Multiple Source Multiple Destination Topology Inference using Network Coding

Pegah Sattari
University of California, Irvine
psattari@uci.edu

Athina Markopoulou
University of California, Irvine
athina@uci.edu

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

Abstract — **In this paper, we combine network coding and tomographic techniques for topology inference. Our goal is to infer the topology of a network by sending probes between a given set of multiple sources and multiple receivers and by having intermediate nodes perform network coding operations. We combine and extend two ideas that have been developed independently. On one hand, network coding introduces topology-dependent correlation, which can then be exploited at the receivers to infer the topology [1]. On the other hand, it has been shown that a traditional (*i.e.,* without network coding) multiple source, multiple receiver tomography problem can be decomposed into multiple two source, two receiver subproblems [2]. Our first contribution is to show that, when intermediate nodes perform network coding, topological information contained in network coded packets allows to accurately distinguish among all different 2-by-2 subnetwork components, which was not possible with traditional tomographic techniques. Our second contribution is to use this knowledge to merge the subnetworks and accurately reconstruct the general topology. Our approach is applicable to any general Internet-like topology, and is robust to the presence of delay variability and packet loss.**

I. Introduction

Network tomography aims at inferring internal network characteristics using end-to-end probes, putting the processing burden on a few end-nodes and keeping internal nodes simple. Such information is useful for network diagnosis and management. There is a body of work in the tomography literature that infers the topology using multicast, unicast, or back-to-back probes and relying on the number and order of received probes.

In this paper, we revisit the problem of topology inference using end-to-end probes between a given set of M sources and N receivers, *i.e.,* in an "M-by-N network", with network coding capabilities. We assume a unique path from each source to each receiver, which is considered known and determined by the routing protocol, as it is the case in the Internet. We show that by allowing internal nodes to do simple network coding operations, we can use the received probes to infer the logical topology accurately and faster than traditional approaches. The key intuition is that network coding introduces topology-dependent correlation in the content of probes, which can then be reverse-engineered to infer the topology.

We break the problem into two steps. First, we build on the observation that an M-by-N network can be decomposed into a collection of 2-by-2 subnetwork components [2, 3], each of which can be of four possible types. We show that when network coding is used, the type of each 2-by-2 component can be exactly identified, which was impossible with previous approaches [2, 3]; furthermore, this can be achieved using a smaller number of probes. The intuition is that, with a simple appropriate code design, the content of the received probes can be used to uniquely identify the type of a 2-by-2 topology. We define the network coding, probing and reconstruction algorithms to be used at the intermediate, source and receiver nodes, respectively, that achieve this goal. The second step is to merge the 2-by-2 components to reconstruct a 2-by-N network in two scenarios. (i) Assuming knowledge of a 1-by-N topology, we identify the points where a second source's topology joins the known topology; this scenario is the one studied in [2, 4], but our algorithm is simpler, faster and more accurate. (ii) Without knowledge of any 1-by-N topology, we show how to infer the 2-by-N network only by merging 2-by-2 components; this scenario is new to our work and makes less assumptions. The 2-by-N case is then generalized to the M-by-N graph. Our approach is applicable to any general graph and is robust to delay variability and packet loss on the links. We showcase the benefits of our approach through simulation of an example Internet topology.

The paper is structured as follows. Section II summarizes related work. Section III states the problem and main intuition behind our algorithms. Section IV presents our algorithms for inferring 2-by-2 components, first considering a lossless (Section IV-A) and then a lossy (Section IV-B) network. Section V explains how to merge the components to reconstruct the entire topology. Section VI presents simulation results that show the advantages of our approach. Section VII concludes the paper.

II. Related Work

There are two bodies of related work: one from the network tomography and the other from the network coding literature. [5] is a good survey of developments in tomography. Here, we focus on inferring the internal topology, *not* link-level characteristics. Tomographic schemes for topology inference require active probing, reliance on the number, order, delay variance and loss of received probes, and heuristic or statistical signal-processing approaches.

Within this body of work, the closest to this paper is the work by Rabbat, Coates and Nowak on multiple source topology inference [2, 3, 4]. They showed that a general M-by-N tomography problem can be decomposed into a collection of 2-by-2 components. They proposed to coordinate transmission of multi-packet probes from the two sources and measure the packet arrival order at the two receivers to infer *some* information about the 2-by-2 topology. Assuming knowledge of a 1-by-N topology, they then use this 2-by-2 information to merge a second source's 1-by-N topology with the first. This way, they infer some information about the M-by-N topology, *i.e.,* bounds on the locations of the joining points, which is not guaranteed to be exact and requires thousands of probes to obtain statistically significant information.

Independently, network coding ideas have been recently applied to tomography problems. In [6], we revisited link-loss (but not topology) tomography using active probing and network coding. The closest to this paper is our preliminary work in [1], where we showed that active probes from two sources and XOR at intermediate nodes are sufficient to infer the topology of a *binary tree.* The insight was that the topology-dependent correlation among

978-1-4244-4724-4/09/$25.00 © 2009 IEEE

Algorithm 1 Operation at Joining Point J. When two sources multicast to N receivers, J knows that it has two incoming links and one outgoing link. Additions are over $\mathbf{F}_q$.

```
for every time window W do
    if (J receives 2 packets within W from its incomings) then
        as soon as the last one arrives, it adds them up, and forwards the resulting packet downstream
    else if (J receives only one packet within W) then
        it forwards the packet downstream
    else if (J does not receive any packet within W) then
        /*nothing to do*/
    end if
end for
```

Algorithm 2 Operation at Branching Point B. While two sources multicast to N receivers, B has one incoming packet and multiple outgoing links.

```
for each incoming packet do
    if the incoming packet is x1 (or x2) then
        forward it only on the outgoing links that are next hops for S1 (S2)
    else
        /* The incoming packet is of the form ax1 + bx2. */
        forward the packet to all outgoing links
    end if
end for
```

probes, introduced by network coding, can be used to cluster the leaves in the binary tree into groups, uniquely connect the groups, and then proceed iteratively in a hierarchical clustering way. This approach generalizes to non-binary trees, but not to arbitrary graphs. In this paper, we use the same basic intuition but a different approach for general graphs: instead of iteratively dividing the network into smaller clusters (top-down), we identify the types of 2-by-2 components and then merge them together in an M-by-N topology (bottom-up).

The following papers also used random network coding for *passive* network tomography. In [7], passive techniques have been used to distinguish among failure patterns. In [8, 9, 10], subspace properties at various nodes have been used for topology inference. In [8], each node passively infers the upstream network topology at no cost to throughput but at high decoding complexity. All these papers consider that random network coding is present for the primary purpose of increasing throughput. Passive topology inference is just a side benefit. In contrast, we propose active probing and a simple, but specifically designed, coding scheme at intermediate nodes, to achieve low-complexity topology inference at the end nodes.

Our work differs from traceroute approaches, which record node ids along a path, in that we identify multiple source multiple receiver network structures. The logical topology is also revealed without revealing the node ids.

III. Problem Statement

Logical Topology. We define an *M-by-N topology* as a directed acyclic graph, between M sources and N receivers, along with a given routing policy that maps each source-destination pair to a single route from the source to the destination. The single-path routing assumption implies the following three *properties of routing.* (1) There is a unique path from each source to each destination. (2) Two paths from the same source to different receivers take the same route until they branch, so that all 1-by-2 components have the "inverted Y" structure; the node where the paths to the two receivers split is called a *branching point.* (3) Two paths from different sources to the same receiver use exactly the same set of links after they join, so that all 2-by-1 components have the "Y" structure; the node where the paths from the two sources merge is called a *joining point.* These assumptions are realistic, the same as in [2], and consistent with the routing behavior in the Internet: the next hop taken by a packet is determined by a routing table lookup on the destination address. We are interested in inferring the *logical* topology, specified by the branching and joining points where the measured end-to-end paths meet.[1] W.l.o.g., we present most of our discussion in terms of $M = 2$, *i.e.,* inferring a 2-by-N topology; an M-by-N topology can then be constructed by merging smaller structures.

Other assumptions. We assume that the link delay has a fixed part, *i.e.,* the propagation delay, and a random part, due to queueing of cross-traffic, and is independent across links. Internet link delays are in the order of a few to hundreds of ms. Regarding packet loss, we consider both absence and presence of i.i.d. random loss. We assume a coarse synchronization (*i.e.,* a synchronization offset on the order of 5-10ms) across the network nodes, which is achievable via a handshaking scheme, *e.g.,* NTP.

Problem Statement. Our goal is to design active probing schemes, *i.e.,* the operation of sources, intermediate nodes and receivers, which allow us to infer the logical topology from the observations.

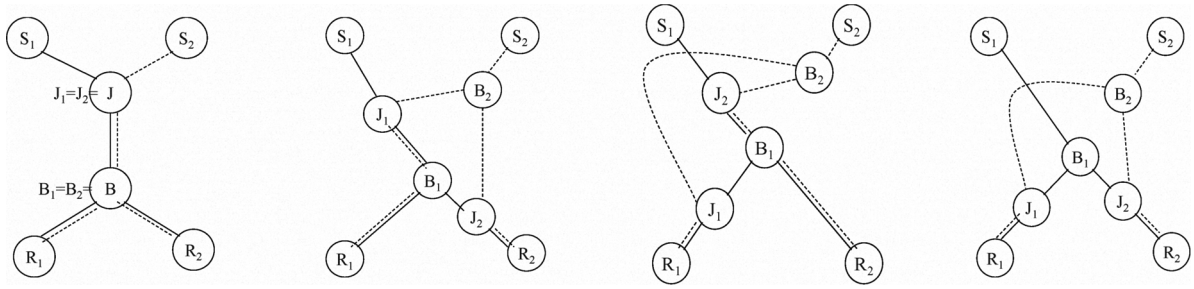
Sources. A pair of sources S_1 and S_2 multicast $x_1 = [1, 0]$ and $x_2 = [0, 1]$, respectively, to all N receivers. They send up to $countMax$ sets of coordinated probes (experiments); or less, if we can deterministically infer the topology earlier. Successive sets of probes are spaced apart by a large interval T, in order to make these experiments independent. Each subnetwork from one source to the N receivers forms a 1-by-N tree; the general graph is a multiple-tree network [2]. We begin with the assumption that the two sources are synchronized and we later relax this assumption by using a sufficiently large time window W at intermediate nodes. Some algorithms also introduce an artificial offset u: a difference in the sending time at the two sources; w.l.o.g., we assume that S_1 sends first and S_2 second. The choice and relation of the timing parameters, W, T, and u, are discussed later.

Intermediate nodes. Their operations are described in Algorithms 1 and 2. Essentially, a joining point (J) adds and forwards packets, while a branching point (B) forwards the single received packet to all "interested" links downstream. A link is "interested" if it is the next hop for at least one source packet in the network code. Additions are performed over a finite field $\mathbf{F}_q$. To ensure that packets from two sources meet at the joining points despite link delays, we require that joining points wait for up to a predetermined time window W. We choose W to be much larger (*i.e.,* on the order of seconds) than the synchronization offset (on the order of 10ms) and than the link delays (on the order of tens or hundreds of ms).

Receivers. Each receiver observes a linear combination of x_1 and x_2, as the result of additions at the joining points. For a 2-by-2 subnetwork, let the observations be $R_1 = c_{11}x_1 + c_{12}x_2$, $R_2 = c_{21}x_1 + c_{22}x_2$. We design algorithms that infer the topology from these observations.

Problem Statement (Refined) and Intuition. The first step in inferring an M-by-N network is to distinguish among all possible 2-by-2 topologies. The second step is to merge these subnetworks to construct the M-by-N network. The intuition behind using network coding in

[1] Intermediate nodes in a logical topology have degree at least three, and in-degree and out-degree at least one. A link in the logical topology may consist of one or more physical links.



(a) type (1): shared (b) type (2): non-shared (c) type (3): non-shared (d) type (4): non-shared

Fig. 1: The four possible types of a 2-by-2 subnetwork. There are two sources (S_1, S_2) multicasting packets x_1, x_2 to two receivers (R_1, R_2). (The 1-by-2 topology of S_1 is a tree composed of S_1, B_1, R_1, R_2. Similarly, The 1-by-2 tree rooted at S_2 is S_2, B_2, R_1, R_2. J_1 and J_2 are the joining points, where the paths from S_2 to R_1 and R_2, join/merge with S_1's topology.)

Tab. 1: **Lossless Case.** Possible observations for types (1) and (4) 2-by-2 topologies. (Observation #1 occurs when the sources are synchronized. Observations #2-4 occur when S_2 sends with offset $u \in [0, W]$ after S_1.)

Observation Number	Type (1)		Type (4)	
	$\mathbf{R_1}$	$\mathbf{R_2}$	$\mathbf{R_1}$	$\mathbf{R_2}$
1	$x_1 + x_2$	$x_1 + x_2$	$x_1 + x_2$	$x_1 + x_2$
2	x_1	x_1	x_1	x_1
3			$x_1 + x_2$	x_1
4			x_1	$x_1 + x_2$

the first step, is that linear operations at internal nodes result in some observations, whose content *uniquely characterizes* the underlying 2-by-2 subnetwork. We use coordinated multicast packets from two sources and carefully adjust the parameters $u, W, countMax$ to create such observations; we design proper algorithms at the receivers to process the observations and infer the topology.

IV. IDENTIFYING 2-BY-2 SUBNETWORK COMPONENTS

We now describe how to distinguish among different 2-by-2 topologies. There exist four 2-by-2 types, as shown in Fig. 1, which were first identified in [2, 3]. We refer to Fig. 1(a), (b), (c), and (d) as type (1), (2), (3), and (4), respectively. Type (1) was called "shared" in [2, 3], since the joining points for both receivers coincide ($J_1 = J_2$) and the branching points for both sources coincide ($B_1 = B_2$). The other three types ((2), (3), (4)) are called "non-shared", since they have two distinct joining points and two distinct branching points. The scheme in [2] only distinguishes between shared and non-shared. In contrast, our approach accurately identifies all four types.

A. Lossless Network

First let us assume that there is no packet loss in the network. In the first experiment, sources S_1, S_2 multicast probes x_1, x_2 to R_1, R_2; S_1, S_2 act simultaneously, or in practice within the synchronization offset. As mentioned in Section III, a choice of large W guarantees that x_1 and x_2 meet at both joining points. J_1, J_2 add the incoming packets over $\mathbf{F}_4$, which suffices for the 2-by-2 case. Thus, R_1, R_2 observe the following in each 2-by-2 type:

- type (1): R_1: $x_1 + x_2$, R_2: $x_1 + x_2$
- type (2): R_1: $x_1 + x_2$, R_2: $x_1 + 2x_2$
- type (3): R_1: $x_1 + 2x_2$, R_2: $x_1 + x_2$
- type (4): R_1: $x_1 + x_2$, R_2: $x_1 + x_2$

Types (2) and (3) result in unique observations that make them distinguishable from any other type, *i.e.,* they are identified in one experiment. However, types (1) and (4) result in similar observations; we need more experiments to get observations that uniquely characterize (1) or (4).

Algorithm 3 Lossless Case - Inferring a 2-by-2 component. Sources S_1, S_2 multicast x_1, x_2. Receivers observe $R_1 = c_{11}x_1 + c_{12}x_2$ and $R_2 = c_{21}x_1 + c_{22}x_2$.

```
1: n=1; /*first experiment*/
2: if c22 > c12 then
3:    Output type (2).
4: else if c22 < c12 then
5:    Output type (3).
6: else
7:    /*It is R1 = R2*/
8:    while n < countMax & R1 == R2 do
9:       Draw offset u uniformly at random out of [0, W].
10:      Send probes; S2 transmits u time later than S1.
11:      if R1 ≠ R2 then
12:         Output type (4); Exit;
13:      end if
14:      n++;
15:   end while
16:   Output type (1); /* It is n == countMax*/
17: end if
```

To design the next experiment, we observe that type (1) is the only 2-by-2 where the two joining points coincide ($J_1 = J_2 = J$). Therefore, the observations at the two receivers are always the same: either $x_1 + x_2$ when the two packets meet at J; or a single packet (x_1 or x_2) when the two packets do not meet at J. In contrast, type (4) has two different joining points $J_1 \neq J_2$. If we force the packets to meet only at one of the joining points but not at the other, the receivers will have different observations. These are observations #3 and #4 in Table 1 and they uniquely characterize type (4). They can be achieved by appropriately selecting the difference between the sources' sending times, *i.e.,* the offset u. u needs to be large enough so that after addition to the link delays, it can affect W. In particular, if D_1, D_2 are the end-to-end delays on the paths from S_2 to J_1, J_2 respectively, then u must be in between $W - D_1$ and $W - D_2$[2] to force different observations at the two receivers.

Alg. 3 summarizes the 2-by-2 inference for lossless networks. Types (2), (3) are identified in the first trial. Type (4) is identified by the first different observation between the two receivers. Otherwise, similar observations at both receivers in $countMax$ trials implies type (1). $countMax$

[2]In 2-by-2 components, this interval is close to W since D_1, D_2 (sum of link delays) are small compared to W. However, in a more general 2-by-N network, there exist multiple links between the sources and joining points. Internet link delays are from a few to hundreds of ms; 500ms is a loose upper bound. W is a few seconds. Thus, we can safely choose $u \in [0, W]$ in the general case.

Tab. 2: **Lossy Case**. Possible observations for all four types of 2-by-2 topologies. (Sources send synchronized and W is large. Observation #13 for types (2) and (3) occurs only when S_2 sends with offset $u \in [0, W]$ after S_1.) We divide the observations into three groups: (i) at least one receiver does not receive any packet (ii) $R_1 = R_2$ (iii) $R_1 \neq R_2$.

Obs. #	Obs. Group	Type (1) $\mathbf{R_1}$	Type (1) $\mathbf{R_2}$	Obs. Group	Type (2) $\mathbf{R_1}$	Type (2) $\mathbf{R_2}$	Obs. Group	Type (3) $\mathbf{R_1}$	Type (3) $\mathbf{R_2}$	Obs. Group	Type (4) $\mathbf{R_1}$	Type (4) $\mathbf{R_2}$
1	(i)	-	-	(i)	-	-	(i)	-	-	(i)	-	-
2		-	$x_1 + x_2$		-	$x_1 + 2x_2$		$x_1 + 2x_2$	-		-	$x_1 + x_2$
3		-	x_1		-	$x_1 + x_2$		$x_1 + x_2$	-		-	x_1
4		-	x_2		-	x_1		x_1	-		-	x_2
5		$x_1 + x_2$	-		-	x_2		x_2	-		$x_1 + x_2$	-
6		x_1	-		$x_1 + x_2$	-		-	$x_1 + x_2$		x_1	-
7		x_2	-		x_1	-		-	x_1		x_2	-
8	(ii)	$x_1 + x_2$	$x_1 + x_2$		x_2	-		-	x_2	(ii)	$x_1 + x_2$	$x_1 + x_2$
9		x_1	x_1	(ii)	$x_1 + x_2$	$x_1 + x_2$	(ii)	$x_1 + x_2$	$x_1 + x_2$		x_1	x_1
10		x_2	x_2		x_1	x_1		x_1	x_1		x_2	x_2
11					x_2	x_2		x_2	x_2	(iii)	x_1	$x_1 + x_2$
12				(iii)	$x_1 + x_2$	$x_1 + 2x_2$	(iii)	$x_1 + 2x_2$	$x_1 + x_2$		$x_1 + x_2$	x_1
13					x_1	$x_1 + x_2$		$x_1 + x_2$	x_1		x_1	x_2
14					x_1	x_2		x_2	x_1		x_2	x_1
15					$x_1 + x_2$	x_2		x_2	$x_1 + x_2$		$x_1 + x_2$	x_2
16											x_2	$x_1 + x_2$

must be large enough to ensure small error probability.[3]

B. Lossy Network

We now consider that packets may be lost on some links. In this case, we can no longer guarantee meetings of x_1 and x_2 at the joining points and predictable observations at the receivers. There are two differences from the lossless case. First, each experiment might result in different outcomes, shown in Table 2, as a result of probabilistic packet loss. Second, there are common observations across all four types, as opposed to just between types (1) and (4). We divide the observations in Table 2 into three groups: (i) at least one of the receivers does not receive any packet ("-") due to loss, (ii) both receivers have the same observation $R_1 = R_2$, and (iii) the two receivers have different observations $R_1 \neq R_2$.

We choose to ignore the observations of *group (i)* because they can be the result of any of the four 2-by-2 types[4] and thus do not help to distinguish among them in the deterministic way adopted in this paper. In future work, we will also consider the likelihood of these observations for each type. Observations of *group (ii)* can also be the result of any 2-by-2 type: unlike the lossless case, where $R_1 = R_2$ is unique to type (1) or (4) topologies, here any of the four topologies may result in such observations if some packets are lost. However, group (ii) are the only possibility for type (1) topology, apart from the group (i) that we ignore, while all other three 2-by-2 types may result in either $R_1 = R_2$ or $R_1 \neq R_2$. Therefore, if after $countMax$ trials, we have only observed group (ii) packets, then the topology is declared type (1).

In observations of *group (iii)*, it is $R_1 \neq R_2$, which means that $c_{12} \neq c_{22}$ and/or $c_{11} \neq c_{21}$. The difference of these coefficients between the two receivers contains topology-related information. W.l.o.g, we focus on the coefficient of x_2 and look at the difference $c_{12} - c_{22}$. Table 2 shows that $c_{12} - c_{22} < 0$ can only occur in type (2) or type (4) topologies; while $c_{12} - c_{22} > 0$ can only occur in a type (3) or (4) topology. By performing several independent experiments and collecting several observations of group (iii), we can distinguish among the candidate topologies. If after $countMax$ experiments, there are only observations of group (ii) or (iii) with $c_{12} - c_{22} \leq 0$, the topology is declared type (2). If there are only observations of group (ii) or (iii) with $c_{12} - c_{22} \geq 0$, it is declared type (3). If there are observations of group (ii) or (iii) with both $c_{12} - c_{22} < 0$ and $c_{12} - c_{22} > 0$, it is type (4).

In our experiments, we try to create observations that reveal the topology. These can occur either naturally, as the result of packet loss, or artificially, by us introducing an offset u in S_2's sending time with respect to S_1. To help these observations occur, especially for small loss rate, and similarly to the lossless case, we use a random offset $u \in [0, W]$. To make these experiments independent, we space apart successive sets of probes by $T = 3W$. Alg. 4 summarizes the 2-by-2 inference for lossy networks.

In summary, our algorithm is simple and follows a deterministic approach: one observation, or a set of observa-

Algorithm 4 Lossy Case - Inferring a 2-by-2 component. Sources S_1, S_2 multicast x_1, x_2. Receivers observe $R_1 = c_{11}x_1 + c_{12}x_2$ and $R_2 = c_{21}x_1 + c_{22}x_2$. *type* is an indicator of the type that gets updated during the experiments.

```
1: n = 1; /*first experiment*/
2: type=0; /*initialization*/
3: while n ≤ countMax do
4:    if R1 ≠ [0,0] & R2 ≠ [0,0] then
5:       if c22 > c12 then
6:          if type ≠ 3 then
7:             type=2;
8:          else
9:             type=4; Break;
10:         end if
11:      else if c22 < c12 then
12:         if type ≠ 2 then
13:            type=3;
14:         else
15:            type=4; Break;
16:         end if
17:      else if type == 0 & R1 == R2 then
18:         type=1;
19:      end if
20:   end if
21:   n++;
22:   Draw offset u uniformly at random out of [0, W].
23:   Send probes; S2 transmits u time later than S1.
24: end while
25: Output type.
```

[3]However, as discussed in Section VI, $countMax$ is much smaller than the number of experiments required in [2, 3], because at least one observation in Table 1 is sufficient to identify the type, and no large number of probes is needed for statistical significance.

[4]The only exceptions are the two cases $(-, x_1 + 2x_2)$ and $(x_1 + 2x_2, -)$ that uniquely identify type (2) or (3) respectively; however, we cannot generalize such observations to the 2-by-N case.

Algorithm 5 Merging Algorithm: Given the two sources S_1 and S_2, a set of receivers $R_1, R_2, ..., R_N$, the 1-by-N S_1 tree topology, and the 2-by-2 results from Algorithm 4 for any pair of receivers R_i, R_j, this algorithm identifies a single link for the location of every J_i (the joining point for R_i), on S_1 topology.

```
for each receiver R_i do
  if ∃k < i such that the S_1, S_2, R_k, R_i 2-by-2 is shared then
    J_i = J_k;
  else
    Let B be the closest branching point to R_i
    while J_i is not localized to a single link do
      Let R_j be any child of B (j ≠ i)
      Based on the type of the 2-by-2 component S_1, S_2, R_i, R_j, locate J_i above/below B
      if (J_i is below B) || ((J_i is above B) && (∄ other branching point above B on S_1's 1-by-N)) then
        J_i is localized to a single link.
        Output this link; Break;
      else
        B = the next upstream branching point
      end if
    end while
  end if
end for
```

tions, is sufficient to uniquely distinguish among types.[5] As a result, we require much less experiments compared to thousands of arrival order measurements required by [2, 3] for statistical significance. In addition and more importantly, we can identify the exact 2-by-2 type, while [2] can only distinguish between shared and non-shared.[6]

C. Inferring all 2-by-2's in a 2-by-N Network

Algorithms 3 and 4 can be directly applied to a 2-by-N network, *i.e.,* a network where two sources multicast to all N receivers. A difference is that intermediate nodes need to perform addition over a larger finite field.[7] Algorithm 3 and Algorithm 4 can then be performed on any pair of receivers among all $\binom{N}{2}$ possible pairs. The same set of 2-by-N probes is used to infer, in parallel and independently, the type of all 2-by-2 topologies. This reduces the number of probes, as we re-use them, instead of sending $\binom{N}{2}$ different sets of probes. The 2-by-N structure is important for the merging algorithm in the next section.

V. Merging Algorithm

Assuming knowledge of 2-by-2 topologies from Section IV, we now infer the general M-by-N network in two scenarios: (i) given knowledge of a 1-by-N tree topology, which is the same problem studied in [2]; and (ii) without knowledge of any 1-by-N, which is new to our work. Exploiting the accurately identified 2-by-2's, we can solve (i) exactly, which was previously only approximately solved; and also solve (ii), which was previously impossible.

A. Building a 2-by-N given a 1-by-N (and 2-by-2's)

In this section, we assume that the 1-by-N from S_1 to N receivers is known, using either the classic methods in [5], or the approach in [1]. Clearly this 1-by-N graph is a tree rooted at S_1 and contains only branching points. We also assume that the 2-by-2's between S_1, a new source S_2, and any pair of receivers are known, using the algorithms of Section IV. Our goal is to locate the joining points where paths from S_2 to the same N receivers join S_1's topology. We use the assumptions in Section III.

This problem was posed in [2, 4] and solved there in an approximate way. Bounds on the joining points locations in the S_1 topology were provided within a sequence of consecutive logical links. This was a result of the fact that 2-by-2's are only identified as shared or non-shared in [2, 3]. We design Algorithm 5, which localizes each joining point for each receiver to a single logical link, between two branching points, in the S_1 topology. Our algorithm is simpler, faster, and more accurate: it can identify *all* joining points for any topology and with lower complexity, thanks to our complete knowledge of the 2-by-2 types.

Let us discuss the example of Fig. 2(a) taken from [4]. This is an Internet topology connecting hosts at academic institutions in the United States and Europe. Consider R_1: it forms a type (1) 2-by-2 with R_2. Thus J_1 must lie above $B_{1,2}$, so that there exists a unique path from each source to R_1. But then $B_{1,7}$ is on the way. R_1,R_7 form a 2-by-2 of type (4), thus J_1 must be below $B_{1,7}$. Now J_1 is localized to one link and the algorithm ends here for R_1. Other receivers are considered similarly. Since a joining point can be placed on any link from the receiver to S_1, the number of steps our algorithm performs for one receiver is at most the height of the S_1 tree topology. Also, when there is a group of receivers within which all pairs are of type (1), the algorithm is run once and the same joining point is assigned to all of them. The total time is thus upper bounded by (#shared groups)·(height of S_1-tree). This is an improvement over the $O(N^3)$ time in [2]. For the same example, the merging algorithm in [4] cannot completely resolve all joining points and provides bounds within a sequence of several logical links instead.

B. Building a 2-by-N by merging 2-by-2's

In this section, we infer a 2-by-N topology without prior knowledge of any 1-by-N, just by directly sending probes over the 2-by-N and merging all $\binom{N}{2}$ 2-by-2 components. Topology inference under these relaxed assumptions is possible thanks to our complete knowledge of 2-by-2 components, and was not possible before [2, 4].

In the first step, we consider all shared (type (1)) 2-by-2 components and assign them the minimum number of branching and joining points required. For example in Fig.2(a), $B_{1,2}, B_{3,4}, B_{5,6}, B_{8,9}$ and $J_1 = J_2, J_3 = J_4, J_5 = J_6, J_8 = J_9$ are identified in this step. In the second step, we consider all non-shared 2-by-2 topologies (of type (2), (3), or (4)). We use the information about the locations of the branching and joining points in each type to: (1) add the minimum number of branching points required to the ones already identified from the shared pairs; and (2) assign joining points to those receivers that have not been already assigned one. In the example of Fig. 2(a), no additional branching point is required: $B_{8,9}$ is connected to all $J_1 = J_2, J_3 = J_4, J_5 = J_6$, and a new J_7 placed on the path to R_7, to satisfy all the non-shared 2-by-2 types.

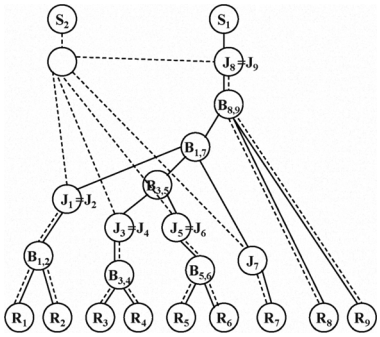
This approach identifies the locations of all joining points, between the S_1 and S_2 1-by-N topologies, but does not identify all the branching points in the S_1 tree topology. Only the "minimum" S_1 topology is identified, *i.e.,* the tree made by the necessary branching points.[8]

[5] *E.g.,* at least one observation of group (iii) rules out the type (1) topology. A pair of group (iii) observations with both $c_{12} - c_{22} > 0$ and $c_{12} - c_{22} < 0$ indicates type (4). Etc.

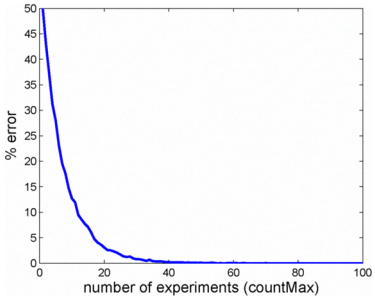
[6] A large W does not guarantee meetings in this case due to the loss effect. Therefore, a large W is not always required in practice.

[7] Additions are performed at joining points. In the worst case, there can be N joining points in a row and thus the field size is the first prime greater than N.

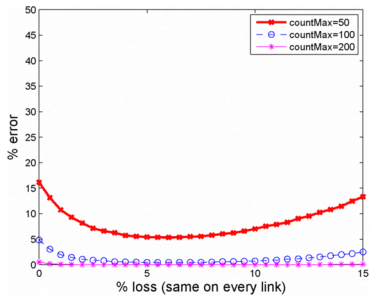
[8] We define as "necessary" branching points the ones located below a joining point of S_1 and S_2 in the 2-by-N structure. An "un-



(a) Example topology from [4]



(b) Lossless case & Alg.3. (Avg over 500 realizations)



(c) Lossy case & Alg.4. (Avg over 10,000 realizations)

Fig. 2: Example logical Internet topology from [4]. Simulation Results for the Lossless (Alg.3) and Lossy (Alg.4) case.

C. From 2-by-N to M-by-N

We can directly extend the 2-by-N inference technique to the M-by-N case [4]: we start from a 2-by-N topology, and add one source at a time, to connect the remaining $M-2$ sources. Assume that we have constructed a k-by-N topology, $2 \leq k < M$. To add the $(k+1)^{th}$ source, we perform k experiments, where at each experiment one different of the k sources and the $(k+1)^{th}$ source send x_1 and x_2. We then glue these topologies together by following the topological rules in Section V-A (with single-source topologies given) or V-B (without such an assumption).

VI. Simulation Results

In this section, we consider the example 2-by-9 topology of Fig. 2(a) and we simulate Algorithms 3 and 4 in the absence and presence of packet loss, respectively. We identify the 2-by-2 types and report the error as a function of the number of experiments $countMax$. We assume that individual link delays vary randomly between 1ms-500ms (a conservative upper bound on the Internet). We choose a large time window $W = 3$sec. Offset u is drawn uniformly at random from $[0, W]$ when needed.

Fig. 2(b) shows the error after applying Algorithm 3 to the lossless network. We infer all $\binom{9}{2}$ 2-by-2 types. Here, the only possible error is to falsely declare a type (4) as type (1). We see that the error probability decreases very rapidly with $countMax$ and reaches 0 at $countMax \simeq 50$.

Fig. 2(c) shows the error assuming that there is packet loss in the network (with prob. p independently on every link) and after applying Algorithm 4. An error in this case can result either from declaring type (2) or (3) or (4) as type (1); or from declaring type (4) as type (2) or (3). We consider values of $p \in [0, 15\%]$ and $countMax = 50, 100, 200$. First, we observe that the error probability is decreasing rapidly with $countMax$: it was negligible with 100 - 200 experiments. This is a significant improvement over [3, 4] for the same example topology: they used 1000 measurements to distinguish only between type (1) and the other three types, for very small loss rates of up to 2%, and they achieved error probability 5-10%. In contrast, with an order of magnitude less probes, we distinguish among all four types, and we have a very small error probability for larger loss rates (up to 15%). Second, we observe that the error probability is not monotonic with p: for small loss rates, Algorithm 4 results in more erroneous cases while Algorithm 3 could give better results. The effect of loss is to increase the number of observations of all three groups. However, for moderate loss rates, we get enough observations of group (iii), thus a small error probability. For larger loss rates, the increase in the observations of group (i), which we ignore, increases the error probability again, especially for small $countMax$.

We note that, in both our approach and in past work [2, 3], the error in the identification of 2-by-2 components may propagate to the Merging algorithm in the next step. However, there is no additional error introduced by our Merging algorithm itself, and thus no need to simulate it.

necessary" branching point is the child of another branching point with no joining point in between. This approach does not identify $B_{3,5}, B_{1,7}$, and directly connects their children ($J_1 = J_2, J_3 = J_4, J_5 = J_6, J_7$) to the upstream branching point ($B_{8,9}$).

VII. Conclusion

In this paper, we designed active probing algorithms that, together with simple network coding at intermediate nodes, can accurately infer a general M-by-N topology. We follow a deterministic approach: we try to create observations that uniquely identify 2-by-2 components; we then merge them to obtain the larger topology. Our approach outperforms traditional multiple source multiple destination tomographic schemes. It can be most beneficial in scenarios where intermediate nodes prefer to support the described operations rather than traceroute.

Acknowledgments

This work was supported by the NSF CAREER grant 0747110.

References

[1] C. Fragouli, A. Markopoulou, S. Diggavi, "Topology Inference using Network Coding," *in Allerton 2006.*

[2] M. Rabbat, M. Coates, R. Nowak, "Multiple Source Internet Tomography," *in IEEE JSAC*, vol. 24(12), pp.2221–2234, 2006.

[3] M. Rabbat, R. Nowak, M. Coates, "Multiple Source, Multiple Destination Network Tomography," *in IEEE INFOCOM 2004.*

[4] M. Coates, M. Rabbat, R. Nowak, "Merging Logical Topologies Using End-to-end Measurements," *in ACM SIGCOMM 2003.*

[5] R. Castro, M. Coates, G. Liang, R. Nowak, B. Yu, "Network tomography: Recent developments," *Statistical Science*, vol. 19, no. 3, pp. 499-517, 2004.

[6] M. Gjoka, C. Fragouli, P. Sattari, A. Markopoulou, "Loss Tomography in General Topologies with Network Coding," *in Proc. IEEE Globecom*, pp. 381–386, November 2007.

[7] T. Ho, B. Leong, Y. Chang, Y. Wen, R. Koetter, "Network Monitoring in Multicast Networks Using Network Coding," *in Proc. of ISIT*, pp. 1977–1981, 2005.

[8] G. Sharma, S. Jaggi, B. K. Dey, "Network Tomography via Network Coding," *in Proc. of ITA Workshop*, UCSD, 2007.

[9] M. Jafarisiavoshani, C. Fragouli, S. Diggavi, C. Gkantsidis, "Bottleneck discovery and overlay management in network coded peer-to-peer systems," *in SIGCOMM INM Wkshp* 2007.

[10] M. Jafarisiavoshani, C. Fragouli, S. Diggavi, "Subspace properties of randomized network coding," *in ITW* 2007.