# NetTomo: A Tomographic Approach towards Network Diagnosis

Paritosh Ramanan Goutham Kamath Wen-Zhan Song
Department of Computer Science
Georgia State University
{pramanan1,gkamath1}@student.gsu.edu, wsong@gsu.edu

***Abstract*—Network diagnosis is a vital aspect in ensuring an efficient and robust functioning of any kind of mesh network. In this paper we present a network diagnosis method which determines the delay map of a mesh network using only end-to-end delay measurements without having the knowledge of the path taken. We model the problem of network diagnosis as an inverse problem and using a concept of ray tracing, solve for the delay in the network. With the help of simulations we show that our algorithm is able to detect nodes in the network based on their delays with reasonable accuracy using only $O(n)$ probes for obtaining measurements. We further demonstrate a real world application of our algorithm in the domain of internet backbone networks by using data pertaining to a major US based network provider.**

***Index Terms*—Network Diagnosis, Network Tomography, Mesh Networks, Linear Algebra, Inverse Problem**

## I. Introduction

Any network typically consists of two important components $nodes$ which process or generate data and $links$ which form the physical connection between different nodes. It is through these $links$ that the various $nodes$ in the network communicate and the flow of information takes place. Network Diagnosis is essentially a process for identifying faulty components within the network with the help of prior information about the network. *Network diagnosis* plays a crucial role in network maintenance and is extremely important for robust performance of the network.

The process of network diagnosis is heavily reliant on gathering consistent information about the network. The information being gathered can be relating to several factors such as the transmission delay at each node, the throughput of each node or the residual energy level at each node. Using such measurements, network diagnosis techniques aim to resolve robustness problems in the network.

One of the ways of performing network diagnosis is through $network\ tomography$ [1]. *Tomography* is the process of reconstructing a cross sectional image of an entity based on observed external parameters. A cross sectional image is constructed by passing rays through an entity and observing its properties before and after its passage. In the case of network tomography, the rays are nothing but probe packets. While traveling from source to destination, the path the probe packet takes depends on parameters like *delay*. By obtaining the end-to-end measurements of such parameters, one can create a cross sectional view of the entire network.

Measurements can be collected from the network in two ways: $actively$ or $passively$ [2]. Active measurement infuses the network with additional overhead tasks to obtain data while passive measurement infers data from the normal functioning of the network. The decision to use active or passive measurement techniques depends on the type of network, the degree of precision required and also the capability of the network to withstand measurement overhead. The fundamental difference between active and passive methods is that of the overhead cost of performing the measurement. In case a very precise and time sensitive analysis is required, then an active measurement is generally preferred. However in case an overhead on the network cannot be tolerated, then it makes more sense to perform a passive measurement. Active measurement involves a trade-off in favor of precision against the overhead. Passive measurement on the other hand offers a trade-off in favor of reduction in overhead costs on the network.

In this paper we present a network diagnosis technique which uses active probing to perform network tomography and finally yields a delay map of the entire network. The key contribution is that, without knowing the path taken by probe packets, we can determine the delays of nodes with good accuracy, by starting out with an initial random guess, using only $O(n)$ probes, where $n$ is the number of nodes in the network. Our technique can be used to diagnose different types of mesh networks like Content Distribution Networks (CDNs), peer-to-peer (P2P) networks, internet backbone networks or simply a generic wired or a wireless network. The work done in [3] classifies the delays observed in internet backbone networks and goes on to talk about the effect delay has on different applications running over an internet backbone network with respect to the end user. Using our technique in conjunction with studies like [3] could enable one to infer more information about the nature of delays in backbone networks. With little modification, our technique can also be adapted to diagnosis with respect to other parameters like throughput.

The rest of the paper is divided into the following sections. Section II gives an idea about related work in the direction of network diagnosis done in recent years in the domain of both

This research is supported by NSF grants (NSF 1066391, NSF 1125165, NSF 1442630) to Sensorweb Research Laboratory at Georgia State University.

978-1-4799-8461-9/15/$31.00 ©2015 IEEE

wired and wireless networks. Section III explains in detail the way we formulate our problem into the $A\chi = B$ system of equations. Section IV describes the techniques we use to solve the problem formulated in the previous section along with the algorithm. Section V deals with the results we obtained by implementing our approach in CORE. In this section we also show the simulation results of applying our algorithm on the backbone network of a major service provider of North America. Finally, in Section VI we summarize our work and briefly outline the direction of future work in network diagnosis.

## II. Related Work

The concept of network tomography was coined in the work [4], which proposed a technique of evaluating network parameters like average delay based on Hidden Markov method when random routing was being followed. The work by *Castro et al* [5] discusses about two types of scenarios pertaining to network diagnosis: one where fixed routing is being followed, and one where dynamic routing is being followed. In case of dynamic routing being followed, the work describes techniques which use a tree based topology for probing using the *sandwich probing* technique [6]. It proposes the use of either the Maximum Likelihood Estimate or the Markov Chain Monte Carlo method to get a best estimate of the actual network topology and then perform network diagnosis using the estimated topology as reference.

There have been attempts at network diagnosis in wired networks using link diagnosis. One of the ways of performing this is the Optimal Sequential Testing [2]. This approach basically presents a heuristic technique which will test each link for failure thereby eliminating many combinations and possibilities. The algebraic method illustrated in [7] provides a method to describe the faults in the entire network on the basis of linearly independent sequences of nodes. The LEND (Least end-to-end Network Diagnosis) [1] approach consists of identifying a MILS (Minimum Identifiable Link Sequence) which will narrow down the region of failure giving a more efficient approach towards network diagnosis. The approach uses the reliability as a log likelihood function.

The proposed network tomography algorithm presented in this paper is a novel way of estimating the network parameters like node delay, based only on end-to-end measurements using *Algebraic Reconstruction Techniques* [8]. Our proposed algorithm gradually converges to an acceptable solution starting from an initial random guess. The strength of the proposed algorithm is that it does not require any knowledge of the routing paths. Our approach also incurs very little overhead in comparison with previously mentioned techniques as only $O(n)$ end-to-end measurements is required, where $n$ is the number of nodes in the network.

## III. Problem Formulation

In order to model the problem of network diagnosis, let us consider a network represented by graph G(V,E) consisting of $n$ nodes. The nodes in the network are represented by set V which is the set of all vertices in G where $|V| = n$ The links between nodes is represented by set E which is the set of all edges in graph G. Let us further consider the set $\Omega$ representing the delays on each node, where the following holds $\omega_k \in \Omega$ $\forall v_k \in V$

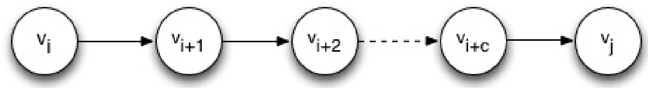



Fig. 1. Path from source to destination showing hops in between

Let us consider Fig. 1 showing a path from vertex $v_i$ to $v_j$, consisting of $c$ nodes in between, with the source being at $v_i$ and destination being $v_j$. Let the set $S_{ij}$ consist of all the nodes encountered while traveling from node $v_i$ to node $v_j$. As the packet traverses from source to destination, it encounters delays at each hop. These delays may be caused at Link Layer or Network Layer mainly due to congestion. Since delays are incurred on a node only while sending a packet, we have the following equation.

$$\sum_{k=i}^{i+c} \omega_k = b_{ij} \tag{1}$$

where, $\omega_k \in \Omega$ and $v_k \in S_{ij}$. Equation (1) represents the method for calculating the end-to-end delay, denoted by $b_{ij}$, by adding up the delays encountered along each node while the packet moves from source $v_i$ to destination $v_j$. In the above network represented by G, the path vector $P_{sd}$ for the path from any $v_s$ to any $v_d$ can be represented as follows.

$$P_{sd} \equiv \{p_1, p_2, \ldots, p_n\} \tag{2}$$

$$|P_{sd}| = n \tag{3}$$

where,

$$p_k = \begin{cases} 1, & \text{if } v_k \in S_{sd} \\ 0, & \text{otherwise} \end{cases}$$

Now let us consider a scenario where we have $m$ such simultaneous source-destination pairs in G. Let $\sigma$ be a set consisting of $m$ such source destination pairs with $m \in O(n)$.

$$\sigma = \{\{s_1, d_1\}, \{s_2, d_2\}, \ldots, \{s_m, d_m\}\} \tag{4}$$

where, $v_{s_i}, v_{d_i} \in V$ , with $1 \leq i \leq n$ and $v_{s_i} \neq v_{d_i}$
We now define matrix $A(m{\times}n)$ in the following way.

$$A = \left[P_{s_1 d_1}, P_{s_2 d_2}, \ldots P_{s_m d_m}\right]^T \tag{5}$$

Let $\chi$ be a $n{\times}1$ vector such that,

$$\chi = \left[x_1, x_2 \ldots x_n\right]^T \tag{6}$$

where,

$$x_k = \omega_k, \forall \omega_k \in \Omega, 1 \leq k \leq n \tag{7}$$

Let $B$ be a $m{\times}1$ vector such that,

$$B = \left[b_{s_1 d_1}, b_{s_2 d_2} \ldots b_{s_m d_m}\right]^T \tag{8}$$

Now, our problem of network diagnosis can be formulated in the form of the following equation.

$$A\chi = B \tag{9}$$

The aim is to now solve for $\chi$. As easy as it might sound, solving for $\chi$ is not really straight forward. In our case only *B* can be gathered from the network using end-to-end measurement. Since we do not have any knowledge about what path a packet takes we do not have any knowledge about *A*. Next section explains the various concepts relating to solving this problem and provides an algorithm for network tomography.

## IV. Network Tomography

In this section we discuss how tomography principles have been used to diagnose the network. We further develop the idea of Network Tomography along the lines of the approach presented in the previous section and introduce the notion of *regularization* to solve *ill-posed* [9] problems such as ours.

### A. Perturbation Model

Equation (9) in Section III represents a system of equations which needs to be solved simultaneously to obtain an indication of delays at each node. Without the knowledge of *A*, it is not possible to solve this equation. In this section we propose a technique to estimate $\chi$ which in turn can lead to an estimate of *A*. Our approach iteratively improves on the estimate of $\chi$ which finally converges to a solution within acceptable tolerance.

A common technique in geophysics is that of ray tracing [10], [11] which involves plotting the trajectory of a seismic ray from the epicenter to the surface of the earth. In a similar way, a path from a source node to a destination node can be thought of as a ray. The time taken for the *ray's* traversal from the epicenter to the surface can be translated to the time taken by a packet to traverse the path.

Our algorithm builds on two aspects of the problem viz. *forward problem* and *inverse problem*. Intuitively, it can be visualized as follows. The *forward problem* consists of estimating the *A* matrix using the best known estimate of $\chi$. Once an estimate of *A* is known, the *inverse problem* then deals with calculating a new estimate of $\chi$ by using an Algebraic Reconstruction Technique (ART). The new estimate of $\chi$ then forms the basis for calculating a new estimate of *A* by the *forward problem* and the process continues until the solution falls below a required threshold. Fig. 2 provides a brief overview of the forward and inverse problem mentioned before along with the sequence in which they are executed.

An approach presented in [11] known as the *perturbation model* can be used in conjunction with the *forward problem* and *inverse problem* to solve Equation (9). Let $\chi^k$ be the $k^{th}$ estimate of $\chi$ and $A^k$ be the $k^{th}$ estimate of the A matrix, one can then find $T^k$ such that,

$$A^k\chi^k = T^k \tag{10}$$

Further,

$$\delta t = B - T^k \tag{11}$$

where $\delta t$ is the residual error. The next estimate $\chi^{k+1}$ is found by solving the equation.

$$\delta x = ART(A^k, \delta t) \tag{12}$$

where ART is any Algebraic Reconstruction Technique [12] and $\delta x$ is the perturbation.

$$\chi^{k+1} = \chi^k + \delta x \tag{13}$$

Equation (13) calculates the next estimate of $\chi$ by considering the difference from observed value. Equation (10), Equation (11), Equation (12) and Equation (13) illustrate the concept of perturbation model.

Let the algorithm which the routing protocol uses to calculate routes from source to destination in the network represented by G be referred to as $\alpha$. Let $\beta$ be another algorithm which can be used to find a path from any two points in G. Both $\alpha$ and $\beta$ try to find a minimum cost path from a source $s$ to a destination $d$.

To solve the forward problem let us consider an estimate $\chi^k$. An estimate $A^k$, is then generated using algorithm $\beta$ for each pair in $\sigma$ by taking $x_i^k \in \chi^k$ as the cost of traveling through node $v_i$. In other words, the minimum end-to-end delay incurring path for each pair in $\sigma$ is calculated to form the estimate of $A^k$. The application of the perturbation model as illustrated in Equations 10 to 13 to compute the new estimate $\chi^{k+1}$ forms the *inverse problem*. Even though our technique is centered around the delay experienced at each node, the same technique will also hold for any other metric like throughput, provided both $\alpha$ and $\beta$ use the same metric to calculate the path.

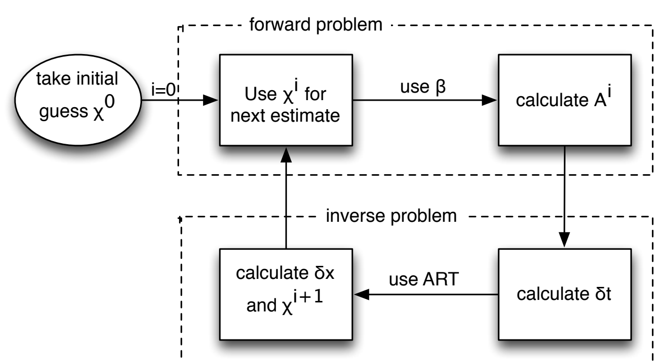



Fig. 2. Overview of the forward and inverse problem

### B. Regularization

In cases where the problem is *ill-posed* i.e. highly sensitive to changes in the *A* matrix, additional constraints need to be added to the system to ensure that a solution nearer to the actual solution is found. One such method of regularization is the Tikhonov regularization [9]. Tikhonov regularization basically involves adding additional rows to $A$ and $B$ such that a new system with an improved set of constraints is formed. To illustrate this let us consider, the system of equations represented by Equation (9). We define a diagonal constraint matrix $\Lambda$ of size $n \times n$, which gives rise to a constraint relation as follows.

$$\Lambda\chi = 0_{n1} \tag{14}$$

where $n$ is the number of nodes in G, $0_{n1}$ is a zero vector of size $n \times 1$ and $\lambda_{jj} > 0, \forall \lambda_{jj} \in \Lambda$ We append Equation (9) with Equation (14) to obtain the following.

$$\begin{bmatrix} A \\ \Lambda \end{bmatrix} \chi = \begin{bmatrix} B \\ 0_{n1} \end{bmatrix} \tag{15}$$

Each diagonal value $\lambda_{jj}$ can be thought of as a constraint parameter on the $j^{th}$ node and setting $\lambda_{jj} = 0, \forall \lambda_{jj} \in \Lambda$ is equivalent to not having any regularization at all. The values of $\Lambda$ is specific to the problem and those yielding the best results is generally chosen.

### C. Data Collection

In this paper we aim to illustrate the idea behind successfully using the Algebraic Reconstruction Technique in mesh networks. We adopt a relatively simple centralized approach with the assumption that there is a central node in the network, which is aware of the entire topology of the network. Fig 3 shows the method illustrated below. The information is gathered in the following way:

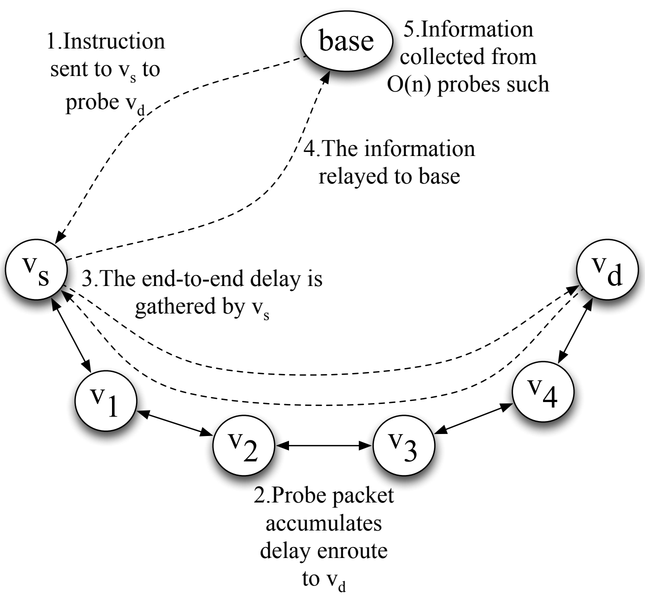



Fig. 3. Pictorial representation of data collection method

1. The central node instructs each node in the network, to send probe packets to specific other nodes in the network.
2. The node then probes the specified destination. The probe packet keeps accumulating delays as it travels to the destination.
3. The end-to-end delay in terms of round trip time is gathered by it.
4. The node then forwards this information to the central node. The information consists of the source node ID, the destination node ID and the end-to-end delay encountered.
5. The central node receives this information from all such nodes and formulates it as an $A\chi = B$ system of equation.

The ART method is then applied on this system of equations to get the delays. In all there are O(n) probes sent out in the network by the nodes. The source and destination pairs can be decided on a number of factors as deemed fit for the application. A common and simple one that can be used is geographical distance which is also what we used in our evaluations.

### D. Network Tomography Algorithm

Based on Section IV-A and Section IV-B we can now define an algorithm which starts out with an initial random guess and converges to the actual value as indicated in the end-to-end measurements. Algorithm 1 describes the Network Tomography algorithm, which we use to diagnose the network. It starts with initialization of $B$ with the measured end-to-end values based on the source destination pairs mentioned in $\sigma$. It uses $\beta$ to populate $A^i$ based on $\chi^i$ as the weight matrix where $\beta$ is an algorithm which most closely resembles $\alpha$, the actual routing protocol being used in the network. After finding the perturbation $\delta t$, it proceeds to calculate $\delta x$ with the help of an ART technique, which is then used to obtain $\chi^{i+1}$. We introduce an adaptive regularization scheme, in which

**Algorithm 1** Network Tomography Algorithm

$\chi^0_{n\times 1} \leftarrow$initial guess
$B_{|\sigma|\times 1} \leftarrow$ end-to-end measurements based on $\sigma$
$\Psi_{n\times n} \leftarrow$ adjacency matrix of G
$\sigma \leftarrow$ initialize to source destination pairs
$B_{(|\sigma|+n)\times 1} \leftarrow [B; 0_{n\times 1}]$ //row concatenation
$\Lambda_{n\times n} \leftarrow I_n$ initialize to identity matrix
$i \leftarrow 1$
**while** $i < MAXITR$ **do**
  $A^i = \beta(\Psi, \sigma, \chi^i)$
  $A^i \leftarrow [A^i; \Lambda^i I_n]$ //row concatenation
  $\delta t = A^i \chi^i - B$
  $\delta x = ART(A^i, \delta t)$
  $\chi^{i+1} = \chi^i + \delta x$
  **for** $j = 1 \ldots n$ **do**
    **if** $(x^{i+1}_j < 0)$ **then**
      $x^{i+1}_j = \theta, 0 \leq \theta \leq 1$
    **end if**
    $\lambda^{i+1}_{jj} = \frac{1}{x^{i+1}_j}$
  **end for**
  $i \leftarrow i + 1$
**end while**

$\Lambda^{i+1} = diag(\frac{1}{\chi^{i+1}})$ indicating that the diagonal elements of $\Lambda^{i+1}$ are inversely proportional to the respective values in $\chi^{i+1}$. This technique of regularization is aimed at imposing a greater penalty on the low delay nodes and lower penalty on the high delay nodes and thereby obtaining a more accurate description of the delays at each individual nodes.

### *E. Scope of the algorithm*

The technique mentioned in this paper is aimed at diagnosis of mesh networks in general. Due to its ubiquitous nature, it could be easily adapted to different networks like content distribution networks, peer-to-peer networks and large scale wireless networks to name a few. This method could also have commercial importance as it is a deterministic approach towards network diagnosis which could help in network maintenance of organizations running large networks. The ART technique being used has the capacity to deal with a large number of probes and will prove vital in keeping the computation cost of this technique low. This in turn could prove beneficial in scenarios where real time network diagnosis is required. The ART technique also has the distinctive capability of handling underdetermined systems too. An underdetermined system is an equation in which the number of rows is less than the number of columns. An underdetermined system is actually representative of less data or absence of reliable data. The ART algorithm however delivers good performance in the case of sufficiently underdetermined systems too [12].

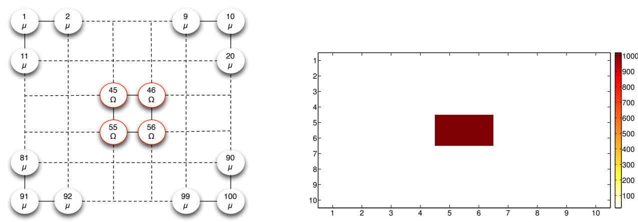
The frequency of executing this algorithm for diagnosis depends mostly on the application and the size of the network. While it might be more useful to run this algorithm periodically in case of traffic intensive large mesh networks, it might suffice to execute it relatively lesser number of times in case of smaller wireless networks. Further since the evolution of delays in the network mostly behaves in a stochastic manner, the frequency of application of the algorithm has a direct bearing on the $freshness$ of the information obtained during probing. Our algorithm is therefore more concerned with how to diagnose a network given some data about it, rather than how to obtain the given data.

## V. Evaluation and Results

To evaluate the network tomography technique, we used two approaches, emulation using synthetic data on a Wireless Mesh Network and one using real data pertaining to the backbone network of a major US provider. It was our intention to evaluate and observe the performance of the network tomography algorithm in diverse settings. Emulations with synthetic data can be helpful in observing any anomalies in the result obtained. It helps us determine whether the algorithm can deliver a robust performance in a condition as set up by us before it can be applied elsewhere. Consequently, we apply our algorithm on real data to further validate our algorithm and observe its performance.

### *A. Synthetic data*

For evaluating our algorithm we emulated a wireless mesh network in CORE [13]. We introduced a high delay in the Data Link layer of some of the nodes in our set up which we refer to as the high delay nodes and the rest as low delay nodes. CORE is a network emulator which creates virtual Network Interface Cards (NICs) for each node in G on a single host machine allowing emulation of actual network settings. The advantage of CORE is that traditional Unix like environment can be obtained on each of the nodes in the network which makes porting code to actual physical devices from the virtual nodes straightforward. To illustrate our way of representing

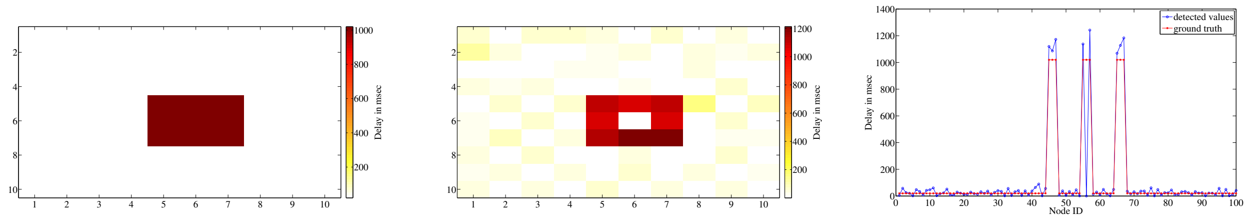



(a) (b)

Fig. 4. Mapping of nodes with their delays

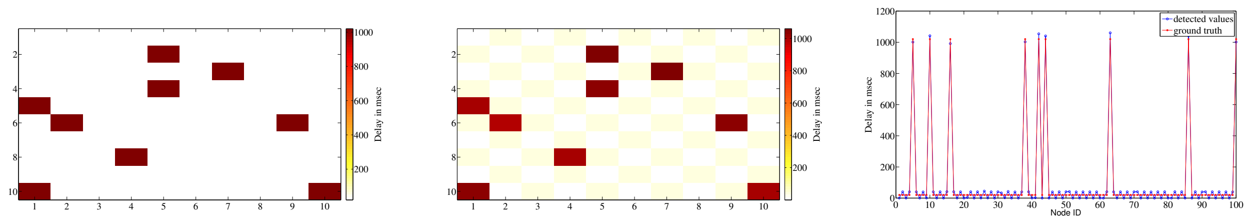
the node to delay mapping consider Fig. 4. It can be observed that a cluster of nodes in Fig. 4(a) i.e. 45, 46, 55 and 56 which correspond to the *red* nodes in Fig. 4(b) are having a high delay. The other nodes in Fig. 4(a) correspond to the rest of the nodes in Fig. 4(b) having a low delay.

To evaluate our technique we used the BATMAN [12] routing protocol ($\alpha$) in the CORE emulation to gather the end-to-end measurements. A shortest path finding algorithm like the Floyd Warshall Algorithm ($\beta$) was used in each successive iteration for path reconstruction. Our CORE setup consisted of 100(10×10) nodes in a mesh topology. We introduce delays in the MAC layer of the virtual nodes in CORE and use an active probing method to obtain the end-to-end measurement of delays using ICMP (Internet Control Message Protocol) packets to obtain the relevant information. We try to maximize the number of hops between $v_{s_i}$ and $v_{d_i}$ for each pair $v_{s_i}, v_{d_i} \in \sigma$. This can be easily accomplished by choosing the source-destination pairs based on a simple parameter like the Euclidean distance between the geographic locations or node IDs. Fig. 5 depicts a case in which a pocket of nodes are having a high delay. Fig. 5(a) shows the ground truth and Fig. 5(b) shows the result of Algorithm 1. It can be observed from Fig. 5(a) and Fig. 5(b) that one node in the centre of the pocket has not been detected as a high delay node. This is more of a limitation with the functioning of the routing protocol as no packets sent by that particular node are able to get through to other nodes. Therefore in case of pockets, our technique is able to detect an outline of a high delay area, and a similar probing technique could be used within the high delay pocket to investigate further. On the other hand Fig. 6 relates to the case wherein we have the high delay nodes scattered randomly across the network. Fig. 6(a) shows the ground truth whereas, Fig. 6(b) shows the values detected by Algorithm 1. Fig. 5(c) and 6(c) plot the value of the ground truth and the detected values for each node in the network in case of pocket




(a) Ground truth delay map (b) Detected delay map (c) Comparison of ground truth with detected values per node

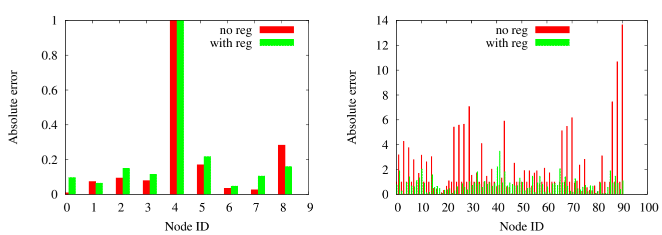
Fig. 5. Scenario where high delay nodes form a pocket




(a) Ground truth delay map (b) Detected delay map (c) Comparison of ground truth with detected values per node

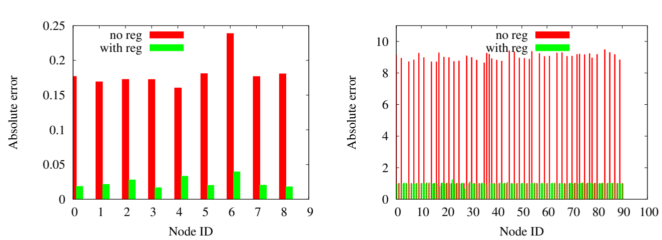
Fig. 6. Scenario where high delay nodes are randomly scattered across the network

delay and randomly incurred delay respectively. It becomes clear from Fig. 5 and 6 that our algorithm is able to segregate the high delay incurring nodes from the low delay incurring ones with a high level of precision using only $O(n)$ end-to-end measurements. Fig. 7 and Fig. 8 depict the error values




(a) High delay nodes (b) Low delay nodes

Fig. 7. Delay detection error comparison in terms of regularization with reference to Fig. 5




(a) High delay nodes (b) Low delay nodes

Fig. 8. Delay detection error comparison in terms of regularization with reference to Fig. 6

when regularization is used and when there is no regularization being used. The error value at the $l^{th}$ node is calculated with the help of the following equation.

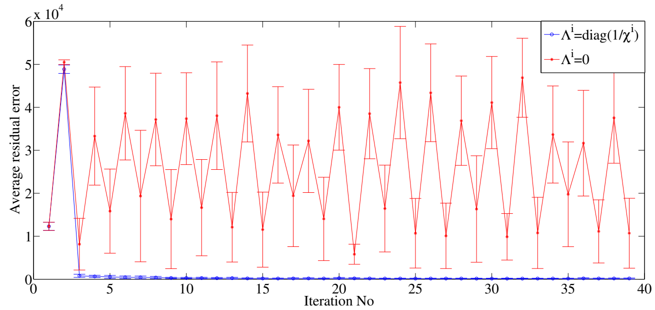
$$error_l = \frac{|x_l^* - x_{gt_l}|}{x_{gt_l}}, \forall 1 \leq l \leq n \tag{16}$$

In Equation (16), $x_l^*$ and $x_{gt_l}$ are the detected delay and ground truth values at the $l^{th}$ node respectively. Fig. 7 and Fig. 8 highlight the fact that regularization is essential for achieving greater accuracy while detecting delays. Fig. 9(a) shows the average residual error $\rho$ and Fig. 9(b) the average absolute error $\eta$ calculated using the following equations.

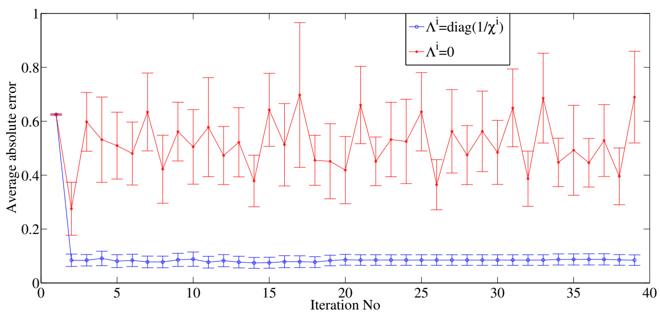
$$\rho_i = \frac{\sum_{j=1}^{p} ||A_j^i \chi_j^i - B_j||_2^2}{p} \tag{17}$$

$$\eta_i = \frac{\sum_{j=1}^{p} \frac{||\chi_j^i - \chi_j^{gt}||}{||\chi_j^{gt}||_2^2}}{p} \tag{18}$$

where $p$ is the number of different cases emulated, $\chi_j^{gt}$ represents the ground truth delays in the $j^{th}$ case, $\chi_j^i$ is the value determined in the $i^{th}$ iteration of the $j^{th}$ case. Similarly, $A_j^i$ represents an estimate of the A matrix in the $i^{th}$ iteration of the $j^{th}$ case. Therefore, $\rho_i$ given by Equation (17) and $\eta_i$ given by Equation (18) represents the average residual error and average absolute error in the $i^{th}$ iteration across all the $p$ cases respectively. From Fig. 9(a) and Fig. 9(b) it can be observed that on an average our method converges to an acceptable error margin within a few iterations. The zig-zag nature of the curve corresponding to $\Lambda^i = 0$ in both is the result of the non-linearity in the forward and inverse problem. However, with adaptive regularization of the form $\Lambda^i = diag(\frac{1}{\chi^i})$, the




(a) Average residual error $\rho$




(b) Average absolute error $\eta$

Fig. 9. Error bar plots showing average residual error and average absolute error

constraint on $\chi$ regularizes the problem to converge to actual linear system represented by Equation (9). Thus the adaptive regularization technique explained in Section IV-B and Section IV-D leads to a very accurate prediction of the delay in nodes.

### *B. Real Data*

To evaluate our algorithm in a real world scenario, we performed simulations with the help of real data obtained from The Internet Topology Zoo project [14]. The project aims at documenting and archiving the backbone topology of major internet providers of the world. We chose ATT North America's backbone network to simulate our algorithm. This simulation is intended to demonstrate how our approach can be used in a real world scenario to provide a delay based map of an entire mesh network. To simulate the delays we extrapolated the data from the work done in [3] which provides useful insights into the delay characteristics of backbone networks of major service providers and states the effect the delay has on certain applications running on the network. To simulate errors during data collection, our simulation incorporated a noise to the measurements giving rise to an equation as follows

$$A\chi = B + \varepsilon \tag{19}$$

where $\varepsilon$ is a 10% Gaussian noise vector.

The Fig. 10 shows the simulation results of our algorithm on ATT's backbone network with the conditions as specified by Equation (19). As is evident from Fig. 10, our algorithm provides a fairly accurate delay map of the entire network. Fig. 11 presents a histogram which provides the per node

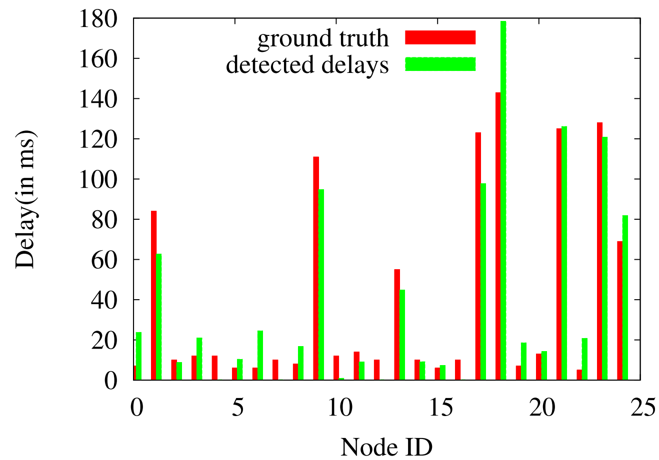



Fig. 11. Per node delay for simulations represented by Fig 10

delay of all the nodes in the network. To determine the source-destination pairs for $\sigma$, we chose pairs such that their geographical distance was maximized while using the same technique for data collection as illustrated in the previous sections. The delays were put into a low and a high delay category. The low delay category had nodes whose delays ranged from $5ms$ to $30ms$ and the high delay nodes had delays which ranged from $50ms$ to $150ms$. These delays were chosen in accordance with the internet backbone delay patterns which have been explained in detail in [3].

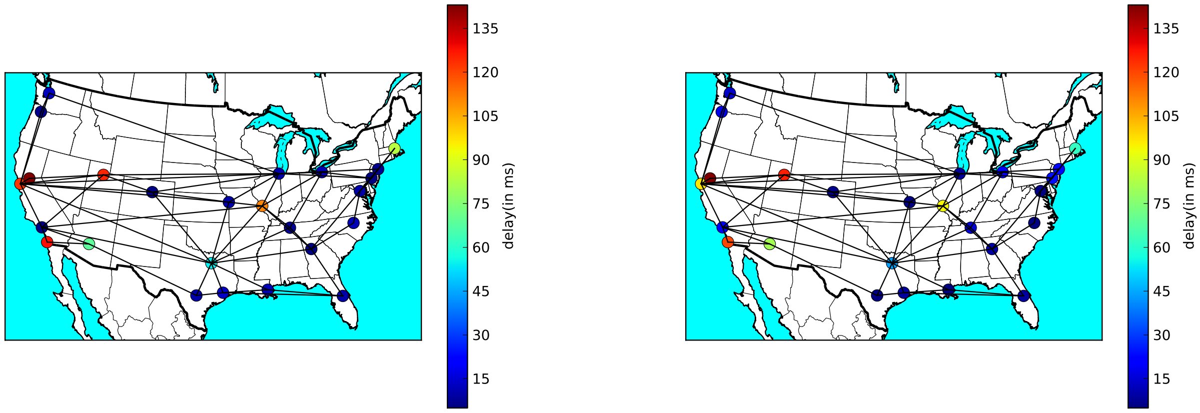
This simulation therefore proves that the algorithm is capable of handling random network topologies and heterogeneous delays which makes it possible to apply it to diverse scenarios such as this one. It can be seen from Fig. 10 and Fig. 11 that our algorithm has the capability to deliver a robust performance in real world scenarios also.

### *C. Limitations*

Although our algorithm gives good results in different scenarios as illustrated above, it has a few limitations. Firstly, the algorithm is not capable of handling blackouts. Blackouts occur when a node powers down and as a result drops all packets. In such a case if the network graph becomes disconnected, our algorithm cannot diagnose the network in its entirety. Secondly, gathering information and ray tracing can be very slow when the number of nodes in the network have been scaled up to a high number. In such a case a distributed version of this algorithm might prove highly beneficial and we intend to pursue this direction in the future as an extension to this current work.

## VI. Conclusion and Future Work

In this paper, we develop a technique which uses the principle of network tomography to perform diagnosis in terms of the delay incurred at each node. This paper presents an approach which tries to identify nodes having a higher than usual delay in the network. The technique attempts to get a closer estimate of the actual ground truth in each




(a) Ground Truth

(b) Detected delays by our algorithm

Fig. 10. Simulation of delays for the ATT North America backbone network

successive iteration by trying to minimize the perturbation between the actual and detected values. Inspired from the concept of seismic tomography we try to get an estimate of the path between a particular source and destination and iteratively keep improving each successive guess. We show with the help of emulation results that, this method can be successfully applied to diagnose a network. By using an active probing technique our approach successfully manages to get an accurate view of the delays in the network using $O(n)$ number of probes and within a reasonable number of iterations.

We also demonstrate a practical application of our algorithm to diagnose the backbone network of a major internet service provider using simulations. These simulations also show that our algorithm can handle different topologies and can handle heterogeneous delays as well. Our algorithm can similarly be adapted to other mesh network domains like peer-to-peer and content distribution networks (CDNs) for diagnosis.

Future work in this direction includes an extension of the aforementioned concept into the domain of other network parameters like throughput. We also aim to study the behavior with link state routing protocols by using an appropriate regularization technique and then performing network diagnosis. Another important direction that we intend to explore is distributed diagnosis, wherein the ART can be run on different nodes in the network. One main advantage of this approach would be scalability when the number of nodes in the network is significantly increased and also faster diagnosis.

## References


[1] Y. Zhao, Y. Chen, and D. Bindel, "Towards Unbiased End-to-End Network Diagnosis," *Networking, IEEE/ACM Transactions on*, vol. 17, pp. 1724–1737, Dec. 2009.

[2] B. Wang, W. Wei, W. Zeng, and K. R. Pattipati, "Fault Localization Using Passive End-to-End Measurement and Sequential Testing for Wireless Sensor Networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON '09. 6th Annual IEEE Communications Society Conference on*, pp. 1–10, June 2009.

[3] A. Markopoulou, F. Tobagi, and M. Karam, "Loss and Delay Measurements of Internet Backbones," *Comput. Commun.*, vol. 29, pp. 1590–1604, June 2006.

[4] Y. Vardi, "Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data," *Journal of the American Statistical Association*, vol. 91, no. 433, 1996.

[5] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: recent developments," *Statistical Science*, vol. 19, pp. 499–517, 2004.

[6] M. Coates, R. Castro, R. Nowak, M. Gadhiok, R. King, and Y. Tsang, "Maximum Likelihood Network Topology Identification from Edge-based Unicast Measurements," in *Proceedings of the 2002 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '02, (New York, NY, USA), pp. 11–20, ACM, 2002.

[7] Y. Chen, D. Bindel, H. H. Song, and R. H. Katz, "Algebra-Based Scalable Overlay Network Monitoring: Algorithms, Evaluation, and Applications," *Networking, IEEE/ACM Transactions on*, vol. 15, pp. 1084–1097, Oct. 2007.

[8] G. T. Herman, *Fundamentals of Computerized Tomography: Image Reconstruction from Projections*. Springer Publishing Company, Incorporated, 2nd ed., 2009.

[9] P. C. Hansen, *Rank-deficient and discrete ill-posed problems*. SIAM Monographs on Mathematical Modeling and Computation, Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 1998. Numerical aspects of linear inversion.

[10] G. Kamath, L. Shi, and W.-Z. Song, "Component-Average Based Distributed Seismic Tomography in Sensor Networks," in *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on*, pp. 88–95, May 2013.

[11] L. Shi, W.-Z. Song, M. Xu, Q. Xiao, J. M. Lees, and G. Xing, "Imaging seismic tomography in sensor network," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on*, pp. 327–335, June 2013.

[12] D. Seither, A. Konig, and M. Hollick, "Routing performance of Wireless Mesh Networks: A practical evaluation of BATMAN advanced," in *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, pp. 897–904, Oct. 2011.

[13] J. Ahrenholz, "Comparison of CORE network emulation platforms," in *MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010*, pp. 166–171, Oct. 2010.

[14] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," *Selected Areas in Communications, IEEE Journal on*, vol. 29, pp. 1765–1775, Oct. 2011.