Loss Tomography in General Topologies with Network Coding

Minas Gjoka
UC Irvine
mgjoka@uci.edu

Christina Fragouli
EPFL, Lausanne
christina.fragouli@epfl.ch

Pegah Sattari
UC Irvine
psattari@uci.edu

Athina Markopoulou
UC Irvine
athina@uci.edu

***Abstract*— Network tomography infers internal network characteristics by sending and collecting probe packets from the network edge. Traditional tomographic techniques for general topologies typically use a mesh of multicast trees and/or unicast paths to cover the entire graph, which is suboptimal from the point of view of bandwidth efficiency and estimation accuracy. In this paper, we investigate an active probing method for link loss inference in a general topology, where multiple sources and receivers are used and intermediate nodes are equipped with network coding, in addition to unicast and multicast, capabilities. With our approach, each link is traversed by exactly one packet, which is in general a linear combination of the original probes. The receivers infer the loss rate on all links by observing not only the number but also the contents of the received probes. In this paper: (i) we propose an orientation algorithm that creates an acyclic graph with the maximum number of identifiable edges (ii) we define probe combining coding schemes and discuss some of their properties and (iii) we present simulation results over realistic topologies using Belief-Propagation (BP) algorithms.**

I. INTRODUCTION

Network monitoring is a necessary component of the diagnosis and operation of any network. Monitoring link loss rates, in particular, is a useful input to various control and traffic engineering decisions at the network and application layers. Over the past decade a significant research effort has been devoted on a class of monitoring problems, known as loss tomography, which aim at inferring link loss rates using active *end-to-end* measurements, i.e., probes sent and collected from/at the network edge [1]–[4]. One of the attractive attributes of tomography is that it does not need the coordination of internal nodes, which may be difficult or impossible in large networks with distributed control.

Although there is a very good understanding of this problem for tree topologies [1], link loss tomography over general graphs with an arbitrary structure is a challenging problem. The existing approaches for general graphs use multiple multicast trees and/or multiple unicast paths to cover the network graph, and then combine the link loss rates estimated from the different paths/trees [2]–[4]. These approaches are suboptimal with respect to the following optimality criteria: (i) how many links of the network we can infer (identifiability), (ii) how well we can infer them (estimation accuracy) and (iii) how many probes we need to send (bandwidth efficiency).

In this paper, we explore a different approach, originally proposed in [5], [6], which uses multiple sources and receivers and assumes that intermediate nodes are equipped not only with multicast and unicast but also with network coding capabilities [7]–[9]. Our approach can be summarized as follows. Given a selected set of sources, an orientation algorithm determines the paths to be followed by probes, so as to avoid cycles, and the nodes that will serve as receivers. Then, sources send appropriately chosen probe packets; intermediate nodes linearly combine incoming probes (over an appropriately chosen finite field) and multicast the output to all outgoing links; finally, receivers use the number and the content of received probes to infer the loss rates of all internal links. The following example illustrates this operation.

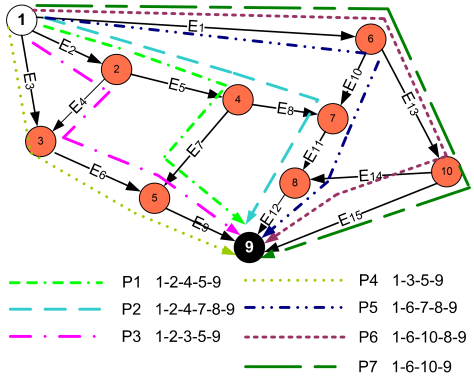


Fig. 1. Example of a general topology (based on Abilene). For one source (node 1), we show the orientation of edges, the resulting receiver (node 9) and the possible paths from the source to the receiver.

Example 1: Consider the configuration shown in Fig. 1: it is based on the Abilene backbone topology [11] and consists of 10 nodes and 15 edges $E_1, ..E_{15}$; our purpose is to infer the loss rates on each edge. If node 1 is used as a source, the orientation algorithm of Section III selects node 9 as receiver and 7 paths $P_1, ...P_7$ from node 1 to node 9 so as to avoid cycles. Once the orientation is determined, several probes are sent from the source and collected at the receiver.

In each experiment, the source sends probes x_1, x_2 and x_3 to the outgoing edges 1, 2 and 3 respectively. Nodes 2, 4, 6, 10 simply forward their incoming packets to all their outgoing links. Node 3 performs coding operations as follows: if within a predetermined time-window it receives only probe x_2, it simply forwards this packet; similarly if it receives only probe x_3. If, however, it receives both x_2 and x_3, it linearly combines them to create packet $x_2 + x_3$ and sends it through outgoing

edge E_6. Nodes 5, 7 and 8 follow a similar strategy: if all links are functioning, node 5 sends packet $3x_2 + x_3$, node 7 sends packet x_1+x_2, and node 8 sends packet $3x_1+x_2$. In the network coding terminology, nodes $3, 7$ use coding coefficients $[1, 1]$ and nodes $5, 8$ use coding coefficients $[1, 2]$; clearly, the choice of coefficients plays an important role.

From the content of the received probes, node 9 can infer the state of the paths and eventually the state of the links. E.g. if it receives only x_3, it infers that all paths from the source S have failed except for path P_4; therefore, edges E_3, E_6, E_9 worked and all other edges failed. Similarly, the receiver can infer the state of the links from any combination of received probes, assuming that the paths and coding scheme are properly selected. This is the goal of this paper. □

This paper builds on our prior work in [5] (where we introduced the idea of using network coding to improve network monitoring) and in [6] (where we studied link loss estimation in tree topologies) and extends it to general topologies. Our approach uses exactly one probe per link and avoids suboptimal combination of observations from different trees, which can have been weaknesses of traditional tomography in general graphs [2].[1] However, our approach also faces novel challenges in dealing with cycles and link identifiability.

The paper is structured as follows. Section II states the problem and discuss the challenges in dealing with cycles and identifiability in general graphs. Sections III and IV present a first attempt to address them. Section VI presents simulation results over realistic topologies and using a message-passing estimation algorithm. Section VII concludes the paper.

II. Problem Statement and Challenges

Consider an undirected graph $G = (V, E)$ where V is the set of nodes and E is the set of edges corresponding to logical links[2] connecting the nodes. Each link $e \in E$ has a loss probability (or rate) α_e associated with each direction, which we are interested in estimating. We want to estimate the link loss rate associated with both directions of all links. We are allowed to use some nodes as sources and receivers; we assume that intermediate nodes are equipped with unicast, multicast and network coding capabilities; we want to infer the loss rate on every edge based on observations on the receivers.

Link loss inference in general graphs faces several novel challenges. Here, we discuss two of them.

A. Challenge I: Graphs with Cycles

In our approach, intermediate nodes simply combine their incoming packets and forward them towards all their outgoing links, in a distributed manner, and without a global view of the network. Employing this mode of operation over a network with cycles may result in probes getting trapped inside a cycle, in a positive feedback loop that consumes network resources without aiding the estimation process. The following example illustrates such a situation.

Example 2: Consider again the network shown in Fig. 1, but now assume that the orientation of edges E_4 and E_6 were reversed. Thus edges E_4, E_5, E_7, and E_6 create a cycle between nodes 2, 4, 5, and 3. The probe packets injected by nodes 3 and 2 would not exit this loop. □

To address this problem, one could equip intermediate nodes with additional functionalities, such as removal of packets that have already visited the same node. This is not practical because it requires keeping state at intermediate nodes; furthermore, such operations would need to be repeated for every set of probes, leading to increased processing and complexity.

In this paper, we take a different approach. We assume that we have the freedom to select -a small number of- nodes that can act as sources or receivers of probe packets. We then propose an algorithm that, starting from a small set of source nodes, selects a graph orientation and a set of receivers such that the resulting graph does not contain any directed cycles. Simulations showed that the resulting number of receivers from the proposed algorithm tends to be quite small. We discuss this approach in Section III.

Given the identified graph orientation, we can estimate the loss rate of all links in one direction. We can then reverse the orientation of all links, and the role of the sources and receivers, to create again an acyclic orientation that allows to estimate the loss rates of the links in the opposite direction.

B. Challenge II: Identifiability

Consider a graph $G = (V, E)$, a set $S \subset V$ of sources and a set $R \subset V$ of receivers of probes. Assume that intermediate nodes in the network are only allowed to perform linear operations over a finite alphabet.

Definition 1: A link is said to be *identifiable* under a given monitoring scheme (choice of sources, receivers, intermediate node operations) if its associated loss rate can be reliably inferred from the measurements observed at the receivers.

In tree networks, we have derived identifiability criteria and proved that it is sufficient for intermediate nodes to simply perform `xor` operations [5]. However, in general graphs, even if there are no cycles, `xor` operations are no longer sufficient.

Example 3: Consider the network and edge orientation shown in Fig. 1, but now assume that intermediate nodes are only allowed to do `xor` operations. Notice that paths P_3 and P_1 overlap twice: on edge E_2, and later on edge E_9. If all links in both paths function, the `xor` operations cancel each other out, resulting in the same observation with the case that both paths are disrupted. More specifically, the following two events become indistinguishable:

(i) *all edges function:* node 5 receives packet x_2 through edge E_7 and packet $x_2 + x_3$ through edge E_6, and sends packet x_3 through edge E_9 to the receiver

[1] Comparison to traditional tomography is out of the scope of this short paper. Here, we assume that the network under study is equipped with unicast, multicast and network coding capabilities, and we look for the best way to exploit these capabilities for link loss tomography. Clearly network coding does not come for free and is not implemented in the routers today; neither is multicast. However, network coding is gaining momentum in wireless and overlay networks and we expect it to be part of these networks in the future.

[2] Logical links result from combining several physical links in cascade into a single link, and thus lead to a graph G where no vertices have degree two. It is well-known that only logical links are identifiable.

Algorithm 1 Orientation Algorithm: Given graph $G = (V, E)$ and senders $S \subset V$, find receivers $R \subset V$ and orientation $\forall\ e \in E$, s.t. there are no cycles and all edges are identifiable.

1: **for all** edges $e = (s, v_2) \in S$ **do**
2: Set outgoing orientation $s \rightarrow v_2$
3: **end for**
4: $R = \{s \in S$ that have incoming oriented edges $\}$
5: $V_1 = S$;
6: $V_2 = \{v_2 \in V - V_1 : s.t.\ \exists edge\ (v_1, v_2)$ from $v_1 \in V_1\}$
7: **while** $V_2 \neq \emptyset$ **do**
8: Identify and exclude receivers: find $r \in V_2$ without unset edges: $R \leftarrow R \bigcup \{r\}$; $V_2 \leftarrow V_2 - \{r\}$
9: Find nodes $U_1 \subset V_2$ that have the smallest number of edges with unset orientation.
10: Find nodes $U_2 \subset U_1$ that have the minimum distance from the sources S. Choose one of them: $v^* \in U_2$.
11: Let $E^* = \{(v^*, w) \in E$ s.t. $w \in V - V_1\}$
12: **for all** edges $(v^*, w) \in E^*$ **do**
13: set direction to $v^* \rightarrow w$
14: **end for**
15: Update $V_1 \leftarrow V_1 \bigcup \{v^*\}$
16: Update $V_2 \leftarrow \{v_2 \in V - V_1 :\ \exists edge\ (v_1, v_2),\ v_1 \in V_1\}$
17: **end while**

(ii) *edges E_4 and E_7 fail and all other edges function:* node 5 only receives packet x_3 from its incoming links, and again sends packet x_3 through edge E_9 to the receiver

On the other hand, if we do as in Example 1, i.e. allow coding operations over a larger alphabet and carefully select coding coefficients $[1, 1]$ at nodes $3, 7$ and $[1, 2]$ at nodes $5, 8$, then the above two events result in observing the distinct packets $3x_2 + x_3$ and x_3 at the receiver. This is a feasible solution but not necessarily unique. □

We discuss identifiability and coding scheme selection (alphabet size and coding coefficients) in Section IV.

III. Removing Cycles

Assume that we are given an undirected graph $G = (V, E)$, where the degree of each node is either one or at least three; this is indeed the case when we consider only logical links. Our goal is, starting from a set of nodes that act as senders $S \subset V$, to select an orientation of the graph and a set of receivers, so that (i) the resulting graph is acyclic, (ii) the orientation allows the maximum number of links to be identifiable, and also (iii) attempts to minimize the number of receivers.

We propose Algorithm 1, which achieves these goals by sequentially visiting the vertices of the graph, starting from the sources, and selecting an orientation for all edges of the visited vertex. This orientation can be thought as imposing a partial order on the vertices of the graph: no vertex is visited before all its parent vertices in the final directed graph.

We now describe Algorithm 1. Lines $1 - 3$ attempt to set all links attached to the sources as outgoing. If we allow an arbitrary selection of sources we may fall into cases where sources contain links to other sources. In this case, one of the sources will also need to act as a receiver, i.e., we allow the set S of sources and the set of receivers R to overlap. In the main part of the algorithm nodes are divided in three sets:

- Set V_1 contains nodes that have been already visited and have orientation already assigned to all their attached edges. Initially $V_1 = S$.
- Set V_2 contains nodes that are one edge away from V_1. These are the next candidates to be added to V_1.
- The remaining nodes are either receivers, R, or nodes that have not been visited yet $V_3 = V - V_1 - V_2 - R$.

In each step of the algorithm, one node $v^* \in V_2$ is selected, all its edges that do not have an orientation are set to outgoing, and v^* is added to $V_1 \leftarrow V_1 \bigcup \{v^*\}$. Notice that the orientation of edges going from V_1 to V_2 is already set. However, a node $v \in V_2$ may have additional unset edges; if it does not have unset edges, then it becomes a receiver $R \leftarrow R \bigcup \{v\}$.

We include two heuristic criteria in the choice of $v^* \in V_2$: (i) we look at nodes with the smallest number of unset edges; (ii) if there are several such nodes, then we select the node with the shortest distance from the sources S; if there are still several nodes, we pick one at random. The rationale behind criterion (i) is to avoid creating too many receivers. The rationale behind criterion (ii) is to create a set of paths with roughly the same path length.[3] The algorithm continues until all nodes are assigned to either R or V_1.

Lemma 3.1: Algorithm 1 produces an acyclic orientation.

Proof: In each step, a node is selected and all its edges which do not already have a direction are set as outgoing. This sequence of selected nodes is a topological ordering. At any point of the algorithm, there are directed paths from nodes considered earlier to nodes considered later. A cycle would exist if and only if for some nodes v_i and v_j: v_j is selected at steps $j > i$ and the direction on the undirected edge (v_i, v_j) is set to $v_i \leftarrow v_j$. This is impossible: if there were an edge (v_i, v_j) it would have been set at the earlier step i at the opposite direction $v_i \rightarrow v_j$; therefore, the resulting directed graph has no cycle. However, there may be nodes without any outgoing edges, which become the receivers. ■

The key point that enables us to create an acyclic orientation for an undirected graph is that the receivers are one of the outputs of the algorithm. Notice that a similar algorithm can be formulated for the symmetric problem, when the receivers R are given and the orientation algorithm produces a (reverse) orientation and a set of sources S, so that there are no cycles. However, if both S and R are fixed, there is no orientation algorithm that guarantees the lack of cycles for all graphs, without introducing additional sources or receivers.

Regarding identifiability, it is easy to see through explicitly constructed examples that in a general undirected graph consisting of logical links, and a fixed given choice of sources,

[3]One could use different criteria to rank the candidate v^*, so as to enforce additional desirable properties on top of identifiability. We used shortest path from the sources to impose a breath-first progression of the algorithm and paths with roughly the same length. We could also optimize for the alphabet size and/or the complexity and performance of the estimation algorithms.

there might not always exist a choice of receivers and orientations such that all links are identifiable. To maximize the number of identifiable edges, Algorithm 1 selects an orientation such that each vertex that is not a source or a receiver, has at least one incoming and at least one outgoing edge. The proof that such an orientation achieves the claimed goal is omitted here for lack of space. We further discuss issues related to identifiability in the next section.

IV. Identifiability and Coding

For a given edge e of a graph, whether it is identifiable or not, depends on two factors, the first being the topological structure and orientation of the network links, and the second being the coding scheme (probe packet combining) employed. Given the set of sources, receivers and link orientation, we can identify the set of paths $\{\mathcal{P}\}$ that connect the sources to all receivers. Let $\mathcal{P}(e)$ denote the set of paths that are routed from a source to a receiver, and employ edge e. The receivers can infer which of these paths were operating during a given experiment and which did not, by observing the received probes. This is the information that they can use to infer link loss rates, together with the knowledge of the topology and the way these paths overlap. It is obvious that two edges e_1 and e_2 are not identifiable if $\mathcal{P}(e_1) = \mathcal{P}(e_2)$. We conjecture that the inverse is also true. Notice that this is a condition on the structural properties of the graph itself.

We additionally require intermediate nodes to employ a suitable probe packet combining scheme. We call such schemes *probe coding* schemes, and we say that a probe coding scheme is *valid*, if it leads to the maximum possible number of identifiable links, which is determined by the structure of the graph. Unlike tree configurations, the probe coding over a general topology may need to perform operations using a larger alphabet, as we discussed in Example 3. In this paper, we restrict our attention to linear operations over finite fields, i.e., additions and multiplications. We will say that a coding scheme over a finite field $\mathbb{F}_q$ employs an *alphabet of size* q.

Assume that receiver nodes only have incoming edges, and let e_R be an edge adjacent to a receiver R. Then $P(e_R)$ is the set of paths that connect sources to receiver R and have e_R as their last edge. We say that a probe coding scheme is valid if, by observing the received probes from edge e_R at a given experiment, R can determine which of the $P(e_R)$ paths were functioning during this experiment and which were not. For valid coding schemes we can derive the following loose lower bound on the required alphabet size.

Lemma 4.1: Let $G = (V, E)$ be acyclic and let P_m denote the maximum number of paths sharing an incoming edge of any receiver R, i.e., $\mathcal{P}_m = \max_{e_R} P(e_R)$. Then the alphabet size q is greater than or equal to $\mathcal{P}_m$.

Proof: Assume that one of the $\mathcal{P}_m$ paths is functioning while all the others are not. Since two paths cannot overlap in all edges, there exists a set of edge failures such that this event occurs. For the receiver to determine which of the $\mathcal{P}_m$ paths is functioning it needs to differentiate between at least $\mathcal{P}_m$ distinct values. ■

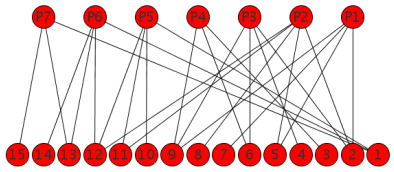


Fig. 2. Factor graph corresponding to the graph of Fig.1 and used for the belief propagation estimation algorithm.

The alphabet size affects the bandwidth efficiency, and is desired to be kept as small as possible. The upper bound depends on the topology as well as on the choice of coding coefficients and is part of ongoing work.[4] In practice, we have noticed through simulation that a random choice of the coding coefficients over a field with size larger than the maximum number of paths sharing an edge leads with high probability to the maximum number of identifiable links.

V. Loss Estimation Algorithm

For our approach to be useful in practice, we need to employ a low complexity algorithm that allows to quickly estimate the loss rate on every link from all the observations at the receiver. Maximum Likelihood Estimator (MLE) is optimal but an efficient implementation is currently known only for multicast trees [1]. Because MLE is quite involved for general graphs, especially, large ones, we use a suboptimal algorithm instead; in particular, we use a Belief Propagation (BP) approach, building on the work in [10]. We briefly summarize this approach in this section and we evaluate its performance through simulations in Section VI.

Following the approach in [10], the first step is to create the factor graph from the original graph. Fig. 2 shows the factor graph for the example network shown in Fig. 1. This is a bipartite graph: on one side there are the links (variable nodes), whose loss rates we want to estimate; on the other side there are the paths (function nodes) that are observed by each received probe. An edge exists in the factor graph between a link and a path if the link belongs to this path in the original graph. Note that, unlike tree topologies considered in [10], in general topologies there might exist multiple paths for every source-receiver pair.

The second step is belief propagation. Each received probe triggers message passing in the factor graph and results in an estimate of link loss probabilities. The estimates from different probes are then combined, using standard methods [10], to provide an estimate ($\hat{\alpha}_e$) for the actual loss probability (α_e) of every link $e \in E$.

This is essentially a parameter estimation problem, and the quality of the estimation for a single link e is captured by the mean-square error $MSE_e = \frac{1}{|k|}\sum_k(|\hat{\alpha}_e - \alpha_e|^2)$, over k realizations. To summarize the quality of the estimation

[4]Using the Sparse Zero Lemma we can show that there exists a field with size $q > L_m$ which guarantees identifiability, where L_m is the maximum number of links in the set of paths $P(e_R)$ for every receiver R. Intuitively this means that we need to distinguish among L_m such possible events. We are currently working on tighter bounds and randomized coding schemes.

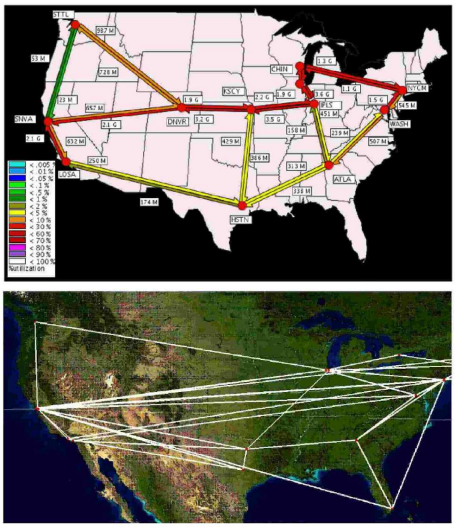


Fig. 3. Topologies used in simulation : Abilene (top) and Exodus (bottom)

across all links $e \in E$, we use an entropy measure ENT that captures the residual uncertainty across all links $ENT = \sum_{e \in E} \log MSE_e$ [6]. These are the metrics we report in our simulations in the next section.

VI. Simulation Results

A. Network Topologies

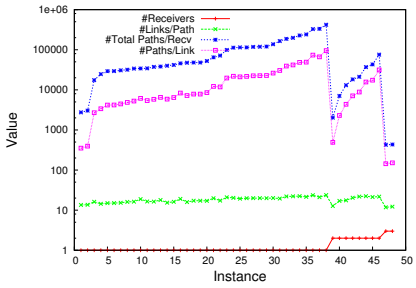
We used two realistic topologies, namely the backbones of Abilene and Exodus shown in Fig. 3. Abilene is a high-speed research network operating in the US and information about its backbone is available at [11]. Exodus is a large commercial ISP, whose backbone map was inferred by the Rocketfuel project [12]. Both topologies were pre-processed to create logical topologies that have degree at least 3. For Exodus, nodes with degree 2 were merged to create a logical link between the neighbors while nodes with degree 1 were filtered; the resulting logical topology contains 48 nodes and 105 links. For the Abilene topology, due to its small size, in addition to some links in tandem merged, more links were added; the modified topology comprises of 10 nodes and 15 links, and is the one shown in Fig. 1 and used as an example throughout this paper.

B. Results on the Orientation Algorithm

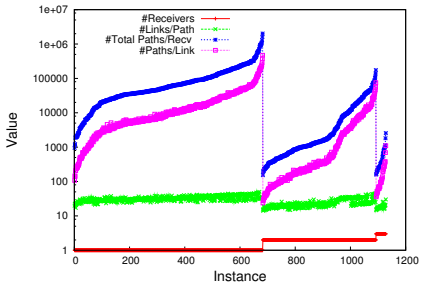
In Fig. 4, we consider the Exodus topology and we run the orientation algorithm for all possible placements of one and two sources; we call each placement an "instance". We are interested in the following properties of the orientation produced by Alg.1:

- the number of receivers: a small number allows for local collection of probes and easier coordination.
- the number of distinct paths per receiver: this affects the alphabet size and it is desired to be small.
- the number of paths per link and links per path: these affect the performance of the belief propagation algorithm.

Fig. 4 shows the above four metrics, sorting the instances first in increasing number of receivers and then in increasing



(a) All possible placements of one source



(b) All possible placements of two sources

Fig. 4. Running the Orientation Algorithm on the Exodus topology.

Topology	Srcs-Recvs	Coding Points	Links / Path	Paths / Link	Edge Disj. Paths
Abilene	{1}-{9}	4	3.85	1.8	3
	{5}-{6}	4	3.71	1.73	3
	{9}-{2}	4	4.28	2.0	2
	{1,9}-{7}	5	3.25	1.73	4
	{3,6}-{9}	5	4	2.13	4
	{9,6}-{4}	5	3.25	1.73	4
	{1,5,9}-{7}	5	3.2	2.13	5
	{1,4,10}-{9}	6	3	2.33	6
Exodus	{39,45}-{30,40}	25	9.47	56.47	4

TABLE I

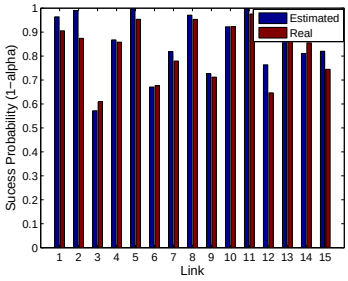
Properties of the orientation graphs produced by Alg.1 for different topologies and choices of sources.

paths/receiver. The following observations can be made. First, the number of receivers produced by our orientation algorithm is indeed very small, as desired. Second the number of links per path is almost constant, because by construction the orientation algorithm tries to balance the paths lengths. Third, the paths/receiver and paths/link metrics, which affect the alphabet size and the quality of the estimation, can be high; however, they decrease by orders of magnitude for configurations with a few receivers; these should be chosen in practice. Finally, Table I considers different choices of sources in the two topologies and shows some properties of the produced orientation.

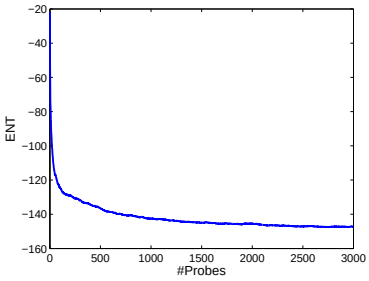
C. Results on Belief-Propagation (BP) Inference

This section presents results on the quality of the link loss estimation for different assignments of loss rates to the links of the two considered topologies. The BP algorithm is used for estimation in all cases. For our simulations the link losses on different links are assumed independent, and may take large values as they reflect losses on logical links, comprising of cascades of physical links, as well as events related to congestion control within the network.

In Fig. 5, we consider the modified Abilene topology with



(a) Estimation per link: estimated vs. real success rate (for 3000 probes.)



(b) Estimation for the entire graph: ENT metric vs. number of probes

Fig. 5. Estimation of loss rates for the Abilene topology, and different loss rates (α's) across links: loss rates have been assumed inversely proportional to the link bandwidth, as reported in [11] (17% average loss rate on average).

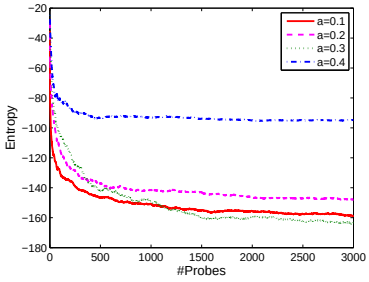


Fig. 6. Abilene topology with same α on all links and one source (at 1).

loss rates inversely proportional to the bandwidth of the actual links. We see that the estimation error for each link (MSE) and for all links (ENT) decreases quickly with increasing number of probes. In Fig. 6 the same topology is considered but with the same α on all links. As expected, ENT decreases with the number of probes, and convergence is faster for larger α's. However, the estimation error is larger for large α's, which is due to the behavior of the belief propagation algorithm.

Similarly, Fig. 7 shows the estimation error ENT for the Exodus topology with uniformly assigned loss rates.

Finally, Table II shows the results for a different number and different placements of sources in the Abilene topology. The case of one source placed at $\{1\}$, discussed as an example throughput the entire paper, is shown in the first row of the table. One can observe that using more sources decreases the total estimation error ENT.

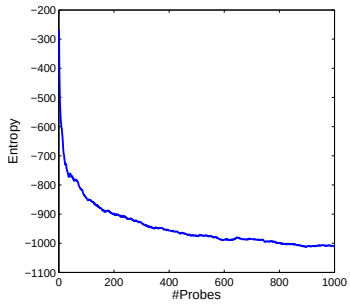


Fig. 7. Exodus topology, with loss rates uniformly distributed in $[1\%, 35\%]$.

Srcs-Rcvs	Entropy for loss rate same over all links					
	a=0.05	a=0.10	a=0.15	a=0.2	a=0.25	a=0.30
{1}-{9}	-178.6	-158.8	-147.9	-147.7	-161.6	-163.5
{5}-{6}	-178.1	-158.3	-149.6	-154.5	-160.4	-156.5
{9}-{2}	-176.1	-163.3	-155.8	-161.2	-166.6	-151.7
{1,9}-{7}	-189.3	-173.9	-166.5	-180.3	-171.7	-156.2
{3,6}-{9}	-186.2	-176.2	-171.3	-177.8	-166.7	-151.4
{9,6}-{4}	-186.9	-174.1	-169.5	-178.7	-173.2	-165.4
{1,5,9}-{7}	-199.8	-190.6	-180.9	-184.4	-172.3	-166.9
{1,4,10}-{9}	-186.4	-183.9	-178.3	-182.3	-177.3	-173.2

TABLE II

QUALITY OF ESTIMATION FOR THE (MODIFIED) ABILENE TOPOLOGY AND FOR DIFFERENT CHOICES OF SOURCE(S).

VII. SUMMARY AND FUTURE WORK

In this paper, we studied the problem of link loss tomography in general graphs that are equipped with unicast, multicast and network coding capabilities. We investigated several aspects including: identifiability as a function of the topology and of the code design, orientation algorithms to avoid cycles, suboptimal estimation using belief-propagation. Future work will (i) further investigate code design to improve identifiability and estimation and (ii) compare our method to traditional tree/path-packing approaches in general graphs.

REFERENCES

[1] R. Caceres, N. G. Duffield, J. Horowitz and D. Towsley, "Multicast-based inference of network-internal loss characteristics", *in IEEE Trans. on Inf. Theory*, vol. 45, pp. 2462–2480, 1999.

[2] T. Bu, N. Duffield, F. Presti, and D. Towsley, "Network tomography on general topologies," *in Proc. ACM Sigmetrics, 2002*.

[3] M. Rabbat, R. Nowak and M. Coates, "Multiple source, multiple destination network tomography", *in Proc. of IEEE Infocom 2004*.

[4] M. Coates and R. Nowak, "Network loss inference using unicast end-to-end measurements", *ITC Seminar on IP traffic, Measurements and Modeling*, Monterey, CA, Sept. 2004.

[5] C. Fragouli, A. Markopoulou, "A network coding approach to overlay network monitoring", *in Proc.of 43rd Allerton*, Sept. 2005.

[6] C. Fragouli, A. Markopoulou, R. Srinivasan, S. Diggavi, "Network Monitoring: It Depends on your Points of View", *in Proc. of ITA Workshop*, San Diego, CA, Jan. 2007.

[7] C. Fragouli. J. Widmer and J.Y. LeBoudec, "Network coding: an instant primer", *in ACM SIGCOMM CCR*, vol. 36(1), Jan. 2006.

[8] R. Ahlswede, N. Cai, S-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. on Inf. Theory*, vol. 46, pp. 1204-1216, July 2000.

[9] S-Y. R. Li, R.W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. on Inf. Theory*, vol. 49, 2003.

[10] Y. Mao, F. R. Kschischang, B. Li and S. Pasupathy, "A factor graph approach to link loss monitoring in wireless sensor networks", *in IEEE JSAC*, vol. 23, pp. 820-829, April 2005.

[11] The Abilene Research Network, *http://abilene.internet2.edu/*.s

[12] N.Spring, R.Mahajan, D.Wetherall, "Measuring ISP Topologies with Rocketfuel," *in Proc. of ACM SIGCOMM 2002*.