swamped by such "immortals".

In theory, under IPv4, time to live is measured in seconds, although every router that forwards the datagram must reduce the TTL by at least one unit. In practice, the TTL field is reduced by one on each hop. Furthermore, most modern OSes use only a few selected initial TTL values, such as 32, 64,128 and 255. This set of initial TTL values covers most of the popular OSes, such as Microsoft Windows, Linux, variants of BSD and many commercial Unix systems. Since Internet traces have shown that few Internet hosts are apart by more than 30 hops [9, 10], one can determine the initial TTL value of a packet by selecting the smallest initial value in the set that is larger than its final TTL [11]. Then difference between the initial value and final TTL means the hop count. For example, if the final TTL value is 120, then the initial TTL value will be 128 and the hop count will be 8. We call the Hop-count-to-IP-set Hierarchical Host Table (HHT).

C. *Sandwich probe*

Sandwich probe were invented by Castro et *al. in* [8] for the similar purpose of topology estimation. Each probe contains three time-stamped packets: two small packets and one big packet sandwiched between the two small ones. The small packets are sent to one of the two leaf nodes, while the large packet is sent to the other [see Fig. 1]. In the packet switching network, store and forward technique is used in networks with intermittent connectivity. Every packet will experience four kinds of delay: processing delay, queuing delay, transmission delay and propagation delay. The transmission of the big packet is the main reason for the different arrival time of the two small packets. In Fig. 1, p1 and p2 are the two small packets whose destinations are node 3; q is the big packet whose destination is node 2. The initial time delay of the two small packets is d. After passing through a router, the time delay increases to d+Δd. If the link bandwidth between node 1 and node 3 is k bps and the size of big packet is m bits, then there would be Δd = m/k+Δ. m/k means the propagation delay of the big packet. Δ means the time interval between the end time of the big packet and the departure time of the second small packet.

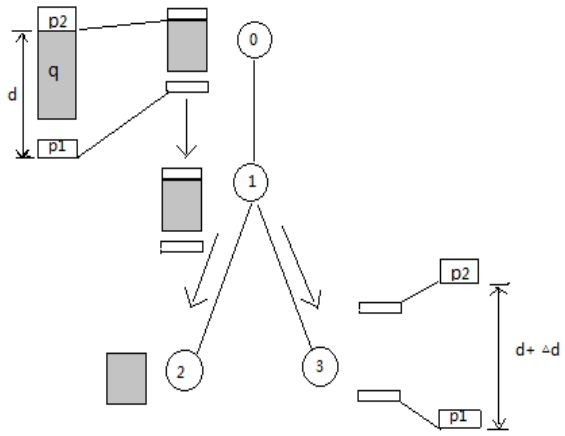

Figure 1. The sandwich transmission examples

D. *Similarity metric*

In topology inference, the concept of metric-induced network topology (MINT) introduced by Bestavros *et al*. [12] provides a framework for defining the similarity metrics. Under the MINT framework, a metric is defined to capture the similarity between all measurement pairs. Any pair of leaf nodes that are connected to the source through common links will have approximately equal similarity according to the metric. Different network topologies will usually generate different clusters of leaf pairs having almost identical similarities. The most important property of the similarity metrics should be monotonic and sensitive to the router. An internal node has a smaller metric value than any of its descendants. The metric shares a positive correlation with the length of the common links between the two nodes.

The packet loss rates may be an effective similarity metric in a highly congested network. While in a normal network environment, packet loss rates don't possess a better differentiation. But the time difference between the two small packets in a sandwich probe is a metric closely related to the router and sensitive to the number of the routers on each link. For any sandwich probe, we define t_1 as the arrival time of the first small packet, t_2 as the arrival time of the second small packet, the time difference representing the similarity can be defined as t_2-t_1. So we choose the mean time difference as the similarity metric in this paper.

III. TOPOLOGY INFERENCE ALGORITHM

Network topology inference problem can be divided into three steps to achieve. (1) Constructing the HHT mapping table based one the method described in Section II-B, (2) Clustering all the leaf nodes based on similarity, layer by layer. This step leads to hierarchical topology. (3) Transforming the

hierarchical topology into tree network topology. Next we will introduce the algorithms in step 2 and step 3.

A. Similarity clustering algorithm

We define the time difference between the two small packets in sandwich probe as similarity. In our model, we also assume the following statistical properties on the network environment:

- Spatial independence : the packet delays over different links are independent;
- Temporal independence and stationarity: the packet delays over a link are identically and independently distributed.
- Delay consistency: the queueing delays of the packet pair are identical with probability 1 when they travel along the shared path.

Definition 1 The node pair to similarity (NP2S) mapping format is as follows:

(node1, node2, similarity)

Where **node1** is the destination node of the small packet in sandwich probe, **node2** is the destination node of the big packet in sandwich probe and **node1**$\in \forall f$, **node2**$\in \forall f$. **Similarity** is time difference between the two small packets.

In experiment, by sending a series of sandwich probes to a pair of the leaf nodes, layer by layer, one can build the NP2S mapping table. Through the similarity clustering algorithm, we can assemble different leaf nodes into different clusters. All leaf nodes in the same cluster correspond to a unique internal node in the topology. This internal node is the nearest common ancestor shared by each other. The detailed description of similarity clustering algorithm is as follows.

Input: the NP2S mapping table **NP2ST**

Output: the set of clusters that contain all the nodes sharing the same ancestor in the particular layer of the tree.

Description in pseudo code:

```
let S be the set which contains all the NP2S items from the table NP2ST;
let R={} be the returned set of clusters that contain all the nodes sharing the same ancestor;
let T={} be an empty set of leaf nodes;
let N be the number of the leaf nodes from the particular layer of the tree;
sort the set S by the field of similarity in descending order;
While (S!= empty set) {
    get a NP2S item (m,n,d) with the maximum similarity;
    S=S-{(m,n,d)};
    S'=S' ∪ {m,n};
    if (|S'|==N) break;
}
While (S'!=empty set) {
    get an item (m',n',d');
    S'=S'-{( m',n',d')};
    T_i={m',n'};
    flag=false;
    for (each set T_j && j<i) {
    if (m' ∈T_j || n' ∈T_j) {
        merge the two sets ,T_j={T_i} ∪ {T_j};
        flag=true;
        break; }
    }// for
    if ( flag == false)
        i++;
}// while
R={T_1} ∪ {T_2}…{T_{i-1}};
if( flag==true) R=R ∪ {T_i};
```

The element in the returned set R is a cluster that contains all the nodes sharing the same ancestor in the same layer of the tree. The relationship of the pair nodes in different clusters are cousinship. According to the clusters and the HHT information, we can build the hierarchical topology.

B. Tree network topology inference from hierarchical topology

In hierarchical topology, we have identified the clusters of leaf nodes that share certain properties. In particular, we want to identify the clusters of leaf nodes whose paths from the source node are the same up to a certain point which can be defined as the number of shared routers. This is also the key point of transforming from hierarchical topology to tree network topology. In Fig. 1, the existence of the big packet in the sandwich probe is the reason for the change of similarity. Furthermore, TTL in the IP header is an attribute that can control the life of a packet. Based on the principle of traceroute, we can use the TTL to control the running state of the big packet in sandwich to record the similarity with the change of the TTL. Theoretically, when the TTL of the big packet is less than the number of shared routers, the similarity is positively related to TTL, otherwise the similarity tends to be stable. So we can get the number of the shared router through the corresponding TTL of the turning point of similarity. The detailed method is as follows.

1) Send a series of sandwich probes to a pair of nodes from different clusters and the TTL of the big packet

will be indexed, starting with 1 and increasing by 1. If the two nodes are from different layers, maintain the node from the lower layer as the destination of the two small packets.

2) Analyze the similarities with the change of TTL, record the corresponding TTL of the turning point of similarity, which means the number of shared routers between the two nodes.

3) Transform from hierarchical topology to tree network topology according to the HHT and the shared routers information between different clusters.

The number of maximum shared routers must be less than the lower layer of the two nodes in the sandwich probe. So by sending limited sandwich probes, one can get the number of shared routers between any clusters.

IV. SIMULATION

In experiment, we simulate a network topology in NS-2[see Fig. 2].

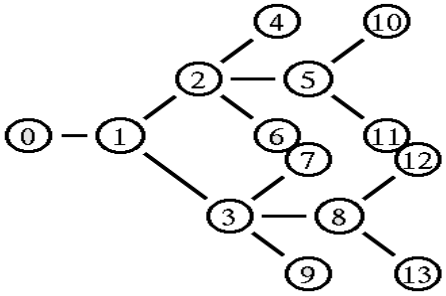

Figure 2. The simulation network topology

In this topology, root node V_0=0, internal node set Vi={1,2,3,5,8}, leaf node set Vf={4,6,7,9,10,11,12,13}. At first we build the HHT table [see TABLE I] according to hop information inferred by TTL values.

TABLE I. HHT

Layer of node	Set of nodes
2	4，6，7，9
3	10，11，12，13

By sending a series of sandwich probes to a pair of the leaf nodes, layer by layer, we get all the similarity metric values [see TABLE II].

TABLE II. SIMILARITY OF NODE PAIRS

Node pair	similarity	Node pair	similarity
(4,6)	0.006526	(10,11)	0.009004
(4,7)	0.001677	(10,12)	0.000969
(4,9)	0.001350	(10,13)	0.000958
(6,7)	0.001631	(11,12)	0.001076
(6,9)	0.001257	(11,13)	0.001310
(7,9)	0.007577	(12,13)	0.010287

Based on similarity clustering algorithm, we can identify clusters of leaf nodes that share the same similarity. In Fig. 3, we get four clusters, such as {4, 6}, {7, 9}, {10, 11} and {12, 13}. According to the clusters and HHT, the hierarchical topology is described in Fig. 3.

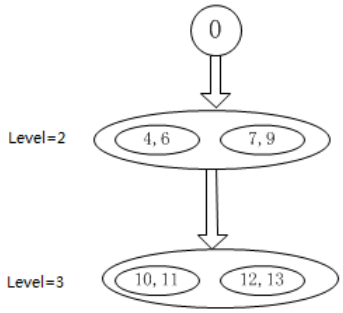

Figure 3. The hierarchicla topology

When considering the transformation from hierarchical topology to a tree network topology, we choose the smallest label in one cluster as the cluster's representative. For example, the node 4 represents the cluster {4, 6}. By sending a series of sandwich probes with changing TTL, the variation of the similarity show in Fig. 4.

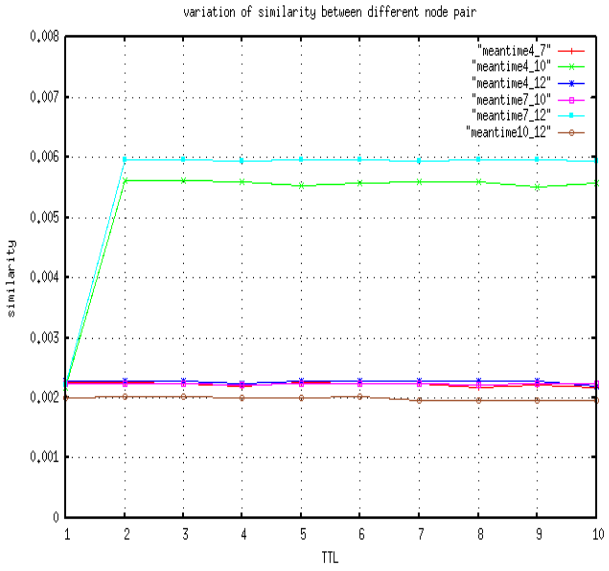

Figure 4. The variation of similarity with the changing TTL

From the Fig. 4, we know that there are two shared routers between node 4 and node 10, so do the node 7 and node 12. All the other nodes share only one common router. Then the transformation from hierarchical topology to tree network topology showed in Fig. 5.

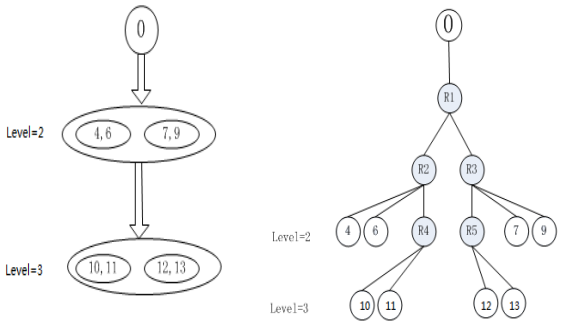

Figure 5. The transformation from hierarchical topology to tree network topology

The result shows that we get the right tree network topology. Our work focuses on the connections between routers and the clusters of leaf nodes attached to any router, instead of the identification of all the routers. So the lables R1-R5 in Fig. 5 simply mean the existence of router rather than IP addresses.

V. CONCLUSIONS

This paper provides a method of topology inference based on a hierarchical host table. On one hand, without any cooperation from the internal routers, this method can avoid the problem of anonymous routers; on the other hand, through constructing a hierarchical host table based on TTL information, this method can reduce the measurement cost effectively. For example, given N leaf nodes in an asymmetric network without anonymous routers, traditionally full N(N-1) traceroutes are needed to determine the underlay topology. Using tomography techniques, the number of detected node pair is N*(N-1)/2, however in this method, assuming the average size of each subnet hosts is S, then the number of the detected node pair consists of two parts. One is between hosts from each router level, which equals L*(L-1)/2,L means the average hosts number in the same router level. The other is between subnets, which equals M*(M-1)/2,M=N/S;Furthermore, this method limits the clustering problem on the same router level, which can also improve the similarity clustering accuracy and avoid the extension to a gernel tree from a binary logical tree.

The limitations of our method are (i) the similarity which is described as the time difference between the two small packets in the sandwich probe may face some deviation when the network load is large; (ii) the tree network topology only describes the connections between routers instead of the identifications of those routers which needs the cooperation with other networking protocols, for example SNMP.

ACKNOWLEDGMENT

This paper is supported by the National Basic Research Program of China under Grant No. 2009CB320505 and the National Key Technology R&D Program of China under Grant No.2008BAH37B04.

REFERENCES

[1] Donnet D,Friedman T. Internet topology discovery: A survey.IEEE Communications Surveys and Tutorials,2007,9(4):2-15.

[2] X. Jin, Y. Wang, and S.-H. G. Chan, “Fast overlay tree based on efficient end-to-end measurements,” in Proc. IEEE ICC, May 2005, pp.1319–1323.

[3] V.Jacobson,“Pathchar”,1997.[Online].Available: http://www.caida.org/tools/utilities/others/pathchar

[4] Traceroute. [Online]. Available: http://www.traceroute.org/

[5] B. Yao, R. Viswanathan, F. Chang, and D. G. Waddington, “Topology inference in the presence of anonymous routers,” in Proc. IEEE INFOCOM,Apr. 2003, pp. 353–363.

[6] S. Ratnasamy and S. McCanne, “Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements,” presented at the IEEE INFOCOM 1999, New York, NY, Mar. 1999.

[7] N. G. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley, “Multicast topology inference from measured end-to-end loss,” IEEE Trans. Inf.Theory, vol. 48, pp. 26–45, Jan. 2002.

[8] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, “Internet tomography:Recent developments,” Stat. Sci., vol. 19, no. 3, pp. 499–517,2003.

[9] B. Cheswick, H. Burch, and S. Branigan, “Mapping and visualizing the Internet,” in Proc. USENIX Annu. Technical Conf., 2000, pp. 1–12.

[10] K. Claffy, T. E. Monk, and D. McRobb, “Internet tomography,” Nature,Jan. 7, 1999.

[11] Haining Wang, Cheng Jin and Kang G.Shin, “Defense Against Spoofed IP Traffic Using Hop-Count Filtering”, IEEE/ACM Transaction on Networking, Vol. 15, No. 1, Feb. 2007

[12] A. Bestavros, J. Byers, and K. Harfoush, “Inference and labeling of metric-induced network topologies,” presented at the IEEE INFOCOM 2002, New York