Multiple Source Multiple Destination Topology Inference using Network Coding

Pegah Sattari
EECS, UC Irvine

Joint work with
Athina Markopoulou, at UCI,
Christina Fragouli, at EPFL, Lausanne
Outline

- Network Tomography
- Goal, Main Ideas, and Contributions
- Proposed Approach
- Conclusion
Network Tomography

- In general
  - Goal: obtain a detailed picture of a network from end-to-end probes.

- Our goal:
  - "Topology inference", multiple sources, multiple receivers, and intermediate nodes both network coding and multicast.
Two bodies of related work

Network Tomography
- Multicast trees using loss correlations
- Unicast probes
- Active probing, reliance on the number, order, delay variance and loss of received probes, and heuristic or statistical signal-processing approach.
- Mostly related: Rabbat, Coates, Nowak, “Multiple-Source Internet Tomography,” IEEE JSAC 06.

Inference with Network Coding
- Passive
  - Failure patterns [Ho et al., ISIT 05]
  - Topology inference [Sharma et al., ITA 07]
  - Bottleneck discovery/overlay management in p2p [Jafarisiavoshani et al., Sigcomm INM 07]
  - Subspace properties [Jafarisiavoshani et al., ITW 07]
- Active
  - Loss tomography [Gjoka et al., IEEE Globecom 07]
  - Binary tree inference [Fragouli et al., Allerton 06]
Main idea 1
Network coding: topology-dependent correlation

[Fragouli et al., 2006], [Sharma et al., 2007]

Network coding introduces topology-dependent correlation among the content of probe packets, which can be reverse-engineered to infer the topology.

- Network coding can make the packets “stay together” and reveal the coding point.
Main idea 2
General Graphs (DAG)

- An M-by-N DAG, with a given routing policy that has three properties:
  - A unique path from each source to each destination.
  - All 1-by-2 components: “inverted Y”.
  - All 2-by-1 components: “Y”.
- Consistent with the routing in the Internet.
- **Logical** topology.

![Diagram](image)

Not a logical topology!
A traditional multiple source, multiple receiver tomography problem can be decomposed into multiple two source, two receiver sub-problems.

Four 2-by-2 types.

- Type 1: shared
- Type 2: non-shared
- Type 3: non-shared
- Type 4: non-shared
Main Idea 2, Cont’d
Decomposition into 2-by-2
Previous Work
2-by-2’s and Merging  Rabbat et al., 2006
Weaknesses of Previous Work

- In the 2-by-2 inference step, they can only distinguish between type 1 (shared) and types 2,3,4 (non-shared).

- This results in inaccurate identification of the joining point locations in the merging step.
  - I.e., bounds within a sequence of several consecutive logical links.
Our Contributions

- At the 2-by-2 inference step:
  - Network coding helps us distinguish among all four 2-by-2 types by looking at the content.

- At the merging step:
  - Under the same assumption as in prior work ($S_1$ 1-by-N), we can localize each joining point, for each receiver, to a single logical link.
  - In addition, we can also design another merging algorithm, without such an assumption, and by only using the 2-by-2 information.
Outline

- Network Tomography
- Goal, Main Ideas, and Contributions
- Proposed Approach
  - Assumptions, Node Operations
  - Step 1: 2-by-2 Components (lossless/lossy)
  - Step 2: Merging Algorithms (two scenarios)
  - Simulation Results
- Conclusion
Assumptions

- **Delay:**
  - fixed part (propagation) and random part (queuing); independent across links.

- **Packet loss:**
  - both lossless and lossy cases.

- **Coarse synchronization (~5-10ms) across nodes.**
  - achievable via a handshaking scheme, *e.g.*, NTP.

- We design active probing schemes, *i.e.*, the operation of sources, intermediate nodes and receivers, which allow topology inference from the observations.
Node Operations

- **Sources**: synchronized
  - later relaxed by large time window $W$
  - in some algorithms, an artificial offset $u$
  - up to countMax experiments, spaced by time $T$.

- **Joining point**: adds and forwards packets within $W$ (additions over $F_q$).

- **Branching point**: forwards the single received packet to all interested links downstream (the next hop for at least one source packet in the network code).
Node Operations

- **Sources:** synchronized
  - later relaxed by large time window $W$
  - in some algorithms, an artificial offset $u$
  - up to countMax experiments, spaced by time $T$.

- **Joining point:**
  - adds and forwards packets within $W$
  - (additions over $F_q$).

- **Branching point:**
  - forwards the single received packet to all interested links downstream
  - (the next hop for at least one source packet in the network code).
Node Operations

- **Sources**: synchronized
  - later relaxed by large time window $W$
  - in some algorithms, an artificial offset $u$
  - up to countMax experiments, spaced by time $T$.

- **Joining point**: adds and forwards packets within $W$ (additions over $F_q$).

- **Branching point**: forwards the single received packet to all interested links downstream (the next hop for at least one source packet in the network code).

\[
x_1 = [1, 0], \quad x_2 = [0, 1]
\]

\[
c_{11}x_1 + c_{12}x_2, \quad c_{21}x_1 + c_{22}x_2
\]
Outline

- Network Tomography
- Goal, Main Ideas, and Contributions
- Proposed Approach
  - Assumptions, Node Operations
  - Step 1: 2-by-2 Components (lossless/lossy)
  - Step 2: Merging Algorithms (two scenarios)
  - Simulation Results
- Conclusion
Inferring 2-by-2’s, No Loss

Distinguishing among \{1,4\}, 2 or 3

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

- One probe distinguishes among Types: \{1,4\}, 2 or 3.
Inferring 2-by-2's, No Loss
Distinguishing between 1,4

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

- Type 1: \( J_1 = J_2 = J \).
- Type 4: \( J_1, J_2 \) different.
- Can be achieved by Appropriately selecting \( u \).

<table>
<thead>
<tr>
<th>Observation Number</th>
<th>Type (1)</th>
<th>Type (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( R_1 )</td>
<td>( R_2 )</td>
</tr>
<tr>
<td>1</td>
<td>( x_1 + x_2 )</td>
<td>( x_1 + x_2 )</td>
</tr>
<tr>
<td>2</td>
<td>( x_1 )</td>
<td>( x_1 )</td>
</tr>
<tr>
<td>3</td>
<td>( x_1 )</td>
<td>( x_1 )</td>
</tr>
<tr>
<td>4</td>
<td>( x_1 )</td>
<td>( x_1 )</td>
</tr>
</tbody>
</table>
Inferring 2-by-2’s, No Loss

Selecting the appropriate offset

Type (4) topology

\[ D_1 > D_2, \text{ offset from } [W-D_1,W-D_2] \]

- 2-by-2’s: \( u \in [W-D_1,W-D_2] \)
- More general: \( u \in [0,W] \)
Inferring 2-by-2’s, Lossy Case

- meetings no longer guaranteed, observations no longer predictable!
- There are common observations across all 4 types.
- Each experiment might result in different outcomes.
Inferring 2-by-2’s, Lossy Case

- There are three groups of observations: (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) | | Type (2) | | Type (3) | | Type (4) |
|---|---|---|---|---|---|---|---|
| 1 | (i) | - | - | (i) | - | - | (i) | - |
| 2 | - | $x_1 + x_2$ | - | $x_1 + 2x_2$ | $x_1 + x_2$ | - | - | $x_1 + x_2$ |
| 3 | - | $x_1$ | - | $x_1 + x_2$ | $x_1 + x_2$ | - | - | $x_1$ |
| 4 | - | $x_2$ | - | $x_1$ | $x_1$ | - | - | $x_2$ |
| 5 | $x_1 + x_2$ | - | - | $x_1 + x_2$ | - | $x_1 + x_2$ | - | $x_1 + x_2$ |
| 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 | $x_2$ | $x_2$ | (ii) | $x_1$ | $x_1$ | (ii) | $x_1$ | $x_1$ | $x_2$ | $x_2$ |
| 11 | | | | | | | | | | |
| 12 | (iii) | $x_1 + x_2$ | $x_1 + 2x_2$ | (iii) | $x_1 + 2x_2$ | $x_1 + x_2$ | (iii) | $x_1 + 2x_2$ | $x_1 + x_2$ |
| 13 | | | | | | | | | | |
| 14 | | | | | | | | | | |
| 15 | | | | | | | | | | |
| 16 | | | | | | | | | | |
### Inferring 2-by-2’s, Lossy Case

Some observations of group (iii) help!

- \( c_{12} - c_{22} < 0 \) can only occur for type 2 or 4!
- \( c_{12} - c_{22} > 0 \) can only occur for type 3 or 4, ...

<table>
<thead>
<tr>
<th>Obs. #</th>
<th>Obs. Group</th>
<th>Type (1) R₁</th>
<th>R₂</th>
<th>Type (2) R₁</th>
<th>R₂</th>
<th>Type (3) R₁</th>
<th>R₂</th>
<th>Type (4) R₁</th>
<th>R₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(i)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>( x₁ + x₂ )</td>
<td>-</td>
<td>( x₁ + 2x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>-</td>
<td>-</td>
<td>( x₁ + x₂ )</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>( x₁ )</td>
<td>-</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>-</td>
<td>-</td>
<td>( x₁ )</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>( x₂ )</td>
<td>-</td>
<td>( x₁ )</td>
<td>( x₁ )</td>
<td>-</td>
<td>-</td>
<td>( x₂ )</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>( x₁ + x₂ )</td>
<td>-</td>
<td>-</td>
<td>( x₂ )</td>
<td>( x₂ )</td>
<td>-</td>
<td>-</td>
<td>( x₁ + x₂ )</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>( x₁ )</td>
<td>-</td>
<td>( x₁ + x₂ )</td>
<td>-</td>
<td>-</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ )</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>( x₂ )</td>
<td>-</td>
<td>( x₁ )</td>
<td>-</td>
<td>-</td>
<td>( x₁ )</td>
<td>( x₂ )</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(ii)</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₂ )</td>
<td>-</td>
<td>-</td>
<td>( x₂ )</td>
<td>(ii) ( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
</tr>
<tr>
<td>9</td>
<td>( x₁ )</td>
<td>( x₁ )</td>
<td>(ii) ( x₁ )</td>
<td>( x₂ )</td>
<td>( x₁ )</td>
<td>( x₂ )</td>
<td>( x₁ )</td>
<td>( x₁ )</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>( x₂ )</td>
<td>( x₂ )</td>
<td>( x₁ )</td>
<td>( x₁ )</td>
<td>( x₂ )</td>
<td>( x₂ )</td>
<td>( x₂ )</td>
<td>( x₂ )</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>( x₂ )</td>
<td>( x₂ )</td>
<td>( x₂ )</td>
<td>( x₁ )</td>
<td>( x₁ )</td>
<td>( x₂ )</td>
<td>( x₂ )</td>
<td>(iii) ( x₂ )</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>(iii) ( x₁ + x₂ )</td>
<td>( x₁ + 2x₂ )</td>
<td>(iii) ( x₁ + 2x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>( x₁ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + 2x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>( x₁ )</td>
<td>( x₂ )</td>
<td>( x₁ + 2x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>( x₁ + x₂ )</td>
<td>( x₂ )</td>
<td>( x₂ )</td>
<td>( x₁ + 2x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>( x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td>( x₁ + x₂ )</td>
<td></td>
</tr>
</tbody>
</table>
Inferring 2-by-2’s, Lossy Case

Try to create group (iii) observations!

- Either naturally (loss) or artificially (u).
- Especially for small loss rates and like the lossless case: u ∈ [0,W]

<table>
<thead>
<tr>
<th>Obs. #</th>
<th>Obs. Group</th>
<th>Type (1)</th>
<th>Obs. Group</th>
<th>Type (2)</th>
<th>Obs. Group</th>
<th>Type (3)</th>
<th>Obs. Group</th>
<th>Type (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(i)</td>
<td>-</td>
<td>(i)</td>
<td>-</td>
<td>(i)</td>
<td>-</td>
<td>(i)</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>x₁ + x₂</td>
<td></td>
<td>x₁ + 2x₂</td>
<td></td>
<td>x₁ + 2x₂</td>
<td></td>
<td>x₁ + x₂</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>x₁</td>
<td></td>
<td>x₁ + x₂</td>
<td></td>
<td>x₁ + x₂</td>
<td></td>
<td>x₁</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>x₂</td>
<td></td>
<td>x₁</td>
<td></td>
<td>x₁</td>
<td></td>
<td>x₂</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>x₁ + x₂</td>
<td></td>
<td>x₂</td>
<td></td>
<td>x₁</td>
<td></td>
<td>x₁ + x₂</td>
</tr>
<tr>
<td>6</td>
<td>x₁</td>
<td></td>
<td>x₁ + x₂</td>
<td></td>
<td></td>
<td>x₁ + x₂</td>
<td>x₁</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>x₂</td>
<td></td>
<td>x₁</td>
<td></td>
<td></td>
<td>x₁</td>
<td>x₂</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(ii)</td>
<td>x₁ + x₂</td>
<td>x₁ + x₂</td>
<td></td>
<td></td>
<td></td>
<td>(ii)</td>
<td>x₁ + x₂</td>
</tr>
<tr>
<td>9</td>
<td>x₁</td>
<td>x₁</td>
<td>x₁</td>
<td>x₁</td>
<td>x₁</td>
<td>x₁</td>
<td>x₁</td>
<td>x₁</td>
</tr>
<tr>
<td>10</td>
<td>x₂</td>
<td>x₂</td>
<td>x₁</td>
<td>x₁</td>
<td>x₁</td>
<td>x₁</td>
<td>x₂</td>
<td>x₂</td>
</tr>
<tr>
<td>11</td>
<td>x₂</td>
<td>x₂</td>
<td>x₂</td>
<td>x₂</td>
<td>x₂</td>
<td>x₂</td>
<td>(iii)</td>
<td>x₁</td>
</tr>
<tr>
<td>12</td>
<td>(iii)</td>
<td>x₁ + x₂</td>
<td>x₁ + 2x₂</td>
<td>(iii)</td>
<td>x₁ + 2x₂</td>
<td>x₁ + x₂</td>
<td></td>
<td>x₁ + x₂</td>
</tr>
<tr>
<td>13</td>
<td>x₁</td>
<td>x₁ + x₂</td>
<td>x₁ + x₂</td>
<td>x₁</td>
<td>x₁</td>
<td>x₁</td>
<td></td>
<td>x₁</td>
</tr>
<tr>
<td>14</td>
<td>x₁</td>
<td>x₁ + x₂</td>
<td>x₁ + x₂</td>
<td>x₁</td>
<td>x₁</td>
<td>x₁</td>
<td></td>
<td>x₂</td>
</tr>
<tr>
<td>15</td>
<td>x₁ + x₂</td>
<td>x₂</td>
<td>x₂</td>
<td>x₁ + x₂</td>
<td>x₂</td>
<td>x₁ + x₂</td>
<td></td>
<td>x₁</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>x₁ + x₂</td>
<td>x₂</td>
<td>x₁ + x₂</td>
<td></td>
<td>x₂</td>
<td>x₁ + x₂</td>
<td>x₂</td>
</tr>
</tbody>
</table>
Inferring all 2-by-2's in a 2-by-N

- Important for the merging algorithm.
- 2 sources multicast to N receivers.
- Additions over a larger field.
- Algorithms can be applied to any pair of receivers among all “N choose 2” possible pairs.
Advantages over Prior Work

- More accurate:
  - we can distinguish among all four 2-by-2 types.

- Faster
  - One observation that uniquely characterizes the 2-by-2 type is sufficient.
  - Unlike [Rabbat et al.], we do not need many experiments for statistical significance.

- Less Bandwidth overhead
  - Duplicate packets crossing the same link.
Outline

- Network Tomography
- **Goal, Main Ideas, and Contributions**
- Proposed Approach
  - Assumptions, Node Operations
  - Step 1: 2-by-2 Components (lossless/lossy)
  - Step 2: Merging Algorithms (two scenarios)
  - Simulation Results
- **Conclusion**
Using the 2-by-2 information, we design two merging algorithms to infer the 2-by-N structure under two scenarios:

1. Assuming knowledge of a 1-by-N tree topology (e.g., using classic tomography methods).
   - We can solve exactly (previously approximately solved).

2. No 1-by-N tree topology is given.
   - We can also solve (previously impossible).

We then generalize our approach to the M-by-N network.
Merging Algorithm 1

1-by-N given

Given: 2-by-2's and $S_1$'s 1-by-N.
Merging Algorithm 2
no 1-by-N given

Only the 2-by-2's are given.
Comparison of the two algorithms

Merging Alg. 1

Merging Alg. 2
From 2-by-N to M-by-N

- 2-by-N can be directly extended to M-by-N.
- Starting from a 2-by-N topology, we add one source at a time, to connect the remaining M-2 sources.
  - Assume we have constructed a k-by-N topology, 2≤k≤M:
    - To add the (k + 1)th source, we perform k experiments:
      - At each experiment one different of the k sources and the (k+1)th source send packets x₁ and x₂.
- We then glue these topologies together by following the topological rules previously described.
Outline

- Network Tomography
- Goal, Main Ideas, and Contributions
- Proposed Approach
  - Assumptions, Node Operations
  - Step 1: 2-by-2 Components (lossless/lossy)
  - Step 2: Merging Algorithms (two scenarios)
  - Simulation Results
- Conclusion
An Internet topology connecting hosts at academic institutions in the US and Europe.
Simulation Results

Absence of loss

- Error: type 4 as type 1.
- Error prob. ~0 in countMax ~ 50.
- Prev. Work: type 1 (shared) vs. \{2,3,4\} (non-shared).

Presence of loss

- Error: types 2,3,4 as type 1 or type 4 as type 2 or 3.
- Error prob. decreases rapidly with countMax.
- Prev. work: 1000 probes (only type 1, \{2,3,4\}), loss ~ 2%, error 5-10%.
Conclusion

- **Summary**
  - Tomographic techniques for topology inference in a network with network coding.

- **Future directions**
  - Likelihood of the observations.
  - Structures larger than 2-by-2:
    - More than two sources and two receivers.
    - Expect a faster merging step at the cost of a more complicated inference step.