Network Tomography: A Novel Algorithm for Probing Path Selection

Teresa Pepe, Marzio Puleri
Ericsson Research,
Pisa, ITALY
E-mail: {Teresa.Pepe, Marzio.Puleri}@ericsson.com

***Abstract*—The impressive growth experienced by the Internet in the last years, together with the diversity in network technologies are posing more and more challenges in the field of network monitoring, where novel approaches, able to obtain an integrated view among multi-technological and multi-vendor domains, are required. In such a context, Network Tomography (NT) has emerged as a very promising monitoring technique, being able to infer unobservable network performance just relying on end-to-end (E2E) measurements.**

This work proposes a novel algorithm that permits to overcome one of the main limitations of NT, that is the proper choice of a minimal set of E2E probing connections, which provides a full monitoring coverage of the network. The proposed approach allows to find automatically the minimal set of probing paths thanks to a proper combined use of the Yen algorithm and the Gauss reduction method.

We demonstrate the effectiveness of our algorithm via simulation analysis.

***Index Terms*—Network Tomography, probing paths selection, identifiability**

I. Introduction

Nowadays, with the evolution of the networks for supporting the new emerging services, operators are facing the problem of managing multi-technological heterogeneous networks in an effective and centralized way. In such a context, an accurate and timely knowledge of the internal status of a network (e.g., delays on individual links, congestion level) is essential for various network operations such as route selection, resource allocation, and fault diagnosis. However, the heterogeneous and unregulated structure of the Internet makes it hard to infer this information. Moreover, the lack of cooperation from the internal nodes in collecting the required network measurements makes this task even harder.

In such a scenario, Network Tomography (NT) has emerged as a very promising technique for implementing centralized remote OAM (Operation Administration and Maintenance) functionalities. One of the most interesting characteristics of such an approach is represented by its transparency to the network. Indeed, NT adopts the same concepts applied in medical tomography (from which the name is derived) to obtain an image of the internal behaviour of the network, by scanning it from the outside, as if it were a human body. Moreover, being technology agnostic it can be used to monitor multi-technological and multi-domain networks.

NT is based on the collection of E2E measurements, obtained by establishing several probing connections (i.e., traffic flows exchanged between monitoring points). The availability of such monitoring connections limits the applicability of such a technique. For this reason, the identification of a proper set of E2E paths able to provide a full coverage of the network (i.e., the solution of the so-called "identifiability problem") is crucial to the successful application of NT.

Although in literature there are significant efforts in analyzing the topological conditions to ensure identifiability ([1][2][3]), to the best of our knowledge a method for the selection of the needed set of paths to uniquely identify all the link metrics from E2E measurements in a general scenario is still missing, making the application of NT in field substantially difficult.

The aim of this work is to provide a novel general algorithm for the selection of the minimum set of probing paths to achieve the identifiability of any kind of network, making feasible the application of NT in field. We developed an automatic method to determine a full monitoring coverage of an arbitrary network, while simultaneously minimizing the overhead due to the transmission of probing packets.

The remainder of this paper is organized as follows: Section II presents a formal description of the problem. Section III details the proposed algorithm, also discussing the computational complexity. Section IV reports the results of the tests carried out to evaluate the effectiveness of the proposed method. Finally, Section V concludes the paper with some final remarks.

II. Problem Statement

In this section, we provide a detailed description of the faced problem, describing the network model and formally discussing the identifiability problem. To make the reading of the following easier, we refer the reader to table I, where all used notations are reported.

A. Network Model

First of all, let us suppose that the network topology is well known, or has been reconstructed by using a topology discovery procedure, and can be hence modelled as a directed graph (or digraph).

Formally, a digraph G is represented as $G = (V, E)$ where V is the set of vertices or nodes (i.e., network nodes) and E is a set of directed edges (i.e., communication links between nodes). Moreover, let us suppose that several end-systems are

978-1-4673-6432-4/15/$31.00 ©2015 IEEE

TABLE I
NOTATION

Symbols	Meaning
G	Digraph associated to the network topology
V	Vertices or nodes
E	Edges or links
n_E	edges number
n_V	vertices number
P	Set of probing paths
p_i	Element of the set P
d_i	destination node for path p_i
s_i	source node for path p_i
A	Dependency matrix
B	set of monitoring nodes
M	Number of probing connections
m	Monitoring points
$nPath$	Number of independent path

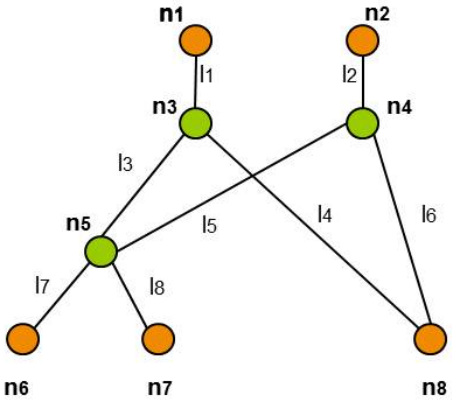

Fig. 1. Example of Network Topology

connected to the network and that they can send and receive probing packets (note that this assumption is reasonable, given that it simply means that the operator is able to connect some monitoring nodes to its own network). Formally, the subset of vertices used for injecting and extracting probing packets is defined as the set of border nodes $B = \{m_i\}$.

We define a path p_i as a sequence of links that starts from a source host s_i and ends at a destination host d_i. A path p_i is a probing path if the nodes s_i and d_i belong to B. All the probing paths in the network form the probing paths set P.

We assume that the operator can freely choose the probing paths in the network. This can be achieved in case of networks supporting traffic engineering, like IP/MPLS or IP networks supporting segment routing (SR).

Hence, given a network $G = (V, E)$ and a probing path set P, we can define a dependency matrix $A_{M\times N}$:

$$A_{i,j} = \begin{cases} 1 & \text{if path } p_i \text{ traverses link } l_j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where, the i^{th} row $A_{i,\cdot}$ of A represents the probing path p_i and the j^{th} column $A_{\cdot,j}$ of A represents the link j of the network. M is the number of distinct probing paths p_i and N is the number of the links in the network.

TABLE II
MONITORING PATH

Monitoring Paths	
p_1	l_1,l_3, l_7
p_2	l_1,l_3, l_8
p_3	l_1,l_4
p_4	l_2,l_5, l_7
p_5	l_2,l_5, l_8
p_6	l_2,l_6

As an example, let us consider the network represented in figure 1 with eight links $\{l_1, l_2, \ldots, l_8\}$ and eight nodes $\{n_1, n_2, \ldots, n_8\}$. Assuming to have five monitoring nodes $\{n_1, n_2, n_6, n_7, n_8\}$ (i.e., nodes among which we can establish probing paths) and six distinct probing paths $\{p_1, p_2, \ldots, p_6\}$ (reported in table II), the dependency matrix is:

$$A = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

B. The Identifiability Problem

The identifiability problem aims at finding the set of probing connections able to provide the coverage required for a full monitoring of the network.

The identifiability of a network depends upon the topology itself, that is, given a network topology, it is not always possible to define a complete set of probing paths so as to be able to provide a full monitoring coverage.

If the network is identifiable, the problem will be that of determining the minimum set of probing paths that guarantees a full coverage of the network.

If we take into consideration additive metrics (e.g., link delay), the relation between probing paths and links can be represented by a system of linear equations, where the unknown variables x are the link metrics, and the known constants y are the end-to-end path measurements, each equal to the sum of the corresponding link metrics. In case the use of a multiplicative metric is required, it can be reduced to an additive one by using the log( ) function of the metric [4].

Given the dependency matrix A, the array X whose element x_j is the unknown value related to the link l_j, and the array Y with element y_i measurement on probing path p_i, we can write:

$$Y = A \cdot X \quad (2)$$

where the i^{th} equation is:

$$y_i = \sum_{j=1}^{M} A_{i,j} \cdot x_j \quad (3)$$

Hence, from a mathematical point of view, the solution of the identifiability problem consists in finding a number of linearly independent equations equal to the number of the parameters to be detected so as to uniquely determine X. Formally, given the linear system in (2), the network G is identifiable iff A has full column rank, i.e., $rank(A) = N$.

It is important to highlight that, as already stated, due to structural limitations in network graph, it is not always possible to define the set of equations that solve the identifiability problem. In more detail, as shown in [5] and [6], the maximum number of independent paths $nPath$ in a network is

$$nPath = n_E - n_V + m + 1 \quad (4)$$

where n_E and n_V are the number of edges and nodes, respectively, and m is the number of monitoring points. Thus, the number of distinct probing paths, corresponding to linear independent equations, that is possible to select in a network is equal to $nPath$.

As a consenquence, we can conclude that a network is identifiable iff $nPath$ is at least equal to the number of network links n_E. Only in this case, it is possible to find a set of probing paths that provide the full monitoring coverage of the network; that is the set of equation which the system (2) is uniquely determined for and solves the variable X by determining the performance of each link.

III. Proposed Algorithm

Our proposal aims at providing a method to determine a proper set P of probing paths for the full monitoring coverage of a generic identifiable network, while simultaneously minimizing the overhead traffic due to the transmission of probing packets finding the minimum set P.

As already stated in the previous sections, we have assumed to use active probing (i.e., by establishing probing connection, in contrast with passive probing that only use already available connections), since the coverage provided by connections already present in the network could not span all the paths of interest.

The basic idea behind our proposal is to determine the distinct probing paths, strictly necessary to uniquely determine the internal status of the network. As said in the previous section, the maximum number of independent paths that we can find is equal to $nPath$ and a network is identifiable iff nPath=n_E.

The proposed algorithm works as follows: given the set of monitoring nodes the algorithm selects a first couple (m_1, m_2) of monitoring points and begins by computing a first probing path. Since there isn't any constraint for the selection of the first path, in the name of simplicity and in order to reduce traffic load, we have decided to select the path according to the shortest path criteria. Thus, the well-known Dijkstra algorithm has been implemented.

Given the first path, the following steps consist in iteratively repeating two additional steps: compute another path between the same nodes (m_1, m_2) and verify that the newly computed path is such to add a linearly independent equation to the system (2). Hence, as before, the idea is that of selecting the shortest path between the two nodes, excluding the already computed ones. For this purpose we have used Yen algorithm. Once a new path has been computed, the algorithm verifies its independency with respect to the previously computed ones, by applying the Gauss reduction method. Hence, given that we want to minimize the number of employed probing paths, if the new path results to be independent from the already accepted ones it will be selected, otherwise it will be discarded.

Once all the possible paths between a given couple of monitoring nodes have been tested, the algorithm selects a new couple of monitoring points and repeats the described procedure.

The algorithm stops when $nPaths$ linearly independent paths are selected. In such a way, we are able to efficiently and automatically select all the paths required for a full coverage of the network, minimizing simultaneously the number and size of selected paths. Indeed, those which are not strictly necessary are automatically discarded, thus reducing the probing traffic.

It is worth noticing that, in case the network is not identifiable (i.e., $nPath < n_E$), the algorithm can still be used to find all the possible linear independent probing path. Alternative techniques to find additional "heuristic" constraints that allow to approximately solve the system (2) may be used [4].

Algorithm 1 Path Computation

```
Input: G = (V, E), B
P = 0
r = 1
for (m_a, m_b) in B do
    j = 1
    Find the jth shortest path (p_j)
    if p_j lin. independent from [p1, p2, , pj − 1] then
        add p_j to P
        j ← j + 1
        r ← r + 1
    else
        discard the path
    end if
    if r==nPaths then
        Break
    end if
end for
Output: P
```

A. Algorithm Complexity

Our algorithm uses Yen algorithm to determine a potential probing path and the Gauss reduction method to verify the independency of the computed paths. The first algorithm has a complexity of $O(k{\cdot}n_V{\cdot}(n_E{+}n_V{\cdot}log(n_V)))$, with k number of computed paths [7], [8], whereas the gauss reduction method, in our case, has a complexity of $O(nPath^3)$.

Thus, the overall complexity of our algorithm is $O(k \cdot n_V \cdot (n_E + n_V \cdot log(n_V)) + nPath^3)$.

We made some evaluation on the scalability of such algorithm considering realistic network scenarios. We took as

reference processing units an Intel desktop i7 3900 processor operating at 3.066 GHz, that, according to Intel specifications, has a processing capacity of 182 GFLOPS in the boosted configuration and a processing acccelerator for workstation based on TESLA K40 GPU by NVIDIA able to reach more than 4 TFLOPS. We considered three networks segmented in 5 regions with $20k$, $50k$ and $100k$ nodes respectively all with a meshing degree of 2, which represent cases close to practical situations.

TABLE III
COMPUTATION TIME ESTIMATION

Network nodes	$i7$	TESLA
$20k$	5 min	1 min
$50k$	6 hours	16 min
$100k$	49 hours	125 min

It is worth noticing that the algorithm, when trying to identify the whole network, typically runs off-line so time is not a big issue. However a common processor is also able to make computation up to 20k nodes in few minutes and, in our opinion, it is still applicable up to 50k nodes. For larger networks a processing accelerator is recommended in order to shorten the computation time. The TESLA GPU provides very interisting results in all cases operating close to real-time up to 20k nodes and with very reasonable execution times for network up to 100k nodes.

IV. RESULTS

In this section we present the results of the simulation carried out to validate our algorithm. The proposed algorithm was tested by simulations using several topologies. The algorithm was coded in C++ and integrated into a main program, written in Java, including a dedicated GUI. The simulation took as inputs the topology G and the set B of probing nodes, and provided as output the set P of probing paths. It is worth noticing that the output set is composed by the minimum number $N = n_E$ of distinct necessary independent probing paths if the network is identifiable, and by $nPath < n_E$ distinct independent probing paths if the network is unidentifiable.

The developed GUI, is used to input the studied topology and to obtain a textual and graphical representation of the probing paths selected by the method. A screenshot of the output GUI is provided in figure 2.

Concerning the tests over simulated topologies, we used the topology generator BRITE [9] [10] to generate different topologies with a number of nodes n_V in the set $\{50, 100, 200, 500, 1000\}$ and a mesh degree in the range $\{2, 5, 10\}$, so as to test the method over a wide range of different kinds of topologies. The probing set B was randomly chosen, considering 10% of the total nodes as possible monitoring points. Moreover, given that the topologies were randomly generated, we obtained both identifiable and

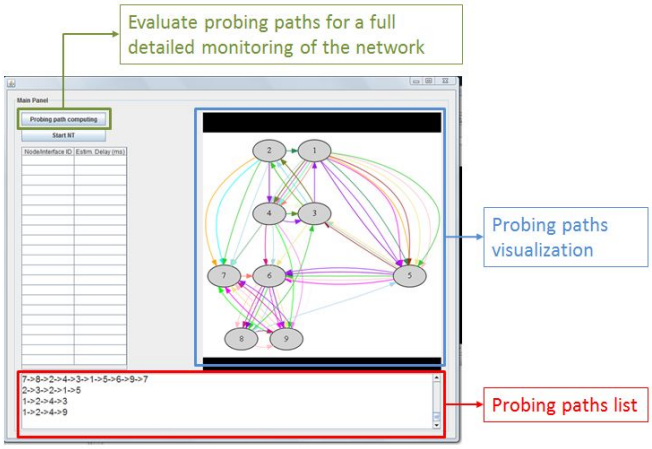

Fig. 2. GUI

unidentifiable topologies. For the first ones the method was always able to determine the minimum set P (i.e., composed of n_E distinct independent probing paths, whereas in case of unidentifiable networks, the method was always able to select $nPath$ distinct independent probing paths (with $nPath$ defined as in (4)).

The results in table IV show that in all network configurations the number of paths selected by our method is consistent with the theoretical value nPath. From table IV it is also possible to notice that for very low mesh degree there is a lower possibility of finding linear independent paths due to poor branching.

TABLE IV
SPERIMENTAL RESULTS

Network	nV	Mesh degree	nPath	Selected Paths
1	50	2	23	23
2	100	2	49	49
3	200	2	94	94
4	500	2	307	307
5	1000	2	456	456
6	50	5	85	85
7	100	5	172	172
8	200	5	336	336
9	500	5	831	831
10	1000	5	1739	1739
11	50	10	212	212
12	100	10	435	435
13	200	10	862	862
14	500	10	2268	2268
15	1000	10	4489	4489

V. CONCLUSIONS

The applicability of active NT is limited by the difficulty of finding the correct and minimal set of probing paths to be used for establishing the monitoring connections.

In this paper we have faced such a problem, by proposing a novel general algorithm, based on the combined use of Yen algorithm and Gauss reduction method, capable of automatically

determining the minimal set of $nPath$ independent probing paths which can provide the largest monitoring coverage possible with the specified network. In case of identifiable networks $nPath = n_E$ (i.e., the minimum possible value of independent path in case of identifiable network), whereas $nPath$ is equal to the maximum number of independent paths in case of unidentifiable networks.

The tests, carried out in a simulated environment, have shown the effectiveness of the proposed method. The scalability analysis made with networks representing typical cases in field has shown that the new algortihm is able to compute the set of probing paths in a reasonable time with almost real-time performances for networks up to 20k nodes.

Being general, practical and scalable, the proposed method improves significantly the possibility of applying NT in field, allowing a better integrated monitoring of heterogeneous networks and monitoring of networks with reduced OAM functionalities, like IP-MPLS ones, with a solution that is completely transparent to the network.

REFERENCES

[1] L. Ma, T. He, K. Leung, A. Swami, and D. Towsley, "Topological constraints on identifying additive link metrics via end-to-end paths measurements," 2012.

[2] Q. Zheng and G. Cao, "Minimizing probing cost and achieving identifiability in network link monitoring," in *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference*, 2010, June, pp. 675–684.

[3] A. Chen, J. Cao, and T. Bu, "Network tomography: Identifiability and fourier domain estimation," in *Signal Processing, IEEE Transactions on*, vol. 58 (12), 2010.

[4] K. Claffy, T. Monk, and D. McRobb, "Internet Tomography." Macmillan Publishers, Jan 1999.

[5] B. Henderson-Sellars, "Modularization and mccabe's cyclomatic complexity," in *CACM*, vol. 35 (12), 1992.

[6] B. Henderson-Sellars and D. Tegarden, "The theoretical extension of two versions of cyclomatic complexity to multiple entrylexit modules," in *Software Quality Journal*, vol. 3(4), 1994.

[7] Yen and Y. Jin, "Finding the k shortest loopless paths in a network," in *Management Science*, vol. 17 (11), 1971.

[8] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," in *Journal of the Association for Computing Machinery*, vol. 34 (3), 1987.

[9] A. Medina, I. Matta, and J. Byers, "Brite: A flexible generator of internet topologies," Boston, MA, USA, Tech. Rep., 2000.

[10] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: An approach to universal topology generation," in *Proceedings of the Ninth International Symposium in Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, ser. MASCOTS '01. IEEE Computer Society, 2001.