Network coding tomography for network failures

Hongyi Yao[1], Sidharth Jaggi[2], Minghua Chen[2]
[1] Tsinghua University [2] Chinese University of Hong Kong

***Abstract*—*Network Tomography* (or *network monitoring*) uses end-to-end path-level measurements to characterize the network, such as topology estimation and failure detection. This work provides the first comprehensive study of passive network tomography in the presence of network failures under the setting that all nodes perform random linear network coding. In particular, we show that it is both necessary and sufficient for all nodes in the network to share *common randomness*, i.e., all local coding coefficients are chosen using a commonly shared random code-book. Without such common randomness, we prove that in the presence of adversarial or random failures, it is either theoretically impossible or computationally intractable to accurately estimate the topology of general networks, and then locate the failures. With such common randomness, we present several sets of algorithms for topology estimation and failure detection, under various settings of adversarial and random failures. For some scenarios our algorithms have polynomial time-complexity, while for others we demonstrate computational intractability. Our main observation from this work is that the linear transforms arising from random linear network coding have some very specific relationships with the network structure, and these relationships can be leveraged to significantly aid tomography. *Key Words:* Network Coding, Network Tomography, Random Errors, Erasures, Adversaries**

I. Introduction

The goal of *network tomography* (or *network monitoring*) is to use end-to-end measurements across a network to infer the network topology, estimate link statistics such as loss rate, and locate network failures [2]. While active network tomography requires a dedicated probing data stream and/or internal node operations, passive network tomography incurs less overhead and complexity as it focuses on revealing network characteristics by passively inferring the information provided by normal data transmission [20]. Most existing work on network tomography are schemes for routing-only networks [2], [20]. By diagnosing the correlated patterns among multiple measurements, such as packets' failure and time delay, a certain amount of information about the internal network can be inferred. The success of these schemes over routing-only networks, however, is limited to simple and structural networks, e.g., trees of moderate size [2].

In essence, the store-and-forward operations internal nodes perform on packets is too simple to allow end-to-end measurements to reveal the network's structure and characteristics to a satisfactory level. The situation becomes even worse in the presence of erasures or errors, or if there are adversaries introducing arbitrary edge erasures and errors.

In the past decade, network coding has attracted much interest since the pioneering work in [1], [19]. By allowing internal nodes to mix incoming packets and output the resulting coded packets, they showed the optimal multicast throughput can be achieved. The optimal multicast throughput is a natural generalization of the well-known *max-flow min-cut theorem* [13], and it can be arbitrarily larger than that achieved by routing based transmission [10]. Moreover, the authors in [19] demonstrated that using *linear coding* can achieve the optimal throughput, and the authors in [6]-[9] showed that even *random linear network coding* (*i.e.*, each node chooses coding coefficients independently and randomly) is sufficient to attain the optimal throughput. Such codes have extremely low design and implementation complexity, in addition to their desirable distributed nature.

Network coding also helps network tomography. With network coding enabled, internal nodes not only store and forward packets, but also mix and code them. Several existing works have shown that such additional coding operations in fact allow end-to-end measurements to reveal more detailed information about the network.

For instance, G.S.Sharma et al. [16] show that the topology of a zero-error network can be determined by an exponentially complex algorithm if random linear network coding is in use and certain global knowledge of coding coefficients is assumed. Similarly, Fragouli et al. [5] and Ho et al. [7] can detect erasure/errors edges using high complexity algorithms under the random linear network coding setting, when the topology and coding coefficients are known *a priori* to the receiver. These works reveal that network coding can help network tomography, at least for some specific problems under specific settings.

Despite such encouraging progress, many important questions remain unexplored. For instance, determining topologies of arbitrary networks under edge failures and adversarial failures, and efficient location of those erroneous.

In this paper we carry out a unified study that provides a comprehensive understanding of passive network tomography in the presence of network failures, under the setting of random linear network coding. In particular, we answer the following two sets of fundamental questions:

- Under what conditions is it ever possible to estimate the topology of general networks in the presence of random or adversarial failures, and to find the locations of such failures.
- Under what conditions do there exist corresponding computationally tractable algorithms?

A. Our Contributions

We first show that it is necessary for all nodes in the network to share *common randomness*, i.e., all local coding coefficients are chosen using a commonly shared random code-book $\mathcal{R}$.

We prove that without such common randomness, it is either theoretically impossible or computationally intractable to accurately estimate the topology of general networks, and then locate the failures in the presence of random or adversarial failures.

Assuming common randomness is shared among all nodes, our results are further categorized into three classes, and are summarized in Table I. A brief disussion of our results follows.

For network tomography in the presence of random errors/erasures:

- we provide the first computationally efficient topology tomogaphy algorithms.
- given that the topology is known, we propose a computationally efficient algorithm that detects all edges experiencing errors/erasures.

Network tomography in the presence of adversarial errors turns out to be a computationally hard problem. For example, we show that the adversary localization problem is computationally intractable (*i.e.*, at least as hard as the nearest codeword problem for random linear coding) even when the receiver already knows the topology and common randomness. For this scenario we

- present existence proof and corresponding (exponential-time) algorithm for topology tomography.
- given the topology is known, we detect all the edges with adversarial errors on the coding subgraph from the source to the receiver via an algorithm whose running-time is $|\mathcal{E}|^z$, where $|\mathcal{E}|$ is the number of edges and z is the number of corrupted edges. This is indeed a high complexity algorithm, but as noted above, the problem is comutaitonally intractable.

We have similar results for network tomography In the presence of adversarial erasures:

- the topology can be estimated via the same scheme as for adversarial errors.
- given that the topology is know, all edges with erasures can be computationally efficiently detected.

We derive our results mainly based on the observation that linear transforms arising from random linear network coding have a very specific relationship with the structure of the network, which can be used for network tomography. In particular, we found it useful to define the *impulse response vector* $\mathbf{t}'(\mathbf{e})$ for every link e as the transform vector from link e to the receiver. The $\mathbf{t}'(\mathbf{e})$ can be treated as the fingerprint of link e. Any failure of e expose its fingerprint, allowing us to detect the failure.

The rest of this paper is organized as follows. We formulate the problem in Section II, and present preliminaries in Section III. We then present our main technical results. As shown in Table I, our results for network tomography in the presence of adversarial errors, random errors, and random/adversarial erasures are presented in Section IV, V, and VI, respectively.

TABLE I
SUMMARY OF OUR RESULTS AND COMPLEXITIES

Failure	Goal	Complexity	Section
Adversarial Errors	Topology estimation	Existence Result	IV-B
	Locating failures	$|\mathcal{E}|^z$	IV-C
Random Errors	Topology estimation	Computationally effcient	V-B
	Locating failures	Computationally effcient	V-C
Random Erasures	Topology estimation	Computationally effcient	VI-A
	Locating failures	Computationally effcient	VI-A
Adversarial Erasures	Topology estimation	Existence Result	VI-B
	Locating failures	Computationally effcient	VI-B

II. Problem Formulation

A. Notational convention

Scalars are in lower-case (*e.g.* z). Matrices are in upper-case (*e.g.* X). Vectors are in lower-case bold-face (*e.g.* $\mathbf{e}$). Column spaces of a matrix are in upper-case bold-face (*e.g.* $\mathbf{E}$). Sets are in upper-case calligraphy (*e.g.* $\mathcal{Z}$).

B. Settings

For ease of discussion, we consider an acyclic and delay-free network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges. In principle our results can be extended to more general networks.

Each node has a *unique identification number* known to itself, such a label could correspond to the node's GPS coordinates, or its IP address, or a factory stamp. The capacity of each edge is normalized to equal one symbol of a finite field $\mathbb{F}_q$ per unit time. Edges with non-unit capacity are modeled as parallel edges. We denote $e(u, v, i)$ as the ith parallel edge between nodes u and v.

For ease of discussion, we focus on the unicast scenario where a single source s communicates with a single receiver r over the network, where intermediate nodes perform random linear network coding as defined below. We remark that our results can be generalized to any communication scenario for which random linear network coding suffice (for instance, multicast communications).

Let C be the *min-cut* from s to r, *i.e.*, the size of a minimal set of edges, the removal of which disconnects s from r. Without loss of generality, we assume that both the number of edges leaving the source s and the number of edges entering the receiver r equal C (the more general case can be handled with somewhat more unwieldy notation), and that there is at least one path from the source s to the receiver r via each interior edge e (if not, no information flows through e, and hence e is irrelevant anyway).

C. Network Transmission on Random linear network Coding

In this paper, we consider the following popular distributed random linear coding scheme called $\mathcal{C}$ [8].

Source encoder: The source s arranges the data into a $C \times n$ *message matrix* X over $\mathbb{F}_q$. It then takes C independent and uniformly random linear combinations over $\mathbb{F}_q$ of the rows of X to generate respectively the packets transmitted on each edge outgoing from s (recall that exactly C edges leave the source s). Each packet contains a pre-determined "short" *header*, known in advance to both the source and the receiver.

Network encoders: Each internal node similarly takes linear combinations of the packets on incoming edges to generate packets transmitted on outgoing edges. Let $\mathbf{x}(u,v,i)$ represent the packet traversing edge $e(u,v,i)$. An internal node v generates its outgoing packet $\mathbf{x}(v,w,j)$ as

$$\mathbf{x}(v,w,j) = \sum_{u:e(u,v,i)\in\mathcal{E}} \beta(u,v,w,i,j)\mathbf{x}(u,v,i). \quad (1)$$

For all nodes u such that there exists at least one edge from u to v, the set of *local coding coefficients* $\{\beta(u,v,w,i,j)\}$ is a vector randomly and uniformly chosen in $\mathbb{F}_q$, and it determines the linear combination of the packet $\{\mathbf{x}(u,v,i)\}$s when generating packet $\mathbf{x}(v,w,j)$. For notational simplicity in places where it causes no confusion, we write $\beta(u,v,w,i,j)$ as $\beta(e,v,e')$ where $e=e(u,v,i)$ and $e'=e'(v,w,j)$.

Receiver decoder: The decoder constructs the $C\times n$ matrix Y over $\mathbb{F}_q$ by treating the received packets as consecutive length-n row vectors of Y (recall that exactly C edges reach the receiver r). The network internal linear operations induce a linear transform between X and Y as

$$Y = TX \quad (2)$$

where T is the overall transform matrix. The receiver can extract T by comparing the received packet headers (recall internal nodes mix headers in the same way as the messages) and the pre-determined headers. With probability at least $1-|\mathcal{E}||\mathcal{V}|/q$ over the choice of the local coding coefficients, T can be shown to be invertible [8]. Thus the receiver can retrieve X from Y by inverting (2).

D. Network failure models

Networks may experience disruption as a part of normal operation. Error and erasure failures are considered. For each of error and erasure failures, both adversarial and random disruptions are considered, resulting in four distinct failure models:

1) **Erasures** An erasure on edge e means that the packet $\mathbf{x}(e)$ carried by e is treated as an all-zeroes length-n vector over $\mathbb{F}_q$ by the node receiving $\mathbf{x}(e)$.
   a) *Random erasures* Random erasures are experienced by edges e independently for each $e\in\mathcal{E}$.
   b) *Adversarial erasures* The set of packets erased is adversarially chosen.
2) **Errors** An error on edge e means that a length-n vector is added to the the packet $\mathbf{x}(e)$ carried by e.
   a) *Random errors* Random errors are experienced by edges e independently for each $e\in\mathcal{E}$. A random error on e means that at least one randomly chosen symbol of the packet carried by e is changed to a uniformly random symbol from $\mathbb{F}_q$. [1]
   b) *Adversarial errors* The set of erroneous packets injected into the network is adversarially chosen, both in terms of location of injection, and values of the packets.

E. Tomography Goals

The focus of this work is network passive tomography in the presence of network failures on networks performing random linear network coding. There are two tomographic goals:

1) *Topology estimation* The receiver r wishes to correctly identify the network topology upstream of it (*i.e.*, the graph $\mathcal{G}$).
2) *Failure location* The receiver r wishes to identify the locations where error or erasure failures occur in the network.

To achieve the first goal (which is the preliminary of the second, unless r pre-knows the network design), it is necessary to assume common randomness among all nodes in the network.

F. Common randomness

Common randomness here means that all local coding coefficients are chosen using a common random code-book $\mathcal{R}$, known *a priori* to the receiver. Each internal node, say v, needs to know only the part of the common randomness in R belnging to v.

The code-book comprises of a list of elements from $\mathbb{F}_q$, with each element chosen uniformly at random. This common random code-book can be securely broadcasted by the source before communication using a common public key signature scheme such as RSA [14], or part of network design.

Common randomness is both necessary and sufficient for network tomography in the presence of failure. On one hand, with common randomness, the work of [16] shows that the topology of zero-error networks can be computed exactly from the transform matrix T. On the other hand, we show in this paper that, without common randomness, determining the topology is either theoretically impossible or computationally intractable.

Depending on the type of failures in the network, we consider two possible types of common randomness. Recall that the local coding coefficient $\beta(u,v,w,i,j)$ transforms information from nodes u to v to w, via the ith and jth parallel edge respectively.

[1] Note the difference of this model from the usual model of *dense random errors* on $\mathbb{F}_q$ [18], wherein each packet is replaced with another packet uniformly at random. The model described in this work is more general in that it can handle such errors as a special case. However, it can *also* handle what we call "sparse" errors, wherein only a small fraction of symbols in a packet get uniformly corrupted. Such a sparse error may be harder to detect. In our model we consider the worst-case sparsity of 1.

1) *Topology estimation with random failures*[2]*:* For each node v, the local coding coefficient $\beta(u, v, w, i, j)$ is the element $\mathcal{R}(u, v, w, i, j)$. Here each distinct setting of the parameters (u, v, w, i, j) indexes a distinct element in $\mathcal{R}$.
2) *Topology estimation with adversarial failures:* For each node v, the local coding coefficient $\beta(u, v, w, i, j)$ is $\mathcal{R}(u, v, \mathcal{A}(v), \mathcal{B}(v), w, i, j)$, where $\mathcal{A}(v)$ and $\mathcal{B}(v)$ are respectively the nodes immediately upstream[3] and downstream of v. Here each distinct setting of the parameters $u, v, \mathcal{A}(v), \mathcal{B}(v), w, i, j$ indexes a distinct element[4] in $\mathcal{R}$.

III. Preliminary

A. Linear transforms in the network

For each edge e there exists a length-C row vector over $\mathbb{F}_q$ called the *global encoding vector* (GEV) $\mathbf{t(e)}$ such that the packet carried by e equals $\mathbf{t(e)}X$. Each $\mathbf{t(e)}$ can be inductively calculated in terms of the linear operations carried out by each node of the network as

$$\mathbf{t(e')} = \sum_{j=1,2,\ldots,d_1} \beta(e_j, v, e')\mathbf{t(e_j)},$$

where $e_1, e_2, ...e_{d_1}$ are the incoming edges of v while e' is an outgoing edges of v. For a set of edges $\mathcal{Z} \subseteq \mathcal{E}$ with cardinality z, the rows of the $z \times C$ *global encoding matrix* $T(\mathcal{Z})$ consists of the vectors $\{\mathbf{t(e)} : e \in \mathcal{Z}\}$. In particular, the rows of the transfer matrix T from the source s to the receiver r are respectively the GEVs of the edges incoming to r.

Corresponding to each edge $e \in \mathcal{E}$ we also define the length-C *impulse response vector* (IRV) $\mathbf{t'(e)}$. In particular, let the source s transmit the *all-zeroes vector* $\mathbf{0} \in (\mathbb{F}_q)^C$ on all outgoing edges, let edge e inject an *all ones packet* $\mathbf{1} \in (\mathbb{F}_q)^n$, and let each internal node perform the encoding operations of code $\mathcal{C}$. Then each column of Y received by the receiver r is identical, and equals the impulse response vector $\mathbf{t'(e)} \in (\mathbb{F}_q)^C$. So $\mathbf{t'(e)}$ can be thought of as a "unit impulse response" from e to r. For a set of edges $\mathcal{Z} \subseteq \mathcal{E}$ with cardinality z, the columns of the $C \times z$ *impulse response matrix* $T'(\mathcal{Z})$ consists of the vectors $\{\mathbf{t'(e)} : e \in \mathcal{Z}\}$.

Note that GEVs are in some sense dual to IRVs – the former represent the linear transforms from the source s to the edge e, while the latter are the linear transforms from the edge e to the receiver r. Just like global encoding vectors, all IRVs

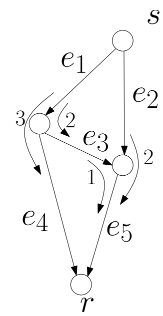

Fig. 1. An example of IRVs of a network. The local encoding coefficients are as shown, and the IRVs are as follows: $\mathbf{t'(e_4)} = [1, 0]$, $\mathbf{t'(e_5)} = [0, 1]$, $\mathbf{t'(e_3)} = \mathbf{t'(e_5)} = [0, 1]$, $\mathbf{t'(e_2)} = 2\mathbf{t'(e_5)} = [0, 2]$ and $\mathbf{t'(e_1)} = 3\mathbf{t'(e_4)} + 2\mathbf{t'(e_3)} = [3, 2]$. Edges e_2 and e_3 are not out-independent, so the IRV $\mathbf{t'(e_2)}$ equals the $\mathbf{t'(e_3)}$ (up to a scalar multiple). Conversely, e_1 and e_5 are out-independent, so $\mathbf{t'(e_1)}$ is linearly independent from $\mathbf{t'(e_5)}$.

can be inductively computed from the incoming edges of the receiver. For example, assume e is an incoming edge of v while $e'_1, e'_2, ..., e'_{d_2}$ are the outgoing edges of v, then we have

$$\mathbf{t'(e)} = \sum_{j=1,2,\ldots,d_2} \beta(e, v, e'_j)\mathbf{t'(e'_j)}.$$

We normalize GEVs and IRVs so that any two vectors that are scalar multiples of each other are said to be equivalent. Thus, unless otherwise specified, the GEV and the IRV of e are both one-dimensional subspaces in $(\mathbb{F}_q)^C$.

Also note that both the GEVs and the IRVs of any edge e are independent of the length of the packet.

B. Relationships between topology and linear transforms

We demonstrate some correspondences between graph properties of the network and linear agebraic properties of random linear network coding.

Any set of z edges $e_1, e_2, ..., e_z$ is said to be *out-independent* if there is a path from the tail of each to the receiver r, and these z are edge-disjoint. They are said to be *in-independent* if the source node has z edge-disjoint paths to the heads of them. The *out-rank* of an edge set $\mathcal{Z}$ equals the max-flow from the tails of $\mathcal{Z}$ to the receiver r. A collection of edge sets $\mathcal{Z}_1, \mathcal{Z}_2, ..., \mathcal{Z}_n$ is said to be *out-independent* if *out-rank*$(\cup_{i=1}^n \mathcal{Z}_i) = \sum_{i=1}^n$ *out-rank*$(\mathcal{Z}_i)$. The *in-rank* of $\mathcal{Z}$ equals the max-flow from the source to the heads of $\mathcal{Z}$. A collection of edge sets $\mathcal{Z}_1, \mathcal{Z}_2, ..., \mathcal{Z}_n$ is said to be *in-independent* if *in-rank*$(\cup_{i=1}^n \mathcal{Z}_i) = \sum_{i=1}^n$ *in-rank*$(\mathcal{Z}_i)$. The in-rank (or out-rank) of an internal node equals to the in-rank (or out-rank) of its incoming (or outgoing) edges. For the set $\mathcal{Z} \subseteq \mathcal{E}$ with in-rank (or out-rank) z, the *in-extended set* (or *out-extended set*) $ExtIn(\mathcal{Z})$ (or $ExtOut(\mathcal{Z})$) is the set that is of in-rank (or out-rank) z, includes $\mathcal{Z}$ and is of maximum size. Note that $ExtIn(\mathcal{Z})$ (or $ExtOut(\mathcal{Z})$) is well-defined and unique[3].

The relationship between the edges and their IRVs and GEVs are:

Lemma 1: 1) The rank of the impulse response matrix $T'(\mathcal{Z})$ of an edge-set $\mathcal{Z}$ with out-rank z is at most z.

[2]In fact, the code-book $\mathcal{R}$ can be replaced by a pseudorandom generator $\mathcal{PRG}$ in the random failure case. The randomness of node v is the output of $\mathcal{PRG}(v)$. It requires each node has an access to $\mathcal{PRG}$, and the source needs to securely broadcast a key of size $\mathcal{O}(|\mathcal{V}|)$. Since all algorithms for random failures have polynomial-time computational complexity in network parameters, codes designed via the PRG must have the same performance as those designed randomly. Otherwise it is possible to design a polynomial-time distinguisher that can break PRG, which is strongly believed to be impossible [11].

[3]If u has k parallel edges to v, u will occur k times in $\mathcal{A}(v)$. The same is true for $\mathcal{B}(v)$.

[4]In the latter of the cases above, the size of $\mathcal{R}$ depends also on the sets $\mathcal{A}(v)$ and $\mathcal{B}(v)$, and hence may be extremely large since each of these sets can take a number of values that grows exponentially with the network size. As justification, we note in Theorem 2 that in fact without common randomness, topology estimation is not possible. Our current techniques require common randomness of this scale, but it is unclear if this is necessary.

2) The rank of the global encoding matrix $T(\mathcal{Z})$ of an edge-set $\mathcal{Z}$ with in-rank z is at most z.
3) The probability that the IRVs of an out-independent set are independent is at least $1 - |\mathcal{E}|/q$.
4) The probability that the GEVs of an in-independent set are independent is at least $1 - |\mathcal{E}|/q$.

Proof:

1) When the out-rank of $\mathcal{Z}$ is z, the max-flow from $\mathcal{Z}$ to r is at most z. If the rank of $T'(\mathcal{Z})$ is larger than z, say $z+1$, $\mathcal{Z}$ can transmit information to r at rate $z+1$, which is a contradiction.
2) The proof is similar to that of 1.
3) For an out-independent set $\mathcal{Z}$ with cardinality z, assume a virtual source node s' has z virtual edges connected to the heads of $\mathcal{Z}$, and the outgoing edges of the heads of $\mathcal{Z}$ are deleted except for $\mathcal{Z}$. The max-flow from s' to r is z and $\mathcal{Z}$ is a cut. Then $T'(\mathcal{Z})$ has rank z if and only if s' can transmit information to r at rate z. But by [8] this happens with probability at least $1 - |\mathcal{E}|/q$.
4) The proof is similar to that of 3.

□

Thus for a large enough field-size q, properties of the edge-sets map to the similar properties of the IRVs and GEVs. For instance, *out-rank*$(\cup_{i=1}^{n}\mathcal{Z}_i) = \sum_{i=1}^{n}$ *out-rank*$(\mathcal{Z}_i)$ if and only if $rank(\cup_{i=1}^{n}T'(\mathcal{Z}_i)) = \sum_{i=1}^{n} rank(T'(\mathcal{Z}_i))$.

The example in Figure 1 shows the relationship of out-independence and IRV-independence.

C. Network error-correcting codes

Consider the scenario where a randomly or maliciously faulty set of edges $\mathcal{Z}$ of size z *injects* faulty packets into the network. As in [17], the network transform (2) then becomes

$$Y = TX + T'(\mathcal{Z})Z, \quad (3)$$
$$= TX + E. \quad (4)$$

Note that Z is a $z \times n$ matrix whose rows are respectively the faulty packets injected by edges in $\mathcal{Z}$, and the $C \times n$ *error matrix* E is defined as $T'(\mathcal{Z})Z$. The goal for the receiver r in the presence of such errors is still to reconstruct the source's message X.

In particular, in this work we use the algorithms of [17].

One desirable feature of the codes in [17] is that they enable the receiver to successfully reconstruct not only the source's message X but also the column-space of the error-matrix $\mathbf{E}$, which is needed for network topology tomography.

D. Computational hardness of NCP

Several theorems we prove regarding the computational intractability of some tomographic problems depend on the following well-studied computational problem. The Nearest Codeword Problem for Random Linear Codes (NCPRLC) is defined as follows:

- *NCPRLC:* $(H, d, \mathbf{z})$
  Given a uniformly random *parity check matrix* $H \in \mathbb{F}_q^{l_1 \times l_2}$ with $l_2 > l_1$, a constant d, and a vector $\mathbf{e} \in \mathbf{H}$, the algorithm is required to output a vector $\mathbf{z} \in F_q^{l_2}$, such that $\mathbf{e} = H\mathbf{z}$ and $\mathbf{z}$ has at most d nonzero elements. Such a vector $\mathbf{z}$ is promised to exist for the given $\mathbf{e}$.

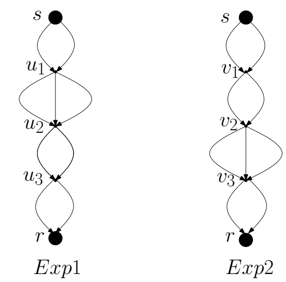

Fig. 2. Two networks that are hard to distinguish between.

The minimum codeword problem is known to be computationally hard even to approximate [4]. In fact, the problem for random parity check matrix H is also believed to be hard [4] ,[21]– no efficient algorithm is known.

IV. Adversarial errors

A. Necessity of common randomness

We first show the necessity of using common randomness for topology inference.

Theorem 2: There exist two networks such that if internal nodes choose local coding coefficients independently and randomly *without* using the common random code-book $\mathcal{R}$, the receiver cannot infer the topology in the absence of errors.

Proof: We are going to show the transform matrixes of $Exp1$ and $Exp2$ in Figure 2 are indistinguishable.

For $Exp1$, $T_s(1) \in \mathbb{F}_q^{2\times 2}$ is the transform matrix from s to u_1, *i.e.*, u_1 receives $T_s(1)X$. $T_{u_1} \in \mathbb{F}_q^{3\times 2}$, $T_{u_2} \in \mathbb{F}_q^{2\times 3}$, and $T_{u_3} \in \mathbb{F}_q^{2\times 2}$ are the transform matrix from u_1, u_2, u_3 to the adjacent downstream nodes respectively. Thus, the transform matrix $T(1)$ from s to r in $Exp1$ is $T(1) = T_{u_3}T_{u_2}T_{u_1}T_s(1)$.

For the similar reason, the transform matrix $T(2)$ from s to r in $Exp2$ is $T(2) = T_{v_3}T_{v_2}T_{v_1}T_s(2)$.

Since each element in T_{u_3}, T_{u_2}, T_{u_1}, $T_s(1)$, T_{v_3}, T_{v_2}, T_{v_1}, $T_s(2)$ is an independently and uniformly random variables, $T(1)$ is not statistically distinguishable from $T(2)$. □

B. Topology inference

In this section, we use an error-correcting code approach [17] to infer the topology of a network containing an adversary. At a high level, the idea is that in sufficiently strongly connected networks, each pair of networks generates transform matrices that look "very different". Hence no matter what the adversary does, he is unable to make the transform matrix for one network resemble that of any other. The proof techniques are similar in flavour to those from algebraic coding theory.

As is common in the network error-correcting literature, we assume that the adversary is bounded, and therefore corrupts no more than z edges in the network. The *rank-distance* between any two matrices $A, B \in \mathbb{F}_q^{C\times C}$ is defined as

$r_m(A, B) = rank(A - B)$. We note that rank-distance indeed satisfies the properties of a distance function; in particular it satisfies the triangle inequality [17].

Assumptions and justifications:

1) *Strong connectivity*. A set of networks satisfies "strong connectivity" if the following is true: each node has both in-rank and out-rank at least $2z + 1$. We motivate this strong connectivity requirement by showing in Theorem 5 lower bounds on the connectivity required for *any* tomography scheme to work in the presence of an adversary.
2) *Knowledge of local topology.* We assume that each node knows the ID numbers of the nodes exactly one hop away from it, either upstream or downstream of it.

Let the transfer matrices of networks $\mathcal{G}$ and $\mathcal{G}'$ be $T(\mathcal{G})$ and $T(\mathcal{G}')$ respectively. We then have the following lemma that shows that the transfer matrices are "very different".

Lemma 3: With probability at least $1 - |\mathcal{V}|^4/q$, $r_m(T(\mathcal{G}), T(\mathcal{G}')) \geq 2z + 1$.

Proof: From r to s, using broad-first searching, we assume u and u' are the first pair of nodes which are different (either the node ID or its adjacent nodes), and belonging to $\mathcal{G}$ and $\mathcal{G}'$, respectively.

We first show the there exists a $(2z + 1) \times (2z + 1)$ a submatrix in $T(\mathcal{G}) - T(\mathcal{G}')$, such that its determinant can be a nonzero value depending on the choices of local coding coefficients.

For the subnetwork below u and u' (recall that it's the same subnetwork in $\mathcal{G}$ and $\mathcal{G}'$.), the coding coefficients are chosen such that only the $2z + 1$ edges disjoint pathes from u (also u') to r transmit the packets using routing. In particular, one edge in a path transmits the exactly packet from the incoming edge of the same path.

Assume for any $i = 1, 2, ..., 2z + 1$, the incoming edge e_i of u is on the i'th disjoint path from s to u. The coding coefficients of u are chosen such that, u sends packet of e_i to the i'th edge-disjoint path to r. In the meantime, the coefficients of u' are all zero, *i.e.*, u' send nothing.

For the subnetwork from s to u in $\mathcal{G}$, the coding coefficients are chosen such that only the $2z+1$ edges disjoint pathes from s to u_1 transmit the packets using routing. In particular, one edge on a path transmits the exactly packet from the incoming edge on the same path.

Then $T(\mathcal{G})$ will contain a $(2z + 1) \times (2z + 1)$ identity submatrix, which of $T(\mathcal{G}')$ is a zero matrix. Thus $T(\mathcal{G})-T(\mathcal{G}')$ has a $(2z + 1) \times (2z + 1)$ sub-matrix with determinant 1.

The determinant of the submatrix is a polynomial with variables being local coding coefficients, it's degree is at most $|\mathcal{E}| \times (2z + 1) \leq |\mathcal{E}|^2 \leq |\mathcal{V}|^4$. Using Schwarth-Zippel lemma [15], with probability at least $1 - \frac{|\mathcal{V}|^4}{q}$, the determinant of the submatrix is nonzero, i.e., $rm(T(\mathcal{G}) - T(\mathcal{G}')) \geq 2z+1$. □

Since there are at most $2^{|\mathcal{V}|^2/2}$ acyclic graphs and $2^{|\mathcal{V}|^2}$ pairs of them, the lemma is true for any pair of networks with a probability at least[5] $1 - |\mathcal{V}|^4 2^{|\mathcal{V}|^2}/q$.

As in (3), after transmission, the erroneous transfer matrix T_e received by r is actually

$$T_e = T + T'(\mathcal{Z})Z(H), \quad (5)$$

where $Z(H)$ represents the errors injected by the adversary in the packet headers, *i.e.*, the first C columns of Z. This combined with Lemma 3 enables us to obtain a decoding rule that estimates the network topology with high probability.

Theorem 4: With probability at least $1 - |\mathcal{V}|^4 2^{|\mathcal{V}|^2}/q$, the network $\mathcal{G}$ is the unique network satisfying

$$\arg \min_{\mathcal{G} \text{ has } |\mathcal{V}| \text{ nodes.}} r_m(T_e, T(\mathcal{G})). \quad (6)$$

Further, this network $\mathcal{G}$ can be identified by the receiver r by using the common randomness available to it. Lastly, all IRVs in the network can be computed using this common randomness.

Proof: We assume lemma 3 is true for any pair of graphs, which happens with a probability at least $1 - |\mathcal{V}|^4 2^{|\mathcal{V}|^2}/q$ as stated above.

By (5), the rank distance $r_m(T_e, T(\mathcal{G}))$ equals $rank(T'(\mathcal{Z})Z(H)) \leq rank(T'(\mathcal{Z}))$, which in turn is at most z. For any transfer matrix $T(\mathcal{G}')$ corresponding to a different network $\mathcal{G}$, by the triangle inequality of the rank distance, $r_m(T(\mathcal{G}'), T_e) \geq r_m(T(\mathcal{G}'), T(\mathcal{G})) - r_m(T(\mathcal{G}), T_e)$. But as shown in Lemma 3, $r_m(T(\mathcal{G}'), T(\mathcal{G})) \geq 2z + 1$ with high probability. Hence $r_m(T(\mathcal{G}'), T_e) \geq z + 1$ with the same probability.

The receiver can find the matrix satisfying (6) by enumerating each possible network configuration $\mathcal{G}$, using the common randomness available to it to estimate the corresponding network transform matrices $T(\mathcal{G})$, and then computing $r_m(T(\mathcal{G}), T_e)$.

Finally, to compute the IRVs of the network, the receiver uses the common randomness available to it to construct the local coding coefficients at each node, and then sequentially construct all IRVs in the network. □

In the end, we show that the strong connectivity requirements we require for Theorem 4 are "almost" tight[6].

Theorem 5: For any network $\mathcal{G}$ that has fewer than $z + 1$ edges from the source s to each node, or fewer than $2z + 1$ edges from each node to the receiver r, there exists an adversarial action that makes any tomographic scheme fail to detect the network structure.

Proof: If node v has a min-cut $2z$ to the the receiver r, and the adversary controls a set $\mathcal{Z}$ of size z of them and runs a fake version of the tomographic protocol announcing that v is not

[5]We do not consider the the network with parallel edges for clarity of exposition. When parallel edges is taken into count, the length of field size should be $\mathcal{O}(|\mathcal{V}|^2 \log(|\mathcal{E}|))$ to make the failure probability of tomography negligible.

[6]There is a mismatch between the sufficient connectivity requirement of Theorem 4 (that there be $2z + 1$ edges between s and each node), and the necessary connectivity requirement of Theorem 5 (that there be $z + 1$ edges between s and each node). The true connectivity requirement for tomographic schemes is still open.

connected to the edges in $\mathcal{Z}$, the probability that r incorrectly infers the presence or absence of v is $1/2$.

On the other hand, if v has only z incoming edges, the adversary can cut these off (*i.e.* simulate erasures on these edges). Since the node can only transmit the message from its incoming edges, this implies that all messages outgoing from u are also, essentially, erased. Hence the presence of u cannot be detected by r. □

In fact, the proof of Lemma 3 only requires $\mathcal{G}$ and $\mathcal{G}'$ differs at a node with high in-rank and out-rank. If we pre-know the possible topology set, we can loose the connectivity requirement. The following corollary shows the idea:

Corollary 6: For a set of possible networks $\{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_d\}$, if any two of them differs at a node which has in-rank and out-rank at least $2z+1$, the receiver can find the correct topology by the receiving transform matrix with a probability at least $1 - d^2|\mathcal{V}|^4/q$.

C. Locating adversarial errors

In this section we demonstrate how to detect the locations in the network where the adversary injects errors.

Assumptions and justifications:

1) *The topology and IRVs of the network are known to the receiver.* This might be because of the scheme in Theorem 4, or perhaps because the network design is known *a priori*.
2) *Every set of $2z$ edges in $\mathcal{G}$ are out-independent.* While this assumption seems strong, we demonstrate in Theorem 8 that such a condition is necessary for r to identify the location of z corrupted edges. Note that this assumption, with high probability, gives a similar statement about the rank of the corresponding IRVs. Using the union bound [12] on the result of Lemma 1.3 gives us the stronger result that any $2\mathcal{Z}$ IRVs are independent with probability at least $1 - |\mathcal{E}|\binom{|\mathcal{E}|}{2z}/q$.
3) *Network error-correcting codes are used to transmit information from s to r.* Network error-correcting codes have provably optimal communication capability in the presence of adversaries. r can then reconstruct both the source's message X and $E = Y - TX$.

Let E has rank η. Let $\{\mathbf{e_1}, \mathbf{e_2}, ..., \mathbf{e}_\eta\}$ be a set of independent columns of E. The receiver r performs the following computation, denoted MIN-INT:

- MIN-INT: Takes as input E. For each column vector $\mathbf{e_i}$ in the basis as defined above, MIN-INT finds a set of edges $\mathcal{Z}_i$ with minimal cardinality such that $\mathbf{e_i}$ is in the column space of the corresponding impulse response matrix $T'(\mathcal{Z}_i)$. MIN-INT outputs each set $\mathcal{Z}_i$.

We show that with high probability MIN-INT finds the location of edges with adversarial errors.

Theorem 7: With probability at least $1 - |\mathcal{E}|\binom{|\mathcal{E}|}{2z}/q$ the solution of MIN-INT results in $\{\mathcal{Z}_i\}$ such that the set of edges $\mathcal{Z}$ on which the adversary injects non-zero errors equals $\cup_{i=1}^{\eta}\mathcal{Z}_i$.

Proof: First of all, since each $\mathbf{e_i}$ is in $\mathbf{T}'(\mathcal{Z})$, we have $|\mathcal{Z}_i| \leq z$ for each $i = 1, 2, ..., \eta$.

We claim that for each $i \in \{1, 2, \ldots, \eta\}$, $\mathcal{Z}_i$ must be a subset of $\mathcal{Z}$. If not, say $e \in \mathcal{Z}'$ is not in $\mathcal{Z}$. By the definition of MIN-INT, $\mathbf{t}'(\mathbf{e})$, the IRV of e, is in the span of the columns of $T'(\mathcal{Z})$ and the IRVs of other edges in $\mathcal{Z}_i$. Thus a non-trivial combination of the at most $2z-1$ IRVs result in $\mathbf{t}'(\mathbf{e})$. However, by Assumption 2 above, any $2z$ IRVs are linearly independent.

We prove next that for any edge $e \in \mathcal{Z}$ on which the adversary injects a non-zero error, MIN-INT outputs at least one $\mathcal{Z}_i$ such that $e \in \mathcal{Z}_i$. Without loss of generality, assume e is the first edge in $\mathcal{Z}$. Then $E = T'(\mathcal{Z})Z$ and the first row of Z is nonzero. Since any z IRVs are independent, $T'(\mathcal{Z})$ is of full column rank. Then for any independent η vectors in $\mathbf{E}$ there must be at least one, say $\mathbf{e}'$, such that the IRV $\mathbf{t}'(\mathbf{e})$ of e has nonzero contribution to it. That is, $\mathbf{e}' = T'(\mathcal{Z})(c_1, c_2, ..., c_z)^T$ with $c_1 \neq 0$. Hence running MIN-INT on $\mathbf{e}'$ will find $\mathcal{Z}_i \subseteq \mathcal{Z}$ and the corresponding edges, including $e \in \mathcal{Z}$. Otherwise, $\mathbf{t}'(\mathbf{e})$ is in the space of $\mathbf{T}'(\mathcal{Z} - \mathbf{e}, \mathcal{Z}_\mathbf{i})$, which is a contradiction. □

We now show matching converses for Theorem 7. In particular, we demonstrate in Theorem 8 that Assumption 2, *i.e.*, that any $2z$ edges be out-independent, is necessary.

Theorem 8: Any z corrupted edges are detectable if and only if any $2z$ IRVs are out-independent.

Proof: The "if" direction is a corollary of Theorem 7. For the "only if" direction, suppose there exist $2z$ edges such that they are not out-independent. Then the corresponding IRVs cannot be linearly independent by Lemma 1.1. Then there must exist a partition of these $2z$ edges into two edge sets $\mathcal{Z}_1$ and $\mathcal{Z}_2$ each of size z such that $\mathbf{T}'(\mathcal{Z}_\mathbf{1}) \cap \mathbf{T}'(\mathcal{Z}_\mathbf{2}) \neq \{0\}$, *i.e.,* the *spanning spaces* of the corresponding IRVs in the two sets intersect non-trivially. Then the adversary can choose to corrupt $\mathcal{Z}_1$ in a manner such that the columns of $T'(\mathcal{Z}_1)Z$ are in $\mathbf{T}'(\mathcal{Z}_\mathbf{2})$. This means r cannot distinguish whether the errors are from $\mathcal{Z}_1$ or $\mathcal{Z}_2$. □

Theorem 8 deals with the case that any z edges can be corrupted. If only some sets of edges are candidates for adversarial action (for instance the set of outgoing edges from some "vulnerable" nodes) we obtain the following corollary.

Corollary 9: Let $\mathcal{S}(\mathcal{Z}) = \{\mathcal{Z}_1, \mathcal{Z}_2, ..., \mathcal{Z}_t\}$ be disjoint sets of edges such that exactly one of them is controlled by an adversary. Then r can detect which edge-set is controlled by the adversary if and only if any two sets $\mathcal{Z}_i$ and $\mathcal{Z}_j$ in $\mathcal{S}(\mathcal{Z})$ are out-independent.

Note: The out-independence between edge sets $\mathcal{Z}_i$ and $\mathcal{Z}_j$ in $\mathcal{S}(\mathcal{Z})$ does not require the edges within each of $\mathcal{Z}_i$ and $\mathcal{Z}_j$ to be out-independent. It merely requires that out-rank$(\mathcal{Z}_i)$ + out-rank$(\mathcal{Z}_j)$ = out-rank$(\mathcal{Z}_i \cup \mathcal{Z}_j)$.

Note that running MIN-INT might require checking all sets of $\binom{\mathcal{E}}{z}$ subsets of edges in the network – this is exponential in z. We now demonstrate that for networks performing random linear coding, the task of locating the set of edges corrupted by an adversary is in fact computationally intractable even when the receiver knows the topology and local encoding coefficients in advance.

Theorem 10: If, knowing the network $\mathcal{G}$ and all encoding

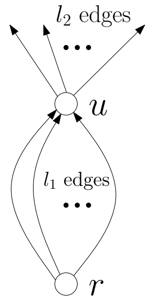

Fig. 3.

operations of $\mathcal{C}$, the receiver r can correctly estimate all adversarial locations in time polynomial in network parameters, NCPRLC can be solved in time polynomial in problem parameters.

Proof: Given a NCPRLC instance $(H, z, \mathbf{e})$, as shown in Figure 3, we construct a network with capacity l_1 and l_2 edges to node u.

Since H is a matrix chosen uniformly at random over $\mathbb{F}_q$, it corresponds to a random network coding $\mathcal{C}$, where each column of H corresponds to an IRV of an incoming edge of u.

Assume the adversary corrupts no more than z incoming edges of u. Adversary can choose the errors Z such that each column of $E = T'(\mathcal{Z})Z$ equals $\mathbf{e}$. But E is all the information about the adversary's behavior known by r when the internal nodes only do random linear network coding. Any algorithm that outputs the corrupted set $\mathcal{Z}'$ must satisfy $\mathbf{e} \in \mathbf{T}'(\mathcal{Z}')$ and $|\mathcal{Z}'| \leq z$. Once $\mathcal{Z}'$ is found, r can efficiently compute Z such that $E = T'(\mathcal{Z}')Z$, which implies that r can solve the NCPRLC instance $(H, z, \mathbf{e})$. $\square$

V. RANDOM ERRORS

A. Necessity of common randomness

We first show that even if each edge suffers random errors independently, topology estimation is at least as computationally intractable as NCPRLC without common randomness.

As in (3), the receiver gets $Y = TX + E$, where $E = T'(\mathcal{Z})Z$. Then $\mathbf{E}$ and T are all the information that can be retrieved by r. Assume the set $\mathcal{I}_{IRV}$ contains all the IRVs of the edges in the network. When the edge suffers random errors independently, $\mathbf{E} \subseteq \mathbf{T}'(\mathcal{Z})$ can not provide more information than $\mathcal{I}_{IRV}$. So it suffices to prove:

Theorem 11: If, knowing T and $\mathcal{I}_{IRV}$, the receiver r can correctly estimate the topology in polynomial time in network parameters, NCPRLC can be solved in time polynomial in problem parameters.

Proof: The proof is similar to that of Theorem 10, and we use the same NCPRLC instance $(H, z, \mathbf{e})$ and network gadget shown in Figure 3. Assume an edge e is connected to z incoming edges of u and has IRV $\mathbf{e}$. If the receiver r can recover the topology, r can tell how e is connected to the z incoming edges of u. Thus r can find a linear combination of z columns of H resulting in $\mathbf{e}$ and thus solve $(H, z, \mathbf{e})$. $\square$

B. Topology inference

We provide a polynomial-time scheme to recover the topology of the network that suffers random network errors. The receiver r proceeds in two steps. Firstly, r recovers the IRVs during several rounds of network communications suffering random errors. In the second step r uses the IRV information obtained to computationally efficiently recover the topology. An interesting feature of the algorithms proposed here is that random network failures actually make it *easier* to efficiently detect the topology. For instance, if no edges in the network ever fail, then the transfer matrix at the receiver is always identical.

Assumptions, justifications, and notation:

1) *Multiple communication rounds.* The protocol runs for t independent communication rounds, where t is a design parameter chosen to trade off between the probability of success and the computational complexity of the topology estimation protocol. The probability of failure of each edge is independent across rounds. Let $\mathcal{Z}(i)$ denote the edge-set that suffers from failures in the ith communication round. The multiplicity of communication rounds enables the variability of IRVs that the receiver leverages to perform topology estimation.
2) *"Weak connectivity requirement".* We assume each edge is *distinguishable* from every other edge, *i.e.*, any pair of edges have out-rank at least 2. (For instance, if each node has out-rank at least 2, then this condition is automatically satisifed.) Using Lemma 1.3 and the union bound over all pairs of edges, any pair of such edges can be shown to have independent nonzero IRVs with a probability of at least $1 - |\mathcal{E}|^3/q$. Lemma 17 shows that if there are two edges e and e' that are not out-independent, then their IRVs are indistinguishable during network communication with random errors.
3) *Each node knows its local topology.* As in Section IV-B, Assumption 2.
4) *Each edge e independently has random errors with probability at least p.* The lower bound p ensures that a moderate value of t is enough such that each edge suffers enough errors to expose its IRV.
5) *The probability $1-p_c$ of independent edge-sets failing is not negligible.* Let $\mathcal{Z}(i)$ and $\mathcal{Z}(j)$ be the edge-sets that experience failures in communication rounds i and j respectively. The probability that $\mathcal{Z}(i)$ is out-independent to $\mathcal{Z}(j)$ (*i.e.*, *out-rank*$(\mathcal{Z}(i), \mathcal{Z}(j)) =$ *out-rank*$(\mathcal{Z}(i)) +$ *out-rank*$(\mathcal{Z}(j))$) is denoted $1 - p_c$. (Note that p_c is independent of i and j due to the assumption of independence of the failures of edges across each communication round.) We require that $1 - p_c$ be bounded away from 0. For example, consider 'noodle' networks (*i.e.*, high-depth h and narrow-width w). For such networks one can show that if the typical size z of $\mathcal{Z}$ is comparable to w then $1 - p_c$ is close to zero. At a high-level, the problem lies in the fact that such networks have high description complexity (dominated by h), but can only

support a low information rate (dominated by w).

6) *Network error-correcting codes* [17] are used. As noted in Section III-C, r can decode source messages X as well as the column space of error matrix $\mathbf{E}$. For $i \in \{1,\ldots,t\}$, $\mathbf{E(i)}$ is the column-space of the error matrix for the ith communication round.

Step I: Find candidate IRVs

The algorithm FIND-IRV that finds a set of candidate IRVs is as follows:

- FIND-IRV: The intersection of the column-spaces $\mathbf{E(i)} \cap \mathbf{E(j)}$ is computed for each pair $i,j \in \{1,\ldots,t\}$. If $rank(\mathbf{E(i)} \cap \mathbf{E(j)}) = 1$ for any (i,j) pair, $\mathbf{E(i)} \cap \mathbf{E(j)}$ is added to the list of IRV candidates.

Recall from the definition (4) of $E = T'(\mathcal{Z})Z$, the column-space $\mathbf{E}$ is a subset of the column-space $\mathbf{T}'(\mathcal{Z})$, and Z represent the injected errors. Lemma 12 shows when we have Z has full row rank z, and then $\mathbf{T}'(\mathcal{Z}) = \mathbf{E}$ is true. Let $(1-z/q)(1-z^2/n)$ be denoted by $1-p_s$.

Lemma 12: For random injected errors Z, Z has full row rank with probability at least $1-p_s$.

Proof: Each failing edge has at least one randomly chosen location in the packet where a random error is inserted. The "Birthday Paradox" [12] implies that with probability at least $1-z^2/n$, for each failing edge-set $\mathcal{Z}$, the following happens: there are z *distinct* locations $l_1,\ldots,l_z \in \{1,\ldots,n\}$ such that l_i has an error in the packet on the i'th edge in $\mathcal{Z}$. On the locations where the injected errors are non-zero, for $i=1,\ldots,z$, Z_{i,l_i} are uniformly non-zero random variables over $\mathbb{F}_q$. Then the determinant of the sub-matrix of the $\{l_1,\ldots,l_z\}$th columns of Z is a nonzero polynomial of degree z of uniformly random variables over $\mathbb{F}_q$. By the Schwartz-Zippel Lemma [15] this determinant is non-zero with probability at least $(1-z/q)$. Thus Z has z independent columns with probability at least $(1-z/q)(1-z^2/n) = 1-p_s$. □

Thus p_s asymptotically approaches 0 with increasing block-length-n and field-size-q.

Theorem 13 characterizes the probability that all IRVs are in the list of candidate IRVs output by FIND-IRV. Let p_a denote $p_c + 2p_s + |\mathcal{E}|/q$.

Theorem 13: The list output by FIND-IRV includes the IRVs for each edge in the network with probability at least $1-|\mathcal{E}|p_a^{tp/2}$.

Proof: We first compute the probability of the event $\mathcal{D}(e,i,j)$ that for any edge e and any $i,j \in \{1,\ldots,t\}$ such that $e \in \mathcal{Z}_i \cap \mathcal{Z}_j$, the IRV $\mathbf{t}'(\mathbf{e})$ equals the one-dimensional vector-space spanned by $\mathbf{E_i} \cap \mathbf{E_j}$.

By Assumption 5, with probability at least $1-p_c$, $\mathcal{Z}_i - e$ is out-independent of $\mathcal{Z}_j - e$. Conditioned on this, Lemma 1.3 implies that with probability at least $1-p_c-|\mathcal{E}|/q$, $\mathbf{T}'(\mathcal{Z}_\mathbf{i}\backslash\mathbf{e})$ is linearly independent of $\mathbf{T}'(\mathcal{Z}_\mathbf{j}\backslash\mathbf{e})$. Hence $\mathbf{T}'(\mathcal{Z}_\mathbf{i}) \cap \mathbf{T}'(\mathcal{Z}_\mathbf{j})$ equals the span of $\mathbf{t}'(\mathbf{e})$. But by Lemma 12, either of $\mathbf{E(i)} \neq \mathbf{T}'(\mathcal{Z}(\mathbf{i}))$ and $\mathbf{E(j)} \neq \mathbf{T}'(\mathcal{Z}(\mathbf{j}))$ with probability at most p_s. Conditioning on all the events implies that the probability of event $\mathcal{D}(e,i,j)$ is at least $1-p_c-2p_s-|\mathcal{E}|/q$.

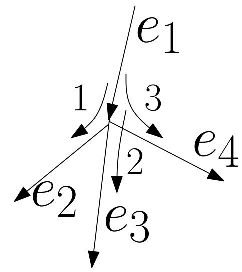

Fig. 4. Let $\mathcal{Z}(1) = \{e_1,e_4\}$ and $\mathcal{Z}(2) = \{e_2,e_3\}$ and $rank(\mathbf{t}'(\mathbf{e_2}),\mathbf{t}'(\mathbf{e_3}),\mathbf{t}'(\mathbf{e_4})) = 3$, then we have $\mathbf{T}'(\mathbf{e_2},\mathbf{e_3}) \cap \mathbf{T}'(\mathbf{e_1},\mathbf{e_4}) = [\mathbf{t}'(\mathbf{e_2}) + \mathbf{2t}'(\mathbf{e_3})]$, which is not an IRV for any edge.

When t is large enough, by the Chernoff bound [12] e will fail at least $tp/2$ times with probability at least $1-p^{\mathcal{O}(t)}$. Conditioned on these many failures, there are $tp/4$ probabilistically independent $\mathcal{D}(e,i,j)$, and FIND-IRV accepts $\mathbf{t}'(\mathbf{e})$ with probability at least $1-(p_a^{tp/4}+p^{\mathcal{O}(t)})$. Taking the union bound over all edges gives the required result. □

Note 1: Since $2p_s + |\mathcal{E}|/q$ is asymptotically negligible for large block-length n and field size q, p_a approximately equals p_c. Also Lemma 1 and Lemma 12 imply that for large n and q, any two failing edge-sets $\mathcal{Z}(i)$ and $\mathcal{Z}(j)$ across multiple communication rounds are out-independent if and only if the corresponding error-matrices $\mathbf{E(i)}$ and $\mathbf{E(j)}$ are linearly independent. Thus r can estimate $1-p_a$ and hence $1-p_c$ by estimating the probability that pairs of $\mathbf{E(i)}$ and $\mathbf{E(j)}$ are linearly independent. This enables r to decide how many communication rounds t are needed so that FIND-IRV has the desired probability of success.

Note 2: If $1-p_a$ is bounded away from zero and $t = \mathcal{O}(\max\{(\log(|\mathcal{E}|)/p, |\mathcal{E}|\})$, the probability that each edge's IRV is accepted is large, say $1-o(1)$. Without loss of generality, we henceforth assume $t = \mathcal{O}(|\mathcal{E}|)$.

Note 3: The set of vectors output by FIND-IRV can also include some "fake" IRVs, as demonstrated in the example in Figure 4. This is not a cause for concern, since finding candidates of IRVs is merely an intermediate step in the process of finding the topology.

Stage II: Topology recovery via candidate IRVs

We now detail the algorithm FIND-TOPO that determines the network topology, using the candidate IRVs generated in Step I by FIND-IRV.

Let $\mathcal{I}_{IRV}$ be the set of candidate IRVs. It is merely a set of one-dimensional subspaces, and as such, individual elements may have no correspondence with the actual IRV of any edge in the network. At any point in FIND-TOPO, let the set $\bar{\mathcal{G}}$ denote the network topology recovered thus far. Let $\bar{\mathcal{V}}$ and $\bar{\mathcal{E}}$ be the corresponding sets of nodes and edges respectively in $\bar{\mathcal{G}}$, and $\bar{\mathcal{I}}_{IRV}$ be the set of IRVs of the edges in $\bar{\mathcal{E}}$, which are computed from $\bar{\mathcal{G}}$ and the random code-book $\mathcal{R}$. We note that the IRVs in $\bar{\mathcal{I}}_{IRV}$ are vectors rather than one-dimensional subspaces, *i.e.*, the unit impulse response vector from the corresponding edge to the receiver (see Section III-A for reference).

- FIND-TOPO: Step A: The set $\bar{\mathcal{V}}$ is initialized as the receiver r, all its immediate upstream neighbours, and the source s. The set $\bar{\mathcal{E}}$ is initialized as the set of edges incoming to r. Hence $\bar{\mathcal{G}} = (\bar{\mathcal{V}}, \bar{\mathcal{E}})$. The initial set of $\bar{\mathcal{I}}_{IRV}$ are the IRVs of the incoming edges of r, initialized to a set of distinct unit vectors.
- Step B: For each node $v \neq s$ in $\bar{\mathcal{V}}$, call function $FindEdge(v, \bar{\mathcal{G}})$ (Step C). Repeat until no new edge is found after going to Step C for each node in $\bar{\mathcal{V}}$; if so go to Step Z.
- Step C: (Function $FindNewEdge(v)$) Let $e_1, \ldots, e_d$ be the outgoing edges of v in $\bar{\mathcal{G}}$. If $\{\bar{\mathbf{t}}'(\mathbf{e_1}), \ldots, \bar{\mathbf{t}}'(\mathbf{e_d})\}$ from $\bar{\mathcal{I}}_{IRV}$ has
  - rank 1, return to Step B.
  - rank greater than 1, call function $CheckIRV(v)$ (Step D).
- Step D: (Function $CheckIRV(v)$) For each candidate $e = (u, v, i)$ not already in $\bar{\mathcal{E}}$ use $\mathcal{R}$ to compute the IRV of e as $\bar{\mathbf{t}}'(\mathbf{e}) = \sum_{j=1}^{d} \beta(e, v, e_j)\bar{\mathbf{t}}'(\mathbf{e_j})$. Check whether $\bar{\mathbf{t}}'(\mathbf{e})$ is in one of $\mathcal{I}_{IRV}$. If so,
  1) Output "New edge found for v".
  2) If $u \notin \bar{\mathcal{V}}$, add u to $\bar{\mathcal{V}}$.
  3) Add $e = e(u, v, i)$ to $\bar{\mathcal{E}}$.
  4) Update $\bar{\mathcal{I}}_{IRV}$ from $\bar{\mathcal{G}}$ and $\mathcal{R}$[7].

  Return to Step D and evaluate for a new $e = (u, v, i)$.
- Step Z: End FIND-TOPO.

We now prove correctness of FIND-TOPO. We begin by proving in Lemma 14 that with high probability FIND-TOPO inserts an edge e in its candidate set of edges $\bar{\mathcal{E}}$ if and only if e is actually in the network $\mathcal{G}$.

Lemma 14: If edge $e = (u, v, i)$ exists in $\mathcal{G}$, $\mathbf{t}'(\mathbf{e})$ is in one of $\mathcal{I}_{IRV}$, $e_1, \ldots, e_d$ are exactly all the outgoing edges of v in $\mathcal{G}$ and $\bar{\mathbf{t}}'(\mathbf{e_i}) = \mathbf{t}'(\mathbf{e_i})$ for $i = 1, 2, ..., d$, FIND-TOPO accepts e as a new edge in $\bar{\mathcal{E}}$ with probability 1. If edge e does not exist in $\mathcal{G}$, FIND-TOPO accepts e as a new edge in $\bar{\mathcal{E}}$ with probability $\mathcal{O}(|\mathcal{E}|^2)/q$.

Proof:

1) If $e = (u, v, i)$ exists in $\mathcal{G}$, $\mathbf{t}'(\mathbf{e})$ is in one of $\mathcal{I}_{IRV}$, $e_1, \ldots, e_d$ are all the outgoing edges of v and have correct IRVs in $\bar{\mathcal{I}}_{IRV}$, we have $\mathbf{t}'(\mathbf{e}) = \sum_{j=1}^{d} \beta(e, v, e_j)\mathbf{t}'(\mathbf{e_j})$ and will be accepted.
2) If e does not exist in $\mathcal{G}$, the coding coefficients $\beta(e, v, e_j)$, $j \in \{1, \ldots, d\}$, are not used. Hence from the perspective of any candidate $\mathbf{v}$ in $\mathcal{I}_{IRV}$, $\sum_{j=1}^{d} \beta(e, v, e_j)\mathbf{t}'(\mathbf{e_j})$ is an independently and uniformly chosen vector in the span of the vectors $\{\mathbf{t}'(\mathbf{e_j}) : \mathbf{j} \in \{\mathbf{1}, \ldots, \mathbf{d}\}\}$. Since $CheckIRV(v)$ is triggered, hence the rank of $\{\mathbf{t}'(\mathbf{e_j}) : \mathbf{j} \in \{\mathbf{1}, \ldots, \mathbf{d}\}\} \geq 2$, so that $\mathbf{t}'(\mathbf{e}) \in \mathbf{v}$ (note that the candidate in $\mathcal{I}_{IRV}$ is one-dimensional subspace) with probability at most $1 - 1/q$. Since FIND-IRV in Stage I needs at most $t = \mathcal{O}(|\mathcal{E}|)$ communication rounds to construct the candidates of IRVs[8], there are at most $\mathcal{O}(|\mathcal{E}|^2)$ candidate IRVs. Using the union bound over all such candidates, $\mathbf{t}'(\mathbf{e}(\mathbf{u}, \mathbf{v}, \mathbf{i}))$ is in one of $\mathcal{I}_{IRV}$ with probability $\mathcal{O}(|\mathcal{E}|^2)/q$. □

Lemma 14 is now an important module in proving the major result in the subsection.

Theorem 15: If FIND-IRV recovers the IRVs of each edge in $\mathcal{I}_{IRV}$, with probability $1 - \mathcal{O}(|\mathcal{E}|^4|\mathcal{V}|)/q$, FIND-TOPO recovers the topology by performing $\mathcal{O}(|\mathcal{E}|^4|\mathcal{V}|)$ operations over $\mathbb{F}_q$.

Proof: Consider FIND-TOPO. Note that if no errors occur, Step B can be find at most $|\mathcal{E}|$, each time of finding a new edge of Step B needs at most $|\mathcal{V}|$ invocations of Step C (once for each node), and each invocation of Step C results in at most 1 invocations of Step D, each invocation of Step D from Step C results in at most $|\mathcal{E}|$ (once for each possible candidate edge) self-invocations . Thus Step D can be invoked at most $|\mathcal{E}|^2|\mathcal{V}|$ times, and Lemma 14 demonstrates that each invocation results in an error with probability $\mathcal{O}(|\mathcal{E}|^2)/q$. Note further that this is the only possible error event. Hence by the union bound [12] the probability that FIND-TOPO results in an erroneous reconstruction of $\mathcal{G}$ is $1 - \mathcal{O}(|\mathcal{E}|^4|\mathcal{V}|)/q$. Also, each computation of Step D takes at most $\mathcal{O}(|\mathcal{E}|^2)$ finite field comparisons to determine membership of $\bar{\mathbf{t}}'(\mathbf{e})$ in the set of candidate IRVs $\mathcal{I}_{IRV}$. Hence, given that the bound on the number of invocations of Step D and that this can be verified to be the most computationally expensive step, the running-time of FIND-TOPO is $O(|\mathcal{E}|^4|\mathcal{V}|)$ operations over $\mathbb{F}_q$.

Finally, we note that $\mathcal{G}$ is acyclic and the assumption that all the IRVs of $\mathcal{G}$ are in $\mathcal{I}_{IRV}$. Hence conditioning on no incorrect edges being accepted, for each invocation of Step B, unless $\bar{\mathcal{G}} = \mathcal{G}$, there exists an edge e such that all edges e' downstream of e in $\mathcal{G}$ are in $\bar{\mathcal{E}}$, which implies all the corresponding $\bar{\mathbf{t}}'(\mathbf{e}')$s are correctly computed. Thus by Lemma 14 edge e is accepted into $\bar{\mathcal{E}}$ by FIND-TOPO with probability 1. Hence, each edge actually in $\mathcal{G}$ also eventually ends up in $\bar{\mathcal{G}}$, and FIND-TOPO terminates. □

C. Locating random errors

We now consider the problem of finding the set of edges $\mathcal{Z}$ that experience random errors. Since $\mathbf{T}'(\mathcal{Z}) = \mathbf{T}'(\mathbf{ExtOut}(\mathcal{Z}))$, the receiver can not distinguish whether the errors are from $\mathcal{Z}$ or $ExtOut(\mathcal{Z})$. So rather than finding $\mathcal{Z}$, we provide a computationally tractable algorithm to locate $ExtOut(\mathcal{Z})$, a proxy for $\mathcal{Z}$.

Assumptions Only Assumptions 1 and 3 (prior knowledge of topology and IRVs, and use of network error-correcting codes) from the corresponding Section IV-B on location of adversarial errors are needed.

The algorithm LOCATE that finds the location of $ExtOut(\mathcal{Z})$ is as follows:

- LOCATE: The input of LOCATE is the error matrix E, and the IRVs of each edge. The receiver checks for each IRV $\mathbf{v}$ whether it lies in $\mathbf{E}$. If so, the edge corresponding to $\mathbf{v}$ is added to $ExtOut$. At the end, $ExtOut$ is output.

[7] The reason that $\bar{\mathcal{I}}_{IRV}$ needs to be updated is that: when e is found as a new edge in $\bar{\mathcal{G}}$, the IRVs of the edges upstream of e will change.

[8] As pointed out in Note 2 after Theorem 13

Theorem 16: If z is no more than $C-1$, LOCATE locates the correct extended set $ExtOut(\mathcal{Z})$ with probability at least $1-3|\mathcal{E}|^2/q-z^2/n$. The computational complexity is $\mathcal{O}(|\mathcal{E}|C^2)$ operations over $\mathbb{F}_q$.

Proof: Lemma 1.3 and 12 implies that $rank(E)$ equals $rank(T'(\mathcal{Z})) = \textit{out-rank}(\mathcal{Z})$, *i.e.*, the number of maximum out-independent edges in $\mathcal{Z}$, with probability at least at least $1-2|\mathcal{E}|/q - z^2/n$. Also, Lemma 1.1 implies that $\mathbf{T}'(\mathbf{ExtOut}(\mathcal{Z})) = \mathbf{T}'(\mathcal{Z}) = \mathbf{E}$. Using the union bound over all $|\mathcal{E}|$ edges on Lemma 1.3, with probability at least $1-|\mathcal{E}|^2/q$, for any edge $e \notin ExtOut(\mathcal{Z})$, $\mathbf{t}'(\mathbf{e})$ is not in $\mathbf{E}$.

For each IRV $\mathbf{t}'(\mathbf{e})$, it cost at most C^2 operations over $\mathbb{F}_q$ to check whether it's in $\mathbf{E}$, then the total complexity of LOCATE is $\mathcal{O}(|\mathcal{E}|C^2)$ operations over $\mathbb{F}_q$. ☐

We next estimate of probability of errors of each edge. To do this, we need to identify which edges are *error identifiable*[9]. That is: an edge $e \in \mathcal{E}$ experiences an error, this event is distinguishable from any edge $e' \neq e$ experiencing an error. Lemma 17 provides necessary and sufficient condition to determine error identifiability.

Lemma 17: If $C > 1$, e is error identifiable if and only if for any different edge e', e and e' are out-independent.

Proof: If each edge $e' \neq e$ satisfies e and e' are out-independent, using Lemma 1.3 and the union bound [12] over each possible $e' \in \mathcal{E}$, with probability at least $1-|\mathcal{E}|^2/q$, IRV $\mathbf{t}'(\mathbf{e}')$ is independent with IRV $\mathbf{t}'(\mathbf{e})$. And then using LOCATE and Theorem 16, we can find e exactly the only edge in the network that is experiencing an error.

In the other direction, if e and e' are not out-independent, $\mathbf{t}'(\mathbf{e}) = c\mathbf{t}'(\mathbf{e}')$ where c is scalar. Thus $\mathbf{t}'(\mathbf{e})Z$ is not distinguishable from $c\mathbf{t}'(\mathbf{e}')Z$. ☐

VI. Erasures

A. Random Erasures

Our techniques can be also used for network erasures (both random and adversarial), resulting in polynomial-time algorithms for tomography.

As in Lemma 12, the event that the injected error matrix $Z \in \mathbb{F}_q^{z\times n}$ has full row rank z is important for this section. Note that an erasure on edge e is equivalent to an error on that edge e, where the error equals the negative of the message flowing on that edge.

Lemma 18: If $\mathcal{Z}$ has in-rank z, with probability at least $1-|\mathcal{E}|/q$, the injected error matrix Z has full row rank z. Otherwise, with probability 1, Z has rank strictly less than z.

Proof: Since the network is directed and acyclic, for ease of analysis we impose an partial order on the edges of $\mathcal{Z} = \{e_1, e_2, ..., e_z\}$. In particular, for any $j > i$, e_j can not be upstream of e_j.

Lemma 1.4 implies that if $\mathcal{Z}$ has in-rank z, with probability at least $1-|\mathcal{E}|/q$, $T(\mathcal{Z})$ has full row rank z.

We now analyze the structure of E. The error corresponding to the erasure on e_1 equal $-\mathbf{t}(\mathbf{e_1})X$. The packet traversing e_2 may be effected by the first erasure. Hence the error corresponding to the erasure on e_2 equal $-(\mathbf{t}(\mathbf{e_2})-a_{1,2}\mathbf{t}(\mathbf{e_1}))X = -\bar{\mathbf{t}}(e_2)X$, where $a_{1,2} = c_{1,2}$ is the *effect from* e_1 *to* e_2. In general, the error corresponding to the erasure on e_i equals

$$\begin{aligned}\bar{\mathbf{t}}(e_i)X &= -(\mathbf{t}(\mathbf{e_i}) - \sum_{j=1,2,\dots,i-1} c_{j,i}\bar{\mathbf{t}}(e_j))X \\ &= -(\mathbf{t}(\mathbf{e_i}) - \sum_{j=1,2,\dots,i-1} a_{j,i}\mathbf{t}(\mathbf{e_j}))X,\end{aligned}$$

where $c_{j,i}$ is the unit effect from e_j to e_i.

Thus $Z = -AT(\mathcal{Z})X$, where $A \in \mathbb{F}_q^{z\times z}$ and the (i,j)'th element of A equal $-a(j,i)$ with $j < i$, 0 if $j > i$, 1 if $i = j$. Then A is invertible. If $T(\mathcal{Z})$ has full row rank z and X has an invertible $C \times C$ sub-matrix (for instance, the head), Z has full row rank z.

Lemma 1.2 implies that if $\mathcal{Z}$ has in-rank less than z, $T(\mathcal{Z})$ has rank less than z. But since $Z = -AT(\mathcal{Z})X$, Z has rank less than z. ☐

When the edges have random erasures, if $rank(Z) = z$ with a non-negligible probability, we can use the same algorithms FIND-IRV and FIND-TOPO as in Section V-B to recover the network's IRVs and then the topology.

In fact, if $rank(Z) = z$, algorithm LOCATE from Section V-C used to find the locations of errors in the network can also be used to find the location of erasures in the network.

B. Adversarial Erasures

Topology inference in the case of adversarial erasures is at least as hard as topology inference in fault-free networks (Section IV-A), and can therefore be handled via the scheme in Section IV-B.

If $rank(Z) = z$, the algorithm LOCATE in Section V-C also works for adversarial erasures.

To efficiently locate the z adversarial erasures $\mathcal{Z}$, Lemma 18 provides a necessary and condition for $rank(Z) = z$, that any set of z edges is in-independent. Assuming such event happens and the adversary chooses z edges to experience erasures, the following theorem characterizes the performance of LOCATE:

Theorem 19: With probability at least $1 - \binom{|\mathcal{E}|}{z}|\mathcal{E}|/q$, no matter what edge set $\mathcal{Z}$ chosen by the adversary to experience erasures, the receiver detects $ExtOut(\mathcal{Z})$ with $\mathcal{O}(|\mathcal{E}|C^2)$ operations over $\mathbb{F}_q$.

Proof Using lemma 18 and the union bound [12] over all possible choices, with probability $1 - \binom{|\mathcal{E}|}{z}|\mathcal{E}|/q$, for any $\mathcal{Z}$ we have $Z = -AT(\mathcal{Z})X$ with full row rank, which implies $\mathbf{E} = \mathbf{T}'(\mathcal{Z})$. Finally, r locates $ExtOut(\mathcal{Z})$ using LOCATE. Theorem 16 guarantees that LOCATE has the desired performance. ☐

If some sets of z edges are *not* in-independent, we cannot guarantee that $rank(Z) = z$. In this case, as in [5], an exhaustive-search (and hence high-complexity) scheme can be used to find the locations of the errors.

[9] This notion is similar to that of *loss identifiability* of [5].

VII. Conclusion and Future Works

This work examines passive network tomography on networks performing random linear network coding, in the presence of network failures. We consider both random and adversarial errors and erasures. We give characterizations of when it is possible to find the topology, and thence the locations of the netork failures, in the presence of such failures. Many of the algorithms we provide have polynomial-time computational complexity in the network size; for those that are not efficient, we prove intractability by showing reductions to computationally hard problems.

Possible future work can proceed in many directions. For one, it would be interesting to characterize how much the connectivity requirements we require can be relaxed if one only wishes to reconstuct the topology approximately. For another, we would like to understand what topological properties can be estimated *without* assuming common randomness. Lastly, we believe that the computational complexity of even some of our polynomial-time schemes can be improved for interesting classes of networks, such as hierarchical networks.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 2000.

[2] R. Castro, M. Coates, G. Liang, R. D. Nowak, and B. Yu. Network tomography: recent developments. *Statistical Science*, 2004.

[3] R. Diestel. *Graph Theory*. Springer-Verlag, Heidelberg, 2005.

[4] I. Dumer, D. Micciancio, and M. Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Transanction on Information Theory*, 49(1):22–37, 2003.

[5] C. Fragouli and A. Markopoulou. A network coding approach to network monitoring. In *Proc. of the 43nd Allerton Conference*, 2005.

[6] T. Ho, R. Koetter, M. Mdard, D. R. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *Proc. of ISIT*, 2003.

[7] T. Ho, B. Leong, Y. H. Chang, Y. G. Wen, and R. Koetter. Network monitoring in multicast networks using network coding. In *Proc. of ISIT*, 2005.

[8] T. Ho, M. Medard, J. Shi, M. Effros, and D. R. Karger. On randomized network coding. In *Proc. of 41st Annual Allerton Conference*, 2003.

[9] S. Jaggi, M. Effros, T. Ho, and M. Médard. On linear network coding. In *Proc. of the 42nd Allerton Conference*, 2004.

[10] S. Jaggi, P. Sanders, PA. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, 51(6):1973–1982, 2005.

[11] Y. Lindell and J. Katz. *Introduction to Modern Cryptography*. Chapman and Hall/CRC press, 2007.

[12] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[13] C. H. Papadimitriou and K. Steiglitz. The max-flow, min-cut theorem. *Combinatorial Optimization: Algorithms and Complexity*, pages 120–128, 1998.

[14] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[15] S. Rudich and A. Wigderson. *Computational complexity theory*. American Mathematical Society, 2000.

[16] G. Sharma, S. Jaggi, and B. K. Dey. Network tomography via network coding. In *Proc. of Information Theory and Applications Workshop*, 2008.

[17] D. Silva, F. R. Kschischang, and R. Kotter. A rank-metric approach to error control in random network coding. *IEEE Transactions on Information Theory*, 54(9):3951–3967, 2008.

[18] D. Silva, F. R. Kschischang, and R. Ktter. Capacity of random network coding under a probabilistic error model,. In *24th Biennial Symposium on Communications, Kingston, ON, Canada*, 2008.

[19] S.R.Li, R. Yeung, and N. Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49(2):371–381, 2003.

[20] Y. Tsang, M. Coates, and R. Nowak. Passive network tomography using em algorithms. In *Proc. of ICASSP*, 2005.

[21] A. Vardy. The intractability of computing the minimum distance of a code. *IEEE Transanction on Information Theory*, 43(6):1757–1766, 1997.