Link Failure Monitoring via Network Coding

Mohammad H. Firooz, Sumit Roy, Linda Bai, and Christopher Lydick
Electrical Engineering Department
University of Washington
Seattle, WA 98105
Email: {firooz,sroy,lyb3,lydick}@u.washington.edu

***Abstract*—In network tomography, we seek to infer link status parameters (delay, congestion, loss rates etc.) *inside a network* through *end-to-end* measurements at (external) boundary nodes. As can be expected, such approaches generically suffer from identifiability problems; i.e., status of links in a large number of network topologies is not identifiable. We introduce an innovative approach based on linear network coding that overcomes this problem. We provide sufficient conditions on network coding coefficients and training sequence under which any *logical network* is guaranteed to be identifiable. In addition, we show that it is possible to locate any congested link inside a network during an arbitrary amount of time by increasing size of transmitted packets, leading to raise in complexity of the method. Further, a probability of success is provided for a random network. OPNET is used to implement the concept and confirm the validity of the claims - simulation results confirm that LNC correctly detects the congested link in situations where standard probing based algorithm fails.**

***Index Terms*—Network Coding, Network Tomography, Graph Theory, Finite Field**

I. Introduction

The need for accurate and fast network monitoring method has increased further in recent years due to the complexity of new services (such as video-conferencing, Internet telephony, and on-line games) that require high-level quality-of-service (QoS) guarantees. This would help network engineers and Internet Service Providers (ISP) to keep track of network utilization and performance. The term *network tomography* was coined by Vardi [1] to encompass these class of approaches that seek to infer internal link parameters and identify link congestion status.

Current network tomography methods can be broadly categorized as follows[2]:

- **Node-oriented:** These methods are based on cooperation among network nodes on an end-to-end route using *control* packets such as ping or traceroute[3]. The challenges of such node-oriented methods arise from the fact that many service providers do not own the entire network and hence do not have access to the internal nodes[4].
- **Path-oriented:** In networks with a defined *boundary*, it is assumed that access is available to all nodes at the edge (and not to any in the interior). A boundary node sends probes to all (or a subset) of other boundary nodes to measure packet attributes on the path between network end-to-end points. Clearly, these *edge-based* methods do not require exchanging special control messages between interior nodes. The primary challenge of such end-to-end probe data [5],[6] to estimate *link level* status is that of *identifiability*, as will be discussed later.

As the Internet evolves towards decentralized, uncooperative, heterogeneous administration and edge-based control, node-oriented tools will be limited in their capability. Accordingly, in this work we only focus on path-oriented methods which have recently attained more attention due to their ability to deal with uncooperative and heterogeneous (sub)networks.

In path-oriented network tomography, probes are sent between two boundary nodes on pre-determined routes (typically the shortest path) between the nodes. For some parameters like delay or failure status, an additive linear model adequately captures the relation between end-to-end and individual link attributes, and can be written as [7], [8]

$$\mathbf{y} = \mathbf{R}\mathbf{x} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^L$ is the L-vector of individual link attributes. The $J \times L$ *binary* matrix $\mathbf{R}$ denotes the routing matrix for the network graph corresponding to the measurements and $\mathbf{y} \in \mathbb{R}^J$ is the measured J-vector of end-to-end path attributes.

Let us assume that the only desired solutions in this framework are binary vectors $\mathbf{x}$ whereby $x_i = 1(0)$ indicates if the corresponding link is congested (not congested). While this is a simplifying assumption - that only congested link experience significant delay and is indicated by a corresponding entry of large magnitude in $\mathbf{x}$, whereas the other entries of $\mathbf{x}$ are relatively small, corresponding to low delays for non-congested links, it has been adopted for modeling in the literature, notably by [9].

In Eq. (1), typically, the number of observations $J \ll L$, because the number of accessible boundary nodes is much smaller than number of links inside the network. Thus the number of variables in Eq. (1) to be estimated is much larger than number of equations in the linear model [10], leading to generic non-uniqueness for any solution to Eq. (1),i.e., inability to uniquely specify which links are congested [11].

A potential solution to the above problem comes from the physics of the problem. Essentially, for a well-designed network there are few simultaneously congested links. That means the maximum number of simultaneously congested links inside the network is limited (at most k). This observation allows an important side constraint to be imposed on any solution to Eq. (1) - namely, we are interested in binary

978-1-4244-8388-4/10/$26.00 ©2010 IEEE

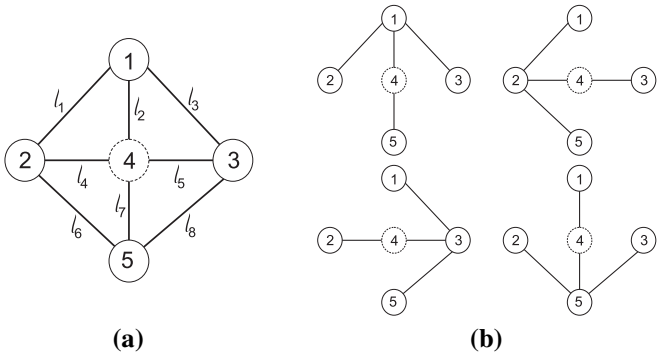


Fig. 1. An example of graph where end-to-end probe sending methods fail. In end-to-end measurement methods, probes are sent from one boundary node to the others. (a)Network topology (b)Spanning trees rooted at boundary nodes

vectors with at most k non-zero entries in $\mathbf{x}$. In this work, we concentrate on $k = 1$; i.e. only a single congested link exists inside the network. This is the simplest possible class of identifiability problems and yet is sufficiently challenging as our investigations will show. A network is defined to be 1-identifiable if congestion status of a single link inside the network can be inferred from the measurements at the receiver(s) [12].

As illustration, consider the network graphs in Fig. 1 and suppose that one of the links l_2 or l_7 is congested (and all others are uncongested). Fig. 1-b shows all the possible shortest paths to all other boundary nodes starting with a source (boundary) node in the network. Obviously, a probe either goes through both l_1 and l_7 or neither. This results in two identical columns (columns 4 and 5) in the network routing matrix as follows:

$$\begin{array}{l} 1 \rightsquigarrow 2 \\ 1 \rightsquigarrow 3 \\ 1 \rightsquigarrow 5 \\ 2 \rightsquigarrow 3 \\ 2 \rightsquigarrow 5 \\ 3 \rightsquigarrow 5 \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

It is evident that it is not possible to identify which of l_2 or l_7 is the congested link using end-to-end measurement. In general, it is proved by Xi et. al. in [13] that a network is 1-identifiable if and only if there are no two identical columns in the network routing matrix.

However, suppose that in place of shortest path routing, we are allowed to route packets from n_1 to n_2 through a longer path that includes l_2, l_4. It is now possible to distinguish between congestion status of links l_2 and l_7. This exemplifies an intrinsic limitation for end-to-end measurement methods based on shortest path routes (or any fixed routing); probes transmitted along such paths contain only minimum information. If it were possible to exchange probes between boundary nodes via other (non-shortest) paths, it would improve network identifiability. However, this is often impractical in path-oriented methods that have fixed routing tables.

We accordingly propose a new approach - based on linear network coding - to achieve the same purpose of enhanced identifiability. The basic idea of linear network coding is to allow an intermediate node to linearly combine packets received on its incoming links and broadcast the result on all of its outgoing links. Due to the broadcasting nature of network coding, a transmitted probe will traverse almost every path between two boundary nodes. Thus, probes received at the destination(s) contain the most information possible about the inside of the network. In addition, because of the linear combination of packets at the intermediate nodes, the packet received at the destination can be written as a linear combination of packets sent out by the source. In this manuscript, we will study the effect of link failure on packets received at the destination(s), when the source packets are given. The key idea of our work is to show there is coding that uniquely changes packets received at the destination when a link goes congested. Moreover, we will show how to design such a code. Since in a network coding paradigm, interior nodes affect the outgoing packets (in contrast to store-and-forward routing), tomography using network coding can be thought of as a combination of node-oriented and path-oriented methods; thereby reaping the advantages of both.

Network coding has received considerable attention in recent years for its potential to achieve the theoretical upper bound (max-flow, min-cut) of network resource utilization. Our intent in this work is to redirect network coding concepts towards a novel application; network status monitoring for graphs hitherto assumed un-identifiable by other means(such as network in Fig. 1). Similar works in this matter include Ho et al. in [12]. However, their approach is based on a multicast tree to find the congested link. This implies the method is restricted to probing, for which they use the shortest path between source-destination pairs. As mentioned, monitoring methods based on shortest path routing suffers from un-identifiability problem. Our approach differs from those of [12] and [14] in that we present how to construct a training sequence to locate a congested link within the network. Moreover, we specify conditions on network coding coefficients of intermediate nodes under which the network is 1-identifiable. This allows us to calculate probability of identifiability when nodes pick their NC coefficients randomly (also known as random network coding). We also show that time to identify the congested link can be reduced by increasing the size of NC packets[1]. Our specific contributions in this work are summarized as follows:

1) We provide conditions under which network coding can be used to locate a single congested link in any *logical network* - i.e. a network containing no nodes with degree two [12].
2) We provide how to generate a training sequence to locate a congested link inside the network.
3) We provide a relation between length of training sequence needed (time to identifiability) and size of network coding packets (complexity of method) to establish

[1]It is noteworthy that our method can be combined with the approach proposed in [14] to infer loss rate within a network.

a fundamental speed/complexity tradeoff.

4) We provide a lower bound for probability of success of our method in a random graph.
5) We implement our proposed method within OPNET simulator - the first implementation of network coding within actual network simulator to the best of our knowledge - and demonstrate the validity of our ideas.

The paper is organized as follows: In Section II, we develop the principles for applying network coding to tomography for a logical network. Implementation of linear network coding (LNC) for such a network in a commercially available simulator (OPNET) is described in Section III for schemes proposed in Section II. The paper concludes with reflections on future work in Section IV.

Notations: We use bold capitals (e.g. $\mathbf{R}$) to represent matrices and bold lowecase symbols (e.g. $\mathbf{y}$) for vectors. The i-th entry of a vector $\mathbf{x}$ is denoted by x_i. A set of sets is denoted by a calligraphic capitalized symbol, e.g. $\mathcal{P}$ and the i-th element, which is itself a set, is denoted by regular capital symbol with i as superscript (e.g. P^i).

II. Tomography with Linear Network Coding

In principle, Linear Network Coding (LNC) is a block code operating on *IP layer* frames, implemented by routers inside the network. The coding is conducted over the finite field $\mathbb{F}_{2^q}$ whereby each coded symbol can be represented by q-bits within an IP layer frame[15]. For the sake of simplicity, we initially consider a delay-free network as in [16], [17], [18] in which information reaches every node instantaneously; our method is readily adapted to a real network where links have finite (non-zero) delay using the buffering method proposed in [18]. LNC has been exploited in [19] and [20] to infer network topology. Consistent with these approaches, we assume that in addition to LNC coefficients at each node, destination nodes are aware of the entire network topology.

We model a communication network consisting of directional links connecting transmitters, switches, and receivers as an directed graph $G(V, E)$ where V is a set of vertices and E is a set of directed edges. Only networks with logical topology are considered here; i.e. degrees of all (interior) nodes in the network (except sources and destinations) are greater than or equal to three, since networks with degree 2 nodes are well-known to be un-identifiable by any end-to-end probing techniques.

A. Network Code Design

Consider a source $s \in V$ and a destination $d \in V$ pair in the network. For a given network graph $G(V, E)$, all nodes apart from the source-destination pair $v \in V - \{s, d\}$, support LNC. In LNC [18], The signal Y_l on an outgoing link $l \in E$ for node v is a linear combination (in finite field $\mathbb{F}_{2^q}$) of the signals Y_j on the incoming links of v, i.e.,

$$Y_l = \sum_{\{j \in E | d(j) = v\}} \gamma_j Y_j \quad , v = o(l), l \in E \tag{3}$$

where $o(l)$ and $d(l)$ represent origin and destination nodes of link $l \in E$, respectively. Operations (addition and multiplication) in Eq. (3) are in finite field $\mathbb{F}_{2^q}$ (for more details refer to [21]).

Let N be the number of paths from s to d and $\mathcal{P}(d)$ be the collection of all those paths. The i-th element $\mathcal{P}(d)$, $P^i(d)$, is the i-th path between s and v. Further suppose that the source s has K outgoing links $e_1, e_2, ..., e_K$ and the symbol α_k is sent over the e_k, $k = 1, 2, \ldots, K$. Let $\mathcal{P}_{e_k}(d)$ denote the collection of paths from source to destination that share the k^{th} outgoing link from the source, i.e.,

$$\mathcal{P}_{e_k}(d) = \{P^i(d) : e_k \in P^i(d) \; s.t. \; o(e_k) = s\} \tag{4}$$

If the source sends a symbol $\alpha \in \mathbb{F}_{2^q}$ over $P^i(d)$, the destination would receive

$$y[n] = \alpha \prod_{l \in P^i(d)} \gamma_l = \alpha \beta_i(G) \tag{5}$$

where $\gamma_l \in \mathbb{F}_{2^q}$ is the coefficient of the link l on path $P^i(d)$ and $\beta_i(G) \in \mathbb{F}_{2^q}$ is the product of LNC coefficients of all links lying on the i-th path from source to destination, $P^i(d)$. The argument G in $\beta_i(G)$ highlights the dependency of β_i on topology G. Now, suppose s sends the symbol $\alpha_k[n]$ over the k-th outgoing link $e_k, k = 1, \ldots, K$ in time slot n (which implies in turn that $\alpha_k[n]$ traverses over all paths $P^i(d) \in \mathcal{P}_{e_k}$). Since network coding is a linear operation, the destination receives, by (5) the following super-imposed symbol in time slot n :

$$y[n] = \sum_{k=1}^{K} \alpha_k[n]. \sum_{P^i(d) \in \mathcal{P}_{e_k}} \prod_{l \in P^i(d)} \gamma_l \tag{6}$$

Eq. (6) can be rewritten as the vector product

$$y[n] = \boldsymbol{\alpha}^T[n] \, \boldsymbol{\beta}(G) \tag{7}$$

where

$$\boldsymbol{\alpha'}_k^T[n] = \overbrace{[\alpha_k[n] \; \ldots \; \alpha_k[n]]}^{|\mathcal{P}_{e_k}| \text{ times}}, \; \boldsymbol{\alpha}^T[n] = [\boldsymbol{\alpha'}_k^T[n]]_{k=1}^{K} \tag{8}$$

and

$$\boldsymbol{\beta}^T(G) = [\prod_{l \in P^i(d)} \gamma_l]_{i=1}^{N} = [\beta_i(G)]_{i=1}^{N} \tag{9}$$

We call $\boldsymbol{\beta}(G)$ the *total network coding vector of the graph* G. If M symbols that constitute a packet are sent in M time slots, the destination receives:

$$\mathbf{y}_{M \times 1} = \mathbf{A}_{M \times N} \, \boldsymbol{\beta}(G)_{N \times 1} \tag{10}$$

where $\mathbf{A}$ is a $M \times N$ matrix whose n^{th} row is $\boldsymbol{\alpha}^T[n]$, the training symbols sent in time slot n (8). It follows by construction that columns of $\mathbf{A}$ corresponding to $\mathcal{P}_{e_k}$ are equal, for $k = 1, 2, ..., K$.

Now, if a link is severely congested, packets are significantly delayed and assumed lost at the destination. Hence, we can model the network with link l in congestion state by its *edge deleted subgraph* denoted by $G_l(V, E_l)$ (for previous definition

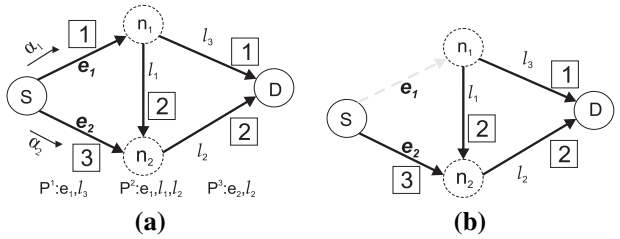


Fig. 2. (a) Intermediate nodes are presented using dashed circle. LNC coefficient of each link is shown in a box next to the link. (b) G_{e_1}: link e_1 is congested. Dashed arrow represents deleted link from network G.

of an edge deleted subgraph refer to [22]). The total network coding vector of the graph G_l, denoted by $\boldsymbol{\beta}(G_l)$, is related to the vector $\boldsymbol{\beta}(G)$, defined in (9) as follows. Clearly, if the congested link doesn't belong to i-th path from source to destination, $P^i(d)$, it will not affect packets going through those paths. That means the i-th entry of $\boldsymbol{\beta}(G_l)$ equals i-th entry of $\boldsymbol{\beta}(G)$ and it is zero if $P^i(d)$ includes the deficient link, i.e.,

$$\begin{aligned} \beta_i(G_l) &= \begin{cases} \beta_i(G) & if \;\; l \notin P^i(d) \\ 0 & o.w. \end{cases} \\ (\boldsymbol{\beta}(G_l))^T &= [\beta_i(G_l)]_{i=1}^{i=N} \end{aligned} \tag{11}$$

where $\beta_i(G)$ is the i-th entry of $\boldsymbol{\beta}(G)$ and l is the congested link.

Now suppose, s sends $\alpha_k[n]$ over path $\mathcal{P}_{e_k}$ in time slot n given link l is congested. The destination node receives

$$y^l[n] = \sum_{k=1}^{K} \alpha_k[n]. \sum_{P^i(d) \in \mathcal{P}_{e_k}} \beta_i(G_l) \tag{12}$$

in time slot n. Using (11) and (12), the following vector form equation can be derived when link l is in congestion and probes are sending in M contiguous time slots:

$$\mathbf{y}^l_{M\times 1} = \mathbf{A}_{M\times N}\, \boldsymbol{\beta}(G_l)_{N\times 1} \tag{13}$$

where $\mathbf{y}^l$ is vector of symbols observed at the destination. Comparing equations (10) and (13) shows that for given matrix $\mathbf{A}$, the received symbols may change in response to link congestion. In the next subsection we will prove that this occurs if $\boldsymbol{\beta}(G)$ and $\mathbf{A}$ satisfy certain conditions, leading to the potential for identifying the congested link. We next provide an illustrative example.

Example 1: Consider the topology in Fig. 2 that consists of 4 nodes and 3 paths between the source and destination. The source has two output links, e_1 and e_2. Hence, using (11), $\mathcal{P}_{e_1}$ and $\mathcal{P}_{e_2}$ are:

$$\mathcal{P} = \{P^1, P^2, P^3\}\; ; \mathcal{P}_{e_1} = \{P^1, P^2\}\; ; \mathcal{P}_{e_2} = \{P^3\} \tag{14}$$

where the end-to-end paths P^1, P^2, P^3 between the source-destination are as defined in Fig. 2-a.

Suppose α_1 and α_2 traverse over e_1 and e_2 respectively. Further, suppose symbols are from $\mathbb{F}_{2^2}$ implying they are 2 bits long. In that case, the output of each intermediate node and the destination is depicted in Fig. 3. Since the network is considered to be delay-free, all nodes receive their information instantaneously. From Fig. 3, the received symbol at destination is

$$y[n] = \alpha_1(1\times 1 + 1\times 2\times 2) + \alpha_2(3\times 2) = 2\alpha_1 + \alpha_2 \tag{15}$$

As defined in (5), $\beta_i(G)$ is defined as product of coefficients of all links on path $P^i(d)$. For example $\beta_2(G)$ is as follows:

$$\beta_2(G) = 1\times 2\times 2 = 3 \tag{16}$$

where the operation is over the finite field $\mathbb{F}_{2^2}$ [21].

Suppose at time slot 1, the source in Fig. 2 sends symbols $\alpha_1 = 1 \in \mathbb{F}_{2^2}$ and $\alpha_2 = 2 \in \mathbb{F}_{2^2}$, respectively. At time slot 2, it transmits $\alpha_1 = 3$ and $\alpha_2 = 3$. In this case, (10) becomes:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 2 \\ 3 & 3 & 3 \end{bmatrix}}_{\text{training sequence } \mathbf{A}} \cdot \underbrace{\begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}}_{\boldsymbol{\beta}(G)} = \underbrace{\begin{bmatrix} 0 \\ 2 \end{bmatrix}}_{\text{received symbols}} \tag{17}$$

Now suppose link e_1 is congested; then G_{e_1} depicted in Fig. 2-b, represents the graph model for this case. As defined in (11), for edge deleted subgraph G_{e_1}, the total network coding vector is now given by

$$\boldsymbol{\beta}(G_{e_1}) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{18}$$

The first and second entries of $\boldsymbol{\beta}(G_{e_1})$ are zero because $e_1 \in P^1$ and $e_1 \in P^2$ (refer to Eq. (11)). For the same symbols sent by source in case of congested link e_1, the destination receivers

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 2 \\ 3 & 3 & 3 \end{bmatrix}}_{\text{training sequence } \mathbf{A}} \cdot \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{\boldsymbol{\beta}(G_{e_1})} = \underbrace{\begin{bmatrix} 2 \\ 3 \end{bmatrix}}_{\text{received symbols}} \tag{19}$$

which is different from the case of no link failure as in (17). For the same transmitted symbols, Table I contains the received symbols at destination for all cases of single link congestion in the network. Such a table may then be used as a lookup to locate/identify the congested link inside the network. It follows that in general, the number of transmission time slots required depends on the choice of q, which can be treated as a design variable. For example if we increase number of bits for LNC from $q = 2$ to $q = 3$, the topology in Fig. 3 is identifiable by transmission in just one time slot. Table III shows received symbol at destination for each link congestion state with $\alpha_1 = 1$ and $\alpha_2 = 4$. We formalize this later in Corollary 5.

TABLE I
SYMBOLS RECEIVED AT DESTINATION FOR TOPOLOGY IN FIG. 2 WITH TRAINING SEQUENCE TRANSMITTED GIVEN IN EQ.(17)

Congested link	None	e_1	e_2	l_1	l_2	l_3
1st time slot	0	2	2	3	1	1
2nd time slot	2	3	1	0	1	3

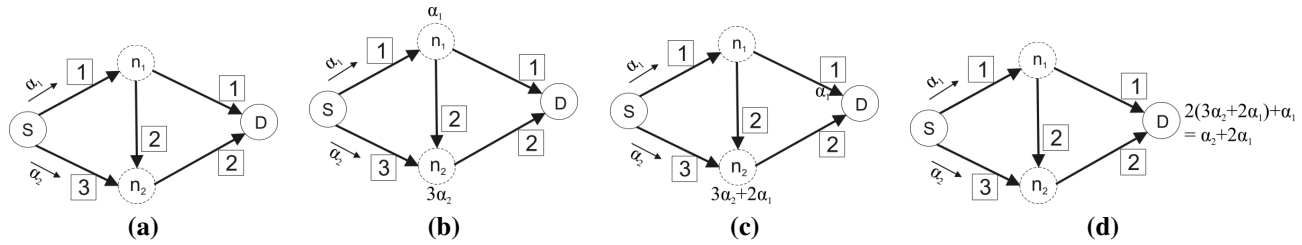


Fig. 3. Output of each intermediate node when source sends symbols α_1 and α_2 over its outgoing links. For delay-free network this is instantaneous.

TABLE II
SYMBOLS RECEIVED AT DESTINATION FOR TOPOLOGY IN FIG. 3 WITH TRAINING SEQUENCE TRANSMITTED IN ONE TIME SLOT $\mathbf{A} = [1\ \ 1\ \ 4]$

Congested link	None	l_1	l_2	l_3	l_4	l_5
1st time slot	6	4	2	5	7	1

B. Link Identifiability Results

As explained via the example above, received symbols at the destination change in the event of a link failure. The following theorem describes the conditions on training sequence ($\mathbf{A}$) and LNC coefficients, under which there exists a one-to-one correspondence between link congestion states and received symbols at the destination.

***Theorem** 1:* Consider a logical network $G(V, E)$, with a source-destination pair $s \in V$, $d \in V$. Let N be the number of paths from s to d, and K the number of outgoing links of s. Suppose each intermediate node has fixed network coding coefficients and the total network coding vector for the graph G is $\boldsymbol{\beta}_{N\times 1}$. Let $\mathbf{A}_{M\times N}$ be constructed from the *training* symbols sent from source to destination over M time slots. Then, for $M \geq K$, there exists a matrix $\mathbf{A}$ and vector $\boldsymbol{\beta}$ such that $G(V, E)$ is 1-identifiable. Moreover rank of $\mathbf{A}$ is K.

Proof: See [23].

The theorem provides *sufficient* conditions that guarantees 1-identifiability of any logical topology $G(V, E)$ using network coding under certain conditions on the training sequence and total LNC vector. Under these sufficient conditions, the congestion states corresponding to different single link failures results in different symbols received at destination. Therefore, by having a simple lookup table, it is possible to identify the congested link in any logical network. The following theorem shows how to construct a training sequence to locate a single congested link inside a network.

Corollary *2:* Given an acyclic, directed and connected graph $G(V, E)$ with source node s and destination node d where s has K outgoing links; Then $\mathbf{A}_{M\times N}$ can be used as training sequence to locate a single failed link in $G(V, E)$ if and only if rank$(\mathbf{A}) = K$.

Theorem 1 talks about possibility of using network coding in the application of link monitoring when number of time slots (M) is greater or equal to number of outgoing links of the source (K). However, by increasing the number of bits assigned to network coding coefficients, it is possible to locate a failure link using fewer number of time slots.

***Theorem** 3:* Assume an acyclic, directed and connected graph $G(V; E)$ with source-destination pair $s \in V$ and $d \in V$ as before. Let the K outgoing links of s be represented by $e_1, e_2, \ldots e_K$. Let N_i be total number of paths from s to d that starts with e_i, i.e., $\sum_{i=1}^{K} N_i = N$. Assume q bits per symbol are used in network coding and M is number of time slots used to send training sequence. Further, assume $Z = \{1, 2, \ldots, K\}$ and $\mathcal{Z}_M$ is the collection of all partitions of Z with size M;

$$\mathcal{Z}_M = \{\, \{H_1, H_2, ..., H_M\} |\, \cup_{i=1}^{M} H_i = Z\}$$

Then $G(V; E)$ is identifiable using NC in field $\mathbb{F}_{2^q}$ if q satisfies the following inequality:

$$q \geq \min_{\{H_i, i=1,\ldots,M\}\in\mathcal{Z}_M} \max_{i} \sum_{j\in H_i} N_j \tag{20}$$

Proof: See Appendix.

Theorem 3 provides a tradeoff between number of time slots for training sequence (speed of the method) and size of network coding coefficient (complexity) to make a network $G(V, E)$ identifiable. In other words, by increasing complexity of the method (increasing q) it is possible to save some time slots (decreasing M). The following example further illustrates the above theorem.

Example 2: Consider the topology in Fig. 4 that consists of 5 paths between the source and destination; i.e. $N = 5$. Source has 3 outgoing links, $K = 3$, and using the definition of N_i we have, $N_1 = 2, N_2 = 1, N_3 = 2$. Suppose we want to locate a failure link by sending symbols from source to destination in 2 time slots, i.e. $M = 2$. Theorem 3 helps us find the minimum number of bits q, that guarantees identifiability in Fig. 4.

Let $Z = \{1, 2, 3\}$. Since $M = 2$, we enumerate all the 2-partitions of Z as given below:

$$\mathcal{Z}_2 = \{\, \{\, \{1\}, \{2, 3\}\}, \{\, \{2\}, \{1, 3\}\}, \{\, \{3\}, \{1, 2\}\}\, \} \tag{21}$$

The outer maximization in Theorem 4 is over $\{N_1 = 2, (N_2 + N_3) = 3\}$ which is 3; our network is found to be identifiable with $M = 2$ and $q \geq 3$.

□

As mentioned before, Theorem 3 implies that increasing the number of time slots used for identifiability of a network $G(V, E)$ decreases number of bits per symbol for NC coefficients. If number of bits of network coding coefficients are large enough, it is even possible to locate a congested link in one time slot. The following two corollaries states these observations more precisely:

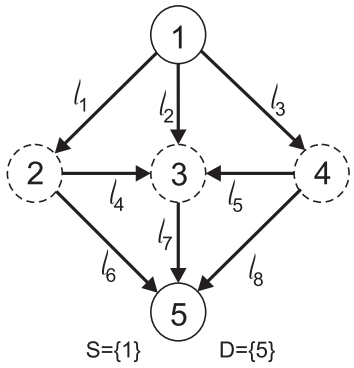


Fig. 4. A network with 5 nodes and five paths, N=5, paths from source to destination. Source has K=3 outgoing links. It is possible to locate a failure link in this network by sending training sequence in 2 time slots if q, number of bits in NC coefficients, are greater than 3.

Corollary *4:* Suppose $G(V, E)$ is identifiable using network coding with q_1 bits per symbol and training sequence in M_1 time slots. Then G is identifiable in $M_2 > M_1$ time slots using NC values having q_2 bits per symbol where $q_2 \leq q_1$.

Corollary *5:* It is possible to locate a congested link in network $G(V, E)$ in one time slot, if $q \geq N$.

In all the results thus far, we have assumed that the network coding coefficients are fixed and chosen so as to satisfy the identifiability conditions. What if the nodes choose the LNC coefficients randomly? In that case, identifiability can be described as a random event. The following theorem provides a lower bound for probability of identifiability of a random graph in such a scenario.

***Theorem** 6:* Let graph $G(V, E)$ be a network with two nodes $s \in V$ and $d \in V$ as source and destination respectively. If each intermediate node choose their NC coefficients uniformly from the elements of $\mathbb{F}_{2^q}$, then the probability that $G(V, E)$ is 1-identifiable is bounded from below by $1 - |E|(|E| + 1)(\frac{1}{2^q})^M$

Proof: See [23].

C. Multi-source, Multi-destination Networks

In previous subsections, we established a novel link failure monitoring method based on single source-destination pair. It may be possible to send probes between an arbitrary number of sources and destinations; the identifiability of a network in such scenarios is discussed next.

Let us consider a logical network $G(V, E)$ with sets $S \subset V$ and $D \subset V$ of sources and destinations. Note that we have access to all nodes in S and D simultaneously and consequently we may consider them as a *super nodes* as presented in Fig. 5 . Hence, a multi-source multi-destination network can be studied as an equivalent single source, single destination network by substituting set of source and destinations with a single super node source and destination, respectively, represented by the new graph $G'(V, E)$. It follows that if G' is identifiable (unidentifiable), then so is G because every edge in G has 1:1 correspondence with an edge in G'.

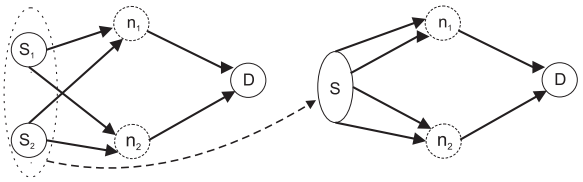


Fig. 5. In multi-source multi-destination network, sources S can be thought of a super node. Similarly set of destinations D can be seen as a super node.

Therefore, results in previous section may be extended to a multi-source multi-destination network $G(V, E)$ in this way.

III. Simulation results

In this Section we briefly describe our LNC simulator [24] constructed within OPNET^{TM}14.5 aided by Matlab 7.1 that is used for finite field calculations necessary for network coding. The simulation results from applying the proposed link failure monitoring scheme to the network in Fig. 1 (which is not identifiable by end-to-end measurements) is presented. Finally, we apply the method to the network graph for the University of Washington's Electrical Engineering department network of servers and present identifiability results.

A. Simulation Environment

Ours is the first known implementation of Network Coding (NC) within OPNET. OPNET was selected because of its wide acceptance as a network modeling tool within both the academic and commercial communities [25].

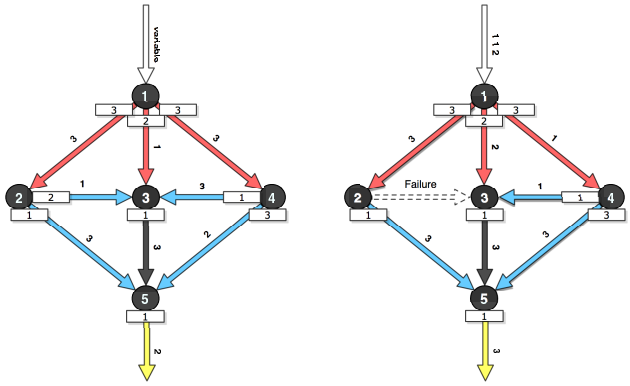
We employed network coding at transport layer (instead of IP layer) largely for convenience - it readily allows adding *hidden* data within the TCP/UDP frame in OPNET, which is invisible to the end-user and to the simulation statistics [24]. Any binary vector of length q, can be interpreted as an element in $\mathbb{F}_{2^q}$, the finite field with 2^q elements. In our network coding implementation, we assign a q-bit field called *LNC field* within the TCP/UDP header, for linear network coding. Only the contents in LNC field is used for network coding operation. In addition, a 1-bit flag within TCP/UDP header determines if the packet is a network coding packet. Once a router receives a packet, it identifies the packet type by looking at the 1-bit flag embedded in TCP/UDP header. If the packet is a network-coded packet, the data in LNC field is extracted and queued at a buffer for a predetermined amount of time after which they are combined and the router clears the buffer. The result of linear combining is written in LNC field of the outgoing packet which is forwarded on all of the outgoing links via unidirectional broadcast.

B. Validation

Fig. 6 demonstrates a test scenario that was run to validate our findings - this topology (same as in Fig. 1) is not identifiable using end-to-end probe monitoring. The arrows indicate the network coding graph which overlays the 100 Mbps Full Duplex connection between routers. Linear NC coefficient assigned to each link - γ_l in Eq. (3)- are shown in a box next to the link. Table III presents received values

TABLE III
SYMBOLS RECEIVED AT DESTINATION FOR TOPOLOGY IN FIG. 6 WITH TRAINING SEQUENCE GIVEN IN EQ. (22).

Congested link	None	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8
1st time slot	2	0	0	0	3	3	1	0	1
2nd time slot	0	2	3	1	1	3	3	1	2
3rd time slot	0	2	1	3	1	2	3	2	1



(a) **(b)**

Fig. 6. Topology given in Figure 1 which is not identifiable using end-to-end measuring (a)Snapshot of network coding simulation; NC coefficients, in $\mathbb{F}_{2^2}$ (γ_l), for each link is shown in a box next to the link (b)Snapshot of simulation when l_3 is congested.

at destination for different link congestion states when the training sequence given by elements of matrix $\mathbf{A}$ is used; it is the lookup table used by destination (node 5) to identify any congested link in the network.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 3 & 3 & 3 \end{bmatrix} \tag{22}$$

Further tests of our approach was conducted on a larger graph resembling that of the University of Washington's Electrical Engineering network shown in Fig. 7. Thirteen subnets (represented by the numbers 1-13) are all connected through Full Duplex Ethernet links to backbone routers (represented as A, B, C and D), which then connect to the rest of campus. Fig. 7-(a) depicts the network coding coefficients of each node and the training sequence used is given below.

$$\mathbf{A} = \begin{bmatrix} 5 & 5 & 2 & 5 & 2 & 1 & 5 & 3 & 5 & 4 & 3 & 6 & 7 \\ 4 & 7 & 4 & 1 & 2 & 7 & 7 & 1 & 2 & 2 & 6 & 4 & 2 \end{bmatrix}$$

Fig. 7-(b) shows the received symbols at destination (node A) for different link congestion scenarios. Symbols next to each link represent values at the destination when that link is congested. Destination node A uses these symbols to uniquely locate any congested link in the network; *i.e.* UW EE network is seen to be identifiable under proposed network coding monitoring.

IV. CONCLUSION

This work has presented a novel approach to link status monitoring based on a deterministic approach that exploits

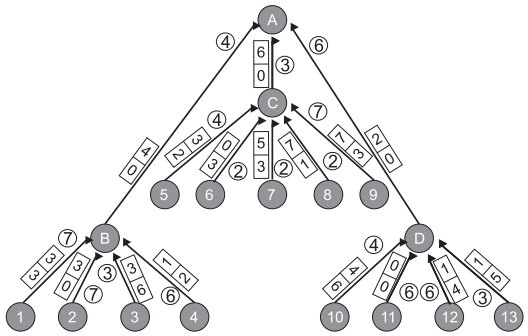


Fig. 7. UW Electrical Engineering network topology. Network coding coefficients of each link, in $\mathbb{F}_{2^3}$ (γ_l), is presented in a circle next to that link. Numbers next to each link shows received symbols at destination when the link is congested.

LNC at the internal nodes in a network. The key problem of identifiability for such approaches was highlighted and various insights provided regarding this concept. New sufficient conditions were derived for successfully identifying a congested link in any *logical network*, and trade-offs between length of training slots and size of the network coding alphabet established. Finally, the method is verified by implementation within OPNET to confirm the validity of the claims.

REFERENCES

[1] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *J. of the American Statistical Association*, vol. 91, no. 433, 1996.

[2] E. Lawrence, G. Michailidis, V. Nair, and B. Xi, "Network tomography: a review and recent developments," *Ann Arbor*, vol. 1001, pp. 48 109–1107.

[3] S. W. Richard, "TCP/IP illustrated," *Addison-Welsey Publishing Company*, 1994.

[4] Y. Bejerano and R. Rastogi, "Robust monitoring of link delays and faults in IP networks," in *22nd Annual Joint Conf. the IEEE Computer and Communications Societies (INFOCOM'03)*, vol. 1, 2003, pp. 134–144.

[5] R. Caceres, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal loss characteristics," *IEEE Trans. Information theory*, vol. 45, no. 7, pp. 2462–2480, 1999.

[6] Y. Tsang, M. Coates, and R. Nowak, "Passive network tomography using EM algorithms," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing(ICASSP'01)*, vol. 3, 2001.

[7] A. Coates, A. Hero III, R. Nowak, and B. Yu, "Internet tomography," *IEEE Signal processing magazine*, vol. 19, no. 3, pp. 47–65, 2002.

[8] T. Bu, N. Duffield, F. Presti, and D. Towsley, "Network tomography on general topologies," *ACM SIGMETRICS Performance Evaluation Review*, vol. 30, no. 1, pp. 21–30, 2002.

[9] N. Duffield, "Network tomography of binary network performance characteristics," *IEEE Trans. Information Theory*, vol. 52, no. 12, pp. 5373–5388, 2006.

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

[11] Y. Xia and D. Tse, "Inference of link delay in communication networks," *IEEE J. on Selected Areas in Communications*, vol. 24, no. 12, pp. 2235–2248, 2006.

[12] M. Gjoka, C. Fragouli, P. Sattari, and A. Markopoulou, "Loss tomography in general topologies with network coding," in *Proc. IEEE Conf. Global Telecommunications(GLOBECOM'07)*, 2007, pp. 381–386.

[13] B. Xi, G. Michailidis, and V. Nair, "Estimating network loss rates using active tomography," *J. of the American Statistical Association*, vol. 101, no. 476, pp. 1430–1448, 2006.

[14] C. Fragouli and A. Markopoulou, "A network coding approach to overlay network monitoring," in *Allerton Conference*, 2005.

[15] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.
[16] T. Ho, M. Medard, J. Shi, M. Effros, and D. Karger, "On randomized network coding," in *Proc. Annual Allerton Conf. on Communication Control and Computing*, vol. 41, no. 1, 2003, pp. 11–20.
[17] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM trans. Networking*, vol. 11, no. 5, pp. 782–795, 2003.
[18] S. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.
[19] G. Sharma, S. Jaggi, and B. Dey, "Network tomography via network coding," in *Information Theory and Applications Workshop*, 2008, pp. 151–157.
[20] C. Fragouli, A. Markopoulou, and S. Diggavi, "Topology inference using network coding," in *Proc. of 44th Annual Allerton Conference on Communication, Control and Computing*, 2006.
[21] E. Horowitz, "Modular arithmetic and finite field theory: A tutorial," in *Proceedings of the second ACM symposium on Symbolic and algebraic manipulation*, 1971, pp. 188–194.
[22] J. Clark and D. Holton, *A first look at graph theory*. World Scientific Publishing Company, 1991.
[23] M. Firooz, S. Roy, L. Bai, and C. Lydick, "Link status monitorin using network coding," *UW Technical Report*, Sept 2009.
[24] C. Lydick, "Tutorial: Network coding using opnet and matlab," *Thechnical report*, May 2009.
[25] D. Cavin, Y. Sasson, and A. Schiper, "On the accuracy of manet simulators," in *Proceedings of the second ACM international workshop on Principles of mobile computing*, 2002, pp. 38–43.