Network RS codes for Efficient Network Adversary Localization

Hongyi Yao*,Sidharth Jaggi†, Minghua Chen†, Tracey Ho*
*Department of Electrical Engineering and Computer Science, California Institute of Technology, USA
†Department of Information Engineering, Chinese University of Hong Kong, Hong Kong

***Abstract*—Network error localizations uses end-to-end observations to localize random or adversarial glitches. For error localization under random linear network codes (RLNCs) the schemes proposed in previous literatures require the *priori* knowledge of the network topology. Moreover, adversarial error localization is proved *computationally intractable* under RLNCs. The goal of current paper is designing new network coding schemes to improve the error localization performance of RLNCs while maintaining the key advantages of RLNCs. To be concrete, we introduce Network Reed-Solomon codes (NRSCs), which have the following features: 1. NRSCs are low-complexity distributed linear network codes. 2. NRSCs achieve the network multicast capacity with high probability. 3. In adversarially faulty networks NRSCs enable the receiver to locate a *maximum* number of adversarial errors in a computationally efficient manner. Moreover, the error localization schemes under NRSCs do not need the priori knowledge of the network topology and thus are robust against dynamic network updating.**

Key Words: Reed-Solomon codes, tomography, error localization, Byzantine adversaries.

I. Introduction

In networks using *linear network coding* each node mixes its receiving packets and outputs their linear combinations, which is proved to attain optimal multicast throughput [1]. In fact, random linear network codes (RLNCs), where each node independently and randomly combines the receiving packets and transmits on outgoing links, suffice to attain the optimal multicast throughput [2], [3].

The goal of *network error localization* is to use *end-to-end measurements* observed by network terminals to locate the errors suffered by the interior nodes in the network [4]. The exploring works [5], [6], [7] showed that receivers in networks performing network coding are able to locate more errors compared with those in networks that only perform routing. Later under RLNCs, the work in [8] provided efficient schemes for locating random errors, and proved that locating the adversarial errors is computationally intractable.

This work was supported in part by National Natural Science Foundation of China Grant 60553001, the National Basic Research Program of China Grant 2007CB807900 and 2007CB807901, NSF grant CNS 0905615, RGC GRF grant 412608, 411008, and 411209, RGC AoE grant on Institute of Network Coding, established under the University Grant Committee of Hong Kong, CUHK MoE-Microsoft Key Laboratory of Human-centric Computing and Interface Technologies, Direct Grant (Project Number 2050397) of The Chinese University of Hong Kong, and two gift grants from Microsoft and Cisco. Part of Hongyi Yao's work was done when he was in Tsinghua University.

More precisely, for directed acyclic networks such that each internal node has at least d outgoing edges, RLNCs are able to determine the identity of up to $d/2$ faulty network edges [8]. The same work shows that this is the best result for network adversarial error localization. Despite such encouraging progress, some challenges remain for RLNCs.

1) Is it ever possible to locate network adversarial errors (*i.e.*, errors that are worst-case in terms of location and content) in a computationally efficient manner? Unfortunately, it is shown in [8] that locating adversarial errors is computationally intractable[1] for RLNCs.
2) Is it ever possible to locate network errors without the prior knowledge of network topology? That is, can the error location algorithm simply outputs some unique identifying characteristic of the error locations (say the IP addresses of the nodes adjacent to the links that experience failure) without knowing in advance the physical connections between nodes in the network? Due to the distributed and random design of RLNCs, all algorithms that we are aware of under RLNC ([6], [8], [5], [7], [10]) is based on the prior knowledge of network topology. However, the topology estimation under RLNC costs exponential time for the networks with adversarial errors [8].

The current paper studies error localization along another direction. To be concrete, we introduce Network Reed Solomon Codes (NRSCs), which are linear network codes that address the two challenges above for network error localization, while preserving the low-complexity, distributed, and high throughput features of RLNC.

A. The contributions of NRSCs

NRSCs address the twin challenges for network error localization faced by RLNCs: the high computational complexity of locating network adversarial errors, and the necessity of prior topology information. To be concrete, NRSC has the following advantages:

- *Low implementation complexity*. The proposed NRSC is a linear network coding scheme (see Section II-C for details), and can be implemented in a *distribute and*

[1]In fact it is shown to be as hard as the well-studied minimum-codeword-problem for random linear error-correcting codes [9].

efficient manner where each network node only needs to know the node-IDs of its adjacent neighbors.

- *High throughput*. The capacity of multicast is achieved with high probability.
- NRSC aids tomography in the following two aspects:
  i) *Computational efficiency*. For the adversarial error model, the receiver under NRSC can locate a *maximum* number of adversarial errors in a computationally efficient manner.
  ii) *The robustness for dynamic networks*. For adversarial error localization, the algorithms under NRSC do not require the priori knowledge of the network topology and thus are robust against edge and node updating.

Besides above theoretical analysis of NRSCs, we further note that NRSCs are flexible with different practical scenarios. More details can be found in Section VI.

The comparison on error localization performance between NRSCs and RLNCs is summarized in Table I. Note that the terms "polynomial" and "exponential" are all in the size of the network.

TABLE I
COMPARISON OF NRSCS AND RLNCS ON ERROR LOCALIZATION PERFORMANCE

Network codes	Field Size required	Computational complexity	Knowledge of network topology
RLNCs	Exponential	Exponential	Required
NRSCs	Polynomial	Polynomial	Not required

The rest of this paper is organized as follows. We formulate the network error localization problem in Section II and present preliminaries in Section III. We then present our main technical results. In Section IV NRSCs are constructed for the multicast scenario and proved to attain optimal throughput. In Section V the error localization scheme based on NRSCs is provided. Section VI generalizes NRSCs to the scenario in which a subset of network edges are trustable.

II. PROBLEM SETTING

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.* $\mathbf{e}$). Sets are in upper-case calligraphic font (*e.g.* $\mathcal{Z}$).

B. Network model

For ease of discussion, we consider a direct 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. For each node $u \in \mathcal{V}$ let $\Gamma_O(u)$ be the set of outgoing edges of u and $\Gamma_I(u)$ be the set of incoming edges of u. The out-degree (or in-degree) of a node u is defined to be $|\Gamma_O(u)|$ (or $|\Gamma_I(u)|$).

The capacity of each edge is normalized to equal one symbol of a finite field of size q, $\mathbb{F}_q$, per unit time. Edges with non-unit capacity is modeled as parallel edges. An edge e with head u and tail v in $\mathcal{V}$ is denoted as $e(u, v)$.

We focus on the multicast scenario where a single source s communicates with a set of receivers $\mathcal{R} \subseteq \mathcal{V}$ over the network. In general our results can be generalized to other scenarios where RLNCs suffice, such as multiple-source multicast network. For ease of notation we assume that the source has no incoming edges and that the receivers have no outgoing edges. Also, that each internal node has at least one incoming edge and one outgoing edge. Otherwise u is isolated and not useful to the communication problem.

C. Network transmission via linear network codes

In this paper we consider the linear network coding scheme proposed in [11]. Let C be the capacity of the multicast network, *i.e.*, $C = \min_{r\in\mathcal{R}} \text{max-flow}(s, r)$. Let each packet have n symbols from $\mathbb{F}_q$, and each edge have the capacity of transmitting one packet, *i.e.*, a row vector in $\mathbb{F}_q^{1\times n}$.

Source encoder: The source s arranges the data into a $C \times n$ *message matrix* X over $\mathbb{F}_q$. Then on each outgoing edge of s a linear combination over $\mathbb{F}_q$ of the rows of X is transmitted. X contains a pre-determined "short" *header*, known in advance to both the source and the receiver, to indicate the linear transform from the source to the receiver.

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

$$\mathbf{x}(e') = \sum_{e\in\Gamma_I(v)} \beta(e, v, e')\mathbf{x}(e), \tag{1}$$

where $\beta(e, v, e')$ is the linear coding coefficient from the packet $\mathbf{x}(e)$ to the packet $\mathbf{x}(e')$ via v.

Receiver decoder: The decoder $r \in \mathcal{R}$ constructs the $d \times n$ matrix Y over $\mathbb{F}_q$ by treating the received packets as consecutive length-n row vectors of Y, where d is the in-degree of r. The network's internal linear operations induce a linear transform between X and Y as

$$Y = TX, \tag{2}$$

where T is the overall transform matrix. The receiver r can extract T by comparing the received packet headers (recall that internal nodes mix headers in the same way as they mix messages) and the pre-determined headers. Once T has rank no less than C the receiver can decode X by $X = T^{-1}Y$.

In the well-known random linear network codes introduced by Ho et al. [3] all linear coding coefficients are chosen uniformly at random and independently by the source and internal nodes. It is shown that in such a code with high probability T has rank no less than C for each receiver.

D. Adversarial models and adversary localization

Networks may experience disruption as a part of normal operation. Edge errors are considered in this work – node errors may be modeled as errors of its outgoing edges. Let $\mathbf{x}(e) \in \mathbb{F}_q^{1\times n}$ be the input packet of e. For each edge $e \in \mathcal{E}$ a length-n row-vector $\mathbf{z}(e)$ is added to $\mathbf{x}(e)$. Thus the output

packet of e is $\mathbf{y}(e) = \mathbf{x}(e)$+$\mathbf{z}(e)$. Edge e is said to suffer an error if and only if $\mathbf{z}(\mathbf{e})$ is a not zero vector.

Let $\mathcal{E}_R$ be the edges incoming to the receivers in $\mathcal{R}$. We define error localization as follows.

Definition 1: The set of edges $\mathcal{E}_R$ is said to be able to locate z *worst-case* errors if all z (or fewer) edges in $\mathcal{E}$ suffering non-zero injected errors can be located from the output packets of $\mathcal{E}_R$.

The error-localization performance of linear network codes was first proved in [8] as follows:

Theorem 2: The set of edges $\mathcal{E}_R$ can locate z errors if and only if each internal node has out-degree at least $d = 2z$.

Remark 1: Note that Theorem 2 is for acyclic networks. The result for networks with cycles is still unknown.

Remark 2: In Definition 1 the adversary is assumed to be able to corrupt any z edges in the network. In Section VI we consider the scenario where a subset of network edges can be trusted.

III. Decoding of Reed-Solomon codes

We begin by recalling some properties of the well-studied Reed-Solomon codes (RSCs) [12], used in particular for worst-case error-correction for point-to-point channels. A Reed Solomon code (RSC) is a linear error-correcting code over a finite field $\mathbb{F}_q$ defined by its parity check matrix $H \in \mathbb{F}_q^{d\times n}$. Here $d+1$ denotes the *minimum Hamming distance*, *i.e.*, the minimum number of nonzero components among the codewords belonging to the code. In particular, H is formed as

$$H = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n], \tag{3}$$

where $\mathbf{h}_i = [h_i, (h_i)^2, ..., (h_i)^d]^T \in \mathbb{F}_q^d$ and $h_i \neq 0$ for each $i \in [1, n]$ and $h_i \neq h_j$ for $i \neq j$.

Given $\mathbf{v}$ which is a linear combination of any $z \leq d/2$ columns of H, the decoding algorithm of RS-CODE, denoted as **RS-DECODE**$(H, \mathbf{v})$, outputs a *z-sparse solution* of $H\mathbf{b} = \mathbf{v}$ with $O(nd)$ operations over $\mathbb{F}_q$ (see [12]). That is, $\mathbf{b} \in \mathbb{F}_q^n$ has at most z non-zero components and $\mathbf{v} = H\mathbf{b}$. Further more, for any $\mathbf{b}' \neq \mathbf{b}$ either $\mathbf{v} \neq H\mathbf{b}'$ or $\mathbf{b}'$ has more than z non-zero components, *i.e.*, $\mathbf{b}$ is the unique z-sparse solution of $H\mathbf{b} = \mathbf{v}$.

IV. The construction of NRSCs

A. Node and edge IDs

Each pair of nodes (u, v) in $\mathcal{V} \otimes \mathcal{V}$ has an ID $id(u, v)$ chosen independently and uniformly at random from $\mathbb{F}_q$. These IDs can be broadcast by the source using digital signature schemes such as RSA [13], or outputted by a pseudorandom hash function[2] (with input as a pair of nodes) such as AES that can be accessed by all parties. Thus this set of $|\mathcal{V}|^2$ IDs is publicly known *a priori* to all parties, even though they may not know *which* nodes and edges are actually in the network.

[2]Note that the randomness of the IDs is used in proving Lemma 3 and Theorem 4, which (the distinctness of node-pair IDs and the throughput of multicast) are polynomial time distinguishable. Thus pseudorandomness suffices [13].

The following lemma shows that each node pair has a distinct ID with high probability:

Lemma 3: With probability at least $1 - |\mathcal{V}|^4/q$, for any $(u, v) \neq (u', v')$ in $\mathcal{E}$, $id(u, v) \neq id(u', v')$.

Proof: For any $(u, v) \neq (u', v')$, $id(u, v) \neq id(u', v')$ with probability at most $1/q$. Since $\mathcal{V}\times\mathcal{V}$ has size $|\mathcal{V}|^2$, there are at most $\binom{|\mathcal{V}|^2}{2} < |\mathcal{V}|^4$ distinct pairs in $\mathcal{V}\times\mathcal{V}$. Using Union Bound over all these $|\mathcal{V}|^4$ pairs the lemma is true with probability at least $1 - |\mathcal{V}|^4/q$. □

For each edge $e(u, v) \in \mathcal{E}$ the ID of e is $id(e) = id(u, v)$. Thus the ID of edge $e(u, v)$ can be figured out by both u and v if they know their adjacent neighbors. Thus a direct corollary of Lemma 3 is that each edge has a distinct ID with high probability. We henceforth assume that this is indeed the case.

Note that for graphs with parallel edges, each pair of nodes has multiple IDs, the ith of which is for the ith parallel edge.

For each edge e the *virtual impulse response vector* (VIRV) is $\mathbf{t}''(e, i) \in \mathbb{F}_q^i$, which is $[id(e), (id(e))^2, ..., (id(e))^i]^T$. For any set of edges $\mathcal{Z}$ with size z, the virtual impulse-response-matrix (VIRM) is $T''(\mathcal{Z}, i) \in \mathbb{F}_q^{i\times z}$, with the columns comprised of $\{\mathbf{t}''(e, i), e \in \mathcal{Z}\}$. Note that $T''(\mathcal{Z}, z)$ is a Vandermonde matrix and invertible when the edges in $\mathcal{Z}$ have distinct IDs.

B. Code construction of NRSCs

We assume by default that the edges in $\mathcal{E}$ have distinct IDs, which happens with probability at least $1 - |\mathcal{V}|^4/q$ by Lemma 3. Let C be the capacity of the multicast network, *i.e.*, $C = \min_{r\in\mathcal{R}}$ max-flow(s, r).

The construction of NRSCs is then as follows.

Source encoder: Let $\Gamma_O(s) = \{e_1, e_2, ..., e_p\}$ be the outgoing edges of the source s and $X \in \mathbb{F}_q^{C\times n}$ be the source message matrix. The source s computes $M = T''(\Gamma_O(s), p)^{-1}X$ and sends the ith row of M as the packet over e_i. Similar to RLNCs [3], the matrix X contains a known "header" to indicate the network transform to the receiver.

Network encoders: Let $\Gamma_O(v) = \{e_1, e_2, ..., e_d\}$ be the outgoing edges of node v. For an incoming edge e of v, v computes $\mathbf{b}(e) = T''(\Gamma_O(v), d)^{-1}\mathbf{t}''(e, d)$. For the coding coefficient $\beta(e, v, e_i)$ from e via v to e_i, v sets $\beta(e, v, e_i)$ to be the ith component of $\mathbf{b}(e)$.

Receiver decoder: The receiver receives

$$Y = TX, \tag{4}$$

where $T \in \mathbb{F}_q^{C\times C}$ can be indicated by the header of Y. If T is invertible the receiver can decode X correctly.

Thus, just like random linear network codes [3], NRSCs can be implemented in a distributed manner once each node knows its local topology, *i.e.*, the adjacent neighbors. If an edge/node has been added/deleted, only local adjustments are needed.

C. Optimal throughput for multicast scenario

Theorem 4 below shows that with high probability NRSCs achieve the multicast capacity.

Theorem 4: With probability at least $1-C|\mathcal{E}|^4|\mathcal{R}|/q$, each receiver in $\mathcal{R}$ can decode X correctly.

Proof: We first prove that for any receiver $r \in \mathcal{R}$, receiver r can correctly decode X with probability at least $1-C|\mathcal{E}|^4/q$.

Let $\mathcal{X}$ be the set of all random variables involved, *i.e.*, $\mathcal{X} = \{id(u,v), (u,v) \in \mathcal{V} \otimes \mathcal{V}\}$. By default we assume that any polynomial mentioned in the proof has variables in $\mathcal{X}$.

Let $det_G = \Pi_{u\in\mathcal{V}} det(u)$, where $det(u)$ is the determinant of the matrix $T''(\Gamma_O(u), |\Gamma_O(u)|)$ for node $u \in \mathcal{V}$. For each $u \in \mathcal{V}$, since each component of $T''(\Gamma_O(u), |\Gamma_O(u)|)$ is a polynomial of degree at most $|\Gamma_O(u)|$, $det(u)$ is a polynomial of degree at most $|\Gamma_O(u)|^2$. Thus det_G is a polynomial of degree at most $\sum_{u\in\mathcal{V}} |\Gamma_O(u)|^2 \leq (\sum_{u\in\mathcal{V}} |\Gamma_O(u)|)^2 = |\mathcal{E}|^2$.

Let T be the transform matrix from s to r defined in Equation (4). We claim each element of $det_G T$ is a polynomial of degree at most $|\mathcal{E}|^4$. To see this, we first note that each component in $det(u)T''(\Gamma_O(u), |\Gamma_O(u)|)^{-1}$ is a polynomial of degree at most $|\Gamma_O(u)|^2 - |\Gamma_O(u)|$ (see Cramer's rule in [14]). Thus in the construction of NRSCs each local coding coefficient $\beta(e,u,e')$ used by $u \in \mathcal{V}$ is $Poly_{(e,u,e')}/det(u)$, where $Poly_{(e,u,e')}$ is a polynomial of degree at most $|\Gamma_O(u)|^2$. Each element in T can be expressed as $\sum_\alpha \bar{\beta}(\alpha)$, where $\bar{\beta}(\alpha) = \Pi_{(e,u,e')\in\alpha}\beta(e,u,e')$ and α is a path from s to r (see [3] for references). Thus each element in T can be expressed as $Poly_\alpha/(\Pi_{u\in\alpha} det(u))$, where $Poly_\alpha = \Pi_{(e,u,e')\in\alpha} Poly_{(e,u,e')}$. Thus $Poly_\alpha$ is a polynomial of degree at most $\sum_{u\in\alpha} |\Gamma_O(u)|^2 \leq \sum_{u\in\mathcal{V}} |\Gamma_O(u)|^2 \leq |\mathcal{E}|^2$. Since no node appears twice in a path of an acyclic network, det_G is divisible by $\Pi_{u\in\alpha} det(u)$ for each path α. Thus $det_G \sum_\alpha Poly_\alpha(\mathcal{X})/(\Pi_{u\in\alpha} det(u))$ is a polynomial of degree at most $|\mathcal{E}|^4$. This completes the proof of the claim that each element of $det_G T$ is a polynomial of degree at most $|\mathcal{E}|^4$.

Now we prove $det_G T$ is invertible with high probability. The determinant of $det_G T$ is denoted as det_r, which is therefore a polynomial of degree at most $|\mathcal{E}|^4 C$.

Without loss of generality let $\{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_C\}$ be the edge-disjoint paths from the source s to the receiver r. We first prove that det_r is a nonzero polynomial, *i.e.*, that there exists an evaluation of $\mathcal{X}$ such that $det_G \neq 0$ (*i.e.*, the edges in $\Gamma_O(u)$ have distinct IDs for each $u \in \mathcal{V}$) and the source can transmit C linearly independent packets via $\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_C$.

The evaluation of $\mathcal{X}$ is described as follows: First, assume each edge has a distinct ID. Second, since the ith outgoing edge of the source sends the ith row of $M = T''(\Gamma_O(s), C)^{-1}X$, the paths $\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_C$ carry linearly independent packets on their initial edges. Third, the IDs of edges in $\mathcal{P}_i$ are all changed to be the ID of the first edge in $\mathcal{P}_i$. Note that this operation preserves the property that the edges in $\Gamma_O(u)$ have distinct IDs for each $u \in \mathcal{V}$ (*i.e.*, $det_G \neq 0$). Finally in fact the network uses routing to transmit the C independent source packets via $\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_C$.

Thus under the above evaluation of $\mathcal{X}$ the matrix $det_G T$ is invertible and therefore $det_r \neq 0$. Using Schwartz-Zippel lemma [3] $det_r \neq 0$ with probability at least $1 - |\mathcal{E}|^4 C/q$ over the choices of $\mathcal{X}$. In the end using Union Bound over all receivers, with probability at least $1 - |\mathcal{E}|^4|\mathcal{R}|C/q$ each receiver can recover the source message X. □

Therefor the techniques over random linear network codes in multicast scenario can be directly moved into NRSC. For instance using network error-correcting codes [15], [16] NRSC are able to attain the optimal throughput for multicast with network errors.

V. Locating Adversarial errors under NRSCs

In this section the error-locating model defined in Section II-D is assumed. The NRSCs described in Section IV are used for network communications. For networks satisfying $|\Gamma_O(u)| \geq d$ for each node $u \in \mathcal{V} - \mathcal{R}$, to simplify notation we use $\mathbf{t}''(e)$ for each VIRV $\mathbf{t}''(e,d)$ in the following. Recall that $\mathcal{E}_R$ is the set of incoming edges of $\mathcal{R}$ and $\mathbf{y}(e)$ is the output packet on any edge $e \in \mathcal{E}$. We define:

Definition 5: The *Reed-Solomon matrix of* $\mathcal{E}_R$ is

$$Y_R = \sum_{e\in\mathcal{E}_R} \mathbf{t}''(e)\mathbf{y}(e). \tag{5}$$

Assumptions and Justifications

1) At most z edges in $\mathcal{Z}$ suffer errors, *i.e.*, $\{e : e \in \mathcal{E}, \mathbf{z}(e) \neq 0\} = \mathcal{Z}$ and $|\mathcal{Z}| \leq z$. Recall that $\mathbf{z}(e)$ is the error packet injected on edge e. When $2z+1 \leq C$, network error-correcting-codes (ECC) (see [15], [16] for details) are used so that the source message X is provably decodable .
2) Each node in $\mathcal{V}-\mathcal{R}$ has out-degree at least $d = 2z$. Note that such connectivity requirement is shown necessary in Theorem 2.
3) The elements in $\mathcal{V} \otimes \mathcal{V}$ are indexed by $\{1, 2, ..., |\mathcal{V}|^2\}$. The parity check matrix $H \in \mathbb{F}_q^{d\times|\mathcal{V}|^2}$ is defined as $H = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_{|\mathcal{V}|^2}]$. Here $\mathbf{h}_i = [h_i, (h_i)^2, ..., (h_i)^d]^T$ and h_i is the ID for the ith element in $\mathcal{V} \otimes \mathcal{V}$.

The error locating algorithm is:

- **LOCATE**: The *input* of the algorithm is the source matrix X, the parity-check matrix H, the source message X (which is decoded by network ECC [15], [16]) and the output packets of $\mathcal{E}_R$, *i.e.*, $\{\mathbf{y}(e) : e \in \mathcal{E}_R\}$. The *output* of the algorithm is a set of edges $\mathcal{Z}'$ (initialized as an empty set).
- Step A: Compute Y_R by Equation (5) and $L = Y_R - (X)_d$, where $(X)_d$ comprises of the first d rows of X.
- Step B: For each column of L, say $\mathbf{v}$, compute $\mathbf{b} =$ **RS-DECODE**$(H, \mathbf{v})$. If the ith component of $\mathbf{b}$ is nonzero, the ith node pair (u,v) in $\mathcal{V}\otimes\mathcal{V}$ is added as an edge $e = (u,v)$ into $\mathcal{Z}'$.
- Step C: End **LOCATE**.

We state the main theorem in the following.

Theorem 6: The edge set $\mathcal{Z}'$ output by **LOCATE** equals $\mathcal{Z}$. The computational complexity of **LOCATE** is $\mathcal{O}(n|\mathcal{V}|^2 d)$.

Before the proof we show the following lemma:

Lemma 7: If the source message matrix X equals 0,

$$Y_R = \sum_{e\in\mathcal{E}} \mathbf{t}''(e)\mathbf{z}(e). \tag{6}$$

Proof: We proceed inductively. Throughout the proof let $\mathcal{E}_T$ be the set of edges satisfying the theorem, *i.e.*, $Y_R = \sum_{e\in\mathcal{E}} \mathbf{t}''(e)\mathbf{z}(e)$ when $\mathbf{z}(e) = 0$ for all $e \in \mathcal{E} - \mathcal{E}_T$.

Step A: If $\mathcal{E}_T = \mathcal{E}_R$, the theorem is true by the definition.

Step B: Since the network is acyclic, unless $\mathcal{E}_T = \mathcal{E}$, there must exist an edge $e \in \mathcal{E} - \mathcal{E}_T$ with an outgoing edge set $\Gamma_O(e) \subseteq \mathcal{E}_T$. Let $\Gamma_O(e) = \{e_1, e_2, ..., e_k\}$ with $k \geq d$ be the set of all outgoing edges of the tail node of e. If only e suffers non-zero injected errors $\mathbf{z}(e)$, the output of e is $\mathbf{z}(e)$. Thus for each $i \in [1, k]$ the output of e_i is $\beta_i\mathbf{z}(e)$, where β_i is the ith component of $\mathbf{b}(e) = T''(\Gamma_O(e), k)^{-1}\mathbf{t}''(e, k)$ (see Section IV-B). Thus $\sum_{i\in[1,k]} \beta_i\mathbf{t}''(e_i, k) = \mathbf{t}''(e, k)$. Since NRSCs are linear network codes and $d \leq k$, $Y_R = \sum_{i\in[1,k]} \beta_i\mathbf{t}''(e_i)\mathbf{z}(e) = \mathbf{t}''(e)\mathbf{z}(e)$. Therefore Equation (6) is true for the case where only e suffers non-zero injected error $\mathbf{z}(e)$. Also, since NRSCs are linear codes, e can be added into $\mathcal{E}_T$.

Step C: Since the network is acyclic and each node (or edge) in $\mathcal{V}$ (or $\mathcal{E}$) is connected to $\mathcal{R}$, we can repeat Step B until $\mathcal{E}_T = \mathcal{E}$. ∎

For the case where no error happens in the network and the source s transmits the $C \times n$ message matrix X with $C \geq d$, by Theorem 7 above we have $Y_R = \sum_{i\in[1,C]} \mathbf{t}''(e_i)\mathbf{x}(e_i)$, where $\mathbf{x}(e_i)$ is the ith row of $M = T''(\Gamma_O(s), C)^{-1}X$, *i.e.*, the packet transmitted on the ith outgoing edge of the source s (see Section IV-B). Thus $Y_R = T''(\Gamma_O(s), d)M = (X)_d$, where $(X)_d$ is the matrix consisting of the first d rows of X.

Then we have the corollary:

Corollary 8: When the source message is X, $Y_R = (X)_d + \sum_{e\in\mathcal{E}} \mathbf{t}''(e)\mathbf{z}(e)$.

Then we can prove Theorem 6 as:

Proof of Theorem 6: Using Corollary 8 we have $L = \sum_{e\in\mathcal{Z}} \mathbf{t}''(e)\mathbf{z}(e)$. Since $|\mathcal{Z}| = z \leq d/2$, each column of L is a linear combination of at most $d/2$ columns of H. Additionally, since H is also a parity check matrix of a Reed-Solomon code, **RS-DECODE** correctly finds all the edges with nonzero injected errors, and therefore $\mathcal{Z}' = \mathcal{Z}$. For each column of L, **RS-DECODE** runs in time $\mathcal{O}(|\mathcal{V}|^2 d)$. Thus the overall time complexity of the algorithm is $\mathcal{O}(n|\mathcal{V}|^2 d)$. □

VI. Generalization of NRSCs

In the following, assume $\mathcal{E}' \subseteq \mathcal{E}$ to be the set of edges that are candidates for adversarial corrupting, and $\mathcal{E} - \mathcal{E}'$ to be the set of trustable edges, and no more than z edges in $\mathcal{E}'$ are corrupted. In such scenario, directly using the NRSCs constructed above requires that any node in the network has max-flow at least $2z$ to the receivers. To release such high connectivity requirement, we generalize NRSCs as follows.

• For each receiver, say R, the IRVs of its incoming edges $\Gamma_I(R)$ are set to be the desired VIRVs.

• For each node, say V, if the outgoing edges $\Gamma_O(V)$ of V have rank at least $2z$, NRSC is used by V such that the IRVs of edges in $\Gamma_I(V)$ equals the desired VIRVs. Otherwise, node V performs random linear network coding (RLNC) to choose the coding coefficients. Node V then informs the IRV of each edge e in $\Gamma_I(V)$ to the upstreaming node of e.

• The receivers perform the same algorithm **LOCATE** in Section V to locate adversarial edges in $\mathcal{E}'$.

Using Theorem 4 of this paper and the capacity achievable results for RLNC [3], we conclude for any node (say V) with max-flow at least $2z$ to the receivers, with high probability the IRVs of edges in $\Gamma_O(V)$ have rank at least $2z$. Thus the IRVs of edges in $\Gamma_I(V)$ all equal the desired VIRVs.

Thus, to locate z adversarial edges in $\mathcal{E}'$, only the nodes in $\{V : \Gamma_I(V) \cap \mathcal{E}' \neq \emptyset\}$ need to meet the connectivity requirement. We note that this condition well fits the practical network scenarios. In practical networks (e.g. Internet), local area networks are more trustable than the public networks, which on the other hand have better connectivity conditions.

VII. Conclusion and Future Work

The paper designs network Reed-Solomon codes (NRSCs) addressing the negative tomography results of adversary localization arising at RLNCs, and meanwhile preserving the advantages of RLNCs. In fact, instead of Reed-Solomon codes we can implant other traditional linear error-correction-codes (*e.g.* BCH codes) into network coding to achieve the same goals. Thus more benefits are hoped to be explored from such network coding structure.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[2] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. of Allerton*, 2003.

[3] T. Ho, M. Mdard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

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

[5] C. Fragouli and A. Markopoulou, "A network coding approach to network monitoring," in *Proc. Allerton*, 2005.

[6] 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.

[7] M. Gjoka, C. Fragouli, P. Sattari, and A. Markopoulou, "Loss tomography in general topologies with network coding," in *Proc. of IEEE Globecom*, 2005.

[8] H. Yao, S. Jaggi, and M. Chen, "Network coding tomography for network failures," in *Proc. of INFOCOM, mini-conference*, 2010.

[9] A. Vardy, "The intractability of computing the minimum distance of a code," *IEEE Transactions on Information Theory*, vol. 43, no. 6, pp. 1757–1766, 1997.

[10] M. J. Siavoshani, C. Fragouli, and S. Diggavi, "On locating byzantine attackers," in *Network Coding Workshop: Theory and Applications*, 2008.

[11] S.R.Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[12] U. K. Sorger, "A new reed-solomon code decoding algorithm based on newton's interpolation," *IEEE Transactions on Information Theory*, vol. 39, no. 2, pp. 358–365, 1993.

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

[14] C. B. Boyer, *A History of Mathematics*, 2nd ed. Wiley, 1968.

[15] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Medard, "Resilient network coding in the presence of byzantine adversaries," in *Proc. of INFOCOM*, 2007.

[16] D. Silva, F. R. Kschischang, and R. Kotter, "A rank-metric approach to error control in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 9, pp. 3951–3967, 2008.