Unicast Network Topology Inference Algorithm Based on Hierarchical Clustering

Xiao Fu(肖甫)[1,2*], *Shi Chenhang*(是晨航)[1], *Huang Kaixiang*(黄凯祥)[1], *Wang Ruchuan*(王汝传)[1,2]

1. College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, P. R. China;
2. Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing University of Posts and Telecommunications, Nanjing 210003, P. R. China

(Received 16 December 2014; revised 17 August 2015; accepted 8 October 2015)

Abstract: Network topology inference is one of the important applications of network tomography. Traditional network topology inference may impact network normal operation due to its generation of huge data traffic. A unicast network topology inference is proposed to use time to live (TTL) for layering and classify nodes layer by layer based on the similarity of node pairs. Finally, the method infers logical network topology effectively with self-adaptive combination of previous results. Simulation results show that the proposed method holds a high accuracy of topology inference while decreasing network measuring flow, thus improves measurement efficiency.

Key words: network topology inference; network tomography; hierarchical clustering; time to live(TTL)

CLC number: TP393 **Document code**: A **Article ID**: 1005-1120(2015)06-0591-09

0 Introduction

Network topology inference is to identify the logical connection relationships between network elements using a variety of measurements and then to speculate the network topology. As the key technology in the field of network measurement, it is of great significance for network management, network operations and network security. Traditional network topology inference methods, including active measurement method like Traceroute[1,2], usually need collaboration of intermediate nodes and protocol support. With an expanding network size and an increasing security requirements, the collaboration between nodes has become more and more difficult, resulting in difficult implement of traditional network topology inference method . Therefore, network topology inference based on end-to-end measurement, also known as network topology inference based on network tomography, has been the focus of scholars.

Network tomography can obtain network internal characteristics based on end-to-end measurement and does not need the collaboration among network internal nodes. In the tree structure network, the corresponding characteristics will be more relevant with the increasing shared link of nodes[3]. Network topology inference, as a typical application of network tomography, can infer the network topology according to the relevance of the performance characteristics of network nodes. Network topology inference based on network tomography was first applied to the multicast network. Tian et al.[4] proposed a method based on hamming distance and hop count to infer multicast network topology, where hop count was used to obtain topology level information, while hamming distance was used to identify multicast network topology. However, this algorithm can be applied only to the network with lighter load, and the need for clock synchroniza-

* **Corresponding author**: Xiao Fu, Professor, E-mail: xiaof@njupt.edu.cn.

How to cite this article: Xiao Fu, Shi Chenhang, Huang Kaixiang, et al. Unicast network topology inference algorithm based on hierarchical clustering[J]. Trans. Nanjing U. Aero. Astro., 2015,32(6):591-599.
http://dx.doi.org/10.16356/j.1005-1120.2015.06.591

tion between nodes is also the restriction for this method to practical applications. Due to the less equipment support muticast in actual network, this results in the research on unicast network topology inference have become more practical and valuable. Zhao et al. [5] proposed a "transport train" measurement which just needed one measurement source without clock synchronization and inferred network topology based on queuing delay. In order to decrease restrict of single parameter in network topology, another multi-parameter topology inference algorithm was developed by combining time delay with packet loss rate[6]. Su et al. [7] divided leaf nodes into mutually disjoint groups through fan-out attenuation mechanism, then inferred the general structure of network topology based on SBA sorting method. Brian et al. clustered terminal nodes through depth first ordering, and reconstructed the logical topology on the basis of depth-first-search (DFS) ordering[8]. Recently, Brett et al. [9,10] proposed a prototype iTop, an algorithm for inferring the network topology when only partial information was available, by construcing a virtual topology, and then repeatedly merged links in this topology toward true network structure. Zhang[11] proposed a novel binary tree pruning algorithm based on t-test to infer the network topology and a lower bound on the correctly identified probability of the proposed method as well. However, network topology inference based on network tomography assumes that the intermediate router nodes are all cooperative, and that it will cause excessive probe packets and large network measuring flow seriously affecting normal operation of the network. Therefore, how to decrease the measuring flow without influencing the accuracy of topology inference deserves our exploration.

A unicast network topology inference based on hierarchical clustering is proposed. This method uses TTL field of probe packets for layering on leaf nodes at first, and then clusters leaf nodes on each layer based on similarity clustering algorithm. Finally, it infers the whole network topology based on hierarchical clustering results and the changing TTL value.

1 Measurement of Network Topology Inference

1.1 Related definition

Network topology, corresponding to a single-source multi-objective tree structure, is set as $T=(V,E)$, where V the node set and E is the link set. v_0 is set as the root node of the tree, and also the sending node of probe packets. R is the leaf nodes set of the tree, and also the target nodes set that will be measured. The internal nodes set is defined as I, except for the root node and the leaf nodes. Then we get $V=\{v_0\}\cup I\cup R$. The parent node of each node v except for the root node is defined as $p(v)$. If $p(v_i)$ is the parent node both of node v_i and node v_j, v_i and v_j are brother nodes. Furthermore, in combination with network topology inference, we assume that root node has only one child node root node while other internal ones have at least two children.

1.2 Sandwich probe measurement method

Sandwich probe measurement method was first proposed by Castro et al.[12] Each sandwich probe packet is composed of two short packets, and a long packet and the length of the long packet has to be considerably longer than that of the short ones. The long packet is located in the middle of the two short packets with a same destination address, but the destination address of the long packet is differ from that of the short ones. As shown in Fig. 1, short packets p_1,p_2 have a same destination address which is node 3, while the destination address of the long packet q is node 5. The initial interval between two short packets is d. Due to the queue delay of long packet by router, the time interval between two short packets reaching the destination becomes large. In Fig. 1, long packet q generates queuing delay when being forwarded by node 1, which results in increase of d to $d+\Delta d$ eventually. The more shared links which the short packets and the long packet go through, the longer queuing delay will be generated by the long packet, and the lar-

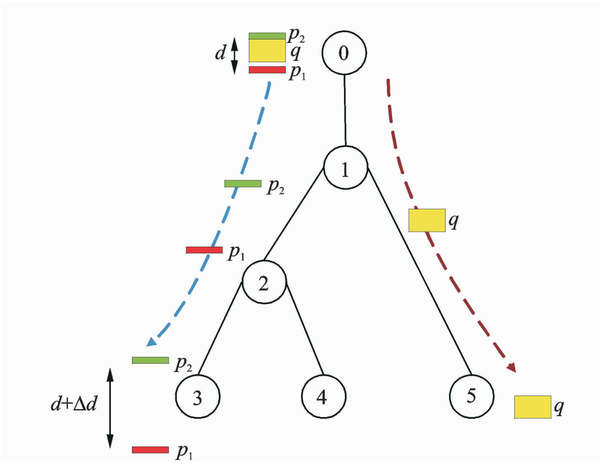


Fig. 1 Sandwich probe packets

ger the interval between p_1 and p_2 comes.

1.3 Calculation for similarity of node pair

The sandwich probe measurement method shows that the eventual interval between the two short packets depends on the queuing delay generated by the long packet in transmission, and is also related with the shared links between the long packet and the short packets. While in tree structure network, the more links a pair of nodes share, the greater their similarity is. We consider the queuing delay as the metric for the similarity of node pair and the queuing delay is based on the eventual interval between two short packets. In order to improve the accuracy, the calculation for the similarity of node pair is as follows: Suppose the source node sends sandwich probe packets to the target node pair (v_i, v_j), and the target node of the short packets is v_i while the target node of the long packet is v_j, and the initial interval between the two short packets is d. Suppose the source node sends N sandwich probe packets in total, and the time when the target node v_i receives the short packets is $T_{1,1}, T_{1,2}, T_{2,1}, T_{2,2}, \cdots, T_{N,1}, T_{N,2}$, then the similarity of node pair (v_i, v_j) is

$$S(v_i, v_j) = \frac{1}{N}\sum_{k=1}^{N}(T_{k,2} - T_{k,1} - d)$$

2 Network Topology Inference Based on Hierarchical Clustering

Network topology inference algorithm based on hierarchical clustering consists of the following three steps: First, the source node sends probe packets to all leaf nodes, and the leaf nodes are layered by time to live (TTL) fields of the received packets; Then each layer of leaf nodes is clustered by similarity clustering algorithm; Finally, according to the result of hierarchical clustering and the changing TTL value, the network topology is inferred.

2.1 TTL hierarchical algorithm

TTL field of 8 bit in IP datagram header is mainly used in TTL hierarchical algorithm. TTL indicates the amount of routers through which the packet passes at most, and it is also the lifetime of the packet in the network. As stipulated in IP protocol, router subtracts 1 from TTL field of the packet before forwarding it. If the TTL value is 0, the router will discard the packet and never forward it. TTL field is set by source point to prevent the waste of network resources caused by undeliverable packet forwarding indefinitely in the Internet. In practice, most OSes, including Microsoft Windows, Linux, and Unix systems, only select a few figures, including 32, 64, 128 and 255, as initial TTL value. The difference between an initial TTL value and its final TTL value is the number of routers which the packet goes through in the network, also known as the hop count. Since the differences between the above initial TTL values are large, and practically few Internet hosts are apart by more than 30 hops[13], one can determine the initial TTL value of a packet as the smallest one in the above set but larger than its final TTL. Therefore, firstly the source node sends probe packets to all leaf nodes with a set initial TTL values. And then final TTL fields of received packets in leaf nodes are recorded. Finally, leaf nodes are stratified according to the hop count between source node and the destination node.

2.2 Similarity clustering algorithm

Leaf nodes are divided into different layers according to the TTL hierarchical algorithm. Then leaf nodes are clustered by similarity layer by layer. First of all, the similarities of all node pairs in a certain layer are obtained by sandwich probe measurement and sorted ascendingly. Then minimum similarity set is calculated by variance ratio. The incompatible K-Bucket is built. Finally, leaf nodes are clustered.

2.2.1 Minimum similarity set

For a set of similarity values in ascending order, the first element is the lower bound of the minimum similarity set and the key is to find the upper bound of the minimum similarity set. Variance is used to measure the volatility of a batch of data. By analyzing experimental data, we discovered that the difference between elements in the minimum similarity set and other elements was generally large. Therefore, variance ratio is selected to obtain minimum similarity set.

The definition of variance is as

$$V=((x_1-\bar{x})^2+(x_2-\bar{x})^2+\cdots+(x_n-\bar{x})^2)/n$$

where $x_1, x_2, \cdots, x_n$ are the data, $\bar{x}$ is the average, and n the number of data. In our scheme, ascendingly-sorted data in similarity set are selected one by one to calculate variance ratio, and then the upper bound of a minimum similarity set is determined by the comparison between variance ratio and the threshold. After several experiments, we revealed that before joining the upper bound element of the minimum similarity set, variance ratios were within ten and with smaller changes, but after joining the upper bound element, a mutation of hundreds even thousands of variance ratio emerged. Therefore, the threshold of variance ratio is selected as 10 here. The specific algorithm in pseudo-code is described as:

Input: The number of leaf nodes on layer i: n

Similarity set of all node pairs on layer i:

$S_i=\{s_{1,2}, s_{1,3}, \cdots, s_{1,n}, s_{2,3}, s_{2,4}, \cdots, s_{n-1,n}\}$

The size of the set S_i: $L=n\times(n-1)/2$

The threshold of variance ratio: $R=K$

Minimum similarity set of leaf nodes on layer i: $M_i=\Phi$

Output: M_i

Begin

Sort similarity in S_i ascendingly and get similarity set in ascending order $S_A=\{s_{a,b}, s_{c,d}, s_{e,f}, \cdots\}$, where $a,b,c,d,e,f\in[1,n]$.

Select the first and the second smallest similarities from S_A (that is $s_{a,b}, s_{c,d}$) to calculate their variance V_2 according to the variance formula; In the same way select the first three smallest similarities from S_A (that is $s_{a,b}, s_{c,d}, s_{e,f}$) to calculate their variance V_3;

If $V_2\neq 0$

then calculate variance ratio of V_3 and V_2: $R_{3,2}=\dfrac{V_3}{V_2}$;

For $j=\{4,5,\cdots,L\}$ do

Begin

If $R_{j-1,j-2}\geqslant K$ then

Begin

Take $j-2$ as the upper bound of the minimum similarity set of leaf nodes on layer i;

Add the first $j-2$ smallest similarities from S_A to set M_i;

Break;

End

If $R_{j-1,j-2}<K$ then

Begin

Calculate the variance V_j of the first j smallest similarities from S_A to;

If $V_{j-1}\neq 0$

then calculate variance ratios V_j and V_{j-1}:

$$R_{j,j-1}=\frac{V_j}{V_{j-1}};$$

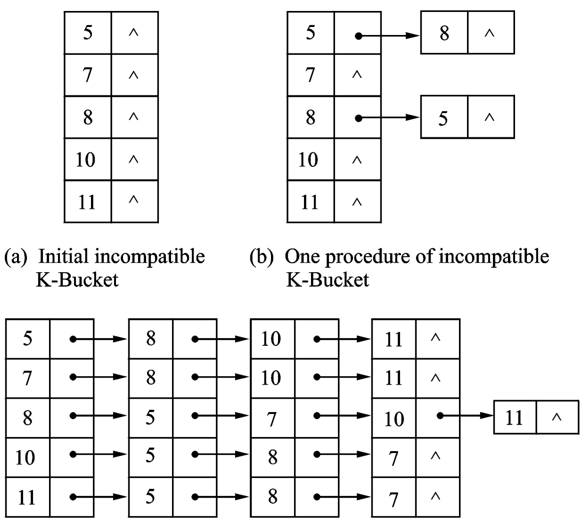
End;

End;

End;

2.2.2 Incompatible K-Bucket

Incompatible K-Bucket is an array of linked lists based on the minimum similarity set. The leaf node pairs in the minimum similarity set uniformly map to the incompatible relations between the first node and its subsequent nodes in each linked list, while the first node in each linked list constitutes the set of leaf nodes in layer i. Take the set of leaf nodes in a certain layer $\{5,7,8,10,11\}$ for example. Firstly, build initial incompatible K-Bucket and only one node leads each linked list, as shown in Fig. 2 (a). Assume that the minimum similarity set is {(5, 8),(5, 10),(7, 8),(8, 10),(7, 10),(5, 11),(8, 11),(7, 11)}. For the first node pair (5, 8), add incompatible node 8 to the linked list with the first node of 5, and add incompatible node 5 to the linked list with the first node of 8, as shown in Fig. 2(b). Similarly, add the rest node pairs in the minimum similarity to the lists and the final incompatible K-Bucket is shown in Fig. 2(c).



(a) Initial incompatible K-Bucket (b) One procedure of incompatible K-Bucket

(c) Final incompatible K-Bucket

Fig. 2 Establishment of incompatible K-Bucket

2.2.3 Leaf node clustering algorithm

Leaf nodes in layer i are clustered based on incompatible K-Bucket. First, suppose the leaf nodes in layer i are divided into two categories, set 1 and set 2, whose representative elements are the first and the second nodes of the first linked list in incompatible K-Bucket, respectively. For the leaf node in addition to the two representative elements, its incompatible nodes in incompatible K-Bucket (that is all subsequent nodes of the linked list where it is the first node) are compared with all the elements in set 1. If they are all different, then add the leaf node to set 1. If there is at least one same node, then compare its incompatible nodes with all the elements in set 2. If they are all different, then add the leaf node to set 2, otherwise build a new category set 3 and add the leaf node to set 3. By that analogy, we finally get the clustering sets of leaf nodes in layer i. The specific algorithm in pseudo-code is described as follows.

Leaf node clustering algorithm based on incompatible K-Bucket.

Input: the number of leaf nodes in layer i: n

Set of leaf nodes in layer i : $N_i=\{x_1, x_2, \cdots, x_n\}$.

Incompatible K-Bucket of leaf nodes in layer i : K_Buffer$_i$, where the order of the first node in each linked list is same as set N_i , that is $x_1, x_2, \cdots, x_n$.

Output: clustering sets of leaf nodes in layer i: Set$_1$, Set$_2$, $\cdots$.

First, suppose the leaf nodes in layer i are divided into two categories, Set$_1=\varnothing$ and Set$_2=\varnothing$. And the number of leaf nodes category is SetNum = 2. Assign the first and the second nodes (assumed to be a and b , $a, b \in N_i$ and $a \neq b$) of the first linked list (K_Buffer$_i$[0]) in incompatible K-Bucket to Set$_1$ and Set$_2$, respectively, that is, Set$_1=\{a\}$, Set$_2=\{b\}$.

```
    For j = {1,2,⋯,n}    do
Begin
    If x_j ≠ a and x_j ≠ b    then
        Begin
            For k = {1,2,⋯,SetNum}    do
            Begin
```

Compare incompatible nodes of the first node x_j of the j th linked list (K_Buffer$_i$[$j-1$]) in incompatible K-Bucket (that is all subsequent nodes after x_j of the jth linked list) with all the elements in Set$_k$ one by one;

If incompatible nodes of x_j and all the elements in Set $_k$ are all different then

```
                    Begin
                      add x_j to Set_k, that is
                     x_j ∈ Set_k ;
                      Break;
                    End
```

If there is at least one node in Set$_k$ same with x_j and incompatible nodes of x_j then

```
                    Continue;
            End
```

If incompatible nodes of x_j and the elements in existing SetNum categories all have the same then

Begin

SetNum=SetNum+1;

Build a new category $\mathrm{Set}_{\mathrm{SetNum}}$;

Add x_j to $\mathrm{Set}_{\mathrm{SetNum}}$, that is $x_j \in \mathrm{Set}_{\mathrm{SetNum}}$;

End;

End;

End;

2.3 Inference of hierarchical clustering network topology with changing TTL

Now a layered network topology which is clustered in each layer is obtained by TTL hierarchical algorithm and similarity clustering algorithm. It is essential that how to merge and connect the lower clustering set and the upper clustering set to obtain a complete network topology. For instance, Fig. 3 shows a hierarchical clustering network topology. The first layer is source node. The second layer contains two clustering sets A and B while the third layer contains sets C and D. There are 9 solutions to merge and connect the lower and the upper clustering sets together. Suppose it is corresponding to an unordered tree. There are 6 merge connection solutions which are shown in Fig. 4. Through analyzing the network topology structure in these solutions, we find that the number of shared links and shared routers are not the same between each clustering set in each solution. On basis of this and the traceroute method, we obtain the number of shared routers between each clustering sets through changing the TTL value of long sandwich probe packets to infer the network topology.

The sandwich probe measurement method shows that the queuing delay is generated by store-and-forward when the long packets pass through the routers, which is related with the number of shared routers. As long as the long packet and the short packets still share links, queuing delay will increase with an increasing number of shared routers. Once the long packet is separated from the short packets, queuing delay will remain unchanged as the second short packet is no long affected by long packet.

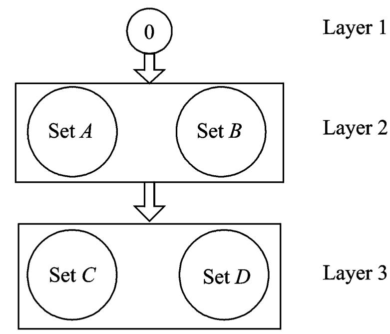


Fig. 3 Hierarchical clustering network topology

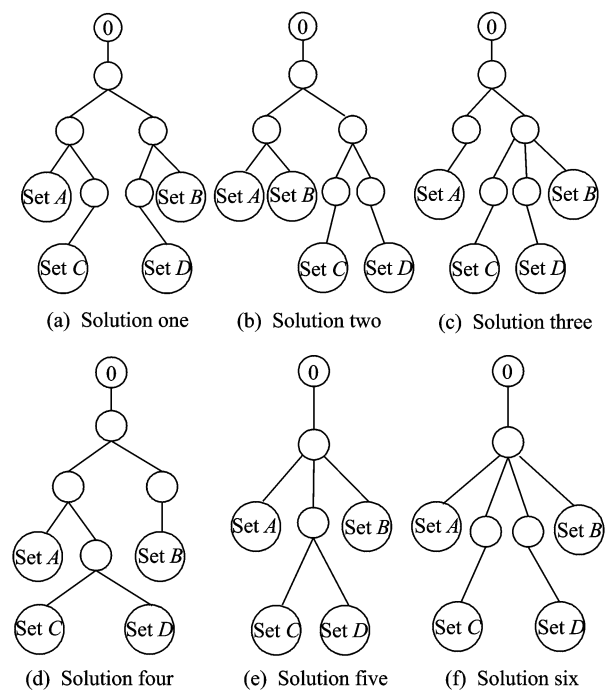


Fig. 4 Merge connection solutions to hierarchical clustering network

In our method, the source node sends sandwich probe. TTL value of the long packet starts from 1 and pluses one by one to two different clustering sets. Meanwhile, the similarity of the two clustering sets is calculated. The TTL value at the turning point where the similarity stops rising and begins to be unchanged equals to the number of shared routers of the two clustering sets. Since the TTL value of the long packet is no larger than hop count of nodes in clustering sets, with limited sandwich probe packets we can get the number of shared routers among all clustering sets and then infer the complete network topology.

3 Experimental Evaluations

In order to verify network topology inference method based on hierarchical clustering, we conducted simulations with lighter and moderate network load based on NS-2. 26 by controlling the size of background traffic. The simulation topology is shown in Fig. 5. Simulation results in the case of moderate network load are as follows.

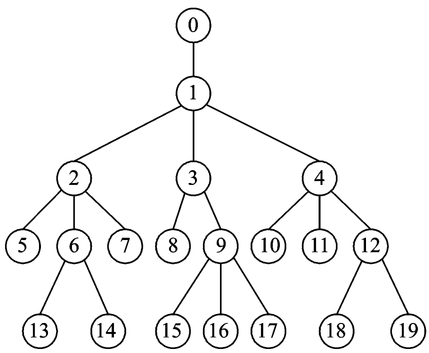


Fig. 5 The simulation network topology

The initial TTL value of the probe packet was set as 128 based on the TTL hierarchical algorithm. The final TTL value was recorded in leaf nodes. Thus the hop count was calculated. The layering result based on hop count is shown in Table 1 and the source node 0 was set as layer 1.

Table 1 Layering result of leaf nodes

Layer	Hop count	Set of leaf nodes
2	2	{5, 7, 8, 10, 11}
3	3	{13, 14, 15, 16, 17, 18, 19}

We sent sandwich probe packets to all node pairs in each layer and the similarity of node pairs are shown in Tables 2, 3.

Then we clustered the leaf nodes in each layer based on similarity clustering algorithm, and the clustering result is shown as follows(Fig. 6).

Table 2 Similarity of node pairs in layer 2

Node pair	Similarity	Node pair	Similarity
(5,7)	0.004 018 70	(7,10)	0.002 809 76
(5,8)	0.002 675 85	(7,11)	0.003 150 48
(5,10)	0.002 740 73	(8,10)	0.00 280 488
(5,11)	0.002 851 09	(8,11)	0.002 975 24
(7,8)	0.002 800 00	(10,11)	0.004 023 58

Table 3 Similarity of node pairs in layer 3

Node pair	Similarity	Node pair	Similarity
(13,14)	0.007 380 70	(15,16)	0.007 615 60
(13,15)	0.002 897 65	(15,17)	0.007 617 00
(13,16)	0.002 423 25	(15,18)	0.003 511 60
(13,17)	0.002 424 45	(15,19)	0.003 535 67
(13,18)	0.002 249 74	(16,17)	0.007 617 00
(13,19)	0.002 355 04	(16,18)	0.003 037 20
(14,15)	0.003 167 65	(16,19)	0.003 059 67
(14,16)	0.002 693 25	(17,18)	0.002 710 12
(14,17)	0.002 694 45	(17,19)	0.002 822 27
(14,18)	0.002 519 74	(18,19)	0.007 113 65
(14,19)	0.002 585 04		

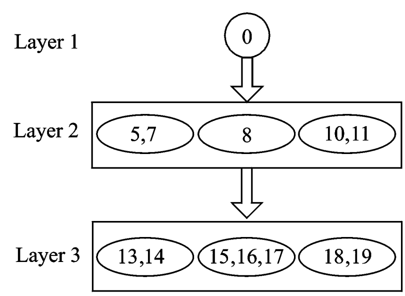


Fig. 6 Hierarchical clustering network topology

Each clustering set was represented by a capital letter, namely $A=\{5,7\}$, $B=\{8\}$, $C=\{10,11\}$, $D=\{13,14\}$, $E=\{15,16,17\}$, $F=\{18,19\}$. After successively sending sandwich probe packets with changing TTL value to the clustering sets, we obtained the similarity of each two cluster sets, as shown in Fig. 7. The number of shared routers between clustering sets A and D, B and E, C and F was two while the number of shared routers between the other clustering sets was one. It is easy to infer the whole network topology, as shown in Fig. 8, the number of shared routers.

Compare Fig. 8 with Fig. 5, the network topology inferred from network topology inference method based on hierarchical clustering is exactly the same with origin network topology. And topology inference method based on hierarchical clustering for 4-layer network topology can be verified. In order to verify inference algorithm

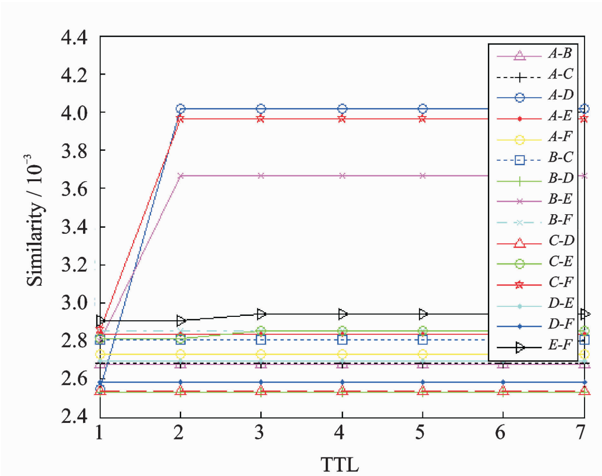


Fig. 7 Variation of similarity between clustering sets and changing TTL

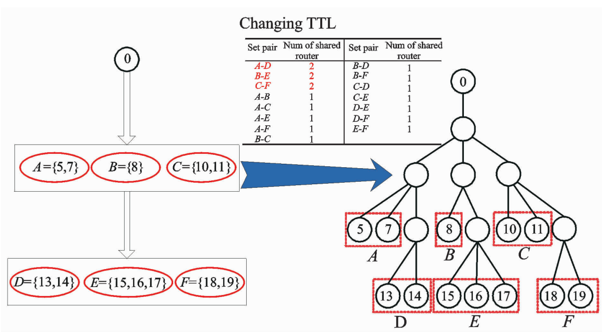


Fig. 8 Network topology inference based on hierarchical clustering network topology

when increase the level of network topology, different simulations were conducted for 5-layer and 6-layer network topology under lighter and moderate network loads, and the simulation results are shown in Table 4.

Table 4 Topology inference accuracy rate with different topology levels under different network loads

Topology level	Lighter load	Moderate load
4	100	100
5	100	93.33
6	86.67	73.33

Accuracy rate of hierarchical clustering network topology inference algorithm in lighter load case was higher than that of moderate load case. We infered that the algorithm works well in lighter load network environment. Table 4 also shows that as the network level increases, accuracy rate of the algorithm will be slightly reduced, but still relatively high.

We further discovered that the algorithm could reduce the measuring flow. Traditional network topology inference based on network tomography has to measure the similarity of all node pairs with $(N^2-N)/2$ measuring flow when the network topology contains N leaf nodes. However, the proposed method assumes that the leaf nodes are divided into T layers and each layer contains M leaf nodes, and $N=MT$. After similarity clustering, all leaf nodes are divided into S clustering sets. The measuring flow includes two parts: One part is the similarity of node pairs in each layer, where the flow is $(M^2-M)/2$, so the flow $(M^2T-MT)/2$ with T layers; the other is the similarity of all the clustering sets where the flow is $(S^2-S)/2$. Thus the total measuring flow is $(M^2T-MT+S^2-S)/2$. In general, $M\ll N/2$, $S\ll N$, so there is S^2N^2, $S\ll N^2/2$, $S^2-S\ll N^2/2$, $(M^2T-MT)=(NM-N)\leqslant\left(\frac{N^2}{2}-N\right)$. Then $\frac{1}{2}(M^2T-MT)+\frac{1}{2}(S^2-S)<\frac{1}{2}N(N-1)$. The complexity is $O(N^2)$.

While in Refs. [9, 10], the complexity is dominated by the calculation of the merging options, which corresponding to $O(|EVT|^2\times|VVT|^3)$, where $|VVT|$ the number of nodes in the network virtual topology and $|EVT|$ the number of links connecting them. $|VVT|$ equals to N. Compared with these network topology inference based on network tomography, our method can basically ensure high accuracy of topology inference with effectively decreasing measuring flow, thus improves the efficiency of topology inference.

4 Conclusions

A unicast network topology inference algorithm based on hierarchical clustering is proposed and the simulation test is conducted on the NS-2. The experiment result shows that the network topology inference can basically ensure high accuracy of topology inference while effectively decreases measuring flow, thus improves efficiency of topology inference. How to further infer the physical topology of network is our future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 61373137, 61373017, 61373139), the Major Program of Jiangsu Higher Education Institutions (No. 14KJA520002), the Six Industries Talent Peaks Plan of Jiangsu (No. 2013-DZXX-014), and the Jiangsu Qinglan Project.

References:

[1] Luckie M, Hyun Y, Huffaker B. Traceroute probe method and forward IP path inference[C]// Proceeding of the 8th ACM SIGCOMM. New York, USA: ACM, 2008: 311-324.

[2] Jin X, Tu W, Chan S H G. Traceroute-based topology inference without network coordinate estimation [C]// IEEE International Conference on Communications. Washington, DC, USA: IEEE, 2008: 1615-1619.

[3] Zhao Honghua, Chen Ming. Topology Inference Based on Network Tomography [J]. Journal of Software, 2010, 21(1):133-146 . (in Chinese)

[4] Tian H, Shen H. Hamming distance and hop count based classificcation for multicast network topology inference[C]// 19th International Conference on Advanced Information Networking and Applications. Washington DC, USA: IEEE, 2005 (1): 267-272.

[5] Zhao Honghua, Ding Ke, Chen Ming. Topology inference algorithm by using one measuring node [J]. Journal of University of Electronic Science and Technology of China, 2010, 39(2): 275-278. (in Chinese)

[6] Zhao Honghua, Chen Ming, Qiu Xiaofeng, et al. Multiple parameters network topology inference based on tomography [J]. Journal of Beijing University of Posts and Telecommunications, 2008, 31(4): 24-28. (in Chinese)

[7] Su H B, Li Y, Lin S J, et al. A sort-based approach to infer the network topology[C]// 2010 IEEE International Conference on Communications. Washington, DC, USA: IEEE, 2010: 1-6.

[8] Eriksson B, Dasarathy G, Barford P, et al. Efficient network tomography for internet topology discovery [J]. IEEE Transactions on Networking, 2012, 20 (3): 931-943.

[9] HolbertB, Tati S, Silvestri S, et al. Network topology inference with partial information [J]. IEEE Transactions on Network and Service Management, 2015, 12(3):406-419.

[10] Holbert B, Tati S, Silvestri S, et al. Network topology inference with partial information[C]// International Conference on Computing, Networking and Communications, Network Algorithm & Performance Evaluation Symposium. Washington, DC, USA: IEEE, 2015:796-802.

[11] Zhang Runsheng, Li Yanbin, Li Xiaotian. Topology inference with network tomography based on t-Test [J]. IEEE Communications Letters, 2014, 18(6): 921-924.

[12] Castro R, Coates M, Liang G, et al. Network tomography: Recent developments [J]. Statistical Science, 2003, 19(3): 499-517.

[13] Wang H, Ding W, Zhu H. A method of tree network topology inference based on hierarchical host table[C] // Systems and Informatics (ICSAI), 2012 International Conference on. USA: IEEE, 2012: 1477-1481.

(Executive Editor: Zhang Bei)