Model for Advanced Sandwich Probe Topology Inference Scheme in Network Tomography

Mrs. Vaishali Navnath Pansambal, Prof.R.H.Kulkarni.

Abstract— To improve the performance of the network, we have to examine the network and its function; we studied about what are the blockage in the network, and studied about the resources which are not consumed properly. For improving the performance of the network after examine the functionality of the network we design the solution. We used the Trace-Route utility tool for getting the internal information of the network, this tools required cooperation of the internal nodes in the network. In the public networks like the routers of the internet, it required cooperation of the internal node in the network, this functionality is not exist in Trace route for that reason we can't used trace route in the public network. The term network tomography means the study of internal characteristics of the network by using the information derived from the end points of the network. In this paper, we proposed a method called as Advanced Sandwich Probe. This method collects the needed information from the network to generate the topology from the collected information. The proposed method is the extension of the sandwich probe method with the functionality of Trace-route method. Simulation result of the proposed method as comparing with the previous method shows that the method improves the topology identification process. The proposed method overcomes the functionality of existing method with the variation of differences.

Index Terms—Network tomography, Topology inference, Sandwich probe scheme.

1. INTRODUCTION

Distributed Internet applications often need to know information about the characteristics of the network. Network administrators require some important network performance parameters, such as loss rate, link delay, etc., in order to monitor, predict, and diagnose the state of the network and provide better service to their users. However, with the Internet becoming more heterogeneous and unregulated, it is challengeable to acquire such information. Although administrators of small-scale networks can monitor local traffic conditions and identify congestion points and performance bottlenecks, very few networks are completely isolated. The user perceived performance of a network thus depends heavily on the performance of an internetwork, and monitoring this internetwork is extremely challenging. [1] Currently most of the well known network topology and behavior identification tools such as Traceroute need help from internal nodes of the network, such as routers.

- *Mrs. Vaishali Navnath Pansambal is currently pursuing masters degree program in Computer engineering in Pune University, India, E-mail: vaishu.bscoer@gmail.com*

They send requests and messages to the internal nodes and receive feedback from those nodes. However, one cannot rely on the cooperation of nodes administered by someone else. For example, Traceroute uses Internet Control Message Protocol (ICMP) error messages sent by the routers. On a public network like the Internet, routers are often configured not to respond to ICMP, and not to send these messages [2]. Depending on the cooperation of internal nodes, and relying on specific settings and configuration details within them, makes this approach very limited. Due to the problems with cooperation-based methods, some researchers have started to work on the problem of inferring internal network structure and statistics without relying on internal nodes.

Network tomography is the study of a network's internal characteristics using information derived from end point data. The word tomography is used to link the field, in concept, to other processes that infer the internal characteristics of an object from external observation, as is done in magnetic resonance imaging or positron emission tomography (even though the term tomography strictly refers to imaging by slicing). The field is a recent development in electrical engineering and computer science, founded in 1996.Network tomography advocates that it is possible to map the path data takes through the Internet by examining information from "edge nodes," the computers where data is originated and requested from. To get all needed information about network, first step is to infer network topology. [3] Our paper focuses on inferring network topology by using information derived from end point data. We focus on the case where we have one source and several receivers, and we use uni-cast end-to-end messages to probe the network. Also

we assume the routes in the network are fixed during the time that these messages are sent.

2. RELATED WORK

In network tomography, we infer network information by using information derived from end points of network. So in network tomography, we first gather information from end points in the network i.e. Data gathering phase and then we use this gathered information to infer topology of the network i.e. Network topology inference.

2.1 Data Gathering Phase

Data collection methods for network topography depend on whether internal nodes in network are cooperating or not. If the network on which the tomography method is performed is a public network, and we do not have control of the internal nodes (routers, switches, etc.), we can only expect a limited amount of information from those nodes [2].If we have our own private network we can program the internal nodes to collect and report the information we need. If internal nodes are cooperating then it is easy to gather network information than when internal routers are not cooperating.

2.2 Active Data Gathering

Active methods send probe messages across the network to elicit information. Active methods can use end-to-end probes so they do not need to rely on help from internal nodes. The downside of active methods is that they impose extra load on the network; these methods try to keep the extra load as low as possible so as not to affect the network's quality of service. The probes sent by active methods can be multicast or unicast messages. Although not all networks support multicast, some methods require multicast messages to collect the data they need [4]. Also there are some methods that try to optimize the size of the data sent with each probe using approaches like network coding [4], [5].

2.3 Passive Data Gathering

Passive methods do not interfere with the network's normal traffic – they just act as monitors and try to reach a conclusion using the available characteristics of the traffic. This is only possible if we are able to use the internal nodes to collect data.

2.4 Topology Inference

There are two major two types of methods for inferring network topology. First is maximum likelihood method and another is constructive method.

2.5 Maximum-Likelihood Approach

Vardi was one of the first researchers working on network tomography. He was working on the problem of estimating end-to-end path properties using link measurements. In 1996 he suggested the following equation to model the problem of estimating the number of packets sent between each pair of end nodes [12].

$$Y = AX, \qquad (1.1)$$

Where Y is a vector of the measured data, which is the number of packets passing through each direct link during a specific time period. So Y has an element for each direct link of the network. X is a vector containing the amount of traffic on each end-to-end path during the time period. A is a binary routing matrix. Let p be the number of end-to-end paths whose traffic is being estimated and $|L|$ is the number of direct links in the network. A is a $|L| \times p$ binary matrix in which the element in row i and column j is 1 if and only if i-th end-to-end path contains the J-th direct link. Some other researchers used this model for other network tomography problems. In the problem Vardi was working on path characteristics were desired and link characteristics were given. In other problems with the same model, link characteristics (delay, loss, etc.) or the routing matrix may be desired. The measurements are usually path-based or link-based. Y = (Y1, Yt) is a vector of the measured data, X = (X1, Xs) is the vector of parameters to be estimated and A is a $|t \times s|$ binary routing matrix. Usually Xi is a random variable parameterized by some Qi and Qs = (Q1... Qs) is what we need to estimate. Maximum-likelihood approaches are relatively computationally complex, and they are feasible only for small routing trees. Therefore, there are some methods to deal with the trade-off between complexity and accuracy of these approaches. One of these ideas is to use pseudo-likelihood models. This means dividing the problem described by Eq. 1.1 into several sub-problems, solving each separately, and then combining the solutions to get to the final solution. For the topology inference problem it means finding the logical topology induced by different groups of receivers and then merging the topologies to create the complete logical topology.

2.6 Constructive Methods

The constructive algorithms require much less computation than the maximum-likelihood approaches, but usually give less accurate results [6]. There are several proposed constructive methods; most of them have a similar bottom-up approach. First, we find a group of leaves that seem to be neighbors. Then we join them as a single node, set proper estimated values for the new node, and continue until only a single node remains. Coates et al. [6] proposed such a

method in 2004 named ALT. ALT uses unicast measurements and group's two nodes in each step so the resultant tree is always a binary tree. Therefore, this method is suited only for inferring binary topologies. Ratnasamy et al. [7] suggested a similar method which uses multicast packet loss measurements. In 2002, Duffield et al. [4] extended Ranasamy's algorithm to make it suitable for general, non-binary trees. More recently, Ni et al. [8] suggested a joining algorithm for general trees which uses unicast measurements. They also suggested a dynamic algorithm to add or delete a receiver from the network. Using their method, it is possible to add nodes one by one and construct the whole topology.

3. EXISTING METHOD

3.1 Sandwich Probe

Sandwich probe [[9] is active type of tomography technique, in which we send probes in the network to get information about network. Probe is an object or message used for the purpose of learning something about the state of the network. For example, an empty message can be sent simply to see whether the destination actually exists. Ping is a common utility for sending such a probe. Sandwich probe measure delays of common paths for every pair of nodes in network and delay differences are measured at single node, so no time synchronization is required. Sandwich probe method works as follows.

Assume we have network as shown in figure 1. As shown in fig, network has two receivers namely 2 and 3, they have common segment from root node to the 1 node. Aim of sandwich probe is to find to measure delay on this common segment i.e. from 0 to node 1.Probe used in this method made up of three packets. Two of them are small packets namely p1, p2 and one large packet name q.

Step 1) Send small packet p1 to the node 2, when p1 received at node 2 record time at which it has arrived.

Step 2) After step1 wait for some time d and then send large packet q to node 3,after that immediately send packet p2 to the 1. Packet *p*2 gets queued behind *q* in every node from the 0 to *1*, where their paths diverge. As a result, it takes more time for *p*2 to be delivered than *p*1.Record time when packet p2 is received, assume it reached at 2 after t+Δt.

Then Δt is delay for the common segment (root to n3) for nodes n1 and n2. For two receivers *i* and *j*, we define Xi, j as the estimated delay of their common path to the root

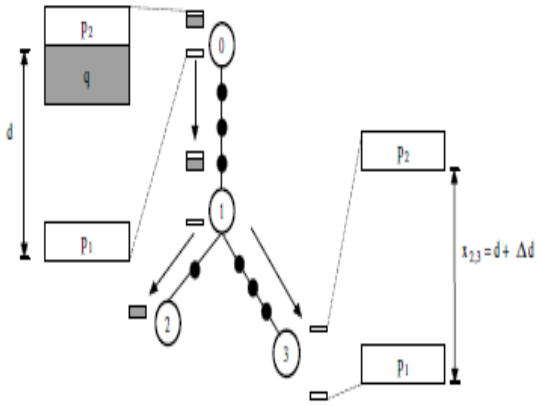


Fig. 1 an example of sandwich probe measurement. The large packet is destined for node 2, the small packets for node 3. The black circles on the links represent physical queues where no branching occurs. In the absence of cross-traffic, the initial spacing between the small probes d is increased along the shared path from nodes 0 to 1 because the second small probe p2 queues behind the large packet. The measurement x2, 3 for this receiver pair is equal to d + Δd. A larger initial spacing d reduces the chance of p2 catching p1 because of a bottleneck or cross-traffic on the path from node 1 to 3

A. Traceroute Technique

Above sandwich probe technique gives very less information about network. The Traceroute [10] utility is one of the most commonly used, not to mention useful, diagnostic tools available to any network operator. Traceroute allows you to examine the path a packet takes across the Internet, showing you each of the individual routers that handle the packet, as well as measuring the time (network latency) it takes to deliver the packet to each router. Looking at a Traceroute is similar to having a bird's eye view of a car drive from one location to another, showing you each of the roads (paths) and intersections (routers) encountered along the way.

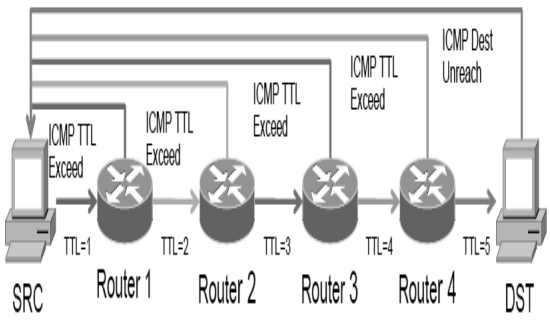


Fig. 2 Traceroute Working

Every Traceroute probe follows this basic pattern:

- Traceroute launches a probe packet towards the final destination, with an initial TTL value of 1.
- Each router that handles the packet along the way decrements the TTL by 1, until the TTL reaches 0.
- When the TTL value reaches 0, the router which discarded the packet sends an ICMP TTL Exceed message back to the original sender, along with the first 28 bytes of the original probe packet with its own IP address.
- The Traceroute utility receives this ICMP TTL Exceed packet, and uses the time difference between the original probe packet and the returned ICMP packet to calculate the round-trip latency for this router "hop".
- This process again from step 1, with a new initial TTL value of N+1, until destination comes.

4. PROPOSED METHOD: ADVANCE SANDWICH PROBE

Proposed scheme is combination of current Sandwich probing scheme and Traceroute technique, so we saw what sandwich probe is and Traceroute technique in above scheme, now we will see our proposed scheme Advanced Sandwich Probe method for Data gathering and then we will constructive method to infer network topology.

We assume that in our work internal nodes in the network are not cooperating, means they don't send reply to the ICMP packets, so we can't use Traceroute technique and sandwich probe method alone is not sufficient for inferring topology of network, so we have combined this above two techniques to infer topology of network and named Advanced Sandwich Probe with Traceroute.

4.1 Data Gathering:

In our scheme we first estimate all Xi, j i.e. delay of common path for all receivers in the network same as per sandwich probe method described above.

In second step, we send sandwich probe to single receiver(i) instead of two different nodes and initially set TTL value of large packet(q) is to certain value k, after k hops q will be drop, and we measure delay in p1 and p2 i.e. Yi,k. After that we will increment value of TTL of q packet (suppose TTL=k+1) then q will dropped after k+1 hops and again we will measure delay in packets to reach i. This process of incrementing TTL value of q and measuring delay in packets to reach i is continued until q reaches to i.

We do above step for each receiver in the network and get Yi, k for each receiver. Example of measuring Yi,k is shown in fig 3.As shown in figure we set TTL of q packet =2 i.e. after two hops packet q will be drop and delay in packets to reach final node is estimated. Now value of k is incremented by 1 i.e. 3 and again time delay is measured this process is continued till q reaches successfully to final node.

The time measurements are performed in the end node instead of the starting node. This method gives us one way delays to the nodes along the path, as opposed to the round trip delay times elicited by Traceroute. The ASP does not give us any information about the addresses of the nodes along the path. Delay varies as per hop so we take multiple reading and takes average of it.

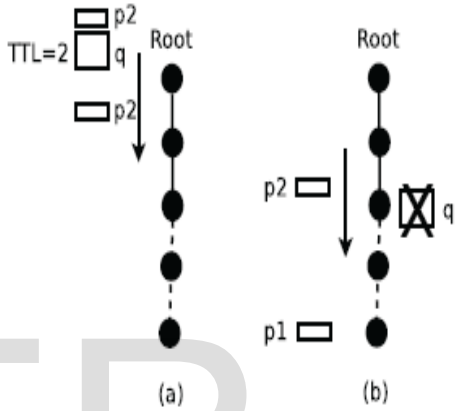


Fig. 3 Step2 in Advanced Sandwich Probe

4.2 Topology Inference

When Data gathering is completed we have values if Xi,j and Yi,k for all receivers in the network. Xi,j contains common segment delay for each receiver pair in network. Yi,k is the set of delay difference for packets for each receiver. We use constructive method for inferring topology of the network. We start with a tree containing only the root, and try to add leaves to the tree one by one. Assume we have a tree, which is a partial tree of the whole network. Assume we want to add a new leaf n to the tree.

First we find the leaf n' in the tree that maximizes Xn', n. The paths from the nodes n and n' to the root have a common segment, and Xn', n is our estimate of the delay along that segment. We need to find out at which node in the path from n' to the root these two paths separate. The delay from the separation node to the root has to be close to Xn', n. So we find the number k which minimizes

$$|Xn', n - Yn', k|,$$

Which means the k-th node in the path of the root to n' has the closest estimated delay to Xn', n. From above we got common segment in n and n' and where they are separating,

so we can add n to the network. Same like this we have to add node one by one and infer network topology.

Figure 4 shows the network topology we are studying. The input of the inference algorithm includes sandwich probes' output i.e., pair wise common path estimations or X = (xi, j) and Y = (Yi, k). Table 1 and Table 2 show X and Y calculated from Figure 4. In Table 1 the element in row i and column j shows xi, j. In Table 2 each row represents the measurements for one leaf. In row i the first element is yi, 1, the second one is yi, 2, etc.

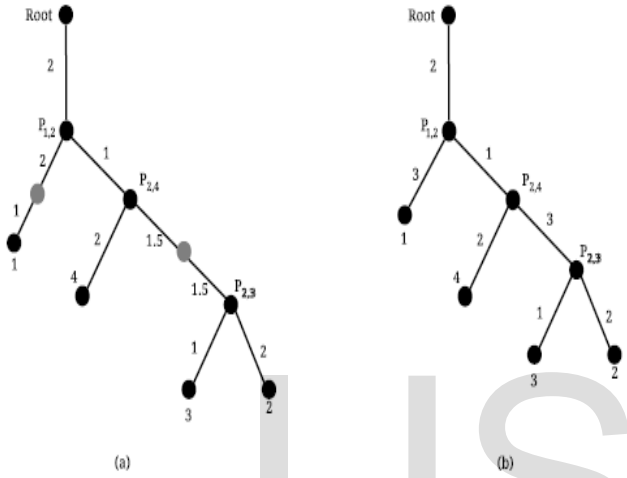


Fig. 4 Actual Tree

Table 1 Actual x values of the tree

$X_{i,j}$	1	2	3	4
1	0	2	2	2
2	2	0	6	3
3	2	6	0	3
4	2	3	3	0

Table 2 Actual Y values of tree

$X_{i,j}$	1	2	3	4
1	0	1.90	2.00	2.05
2	1.90	0	6.10	2.90
3	2.00	6.10	0.0	3.10
4	2.05	2.90	3.10	0.0

Table 3 Estimated X values of tree

$Y_{I,K}$	
1	1.90 3.8 4.90
2	1.90 3.05 4.60 5.90 7.85
3	2.00 2.90 4.45 6.10 6.90
4	2.10 3.10 4.90

Table 4 Estimated Y values of the tree

We start by adding a single node as the root and the leaves are added one by one to the tree. Inserting the first node gives us the tree in Figure 5-a, the number of the nodes and the link delays come from the first row in Table 4. The length of the first link is 1.9, which is the first element of the row. The length of the second link is the difference between the second and the first elements i.e., 3.8−1.9 = 1.9 and the length of the third one is the difference between the fourth and the third elements. To insert the second leaf we first look into Table 3 and find X (1, 2) which is 1.9. Now we should find the node in the path from leaf 1 to the root whose distance to the root is closest to 1.9. This is the first node from the root and is called P (1, 2) in Figure 5-b. This means the paths of leaf 1 and 2 separate at this node. Now using the second row of Table 4 we create the rest of the path of leaf 2 and we get the tree in Figure 5-b. The separation point is P (1, 2) whose distance to the root is 1.9. The distance from next node in the path of the leaf 2 to the root is 3.05, so the length of the next link is 3.05 − 1.9 = 1.15 and the rest of the path is created with the same process for the first leaf.

Finally to add the last leaf, the procedure is the same as for the third one. First we find the node n in Table 3 that maximizes x (4, n) which is leaf 3. The node with the closest

distance to x (4, 3) is the node called P (2, 4) in Figure 5-d; x (4, 3) = 3.1 and the root's distance to P (2, 4) is 3.05. According to Table 4 the path of the leaf 4 contains one more link and we create that using the fourth row in Table 4 The final result of the algorithm is shown in Figure 5-d.

Yi,k	
1	2 4 5
2	2 3 4 6 8
3	2 3 4 6 7
4	2 3 5

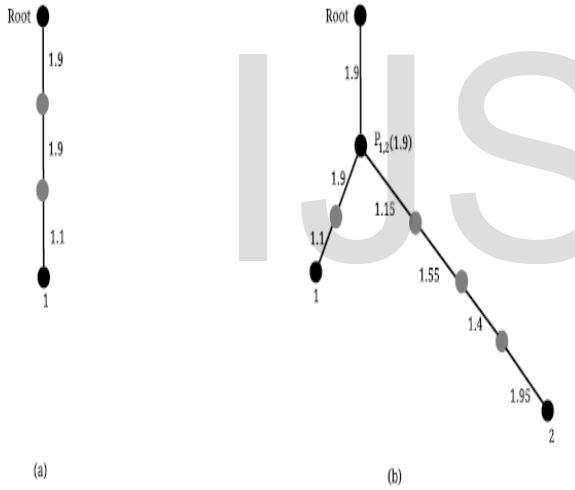


Fig. 5 (a,b) Example of topology inference

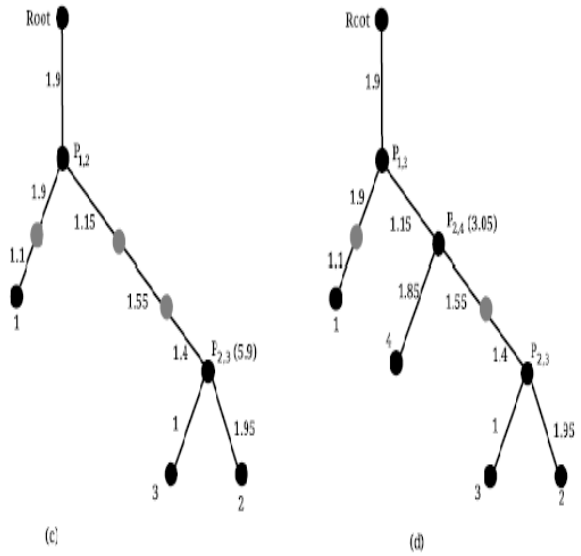


Fig. 6 (c, d) Example of topology inference

5. IMPLEMENTATION DETAILS

5.1 Modules

The proposed application is implemented in the following environment

Simulation Set: The application is implemented with the help of java language. Jung libraries are used for the network topology. For generating the sensor nodes and networks the Jung libraries are used.

Network: In the proposed network system maximum 50 nodes are used for implementation and simulation. These nodes are dynamically loaded.

There are mainly modules of the proposed system. The introductions of these four modules are as follows:

1. Formation of Networks:

The network with minimum 50 nodes is formed. These nodes are dynamically loaded. The nodes are created by using the Jung libraries. These nodes are sends the data from source to destination.

2. Get X Matrix:

This X matrix contains common segment delay for each receiver pair in network. The process of data gathering is completed when we get the values of x and y matrix for all receivers of networks. It means simply the data gathering is the combination of x and y matrix

3. Get Y Matrix:

The Y matrix is the set of delay difference for the packets for each receiver. We get the result of data gathering when we get the result of X matrix and Y matrix for all the receivers' networks.

4. Inferring Topology:

The output of data gathering is used as an input for this process. The constructive method is used for inferring the topology of the network.

5.2 Hardware Requirement

- Hard disk : 80 GB
- RAM : 512 MB
- Processor: Intel Pentium4 or above.

5.3 Software Requirements

A. JAVA

The technology used for designing and implementation of this project is java as a coding language. We use the vector class for implementing the algorithm. The GUI designing uses the swing class. Here we use 1.7 version of jdk. These are used for developing the java applications and applets. It is one of the software development environments.

NetBeans IDE

NetBeans IDE are installed for implementing the java code. NetBeans is an integrated development environment (IDE) for developing primarily with Java, but also with other languages. The NetBeans IDE is written in Java and can run on any platforms supporting a compatible JVM.

B. Jung Tool and Library for Forming Networks on the Frame.

Java Universal Network/Graph Framework is a software library, which is used for visualization, analysis the data which is represented as a graph or network. It is written in java. It is used to design directed or undirected graph, graph with parallel edges, multi-model graph etc. It is an open-source library; JUNG provides a common framework for graph/network analysis and visualization.

C. XML

XML stands for Extensible Markup Language which is much like HTML. It was designed to carry data, not to display data. XML tags are not predefined. You have to define your own tags. Here we use XML files to store the network.

5.4 Network Model

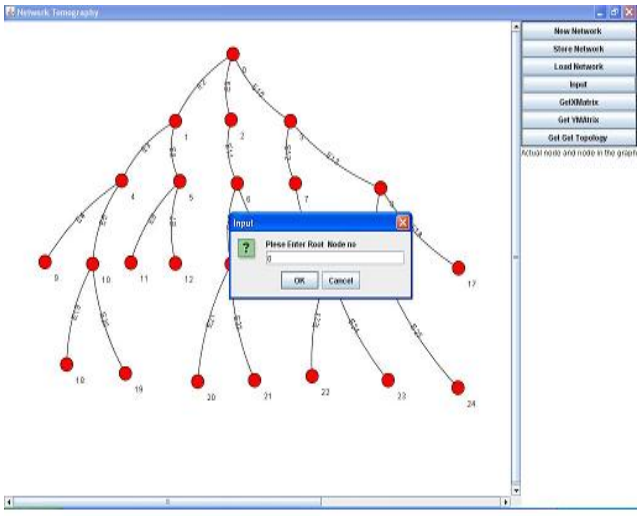
In the proposed network model, we can add new network. We can also form a new network. We can form different types of network model. Here we form a tree network model in which contain root node which is also called as parent node and next is child node. The end nodes also called as leaf nodes. In tree network model contains different levels like level 1; level 2, etc depend on network model. In network may contain nodes which are denoted as a router, hubs, switches etc in the network. It also contains edges which are represented as a link in a network model. We store this network and also use for further process. We can store and load the network in which contain nodes and edges up to 60.

5.5 Simulation

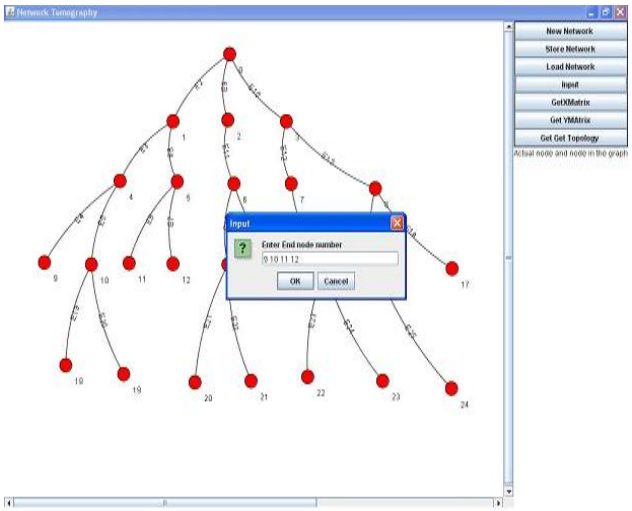
Select the root node from the network model. We can select any node as a root node. Here we select 0 as a root node from network model. Also select the end nodes from the network model. With the help of root node and end node, get a Get Xmatrix. It is useful for data gathering. It contains common delay path for each receiver pair in network. Data gathering is the combination of X and Y matrix. The completion of the data gathering process is done by getting the values from x and y matrix. Next step is Get Ymatrix; the set of delay difference for the packets for each receiver is get from the Ymatrix. We get the topology by calculating the Xmatrix and Ymatrix. And we get the inferring result. In inferring topology process, the output of data gathering is give as an input to the inferring topology.

5.6 Screen Shots

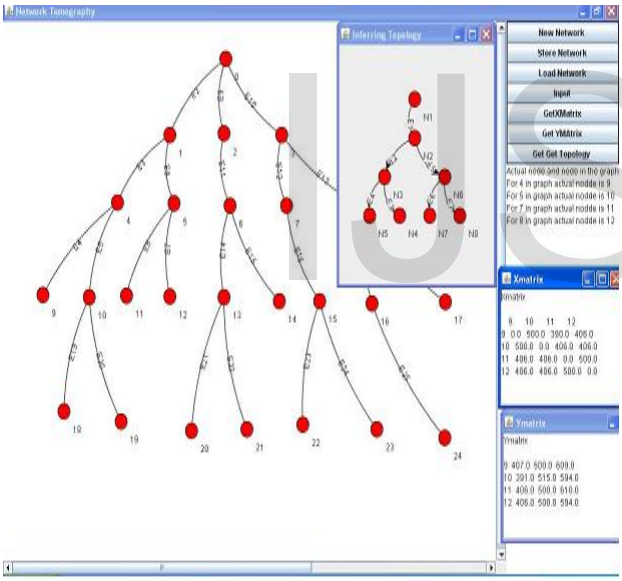
5.6.1.1 Enter root node : 0



5.6.1.2 Enter End Node :9 10 11 12



5.6.1.3 Infer Topology



6 CONCLUSION

In this paper, we have monitored the performance of the network with the help of different techniques. We have used utility tool like Traceroute to get internal information of the system. We proposed the new technique called Advanced switching probe to gather important evaluation of the network.

The performance result shows that our proposed work overcomes the existing work with variations of differences.

REFERENCES

[1] Network Tomography: Recent Developments Rui Castro, Mark Coates, Gang Liang, Robert Nowak and bin Yu.J. Clerk Maxwell, a Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] B. Yao, R. Viswanathan, F. Chang, and D. Waddington, "Topology inference in the presence of anonymous routers," in INFOCOM 2003. Twenty-Second Annual Joint Conferences of the IEEE Computer and Communications. IEEE Societies, **vol**. 1. IEEE, 2003, pp. 353–363K. Elissa, "Title of paper if known," unpublished.

[3] N. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley, "Multicast topology inference from measured end-to-end loss," Information Theory, IEEE Transactions on, vol. 48, no. 1, pp. 26–45, 2002.

[4] N Duffield, J. Horowitz, F. Lo Presti, and D. Towsley, "Multicast topology inference from measured end-to-end loss," Information Theory, IEEE Transactions on, vol. 48, no. 1, pp. 26–45, 2002.

[4] M. Gjoka, C. Fragouli, P. Sattari, and A. Markopoulou, "Loss tomography in general topologies with network coding,"in Global Telecommunications Conference, 2007. GLOBECOM' 07. IEEE. IEEE, 2007, pp. 381–386.

[5] C. Fragouli and A. Markopoulou, "A network coding approach to network monitoring," 2005.

[6] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: recent developments," Statistical Science, pp. 499–517, 2004

[7] S. Ratnasamy and S. McCanne, "Inference of multicast routing trees and bottleneck bandwidths uses end-to-end measurements," in INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings IEEE, vol. 1 IEEE, 1999, pp. 353–360

[8] J. Ni, H. Xie, S. Tatikonda, and Y. Yang, "Efficient and dynamic routing topology inference from end-to-end measurements," IEEE/ACM Transactions on Networking (TON), vol. 18, no. 1, pp. 123–135, 2010.

[9] 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 Performance Evaluation Review, vol. 30, no. 1,
pp. 11–20, 2002.

[10] http://cluepon.net/ras/traceroute.pdf