Application of Network Coding and Compressed Sensing in Networking

Mohammad Hamed Firooz

A dissertation submitted in partial fulfillment of
the requirements for

Doctor of Philosophy

University of Washington

2013

Reading Committee:

Sumit Roy, Chair

Professor Hui Liu

Professor Radha Poovendran

Professor Maryam Fazel

Professor Rekha Thomas

Program Authorized to Offer Degree:
Electrical Engineering

University of Washington

Abstract

Application of Network Coding and Compressed Sensing in Networking

Mohammad Hamed Firooz

Chair of the Supervisory Committee:
Professor Sumit Roy
Electrical Engineering Department

The growing adoption of data communication has resulted in the dramatic increase for higher capacity and performance. Communications engineers have historically optimized Physical/MAC layers so as to increase aggregate network throughput. However, in recent years the researchers have focused their attention increasingly towards understanding the other layers in a network to improve the efficiency of the networks. The main objective of this work is to explore applications of recently developed ideas in coding and data acquisition for networking.

Network coding has received considerable attention in recent years for its potential for achieving the theoretical upper bound (max-flow, min-cut) of network resource utilization via the introduction of coding concepts at the network layer (IP layer). Instead of just receiving a packet and forwarding it to suitable next hop, intermediate nodes perform a linear operation upon receiving packets and broadcast the result to all of their neighbors. In this work we exploit network coding for various applications in wired and wireless networks. First, we use NC to locate congested links inside a wired network. Then, we investigate application of network coding in wireless relay networks and wireless broadcasting. Finally. we explore employing network coding for data sharing in wireless ad-hoc networks.

The idea of compressed sensing is based on the fact that with some minimal prior knowledge about the data vector of a signal, it is possible to reconstruct that signal (efficiently) from a very limited number of measurements (samples). Compressed Sensing has gained

a fast-growing interest in many applications. In this work we explore the application of CS in network monitoring and tomography. We will show that most of networks routing matrices can be used as measurement matrix in compressed sampling. We provide an upper-bound for delay recovery when compressed sensing is used to recover link delay inside the networks. Finally, we provide an algorithm to design network routing matrix such that network monitoring (or tomography) put minimum burden into the network.

TABLE OF CONTENTS

Page

LIST OF FIGURES

DEDICATION

To my parents and brothers for their love and support throughout my life.
To my advisor who gave me the freedom to explore on my own, and at the same time the guidance to recover when my steps faltered.

Chapter 1

INTRODUCTION

1.1 Network Coding

Network coding operation (usually) is in finite field $\mathbb{F}_{2^q}$ in which a packet is divided to q-bit long sections where each q-bit represents a symbol in the finite field. If the length of a packet is not dividable by q, zero padding may apply. A packet which has rq bits can be thought of a vector in $\mathbb{F}_{2^q}^r$. Every node linearly combines receiving packets (in field $\mathbb{F}_{2^q}$) and broadcasts it to all nodes it is connected; i.e. output symbol Y_o in every output link of an arbitrary node v inside the network can be written as a linear combination of incoming signals as it is depicted in Figure 1.1, where γ_is are called NC coefficients of node v.

In linear network coding the receiver(s) need only know the overall linear combination of source processes in each of their incoming signals [86][102]. This information can be sent through the network as a vector, for each signal, of coefficients to each of the source processes, updated at each coding node by applying the same linear mappings to the coefficient vectors as the information signals. Because of broadcast nature of network coding, this method can be used in networks in which there are multiple paths between a source and a destination.

Our intent in this work is to redirect network coding concepts towards some novel applications in wired and wireless networks. We first start with application of network coding

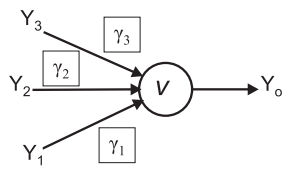


Figure 1.1: In network coding, output signal is a linear combination of inputs: $Y_o = \gamma_1 Y_1 + \gamma_2 Y_2 + \gamma_3 Y_3$.

in network tomography for wired networks. Then we propose some applications in wireless networks in which network coding brings significant improvement.

We propose an active link monitoring method using network coding which is able to find a single congested link in any logical network[1] (any network with logical topology [136], [83]). It is shown in [145] and [36] that using network coding makes it possible to infer network topology and coefficients of intermediate nodes in the network. Based on these works, we assume that destination nodes are aware of network topology and NC coefficients of each node in the network. Similar works in this matter include Ho et al. in [83]; however, their work suffers from using multi-cast tree to find the congested link which implies unique path between source(s) and destination(s). We shall relax this assumption in our work.

As mentioned before network coding was first introduced for wired networks and our link monitoring problem follows that intention. However, there are some works in the literature that shows advantage of network coding in wireless networks. The main challenge of extending network coding to a wireless network can be categorized as follows:

- Omnidirectional transmission property of wireless networks
- No simultaneous packet transmission and reception by any node
- Possible destructive interference effects among concurrent transmissions

Omnidirectional transmissions have been modeled in [121] and [119] as hyperlinks with additional constraint sets that can prevent nodes from transmitting and receiving packets simultaneously. Authors in [118] study the problem through the use of linear programs to optimize the network resources based on link costs.

The interference effects have been incorporated by [119] for joint optimization of MAC and network flows while preventing simultaneous transmission and reception by any node. The transmission-reception pairs are based on the signal-to-interference-plus-noise-ratio (SINR) threshold criterion. This implies that a node would not be able to receive and

[1] network containing no nodes with degree two

transmit packets at the same time, as the SINR at the receiver would be very small, since the signal of the transmitter would dominate the interference.

There are also some works using network coding for wireless multicasting. A decentralized solutions to achieve minimum cost multicast connections have been introduced in [120]. Wu et. al. in [167] exploit network coding for energy-efficient multicasting. To avoid interference, they consider only one isolated link at a time. For energy efficiency purposes only, there is no need to consider interference effects and simultaneous transmissions. However, if we consider additional objectives such as throughput or delay efficiency, then network codes must be jointly designed with MAC. Authors in [139] study the cross-layer design possibilities of joint MAC and network coding to operate in wireless ad hoc networks.

In this work, we first consider data dissemination in a wireless network using network coding. By increasing the deployment of ad-hoc networks and sensor networks, there is an increasing demand in fast and reliable data dissemination algorithms for these networks. We provide an upper bound on the number of time slots needed for fully dissemination of data in a network with finite number of nodes given the reception probability between each pair of nodes.

Then we consider data exchange in a relay network. It has been shown in the literature that throughput of a relay network would be increased by deploying network coding either at MAC layer [166] or PHY layer [174]. We will investigate this result more by implementing network coding at MAC layer of a WiFi framework. We will explore the practical challenges of having network coding at the MAC layer and the actual gain comes out of it.

1.2 Compressed Sensing

Data acquisition and reconstruction used to follow Nyquist density sampling theory. This principle states that to reconstruct a signal, the number of samples needed to reconstruct a signal without error is dictated by its bandwidth, the length of the shortest interval which contains the support of the spectrum of the signal under study [23]. In fact, inadequate sampling of a signal can induce aliasing, meaning that the same set of samples may describe many different signals. However, for some signals of interest, it is possible to recover the signal accurately or sometimes even exactly from a number of samples which is far smaller

than the desired resolution of the signal. To avoid aliasing a preconditioning step is necessary to eliminate spurious (incorrect) solutions. This preconditioning step introduces a new approach in data sampling called compressed sensing.

According to Compressed Sensing (CS) theory any sufficiently compressible signal can be accurately recovered from a small number of nonadaptive linear projection samples. Essentially, for any signal $\mathbf{x} \in \mathbb{R}^n$, the its linear projection representation is equal to $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $m \times n$ matrix $\mathbf{A}$ ($m \ll n$) is referred to as measurement matrix. The main challenge in traditional compressed sensing is to construct $\mathbf{A}$ with the following desirable (and conflicting) properties: a) achieve maximum possible compression (m/n small) and allow b) accurate reconstruction of $\mathbf{x}$ from $\mathbf{y}$ based on the sparsity of $\mathbf{x}$ using c) a fast decoding algorithm.

In applications like medical imaging or seismology, Compressed Sensing has seen quick acceptance since the breakthrough works. However its role in networking is still limited. Authors in [80] describes how to apply compressed sensing to the decentralized compression of networked data. Coates et. al. use CS and the fact that performance on two overlapping paths should be correlated, to estimate performance metrics such as delay on many end-to-end paths in a network using measurements taken on only a few of these paths [40]. Reconstruction of missing values in traffic measurement matrices is proposed in [181]. Traffic measurement matrices specify the traffic volumes between origin and destination pairs in a network. However sometimes some measurements are missing due to failure of data collection system or unreliability of exploited transport protocol. They use idea of compressed sensing to recover missing entries in traffic matrices.

Compressed sensing has been recently used in the literature for network tomography [40, 40, 173, 164]. Authors in [40] use compressed sensing to estimate metrics (such as delay) for unmeasured paths. They assume that they have access to part of the end-to-end measurement data and their goal is to infer metrics for the remaining unobserved paths. Xu et. al. in [173] study the application of compressed sensing for sparse vectors over graphs. Their work however largely suffered from assumptions that are unsuited to delay estimation in practice. They perform a standard random walk over a *sufficiently* connected graph to take the measurements. However, in network tomography the measurement matrix

(i.e., routing matrix) is already given and it is fixed. Besides, most of the networks are not sufficiently connected [56, 129]. In this work we assume that the routing path between any pair of boundary nodes is predetermined–usually by shortest path algorithm–and we do not put any constraint on network topology.

Here, by using the concept of compressed sensing, we first try to estimate link delay inside a network. In this application, the measurement matrix is network routing matrix and the end-to-end measurement is the projection. Here we assume that the measurement matrix is already pre-determined, and is not a design choice. Hence, the question is whether a given routing matrix is an appropriate measurement matrix for compressed sensing, i.e. if it satisfies objective b) above.

In above problem the routing matrix is assumed to be fixed and given. However, as a network designer or network administrator, one has the opportunity to set the routing matrix. That means, there are a number of end-to-end paths between boundary nodes that the routing matrix can be built on. On the other hand, injecting monitoring packet put additional burden on the network which, if not controlled, could decrease network throughput. Thus we want to minimize number of paths selected such that it is still possible to recover link delay. We will show that this is a integer programming problem and we will introduce a simple heuristic algorithm to solve it.

1.3 Notations

The following notations are used throughout this manuscript.

- Bold calligraphic capitalized symbols (e.g. $\boldsymbol{\mathcal{U}}$) represents random variables.
- Bold capitals (e.g., $\mathbf{R}$) represents matrices.
- For the matrix $\mathbf{R}$, $\mathcal{N}(\mathbf{R})$ denotes its null space.
- For the matrix $\mathbf{R}$, superscript ^{T} denotes its transpose.
- Bold lowercase symbols (e.g., $\mathbf{y}$) is for vectors.

- The i-th entry of a vector $\mathbf{x}$ is denoted by x_i.
- If $\mathbf{x} \in \mathbb{R}^n$, the l_p-norm of $\mathbf{x}$ is defined as follows:

$$\| \mathbf{x} \|_p = \left(\sum_{i=1}^{n} x_i^p \right)^{\frac{1}{p}}. \tag{1.1}$$

- A set is denoted by a normal capitals (e.g. V).
- The cardinality of a set S is defined as number of elements S has and it is denoted by $|S|$.
- A set of sets is denoted by a calligraphic capitalized symbol. For example, $\mathcal{R}$ is the set of all end-to-end paths in the network[2].
- The i-th set, which is itself a set, is denoted by capital symbol with i as superscript (e.g., P^i).
- An empty set is denoted by $\emptyset$.
- For any set $S \subset \{1, 2, 3, ...n\}$, S^c represents the complement.
- The number of i-combinations from a given set of N elements, when order doesn't matter, is denoted by $C(N, i)$.
- For a set $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the subspace spanned by elements of S is called the subspace of S and $dim(S)$ denotes the dimension of that subspace
- The equality between two subspaces S_1 and S_2 is denoted by $S_1 \equiv S_2$.
- For any vector $\mathbf{x} \in \mathbb{R}^n$, vector $\mathbf{x}_S \in \mathbb{R}^n$ has entries defined as follows:

$$(x_S)_i = \begin{cases} x_i & \text{if } i \in S \\ 0 & \text{o.w.} \end{cases}. \tag{1.2}$$

[2]Each path is itself set of nodes and edges of the network.

Chapter 2
NETWORK TOMOGRAPHY

Monitoring of link properties (delay, loss rates etc.) within the Internet has been stimulated by the demand for network management tasks such as fault and congestion detection. 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. In 1996, the term *network tomography* was coined by Vardi [159] to encompass a 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:

- **Node-oriented:** These methods are based on cooperation among network nodes on a route using *control* packets. For example, active probing tools such as ping or traceroute, measure and report attributes of the round-trip path (from sender to receiver and back) based on separate probe packets [137]. The challenges of such node-oriented methods arise from the fact that service providers typically do not own the network and hence do not have access to the internal nodes[10].

- **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 points. Clearly, these *edge-based* methods do not require exchanging control messages with any interior nodes. The primary challenge confronting such end-to-end probe data [21],[156] based link status estimation 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. Accord-

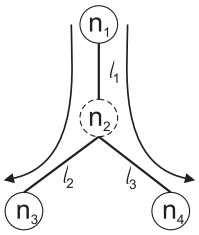


Figure 2.1: In end-to-end measurement methods, probes are sent from one boundary node to the others.

ingly, 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 these are the shortest paths between the nodes. For some parameters like delay (which is the main concern in this manuscript), an additive linear model adequately captures the relation between end-to-end and individual link delays, and can be written as [38, 19]

$$\mathbf{y} = \mathbf{R}\mathbf{x} \tag{2.1}$$

where $\mathbf{x}$ is the $n \times 1$ (unknown) vector of individual link delay. The $r \times n$ *binary* matrix $\mathbf{R}$ is the routing matrix for the network graph corresponding to the paths chosen for the probes (each row of the matrix is a path) and $\mathbf{y} \in \mathbb{R}^r$ is the measured r-vector of end-to-end path delays.

In Eq. (2.1), typically the number of observations $r \ll n$, because the number of accessible boundary nodes is much smaller than number of links inside the network. Thus the number of variables in Eq. (2.1) to be estimated is much larger than number of equations ($rank(\mathbf{R}) < L$) [28] in the linear model, leading to generic non-uniqueness for any solution to Eq. (2.1), i.e., inability to uniquely specify which links are congested [169].

Definition: A network is said to be identifiable for a given monitoring scheme (choice of sources, receivers, whether intermediate nodes collaborate) if status (congestion, delay, packet loss) of all links inside the network can be inferred from the measurements at the receiver(s) [72].

As mentioned before, networks are not identifiable in general since the routing matrix $\mathbf{R}$ in 2.1 is not invertible. A potential solution to this problem is to have side information about the network. For example, we can limit number of congested links inside the network or we can only focus on links with high packet loss. In this chapter, we first study the application of network coding in locating congested links inside the networks. We will show that using network coding will enhance networks identifiability; i.e. network coding is able to identify a congested link in a network that is deemed un-identifiable using end-to-end probe based methods. Then, we investigate using compressed sensing in delay estimation in networks using end-to-end measurement method. Theory of Compressed Sensing enables us to put conditions on routing matrix of identifiable networks and having an upper-bound on estimation error.

2.1 Link Status Monitoring Using Network Coding

consider the network graph in Figure 2.2 with the following routing matrix. We are assuming the shortest path for the routing algorithm but our argument is not limited to this.

$$\mathbf{R} = \begin{array}{c} \\ 1 \rightsquigarrow 2 \\ 1 \rightsquigarrow 3 \\ 1 \rightsquigarrow 5 \\ 2 \rightsquigarrow 3 \\ 2 \rightsquigarrow 5 \\ 3 \rightsquigarrow 5 \end{array} \begin{array}{c} \begin{array}{cccccccc} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 & l_7 & l_8 \end{array} \\ \left[\begin{array}{cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array} . \tag{2.2}$$

Suppose that one of the links l_2 or l_7 is congested and all others are uncongested. Figure 2.2-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_2 and l_7 or neither. It is not possible to identify which link, l_2 or l_7 is congested, by using end-to-end measurement. The routing matrix of the network is given in Eq. (2.2). Since columns 2 and 7 are identical, network is unidentifiable. However, suppose that in place of the shortest path routing, we are allowed to route packets from n_1 to n_2 through a longer

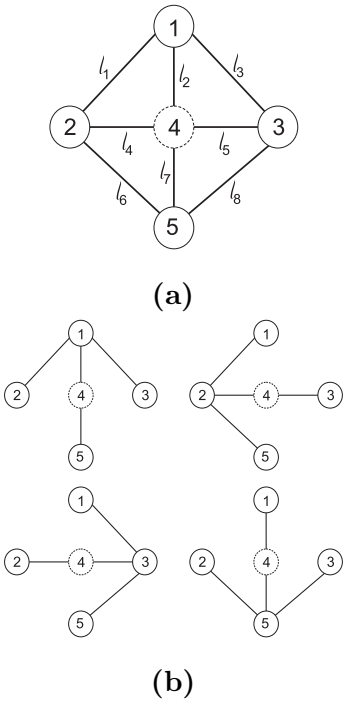


Figure 2.2: An example of graph where end-to-end probe sending methods fail (a)Network topology (b)Spanning trees rooted at boundary nodes

path that includes l_2 and l_4. It is now possible to distinguish between congestion statuses of links l_2 and l_7. This exemplifies the intrinsic limitation of end-to-end measurement; probes transmitted along such paths contain only minimal information.

Our intuition suggests that, if it were possible, exchanging probes between boundary nodes via additional paths would improve network identifiability. However, this is often impractical in path-oriented methods that have fixed routing tables.

Motivated by the above example, we use the concept of network coding for link monitoring. We show that a monitoring scheme based on network coding is able to identify a congested link that is deemed unidentifiable using end-to-end probe based observation methods (such as in the network in Figure 2.2).

It is worth pointing out that scalability is largely an unsolved problem in network tomography. Most of the algorithms take advantage of the hierarchical nature of networks to overcome this issue [44, 111]. Almost every large network is divided using the subnetting

technique into small subnetworks that are arranged in a hierarchical architecture [153, 105]. This approach not only increases network performance but also has the advantage in network management and scalability. Based on this fact, network tomography algorithms focus on small networks [178, 108]. The algorithm, then, is applied to each level of network hierarchy to seek link properties in that level.

Similarly in this work, we first target small wired networks. With wired networks, the probability of having a high delay link is very low[1] [15, 134], so the probability of having more than one congested link (i.e., $k > 1$) is negligible. Hence, we derive our analytical results for $k = 1$-identifiable networks. Then we provide simulation evidence for $k > 1$-identifiability to show the scalability of the algorithm.

2.1.1 Related Works

The intuition that the failure of a single link will change LNC-received symbols at the receivers has been introduced in the literature. Ho et al. [83] consider network monitoring using network coding based on multicast tree. Fragouli et al. in [65] show identifiability criteria for networks with tree topologies. However, these works do not consider networks with general topology. The idea of using network coding for monitoring networks with general topology has been discussed in [66, 72, 145] and[175]. For example, Yao et al. in [175] establish a one-to-one correspondence between network topology and network coding transfer function. Our work differs from these works in that we propose an active network monitoring algorithm in which a training sequence is exchanged between the boundary nodes to identify the congested link. In other words, we show how to design network coding coefficients and a training sequence to locate a single congested link. One of our main concerns here is to design an efficient training sequence that can locate a congested link as fast as possible. For that, we provide our speed/complexity tradeoff, which shows the relation between the size of the network coding coefficients and the number of time slots needed for the training sequence transmission.

For the sake of simplicity, we initially consider a delay-free network as in [87, 103]

[1] Usually of order 10^{-3}.

and[112], in which information reaches every node instantaneously; as we explain in Section 2.1.8, our method is readily adapted to a real network where links have finite (non-zero) delay. LNC has been exploited in [145], [36] and [144] 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. However, the source node and interior nodes do *not* need to know the topology or NC coefficients of the network.

2.1.2 Tomography with Linear Network Coding

2.1.3 Formulation

As is customary in network coding literature, a communication network consisting of links connecting transmitters, switches, and receivers can be modeled as an directed graph $G(V, E)$ where V is a set of vertices and E is a set of 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[2], since networks with degree 2 nodes are well-known to be unidentifiable by any end-to-end probing techniques [72][66].

In LNC [112], intermediate nodes linearly combine packets received on their incoming edges and broadcast the result over all outgoing edges. The coded symbols are transmitted as vectors of bits of length q, represented as an element of the finite field $\mathbb{F}_{2^q}$. The length of vectors is equal in all transmission and all links are assumed to be synchronized [82, 85] with zero delay. It is worth mentioning once again that a zero delay network is a common assumption in network coding related works since it reduces the complexity of the analysis. In Section 2.1.7, we talk about how to implement network coding in networks with non-zero but finite delay.

2.1.4 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 network coding. The signal Y_l on an outgoing link $l \in E$ for node v is a linear combination

[2]Degree of a node is defined as number of links (incoming or outgoing) connected to it.

(in finite field $\mathbb{F}_{2^q}$) of the signals Y_j on the incoming links of v, i.e.,

$$Y_l = \sum_{\{j \in E | d(j)=v\}} \gamma_j Y_j \quad , v = o(l), l \in E, \tag{2.3}$$

where $o(l)$ and $d(l)$ represent origin and destination nodes of link $l \in E$, respectively. Operations (addition and multiplication) in Eq. (2.3) are in finite field $\mathbb{F}_{2^q}$. A self-contained summary is given in [58]; for more details, readers should refer, for example, to [89].

For each node $v \in V - \{s, d\}$, $\mathcal{P}(v)$ is defined as the collection of all paths from s to v in the directed graph $G(V, E)$; it's i-th element $P^i(v)$ is the i-th path between s and v. Suppose there are total of N paths from source s to destination d, i.e., $N = |\mathcal{P}(d)|$. 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 k-th outgoing link 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.,

$$\mathcal{P}_{e_k}(d) = \{P^i(d) : e_k \in P^i(d) \; s.t. \; o(e_k) = s\}. \tag{2.4}$$

In fact sets $\mathcal{P}_{e_k}$, $k = 1, ..., K$ are partitions of $\mathcal{P}(d)$, i.e., $\mathcal{P}(d) = \cup_{k=1}^{K} \mathcal{P}_{e_k}(d)$ and $\mathcal{P}_{e_i}(d) \cap \mathcal{P}_{e_j}(d) = \phi$, $i \neq j$.

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

$$\begin{aligned} y[n] &= \alpha \prod_{l \in P^i(d)} \gamma_l & (2.5) \\ &= \alpha \beta_i(G), & (2.6) \end{aligned}$$

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 (2.6) the following super-imposed symbol in time slot n :

$$y[n] = \sum_{k=1}^{K} \alpha_k[n]. \sum_{P^i(d) \in \mathcal{P}_{e_k}} \prod_{l \in P^i(d)} \gamma_l. \tag{2.7}$$

Eq. (2.7) can be rewritten as the vector product

$$y[n] = \boldsymbol{\alpha}^T[n]\,\boldsymbol{\beta}(G), \tag{2.8}$$

where

$$\begin{aligned} \boldsymbol{\alpha'}_k^T[n] &= \overbrace{[\alpha_k[n]\,\alpha_k[n]\,\ldots\,\alpha_k[n]]}^{|\mathcal{P}_{e_k}| \text{ times}}, & (2.9)\\ \boldsymbol{\alpha}^T[n] &= [\boldsymbol{\alpha'}_k^T[n]]_{k=1}^{K}, & (2.10) \end{aligned}$$

and

$$\boldsymbol{\beta}^T(G) = [\prod_{l \in P^i(d)} \gamma_l]_{i=1}^{N} = [\beta_i(G)]_{i=1}^{N}. \tag{2.11}$$

We call $\boldsymbol{\beta}(G)$ the *total network coding vector of the graph* G; the i-th entry of $\boldsymbol{\beta}$ is the product of LNC coefficients of all nodes lying on the i-th path from source to destination, $P^i(d)$. Note that $\boldsymbol{\alpha'}_k[n]$ is a repeated length-$|\mathcal{P}_{e_k}|$ vector comprising of the symbol $\alpha_k[n]$ (sent over the k-th outgoing link of the source at n-th time slot). If M symbols that constitute a packet are sent in M time slots, the destination receives:

$$\mathbf{y}_{M\times 1} = \mathbf{A}_{M\times N}\,\boldsymbol{\beta}(G)_{N\times 1}, \tag{2.12}$$

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 as given in Eq. (2.10). 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 precise definition of an edge deleted subgraph refer to [37]). In other words, for the original network $G(V, E)$, $G_l(V, E_l)$ represents the same network with link l is congested such that packets sent on that link are dropped. 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 (2.11) 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.,

$$\begin{aligned} \beta_i(G_l) &= \begin{cases} \beta_i(G) & \text{if } l \notin P^i(d) \\ 0 & \text{otherwise} \end{cases}, \\ (\boldsymbol{\beta}(G_l))^T &= [\beta_i(G_l)]_{i=1}^{i=N}, \end{aligned} \tag{2.13}$$

where $\beta_i(G)$ is the i-th entry of $\boldsymbol{\beta}(G)$ (or equivalently product of coefficients of all links on path $P^i(d)$) defined in (2.11) 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

$$y^l[n] = \sum_{k=1}^{K} \alpha_k[n]. \sum_{P^i(d) \in \mathcal{P}_{e_k}} \beta(G_l)_i, \tag{2.14}$$

in the time slot n. Using (2.13), (2.10) and (2.14), the following vector form equation can be derived when link l is in congestion:

$$y^l[n] = \boldsymbol{\alpha}^T[n] \boldsymbol{\beta}(G_l). \tag{2.15}$$

Sending probe packets in M contiguous time slots results in M linear equations, which can be written as:

$$\mathbf{y}^l_{M\times 1} = \mathbf{A}_{M\times N}\, \boldsymbol{\beta}(G_l)_{N\times 1}, \tag{2.16}$$

where $\mathbf{y}^l$ is vector of symbols observed at the destination in M time slots with link l congested. Comparing equations (2.12) and (2.16) shows that for given matrix $\mathbf{A}$, the received symbols change in response to link congestion. In the next subsection we will prove that this occurs if the network coding vector of the graph G are chosen to satisfy certain conditions, leading to the potential for identifying the congested link. We next provide an illustrative example.

Example 1: Consider the topology in Figure 2.3 that consists of 4 nodes and 3 paths between the source node, S, and destination node, D. The source has two output links, e_1 and e_2. Hence, using (2.13), $\mathcal{P}_{e_1}$ and $\mathcal{P}_{e_2}$ are:

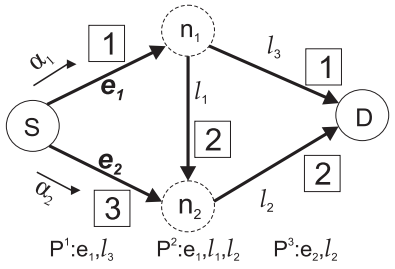


Figure 2.3: A network with two boundary nodes and two interior nodes which contains three paths from source (node S) to destination (node D). Linear network coding coefficient of each link is shown in a box next to the link.

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

where the end-to-end paths P^1, P^2, P^3 between the source-destination pair are as defined in Figure 2.3.

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. By Eq. (2.7), the received symbol at destination is

$$\begin{aligned} y[n] &= (\alpha_1 \times (1 \times 1 + 1 \times 2 \times 2)) + (\alpha_2 \times (3 \times 2)) \\ &= 2\alpha_1 + \alpha_2. \end{aligned}$$

As defined in (2.6), $\beta_i(G)$ is defined as product of coefficients of all links on path $P^i(d)$. For example $\beta_2(G)$ is as follows:

$$\beta_2(G) = 1 \times 2 \times 2 = 3, \tag{2.17}$$

where the operation $\times$ is as defined over the finite field $\mathbb{F}_{2^2}$.

From Equations (2.11) and (2.10), the vector $\boldsymbol{\alpha}$ and *total network coding vector* $\boldsymbol{\beta}$ are

obtained as:

$$\boldsymbol{\alpha}^T = \begin{bmatrix} \alpha_1 & \alpha_1 & \alpha_2 \end{bmatrix}, \tag{2.18}$$

$$\boldsymbol{\beta}(G) = \begin{bmatrix} 1 & 3 & 1 \end{bmatrix}^T, \tag{2.19}$$

where α_1 and α_2 are two symbols in $\mathbb{F}_{2^2}$.

For example, suppose at time slot 1, the source in Figure 2.3 sends symbols $\alpha_1 = 1 \in \mathbb{F}_{2^2}$ and $\alpha_2 = 0 \in \mathbb{F}_{2^2}$, respectively. At time slot 2, it transmits $\alpha_1 = 0$ and $\alpha_2 = 1$. In this case, (2.12) becomes:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{training sequence } \mathbf{A}} \cdot \underbrace{\begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}}_{\boldsymbol{\beta}(G)} = \underbrace{\begin{bmatrix} 2 \\ 1 \end{bmatrix}}_{\text{received symbols}} . \tag{2.20}$$

Now suppose link e_1 is congested; then G_{e_1} represents the graph model for this case. As defined in (2.13), for edge deleted subgraph G_{e_1}, the total network coding vector is now given by

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

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. (2.13)). For the same symbols sent by source in case of congested link e_1, the destination receives:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{training sequence } \mathbf{A}} \cdot \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{\boldsymbol{\beta}(G_{e_1})} = \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\text{received symbols}} , \tag{2.21}$$

which is different from the case of no link failure as in (2.20). For the same transmitted symbols, Table 2.1 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.

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

Table 2.1: Symbols received at destination for topology in Figure 2.3 with training sequence transmitted in two time slots given in Eq.(2.20)

For $q = 2$, each received symbol at the destination belongs to the symbol set $\{0, 1, 2, 3\}$ representing the elements of $\mathbb{F}_{2^2}$. Therefore, for 2-bit long network coding, in each transmission (in one time slot) received symbol conveys only 4 different values. On the other hand, there are five links in the network which means receiver has to distinguish between six different states: single link congestion for each of the five links and no congestion case. These observations leads to the fact that for $q = 2$, the number of transmissions required in this simple network cannot be less than two time slots.

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 the length of network coding by one bit (i.e., $q = 3$) the topology in Figure 2.3 is identifiable by transmission in just one time slot with the same NC coefficients for each interior node. In $\mathbb{F}_{2^3}$, the the total network coding of the graph changes to the following[3]:

$$\boldsymbol{\beta}(G_{e_1}) = \begin{bmatrix} 1 & 4 & 6 \end{bmatrix}^T.$$

Let change the training sequence to $\alpha_1 = 1$ and $\alpha_2 = 1$. Table 2.3 shows received symbol at destination for each link congestion state in this case. We formalize this later in Theorem 4.

2.1.5 Link Identifiability Results

As explained via the example above, received symbols at the destination change in the event of a link failure. The following theorems describe the conditions on training sequence ($\mathbf{A}$)

[3]Clearly, when the field of operation is changed, the results of multiplication and summation operations are changed as well

Congested link	None	e_1	e_2	l_1	l_2	l_3
1st time slot	3	6	5	7	1	2

Table 2.2: Symbols received at destination for topology in Figure 2.3 with training sequence transmitted in one time slot $\mathbf{A} = [1 \ 1 \ 1]$

and LNC coefficients, under which there exists a one-to-one correspondence between link congestion states and received symbols at the destination.

lemma 1 *Consider a directed, acyclic and connected graph $G(V, E)$, with a source- destination pair $s \in V$, $d \in V$. Let N be the number of paths from node s to d, and K the number of outgoing links of s (equivalently, the degree of s). Suppose each intermediate node $j \in V - \{s, d\}$ has fixed network coding coefficients and the total network coding vector for the graph G is $\boldsymbol{\beta}_{N\times 1}$. Let $\mathbf{A}_{M\times N} = [a_{ij}]$ be constructed from the* training *symbols sent over N different paths from source to destination over M time slots. Then, for $M \geq K$ and q large enough, there exists a matrix $\mathbf{A}$ and vector $\boldsymbol{\beta}$ such that the following conditions hold:*

1 For each subgraph $G_l(V, E_l)$, $\mathbf{A}\boldsymbol{\beta} \neq \mathbf{A}\boldsymbol{\beta}(G_l)$

2 For each pair of subgraphs $G_{l_1}(V, E_{l_1})$ and $G_{l_2}(V, E_{l_2})$, $\mathbf{A}\boldsymbol{\beta}(G_{l_1}) \neq \mathbf{A}\boldsymbol{\beta}(G_{l_2})$

Proof: See A.1.

The following two Theorems give how to design training matrix and network coding coefficients for a graph $G(V, E)$.

Theorem 1 *Consider a directed, acyclic and connected graph $G(V, E)$ with the source-destination pair $s \in V$ and $d \in V$. Let $e_1, e_2, ...e_K$ be the K outgoing links of the source s. Let $\mathcal{P}_{e_i}$ be the set of paths from source to destination that have e_i in common. Further, assume β_{k,e_i} is the product of linear network coding coefficients of all links on k-th path in $\mathcal{P}_{e_i}$. Then $G(V, E)$ is 1-identifiable if the following holds for every outgoing links of s:*

$$\sum_{k=1}^{N_i} \zeta_k \beta_{k,e_i} = 0, \zeta_k \in \{0, 1\} \iff \zeta_k = 0, \ \forall k. \tag{2.22}$$

In other words β_{k,e_i} for all k should be independent numbers in the finite field $\mathbb{F}_{2^q}$ [4].

Proof: See A.1.

Theorem 2 *Let $G(V,E)$ be a directed, acyclic and connected graph with the source node s and the destination node d. Let $e_1, e_2, ...e_K$ be K outgoing links of the source s. Moreover, let $\alpha_i[j]$ be the training symbol transmitted on the link e_i at the time slot j. Assume NC coefficients of G satisfy the condition in Eq. (2.22). Then it is possible to locate any congested link in $G(V,E)$, if*

$$\alpha_i[i] = 1, \alpha_i[j] = 0 \;\; i \neq j, i = 1, 2, ..., K \tag{2.23}$$

Proof: See A.1.

Theorems 1 and 2 gives sufficient conditions to create NC coefficients and training sequence such that the graph G be 1-identifiable.

Lemma 1 talks about the possibility of using network coding in the application of link monitoring. Then Theorems 1 and 2 provide *sufficient* conditions, respectively, on NC coefficients and the training sequence that guarantee identifiability of any logical topology using network coding. 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.

As will be discussed more in simulation section, for a given network $G(V,E)$, one way to design NC coefficients for the network is to select them randomly and then check to see if they satisfy condition in Eq. (2.22). The probability that a set of randomly selected coefficients satisfies the condition in Theorem 1 is given by the following theorem.

Theorem 3 *Let graph $G(V,E)$ be a directed, acyclic and connected graph with two nodes $s \in V$ and $d \in V$ as source and destination respectively. If each intermediate node $j \in V - \{s,d\}$ choose their NC coefficients uniformly from the elements of $\mathbb{F}_{2^q}$, then the probability that all links in $G(V,E)$ are identifiable is bounded from below by $1 - |E|(|E|+1)(\frac{1}{2^q})^M$.*

[4] For more information about independent numbers refer to [123].

Proof: See A.1.

In all of the above results the number of time slots (M) is greater or equal to number of outgoing links of the source (K). However, as we illustrated in Example 1, by increasing the number of bits assigned to network coding coefficients, it is possible to locate a congested link in fewer number of time slots.

Theorem 4 *Assume a directed, acyclic and connected graph $G(V, E)$ with the source-destination pair $s \in V$ and $d \in V$ as before. Let the K outgoing links of s be represented by $e_1, e_2, \dots e_K$. Let N_i be total number of paths from s to d that starts with e_i, i.e., $N_i = |P_{e_i}(d)|$. Assume q bits per symbol are used in network coding and M is number of time slots used to send training sequence. Further, assume $Z = \{1, 2, \dots, K\}$ and $\mathcal{Z}_M$ is the collection of all partitions of Z with size M;*

$$\mathcal{Z}_M = \{\, \{H_1, H_2, ..., H_M\} | \cup_{i=1}^{M} H_i = Z, H_i \cap H_j = \emptyset \; i \neq j, H_i \neq \emptyset \; \forall i\} \tag{2.24}$$

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

$$q \geq \min_{\{H_i, i=1,...,M\} \in \mathcal{Z}_M} \max_{i} \sum_{j \in H_i} N_j. \tag{2.25}$$

Proof: See A.1.

The following example further illustrates the above theorem.

Example 2: Consider the topology in Figure 2.4 that consists of 5 paths between the source and destination, i.e., $N = 5$. The 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 4 helps us find the minimum number of bits q, that guarantees identifiability in Figure 2.4.

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

$$\mathcal{Z}_2 = \{\, \{\, \{1\}, \{2, 3\}\}, \{\, \{2\}, \{1, 3\}\}, \{\, \{3\}, \{1, 2\}\}\, \}. \tag{2.26}$$

The inner maximization in Theorem 4 is over $\{N_1 = 2, (N_2 + N_3) = 3\}$ which is 3; our network is found to be 1-identifiable with $M = 2$ and $q \geq 3$.

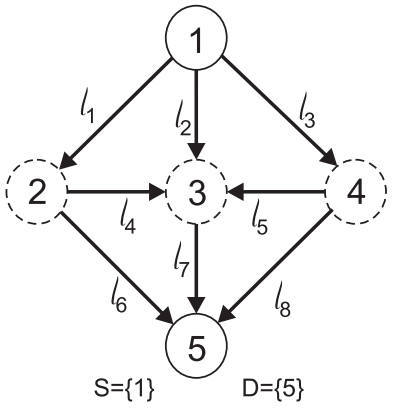


Figure 2.4: A network with 5 nodes and five paths, N=5, paths from source to destination. Source has K=3 outgoing links. It is possible to locate a failure link in this network by sending training sequence in 2 time slots if q, the number of bits in NC coefficients, are greater than 3.

□

Theorem 4 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 the 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:

Corollary 5 *Suppose $G(V, E)$ is identifiable using network coding with q_1 bits per symbol and a training sequence in M_1 time slots. Then G is identifiable using a training sequence in $M_2 > M_1$ time slots with NC values having q_2 bits per symbol where $q_2 \leq q_1$.*

Corollary 6 *It is possible to locate a congested link in network $G(V, E)$ in one time slot, if q number of bits assigned to LNC is greater than or equal to total number of paths between source and destination, i.e., $q \geq N$.*

Above theorem and corollaries provide a tradeoff between number of time slots for training sequence (speed of the method) and size of the 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 theorems gives the sufficient condition on the NC coefficients and the training sequence to make the network $G(V,E)$ 1-identifiable when $M < K$.

Theorem 7 *Consider a directed, acyclic and connected graph $G(V,E)$ with the source-destination pair $s \in V$ and $d \in V$. Let $e_1, e_2, ...e_K$ be the K outgoing links of the source s, and $\mathcal{P}_{e_i}$ be the set of paths from source to destination that have e_i in common. Moreover, let M and q satisfy the inequality in Eq.* (2.25)*, and*

$$\{H_i^*, i = 1, ...M\} = arg \min_{\mathcal{Z}_M} \max_i \sum_{j \in H_i} N_j. \tag{2.27}$$

Finally, assume β_{k,e_i} is the product of linear network coding coefficients of all links on k-th in $\mathcal{P}_{e_i}$. Then $G(V,E)$ is 1-identifiable if the following holds for every $H_i^*, i = 1, 2, ..., M$:

$$\sum_{j \in H_i^*} \sum_{k=1}^{N_i} \zeta_{k,j} \beta_{k,e_j} = 0, \zeta_{k,j} \in \{0,1\} \iff \zeta_{k,j} = 0, \ \forall k \ \forall j. \tag{2.28}$$

In other words β_{k,e_j} for all $k = 1, 2, ..., N_i$ and all $j \in H_i^*$ should be independent numbers in the finite field $\mathbb{F}_{2^q}$.

Proof: See A.1.

Theorem 8 *Given a directed, acyclic and connected graph $G(V,E)$ with the source node s and the destination node d. Let $e_1, e_2, ...e_K$ be the K outgoing links of the source s, and M be the desired number of time slots used to send training sequence, and*

$$\{H_i^*, i = 1, ...M\} = arg \min_{\mathcal{Z}_M} \max_i \sum_{j \in H_i} N_j. \tag{2.29}$$

Assume that the number of bits assigned for NC, q, satisfies inequality (2.25)*. Then it is possible to locate any congested link in $G(V,E)$, if*

$$\alpha_k[i] = 1, \alpha_{k'}[i] = 0, k \in H_i^*, k' \notin H_i^*, \tag{2.30}$$

where $\alpha_{k'}[i]$ is the training symbol transmitted on the link e_k at the time slot i.

Proof: See A.1.

Note that as we defied before $\{H_i^*, i = 1, ...M\}$ is one of the partitions of $Z = \{1, 2, ..., M\}$.

For link failure monitoring, one can decide how fast s/he wants to locate the congested link inside the network. Then, for that desired number of time slots M, Theorem 4 gives a lower bound of the size of the network coding. Then Theorem 7 gives sufficient conditions on NC coefficients such that the network is 1-identifiable. To find the NC coefficient we can use the same approach followed for the case $M = K$, i.e., NC coefficients are selected randomly and checked to see if they satisfy Eq. (2.28). Finally, to locate the congested link inside the network the training sequence given in Theorem 8 is used.

2.1.6 Multi-source, Multi-destination Networks

In previous sections, we established a novel linear network coding based approach which guarantees identifiability of a logical network using probes between only a 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. We do *not* claim any optimality for the proposed approach; our aim here is to simply show that the method in the previous section for a single source-destination pair is readily extended to multi-source, multi-destination scenario with minor modifications[5].

As an example, consider the network given on the right hand side of Figure 2.5. In this graph, there are two sources, S_1 and S_2, and one destination D. Note that we have access to both S_1 and S_2 simultaneously and consequently we may consider the pair as a *super node* as presented in Figure 2.5 . Hence, a multi-source multi-destination network can be studied as an equivalent single source, single destination network by substituting set of sources and destinations with a single super node source and destination, as 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'. Therefore, results in previous section may be extended to a multi-source multi-destination network $G(V, E)$ in this way.

[5]Exploring the advantages from proper training sequence design for multi-source, multi-destination case is left for future work.

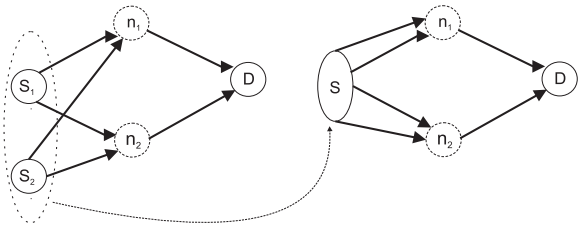


Figure 2.5: In multi-source multi-destination network, sources S can be thought of a super node. Similarly set of destinations D can be seen as a super node.

2.1.7 Simulation results

In this Section we describe our linear network coding simulator constructed within OPNET^{TM}14.5 aided by MATLAB 7.1 that is used for finite field calculations necessary for network coding. We first discussed how our result for delay-free networks can be applied to networks with finite delay. Then, the simulation results from applying the proposed link failure monitoring scheme to the network in Figure 2.2 (which is not identifiable by end-to-end measurements) is presented. Then we preset a network with two sources and one destination to evaluate our proposed method for multi-source multi-destination networks in Section 2.1.6.

In this manuscript, we only focus our attention on 1-identifiability. However, the idea of using network coding for failure monitoring can be extended to $k > 1$, which is an avenue for future research. To support this claim, In Section 2.1.11 we present a 2-identifiable network which can be monitored by network coding.

2.1.8 Networks with Finite Delays

Given an acyclic network with delay, we can either operate the network in a continuous mode (where information is continuously injected into the network) or with burst mode. In the former, the same injected information can take different routes causing different delays through the network. Koetter et al. in [103] treat this case by using memory at the receiver node(s). In burst-oriented mode, each node transmits information on an outgoing link only if an input has been observed on all incoming links or a timer times out. This approach is taken by Li et al. [112], and leads to a situation where a network with delay can be thought

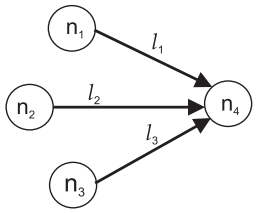


Figure 2.6: waiting time of node n_4 is: $\tau_w(n_4) = \max_{i=1,2,3}\{\tau_w(n_i) + \tau_d(l_i)\}$.

of being equivalent to a delay-free network and the results in previous subsection therefore apply. We use this method jointly with the buffering model proposed in [36], where an index called generation number is added to header of packets to distinguish between packets sent in successive time slots.

Let $\tau_d(l)$ be the delay of link l when it is not congested. In that case, a packet traversing over path $P^i(d)$, defined in Section 2.1.4, encounters net delay of $\sum_{l\in P^i(d)} \tau_d(l)$. We assume that the processing time in each node is negligible. We define $P^{max}(v)$ as the longest path between source and node $v \in V$:

$$P^{max}(v) \in \mathcal{P}(v) \text{ and } |P^{max}(v)| \geq |P^j(v)|. \tag{2.31}$$

Let $\tau_w(v)$ be the wait time of node v, *i.e.*, the amount of time node v has to wait before combining the received packets. In other words, at the start of each time slot (every T seconds), node $v \in V - \{s\}$ waits for $\tau_w(v)$, and then linearly combines all the received packets in that duration and broadcasts the result on all of its outgoing links. Packets received after $\tau_w(v)$ and before the start of the next time slot are dropped.

The waiting time of each node linearly depends on the following two parameters: 1- waiting time of its neighbors from which it receives packets, and 2- delays of its incoming links. This dependency can be written as below(see Figure 2.6):

$$\tau_w(v) = \max_{\{l\in E: v=d(l)\}} \{\tau_d(l) + \tau_w(o(l))\}, \tag{2.32}$$

where $o(l)$ and $d(l)$ represent origin and destination of link $l \in E$, respectively.

If the delay caused by congestion (which is ideally defined to be infinite) is greater than waiting time at the destination, the above buffering renders the delay network equivalent to

a delay-free network. Note that by definition of waiting time at each node, if the congestion delay exceeds the waiting time at destination, it is greater than the waiting time of each interior node.

2.1.9 Simulation Environment

Ours is the first publicly available implementation of Network Coding (NC) in the network simulation environment OPNET. OPNET was selected because of its wide acceptance as a network modeling tool within both the academic and commercial communities [29]. To implement network coding in OPNET, the primary need was a method for queueing packets within each router to account for variable transmission and reception times, and to ensure that packets were combined correctly. The routers model were modified to be able to distinguish between non-network coded and network coded packets through the use of a *flag bit* within the UDP header of received packets; thereafter, the intercepted NC (network-coded) packets are sent for separate processing while the non-NC packets are processed normally. In addition, because of the inherent necessity of network coding to operate on unidirectional links, each interface within a router model is designated as a SEND or RECEIVE interface only for the network coded packets while operating regularly with non-network coded packets. For interfaces with only SEND NC packets, any received NC packets were discarded. Conversely, the RECEIVE interfaces intercepted and processed the NC packets.

Network coding - by definition- is intended for implementation in Layer 3, over IP frames. For the purposes of this simulation, it was not necessary to implement those extensive features; rather it was convenient to develop an implementation which takes advantage of pre-existing OPNET functionality. Therefore, we employ network coding at transport layer (instead of IP layer) largely for convenience - it readily allows adding *hidden* data within the TCP/UDP frame header in OPNET, which is invisible to the end-user and does not impact simulation statistics [122]. As mentioned in Section 2.1.2, 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 (refer to Section 2.1.8) 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. Note that this network-coding approach is different from the bidirectional physical layer broadcast as described in Section 2.1.2.

To simulate congestion within a link, the identified NC packets were either dropped or significantly delayed. Our implementation thus only affects the NC packets, and is transparent to all non-NC coded packets. Since NC involves finite field calculations which is not explicitly supported in OPNET, we implemented the MATLAB API within OPNET [46] to use the Galois Field functions within MATLAB's Communications Toolbox. A tutorial [122] describing our implementation and the OPNET and Matlab codes are available for download at [63].

2.1.10 Validation

Figure 2.7 demonstrates a test scenario that was run within OPNET to validate our findings - this topology (same as in Figure 2.2) 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.

For this network we choose the number of time slots to be $M = K = 3$. Given that, Theorem 4 derives sufficient lower bound on the number of bits for NC, i.e., $q \geq 3$. Starting from $q = 3$, we run a decreasing order iteration on q to find NC coefficients and training sequence that make the network 1-identifiable (code available at [122]). The goal is to find the minimum number of bits (lowest complexity) for NC which still can locate any single congested link in the network. For any q in the iteration, the network coding coefficients are selected uniformly random in $\mathbb{F}_{2^q}$ and checked to see if they satisfy the condition in Eq.

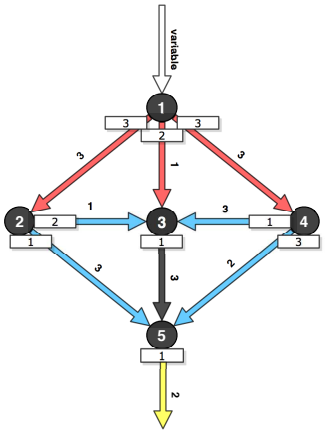


Figure 2.7: Topology given in Figure 2.2 which is not identifiable using end-to-end measuring. This is a snapshot of our simulation. The network coefficient of each link is in a box on that link. The finite field is $\mathbb{F}_{2^2}$.

2.22. The minimum q that satisfies this condition happens to be $q = 2$. Then the training symbols, α_is, are selected as explained in Theorem 2.

Linear NC coefficient assigned to each link are shown in a box next to the link and the training sequence given by elements of matrix A is as follows.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}. \tag{2.33}$$

Table 2.3 presents received values at destination for different link congestion states. it is a lookup table used by destination (node 5) to identify any congested link in the network. Now consider the network in Figure 2.8 with two sources. As mentioned in Section 2.1.6, one way to apply the results in Section 2.1.2 developed for one source one destination networks, is to consider both sources as a single virtual super-node. The virtual super node for this network is depicted as dashed circle around the two sources. This network is 1-identifiable in $M = 2$ time slots with NC for $q \geq 5$. This is achieved using the same approach as before with a decreasing order iteration starting from $q = 8$ (derived from Theorem 4) and determining the minimum value for which the NC coefficients satisfy the sufficient conditions

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

Table 2.3: Symbols received at destination for topology in Figure 2.7 with training sequence given in Eq. (2.33)

for identifiability in Theorem 1. The super-node and training sequence are given in Figure 2.8. The network coding coefficients are given bellow and the lookup table is omitted for lack of space and is available at [58].

	S	S_1	S_2	n_1	n_2	n_3	n_4	n_5	n_6	D
S	0	21	15	0	0	0	0	0	0	0
S_1	0	0	0	31	15	2	0	0	0	0
S_2	0	0	0	11	2	20	0	0	0	0
n_1	0	0	0	0	0	1	12	5	0	0
n_2	0	15	2	0	0	7	22	0	5	0
n_3	0	2	20	1	7	0	19	0	0	0
n_4	0	0	0	12	22	19	0	11	31	0
n_5	0	0	0	5	0	0	11	0	0	16
n_6	0	0	0	0	5	0	31	0	0	22
D	0	0	0	0	0	0	0	16	22	0

(2.34)

2.1.11 2-identifiable Network

In this section, we explore $k = 2$ identifiability via simulation to obtain further insights. Consider the network presented in Figure 2.9-(a) with 2 sources, 2 destinations, 4 intermediate nodes and 15 links. This network is 2-identifiable with Network Coding for $q \geq 8$ [6]. The Network Coding coefficients for the links were randomly chosen and are given as below:

[6] This was empirically determined by increasing the size of the finite field.

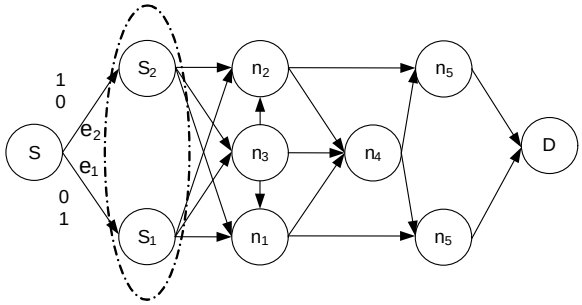


Figure 2.8: A network with two sources to examine the 'super-node' approach. The super node is depicted by a dashed circle around two main sources.

	S_1	S_2	n_1	n_2	n_3	n_4	D_1	D_2
S_1	0	0	9	98	174	0	0	0
S_2	0	0	71	196	168	0	0	0
n_1	0	0	0	0	0	179	63	0
n_2	0	0	0	0	0	228	0	234
n_3	0	0	210	125	0	245	0	0
n_4	0	0	0	0	0	0	51	194
D_1	0	0	0	0	0	0	0	0
D_2	0	0	0	0	0	0	0	0

(2.35)

As discussed in Section 2.1.6, we treat two sources and two destinations as super nodes. Figure 2.9-(b) depicts the network with its super source, S. In the first time slot, S transmits 1 and 0 over e_1 and e_2 respectively[7]. In the second time slot it sends 0 and 1. For $k = 2$, the total number of possible failure events for a network with $|E|$ links equals $|E|(|E|+1)/2+1$, representing no failure, exactly 1 failure and 2 failures. Hence, a node needs a lookup table with $|E|(|E| + 1)/2 + 1$ entries to identify the specific event. Hence the given network requires lookup table has 120 entries, of which the first 5 are shown in Table 2.4.

[7]The training sequence is the same as it is Theorem 2.

Congested link	None	l_1	l_1, l_2	l_1, l_3	l_1, l_4	...
1st time slot	133	140	51	220	93	...
2nd time slot	193	31	105	152	38	...

Table 2.4: Symbols received at destination for topology in Figure 2.9

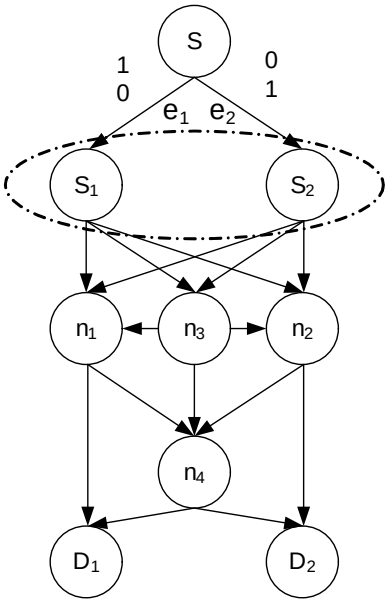


Figure 2.9: A network which is 2-identifiable using Network Coding. The super-node S is depicted by dashed circle around S_1 and S_2.

2.2 Link Delay Estimation Using Compressed Sensing

In Eq. (2.1), usually, the number of observations r is much less than the number of variables n (i.e., $r \ll n$) because the number of accessible boundary nodes is much smaller than the number of links inside the network. Thus, the number of variables in Eq. (2.1) to be estimated is much larger than the number of equations [28], leading to the generic nonuniqueness of solutions to Eq. (2.1), i.e., the inability to uniquely determine link delay [169] from end-to-end measurements. However, the problem of identifying only the (few)

links with large delays (a.k.a congested links[8]) suggests the possibility of improved mechanisms to solve the under-determined system in Eq. (2.1), provided that the *sparsity* of the desired solution can be exploited. In other words, we are interested in solutions $\mathbf{x}$ with only a few large entries, say, up to k. If the other entries are small, we refer to such vector as *nearly k-sparse*, and if they are exactly zero we call it *exactly k-sparse*. Clearly, if vector $\mathbf{x}$ is exactly k-sparse, it is also nearly k-sparse. For the sake of simplicity, we use the terms *nearly k-sparse* and *k-sparse* interchangeably. If we want to emphasize that the sparsity is exact, we use the term *exactly k-sparse*. A network is called k-identifiable if for every *exactly k-sparse delay vector* $\mathbf{x}$, Eq. (2.1) is uniquely solvable.

This work applies the concepts of *expander graphs* and compressed sensing [101, 91, 24, 49] to the estimation of k-sparse delay vectors. This is achieved by fundamentally relating the network routing matrix to a bipartite graph. If the bipartite graph is an *expander graph*, then one can use l_1 minimization to solve Eq. (2.1), that has polynomial complexity in n and is independent of k [13].

In this section, we first establish a novel connection between network delay tomography and binary compressed sensing via the notion of expander graphs. Next, for 1-identifiable networks, we relax the existing result for expansion from $\epsilon \leq 1/6$ to $\epsilon \leq 1/4$ (Lemma 2). Furthermore, we remove the left d-regularity condition in expander graphs (Theorem 10). Then we generalize our result to k-identifiable networks (Theorem 12) and show that the l_1 minimization can be applied to estimate links delay in a network with arbitrary number of congested links. Simulation results indicate that these relaxations increase the number of networks that satisfy the expansion property by 30% compared to the existing conditions. Finally, we provide simulation evidence of applying the proposed method to delay estimation when the delay vector $\mathbf{x}$ is k-sparse for $k \geq 1$ and show that it outperforms the state-of-the-art algorithm in the literature.

[8] A congested link is one with a significantly elevated delay compared to the rest of the links in the network.

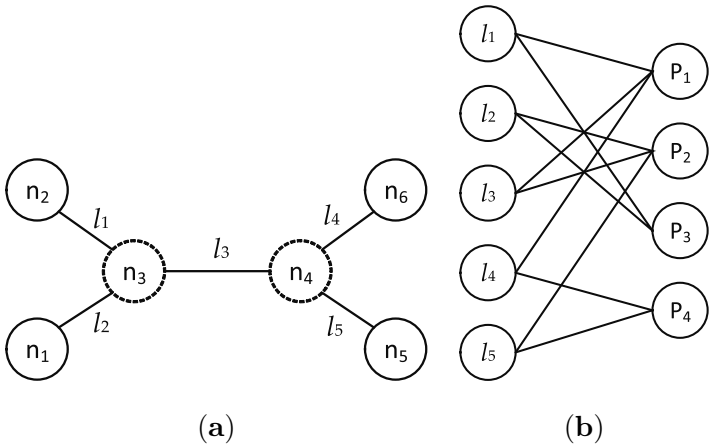


Figure 2.10: (a) A network with 4 boundary nodes, 2 intermediate nodes and 5 links, (b) Bipartite graph corresponding to given routing matrix in Eq. (2.37)

2.2.1 Routing Matrix and Bipartite Graph

As is customary, a network consisting of bidirectional links connecting transmitters, switches, and receivers can be modeled as an undirected graph $N(V, E)$, where V (E) is the set of vertices (edges). Throughout this manuscript, boundary nodes are depicted as solid circles, while intermediate nodes are presented using dashed circles. We use network depicted in Figure 2.10-(a) to illustrate the subsequent definitions.

In this section, we show that the routing matrix of any network can be represented as a *bi-adjacency matrix* of a suitably defined *bipartite* graph. This will help connect the problem of network identifiability with *expander graphs*, a special subset of bipartite graphs.

A bipartite graph is one whose vertices can be divided into two disjoint sets, X and Y, so that every edge connects a vertex in X to one in Y [37]. It is usually represented as a triple $G(X, Y, H)$, where $H \subset X \times Y$ is a set with paired elements from X and Y. The vertex sets X and Y are the left and right sides of the graph, respectively. A bipartite graph $G(X, Y, H)$ can be represented by its *bi-adjacency* matrix $\mathbf{A} = [a_{ij}]$, where $a_{ij} = 1$ if node

$i \in Y$ is connected to node $j \in X$, and is zero otherwise, i.e.,

$$\begin{aligned} a_{ij} &= \begin{cases} 1 & (j,i) \in H \\ 0 & (j,i) \notin H \end{cases}, \\ \mathbf{A} &= [a_{ij}]. \end{aligned} \tag{2.36}$$

In the definition of the bi-adjacency matrix $\mathbf{A}$, rows of $\mathbf{A}$ correspond to Y, which is the right-hand side of the graph; columns of $\mathbf{A}$ correspond to X, which is the left-hand side of the graph. This convention is used throughout the paper.

Assume that a given network $N(V, E)$ has a total of n links (i.e., $n = |E|$), and $\mathcal{R}$ is the (given) set of paths between the boundary nodes of the network and $r = |\mathcal{R}|$. Let $\mathbf{R}_{r \times n}$ denote the routing matrix corresponding to the set $\mathcal{R}$. For example, for the 1-identifiable network in Figure 2.10-(a), suppose the following routing matrix is given:

$$\mathbf{R} = \begin{array}{l} \\ P_1 : n_2 \rightsquigarrow n_6 \\ P_2 : n_1 \rightsquigarrow n_5 \\ P_3 : n_1 \rightsquigarrow n_2 \\ P_4 : n_5 \rightsquigarrow n_6 \end{array} \begin{bmatrix} l_1 & l_2 & l_3 & l_4 & l_5 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \tag{2.37}$$

which is equivalent to the following set of paths $\mathcal{R}$:

$$\mathcal{R} = \{l_1 l_3 l_4, l_2 l_3 l_5, l_1 l_2, l_4 l_5\}. \tag{2.38}$$

$\mathbf{R}_{r \times n}$ can be viewed as a bi-adjacency matrix of a bipartite graph $G(X, Y, H)$, where $X = E$ (set of links in the network) and $Y = \mathcal{R}$ (set of given paths in the network). There exists a connection between a node in X and a node in Y if a path in $\mathcal{R}$ includes the corresponding link in E. Figure 2.10-(b) presents the bipartite graph for the network in Figure 2.10-(a) with the routing matrix $\mathbf{R}$ in Eq. (2.37).

Note that the above routing matrix, or its equivalent set of paths, is not a complete set of routes for the network in Figure 2.10-(a) (e.g., it does not include the path from n_1 to n_6, which is $l_2 l_3 l_4$). However, it is typically a fundamental premise in network tomography that the routing matrix is already chosen and may not be changed. Hence, we initially seek

to investigate the following question: Assuming that the routing matrix is given, when is it possible to identify or estimate link delays?

2.2.2 Expander Graphs and Network Identifiability

In recent years, a new approach–*Compressed Sensing*–for estimating an n-dimensional (signal) vector $\mathbf{x}$ from a lower-dimensional representation has attracted much attention [49, 25, 24]. For any signal $\mathbf{x} \in \mathbb{R}^n$, the reduced dimension representation is equal to $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $m \times n$ matrix $\mathbf{A}$ ($m << n$) is referred to as the *measurement matrix*. The main challenge in traditional compressed sensing is to construct $\mathbf{A}$ with the following desirable (and conflicting) properties: (a) achieve maximum possible compression (m/n small) and yet allow (b) an accurate reconstruction of $\mathbf{x}$ from $\mathbf{y}$ when $\mathbf{x}$ is known to be sparse using (c) a fast decoding algorithm [161, 71, 26, 172].

As discussed above, the routing matrix of a network is the measurement matrix for delay tomography application, and in most scenarios it is predetermined. The main issue, therefore, is to determine whether it is an *appropriate* measurement matrix for compressed sensing, i.e., if it satisfies objective (b) above. In the simulation section, we show that the existing conditions for the measurement matrix, $\mathbf{A}$, do *not* apply to most of the routing matrices. Motivated by this observation, we aim to revisit these conditions and modify them so that they become more suitable to the network tomography problem. Then, we use linear program (LP) optimization to solve Eq. (2.1).

2.2.3 Expander Graphs

Definition 1 *A bipartite graph $G(X, Y, H)$ with a left degree d (i.e., $deg(v) = d \; \forall \, v \in X$) is a $(\phi, d, \epsilon) - expander$ if for any $\Phi \subset X$ with $|\Phi| \leq \phi$, the following condition holds:*

$$|N(\Phi)| \geq (1 - \epsilon) d \, |\Phi|, \tag{2.39}$$

where $N(\Phi)$ is a set of neighbors of Φ [9]*. ϕ and ϵ are the "expansion factor" and the "error parameter," respectively.*

[9] Neighbors of Φ are nodes which are connected to at least one of the nodes in Φ.

Roughly speaking, in an expander graph, the degree of connectivity for a collection of nodes (with cardinality of up to ϕ) on the left-hand side (X) expands by that factor on the right-hand side (Y) [142]. Expander graphs are well-studied; in a key result, Berinde and Indyk in [12, 13] show that the bi-adjacency matrix of a $(2\phi, d, \epsilon) - expander$ graph can be used as the measurement matrix for a ϕ-sparse signal, for $\epsilon < \frac{1}{6}$.

The parameter ϵ in an expander graph is a design variable that is related to recovery error. The existing results require $\epsilon < 1/6$, which, as we will show, does not apply to most of the networks . In a network tomography problem, the measurement matrix is pre-determined, so we need to enlarge the bound on ϵ as high as possible to increase the likelihood that it leads to an identifiable network.

The bipartite graph given in Figure 2.10-(b) coordinates with the 1-identifiable network in Figure 2.10-(a) with the routing matrix in Eq. (2.37). It is easy to see that this bipartite graph is an expander for $\epsilon = 1/4$. Motivated by this example, we relax the existing result for $\epsilon < 1/6$ to $\epsilon \leq 1/4$. In the simulation results (Section 2.3.2), we show that this relaxation increases the number of k-identifiable networks that satisfy the expansion property by 30%. In other words, for more than 30% of k-identifiable networks, we have $1/6 \leq \epsilon \leq 1/4$. For networks that satisfy the expansion property, LP optimization can be used to solve the tomography problem. The analytical results are first derived for 1-identifiable networks because it is easier to give intuitive explanation. Then we generalized the result to k-identifiable networks with arbitrary k.

2.2.4 1-Identifiability

Consider a network $N(V, E)$ with a routing matrix $\mathbf{R}$ that satisfies the expansion property of a $(2, d, \epsilon)$-expander graph. Now, let us consider two links that are in fact two nodes on the left-hand side of bipartite graph $G(E, \mathcal{R}, H)$; i.e., $|\Phi| = 2$. As indicated by the definition of expander graphs in Eq. (2.39), the number of nodes on the right-hand side connected to these two nodes is $2d(1 - \epsilon)$. If the two nodes are connected to exactly the same nodes on the right-hand side, there is no way to distinguish between them because of the symmetry in their connection. That is, if the two nodes are connected to the same nodes, then bi-

adjacency matrix $\mathbf{R}$ is rank deficient. To identify the correct congested link, the paths that pass through these two links must be greater than d. Thus, $2d(1-\epsilon) \geq d+1$, which must be valid for any d. For $d=2$ we obtain $\epsilon \leq \frac{1}{4}$. Here, we exclude case of $d=1$, since each left 1-regular bipartite graph with $N(\Phi) \geq 2$ is an expander graph.

Relaxing the bound on ϵ

Followed by above intution, Lemma 2 provides an upper bound on the error of recovering $\mathbf{x}$ from its lower-dimensional projection $\mathbf{Ax}$ when $\mathbf{A}$ is a bi-adjacency matrix of a $(2, d, \epsilon)$-expander graph and $\epsilon \leq 1/4$.

lemma 2 *Let $\mathbf{A}$ be a bi-adjacency matrix of a $(2, d, \epsilon)$-expander graph with $\epsilon \leq 1/4$. Consider any two vectors, $\mathbf{x}$ and $\mathbf{x}'$, with the same projection under the measurement matrix $\mathbf{A}$, i.e., $\mathbf{Ax} = \mathbf{Ax}'$. Assume that $\mathbf{x}$ is 1-sparse. Further, without loss of generality, suppose that $\| \mathbf{x}' \|_1 \leq \| \mathbf{x} \|_1$. Let S be the singleton set of the largest (in magnitude) element of $\mathbf{x}$. Then,*

$$\| \mathbf{x}' - \mathbf{x} \|_1 \leq f(\epsilon) \| \mathbf{x}_{S^c} \|_1, \tag{2.40}$$

where

$$f(\epsilon) = \frac{2(1+2\epsilon)}{1-2\epsilon}, \quad \epsilon \leq \frac{1}{4}^{10}. \tag{2.41}$$

Proof: See Appendix.

The results from the previous lemma show that under some conditions, the link delay in a network may be estimated as the unique solution to Eq. (2.1). The following theorem relates the problem of delay estimation in a network $N(V, E)$ to results on expander graphs with $\epsilon \leq 1/4$ and shows that Eq. (2.1) can be solved for $\mathbf{x}$ using LP optimization.

Theorem 9 *Let $N(V, E)$ be a network with a set of paths $\mathcal{R}$ and a corresponding routing matrix $\mathbf{R}_{r \times n}$. Suppose that $G(E, \mathcal{R}, H)$ is a bipartite graph with bi-adjacency matrix $\mathbf{R}$.*

[10] In [12, 13], this function is derived as $f(\epsilon) = \frac{2(1-2\epsilon)}{1-6\epsilon}$ and hence it requires $\epsilon < \frac{1}{6}$.

Assume that $\mathbf{x}$ *is the true (unknown) delay vector of* $N(V, E)$ *and* $\mathbf{y} = \mathbf{R}\mathbf{x}$ *is the (given) end-to-end delay measurement. Let* $\mathbf{x}'$ *be a solution to the following LP optimization:*

$$\begin{aligned} &\min \parallel \mathbf{x}' \parallel_1 \\ &s.t. \\ &\mathbf{R}\mathbf{x}' = \mathbf{y}. \end{aligned} \tag{2.42}$$

Then,

$$\parallel \mathbf{x} - \mathbf{x}' \parallel_1 \leq f(\epsilon) \parallel \mathbf{x}_{S^c} \parallel_1, \tag{2.43}$$

if G *is a* $(2, d, \epsilon)$*-expander with* $\epsilon \leq \frac{1}{4}$.

Proof: See Appendix.

Note that if the true delay vector $\mathbf{x}$ is exactly 1-sparse (which almost never happens in practice because links always have nonzero delay), it implies that $\parallel \mathbf{x}_{S^c} \parallel_1 = 0$, which means that $\mathbf{x}' = \mathbf{x}$; i.e., l_1-norm minimization in Eq. (2.42) can recover $\mathbf{x}$ with zero estimation error. In other words, if the delay of all links in the network is zero except for maybe one link, the delay of that link can be exactly recovered from the end-to-end delay measurement. However, if the true delay vector contains links with small but nonzero delays (the more likely scenario), the estimation error is not zero and the above theorem yields an upper bound.

Relaxing d-regularity Condition

One of the conditions for expander graphs is d-regularity on the left-hand side. However, there exists some networks, $N(V, E)$, which are 1-identifiable, but their corresponding bipartite graphs are not d-regular. Hence, the result of Theorem 9 does not apply. An example of such a network is depicted in Figure 2.11-(a) with the following routing matrix:

$$\mathbf{R} = \begin{matrix} \\ P_1 : n_1 \rightsquigarrow n_2 \\ P_2 : n_1 \rightsquigarrow n_3 \\ P_3 : n_2 \rightsquigarrow n_3 \end{matrix} \begin{bmatrix} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}. \tag{2.44}$$

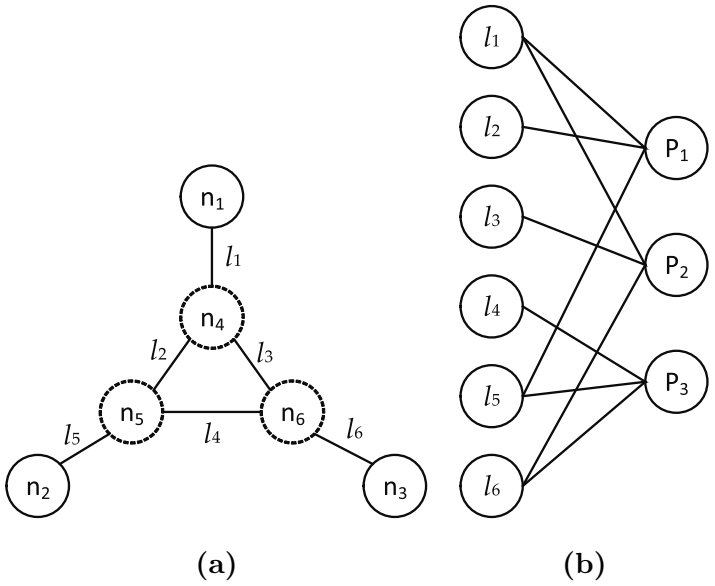


Figure 2.11: An example of 1-identifiable network whose corresponding bipartite graph is not regular on the left side: (a) Network topology (b) Bipartite graph corresponding to the routing matrix $\mathbf{R}$ in Eq. (2.44)

The above routing matrix is a bi-adjacency matrix of the bipartite graph presented in Figure 2.11-(b). This bipartite graph is not regular in the left side because the degree of a node in the left set is either 1 or 2; hence, it cannot be an expander. However, Figures 2.12-(a) and (b), respectively, represent subgraphs of G with regular left degree 1 and 2; these subgraphs are expander graphs. The above observation convinces us that the result in Theorem 9 is extendable for networks whose corresponding bipartite graph is not regular (and is therefore not an expander) but can be partitioned into subgraphs that are expander graphs.

Theorem 10 *Let $N(V, E)$ be a network with routing matrix $\mathbf{R}_{r\times n}$. Let $G(X, Y, H)$ be a bipartite graph with bi-adjacency matrix $\mathbf{R}$. Suppose that $G_i(X_i, Y, H_i)$, $i = 1, 2, ...M$, are d_i-regular bipartite subgraphs of G such that*

- $X = \cup X_i$ *and* $X_i \cap X_j = \emptyset$ *for* $i \neq j$
- $H = \cup H_i$

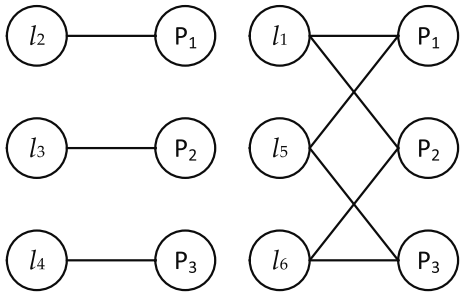


Figure 2.12: Two subgraphs of the bipartite graph in Figure 2.11-b which are regular on their left side

- $d_i \neq d_j$ *for* $i \neq j$

Then, $N(V,E)$ *is* 1*-identifiable if* G_i *is a* $(2, d_i, \epsilon)$*-expander graph with* $\epsilon \leq \frac{1}{4}$*,* $i = 1, 2, \ldots, M$*. Further, the link delay is the solution to the LP optimization in Eq.* (2.42)*.*

Proof: See Appendix.

For future reference, we refer to the conditions in Theorem 10 as *1-identifiability expansion conditions*. If the network $N(V,E)$ satisfies these conditions, we refer to it as the 1-*identifiable expander network*.

Note that the 1-identifiable expansion conditions in Theorem 10 imply the following for any link pair l_i and l_j:

- They belong to different G_i's and hence have different degrees $d_i \neq d_j$.
- They belong to the same subgraph G_i, i.e., $d_i = d_j$. In that case, because G_i is a bipartite graph, they satisfy the expansion property in Eq. (2.39).

We state this observation formally in the following corollary.

Corollary 11 *Let* $N(V,E)$ *be a 1-identifiable expander network with the routing matrix* $\mathbf{R}_{r\times n}$ *and a set of paths* $\mathcal{R}$*. Let* $G(E, \mathcal{R}, H)$ *be its corresponding bipartite graph with the*

bi-adjacency matrix $\mathbf{R}$*. Then, one and only one of the following statements is true for any two links* l_i *and* l_j *in* E*,* $i \neq j$*:*

- $deg(l_i) > deg(l_j)$
- $deg(l_i) < deg(l_j)$
- $deg(l_i) = deg(l_j) = d$ *and* $2d - 4deg(l_i, l_j) \geq 0$

where $deg(l_i, l_j)$ is defined as the number of nodes connected to both l_i and l_j in the bipartite graph $G(E, \mathcal{R}, H)$[11].

proof: See Appendix.

2.2.5 k-identifiability

In this subsection, we extend our results to general k-identifiable networks. We first define k-identifiable expander networks. Then we show that for any k-identifiable network with given routing matrix $\mathbf{R}$, its k-sparse delay vector $\mathbf{x}$ can be recovered from end-to-end measurement $\mathbf{y} = \mathbf{Rx}$ with LP optimization in Eq. (2.42) and the average estimation error (over all combinations of up to k congested links in the network) is bounded from above.

Definition 2 *A* k*-identifiable expander network* $N(V, E)$ *is a network whose routing matrix* $\mathbf{R}$ *is the bi-adjacency matrix of a bipartite graph* $G(X, Y, H)$ *consisting of* d_i*-regular subgraphs* $G(X_i, Y, H_i)$ *with the following properties*

- $X = \cup X_i$ *and* $X_i \cap X_j = \emptyset$ *for* $i \neq j$
- $H = \cup H_i$
- $d_i \neq d_j$ *for* $i \neq j$
- $G(X_i, Y, H_i)$ *is a* $(2k, d_i, \epsilon)$*-expander with* $\epsilon \leq \frac{1}{4}$

[11] Equivalently, $deg(l_i, l_j)$ denotes number of paths going through both links l_i and l_j.

As defined before, for k-sparse delay vector $\mathbf{x}$ there are up to k large entries in $\mathbf{x}$ (representing the congested links in the network) and the rest are close to zero (uncongested links). Let define random variable $\boldsymbol{\mathcal{S}}$ as the set of indexes of large entries in $\mathbf{x}$. Hence, we have $\boldsymbol{\mathcal{S}} \subset \{1, 2, \ldots, n\}$, $|\boldsymbol{\mathcal{S}}| \leq k$ and $\boldsymbol{\mathcal{S}}^c = \{1, 2, \ldots, n\} - \boldsymbol{\mathcal{S}}$. In other words, the random variable $\boldsymbol{\mathcal{S}}$ represents congested links in the network $N(V, E)$[12]. The probability distribution of $\boldsymbol{\mathcal{S}}$ can easily be derived by having the congestion probability of the links and assuming independency among links' congestion events. The following theorem gives the expected estimation error when l_1 optimization is used to recover link delays in a k-identifiable network.

Theorem 12 *Let $N(V, E)$ be a k-identifiable expander network with a set of paths $\mathcal{R}$ and a corresponding routing matrix $\mathbf{R}_{r\times n}$. Assume that $\mathbf{x}$ is the true (unknown) delay vector of $N(V, E)$ and $\mathbf{y} = \mathbf{R}\mathbf{x}$ is the end-to-end delay measurement. Moreover, let $\boldsymbol{\mathcal{S}}$ ($|\boldsymbol{\mathcal{S}}| \leq k$) be the set of indexes of the largest entries in $\mathbf{x}$. If $\mathbf{x}'$ is the solution to the LP optimization in Eq.* (2.42)*, then*

$$\mathbb{E}_{\boldsymbol{\mathcal{S}}}[\| \mathbf{x} - \mathbf{x}' \|_1] \leq f(\epsilon)\mathbb{E}_{\boldsymbol{\mathcal{S}}}[\| \mathbf{x}_{\boldsymbol{\mathcal{S}}^c} \|_1], \tag{2.45}$$

where expectation is over the distribution of the congested links inside the network.

Proof: See Appendix.

2.3 Evaluation Results

In Section 2.2.2, Theorem 12, we showed that if the routing matrix of a network is the bi-adjacency matrix of the union of disjoint expander graphs, that network is k-identifiable. Moreover, we can estimate internal link delay using an LP optimizer in Eq. (2.42). However, a legitimate 'big-picture' question arises: How many networks actually satisfy the conditions of Theorem 12; i.e., how many are k-identifiable expanders? In this section, we generate random Internet-type networks to study this question. Our simulation results show that our relaxation increases number of networks which satisfy expansion property by almost 30%.

[12] $\boldsymbol{\mathcal{S}}^c$ represents the uncongested links in the network

For those networks that are k-identifiable expander–i.e., their routing matrix satisfies the condition in Definition 2–we determine the average normalized estimation error when there is k congested link in the network and show that the average normalized estimation error remains within an acceptable range. Next, we compare our algorithm with a recently developed delay tomography algorithm and show that the proposed algorithm yields a lower estimation error.

Finally, some network monitoring applications need to locate congested links in the network. We show that if the network is k-identifiable, the proposed LP-optimization algorithm can be employed as a congestion detection tool with close to ideal detection performance.

2.3.1 Generation of Networks with Random Topology

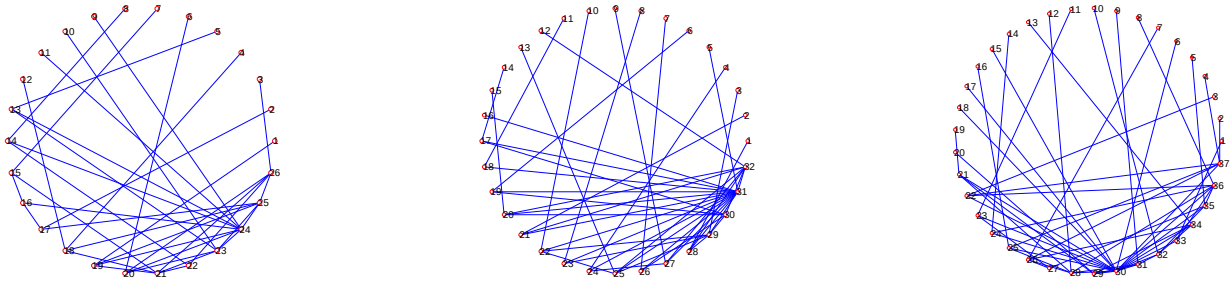
We use Inet version 3.0 [96, 165]–an Internet topology generator software (at AS[13] level)–to generate random graphs with the given power law and a fixed number of boundary nodes [14]. We create networks containing 5000 nodes with 5, 8, 10, 12, 16, and 20 boundary nodes, respectively. The output of Inet, which contains the set of neighbors of each node in the generated graph, is fed to matgraph toolbox in MATLAB [147] for modification. We first create a routing matrix containing the shortest paths between any boundary node pairs in the network. Then we delete all nodes and links that do not contribute to any of the above paths, since if a link is not covered by any end-to-end path, it is not identifiable. The remaining networks constitute our random set. In Figure 2.13, three examples of random networks are depicted.

It is worth pointing out that in Internet topology-based networks, a relationship exists between the number of boundary nodes (number of nodes with degree 1) and the number of links in a network (often referred to as the network size) [**?**]. The literature shows that the degree of nodes in Internet topology has a power law distribution. The majority of Internet topology generation softwares, including Inet 3.0, consider the number of boundary nodes as an input argument [165].

---

[13] Autonomous System.

[14] Boundary nodes are nodes with degree one which act as injection points for probes in our problem.



(**a**) (**b**) (**c**)

Figure 2.13: Output of Inet after our modification in MATLAB with (a) 12 (b) 16 (c) 20 boundary nodes. Nodes with degree 1 represents injection nodes

2.3.2 Networks and Expansion Property

For the routing matrices of these random networks, we first examine how many of them satisfy the k-identifiability expansion conditions in Definition 2. For the network with a fixed number of boundary nodes, fifty different topologies are created. Table 2.5 shows the percentage of them that satisfy the k-identifiability expansion property for $k = 1, 2, 3$.

To show the impact of our relaxation of ϵ in Lemma 2 and Theorem 12, in Table 2.5, we also provide the percentage of networks that are k-identifiable, using $\epsilon < 1/6$. As one can see, by moving the bound on ϵ from $1/6$ to $1/4$, the number of networks satisfying the expansion property increases by almost 30%. In other words, 30% of k-identifiable networks are within $1/6 \leq \epsilon \leq 1/4$.

2.3.3 Delay Estimation: Simulation Experiments

Theorem 12 says that if the routing matrix of a network satisfies k-identifiable expander conditions, then link delays in the network can be estimated using Eq. (2.42). To examine the accuracy of the proposed delay estimation method, for each network created in Section 2.3.1, we calculate the average normalized estimation error for all links as follows. k reference links are randomly selected and assigned a delay of 10 ms to denote congestion, $k = 1, 2, 3$. All other links in the network are assumed to experience i.i.d exponentially distributed

Table 2.5: For networks with fixed number of boundary nodes, comparing the percentage of which are k-identifiable expanders for $k = 1, 2, 3$ with $\epsilon \leq \frac{1}{4}$ and $\epsilon < \frac{1}{6}$

		12	16	20	25	30
$k = 1$	$\epsilon \leq \frac{1}{4}$	72%	74%	72%	70%	72%
	$\epsilon < \frac{1}{6}$	40%	32%	30%	36%	32%
$k = 2$	$\epsilon \leq \frac{1}{4}$	56%	50%	50%	54%	52%
	$\epsilon < \frac{1}{6}$	20%	24%	22%	22%	26%
$k = 3$	$\epsilon \leq \frac{1}{4}$	56%	50%	46%	52%	52%
	$\epsilon < \frac{1}{6}$	0%	0%	22%	20%	24%

delays with average μ, i.e.,

$$f_l(t) = \frac{1}{\mu} exp(-\frac{t}{\mu}) \quad \forall l \in E, \tag{2.46}$$

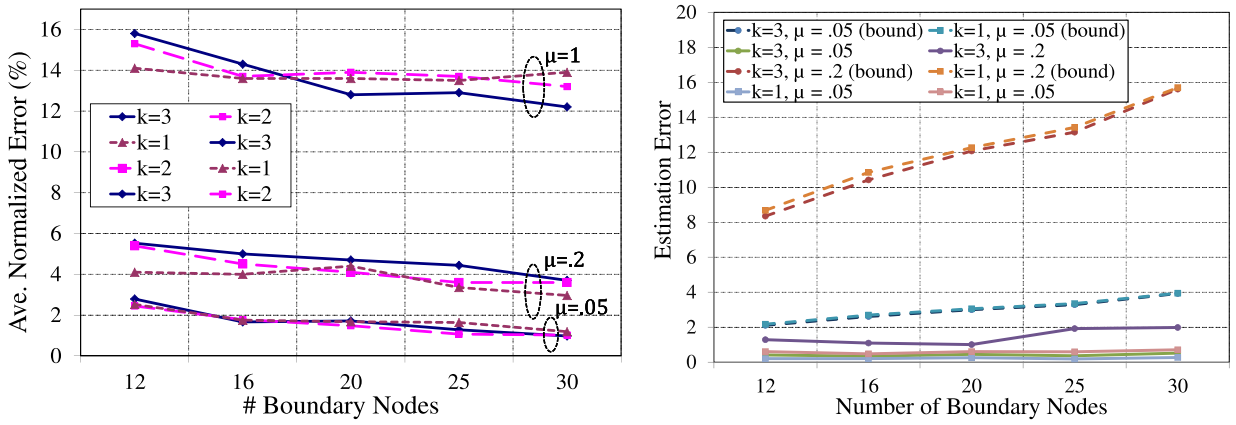
where $f_l(t)$ is the delay for link l and $\mu \in [0, 1]$ to denote that these links do not undergo congestion.

We exploit the proposed LP optimization in Eq. (2.42) to estimate link delays. For the network, the normalized estimation error for each congested link inside the network is calculated as follows:

$$norm.\ err = \frac{\| \mathbf{x} - \hat{\mathbf{x}} \|_2}{\| \mathbf{x} \|_2}, \tag{2.47}$$

where $\mathbf{x}$ and $\hat{\mathbf{x}}$ are the true and estimated delay vectors respectively.

Figure 2.14-(a) presents the average normalized estimation error when there are k congested links inside the network for $k = 1, 2, 3$ and LP optimization Eq. (2.42) is used to estimate the delay. As expected, the average normalized estimation error for different μ (vector $\mathbf{x}$ is nearly k-sparse) mimics the expected trend from Eq. (2.45); i.e., as μ decreases (the average delay of uncongested links goes to zero) the recovery error of the proposed algorithm decrease. This phenomena also can be seen in Figure 2.14-(b) which presents the the normalized estimation error upper bound. From Eq. (2.43) and Eq. (2.45), the upper bound depends only on delay values of uncongested links and it goes to zero for an ideal network (i.e. zero delay for uncongested links).



(**a**) (**b**)

Figure 2.14: (a) Average normalized estimation error in networks satisfying conditions given in Definition 2 when there are k deficient link within the network for different average delay μ in Eq. (2.46) (b) Comparing the simulation estimation error and the derived error upper bound in Eq. (2.45)

To evaluate the algorithm for large networks, we generate networks with 50 and 100 boundary nodes by implementing the process explained at the beginning of this section. For each number of boundary nodes, N = 50 and N = 100, 50 different network samples are generated. The average numbers of links in these samples are around 250 and 350 links, respectively. Figure 2.15 presents the average normalized estimation error for delay recovery, as derived using the proposed algorithm, where $k\%$ of the links are congested. The process of assigning delay to congested and uncongested links is similar to the previous scenario.

It worth mentioning that, for a fix value of μ, as the number of congested links are increasing the value of $\| \mathbf{x} \|_2$ is also increasing. The results in Figure 2.15 shows that the estimation error is not increasing as fast as the value of $\| \mathbf{x} \|_2$. Hence, for a fix value of μ, the normalized estimation error have a decreasing trend as the the number of congested links are increasing while it satisfied the sparsity condition $k \ll |E|$. For the same reason, for fix μ, normalized estimation error for bigger network, $N = 100$, is slightly less than normalized estimation error with $N = 50$ boundary nodes.

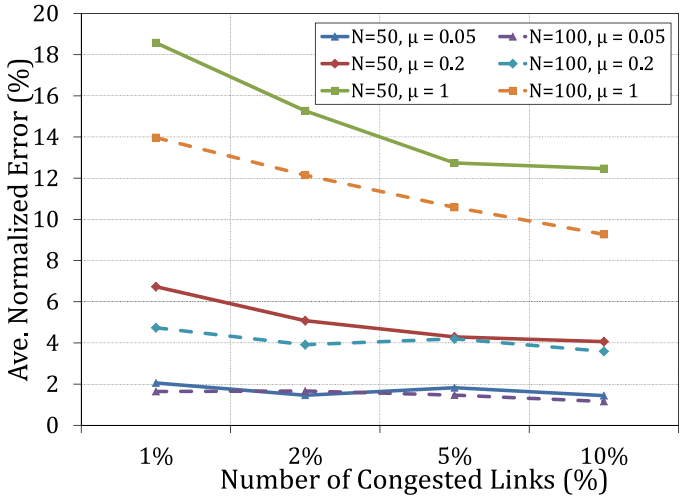


Figure 2.15: Average normalized estimation error in large networks with $N = 50$ and $N = 100$ boundary nodes. Networks are satisfying conditions given in Definition 2 and $k\%$ of the links are deficient.

Finally, we aim to evaluate the proposed algorithm as a detection tool. We study the effectiveness of LP optimization in classifying links as congested and uncongested for k-identifiable networks. To this end, we compare the delay caused by each link with a threshold. Links with delays greater than the threshold are categorized as congested, and vice versa. That is, we consider the simple detection rule:

$$\forall l \in E, \text{if } x_l \geq \tau \Rightarrow l \text{ is congested} \tag{2.48}$$

Figure 2.16 presents the receiver operating characteristic (ROC) for the above-mentioned detection rules, where τ changes from 0 to infinity. For $\mu = 1$, the ROC plot is very close to the ideal receiver, indicating that the LP optimization in Eq. (2.42) can distinguish between congested and uncongested links. An interesting observation in Figure 2.16 is that detection performance depends only on the average delay (μ) of uncongested links and is independent of network size and number of congested links k as long as the sparsity condition is satisfied, i.e., $k \ll |E|$.

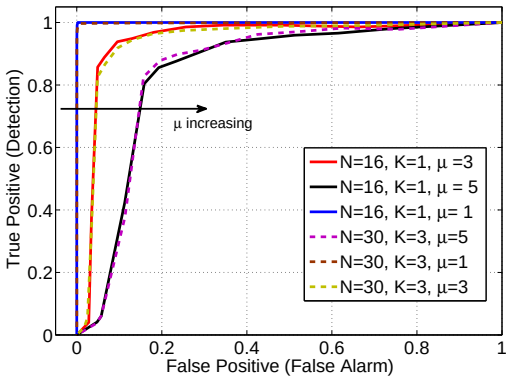


Figure 2.16: Receiver Operating Characteristic for the proposed algorithm.

2.3.4 Delay Estimation: Cumulative Distribution Function

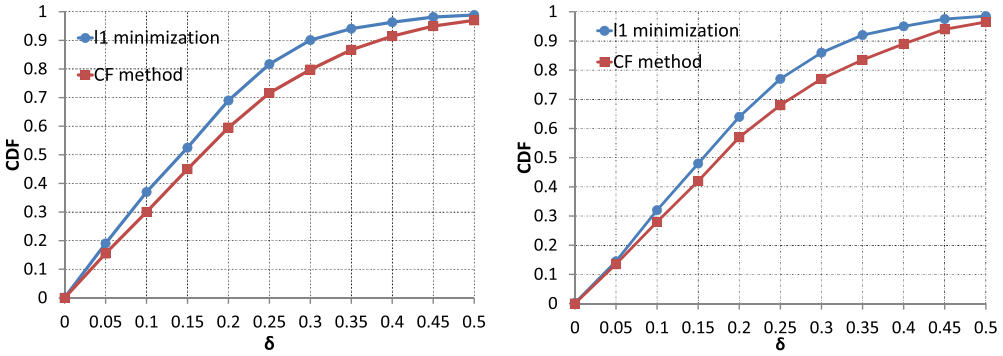
In this section, we compare our results with those produced by CF-estimator, one of the recent and state-of-the-art network delay estimators proposed in [32]. In the study, the authors proposed a mixture model of characteristic functions for delay matrix $\mathbf{x}$ and developed a fast estimation algorithm based on generalized method of moments (GMM). The authors claim that this approach enables the use of more flexible models of heterogeneous network link delays, wherein the delays are nondiscrete and may have different scales across all network links. We provide the cumulative probability of the normalized estimation error using both methods and show that the proposed method provides less probability of error.

Let $\mathbf{x}$ be the actual delay of the links and $\hat{\mathbf{x}}$ be the output of the delay estimator. We aim to calculate the following CDF:

$$P(\frac{\| \mathbf{x} - \hat{\mathbf{x}} \|_2}{\| \mathbf{x} \|_2} \leq \delta). \tag{2.49}$$

Figure 2.17 presents the CDF of the normalized estimation error when there are k congested links inside the network for $k = 1, 3$ and $\delta \in [0, .5]$. For each CDF, 200 random networks[15] are generated and for each generated network, link delays are assigned as we describe in Section 2.3.3. **As one can see, the proposed algorithm outperforms the CF algorithm due to the fact that it uses sparsity information.**

[15] $\mu \in \{0, .1, .2, 1\}$.



(**a**) (**c**)

Figure 2.17: Cumulative distribution function of estimation error for (a)$k = 1$ (b)$k = 3$. The proposed algorithm outperforms CF method.

As one can see the proposed algorithm outperforms CF algorithm. The reason is that it uses sparsity as its side information. It is worth mentioning again that to provide QoS in a network, either it is for multimedia streaming, gaming or any other live application, links with high delays are more important. Hence the l_1 minimization given in Eq. (2.42) focus on finding the k links with the highest delays and it results in a better estimation.

2.4 Designing Routing matrix Using Compressed Sensing

Clearly, the choice of paths in the network used for probing directly affects the estimation accuracy but also constitutes an overhead. For example, for a given estimation accuracy, one could attempt to minimize total number of probes used for network monitoring. We study this problem and arrive at a binary integer programming formulation. Given that this is an NP-hard problem, a heuristic algorithm is developed which results in good approximate results with reasonable complexity.

2.4.1 Minimum Path Set Selection

As an example, consider the network in Figure 2.10, which has six paths between boundary nodes, as depicted in Figure 2.18. These are collected in the following set.

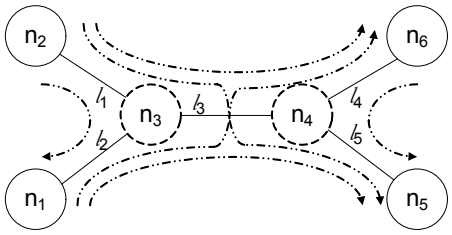


Figure 2.18: Paths between boundary nodes of network in Figure 2.10

$$\mathcal{R} = \{l_1 l_3 l_4, l_1 l_3 l_5, l_2 l_3 l_4, l_2 l_3 l_5, l_1 l_2, l_4 l_5\}. \tag{2.50}$$

However, only four of the above are sufficient for the delay identifiability of any single link. In general, not all paths between boundary nodes are necessary for identifiability. The next subsection shows how to select the set of paths from $\mathcal{R}$ so that a desired objective is achieved.

2.4.2 Network Covering

Consider a network $N(V, E)$ with a collection of end-to-end paths $\mathcal{R}$. We seek the *minimum number of paths between boundary nodes so that each link belongs to at least one path.* This suffices to determine whether the network contains *any* bad link. This problem–known as "network covering"–has been studied in literature [131, 11]. In Section 2.4.3, we present a new algorithmic formulation that not only locates but also estimates the delay value of the congested link with minimum number of paths.

To formulate an optimization problem, we define a variable $I_{\mathcal{R}_i}$, which indicates whether path $\mathcal{R}_i \in \mathcal{R}$ is included or not:

$$I_{\mathcal{R}_i} = \begin{cases} 1 & \mathcal{R}_i \text{ is used} \\ 0 & \text{o.w.} \end{cases}. \tag{2.51}$$

We seek to minimize the number of paths, i.e.,

$$\min \sum_{i=1}^{|\mathcal{R}|} I_{\mathcal{R}_i}, \tag{2.52}$$

over all binary variables $I_{\mathcal{R}_i}$ subject to the fact that each link belongs to at least one path. Let $\mathbf{I}_{\mathcal{R}}^t = [I_{\mathcal{R}_i}]_1^{|\mathcal{R}|}$ be the vector of path indicators. The ith entry of $\mathbf{R}^t\mathbf{I}_{\mathcal{R}}$ is the number of paths that traverse the ith link, each of which should be equal to or greater than 1. We therefore seek the following:

$$\begin{aligned} &\min \sum_{i=1}^{|\mathcal{R}|} I_{\mathcal{R}_i} \\ &\text{s.t.} \\ &\mathbf{R}^t\mathbf{I}_{\mathcal{R}} \geq 1 \\ &I_{\mathcal{R}_i} \in \{0,1\}. \end{aligned} \tag{2.53}$$

The above is a *binary integer programming* that is well-known to be NP-hard. A number of algorithms provide good approximate solutions [138, 17] when the original problem can be "relaxed."

2.4.3 Minimum Path Set for 1-identifiability

Among all available paths in the network, $\mathcal{R}$, what is the subset with minimum cardinality that guarantees the 1-identifiability of the network? To achieve a network that is $k = 1$-identifiable, it is necessary that our minimum set "cover" the network, which means that $\mathbf{R}^t\mathbf{I}_{\mathcal{R}} \geq 1$, as discussed in the previous section. For identifiability, we used Corollary 11 that specifies conditions for any two links $l_i, l_j \in E$ to have the network 1-identifiable. Using the path indicator functions in Eq. (2.51), these conditions can be rewritten as follows:

$$\begin{aligned} &\bullet \quad \sum_{\{\mathcal{R}_k : l_i \in \mathcal{R}_k\}} I_{\mathcal{R}_k} - \sum_{\{\mathcal{R}_k : l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} \geq 1 \\ &\bullet \quad \sum_{\{\mathcal{R}_k : l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} - \sum_{\{\mathcal{R}_k : l_i \in \mathcal{R}_k\}} I_{\mathcal{R}_k} \geq 1 \\ &\bullet \quad \sum_{\{\mathcal{R}_k : l_i \in \mathcal{R}_k\}} I_{\mathcal{R}_k} + \sum_{\{\mathcal{R}_k : l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} - 4 \sum_{\{\mathcal{R}_k : l_i, l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} \geq 0 \end{aligned} \tag{2.54}$$

As mentioned, for any two links l_i and l_j, one and only one of the above inequalities must be satisfied. Hence, we use the notion of *alternative constraints*–a well-known trick in linear binary programming [17]. To that end, we introduce three binary variables $y_{1ij}, y_{2ij}, y_{3ij}$ for any two links l_i and l_j, $i \neq j$:

$$y_{kij} = \begin{cases} 1 & \text{if the } k\text{-th constraint is satisfied} \\ 0 & otherwise \end{cases} . \tag{2.55}$$

Then we rewrite the constraints in Eq. (2.54) in the following format:

$$\begin{aligned} & n(1-y_{1ij}) + \sum_{\{\mathcal{R}_k : l_i \in \mathcal{R}_k\}} I_{\mathcal{R}_k} - \sum_{\{\mathcal{R}_k : l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} \geq 1 \\ & n(1-y_{2ij}) + \sum_{\{\mathcal{R}_k : l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} - \sum_{\{\mathcal{R}_k : l_i \in \mathcal{R}_k\}} I_{\mathcal{R}_k} \geq 1 \\ & n(1-y_{3ij}) + \sum_{\{\mathcal{R}_k : l_i \in \mathcal{R}_k\}} I_{\mathcal{R}_k} + \sum_{\{\mathcal{R}_k : l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} - 4 \sum_{\{\mathcal{R}_k : l_i, l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} \geq 0 \\ & y_{1ij} + y_{2ij} + y_{3ij} = 1. \end{aligned} \tag{2.56}$$

The last equality guarantees that exactly one of y_{kij}'s equals 1, and the others are zero. Note that if $y_{kij} = 0$, the kth constraint is trivially satisfied. Therefore, the constraints in Eq. (2.57) are equivalent to the statement that one and only one in Eq. (2.54) holds. This implies that if a subset of $\mathcal{R}$ satisfies the constraints in Eq. (2.57), the network is 1-identifiable using those paths.

The following theorem summarized all the above findings for the minimum number of paths, which quarantines the 1-identifiability of a given network $N(V, E)$.

Theorem 13 *Suppose that a network $N(V, E)$ with a routing matrix, $\mathbf{R}$, and an equivalent set of paths, $\mathcal{R}$, is given. Determining the minimum number of paths r that guarantees the*

1-identifiability of the network can be written as follows:

$$
\begin{aligned}
&\min \sum_{i=1}^{|\mathcal{R}|} I_{\mathcal{R}_i} \qquad (2.57)\\
&s.t.\\
&\mathbf{R}^t \mathbf{I}_{\mathcal{R}} \geq 0\\
&\forall l_i, l_j \in E (l_i \neq l_j)\\
&n(1-y_{1ij}) + \sum_{\{\mathcal{R}_k : l_i \in \mathcal{R}_k\}} I_{\mathcal{R}_k} - \sum_{\{\mathcal{R}_k : l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} \geq 1\\
&n(1-y_{2ij}) + \sum_{\{\mathcal{R}_k : l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} - \sum_{\{\mathcal{R}_k : l_i \in \mathcal{R}_k\}} I_{\mathcal{R}_k} \geq 1\\
&n(1-y_{3ij}) + \sum_{\{\mathcal{R}_k : l_i \in \mathcal{R}_k\}} I_{\mathcal{R}_k} + \sum_{\{\mathcal{R}_k : l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} - 4 \sum_{\{\mathcal{R}_k : l_i, l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} \geq 0\\
&y_{1ij} + y_{2ij} + y_{3ij} = 1\\
&y_{kij} \in \{0,1\} \;\; k = 1,2,3\\
&I_{\mathcal{R}_i} \in \{0,1\} \;\; \forall i,
\end{aligned}
$$

where n is the number of links inside the network ($n = |E|$).

The number of constraints for the binary optimization problem in Eq. (2.57) is of order $O(n^2)$. Note that the above theorem is based on Theorem 10, which provides sufficient conditions for 1-identifiability. Hence, if Eq. (2.57) does not have a feasible solution, it does not necessarily mean that $N(V, E)$ is not 1-identifiable. However, our simulation results reveal that the chance for this event to occur is low. In other words, if Eq. (2.57) is not feasible, then one can say with high probability that the network is not 1-identifiable.

2.4.4 Heuristic Algorithm

Because the optimization problem in Eq. (2.53) and (2.57) is NP-complete, we introduce a heuristic algorithm by relaxing the binary constraints in (2.57), leading to a linear programming problem that is solvable in polynomial time. This is achieved by substituting the constraints $I_{\mathcal{R}_i} \in \{0,1\}$ and $y_{kij} \in \{0,1\}$, with $I_{\mathcal{R}_i} \in [0,1]$ and $y_{kij} \in [0,1]$, respectively.

$$\min \sum_{i=1}^{|\mathcal{R}|} I_{\mathcal{R}_i} \tag{2.58}$$

s.t.

$$\mathbf{R}^t \mathbf{I}_{\mathcal{R}} \geq 0$$

$$\forall l_i, l_j \in E (l_i \neq l_j)$$

$$n(1 - y_{1ij}) + \sum_{\{\mathcal{R}_k : l_i \in \mathcal{R}_k\}} I_{\mathcal{R}_k} - \sum_{\{\mathcal{R}_k : l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} \geq 1$$

$$n(1 - y_{2ij}) + \sum_{\{\mathcal{R}_k : l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} - \sum_{\{\mathcal{R}_k : l_i \in \mathcal{R}_k\}} I_{\mathcal{R}_k} \geq 1$$

$$n(1 - y_{3ij}) + \sum_{\{\mathcal{R}_k : l_i \in \mathcal{R}_k\}} I_{\mathcal{R}_k} + \sum_{\{\mathcal{R}_k : l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} - 4 \sum_{\{\mathcal{R}_k : l_i, l_j \in \mathcal{R}_k\}} I_{\mathcal{R}_k} \geq 0$$

$$y_{1ij} + y_{2ij} + y_{3ij} = 1$$

$$y_{kij} \in [0, 1] \;\; k = 1, 2, 3$$

$$I_{\mathcal{R}_i} \in [0, 1] \;\; \forall i.$$

Let $I^*_{\mathcal{R}}$ be the solution to the above LP optimization and $I^*_{\mathcal{R}_{i^*}}$ be the maximum entry in $I^*_{\mathcal{R}}$, i.e.,

$$I^*_{\mathcal{R}_{i^*}} = \arg\max_i I^*_{\mathcal{R}_i}. \tag{2.59}$$

We use $\mathcal{R}_{i^*}$ as one of the paths for probing or equivalently set $I^*_{\mathcal{R}_{i^*}} = 1$. Now, we solve the LP in Eq. (2.58) with the additional constraint $I^*_{\mathcal{R}_{i^*}} = 1$. If we run the above algorithm $|\mathcal{R}|$ times, we achieve an approximate solution to the ILP problem in Eq. (2.57). If the sequences of the LP problems are all feasible, it means that there is a zero-one vector $\mathbf{I}_{\mathcal{R}}$ such that all the constraints in Eq. (2.57) are satisfied. This implies that the network $N(V, E)$ is a 1-identifiable expander and $\mathbf{I}_{\mathcal{R}}$ contains the requisite minimum number of paths. However, if one of the LP problems in (2.58) is not feasible, it does not necessarily mean that the ILP is infeasible. In other words, the network $N(V, E)$ may be a 1-identifiable expander although the sequences of the LP problems in Eq. (2.58) are not feasible. Fortunately, our simulation results show that the chance of this event to happen is low. Figure 2.19 summarizes the relationship between the feasible sets for the LP optimization in (2.58) and

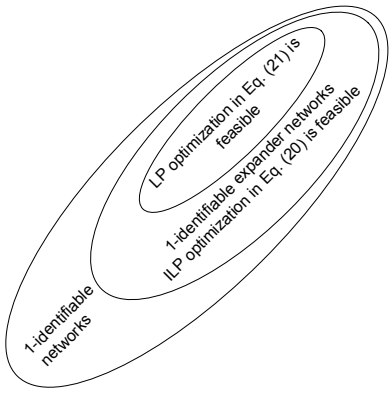


Figure 2.19: Relation between feasibility in Eq. (2.58), in Eq. (2.57) and 1-identifiable networks

the ILP problem (2.57) and the 1-identifiable networks.

2.4.5 simulation results

As mentioned in Section 2.2.2, the goal of compressed sensing is to achieve the maximum possible compression gain. In Section 2.4.1, based on our result on expander graphs, we propose the optimization problem in (2.58), which yields the minimum number of paths (minimum r) that makes the network 1-identifiable. For networks with a fixed number of boundary nodes that are 1-identifiable, Figure 2.20 shows a histogram of the number of paths needed (relative to the number of links in the network) in order to make the network 1-identifiable. After the paths that make the network 1-identifiable are selected, the LP optimization proposed in Eq. (2.42) can be used to find the link delays.

As can be seen in all the histograms in Figure 2.20, most of the mass is concentrated around 0.6, implying that with high probability, $r = 0.6n$ paths (where n is the number of links in the network) suffice for link delay estimation. Thus, the minimum path selection algorithm proposed in Section 2.4.3 will save, with high probability, 40% percent of the overhead probe traffic needed for link delay estimation, compared to the quantity required for exhaustive probing between all boundary pairs.

A second aspect highlighted by Figure 2.20 is that there is almost no mass below 0.4.

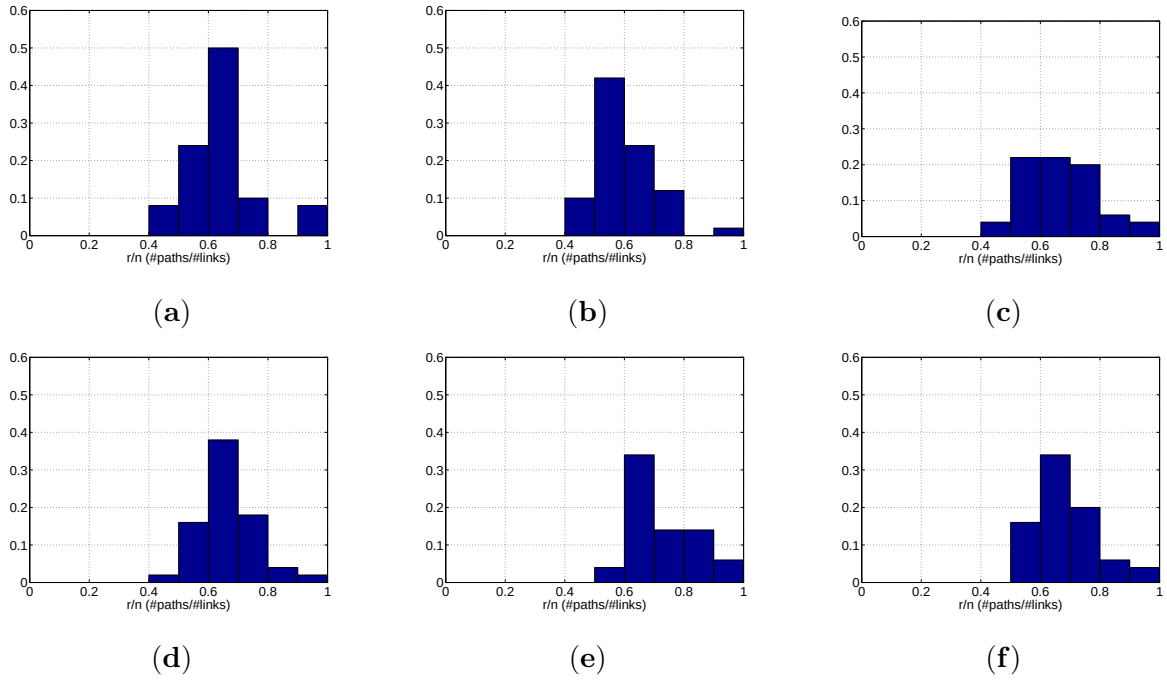


Figure 2.20: Histogram of $\frac{r}{n}$ for 1-identifiable networks with (a) 5 (b) 8 (c) 10 (d) 12 (e) 16 (f) 20 boundary nodes.

This means that for a network satisfying conditions in Theorem 10–i.e., being a 1-identifiable expander–$\frac{r}{n}$ cannot be less than 0.4. Restated, estimating link delay using the proposed algorithm in Eq. (2.42) requires at least $0.4n$ (end-to-end) measurements. The underlying cause is that the routing matrix of a graph exhibits a structure that is distinct from that of a random binary network.

Chapter 3

NETWORK CODING FOR WIRELESS NETWORKS

Network coding was initially proposed as a distributed mechanism for achieving the multicast theoretic (max-flow, min-cut) capacity in wired networks. In wired multicasting, information is sent from a set of source nodes to a set of destination nodes over a multihop network where the intermediate nodes merely forward their received packets via a predetermined look-up table (routing) [1]. Ahlswede et al., in [5] suggested the innovative notion of *coding* on layer-3 packets instead of look-up forwarding on specific outgoing links, and showed that network throughput can be increased.

In a network employing NC, routers perform a (random) linear combination of incoming layer-3 packets and broadcast the result to all its neighbors. Randomized linear network coding schemes were shown to be sufficient in achieving the information theoretic max-flow, min-cut bound on network capacity [113]. Necessary and sufficient conditions for the design of such random linear codes were provided by Koetter et al. [102]. While the concept of NC was developed for the network (IP) layer, it has often been implemented in practice at higher layers, such as the transport or application layers [163, 59].

A fundamental reason as to why network coding is beneficial is based on the premise of *simultaneous transmission* from several (source) nodes to a single (receive) node. While this is feasible in a wired network whereby concurrent transmissions are deemed 'orthogonal', a multi-hop *wireless* network is quite different. Wireless is a *shared* medium (at least for nodes within a common transmission range) and there is no natural spatial orthogonality. Thus wireless multihop networks have relied on other forms of orthogonality – in time (TDMA) or frequency (FDMA) – to achieve interference-free transmission. Wireless Network Coding (WNC) uses non-orthogonal transmissions that, nevertheless, allow recov-

[1] A router (layer 3 device) presently operates by forwarding an incoming IP packet to an appropriate outgoing line, as dictated by its routing table.

ery of multiple packets to enhance aggregate network throughput. Orthogonality is needed for any interference free communication. The broadcast nature of wireless (coupled with network topology) determines the nature of interference. Simultaneous transmissions in a wireless network typically result in all of the packets being lost (i.e., collision). A wireless network therefore requires a scheduler (as part of the MAC functionality) to minimize such interference. Hence any gains from network coding are strongly impacted by the underlying scheduler and will deviate from the gains seen in wired networks [140]. Further, wireless links are typically half-duplex due to hardware constraints; i.e., a node can not simultaneously transmit and receive due to the lack of sufficient isolation between the two paths. In addition, in a wired network noise power is negligible which means that signal-to-noise ration (SNR) in wired networks is significantly high. Another important feature of wired networks is (almost) constant characteristic of communication channels which brings low packet loss and high link reliability.

Extending network coding to wireless networks, however, meets several new and significant challenges. Primary among them is the broadcast nature of wireless that significantly impacts the nature of interference - in direct contract to a wired network where any number of nodes can simultaneously transmit to a single (receive) node successfully whenever a separate direct link exists. However, such simultaneous transmission in a wireless network would typically result in all the packets being lost (collision). A wireless network therefore requires a scheduler (as part of the MAC functionality) to minimize such interference. Hence any gains from network coding is strongly impacted by the underlaying scheduler and will deviate from that in wired networks [140]. Further, wireless links are typically half-duplex due to hardware constraints in contrast with wired links, i.e. a node may not simultaneously transmit and receive due to the lack of sufficient isolation between the transmit and receive paths.

In this chapter, we first investigate the issues and actual performance of network coding in wireless networks. We consider one of the simple and still challenging example of wireless networks, relay networks. We explain how to implement wireless network coding in a relay network, what the challenges are and how to overcome them. At the end we present system throughput and compare it with analytical result in which any inferior parameter in the

system considered to be perfect.

After that, we focus on application of network coding in wireless broadcasting. For wireless broadcasting, fountain codes are introduced as an effective solution to deliver information to a large number of users in the coverage area. In this chapter we aim to analyze the application of network coding as a fountain code and derive its performance in one-way wireless broadcasting scenario.

3.1 Relay Networks Using Network Coding

Another important consideration is the impact of the wireless channel on a transmitted signal – inclusive of channel attenuation which is assumed negligible on a wire, but may not be ignored in wireless. The received signal y over a wireless link can be modeled in general as

$$y = h\,x + z, \tag{3.1}$$

where x is the transmitted symbol, z is the additive noise sample at the receiver, and h is the (narrowband) channel loss between the source and the destination. Some previous work on WNC has incorporated aspects of the features mentioned above. Omnidirectional source transmissions were modeled in [121], [119] as hyper-links with additional constraints that prevent nodes from transmitting and receiving packets simultaneously. Interference effects were incorporated by [119] for joint optimization of MAC and network flows, where successful transmission between a node pair is based on a signal-to-interference-plus-noise-ratio (SINR) threshold, thereby potentially allowing simultaneous successful reception at different receive nodes.

One of the potential applications of WNC is in multicasting. A decentralized formulation to throughput optimization for the multicasting problem was introduced in [120][118]. However, if additional objectives such as maximizing throughput subject to delay constraints are considered, then network codes must be jointly designed with MAC as in [139, 167]. Authors in [6] qualify the impact of random access MAC schemes (such as CSMA/CA) on performance of NC in an all-to-all data dissemination system.

Information exchange via wireless relays is another natural scenario for potential ap-

plication of network coding on top of the MAC layer. In [166], the authors assume a deterministic MAC protocol and generalizes the canonical three-node scenario to the case with an arbitrary number of relays between two source nodes. The main issue with MAC layer NC is that any existing asymmetry in the system may cause performance degradation. In fact, as we will show in our experimental result in Section 3.1.8, even small asymmetry in source nodes power decreases system throughput. Hence, MAC layer NC is naturally suited for symmetric transmission rates between any node pair. Since NC at MAC layer directly operates pointwise on the information symbols in the two packets from the respective sources, equal size packets (hence equal rates) are necessary.

The efforts to extend the idea of MAC layer NC to asymmetric traffic include [116][154]. For example, [154] proposes to interpret NC as a mapping of modulation constellation to match symmetric traffic. However, it has been shown that, even in these schemes, performance is significantly reduced due to a lack of symmetry [33, 182]. To deal with asymmetric traffic, implementing the XOR operation of NC at the antenna has been proposed; this has been dubbed as *Analog Network Coding* (ANC) or *Physical Layer Network Coding* [98]. Recently, Chen et al. [33] proposed a new network coding scheme called Decode-and-Forward with Joint Modulation (DF-JM). They show that DF-JM has the potential to achieve the capacity for asymmetric or symmetric traffic. Simulation-based evidence of the gains from WNC for a cellular network, in terms of achievable rate regions for MAC-layer and PHY-layer network coding, was presented in [174, 115], based on the assumption of perfectly known channels at every user.

The above summary of wireless network coding literature suggests that little effort has been put forth into actually implementing WNC concepts in any lab-scale prototype and measuring performance in a real setting. One might say that, consequently, WNC is a largely unproven idea in practice; our work is thus distinguished by demonstrating some of the first evidence of the effectiveness of WNC in a canonical relay network.

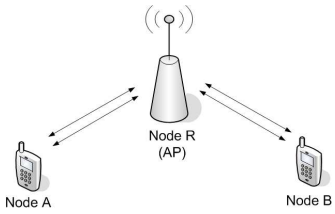


Figure 3.1: A relay network, Nodes A and B want to exchange data through node R

3.1.1 Wireless Relay Networks: A Review

As mentioned, we aim to investigate the challenges of implementing WNC in a simple relay network using a currently available software-defined platform, and suggest potential solutions. We present measured system throughput and compare it with suitable analytical results for benchmarking. Consider a two-way (bi-directional) relay channel with two source/destination nodes (nodes A and B), and one intermediate relay node (node R), as depicted in Figure 3.1. By assumption, the relay node only forwards data and is not a source or destination. Nodes A and B are not within transmission range of each other, and they require the relay node to communicate. For a half-duplex system, a baseline strategy for A and B is to exchange data via TDMA scheduling through node R. In the first time slot, node A transmits data to R. In the next time slot, R relays the received data to B. In the third time slot, node B transmits data to R. Finally, in the fourth time slot, R sends B's information to A. As depicted in Figure 3.2-(a), two bits of information would thus be exchanged in four time slots.

As already shown in [107] and summarized in Figure 3.2-(b), it is possible to exchange two bits of information in three time slots by applying network coding in the MAC layer at the relay node, thereby achieving a 33% throughput improvement relative to pure TDMA. In the first time slot, A sends its data x_A to R. In the second time slot, B transmits its information, x_B to R. In the third time slot, node R broadcasts $x_A \oplus x_B$. Since both nodes know their own data, if they receive $x_A \oplus x_B$ they can extract their desired information.

The above throughput enhancement requires an ideal TDMA MAC, which deterministically assigns a channel to each node in the network. However, one of most common

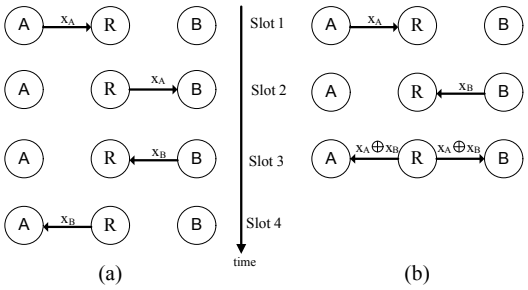


Figure 3.2: (a) 4-step and (b) 3-step information exchange using NC. In both scenarios, MAC protocol is TDMA.

wireless access networks (802.11) uses a random access MAC protocol based on carrier sensing (CSMA/CA) for time-sharing of the common channel. Besides the obvious advantages of a distributed protocol, such random access schemes are efficient and fair at low average traffic loads as in several data applications. Hence in this work, we first explore the challenges of implementing WNC over a system with random MAC protocol. The measured MAC layer throughput with NC shows significant gains in system throughput, close to the analytical predictions.

As discussed in the previous subsection, MAC layer network coding is based on *symmetric* traffic, whereas many data exchange scenarios are asymmetric by nature. Since the relay node can be placed at varying distances between both sources, the data rates transmitted by the relay node to both destinations are typically different. As mentioned, PHY layer NC has been proposed to deal with asymmetric traffic. Unfortunately, there is relatively little attention paid in the literature to system implementation and actual performance measurement of applying NC at PHY layer. Motivated by this observation, here, we extend the concept of Decode-and-Forward with Joint Modulation (DF-JM) developed in [33] (state-of-art PHY layer NC) to the asymmetric relaying problem considered here. In other words, we implement DF-JM to an OFDM-based (similar to 802.11a) PHY layer on a suitable test-bed platform (described later) and measure actual PHY layer NC throughput in a real-world application–a first, to the best of the authors' knowledge.

The current state-of-art of PHY NC treats the addition of two (synchronized) analog

signals at the relay antenna as the XOR operation over the information symbols. However, a closer model to reality for such a superposition should include the respective channel gains, i.e.,

$$y = h_1 x_1 + h_2 x_2 + z, \tag{3.2}$$

where x_i is the symbol from source i, and h_i is narrowband channel gain between source i and the relay. Note that, besides the requirement of synchronism, successful analog network coding also requires full knowledge of channel gains at the relay node.

In summary, we are pursuing the following three main goals in this section: a) to *design and modify* the CSMA-based MAC layer to support network coding, (b) to *develop* new ideas for PHY-NC for asymmetric traffic scenarios and *apply* the new ideas to OFMD-based PHY layer, and (c) to *demonstrate* the utility of WNC via laboratory-scale experimentation using *commodity* wireless radios, notably 802.11a/g. Since much of NC theory has traditionally presumed orthogonal time-scheduled MAC layer (such as TDMA), we believe that our results are some of the first to estimate the benefits from WNC for a CSMA/CA MAC.

The experiments were conducted using the Microsoft's new Software Radio (SORA) platform [151] and the open source 802.11a code. SORA is a fully programmable software radio platform based on general purpose multi-core processors in commodity PC architecture, developed by Microsoft Research Asia (MSRA). With SORA, developers can implement and experiment with high-speed wideband wireless protocols (like IEEE 802.11a/b) using commodity general-purpose PCs [180].

3.1.2 System Description

We propose to apply wireless network coding – both at MAC and at PHY layer - to a bi-directional relay network, as shown in Figure 3.1. Our hardware platform, SORA, implements 802.11a MAC and PHY layer. First, we explain how to apply network coding on top of 802.11 MAC with minimum modifications. Then, we describe how the DF-JM scheme is implemented to support the new PHY layer NC concepts.

3.1.3 802.11 MAC Layer Wireless NC

The achievement of a 33% improvement in idealized network throughput (displayed in Figure 3.2) for the canonical two-node scenario communicating via a relay, is attained based on some key assumptions, notably: a) a scheduled MAC, such as TDMA, with implicit node synchronization, and b) symmetric, constant-rate traffic, whereby source nodes have data to send in every slot. In the next two subsections, we describe an implementation for the SORA with 802.11, which bypasses both of the above constraints. To the best of our knowledge, there has been no demonstration of the benefits of NC in an 802.11 infrastructure network, where the Access Points act as natural relay between sources and sinks. The known short-term unfairness due to carrier sense multiple access type MAC protocols such as 802.11 provides new challenges in implementing NC, and will be explored further [68].

Relay Node

The relay node R (access point in 802.11) receives packets from both sources, implements NC operation, and broadcasts the result. Due to CSMA/CA channel access, the relay may receive many packets from one source before it gets any packets from the other. This necessitates two queues at the relay – one each for packets from sources A and B, respectively. Whenever the relay receives a packet from one of the sources, it looks at the other queue. If it finds the other queue empty, it queues the packet and waits for packets from the other source to arrive. The relay keeps queuing packets from a source node until it receives one from the other source. Then it XORs one of the queued packets from the first source and the (unique) *proper packet* from the other, and broadcasts the result (Figure 3.3). Roughly speaking, a packet from source A (B) is called a proper packet of the packet from source B (A), if they have the same reception order (Figure 3.4). Later, in Section 3.1.6, we discuss the choice of such a proper packet in more detail.

The queue size at the relay node should be sufficiently large enough to achieve an acceptable packet drop rate. Further, when the relay node receives a packet, it has to search the queue to find a *proper packet* to be combined with the received one. Accordingly, since searching a large queue is time and energy consuming, the queue size should also be

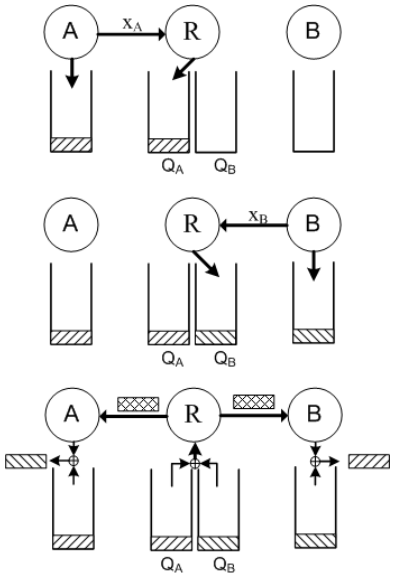


Figure 3.3: Proposed system architecture for MAC layer NC

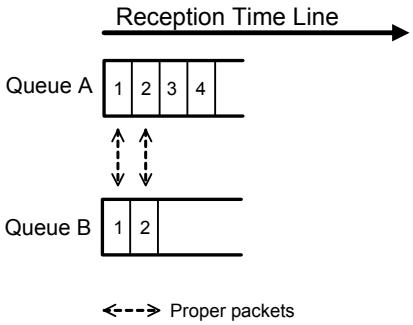


Figure 3.4: Simple illustration of proper packets

sufficiently small enough to meet any NC delay constraint.

Source/Sink Nodes

Upon receiving a packet from the upper layer, the source node starts listening to the channel. Whenever it finds the channel empty, it captures the channel and starts transmitting. A copy of transmitted packet is saved in its buffer, since it is needed to decode information from other sources as a result of NC. Whenever it receives a packet from the relay node, it checks its destination address. If the packet was a broadcast packet, it fetches a *proper packet* from its buffer, calculates the XOR between the two, and sends the result to the upper layer.

3.1.4 PHY Layer

Upon receiving the broadcast messages from the relay node, each destination decodes its intended message with its own signal as side information. As such, the two-way relay can in general be regarded as three separate slots, i.e., two sources send to relay node in slots 1 and 2 resp. and the relay broadcasts with side information in slot 3.

Slot 1 and Slot 2

In slot 1 (slot 2), the information bits $w_A(w_B)$ are encoded and bit-interleaved and applied to modulator, generating the symbols x_A (x_B) from the respective constellations $\mathcal{M}_A(\mathcal{M}_B)$, that is transmitted to the relay node. Since the transmit rate mainly depends on the constellation size, we consider BPSK modulation at both sources for the *symmetric* relaying. For *asymmetric* relaying, B transmits QPSK signals while A uses BPSK modulation or vice versa. During the first time slot, the signal received at the relay node is thus given by

$$y_{AR} = \sqrt{P_A} h_A x_A + z_{AR}, \tag{3.3}$$

where Y_{AR} and Z_{AR} are the received signal and the zero mean complex Gaussian noise of variances σ^2_{AR} of the relay node, respectively. Likewise, the received signal at the relay node during the second time slot is

$$y_{BR} = \sqrt{P_B} h_B x_B + z_{BR}, \tag{3.4}$$

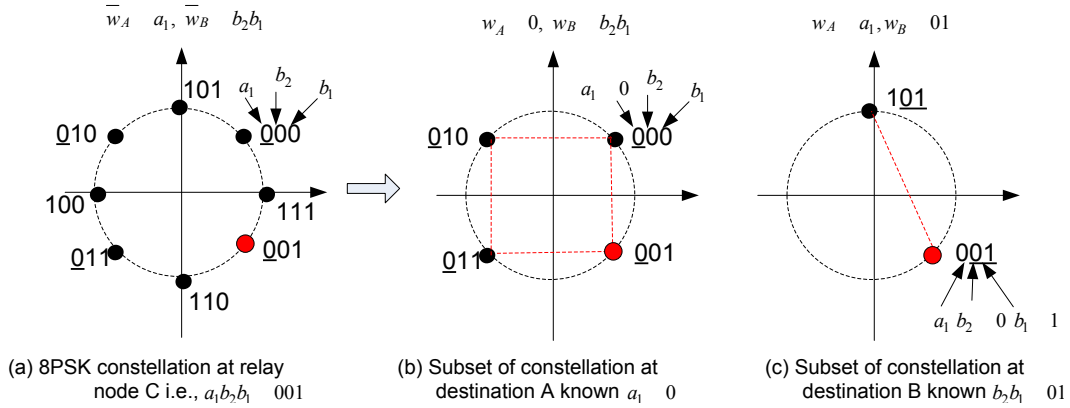


Figure 3.5: Demapping at both destinations for DF-JM schemes based on optimal 8-PSK labeling

where h_i, $i = A, B$, is the channel coefficient of the link between the source/destination i and the relay node, and reciprocal channel can be assumed.

Slot 3

During the third time slot, we apply the DF-JM scheme in the relay node to broadcast the data. The relay node concatenates the decoded information $\overline{w}_A$ and $\overline{w}_B$, into a new sequence $\overline{w}_R = [\overline{w}_A\ \overline{w}_B]$, and then encodes and modulates the resulting sequence jointly, regardless of the sizes of the original messages. As an example, we consider the asymmetric traffic with BPSK and QPSK constellations. Thus, we have $\overline{w}_A = [a_1]$ and $\overline{w}_B = [b_2, b_1]$, where a_1,b_2 and b_1 denote the binary symbols; then $\overline{w}_R = [a_1, b_2, b_1]$. The transmitted signal from R is $x_R = \mathcal{M}_R(\overline{w}_R)$ by using the constellation $\mathcal{M}_R$. The corresponding received signals at stations A and B are respectively $y_A = \sqrt{P_R} h_A x_R + z_A$ and $y_B = \sqrt{P_R} h_B x_R + z_B$, where z_A and z_B are zero mean, complex Gaussian noise of variances σ_A^2 and σ_B^2 .

With the help of known redundant bits w_A, node A can decode the desired bits w_B using subset partitioning, i.e.

$$\widehat{w}_B = \arg\min_{W_B} \left| y_A - \sqrt{P_R} h_A \mathcal{M}_R([w_A W_B]) \right|^2 . \tag{3.5}$$

Likewise, B can perform detection from the subset of the constellation points based on

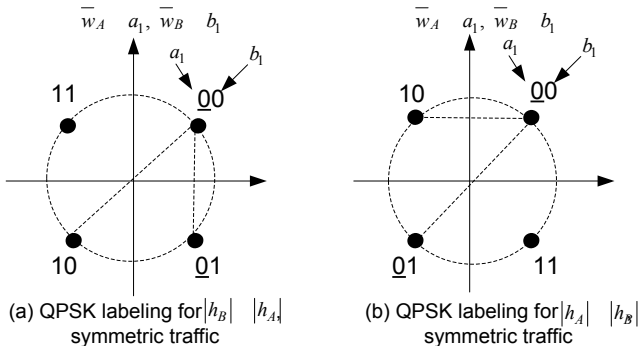


Figure 3.6: Optimal constellation labeling for QPSK constellation at the relay node

the known sequence w_B. This is explained by the example shown in Figure 3.5 using 8-PSK constellation at the relay node. The relay node R needs to forward $\overline{w}_A = [a_1]$ and $\overline{w}_B = [b_2, b_1]$ to both destinations, where we assume that $a_1 = 0$ and $[b_2\ b_1] = [0\ 1]$ without loss of generality. By applying the DF-JM scheme, the relay node R combines them together as $\overline{w}_R = 001$ and transmits $X_R = (001)_{8PSK}$ using the 8-PSK constellation. Thanks to the fact that the destination A knows $a_1 = 0$, it only needs to consider the possible $[b_2\ b_1]$ from all the 8-PSK points for which the first bit is 0, as shown in Figure 3.5. Similarly, the estimated a_1 for destination B can be chosen from the points $(001)_{8PSK}$ and $(101)_{8PSK}$. These results implicates that *signal detection can be performed from a subset of the constellation points instead of the entire constellation.* Therefore, we can see that for DF-JM scheme, the high level constellation is used in the relay node, but low level constellation can be used for de-mapping at sink nodes A and B by exploiting the side information. From Figure 3.5, we also can see that if the constellation labeling map is carefully chosen in the relay node, the intra-subset Euclidean distance (i.e., $(001)_{8PSK}$ and $(101)_{8PSK}$) can be greatly increased through the side information. It can be verified that such labeling for 8-PSK in Figure 3.5 is optimal; the optimal labeling for QPSK is shown in Figure 3.6.

3.1.5 SORA Implementation

The network stack of the legacy 802.11 implementation in SORA is depicted in Figure 3.13 and consists of three parts [151]:

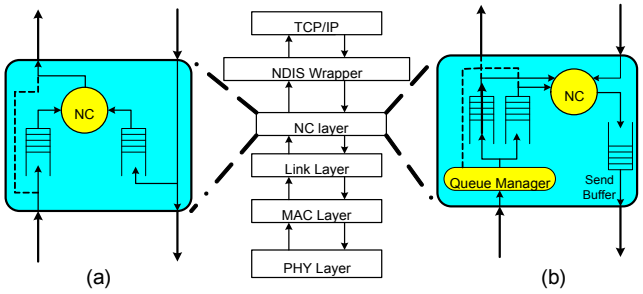


Figure 3.7: Proposed 802.11 network stack for (a) source/sink (b) relay node(s) in SORA

- PHY layer–this layer is OFDM-based and similar to 802.11a PHY layer.

- MAC layer–this layer is simply a state machine modeling CSMA/CA, the core component of Distributed Communication Function (DCF) in 802.11a/b/g.

- Link layer (LL)–this layer is responsible for interfacing with TCP/IP layer. When a node receives a MAC frame, this layer decides what to do with it. If the packet is addressed to this receiver, LL passes it to TCP/IP layer; otherwise, the packet is dropped.

Next, we describe how this stack was changed to support MAC and PHY network coding as described above.

3.1.6 NC at MAC layer

To apply network coding at the MAC layer, the legacy PHY and MAC layers implemented in SORA architecture remain untouched. All changes are applied to Link layer via addition of an NC layer, dedicated to NC operation. As mentioned in Section 3.1.3, this layer is different for source/sink nodes and the relay node. However, they both need a network coding unit, which simply XORs data on its input ports. Figures 3.7-(a) and 3.7-(b) show the new stack protocol for relay and source/sink nodes, respectively.

Packet format

When a source receives a packet from the relay node, it knows that the packet is a result of XOR combination of one of its own packets and the desired packet transmitted from the other source. But it may already have sent a number of packets before receiving any from the relay. Hence it needs to know which one of its packets has been used for encoding at the relay node. To solve this problem, Chou et al. proposed the concept of 'generation' for a packet [35]. Each packet contains a metadata field in its header representing its *generation*. Only packets from the same generation can be combined. Each source has a `counter` which shows its current generation number. At the beginning, both sources reset their generation counter to zero. Whenever they transmit a packet, they insert the current generation number in the related field in a packet header and increment the `counter` by one.

The relay node only combines two packets from two sources if they are from the same generation. If it receives a packet from source A (B) with generation number n, it first searches in the buffer dedicated to source B (A) looking for a packet with the same generation number n. At that point, one of the following two scenarios will ensue:

- There is a packet with generation n. In this case the relay node will combine the two packets and broadcast the result. Then, it look at the buffer dedicated to A (B) and deletes any packet from generation n.

- The buffer is empty or there is no packet from generation n - in this case, relay node will save the packet in the buffer dedicated to A (B). If there is already a packet from generation n in A (B) buffer, it will be rewritten by the new one.

When a source receives a packet, it looks at its generation number. It fetches a packet with the same generation number from its buffer, fulfills network coding operation on the two packets, and sends the result to the upper layer.

Note that the variable `counter` that keeps track of a packet's generation is simply a register in a finite field, which means that the value contained is bounded and will reset after a while. Therefore, when two packets from generation n are combined, there is a possibility

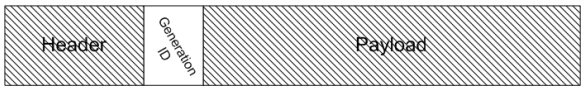


Figure 3.8: Field for generation number in packet header

that they actually belong to different generations. The probability of this occurring depends on the size of the `counter` and is minimized by a sufficiently large value of generation ID (or `counter`). On the other hand, adding generation ID to the packet header increases packet overhead, representing a tradeoff in choice of generation ID field size.

Source/Sink Nodes

Each source/sink node has two major threads, one for receiving a packet (sink part) and one for transmitting (source part). To implement the MAC layer NC, the receiving thread must know about the packet flow in the sending thread, i.e. the receiving thread needs to know exactly which packets have been transmitted to complete decoding. For that reason, as discussed in Section 3.1.3, there is a shared `buffer` with bounded length `BUFFER_SIZE` between the receiving thread and sending thread in each source node.

When a packet is transmitted, the sending thread puts a copy of the packet in the `buffer`. If the buffer is full, it will overwrite the oldest packet in the buffer. If the source receives a packet, the receiving thread looks at its generation ID and fetches a packet, if any, from the `buffer` with the same generation number from its buffer. Then the node XORs the two packets (received packet and one from the buffer) and sends the result to the upper layer. If there is no packet with that generation ID, the receiving thread sends the packet to the upper layer. If the packet has undergone NC (XOR at the relay node), then the CRC check at the upper layer fails and the packet is dropped; only a non-NC directly from the other source is accepted. Note that since the `buffer` is shared between two threads, lock protection between the threads is needed [148].

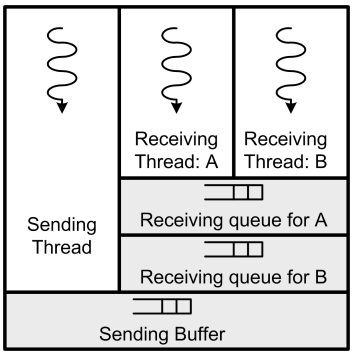


Figure 3.9: Multithreading and shared queue configuration at the relay node

Relay Node

The relay node has three major threads: the receiving thread for A, the receiving thread for B, and the sending thread (Figure 3.9). Two receiving threads share two buffers as depicted in Figure 3.3– one for each source. When the relay receives a packet from A (B), its dedicated thread wakes up and extracts the generation number from the received packet. It then searches inside the buffer dedicated to B (A) for a packet with the same generation ID. If there is such a packet, the relay node combines the two packets, signals the sending thread, and hands the resulting packet to it to broadcast. If there is no packet with the same generation ID as the received packet, it is saved in a buffer dedicated to A. If the buffer is full, the oldest packet would be fetched out, handed to the sending thread for broadcast, and replaced with the new packet.

As depicted in Figure 3.9, the three threads at the relay node share the `sending buffer`. Whenever a packet is ready to be sent, it is placed in the `sending buffer`. If the buffer is full, the oldest packet would be overwritten. The sending thread always monitors the `sending buffer`; while the buffer is not empty, the sending thread fetches the oldest packet and sends it to the PHY layer.

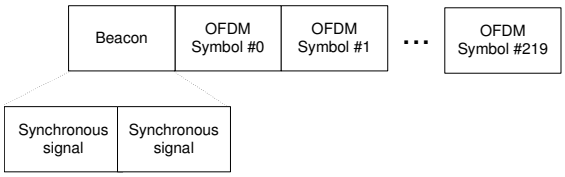


Figure 3.10: PHY symbol frame structure in OFDM system

3.1.7 NC at PHY layer

Frame Structure

As shown in Figure 3.10[2], the physical layer is based on 11.52 ms duration frames, comprising of beacon and data components. The beacon consists of two identical band limited pseudo random signals, mainly used for frame synchronization and frequency offset estimation. The data signal is composed of 220 OFDM symbols. Each OFDM symbol includes 256 subcarriers, but only 198 subcarriers is used, which includes 168 data subcarriers, 12 continuation pilots and 18 scattered pilots. The three types of signals are transmitted over the OFDM time-frequency grid as in Figure 3.11. The information for physical layer such as modulation and coding scheme and the type of network coding are included in continuation pilots. The continuation pilots can also serve for frequency synchronization and phase tracking. The rectangularly distributed scattered pilots serve as reference for channel estimation, and the power of continuation pilots and scattered pilots are set to be 3 dB higher than data signal. Table 3.1 summarizes the detailed PHY parameters that supports BPSK, QPSK, and 8-PSK modulation schemes for different relay scenarios.

Packet Relaying Processing

Since the DF-JM is employed in the relay node, we extend the OFDM PHY with DF-JM scheme; the associated signal processing for the transmitter and receiver is shown in Figure 3.12. At the third time period when the relay node becomes a transmitter, two LDPC encoders will be used to encode the information from the two sources. Based on channel

[2]Note that a basic OFDM PHY is implemented here to support the experiments, which can be extended to the PHY frame in 802.11a.

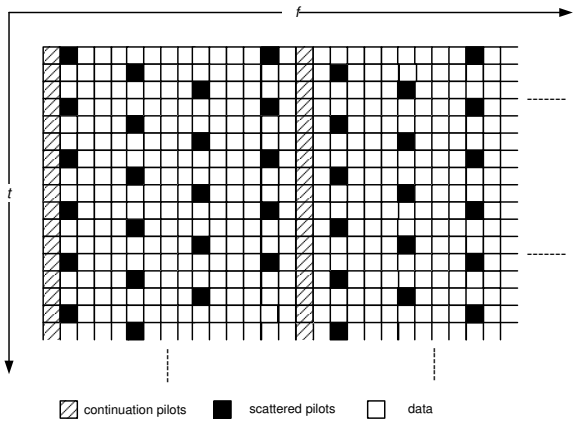


Figure 3.11: Signal types in time-frequency grid

Table 3.1: OFDM PHY parameters

Parameter	Value
Frequency band	2.422 GHz
signal bandwidth	4.254 MHz
subcarrier spacing	21.484 KHz
Symbol duration (data)	46.546 μs
Guard interval duration	5.8182 μs
Frame length	11.52 ms
LDPC length	9216
Code rate	1/2

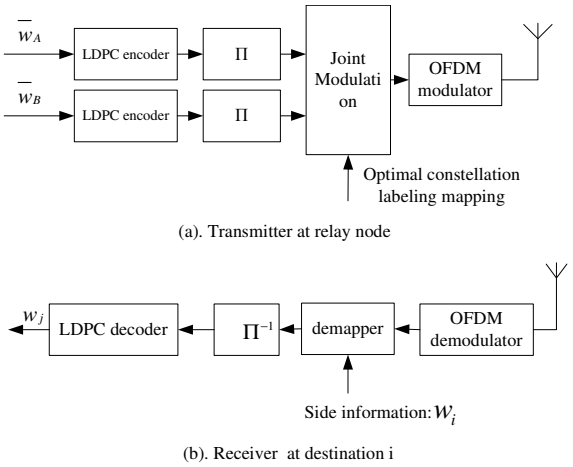


Figure 3.12: Signal processing in the relay node and destinations (source/sink nodes) for Phy layer NC

condition, three types of modulation are considered here: BPSK, QPSK and 8-PSK packets. For symmetric relaying, nodes A and B transmit the BPSK packet to the relay node R, and then QPSK packet can be transmitted by the node R using the DF-JM scheme. When the node R is placed near by node B, the link between B and R can support QPSK transmission, allowing the node R to transmit the 8-PSK packet based on DF-JM scheme in the third time slot. For comparison, optimal labeling and Gray mapping for QPSK and 8-PSK packets are introduced at the node R and the labeling information carried by the continuation pilots.

3.1.8 Experimental Results

3.1.9 Sora Platform

Software defined radios (SDR) have been attracting increasing attention recently. In an SDR system, components that are typically implemented in hardware (e.g., mixers, filters, amplifiers, modulators/demodulators, detectors, etc.) are instead implemented by means of software on a suitable hardware platform [47]. Changing a component implemented in software is easier and faster, leading to consequent flexibility in modifying a communication system built on a SDR.

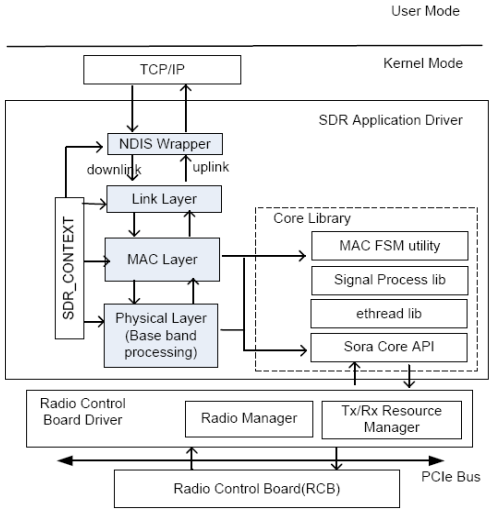


Figure 3.13: Hardware/Software structure in SORA [151]

Microsoft's Software Defined Radio platform (SORA) consists of three fundamental components: [151]:

- RF front-end
- Radio Control Board (RCB)
- SDR application driver

Figure 3.13 illustrates the SORA architecture. The RF front-end represents the well-defined interface between the digital and analog domains. It contains analog-to-digital (A/D) and digital-to-analog (D/A) converters, and necessary circuitry for radio transmission. Since all signal processing is done by the software, the RF front-end design is rather generic. The RF front-end in SORA is interchangeable; in this work, we use the WARP radio board [1].

The RCB is the new PC interface board for establishing a high-throughput, low-latency path for transferring high-fidelity digital signals between the RF front-end and PC memory. To achieve high system throughput, RCB uses a high-speed, low-latency PCIe bus, which supports a throughput of up to 16.7 Gbps with x8 model. The large on-board memory

further allows caching pre-computed waveforms, adding additional flexibility for software radio processing. Finally, the RCB provides a low-latency control path for software to control the RF front-end hardware and to ensure it is properly synchronized with the host CPU.

In SORA, SDR components are located in a separate upper-level driver, called an SDR Application Driver, where the customized Link layer, MAC and PHY are implemented. An SDR application driver accesses the hardware resource via the Sora Core API inside the Core Library. The core library implements the common functions of radio and other hardware resource management, like radio board configuration and control. Specially, it provides the necessary system services and programming support for implementing wireless PHY in a general-purpose multicore processor. Several processing tasks with intensive computation complexity, such as the down-sampling, Fast Fourier Transform (FFT) and LDPC decoder, are optimized by taking advantage of the data-parallelism with the Single Instruction Multiple Data (SIMD) instruction sets of the CPU.

3.1.10 NC at MAC layer: Experiments

We use three i7 personal computers supporting PCIe. Each computer is equipped with a SORA board and a warp radio board as the RF front-end. To remove any hidden node problem, we place each node on a vertex of an equilateral triangle with a side length of 1 m [3]. In this configuration, every node is in coverage range of the others. However, source nodes only accept packets from the relay node.

Each node is performing in 802.11a basic mode with a transmission rate of 1 Mbps. Each source contains a 135 MB file aiming to send it to the other one. Packets are 1 KB long; i.e., it takes roughly 8 ms to send a packet. Length of generation ID is 32-bit, which is a negligible overhead compared to the length of the packets. For the sake of simplicity, in this work, the same amount of memory is assigned to every queue in the system architecture discussed in Section 3.1.6. In other words, all queues would have the same size in each experiment. Having different sizes for each queue and observing the role that plays on

[3] Estimating the impact of hidden node is out of scope of this work.

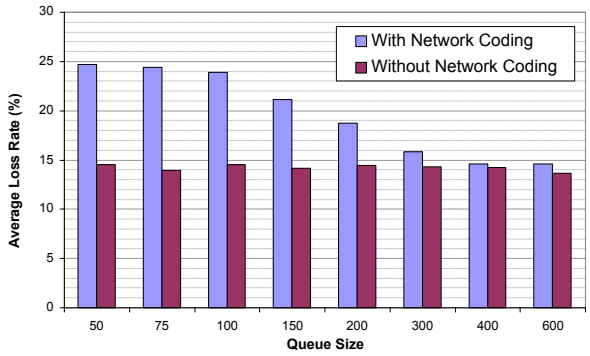


Figure 3.14: Average packet loss rate in two-way relay on 802.11 framework with and without NC at MAC layer

system performance is deferred to future work.

Figure 3.14 compares packet loss rates in the relay network with and without network coding. As we discussed in Section 3.1.6, using network coding increases the likelihood of losing packets. A solution to this problem would be increasing the size of the buffer. As one can see, having a large enough queue would overcome any packet loss caused by MAC layer NC. However, increasing queue size, as Figure 3.15 shows, would increase the average delay in the system. In this instance, we define delay as amount of time needed for a packet to traverse from one source to the other. In this figure, we normalize the y-axis such that a packet length is 1 s. At the cost of increasing the incidence of packet loss and a little more computational complexity at the nodes, MAC layer NC decreases packet delay at low queue size. If one increases size of the queue to decrease system packet loss, that cancels out the advantage of shorter packet delay as in Figure 3.15.

System throughput (packet/s/node) is depicted in Figure 3.16, where throughput is the average number of successfully received packets at each node per second. Comparing to baseline TDMA relay protocol (4-step without network coding), NC at MAC layer increases the throughput by about $20 - 30\%$. While system throughput for applying NC to CSMA/CA MAC is always less than 35 packets/s/node, for ideal TDMA-based system, the throughput would be around 41 packets/s/node. That means, when network coding is applied to MAC layer, randomness in capturing the channel causes nearly 15% reduction in

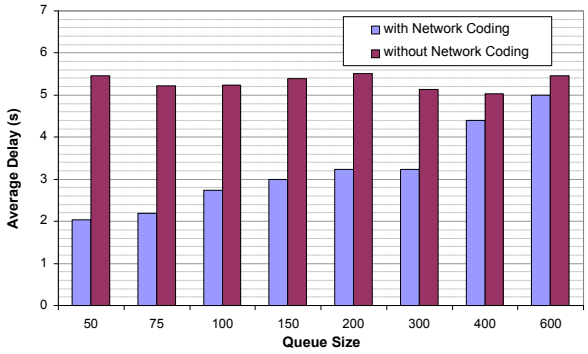


Figure 3.15: Average delay in two-way relay on 802.11 framework with and without NC at MAC layer. Delay axis has been scaled such that the packets are 1 second long (equivalently 1Mbit long).

system throughput compared to an ideal deterministic TDMA MAC.

Finally, to test the sensitivity of MAC layer NC, we change one of the nodes' power and keep the other parameters of the system untouched. In each experiment, we halve the power of node B and measure system throughput and packet loss rate for each source/sink in the system, as depicted in Figure 3.17. Figure 3.17-a presents the following results: when the power of node B decreases, its packet loss rate increases while the packet loss rate of the other node (node A) remains almost the same.

For MAC layer NC, the relay node needs to receive packets from both nodes in order to do the encoding (XOR the packets) and broadcast the result. Hence packets from node A are buffered until a packet successfully arrives from node B. This means delay from A to B increases or, equivalently, throughput decreases. As one can see in Figure 3.17-b, having asymmetry in the system would decrease throughput of node A as well. The reason for that can be justified as follows: buffer dedicated to node A at the relay node would be filled up because of the high packet loss rate from node B.

As discussed in Section 3.1.6, the relay node would start broadcasting uncoded packets from node A to prevent buffer overflow. This means the sending buffer at the relay usually has some packets from A to be sent. Therefore, when a packet is received from B and gets encoded by the correspondent packet (i.e. same generation) from node A, the coded packet

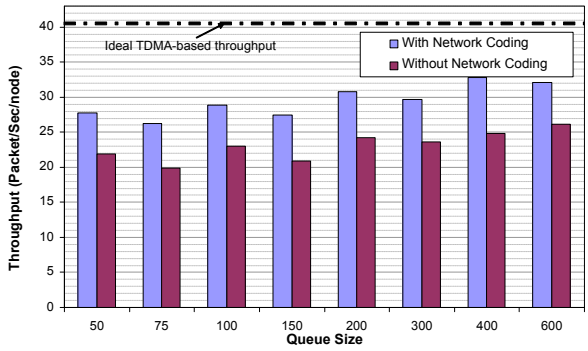


Figure 3.16: System throughput (per node) in information exchange for a 802.11 network when MAC layer NC is applied

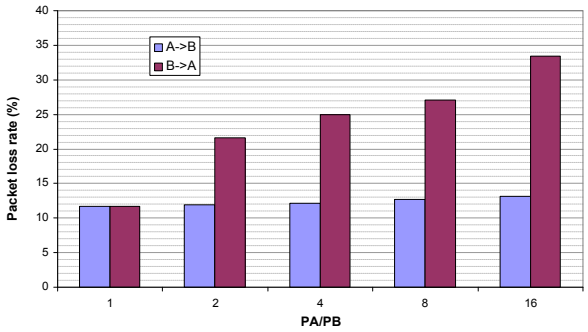
would spend some time in relay's sending buffer, increasing the delay from B to A and degrading throughput at A.

3.1.11 NC at PHY layer

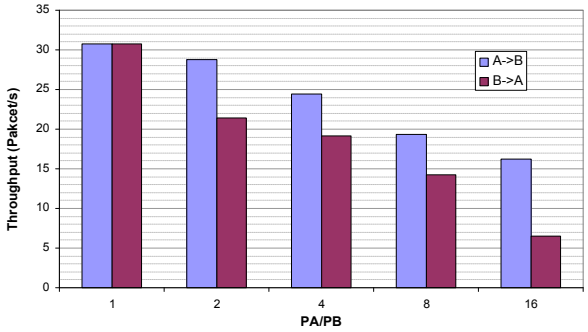
As our experience in the previous subsection shows, unequal packet loss rate would decrease system performance. PHY layer NC is proposed to deal with any asymmetric traffic. In this subsection, the performance of the demonstration platform with the novel PHY prototype is shown. In the test set-up, the platform was implemented on three PCs equipped with external antennas. By adjusting the transmit power and the distance between transmit and receive antenna pairs, the channel condition can be changed for different scenarios.

In Figures 3.18 and 3.19, we illustrate the received signal after channel equalization at node A for the two-way symmetric and asymmetric traffic.

The end-to-end throughput for both symmetric and asymmetric traffic corresponding to different network coding schemes is presented in Figure 3.20. It is observed that irrespective of symmetric or asymmetric traffic, a three-step information exchange scheme based on DF-JM with optimal labeling significantly improves the network throughput, compared to four-step information exchange scheme. It is noteworthy that the optimal labeling provides a nearly 100% gain over the Gray mapping at SNR around 7dB for asymmetric relaying, but does not have the same throughput improvement for symmetric relaying. This behavior



(a)



(b)

Figure 3.17: Reducing transmission power at node B to measure (a) packet loss rate (b) throughput of each node when MAC layer network coding is used

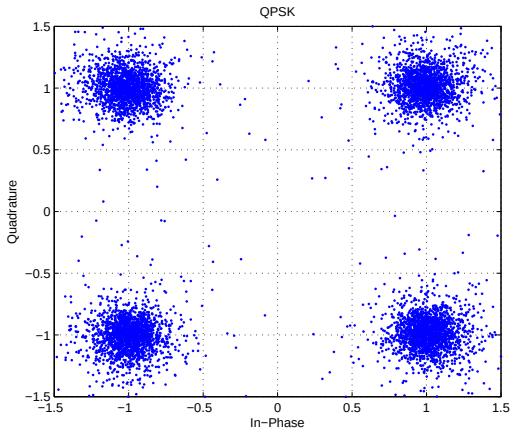


Figure 3.18: The received signal after channel equalization in A for symmetric relaying, SNR =16.0632 dB

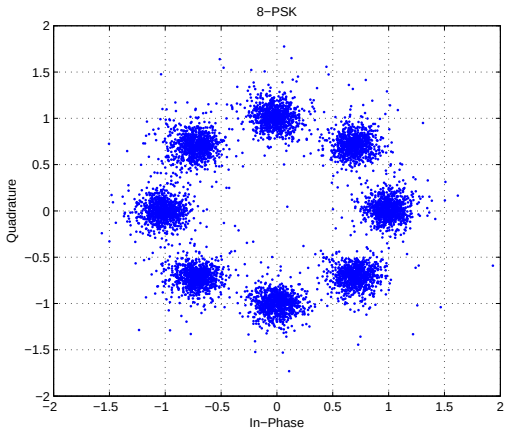


Figure 3.19: The received signal after channel equalization in A for asymmetric relaying, SNR =16.4238 dB

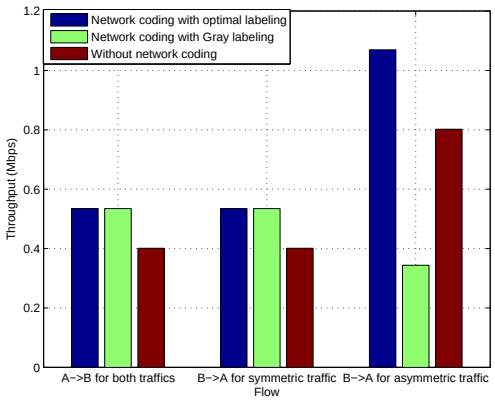


Figure 3.20: The end-to-end throughput (per flow) in information exchange for an OFDM-based network when PHY layer NC is applied

is reasonable because the SNR=7dB is enough to support the transmission of QPSK signal without decoding error.

3.2 Wireless Broadcasting using Network Coding

For point-to-point communication, automatic repeat request (ARQ) is an effective and reliable error control mechanism that utilizes acknowledgments and timeouts to recover from packet losses. It achieves robust, in-order delivery at the cost of increased delay and lower throughput. However, its application is limited largely to *unicast* scenarios; for broadcasting, it does not scale well due to the feedback implosion problem [133] and it's use has largely been limited to small groups of end-users.

An effective class of solutions to the above problem involves methods that do *not* require explicit receiver feedback, such as the notion of *erasure coding* that has been proven to be effective for broadcast channels. The scenario is straightforward: a base station (BS) has a source file that must be sent to all receivers. The file is divided into N *information packets*, which are then encoded into a stream of coded packets and transmitted, without any feedback from the receivers. When a receiver receives $N' = N + E$ encoded packets, it is able to decode the N source packets with probability $1 - \delta(E)$.

The process described explains the core idea of erasure coding. One of the earliest erasure code families is based on Reed-Solomon (RS) coding, in which the receiver needs to receive any N packets from the encoded broadcast packet stream to decode the information packets. RS erasure codes require the source to know the channel characteristic (erasure rate) before they begin the encoding process. In many scenarios such as satellite communications, such knowledge of channel error rate is infeasible or time-consuming to measure. Moreover, in many broadcast applications, the erasure rates between different source-receiver pairs vary. To overcome these issues with RS codes, fountain codes were introduced [**?**]. Fountain codes are considered to be rateless, because the number of encoded packets that can be formed from the source messages is potentially limitless. The source keeps transmitting encoded packets generated from N information packets. When a receiver receives any $N' = N + E$ encoded packets correctly, it is able to recover the information packets (this occurs typically with sufficiently high probability $1 - \delta(E)$).

While in theory a BS can transmit as many coded packets as needed, in practice we desire that the BS only transmits M (fixed maximum number) coded packets, where $M \geq N$ is *as small as possible* subject to sufficiently large fraction of the receiver population being able to decode the full N sequence of information packets, with high probability. In other words, when the BS has transmitted M packets, we require that $(1-\beta)\,\%$ of users in the network are able to recover the source file with probability $1-\eta$. The parameter β represents an *outage fraction*, indicating the maximum percentage of users who are unable to receive service from the BS. The relation between β, η, and M determines the performance of a fountain code. In this paper, we employ random linear network coding (RLNC) [85] as a rateless code and show that RLNC outperforms one of the best-known fountain codes in the literature: the Luby transform (LT) code [**?**], by demonstrating that when RLNC is used, fewer extra packets are needed to achieve the same probability of decoding, $1-\delta(E)$.

3.2.1 Related Work

Erasure codes were introduced for communication over an erasure channel in which a packet is either received without error or is lost ('erased'). A benchmark is the Luby Transform (LT) erasure code that achieves good performance with few extra packets needed for achieving good decoding probability and can be implemented with a fast and efficient decoding algorithm. Fundamentally, the LT code design problem maps to a set of random equations that are sparse, implying that the number of unknowns per equation is small, on average [**?**]. This sparsity allows for substantially more efficient encoding and decoding but the price paid is that (slightly) more than N packets are needed to reconstruct the source data. Designing a proper structure for the system of equations, so that both the number of additional packets and the coding times are simultaneously small, is a challenge.

Recently, NC techniques have also drawn significant attention because of their potential for improving the transmission efficiency in broadcast communications [69, 130, 3, 6]. In [130], several broadcast protocols utilizing NC were presented. The authors found the transmission efficiency of forward error correction (FEC) codes for wireless broadcast and compared it with that of NC. It was observed that when ARQ was used, FEC achieved

higher transmission efficiency than NC when dealing with a larger number of users, and vice versa for a smaller number of users. In other words, NC-based retransmission performed well under good channel conditions and a small number of users. Note that NC-based retransmission strategy is still based on ARQ. Our work is different because we do *not* assume any feedback channel between receivers and BS. We focus on finding statistical relation between number of transmission packet, M, and the outage fraction, β when there is no feedback from users.

3.2.2 System Model

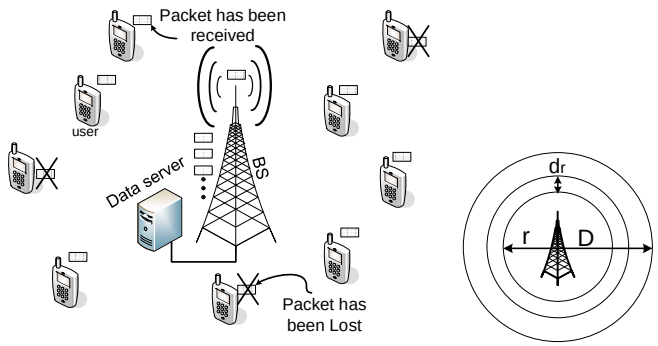
A BS, such as a digital TV tower or a Radio Base Station (RBS) in a cellular network, is serving K users in its coverage area a (common) source data file. The source file is divided into N *information packets*, $\{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N\}$, that is to be delivered to (almost) every user in the area with high probability. Without loss of generality, we assume that the information packets have equal length and can be modeled as vectors of r symbols, where each symbol is an element of finite field $\mathbb{F}_{2^q}$, i.e., $\mathbf{x}_i \in \mathbb{F}_{2^q}^r \forall i$. For convenience, assume that q divides the length of information packets (otherwise, zero padding is applied). Moreover, all packets are linearly independent vectors in $\mathbb{F}_{2^q}$ [123].

The BS seeks to broadcast the N information packets to the users via suitable encoding in the manner of a *rateless code*, *without* any feedback from users. We seek to find the probability that more than $1 - \beta$ of the users can decode all the N information packets when the BS transmits M coded packets. For example, consider the probability that 90% of the users ($\beta = .1$) decode the information when the BS transmits 100 coded packets ($M = 100$) for $N = 20$ information packets? Clearly, increasing M increases the decoding probability, but at the cost of larger service delay.

The users are assumed to be uniformly distributed with a density of ρ throughout the BS coverage area, over a radius D. Assuming K large, the user density and number of users is related via

$$\rho = \frac{K}{\pi D^2} \tag{3.6}$$

The channel between the BS and a node located at a distance of r from the BS, is



(a) (b)

Figure 3.21: A BS serving all the users in its coverage range of D with user density ρ. Users are locating in rings centered at the BS with infinitesimal width

assumed to be an erasure channel with a probability $p(r)$. Hence, any transmitted packet is received by the node with probability $p(r)$, independent of all other transmissions. The reception probability $p(r)$ depends only on the physical channel between the user and the BS and link layer parameters (such as the modulation or channel encoder/decoder), which will be discussed further in Section 4.1.6.

Random Linear Network coding

For the aforementioned erasure channels, fountain codes are a well-known solution to the problem of reliable broadcasting. In traditional fountain codes, source (information) packets are combined over a binary field (or Galois field $\mathbb{F}_2$) [**?**]. In our work, we propose to substitute via the use of Random Linear Network Coding (RLNC), that in turn, allow the use of higher fields, i.e., $\mathbb{F}_{2^q}$ for encoding. The BS transmits a sequence of encoded packets that are obtained by (random) linear combination of the N information packets, $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N$; the i-th transmitted packet is

$$\mathbf{y}_i = \sum_{i=1}^{N} \gamma_{i,k} \mathbf{x}_k = \boldsymbol{\gamma}_i^T \mathbf{X}, \tag{3.7}$$

where γ_is are called *network coding coefficients* that are *randomly* selected from the finite field $\mathbb{F}_{2^q}$. These coefficients are necessary for decoding and are hence inserted in the header

of the packet $\mathbf{y}_i$. As discussed in [35], network coding systems incur this additional header (that contains the network coding coefficients). However, if the size of the information packets is reasonably large, the additional overhead is acceptable.

When a receiver receives a broadcast packet, it extracts the NC coefficients from the header. The NC coefficients constitute a vector of length N, where each entry belongs to $\mathbb{F}_{2^q}$ (the vector itself is in $\mathbb{F}_{2^q}^N$). The receiver adds this packet to its buffer. Suppose that the receiver has N' packets in its buffer at any time. The relation between information packets and the buffered packets can be written as follows:

$$\mathbf{Y}^{(u)}_{N' \times r} = \mathbf{A}^{(u)}_{N' \times N} \cdot \mathbf{X}_{N \times r}, \tag{3.8}$$

where $\mathbf{Y}^{(u)}_{N' \times r}$ denotes encoded packets received thus far by node u, $\mathbf{A}^{(u)}_{N' \times N}$ is the matrix of NC coefficients and $\mathbf{X}_{N \times r}$ is the information matrix whose rows contain the information packets, i.e.,

$$\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2 \dots \mathbf{x}_N]^T, \mathbf{Y}^{(u)} = [\mathbf{y}_1\ \mathbf{y}_2 \dots \mathbf{y}_{N'}]^T,$$
$$\mathbf{A}^{(u)}_{N' \times N} = \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,N} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{N',1} & \gamma_{N',2} & \cdots & \gamma_{N',N} \end{bmatrix}.$$

When the matrix $\mathbf{A}^{(u)}$ is full rank, Eq. (3.8) has a unique solution and the information matrix $\mathbf{X}$ can be recovered from the received encoded packet matrix $\mathbf{Y}$. It has been shown that, if $N' = N + E$ (for any E), the matrix $\mathbf{A}$ is full rank with following probability [**?**]:

$$\begin{aligned} 1 - \delta(E) &= P(rank(\mathbf{A}^{(u)}_{N' \times N}) = N) = \\ &= \prod_{i=1}^{N} (1 - \frac{1}{(2^q)^{E+i}}) \\ &\geq 1 - N\, 2^{-q(E+1)} \end{aligned} \tag{3.9}$$

For $E = 0$, $\delta(E) \leq N\, 2^{-q}$, i.e., when a node receives any N packets, it is unable to recover the entire information N-set with probability at most $N\, 2^{-q}$. Clearly, this bound decreases exponentially with q, the size of finite field, and linearly increases with number of

information packets. Therefore, it is reasonable to assume that with NC broadcasting, if the node receives N encoded packets, then it can recover the information packets, $\{\mathbf{x}_1, ...\mathbf{x}_N\}$ with sufficiently high probability if q is chosen appropriately.

3.2.3 Statistical Analysis of NC Broadcasting

The BS transmits $M \geq N$ encoded packets; a receiver needs to successfully decode at least (any) N packets in order to be able to recover (with high probability) the original information. Therefore, the probability that a user located at a distance r from the BS is able to retrieve the original message, can be calculated as follows:

$$P_{rcv}(r) = \sum_{i=N}^{M} \binom{M}{i} p^i(r) \left(1 - p(r)\right)^{M-i}. \tag{3.10}$$

where $p(r)$ models the probability of reception of any individual packet over an erasure channel and will be discussed in Section 4.1.6.

Let $\boldsymbol{\mathcal{U}}$ be the random variable representing the number of users who can decode the full information. In this section, we first calculate the expectation and variance of $\boldsymbol{\mathcal{U}}$, and then derive the probability mass function for $\boldsymbol{\mathcal{U}}$, that highlights the relation between number of transmitted packets M, number of information packets, N, and outage fraction β.

First Order and Second Order Statistics

Consider users located in the (infinitesimal) annulus centered at the BS, at a radial distance r as in Figure 3.21-(b). The number of users in the ring is $2\pi\rho r dr$, among whom $\boldsymbol{\mathcal{U}}(r)$ are the number of users who can decode the full information. Using Eq. (3.10), the expected value $\mathbb{E}[\boldsymbol{\mathcal{U}}(r)]$, and the variance, $\mathbb{V}ar[\boldsymbol{\mathcal{U}}(r)]$, can be calculated as follows:

$$\begin{aligned} \mathbb{E}[\boldsymbol{\mathcal{U}}(r)] &= 2\pi\rho P_{rcv}(r) r dr \\ \mathbb{V}ar[\boldsymbol{\mathcal{U}}(r)] &= 2\pi\rho P_{rcv}(r)(1 - P_{rcv}(r)) r dr. \end{aligned} \tag{3.11}$$

Averaging over the radius r, the expected number of users who can recover the full information is given by

$$\mathbb{E}[\boldsymbol{\mathcal{U}}] = \int_0^D \mathbb{E}[\boldsymbol{\mathcal{U}}(r)] dr = \int_0^D 2\pi\rho P_{rcv}(r) r dr \tag{3.12}$$

and the corresponding variance of $\boldsymbol{\mathcal{K}}$ is

$$\mathbb{V}ar[\boldsymbol{\mathcal{U}}] = \int_0^D \mathbb{V}ar[\boldsymbol{\mathcal{U}}(r)]dr = \int_0^D 2\pi\rho P_{rcv}(r)(1 - P_{rcv}(r))rdr \tag{3.13}$$

Outage Fraction

First consider a *finite* coverage area of radius D that is divided into S equal concentric annuli as shown, i.e., the width of each annulus equals $\frac{D}{S}$ (Figure 3.21-(b)). The inner and outer radius of the i-th annulus is thus $\frac{(i-1)D}{S}$ and $\frac{iD}{S}$, respectively. We assume S is sufficiently large enough such that the users in i-th annulus experience almost the same experience channel and hence the decoding probability for any user within an annulus can be treated as identical $P(r_i = \frac{iD}{S})$. Let K_i be the number of users in the i-th ring, then

$$K_i = 2\pi\rho r_i dr \tag{3.14}$$

Let $\boldsymbol{\mathcal{U}}(r_i)$ be a random variable denoting the number of users in the i-th ring who can decode the full information. $\boldsymbol{\mathcal{U}}(r_i)$ follows binomial distribution, i.e.,

$$P(\boldsymbol{\mathcal{U}}(r_i) = n) = \binom{K_i}{n} P_{rcv}(r_i)^n (1 - P_{rcv}(r_i))^{K_i - n}. \tag{3.15}$$

A Binomial variable with probability of success p and number of trials N can be approximated in the limit $N\,p = \lambda$ as a Poisson variable with parameter λ. Therefore, treating the $\boldsymbol{\mathcal{U}}(r_i)$ henceforth as a Poisson variable with rate $\lambda_i = \lambda(r_i) = K_i P_{rcv}(r_i) = 2\pi r_i \rho P_{rcv}(r_i)$, the *total* number of the users that can recover the information packets is

$$\boldsymbol{\mathcal{U}} = \sum_{i=1}^{S} \boldsymbol{\mathcal{U}}(r_i) \tag{3.16}$$

The $\boldsymbol{\mathcal{U}}(r_i)$ are independent Poisson random variables with rate parameter λ_i, yielding the fact that $\boldsymbol{\mathcal{U}}$ is a Poisson variable with arrival rate $\lambda = \sum_{i=1}^{S} \lambda_i$ from well-known results. As S approaches infinity, we have:

$$\lim_{S\to\infty} \lambda = \int_0^D 2\pi r \rho P_{rcv}(r) dr \tag{3.17}$$

Therefore, the average number of users who can decode the information packets after M transmission is $\mathbb{E}[\boldsymbol{\mathcal{U}}] = \lambda$ which agrees with our derivation in Eq. (3.12). The Probability

Table 3.2: Simulation parameters for average successful decoding rate

Parameter	Symbol	Value
Num of Info packect	N	10/20
Num of coded packect	M	100-1000
Coverage radius	D	10 m
User density	ρ	0.1
Transmission power	P	10 W
Channel mode		Rayleigh fading
Path loss factor	α	2
rate factor for exp RVs	μ	15-50
Detection Thd	thd	5 W

Mass Function (PMF) for the number of users that can successfully decode the information is thus

$$P(\boldsymbol{\mathcal{U}} = n) = e^{-\lambda}\frac{n^{\lambda}}{n!}, \quad 0 \leq n \leq K \tag{3.18}$$

By using Equation (3.18), the probability that more than $1-\beta$ of the users can decode all the messages after M transmission of the BS can be expressed as

$$1-\eta \quad = \quad \sum_{n=(1-\beta)K}^{K} P(\boldsymbol{\mathcal{U}} = n) \tag{3.19}$$

3.2.4 Evaluation Results

In this section, we evaluate the accuracy of our approximation in Eq. (3.19) via computer simulation. Then, we explore and quantify how NC based Broadcasting outperforms LT codes over erasure channels. The complete list of simulation parameters and system block diagram are presented in Table 3.2.4 and Figure 3.22 respectively.

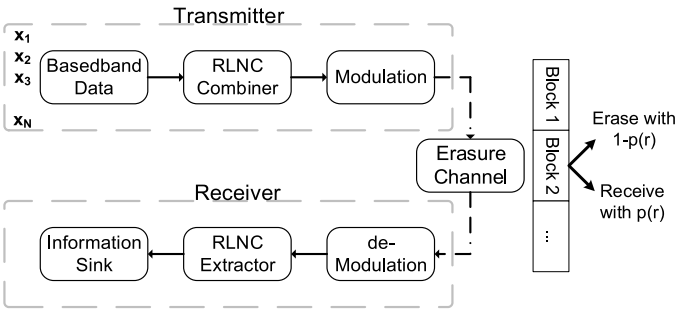


Figure 3.22: System block diagram and the block fading channel used in the simulation.

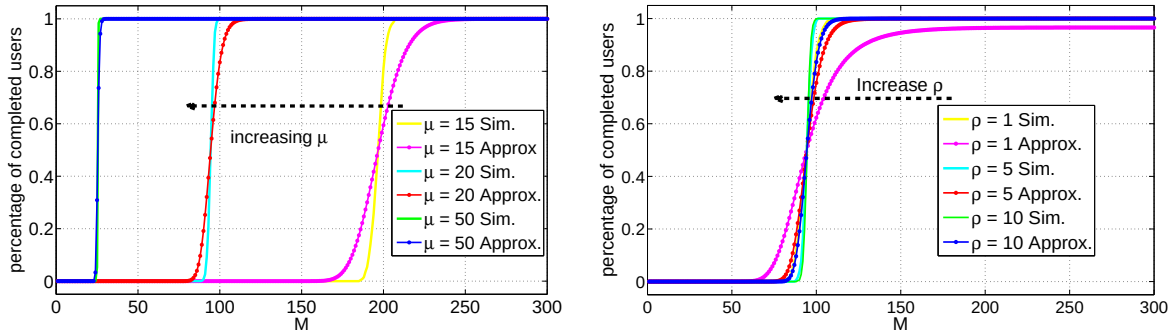
Broadcast Channel Model

We model the channel between the BS and users in its coverage service as independent Rayleigh block-fading channel. Each block is $r \cdot q$ bits (packet size) long and the fading parameter is assumed to be constant during a block transmission duration. If the fading is deep such that the instantaneous channel capacity is lower than the transmission bit rate (R_0), the packet is lost (or erased). This happens with a probability $p(r)$ when the receiver is located at a distance r from the BS. On the other hand, if the instantaneous channel capacity exceeds the transmission bit-rate, we assume that the packet is received correctly; this occurs with probability $1 - p(r)$. In the remain of the section we derive $p(r)$.

Thus the broadcast channel can be effectively modeled as follows: transmission power is P, bit rate over the channel is R_0, W is the system bandwidth, and $\frac{N_0}{2}$ is the noise power spectral density (PSD). The path loss factor is α, and the channel fading coefficients are Rayleigh random variables with unit mean, that are independent across different transmission intervals. For a user at distance r from the BS, the average receive $\overline{SNR}$ equals $\frac{r^{-\alpha}P}{N_0W}$. Let $\{z_{i,j}\}$ be a set of independent exponential random variables with the rate parameter μ. Then the $\overline{SNR}$ at user i during a time interval of j is $z_{i,j} * \frac{r^{-\alpha}P}{N_0W}$. Therefore, the achievable rate over the channel between the BS and the user i during time interval j is given by:

$$R_{i,j}(z_{i,j}) = \log(1 + z_{i,j} * \frac{r^{-\alpha}P}{N_0W}) \tag{3.20}$$

The probability that user i can successfully receive a transmitted packet during time



(**a**) (**b**)

Figure 3.23: Comparing the result of the Poisson approximation in Eq. (3.19) with computer simulation for $\beta = 0.1$. (a) The effect of fading parameter μ in our approximation model when $\rho = 10$. (b) The effect of user density ρ in our approximation model when $\mu = 20$.

interval j, $P_{i,j}(r)$, is:

$$\begin{aligned} P_{i,j}(r) &= Pr(R_{i,j} \geq R_0) \\ &= Pr(z_{i,j} \geq \frac{(e^{R_0} - 1)N_0 W}{r^{-\alpha} P}) \\ &= e^{\frac{-\mu(e^{R_0}-1)N_0 W}{r^{-\alpha} P}} \end{aligned} \tag{3.21}$$

Since the above probability is the same for any (block) time interval, we drop the time index j in what follows. Hence, a user at a distance of r to the BS can successfully receive a packet during any (block) time interval is given by

$$p(r) = e^{\frac{-\mu(2^{R_0}-1)N_0 W}{r^{-\alpha} P}} \tag{3.22}$$

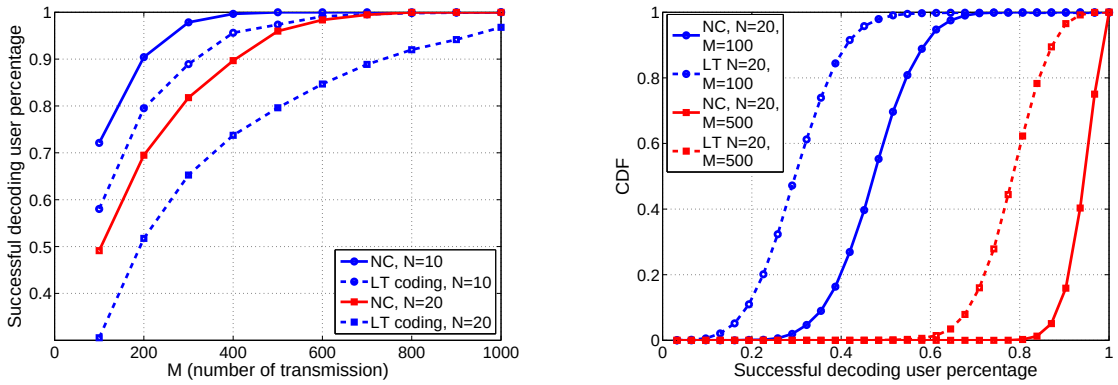
Accuracy of the Poisson Approximation

For a fixed number of transmissions M, Figure 3.23 presents the probability of the event that more than 90% of the users ($\beta = 0.1$) inside the coverage area can decode the entire information packets. As can be seen, our model in Eq. (3.19) follows the simulation results closely. The S-shape of the simulation-based plot is similar to the shape of the cumulative distribution function (CDF) of the Poisson distribution which agrees to our finding in Eq. (3.19).

The probability of success in Eq. (3.22) depends on other parameters such as path loss and fading. Figure 3.23-(a) presents the effect of the fading parameter μ on approximation accuracy in Eq. (3.19). For a fixed r, as μ increases, the reception probability, $p(r)$, also increases. In other words, increasing μ improves the erasure channel between the BS and users located at a distance of r from the BS. Consequently, a single broadcasted packet would be received by more number of users. Hence, the total number of encoded packets that BS needs to send out, M, in order to make sure that $1-\beta$ of the users will be able to decode decreases. In the limit, when $p(r)$ approaches 1, every packet would be received by all the users and therefore $M = 20$. This explains the sharp transition and shifting to the right for the plots in Figure 3.23-(a).

Successful Decoding Rate Comparison

In this section, using computer simulation, we compare the performance of proposed network coded broadcasting with LT fountain codes. The BS broadcasts M coded packets using binary phase shift keying (BPSK) without any channel coding; receiver nodes are always in listening mode; for each transmission, a node may or may not receive the transmitted packet successfully, as modeled in Eq. (3.22). After M transmissions, the nodes attempt decode the full information packet set. The percentage of the nodes which can successfully decode the N information messages are denoted by $1-\beta$. As can be seen from Figure 3.24, NC based broadcast always outperform LT encoding for a given successful decoding rate.



(**a**) (**b**)

Figure 3.24: (a) Average successful decoding rate comparison between NC and LT (b) CDF of successful decoding rate comparison between NC and LT.

Chapter 4

WIRELESS DATA SHARING WITH NETWORK CODING

The information dissemination problem, at its root, is a classical broadcast problem: sharing data residing at one node (source) with all others (destinations) in the network. The significance of this problem can be readily gauged by the extensive history of prior work on this topic, which has contributed to the design of networked communications with protocol stacks that support efficient broadcasting.

The information sharing problem was initially studied exclusively for wired ad-hoc networks as a radically different and appealing alternative for the traditional client-server model [54]. Such networks are possess the following distinguishing characteristics:

1. They (usually) do not have a centralized entity for facilitating computation, communication and time-synchronization

2. The full network topology is not known to the nodes of the network

3. The computational power of the nodes may be limited

These constraints motivate the design of simple decentralized algorithms where each node exchanges information with only a few of its immediate neighbors at each step. Mimicking the spread of a rumor in a society, gossip protocols constitute an important decentralized class of algorithms whereby each node acts based only on its own state information and those of its neighbors [43].

Of late, there has been considerable interest in the sharing problem when nodes are inter-connected wirelessly such as personal mobile devices (cellular phones or PDAs). In this chapter, we investigate the application of network coding in information sharing. We will consider an ad-hoc network in which nodes are willing to share their content with the rest of the network. We will show that using NC has the advantage of decreasing the

dissemination latency, defined as time needed to diffuse information to the entire network. **Then, we move to Vehicular Adhoc networks (VANET). We will show**

4.1 All-to-All Data Dissemination

Of late, a more modern version of the one-to-many (broadcast) problem has gained prominence; this is typically referred to as a *data sharing* among multiple peer-to-peer (p2p) nodes, or the *all-to-all* problem. This problem arises when each node in a network obtains only *a fraction of the total information* (e.g. part of a video-on-demand file or a software update) desired collectively by all. In a simplified version of the all-to-all data-dissemination problem, a source file desired by all, is divided into N mutually exclusive *information packets*, and each packet is stored at exactly one node in the network [100, 43]. Every node wishes to acquire the remaining $N-1$ pieces of the source file; the order in which each node receives the remaining information packets is not relevant.

Traditionally, the data dissemination problem over a decentralized network architecture has focussed on the impact of the *dissemination algorithm* designed to optimize a performance metric such as the dissemination latency, i.e., the time required for *all nodes* to acquire the entire file [127]. Our contribution is to analyze the impact of network coding to this problem. One of the most significant benefits of our approach is that nodes do *not* need extra information from other nodes regarding the state of the network [170] to accomplish dissemination, significantly reducing communication overhead.

The prior analytical models used for this problem have largely suffered from unrealistic assumptions that are unsuited to a wireless network - notably that of pure fail/success $(0-1)$ whereby either each transmission is either successfully received by *all* neighbors or fails. Our analysis advances the state of the art by using a more appropriate link model whereby for each broadcast, sink nodes successfully receive the transmitted packet with a reception probability that depends on the nodes respective locations. Note further that in all broadcast wireless networks, the role of the multiple access or MAC protocol is fundamental to managing interference. In our work, we assume orthogonal access, such as *random TDMA*, thereby obviating the need to model multi-access interference. Such a set-up is admittedly limited but nonetheless useful, as it encompasses practical network scenarios

such as a single-cell CSMA/CA [88, 14].

Finally, we emphasize that we only focus on the dissemination latency (i.e., the time needed for all nodes to communicate and receive the entire file) and do *not* consider the latency caused by encoding/decoding which has been studied separately in the literature [162]. In fact, [160, 114] explores design of a sparse network coding matrix that decreases encoding/decoding time significantly. Clearly, the net latency of data dissemination is the sum of our result and the encoding/decoding time.

4.1.1 Related Works

Data Dissemination in Wireless Networks

Data dissemination initially among wired clients first emerged as an appealing alternative to the traditional client-server model [54]. Due to the exponential growth of wireless networks, there has been considerable interest in the sharing problem between wireless nodes (such as cellular phones or PDAs) that are interconnected *wirelessly*. Clearly, data dissemination in wireless networks differs from the wired scenario in significant ways - the broadcast nature of wireless contributes (potentially) to multi-access interference. As a result, simultaneous reception of multiple packets in a wireless network is typically limited. Furthermore, wireless links are half-duplex due to hardware constraints; i.e., a node may not simultaneously transmit and receive due to the lack of sufficient isolation between the two paths.

Despite these differences, gossip-based protocols - originally applied for dissemination over wired networks - have remained as the algorithmic family of choice in wireless networks for dissemination, as they inherit the desired balance between simplicity and robustness [7]. Some modifications have been applied to make these protocols consistent with wireless properties. For instance, when a node transmits a packet in a wireless network, every node within its transmission range receives the packet [135, 51] in contrast to to gossip algorithms in wired networks that are based on unicast communication with a randomly chosen neighbor. However, the main feature of gossip-based algorithms is preserved - every node in the network acts simply based on its *local* state information.

Network Coding for Dissemination

A relatively new approach to decentralized data dissemination is based on network coding, first introduced for multicasting in wired networks [5, 112, 87, 85, 103]. Authors in [42] exploit random (linear) network coding within a gossip-based dissemination protocol for wired networks and show that, in a fully connected network, network coding can improve dissemination latency at the cost of a small overhead associated with each packet. In fact, they show that, in a network with N nodes, data can be diffused in $O(N)$ time using a coding-based gossip algorithm rather than $O(NlogN)$ without network coding. Mosk-Aoyama and Shah extend the result in [42] to a wired network with arbitrary topology [128] and show how the graph topology affects the performance of the coding-based dissemination algorithm.

For wireless networks, network coding was first exploited for wireless broadcasting – treated as a one-to-many data-dissemination problem – using simple XORs that demonstrated bandwidth efficiency over traditional wireless broadcast approaches. Recently, [67] used network coding within a gossip-based algorithm (rumor-spreading) to diffuse information in an ad-hoc wireless network. This was continued by [6] who studied performance of such dissemination due to real-world MAC schemes.

There is a plausibility argument as to why network coding provides substantial benefits for data dissemination. As mentioned before, gossip algorithms impose restrictions on the information possessed by a node; i.e., every node has only a local view of the system state at any time. This is analogous to the initial state in our scenario, where each node has only a (small) unique part of the full file, and seeks to gather the other pieces. With time, a node gathers some of the other pieces but does not have any information about what pieces the neighboring nodes may possess. It is natural that, at any time, different nodes in the network will have acquired (at least some) non-overlapping pieces. Intuitively, this suggests that if each node encodes all the data it presently contains (via network coding) and broadcasts it, recipient nodes will have acquired coded versions about the missing pieces. In time, each node will be able to decode the full file, after a sufficient number of such encoded packet transmissions from other nodes, as subsequently described.

4.1.2 System Model

As usual, a network graph is denoted as $G(V, E)$, with $|V| = N$ nodes and links $E \subset V \times V$. We assume that the network is slotted (i.e., all nodes are synchronized) for simplicity and that all transmissions occur synchronously with a common clock. Further, without loss of generality, we assume that during each time slot, a node $v \in V$ can broadcast exactly one packet. When node v broadcasts, node $u \in V$ receives the signal correctly with probability P_{vu}. We consider an interference-free (orthogonal) access that allows only one node to transmit at a time. This includes, among others, a single-cell 802.11-type infrastructure network based on CSMA/CA if all nodes lie within the (common) carrier sensing range. The probability of a node capturing the common channel at any time is assumed to be uniform among all nodes. Finally, we only consider networks with fixed topology, i.e., where the topology and link capacities do not change over time.

4.1.3 Data Dissemination using Network Coding: Arbitrary Network Topology

Assume that each node u initially has a single *information packet* x_u to be shared with every other node in the network. Hence, the set of unique (information) packets in the network, initially and at all subsequent times, is given by $\{x_1, \ldots, x_N\}$ for a network with N nodes. Each information packet is a vector of r symbols, where each symbol is an element of a finite field $\mathbb{F}_{2^q}$, i.e., $x_u \in \mathbb{F}_{2^q}^r$ for each node $u \in V$. For convenience, assume that q divides the length of packets transmitted (otherwise, zero padding is applied). Moreover, all packets are linearly independent vectors in $\mathbb{F}_{2^q}$, reflecting the fact that nodes have different information to share[1]. The results and derivations presented in this manuscript can be extended to a case when some nodes have more than one message and some have none or when all the messages are there with one particular node to start with.

With time, each node receives a sequence of linear combinations of information packets at the other nodes. Hence, after a sequence of broadcasts, node $u \in V$ possesses a set of coded messages, $S_u(t)$ at time (slot) t.

$$S_u(t) = \{m_1, m_2, ..., m_{|S_u(t)|}\}, \tag{4.1}$$

[1]For more information about linear independence in finite fields, refer to [89, 123].

Each message m_i is a linear combination of the underlying *information packets*[2], initially possessed by the nodes and can be represented as

$$m_i = \sum_{k=1}^{N} \alpha_{i,k} x_k = \boldsymbol{\alpha}_i^T \mathbf{x}, \quad i = 1, 2, ..., |S_u(t)|, \tag{4.2}$$

where some of the coefficients $\alpha_{i,k}$ may be zero (if the corresponding information packet is not present at the transmitting node at that time). For each message m_i, $\alpha_{i,k}$s are called its *network coding coefficients*, and they are available through the header of the packet containing m_i. As discussed in [35], in a network coding system, each packet consists of two parts: a header that contains the network coding coefficients and a body that carries the encoded message. This header is a price to pay to use the network coding. However, if the size of the information packets (and hence the size of the messages) is reasonably large, this overhead is negligible. That being said, for each message m_i at node u, network coding coefficients are available.

Clearly, $S_u(t)$ (set of messages at node u at time t) spans a subspace in $\mathbb{F}_{2^q}^r$, as observed by rewriting Eq. (4.2) in the following form:

$$\mathbf{m}_u(t) = \mathbf{A}_u(t)\, \mathbf{x}, \tag{4.3}$$

where $\mathbf{A}_u$ is the coefficient matrix consisting of NC coefficients and $\mathbf{x}$ is the N-vector of all the information packets in the network, given by

$$\begin{aligned} \mathbf{x} &= [x_1 \;\; x_2 \;\; \ldots \;\; x_N]^T, \\ \mathbf{m}_u(t) &= [m_1 \;\; m_2 \;\; \ldots \;\; m_{|S_u(t)|}]^T, \\ \mathbf{A}_u(t) &= \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \ldots & \alpha_{1,N} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{|S_u(t)|,1} & \alpha_{|S_u(t)|,2} & \ldots & \alpha_{|S_u(t)|,N} \end{bmatrix}. \end{aligned} \tag{4.4}$$

is defined by $\alpha_{i,j}$, the j-th coefficient used in the i-th message m_i received by node u.

From the received broadcast messages, a node can decode all the information packets at time t if the rank of its coefficient matrix, $\mathbf{A}_u(t)$, equals N, the total number of unique

[2]Recall that each node u starts with a single information packet x_u to share with the rest of the network.

information packets in the network. In the next section, we use this observation to put an upper bound on dissemination latency in the network.

If node u captures the channel at any time slot t, it transmits a linear combination of all the messages that it currently contains, as follows:

$$y_u(t) = \sum_{i=1}^{|S_u(t)|} \beta_i m_i, \tag{4.5}$$

where $y_u(t)$ denotes a transmitted packet at time t, $S_u(t)$ is a set of messages node u contains at time t, and $\beta_i, \;\; i = 1, 2, ..., |S_u(t)|$, are random finite numbers selected from $\mathbb{F}_{2^q}$. Using (4.3), the transmitted message by node u at time t can be written in terms of *information packets* as follows:

$$\begin{aligned} y_u(t) &= \boldsymbol{\beta}\mathbf{m}_u(t-1) \\ &= \boldsymbol{\beta}\mathbf{A}_u(t-1)\mathbf{x} \\ &= \boldsymbol{\alpha}'\,\mathbf{x}, \end{aligned} \tag{4.6}$$

where $\boldsymbol{\alpha}' = \boldsymbol{\beta}\mathbf{A}_u(t-1)$ presents network coding coefficients of information packets used by node u to transmit a message at time t. As mentioned before, these coefficient are send a long with $y_u(t)$.

4.1.4 Stopping Time

The data dissemination algorithm terminates when *all* nodes are able to decode the broadcast messages to recover the underlying N set of information packets, which happens when Eq. (4.3) for all $u \in V$ has a unique solution, i.e., when the coefficient matrix at each node has full rank N. Hence, the stopping time $\mathcal{T}$ is defined as the first time when the full rank condition is achieved:

$$\mathcal{T} = \min_t \{rank(\mathbf{A}_u(t)) = N \;\forall u \in V\}. \tag{4.7}$$

Because matrix $\mathbf{A}_u$ has rank N if and only if $S_u(t)$, the subspace scanned by messages in u at time t, has dimension N. The stopping time can also be defined as follows:

$$\mathcal{T} = \min_t \{dim(S_u(t)) = N \;\forall u \in V\}. \tag{4.8}$$

Clearly, $\mathcal{T}$ is an integer random variable over $[N, \infty)$[3]. We next seek the expected value $\mathbb{E}[\mathcal{T}]$ as a performance metric for algorithm design. In general, $\mathbb{E}[\mathcal{T}]$ is difficult to compute; hence, we resort to bounds.

4.1.5 upper bound for Mean Stopping Time

As mentioned above, $S_u(t)$ is the set of packets that node u possesses at time t. Each packet is an r vector in a finite field $\mathbb{F}_{2^q}$ (i.e., it belongs to $\mathbb{F}_{2^q}^r$). Moreover, we assume that packets in $S_u(t)$ are independent. Hence, the cardinality of $S_u(t)$ is the same as its dimension.

By our formulation, every node initially starts with an information packet. In other words, at $t = 0$, there is one and only one (independent) packet in $S_u(0)$, i.e., $dim(S_u(0)) = 1 \; \forall u \in V$. With time, the information spreads to all nodes upon sharing via broadcast, resulting in a final per node dimension of N at the time of stopping. Hence, each node dimension is raised by $N-1$ during the information dissemination, and the overall dimension increase among all the nodes is $N(N-1)$. Let us define $D(t)$ as the total dimension increase (among all the nodes) at time t. Obviously, $D(t)$ can be written as follows:

$$D(t) = \sum_{u \in V} dim(S_u(t)) - N. \tag{4.9}$$

Clearly, the information has spread to all nodes when $D(t) = N(N-1)$. Now, let $\mathcal{T}_i$ denote the number of time slots until the total dimension increases by $i(N-1)$. It can be written as follows:

$$\mathcal{T}_i = \min_t \{D(t) \geq i(N-1)\}. \tag{4.10}$$

By definition $\mathcal{T}_0 = 0$ and the information spreads to all the nodes at $\mathcal{T}_N$, i.e., $\mathcal{T} = \mathcal{T}_N$.

The following lemma gives an upper bound for the probability of the message sets at two nodes spanning the same subspace when $t = \mathcal{T}_i$.

---

[3]To diffuse N packets, we need at least N transmissions, which in wireless networks require at least N time slots.

lemma 3 *At $t = \mathcal{T}_i$, the probability that any two nodes (e.g., u, v) have the same subspace can be bounded from above as follows*[4]*:*

$$P(S_u(\mathcal{T}_i) \equiv S_v(\mathcal{T}_i)) \leq \sum_{k=1}^{N-1} \min\left(\frac{N-i}{N-k-1}, \frac{i}{k}\right) P(dim(S_v(t)) = k+1). \tag{4.11}$$

Proof: See A.3.

Intuitively, the probability that two nodes have the same subspace at $\mathcal{T}_i$ is calculated using the law of total probability: probability that one of them has dimension k multiplied by the probability that the other one has the same subspace. The latter can be calculated by recalling that $\mathcal{T}_i$ represents the number of time slots at which the total dimension increase is $i(N-1)$. We next explore the dimension of node $u \in V$ at time $\mathcal{T}_i$. The following lemma gives an upper bound on the probability of node u having dimension k at $\mathcal{T}_i$.

lemma 4 *The probability that node u has dimension k at time $\mathcal{T}_i$ is given by:*

$$\begin{aligned} &P(dim(S_u(\mathcal{T}_i)) = k) = \\ &\frac{\sum_{j=0}(-1)^j C(N-1,j) C(i(N-1)+N-2-jN-k, N-2)}{\sum_{j=0}(-1)^j C(N,j) C(i(N-1)+N-1-jN, N-1)}. \end{aligned} \tag{4.12}$$

Proof: See A.3.

By definition, at time $\mathcal{T}_i$, the dimension has increased by $i(N-1)$ and this comes from N nodes. Now the probability that one node has dimension increase k is in fact the traditional ball and bins problem.

The following theorem provides an upper bound on expected number of time slots between $\mathcal{T}_i$ and $\mathcal{T}_{i+1}$.

Theorem 14 *For $0 \leq i \leq N-2$:*

$$\mathbb{E}[\mathcal{T}_{i+1} - \mathcal{T}_i] \leq \frac{N-1}{\frac{1}{2N}(\sum_{u,v \in V} P_{uv})(1 - P(S_u(\mathcal{T}_{i+1}) \equiv S_v(\mathcal{T}_{i+1})))}, \tag{4.13}$$

and for $i = N-1$

$$\mathbb{E}[\mathcal{T}_N - \mathcal{T}_{N-1}] \leq \frac{N(N-1)}{\frac{1}{2N}(\sum_{u,v \in V} P_{uv})}. \tag{4.14}$$

[4]Note that in Eq. A.74 (and in similar cases henceforth), $S_u(t) \equiv S_v(t)$ indicates the equality between two subspaces, the one spanned by elements in S_u and the one spanned by elements in S_v.

Proof: See A.3.

Finally, the following theorem–which is the main result of this section–gives an upper bound on the stopping time $\mathcal{T}$.

Theorem 15 *Let $\mathcal{T}$ be stopping time. Then*

$$\begin{aligned} \mathbb{E}[\mathcal{T}] &= \mathbb{E}[\mathcal{T}_N] \\ &\leq \frac{N-1}{\frac{1}{2N}\sum_{u,v\in V} P_{uv}} \Bigg(\sum_{i=1}^{N-1} \frac{1}{1 - P\big(S_u(\mathcal{T}_i) \equiv S_v(\mathcal{T}_i)\big)} + N \Bigg). \end{aligned} \tag{4.15}$$

Proof: See A.3.

Note that to numerically calculate the upper bound given in Eq. 4.15, we need to combine the results in Eqs. A.74 and 4.12. Since each node is initialized with a single packet, it needs to acquire the remaining $N-1$ packets from other nodes for the process to terminate. Hence a total of $N(N-1)$ successful packet transmissions must occur. Due to the broadcast nature of wireless, multiple receive nodes hear each transmission and may decode the transmitted packet (according to their reception probability; higher reception probability results in a higher chance of decoding). Therefore, the number of time slots required is inversely proportional to reception probability as captured by the first part of the upper bound. The second part of the upper bound represents the fact that a successfully received packet v at a node is only useful if it does not belong to the subspace spanned by existing packets; i.e., it is 'innovative'. Again intuitively, the probability of a packet being innovative at a node decreases with time, as the subspace spanned by existing packets is always monotonic non-decreasing.

In an interference-free network where only one node transmits in each time slot, Theorem 15 gives an upper bound on dissemination latency when NC is used. As expected, the upper bound depends on reception probabilities and the number of nodes inside the network. In the following two lemmas, we consider two extreme cases, a fully connected wireless network and a sparsely connected network. These two cases illustrate how reception probability affects dissemination based on NC in a wireless network. We show that no matter what the

reception probabilities are, the average stopping time is between $O(N)$ and $O(N^2)$ as long as the network stays connected.

A wireless network $G(V, E)$ is considered fully connected when for every two nodes $u, v \in V$, $P_{uv} > 0$. In other words, in a fully connected network, every node is within the transmission range of all the others.

lemma 5 *In a fully connected wireless network in which every node can receive a packet from any other node in the network with a nonzero probability, the average stopping time is of order $O(N)$ when network coding is applied.*

proof: See A.3.

Above, the corollary is consistent with the result of the data dissemination in a wired network when network coding is adapted. The authors in [42] show that the stopping time increases linearly with the size of the wired network (i.e., it is of order $O(N)$) if the network is fully connected; i.e., each node is connected to every other node.

A wireless network $G(V, E)$ is sparsely connected when the network is connected and every node is in transmission range of at most two other nodes. An example of a sparsely connected wireless network is a linear network when each node can only communicate with its close neighbors.

lemma 6 *In a sparsely connected wireless network, the average stopping time is of order $O(N^2)$ when network coding is applied.*

proof: See A.3.

Clearly, Lemma 5 gives the best achievable time [5] (smallest number of time slots), and Lemma 6 gives the worst time (largest number of time slots needed). In other words, the above two lemmas show that for data dissemination in a wireless network with N nodes, the average stopping time is between $O(N)$ and $O(N^2)$, independent of the underlying reception probability of the nodes.

---

[5] especially when p=1

4.1.6 Performance Evaluation

In this section, we present results from a system simulation conducted using MATLAB R2008b that conforms to the data dissemination model described. The objective is to explore tightness of our stopping time upper bound, given in Eq. (4.15), for data dissemination in a wireless network with general topology and interference-free random access scheduler (i.e., only one node captures the channel and transmits, while all others listen). At the physical layer, we use QAM modulation with no channel coding. The results reported are based on simulations conducted for two simple and useful topologies: regular linear and 2-D grid. Such structured topologies help with better understanding of the model behavior as the number of nodes increase.

4.1.7 Reception Probability

We assume that all nodes use the same transmission power $\mathcal{P}$ and modulation scheme to broadcast packets. The wireless channel undergo Rayleigh envelope fading, and the path loss exponent is η. Assume node u sends a packet to node v which is $d(u,v)$ far away. Let $s_{u,v}$ be the received power at v in the clear channel (i.e., no interference). Then $s_{u,v}$ follows an exponential random variable with mean $\mathcal{P} \cdot d(u,v)^{-\alpha}$ given by the following probability distribution function (pdf):

$$f(s_{u,v}) = \frac{1}{\mathcal{P}d(u,v)^{-\eta}} exp(-\frac{s}{\mathcal{P}d(u,v)^{-\eta}}). \tag{4.16}$$

Node v, the receiver, can decode successfully the packet transmitted by u if its received Signal-to-Noise (SINR) ratio exceeds a threshold[6]:

$$P_{u,v} = Pr(\frac{s_{u,v}}{\mathcal{N}_0} \geq z), \tag{4.17}$$

The $\mathcal{N}_0$ is the variance of additive white Gaussian noise, assumed to be 4×10^{-14} at all the receivers [7], and z is the capture reception threshold whose value depends on channel coding and modulation.

---

[6] As mentioned before, we are assuming an interference free communication

[7] Noise Power is calculated for the bandwidth of 10MHz and in temperature 300K. This value, however, does not affect the result of the simulation.

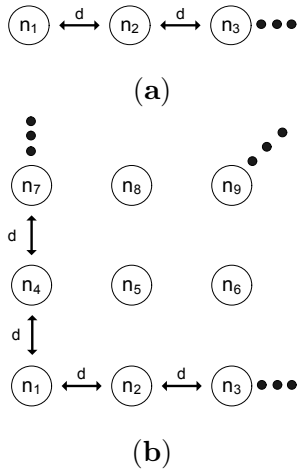


Figure 4.1: Example of a network with (a) linear topology (b) grid topology

4.1.8 Linear Grid

Here, nodes are located in a line, with equal distance between neighbors. At first, we let the transmit power remain fixed and increase the size of the network by adding more nodes. Figure 4.2 presents the simulation result and the analytical upper bound for the linear network. As one can see, the upper bound given in Eq. 4.15 closely follows the trend of the simulation results.

When there are only a few nodes in the network, stopping time has a linear relation with the number of nodes in the network. However, when the size of the network keeps increasing, the linear relation is not valid anymore. This is consistent with our findings in Lemmas 5 and 6. For a small number of nodes in the network, nodes are in transmission range of each other; i.e., a transmitted packet is heard by all of the nodes in the network (with nonzero probability $P_{uv} > 0$); hence, the stopping time is $O(N)$. On the other hand, when the size of the network keeps expanding, after a while we have $P_{uv} = 0$ for some nodes in the network and that affects the trend of the dissemination delay. In Figure 4.2, after $N = 30$, the stopping time (from both the simulation and analytical result) starts to increase nonlinearly. In fact, one can see that the stopping time is $O(N^2)$ after $N = 30$.

Finally, we change transmission power to see its effect on dissemination latency and the

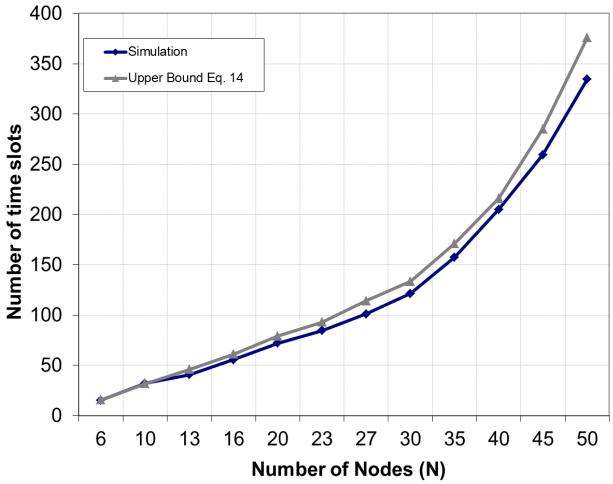


Figure 4.2: Both the analytical upper bound and the simulation result are increasing linearly at first, and after $N = 30$ they start increasing non-linearly versus the number of nodes in the network, $d = 30$, $\mathcal{P} = 20 \times 10^{-6}$, $\mathcal{N}_0 = 4 \times 10^{-14}$.

accuracy of the upper bound in Eq. (4.15). The result is presented in Figure 4.3. Clearly, decreasing transmission power reduces nodes' coverage and results in increased stopping time. However, the relation between the stopping time and the transmission power is very interesting and is sort of hidden in Eq. (4.15).

For a fixed network, we start with $0dB$ power and decrease it to $-40dB$. At first, nodes are in transmission range of each other. In that case, the relation between the transmission power and the stopping time is linear. However, after a point, nodes start falling out of the coverage range of each other. When that happens, stopping time starts increasing nonlinearly with transmission power. Although our formula in Eq. 4.5 does not show this relation directly (counter to the result with size of the network), it has the same trend as the simulation results.

4.1.9 2-D Grid

In a 2-D grid topology, nodes are located on a equispaced 2-D lattice as depicted in Figure 4.1-b. As for the linear network, we first let the transmission power remain fixed while

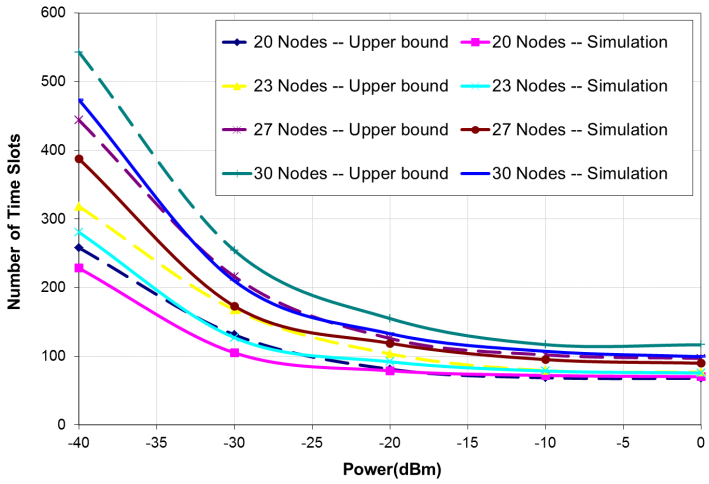


Figure 4.3: Analytical upper bound and simulation results versus nodes' transmission power for linear network, $\mathcal{N}_0 = 4 \times 10^{-14}$.

increasing the size of the network by adding more nodes. Figure 4.4 presents the simulation result and the analytical upper bound for the grid network with five different sizes.

In an $m \times n$ grid network with equispaced d, the distance between every two nodes is less than or equal to $d\sqrt{m^2 + n^2}$, which happens to be smaller than the transmission range of all the nodes in our simulation. In other words, for the fixed transmission power, each node can hear from all other nodes with nonzero probability. It is for this reason that the stopping time has a linear trend with the size of the network (Theorem 5).

Finally, for different transmission power, analytical upper bounds and simulation results are presented in Figure 4.5. As we argued for the network with linear topology, although the relation between the transmission power and the dissemination latency can not be directly inferred from the proposed upper bound in Eq. (4.15), it follows the simulation results by the same trend.

4.1.10 Advantage of Network Coding

In this subsection we compared the dissemination latency with network coding using computer simulation to a baseline *random* non-NC algorithm. The non-NC approach - termed

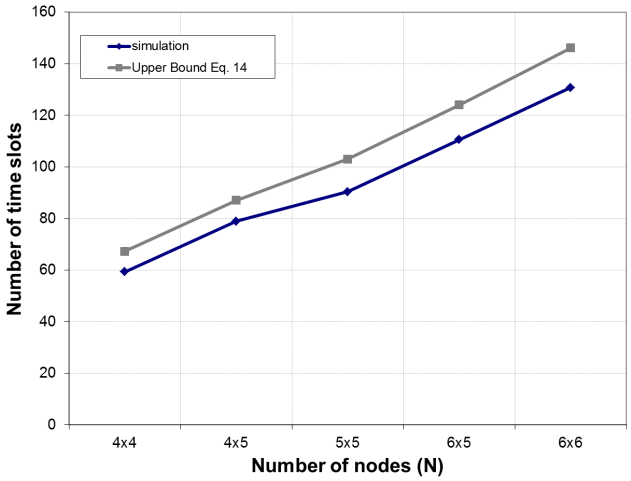


Figure 4.4: Analytical and simulation results when size of network is changing and nodes' transmission power is fixed for grid network, $d = 30$, $\mathcal{P} = 20 \times 10^{-6}$, $\mathcal{N}_0 = 4 \times 10^{-14}$. Since nodes are in coverage range of each other dissemination latency increases linearly as we show in Theorem 5.

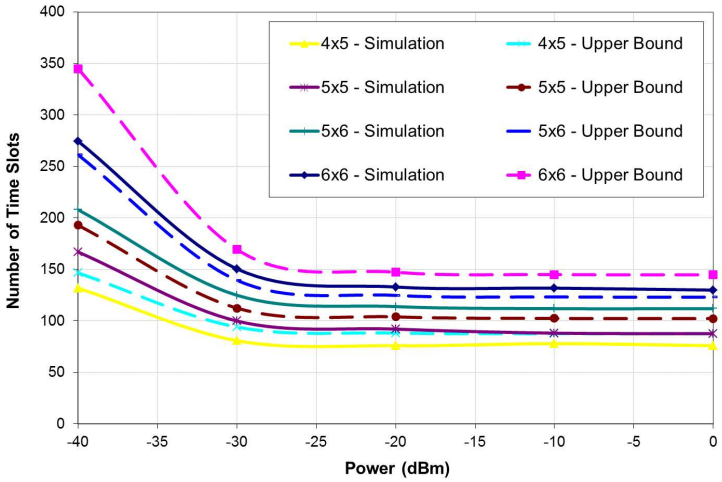


Figure 4.5: Analytical upper bound follows the simulation results trend when nodes' transmission is changing power for grid network, $\mathcal{N}_0 = 4 \times 10^{-14}$.

Table 4.1: Stopping Time for dissemination algorithm in a linear network with and without network coding. $d = 30$, $\mathcal{P} = 20 \times 10^{-6}$

# nodes	23	27	30	35
NC-based	84.46	100.94	121.54	157.59
random-selection	1189.6	2180.8	3148.9	3713.4

Table 4.2: Stopping Time for dissemination algorithm in grip network with and without network coding. we have $d = 30$, $\mathcal{P} = 20 \times 10^{-6}$

# nodes	4×4	4×5	5×5	6×4	6×6
NC-based	59.34	78.88	90.4	110.5	130.72
random-selection	684.3	1044.4	2275.8	2836.7	3724.4

random selection - operates as follows: whenever a node captures the channel it *randomly selects* an information message from its buffer and broadcasts the selected packet. Table 4.1 compares the mean time needed to diffuse data in a linear network with network coding vs. the random selection algorithm. The performance of the random selection algorithm and NC-based algorithm for a 2-D grid topology is compared in Table 4.2. As expected, network coding based dissemination outperforms the random selection algorithm by an order of magnitude. The reception probabilities are calculated as explain in Section V.

4.2 Collaborative Downloading in VANET using Network Coding

Recent development and standardization in vehicular ad hoc networks (VANETs) [79] have motivated the increasing interest in data services for in-vehicle consumption, such as 'commerce and entertainment-on-the-wheel' [2]. Hence, in the near future, the number of vehicles equipped with wireless communication devices is poised to increase dramatically, i.e., we are moving closer to an intelligent transportation system. Such a system would provide a wide variety of applications: local information pushed to vehicles (e.g., traffic notification, map updates, location-based advertisements); or specific data pulled from Internet servers (e.g.,

neighborhood parking, reviews of local restaurants, and video clips of local attractions) [183].

In current intelligent transportation systems, services are provided to vehicles using existing wide-area cellular infrastructure (3G/4G) and/or the roadside infrastructure based on Dedicated Short Range Communications (DSRC) links[8] proposed for VANET networks [2]. However, both of these approaches have their own challenges: the modest data rate of present 3G links and the associated high cost of data downloading on one hand, and the intermittent hotspot-type roadside coverage envisaged with DSRC on the other [95].

The above challenges lead to the following simple premise for content dissemination, called collaborative downloading: if the content available to a subset of vehicles is also desired by (many) others in the network, peer-to-peer content distribution using vehicle-to-vehicle (V2V) ad hoc communications is time and cost efficient.

Collaborative downloading is a data dissemination protocol for distributing information among all nodes inside the network [104], [171], and it has attracted a lot of attention in the VANET community in the past few years [70]. In general, data dissemination in VANET networks consists of two phases [92][93]:

1. Roadside-to-Vehicle (R2V) phase: in this phase, vehicles are communicating with a base station (located in specific location on the road) to receive data.

2. Vehicle-to-Vehicle (V2V) phase: when the vehicles are out of the coverage of the BS, they try to exchange information between each other. If this phase is completed, all the nodes have the same data.

To better understand the concept of collaborative downloading, consider the scenario in Figure 4.6 in which a group of vehicles equipped with DSRC radios are connected to the Internet via roadside DSRC base stations. Assume that these vehicles have a common interest in a large file on the Internet. The intermittency arises from the sparsity of the roadside DSRC base-stations and the hotspot nature of coverage. As a result, data is

[8]DSRC in North America is based on 802.11p wireless link access in 5.9 GHz band, subsequently folded into the IEEE 1609 WAVE standards.

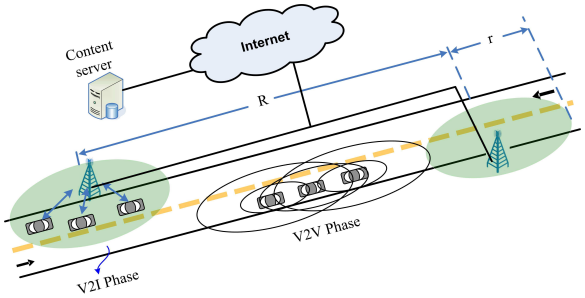


Figure 4.6: System diagram of collaborative downloading among vehicles on the road.

downloaded during the (short) intervals of radio connectivity from the infrastructure to the vehicle. Thus each vehicle in a platoon only obtains a fraction of the overall file in each contact duration. In principle, each vehicle could download the whole file through multiple contacts, requiring significant latency due the sparsity of the contacts. A more effective alternative is for the vehicles to form a coalition for sharing their respective pieces after each round of contacts (collaborative downloading) when they are out of coverage (V2V phase). Here, our focus is on R2V phase. We assume a perfect V2V transmission, i.e., data dissemination is complete after V2V phase. In other words, we assume that the vehicles have the same information after V2V phase is over.

In the state-of-the-art algorithm for collaborative downloading, nodes need to communicate with the BS to let it know which packets they are interested in. That means a partial time of vehicle to BS connection (in R2V phase) has to be assigned to signaling from the vehicles to the BS[9]. Moreover, this scheme requires a lot of synchronization and handshaking between the BS and the receiving vehicles. In this work, we propose to use network coding [5, 112, 64] in downloading data from the infrastructure. Our algorithm omits any kind of signaling from vehicles to the BS. Moreover, we analytically show that the expected time needed to disseminate data from infrastructure to all vehicles is slightly less by using NC. In other words, NC slightly decreases the downloading delay in addition to removing any need for uplink communication.

Network coding has recently been applied to many data dissemination problems in wire-

[9]R2V links are half-duplex, i.e., the base station can not receive and transmit simultaneously.

less networks. In [67], authors use a gossip-based algorithm (called rumor-spreading) to diffuse information in an ad-hoc network. Their work is continued by [6] who study the performance degradation due to actual MAC schemes.

4.2.1 Roadside-to-Vehicle Phase:System Model

Consider a platoon of N vehicles interested in a certain file comprising of M packets $x_1, x_2, \ldots x_M$. Suppose the network infrastructure (i.e., all the roadside base stations) possesses that common file. The goal is to distribute this file to all vehicles in the network. This is achieved by cycling through a succession of R2V+V2V phases. During each R2V phase, we assume that any of the N vehicles downloads a constant number of $m \ll M$ packets, i.e., the duration and rate of download per contact are constant and independent of the vehicle. For security assurance, we assume that communications between the BS and each vehicle is encrypted. In other words, the BS communicates with only one vehicle at a time and the other nodes can not see the communication link between them. That means, in collaborative downloading sharing is limited to V2V phase and we do not use the advantage of broadcasting in R2V phase for the sake of security.

Obviously, the way the m packets are chosen affects the system performance. We consider the following two ways of selecting m packets in each round:

- *Feedback-based scheme*: The m packets are chosen randomly by the serving BS among the *currently unreceived packets* of the vehicle. It is assumed that in each round, nodes can individually signal to the server the specific indices of packets yet to be received.

- *Network coding aided scheme*: In this scheme the BS uses Random Linear Network Coding (RLNC). In each transmission, it sends a linear combination of the M packets, where the combining coefficients are uniformly chosen over the finite field $\mathbb{F}_{2^q}$. In this scenario no feedback is needed from vehicles to the BS.

When the last vehicle leaves the BS coverage area, all nodes are in V2V phase. In this phase, every node tries to share its information with other nodes inside the network. After V2V phase is complete, all vehicles have the same information. When the vehicles enter the

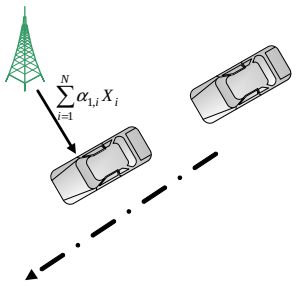


Figure 4.7: In each transmission, BS sends a linear combination of the packets to a vehicle and this transmission can not be seen by any other vehicles.

range of the next BS, they try to obtain the remaining missing packets. This continues until every node has the full message of M packets. We call each R2V and its following V2V phase a *round*. The number of rounds required to send the M packets from the infrastructure to the vehicles is the matter of interest. In this manuscript, we aim to derive probability distribution and expected value of the number of rounds needed to disseminate information to all N vehicles in the network, using either the feedback-based or NC-aided scheme.

For the sake of simplicity, we only consider the case where there are two vehicles in the network, i.e., $N = 2$. Generalizing our result to arbitrary N is considered as one of our future works. For each scheme, we first analyze a simple case of $m = 1$, transmitting only one new packet in each R2V phase of a vehicle. Then, we generalize our result to arbitrary m.

4.2.2 Feedback-based scheme:

Suppose there are $N = 2$ vehicles in the network. As mentioned before, in each R2V phase, a vehicle downloads m packets from the BS. In the following V2V phase, vehicles exchange their uncommon packets. In the i-th round, let $\mathcal{X}_i$ denote the number of *common* packets among the m downloaded packets by each node. Clearly, $\mathcal{X}_i$ is a random variable in the set $\{0, 1, \ldots, m\}$. Further, let define $\mathcal{S}_i$ as number of packets each node has at the end of i-th round. Clearly, at the end of each round (R2V+V2V) $2m - \mathcal{X}_i$ new packets are added to

each node. Thus, It is easy to see the following recursive equation for $\mathcal{S}_i$:

$$\mathcal{S}_i = \mathcal{S}_{i-1} + 2m - \mathcal{X}_i. \tag{4.18}$$

The dissemination algorithm stops when both cars have all the M packets, i.e., the whole information. Let $\mathcal{T}$ denote stopping time defined as follows:

$$\mathcal{T} = \min_t \{\mathcal{S}_t = M, t \geq 0\}. \tag{4.19}$$

In this section we aim to calculate probability distribution and expected value of the stopping time for the feedback-based dissemination algorithm.

4.2.3 $m = 1$, $N = 2$

lemma 7 *The number of common packets at the i-th round given* $\mathcal{S}_{i-1} = s$ *has a probability distribution given by*

$$\begin{aligned} P(\mathcal{X}_i = x | \mathcal{S}_{i-1} = s_i) &= \frac{\binom{1}{x}\binom{M-s_i-1}{1-x}}{\binom{M-s_i}{1}} \\ &= \frac{\binom{M-s_i-1}{1-x}}{M - s_i} \quad x = 0, 1. \end{aligned} \tag{4.20}$$

Proof: see A.4

▲

As mentioned before, a vehicle, in each R2V phase, individually signal to the BS the specific indices of packet yet to be received by it. Thus, in each round number of packets each node has, would be increased by at least one when $m = 1$. Hence number of rounds needed to deliver all the M packets is at most M, i.e., $\mathcal{T}$ is bounded from above by M. On the other hand, at each round at most two new packets can be delivered to the vehicles, i.e. $M/2 \leq \mathcal{T}$. By the same argument, it is easy to see that for each $0 \leq t \leq \mathcal{T}$, $\mathcal{S}_t$ is bounded as $t \leq \mathcal{S}_t \leq 2t$.

By definition of $\mathcal{T}$, we have $\mathcal{S}_\mathcal{T} = N$. We aim to calculate probability distribution of stopping time, i.e., $P(\mathcal{T} = t)$. If the stopping time is $\mathcal{T}$, then at $\mathcal{T} - 1$, one and only one of the followings is correct:

- $\mathcal{S}_{\mathcal{T}-1} = M-1$
- $\mathcal{S}_{\mathcal{T}-1} = M-2$

By conditioning on the above events and by using the law of total probability, $P(\mathcal{T}=t)$ can be calculated as follows:

$$\begin{aligned} P(\mathcal{T}=t) &= P(\mathcal{T}=t|\mathcal{S}_{t-1}=M-1)P(\mathcal{S}_{t-1}=M-1) \\ &\quad + P(\mathcal{T}=t|\mathcal{S}_{t-1}=M-2)P(\mathcal{S}_{t-1}=M-2) \\ &= P(\mathcal{S}_t=M|\mathcal{S}_{t-1}=M-1)P(\mathcal{S}_{t-1}=M-1) \\ &\quad + P(\mathcal{S}_t=M|\mathcal{S}_{t-1}=M-2)P(\mathcal{S}_{t-1}=M-2). \end{aligned} \tag{4.21}$$

Given $\mathcal{S}_{t-1}=M-1$, probability of having $\mathcal{S}_t=M$ is the same as probability of $\mathcal{X}_t=1$, which has been given in Eq. (4.18). Thus by using Lemma 7, we have:

$$P(\mathcal{S}_t=M|\mathcal{S}_{t-1}=M-1) = P(\mathcal{X}_t=1|\mathcal{S}_{t-1}=M-1) = 1,$$

because if $\mathcal{S}_{t-1}=M-1$, it means that the vehicles are one packet short. In the next round, the BS will send them that packet to complete the downloading process.

The above can be summarized to the following lemma:

lemma 8 *Let $\mathcal{S}_t$ be the number of packets each node has at the end of t-th round. Then*

$$\begin{aligned} P(\mathcal{S}_t=s+1|\mathcal{S}_{t-1}=s) &= P(\mathcal{X}_t=1|\mathcal{S}_{t-1}=s) \\ &= \frac{1}{M-s}, \end{aligned} \tag{4.22}$$

and

$$\begin{aligned} P(\mathcal{S}_t=s+2|\mathcal{S}_{t-1}=s) &= P(\mathcal{X}_t=0|\mathcal{S}_{t-1}=s) \\ &= \frac{M-s-1}{M-s}, \end{aligned} \tag{4.23}$$

where (as mentioned before)

$$t \le \mathcal{S}_t \le 2t.$$

▲

To calculate $P(\mathcal{T}=t)$ in Eq. (4.21), we need to calculate $P(\mathcal{S}_t=s)$. It can be calculated by conditioning on the two following collectively exclusive events:

- $\mathcal{S}_{t-1}=s-1$
- $\mathcal{S}_{t-2}=s-2$

Therefor, we have

$$\begin{aligned} P(\mathcal{S}_t=s) &= P(\mathcal{S}_t=s|\mathcal{S}_{t-1}=s-1)P(\mathcal{S}_{t-1}=s-1) \\ &\quad +P(\mathcal{S}_t=s|\mathcal{S}_{t-1}=s-2)P(\mathcal{S}_{t-1}=s-2) \\ &= P(\mathcal{X}_t=1|\mathcal{S}_{t-1}=s-1)P(\mathcal{S}_{t-1}=s-1) \\ &\quad +P(\mathcal{X}_t=0|\mathcal{S}_{t-1}=s-2)P(\mathcal{S}_{t-1}=s-2) \\ &= \frac{1}{M-s+1}P(\mathcal{S}_{t-1}=s-1) \\ &\quad +\frac{M-s+1}{M-s+2}P(\mathcal{S}_{t-1}=s-2). \end{aligned} \tag{4.24}$$

So, $P(\mathcal{S}_t=s)$ can be recursively calculated using above equation and the following initialization:

$$\begin{aligned} P(\mathcal{S}_1=1) &= P(\mathcal{X}_1=1)=\frac{1}{M}, \\ P(\mathcal{S}_1=2) &= P(\mathcal{X}_1=0)=\frac{M-1}{M}. \end{aligned} \tag{4.25}$$

We summarize the above findings to the following theorem:

Theorem 16 *Let $\mathcal{T}$ be the stopping time. Then, it has a probability distribution as follows:*

$$P(\mathcal{T}=t)=P(\mathcal{S}_{t-1}=M-1)+\frac{1}{2}P(\mathcal{S}_{t-1}=M-2), \tag{4.26}$$

where $\frac{M}{2} \leq t \leq M$. $P(\mathcal{S}_t = s)$ *can be calculated using the following recursive formulation:*

$$\begin{aligned} P(\mathcal{S}_t = s) &= \frac{1}{M-s+1} P(\mathcal{S}_{t-1} = s-1) \\ &\quad + \frac{M-s+1}{M-s+2} P(\mathcal{S}_{t-1} = s-2), \end{aligned} \tag{4.27}$$

which has the following initialization expressions:

$$P(\mathcal{S}_1 = 1) = \frac{1}{M}, P(\mathcal{S}_1 = 2) = \frac{M-1}{M}.$$

4.2.4 $N = 2$ and arbitrary m

The result in previous section can be readily generalized to the following. Its proof is omitted here for the sake of space limitation.

Theorem 17 *Let BS have M packets to distribute among two vehicles. In each R2V phase it transmits m packets to each car separately. Let $\mathcal{T}$ denote the stopping time. Then, it has the following probability distribution:*

$$P(\mathcal{T} = t) = \sum_{i=1}^{2m} \frac{\binom{m}{2m-i}}{\binom{i}{m}} P(\mathcal{S}_{t-1} = M-i), \tag{4.28}$$

where $\frac{M}{2m} \leq t \leq \frac{M}{m}$. *Moreover,* $P(\mathcal{S}_t = s)$ *can be calculated using the following recursive formula:*

$$P(\mathcal{S}_t = s) = \sum_{i=m}^{i=2m} \frac{\binom{m}{2m-i}\binom{M-s+i-m}{i-m}}{\binom{M-s+i}{m}} P(\mathcal{S}_{t-1} = s-i), \tag{4.29}$$

which has initialization expressions as follows:

$$P(\mathcal{S}_1 = i) = \frac{\binom{m}{2m-i}\binom{M-m}{i-m}}{\binom{M}{m}}, \;\; i = m, \ldots, 2m. \tag{4.30}$$

Proof: see [57]

4.2.5 Network Coding Aided Scheme

Suppose BS is using linear random network coding. That means, in each transmission it sends a linear combination of the M packets, where combination coefficients are uniformly chosen in finite field $\mathbb{F}_{2^q}$. By the same procedure we did for the feedback-based algorithm,

we first derive the probability distribution and expectation of stopping time for a very simple case of $N = 2$ and $m = 1$. Then, we solve the problem in a more general case of $N = 2$ and arbitrary m.

4.2.6 $m = 1$, $N = 2$

The BS sends a linear random combination of its M packets in R2V phase. For example, in the first round, it sends the following packet to the first vehicle:

$$\sum_{i=1}^{M} \alpha_{1,i} x_i, \;\; \alpha_{1,i} \in \mathbb{F}_{2^q}, \tag{4.31}$$

where x_i's are the BS information packets, as defined in Section 4.2.1. The following combination is sent to the second vehicle:

$$\sum_{i=1}^{M} \alpha_{2,i} x_i, \;\; \alpha_{2,i} \in \mathbb{F}_{2^q}. \tag{4.32}$$

At the end of the first round (after one R2V+V2V), both vehicles have the same information, i.e., they both have packets in Eq. (4.31) and Eq. (4.32). The above can be written as $\mathbf{Ax}$ where $\mathbf{x} = [x_1\, x_2\, \ldots\, x_M]^T$ is an $M \times 1$ vector of BS packets and $\mathbf{A}$ is a $2 \times M$ matrix containing NC coefficients. Clearly, at the end of second round two rows are added to matrix $\mathbf{A}$ and $\mathbf{Ax}$ is a 4×1 vector representing 4 packets in each vehicle. In general at the end of each round, two rows are added to $\mathbf{A}$. Therefore, at the end of round t, matrix $\mathbf{A}$ has dimension $2t \times M$. Clearly, nodes are able to recover BS information whenever matrix $\mathbf{A}$ is invertible or equivalently when $rank(\mathbf{A}) = M$. Thus, we can define the stopping time when NC is applied, $\mathcal{T}_{NC}$, as follows:

$$\mathcal{T}_{NC} = \min_{t}\{rank(A_{2t\times M}) = M\}. \tag{4.33}$$

Obviously $\mathcal{T}_{NC} \geq M/2$. As mentioned before, our goal is to calculate the probability distribution and expected value of $\mathcal{T}_{NC}$. The following lemma gives the probability of rank of matrix $\mathbf{A}_{t\times M}$ with random entries in a finite field.

lemma 9 *Let $\mathbf{A}_{t\times n}$ be a random matrix over finite field $\mathbb{F}_{2^q}$ such that each entry $a_{i,j}$ is picked uniformly from $\mathbb{F}_{2^q}$. suppose $t \geq n$. Then, probability of $\mathbf{A}$ having rank n is given as*

follows:

$$P(rank(\mathbf{A}_{t\times n}) = n) = \prod_{i=1}^{n}(1 - \frac{1}{(2^q)^{t-n+i}}) \approx 1 - \frac{1}{(2^q)^{t-n+1}}, \tag{4.34}$$

where the approximation is valid for sufficiently large q.

Proof: See [126].

▲

Using the above result, an upper-bound on the probability of the stopping time can be calculated as given in following lemma. Its proof is omitted here for the sake of space limitation.

lemma 10 *Let $\mathcal{T}_{NC}$ be stopping time defined in Eq.* (4.33)*. Then,*

$$P(\mathcal{T}_{NC} = t) \leq (1-\frac{1}{q})(1-P(rank(\mathbf{A}_{2t-1\times M})=M)) \leq (1 - \frac{1}{q})\frac{1}{q^{2t-M}}, \tag{4.35}$$

where the last inequality is valid for large q.

Proof: See A.4.

▲

Finally, the following theorem gives an upper-bound for the expected value of the stopping time when q is large enough.

Theorem 18 *Let $\mathcal{T}_{NC}$ be stopping time defined in* (4.33)*. Suppose q is large enough such that the approximation results in equations* (4.34) *and* (4.35) *are valid. Then the expected value of stopping time is bounded from above with the following:*

$$\mathbb{E}[\mathcal{T}_{NC}] \leq \frac{M}{2} + \frac{1}{q-1}. \tag{4.36}$$

Proof: See A.4.

4.2.7 $N = 2$ and arbitrary m

Results in previous section can be readily generalized to arbitrary m, number of packets transmitted from BS to vehicles in each R2V phase. Its proof is omitted here for the sake of space limitation.

Theorem 19 *Let $\mathcal{T}_{NC}$ be stopping time defined in* (4.33)*. Suppose q is large enough such that the approximation results in equations* (4.34) *and* (4.35) *are valid. Then the expected value of stopping time is bounded from above with the following:*

$$\mathbb{E}[\mathcal{T}_{NC}] \leq \frac{M}{2m} + \frac{1}{q-1} \tag{4.37}$$

.

4.2.8 Evaluation Results

In this section, we present results from simulations conducted using MATLAB R2008b. We conduct simulations to confirm the analytical results in Section 4.2.2 and Section 4.2.5, describing data dissemination in R2V phase with and without network coding.

Figure 4.8 depicts the average number of rounds needed to disseminate information from infrastructure to the vehicles. As explained before, we are assuming complete information exchange in V2V phase. In the simulation, the BS contains M packets and in each round it sends m packets to each vehicle separately. As we mentioned before, we limit ourselves to $N = 2$ number of vehicles.

As one can see in Figure 4.8, $\mathcal{T}$ linearly increases with M, the number of total information packets the BS possesses. There is a very small benefit in using NC when the matric is number of rounds. In fact, for only two vehicles, the feedback-based algorithm needs almost one round more than NC aided scheme (note that NC almost achieves minimum number of rounds). Therefore, for dissemination delay, NC has a small advantage compared to feedback-based algorithm. However, NC scheme removes any need for signaling. In the NC aided scheme the BS keeps sending network coded data to the vehicles without any knowledge of what information they possess. On the other hand, for the feedback-based

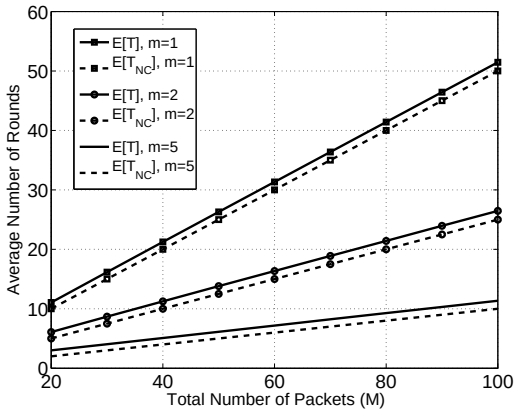


Figure 4.8: Average stopping time for $N = 2$, $m = 1, 2, 5$ and q=8

algorithm vehicles are required to communicate with the BS before receiving any data. They need to inform the BS regarding the packets in which they are interested.

BIBLIOGRAPHY

[1] Rice university wireless open access research platform, warp radio board. `http://warp.rice.edu/trac/wiki/Radio\Board`.

[2] IEEE Std 1609 family, IEEE trial-use standard for wireless access in vehicular environments(WAVE). Nov. 2006.

[3] C. Adjih and S.Y. Cho. Wireless broadcast with network coding: Dynamic rate selection. In *IFIP International Federation for Information Processing*, volume 265, 2010.

[4] M. Adler, T. Bu, R.K. Sitaraman, and D. Towsley. Tree layout for internal network characterizations in multicast networks. *Lecture Notes in Computer Science*, pages 189–204, 2001.

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

[6] A. Asterjadhi, E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi. Toward network coding-based protocols for data broadcasting in wireless ad hoc networks. *IEEE Trans. Wireless Communications*, 9(2):662–673, 2010.

[7] T.C. Aysal, M.E. Yildiz, A.D. Sarwate, and A. Scaglione. Broadcast gossip algorithms for consensus. *IEEE Trans. Signal Processing*, 57(7):2748–2761, 2009.

[8] E.F. Beckenbach, R. Bellman, E.F. Beckenbach, E.F. Beckenbach, R.E. Bellman, and R.E. Bellman. *An introduction to inequalities*, volume 3. Random House, 1961.

[9] L.W. Beineke and R.J. Wilson. *Topics in algebraic graph theory.* Cambridge Univ Press, 2004.

[10] Y. Bejerano and R. Rastogi. Robust monitoring of link delays and faults in IP networks. In *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'03)*, volume 1, pages 134–144, 2003.

[11] Y. Bejerano and R. Rastogi. Robust monitoring of link delays and faults in IP networks. *IEEE/ACM Trans. Networking*, 14(5):1103, 2006.

[12] R. Berinde, A.C. Gilbert, P. Indyk, H. Karloff, and M.J. Strauss. Combining geometry and combinatorics: A unified approach to sparse signal recovery. In *Proc. 46th Annual Allerton Conference on Communication, Control, and Computing*, pages 798–805, 2008.

[13] R. Berinde and P. Indyk. Sparse recovery using sparse random matrices. *preprint*, 2008.

[14] G. Bianchi, L. Fratta, and M. Oliveri. Performance evaluation and enhancement of the csma/ca mac protocol for 802.11 wireless lans. In *Proceedings of Seventh IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 2, pages 392–396, 1996.

[15] CJ Bovy, HT Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, and P. Van Mieghem. Analysis of end-to-end delay measurements in internet. In *Proc. the Passive and Active Measurement Workshop*, 2002.

[16] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *Information Theory, IEEE Trans. on*, 52(6):2508–2530, 2006.

[17] S.P. Bradley, A.C. Hax, and T.L. Magnanti. *Applied mathematical programming*. Addison-Wesley, 1977.

[18] R.A. Brualdi and H.J. Ryser. *Combinatorial matrix theory*. Cambridge Univ Press, 1991.

[19] T. Bu, N. Duffield, F.L. Presti, and D. Towsley. Network tomography on general topologies. *ACM SIGMETRICS Performance Evaluation Review*, 30(1):21–30, 2002.

[20] T. Bu and D. Towsley. On distinguishing between internet power law topology generators. In *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, volume 2, pages 638–647, 2002.

[21] R. Caceres, NG Duffield, J. Horowitz, and DF Towsley. Multicast-based inference of network-internal loss characteristics. *IEEE Trans. Information theory*, 45(7):2462–2480, 1999.

[22] E. Candes and T. Tao. Decoding by linear programming. *IEEE Trans. Information theory*, 51(12):4203–4215, Dec. 2005.

[23] E.J. Candès. Compressive sampling. In *Proceedings of the International Congress of Mathematicians*, volume 3, pages 1433–1452, 2006.

[24] E.J. Candès. Compressive sampling. In *Proc of the International Congress of Mathematicians*, volume 3, pages 1433–1452, 2006.

[25] E.J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Information Theory*, 52(2):489–529, 2006.

[26] E.J. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Information theory*, 52(12):5406–5425, 2006.

[27] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson. Randomness conductors and constant-degree lossless expanders. In *Proc. of the 34th annual ACM symp. theory of computing*, pages 659–668, 2002.

[28] Rui Castro, Mark Coates, Gang Liang, Robert Nowak, and Bin Yu. Network tomography: Recent developments. *Statistical Science*, 19(3):499–517, 2004.

[29] 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*, pages 38–43, 2002.

[30] A. Chen, J. Cao, and T. Bu. Network tomography: Identifiability and fourier domain estimation. *in Proc. IEEE Infocom, Alaska*, 2007.

[31] A. Chen, J. Cao, and T. Bu. Network tomography: Identifiability and fourier domain estimation. In *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, pages 1875–1883, 2007.

[32] A. Chen, J. Cao, and T. Bu. Network tomography: Identifiability and fourier domain estimation. *IEEE Trans. on Signal Processing*, 58(12):6029–6039, 2010.

[33] Z. Chen, H. Liu, and W. Wang. A novel decoding-and-forward scheme with joint modulation for two-way relay channel. *IEEE Communications Letters*, 14(12):1149–1151, 2010.

[34] Z. Chen, C. Yang, H. Liu, and W. Wang. Coded modulation design for two-way relay channels. *submitted to IEEE Trans. Wireless Communications*, April 2011.

[35] P.A. Chou, Y. Wu, and K. Jain. Practical network coding.

[36] P.A. Chou, Y. Wu, and K. Jain. Practical network coding. In *Proc. the annual allerton Conf. Communication control and computing*, volume 41, pages 40–49, 2003.

[37] J. Clark and D.A. Holton. *A first look at graph theory.* World Scientific Publishing Company, 1991.

[38] A. Coates, AO Hero III, R. Nowak, and B. Yu. Internet tomography. *IEEE Signal processing magazine*, 19(3):47–65, 2002.

[39] M. Coates and R. Nowak. Network loss inference using unicast end-to-end measurement. In *Proc. Int. Conf. IP Traffic, Modeling and Management*, 2000.

[40] M. Coates, Y. Pointurier, and M. Rabbat. Compressed network monitoring. In *Proc. IEEE/SP 14th Workshop on Statistical Signal Processing*, pages 418–422, 2007.

[41] M. Coates, Y. Pointurier, and M. Rabbat. Compressed network monitoring for ip and all-optical networks. In *Proc. the 7th ACM SIGCOMM conference on Internet measurement*, pages 241–252, 2007.

[42] S. Deb, M. Médard, and C. Choute. Algebraic gossip: A network coding approach to optimal multiple rumor mongering. *IEEE Trans. Information Theory*, 52(6):2486–2507, 2006.

[43] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic algorithms for replicated database maintenance. In *Proc. of the sixth annual ACM symposium on principles of distributed computing*, pages 1–12, 1987.

[44] L. Denby, J.M. Landwehr, C.L. Mallows, J. Meloche, J. Tuck, B. Xi, G. Michailidis, and V.N. Nair. Statistical aspects of the analysis of data networks. *Technometrics*, 49(3):318–334, 2007.

[45] L. Denby, J.M. Landwehr, C.L. Mallows, J. Meloche, J. Tuck, B. Xi, G. Michailidis, and V.N. Nair. Statistical aspects of the analysis of data networks. *Technometrics*, 49(3):318–334, 2007.

[46] V. Dham. *Link Establishment in Ad Hoc Networks Using Smart Antennas.* Master of Science Thesis, Virginia Polytechnic Institute and State University, 2003.

[47] M. Dillinger, K. Madani, and N. Alonistioti. *Software defined radio: architectures, systems, and functions.* John Wiley & Sons Inc, 2003.

[48] A.G. Dimakis, P.B. Godfrey, Y. Wu, M.J. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *IEEE Trans. Information Theory*, 56(9):4539–4551, 2010.

[49] D.L. Donoho. Compressed sensing. *IEEE Trans. Information theory*, 52(4):1289–1306, 2006.

[50] N. Duffield. Network tomography of binary network performance characteristics. *IEEE Trans. Information Theory*, 52(12):5373–5388, 2006.

[51] S.Y. El Rouayheb, M.A.R. Chaudhry, and A. Sprintson. On the minimum number of transmissions in single-hop wireless coding networks. In *Proceedings of IEEE Information Theory Workshop*, pages 120–125, 2007.

[52] B. Eriksson, G. Dasarathy, P. Barford, and R. Nowak. Toward the practical use of network tomography for internet topology discovery. In *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–9, 2010.

[53] P.T. Eugster, R. Guerraoui, A.M. Kermarrec, and L. Massoulié. Epidemic information dissemination in distributed systems. *Computer*, pages 60–67, May 2004.

[54] P.T. Eugster, R. Guerraoui, A.M. Kermarrec, and L. Massoulié. Epidemic information dissemination in distributed systems. *Computer*, 37(5):60–67, 2005.

[55] Y. Evren Sagduyu and A. Ephremides. On joint MAC and network coding in wireless ad hoc networks. *IEEE Trans. Information Theory*, 53(10):3697–3713, 2007.

[56] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM Computer Communication Review*, volume 29, pages 251–262, 1999.

[57] M. H. Firooz and S. Roy. Collaborative downloading in vanet using network coding. *in Proc. IEEE International Conference on Communications (ICC)*, June 2012.

[58] M. H. Firooz, S. Roy, L. Bai, and C. Lydick. Link status monitoring using network coding. *UW Technical Report*, Aug 2011.

[59] M.H. Firooz, Z. Chen, S. Roy, and H. Liu. Wireless network coding via modified 802.11 mac/phy: Design and implementation on sdr. *IEEE Journal on Selected Area in Communications, (to appear)*, 2012.

[60] M.H. Firooz and S. Roy. Link delay estimation via expander graphs. *Arxiv preprint arXiv:1106.0941*, 2011.

[61] M.H. Firooz and S. Roy. Network tomography via compressed sensing. Dec. 2011.

[62] M.H. Firooz and S. Roy. Data dissemination in wireless networks with network coding. *IEEE letters on Communications (accepted)*, 2012.

[63] Mohammad H. Firooz and Chris Lydick. Opnet implementation of network coding. `http://www.ee.washington.edu/research/funlab/network_coding`, 2009. [Online].

[64] Mohammad H. Firooz, R. Roy, B. Bai, and C. Lydick. Link failure monitoring via network coding. In *Proc. IEEE 35th Conference on Local Computer Networks (LCN)*, pages 1068–1075, 2010.

[65] C. Fragouli and A. Markopoulou. A network coding approach to overlay network monitoring. In *Allerton Conference*, 2005.

[66] C. Fragouli, A. Markopoulou, R. Srinivasan, and S. Diggavi. Network monitoring: it depends on your points of view. In *Proc. of ITA Workshop*, Jan. 2007.

[67] C. Fragouli, J. Widmer, and J.Y. Le Boudec. Efficient broadcasting using network coding. *IEEE/ACM Trans. Networking*, 16(2):450–463, 2008.

[68] M. Garetto, T. Salonidis, and E.W. Knightly. Modeling per-flow throughput and capturing starvation in csma multi-hop wireless networks. *IEEE/ACM Trans. Networking*, 16(4):864–877, 2008.

[69] M. Ghaderi, D. Towsley, and J. Kurose. Reliability gain of network coding in lossy wireless networks. In *IEEE 27th Conference on Computer Communications (INFOCOM)*, pages 2171–2179, 2008.

[70] S. Ghandeharizade, S. Kapadia, and B. Krishnamachari. Pavan: a policy framework for content availabilty in vehicular ad-hoc networks. In *Proc. the 1st ACM international workshop on Vehicular ad hoc networks*, pages 57–65, 2004.

[71] A.C. Gilbert, M.J. Strauss, J.A. Tropp, and R. Vershynin. One sketch for all: fast algorithms for compressed sensing. In *Proc. the thirty-ninth annual ACM symposium on Theory of computing*, pages 237–246, 2007.

[72] M. Gjoka, C. Fragouli, P. Sattari, and A. Markopoulou. Loss tomography in general topologies with network coding. In *Proc. IEEE Conf. Global telecommunications*, pages 381–386, 2007.

[73] M. Gjoka, C. Fragouli, P. Sattari, and A. Markopoulou. Loss tomography in general topologies with network coding. In *Proc. IEEE Globecom*, pages 381–386, 2007.

[74] C. Gkantsidis, P. Rodriguez, et al. Cooperative security for network coding file distribution. In *IEEE INFOCOM*, pages 1–13, 2006.

[75] C. Gkantsidis and P.R. Rodriguez. Network coding for large scale content distribution. In *Proc. of INFOCOM*, volume 4, pages 2235–2245, 2005.

[76] R.A. Golding. *Weak-consistency group communication and membership*. PhD thesis, 1992.

[77] J.D. Hall, N.A. Carr, and J.C. Hart. Cache and bandwidth aware matrix multiplication on the GPU. 2003.

[78] K. Harfoush, A. Bestavros, and J. Byers. Robust identification of shared losses using end-to-end unicastprobes. In *Proc. Int. Conf. Network Protocols*, pages 22–33, 2000.

[79] H. Hartenstein and K. Laberteaux. *VANET Vehicular Applications and Inter-Networking Technologies*. Wiley Online Library, 2009.

[80] J. Haupt, W.U. Bajwa, M. Rabbat, and R. Nowak. Compressed sensing for networked data. *Signal Processing Magazine, IEEE*, 25(2):92–101, 2008.

[81] T. Ho, DR Karger, M. Medard, and R. Koetter. Network coding from a network flow perspective. In *Proc. of IEEE Int. Symp. Information Theory(ISIT'03)*.

[82] T. Ho, R. Koetter, M. Medard, DR Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *Proc. IEEE Int. Symposium on Information Theory*, page 442, Jun-Jul 2010.

[83] T. Ho, B. Leong, Y.H. Chang, Y. Wen, and R. Koetter. Network monitoring in multicast networks using network coding. In *Proc. Int. Symp. Information theory*, pages 1977–1981, 2005.

[84] T. Ho and D.S. Lun. *Network Coding: an Introduction*. Cambridge University Press, 2008.

[85] T. Ho, M. Medard, R. Koetter, D.R. Karger, M. Effros, J. Shi, and B. Leong. A random linear network coding approach to multicast. *IEEE Trans. Information Theory*, 52(10):4413–4430, 2006.

[86] T. Ho, M. Médard, J. Shi, M. Effros, and D.R. Karger. On randomized network coding. In *Proc. the Annual Allerton Conference on Communication Control and Computing*, volume 41, pages 11–20, 2003.

[87] T. Ho, M. Medard, J. Shi, M. Effros, and D.R. Karger. On randomized network coding. In *Proc. Annual allerton Conf. Communication control and computing*, volume 41, pages 11–20, 2003.

[88] T.S. Ho and K.C. Chen. Performance analysis of ieee 802.11 csma/ca medium access control protocol. In *Proceedings of Seventh IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 2, pages 407–411, 1996.

[89] E. Horowitz. Modular arithmetic and finite field theory: A tutorial. In *Proceedings of the second ACM symposium on Symbolic and algebraic manipulation*, pages 188–194, 1971.

[90] P. Indyk and M. Ruzic. Near-optimal sparse recovery in the l1 norm. In *IEEE 49th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 199–207, 2008.

[91] S. Jafarpour, W. Xu, B. Hassibi, and R. Calderbank. Efficient and robust compressed sensing using optimized expander graphs. *IEEE Trans. Information Theory*, 55(9):4299–4308, 2009.

[92] M. Jerbi, P. Marlier, and S.M. Senouci. Experimental assessment of V2V and I2V communications. In *IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–6, 2007.

[93] M. Jerbi and S.M. Senouci. Characterizing multi-hop communication in vehicular networks. In *IEEE Wireless Communications and Networking Conference*, pages 3309–3313, 2008.

[94] A. Jiang. Network coding for joint storage and transmission with minimum cost. In *IEEE Intl. Symposium on Information Theory*, pages 1359–1363, 2006.

[95] D. Jiang and L. Delgrossi. IEEE 802.11p: Towards an international standard for wireless access in vehicular environments. In *Vehicular Technology Conference*, pages 2036–2040, 2008.

[96] C. Jin, Q. Chen, and S. Jamin. Inet: Internet topology generator. 2000.

[97] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking. Randomized rumor spreading. In *Proc. of the 41st annual symposium on foundations of computer science*, page 565, 2000.

[98] S. Katti, S. Gollakota, and D. Katabi. Embracing wireless interference: Analog network coding. In *Proc. Applications, technologies, architectures, and protocols for computer communications*, pages 397–408, 2007.

[99] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. XORs in the air: practical wireless network coding. *IEEE/ACM Trans. Networking*, 16(3):497–510, 2008.

[100] A.M. Kermarrec, L. Massoulié, and A.J. Ganesh. Probabilistic reliable dissemination in large-scale systems. *IEEE Trans. Parallel and Distributed systems*, pages 248–258, 2003.

[101] M.A. Khajehnejad, A.G. Dimakis, W. Xu, and B. Hassibi. Sparse recovery of nonnegative signals with minimal expansion. *IEEE Trans. Signal Processing*, 59(1):196–208, 2011.

[102] R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM Trans. Networking*, 11(5):782–795, 2003.

[103] R. Koetter and M. Medard. An algebraic approach to network coding. *IEEE/ACM trans. Networking*, 11(5):782–795, 2003.

[104] G. Korkmaz, E. Ekici, F. Özgüner, and Ü. Özgüner. Urban multi-hop broadcast protocol for inter-vehicle communication systems. In *Proc. ACM Intl workshop on Vehicular ad hoc networks*, pages 76–85, 2004.

[105] J.F. Kurose and K.W. Ross. *Computer Networking: A Top-Down Approach*. Pearson/Addison Wesley, fifth edition, 2009.

[106] E.S. Larsen and D. McAllister. Fast matrix multiplies using graphics hardware. In *Proc. of the ACM/IEEE conference on Supercomputing*, pages 55–55, 2001.

[107] P. Larsson, N. Johansson, and K.E. Sunell. Coded bi-directional relaying. In *Proc. IEEE Vehicular Technology Conference*, volume 2, pages 851–855, 2006.

[108] E. Lawrence, G. Michailidis, and V.N. Nair. Network delay tomography using flexicast experiments. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(5):785–813, 2006.

[109] E. Lawrence, G. Michailidis, and V.N. Nair. Network delay tomography using flexicast experiments. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(5):785–813, 2006.

[110] E. Lawrence, G. Michailidis, V.N. Nair, and B. Xi. Network tomography: a review and recent developments. *Ann Arbor*, 1001:48109–1107.

[111] E. Lawrence, G. Michailidis, V.N. Nair, and B. Xi. Network tomography: a review and recent developments. *Ann Arbor*, 1001, 2006.

[112] S.Y.R. Li, RW Yeung, and N. Cai. Linear network coding. *IEEE Trans. Information Theory*, 49(2):371–381, 2003.

[113] S.Y.R. Li, R.W. Yeung, and N. Cai. Linear network coding. *IEEE Trans. Information Theory*, 49(2):371–381, 2003.

[114] X. Li, W.H. Mow, and F.L. Tsang. Rank distribution analysis for sparse random linear network coding. In *Proceedings of International Symposium on Network Coding (NetCod)*, pages 1–6, 2011.

[115] C.H. Liu and F. Xue. Network coding for two-way relaying: rate region, sum rate and opportunistic scheduling. In *Proc. IEEE International Conference on Communications*, pages 1044–1049, 2008.

[116] J. Liu, M. Tao, Y. Xu, and X. Wang. Superimposed xor: A new physical layer network coding scheme for two-way relay channels. In *Proc. IEEE GLOECOM*, pages 1–6, Dec 2009.

[117] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[118] D.S. Lun, M. Médard, T. Ho, and R. Koetter. Network coding with a cost criterion. In *Proceedings of International Symposium on Information Theory and its Applications*, pages 1232–1237, 2004.

[119] D.S. Lun, M. Médard, R. Koetter, and M. Effros. On coding for reliable communication over packet networks. *Physical Communication journal*, 1(1):3–20, 2008.

[120] D.S. Lun, N. Ratnakar, R. Koetter, M. Médard, E. Ahmed, and H. Lee. Achieving minimum-cost multicast: A decentralized approach based on network coding. In *Proceedings of IEEE Annual Joint Conference of the Computer and Communications Societies*, volume 3, pages 1607–1617, 2005.

[121] D.S. Lun, N. Ratnakar, M. Médard, R. Koetter, D.R. Karger, T. Ho, E. Ahmed, and F. Zhao. Minimum-cost multicast over coded packet networks. *IEEE/ACM Trans. Networking*, 14(SI):2608–2623, 2006.

[122] Christopher Lydick. Tutorial: Network coding using opnet and matlab. *Thechnical report*, May 2009.

[123] R.J. McEliece. *Finite fields for computer scientists and engineers*. Springer, 1987.

[124] A. Medina, I. Matta, and J. Byers. On the origin of power laws in internet topologies. *ACM SIGCOMM computer communication review*, 30(2):18–28, 2000.

[125] T. Migler, K.E. Morrison, and M. Ogle. Weight and rank of matrices over finite fields. *Arxiv preprint math*, 2004.

[126] Theresa Migler, Kent E. Morrison, and Mitchell Ogle. Weight and rank of matrices over finite fields. *Arxiv*, 2004.

[127] Y. Minsky. *Spreading rumors cheaply, quickly, and reliably.* PhD thesis, 2002.

[128] D. Mosk-Aoyama and D. Shah. Information dissemination via network coding. In *IEEE Intl. Symposium on information theory*, pages 1748–1752, 2006.

[129] M. Najiminaini, L. Subedi, and L. Trajkovic. Analysis of internet topologies: a historical view. In *Proc. IEEE International Symposium on Circuits and Systems*, pages 1697–1700, 2009.

[130] D. Nguyen, T. Tran, T. Nguyen, and B. Bose. Wireless broadcast using network coding. *IEEE Transactions on Vehicular Technology*, 58(2):914–925, 2009.

[131] H.X. Nguyen and P. Thiran. Active measurement for multiple link failures diagnosis in IP networks. *Passive and Active Network Measurement*, pages 185–194, 2004.

[132] S.Y. Ni, Y.C. Tseng, Y.S. Chen, and J.P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *Proceedings of the annual ACM/IEEE Intl. Conf. Mobile computing and networking*, pages 151–162, 1999.

[133] J. Nonnenmacher, E.W. Biersack, and D. Towsley. Parity-based loss recovery for reliable multicast transmission. *IEEE/ACM Transactions on Networking*, 6(4):349–361, 1998.

[134] V. Paxson. End-to-end internet packet dynamics. In *ACM SIGCOMM Computer Communication Review*, volume 27, pages 139–152, 1997.

[135] R. Prakash, A. Schiper, M. Mohsin, D. Cavin, and Y. Sasson. A lower bound for broadcasting in mobile ad hoc networks. *Tech. Rep. IC/2004/37 École Polytechnique Fédérale de Lausanne, Switzerland*, June.

[136] MG Rabbat, MJ Coates, and RD Nowak. Multiple-source internet tomography. *IEEE Journal on Selected Areas in Communications*, 24(12):2221–2234, 2006.

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

[138] R.T. Rockafellar. *Network flows and monotropic optimization.* 1984.

[139] Y.E. Sagduyu and A. Ephremides. On joint MAC and network coding in wireless ad hoc networks. *IEEE Trans. Information Theory*, 53(10):3697–3713, 2007.

[140] Y.E. Sagduyu and A. Ephremides. Cross-layer optimization of MAC and network coding in wireless queueing tandem networks. *IEEE Trans. Information Theory*, 54(2):554–571, 2008.

[141] F. Santosa and W.W. Symes. Linear inversion of band-limited reflection seismograms. *SIAM Journal on Scientific and Statistical Computing*, 7:1307, 1986.

[142] P. Sarnak. What is an expander? *American Mathematical Society.*, 51:762–770, 2004.

[143] C. Sasaki, A. Tagami, and S. Ano. Implementation approach for network coding using external devices in ip multicast. In *IEEE/IPSJ Intl. Symposium on Applications and the Internet, KDDI R&D Labs., Saitama, Japan*, 2010.

[144] P. Sattari, A. Markopoulou, and C. Fragouli. Multiple source multiple destination topology inference using network coding. In *Network Coding, Theory, and Applications, 2009. NetCod'09. Workshop on*, pages 36–41. IEEE.

[145] G. Sharma, S. Jaggi, and BK Dey. Network tomography via network coding. In *Information theory and applications workshop*, pages 151–157, 2008.

[146] G. Sharma, S. Jaggi, and BK Dey. Network tomography via network coding. In *Information Theory and Applications Workshop*, pages 151–157, 2008.

[147] ER Sheinerman. MATGRAPH: A matlab toolbox for graph theory. *Software available at: http://www. ams. jhu. edu/˜ ers/matgraph*, 2007.

[148] A. Silberschatz, P.B. Galvin, G. Gagne, and A. Silberschatz. *Operating system concepts*. Addison-Wesley New York, 1994.

[149] M. Stojnic, W. Xu, and B. Hassibi. Compressed sensing-probabilistic analysis of a null-space characterization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008.

[150] A. Ta-Shma, C. Umans, and D. Zuckerman. Lossless condensers, unbalanced expanders, and extractors. *Combinatorica*, 27(2):213–240, 2007.

[151] K. Tan, H. Liu, J. Zhang, Y. Zhang, J. Fang, and G.M. Voelker. Sora: high-performance software radio using general-purpose multi-core processors. *Communications of the ACM journal*, 54(1):99–107, 2011.

[152] A.S. Tanenbaum. Computer networks, 2003. *ISBN-10*, 130661023.

[153] A.S. Tanenbaum. *computer networks*. Prentice Hall, fifth edition, 2010.

[154] S. Tang, H. Yomo, T. Ueda, R. Miura, and S. Obana. Achieving full rate network coding with constellation compatible modulation and coding. In *Proc. IEEE GLOBECOM*, Dec 2010.

[155] J.A. Tropp and A.C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Information Theory*, 53(12):4655–4666, 2007.

[156] 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)*, volume 3, 2001.

[157] Y. Tsang, M. Coates, and R.D. Nowak. Network delay tomography. *IEEE Trans. Signal Processing*, 51(8):2125–2136, 2003.

[158] P. Van Mieghem and FA Kuipers. On the complexity of qos routing. *Computer Communications*, 26(4):376–387, 2003.

[159] Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the American Statistical Association*, 91(433), 1996.

[160] D. Vukobratović, Č. Stefanović, and V. Crnojević. On low-complexity network coding for data collection in wireless sensor networks. 2008.

[161] M.B. Wakin. An introduction to compressive sampling. *IEEE Signal Pprocessing Magazine*, 25(2):21–30, 2008.

[162] M. Wang and B. Li. How practical is network coding? In *Proceedings of 14th IEEE International Workshop on Quality of Service*, pages 274–278, 2006.

[163] M. Wang and B. Li. R2: Random push with random network coding in live peer-to-peer streaming. *IEEE J. Selected Areas in Communications*, 25(9):1655–1666, 2007.

[164] M. Wang, W. Xu, E. Mallada, and A. Tang. Sparse recovery with graph constraints: Fundamental limits and measurement construction. *Arxiv preprint arXiv:1108.0443*, 2011.

[165] J. Winick and S. Jamin. Inet-3.0: Internet topology generator. 2002.

[166] Y. Wu, P.A. Chou, and S.Y. Kung. Information exchange in wireless networks with network coding and physical-layer broadcast. In *Proc. of Conference on Information Sciences and Systems*, Mar. 2005.

[167] Y. Wu, P.A. Chou, and S.Y. Kung. Minimum-energy multicast in mobile ad hoc networks using network coding. *IEEE Trans. Communications*, 53(11):1906–1918, 2005.

[168] B. Xi, G. Michailidis, and V.N. Nair. Estimating network loss rates using active tomography. *Journal of the American Statistical Association*, 101(476):1430–1448, 2006.

[169] Y. Xia and D. Tse. Inference of link delay in communication networks. *IEEE Journal on Selected Areas in Communications*, 24(12):2235–2248, 2006.

[170] W. Xiumin, W. Jianping, and X. Yinlong. Data dissemination in wireless sensor networks with network coding. *EURASIP Journal on Wireless Communications and Networking*, 2010.

[171] B. Xu, A. Ouksel, and O. Wolfson. Opportunistic resource exchange in inter-vehicle ad-hoc networks. In *Proc. of IEEE Intl Conf. Mobile Data Management*, pages 4–12, 2004.

[172] W. Xu and B. Hassibi. Efficient compressive sensing with deterministic guarantees using expander graphs. In *IEEE information theory workshop*, pages 414–419, 2007.

[173] W. Xu, E. Mallada, and A. Tang. Compressive sensing over graphs. In *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, pages 2087–2095, 2011.

[174] F. Xue, C.H. Liu, and S. Sandhu. MAC-layer and PHY-layer network coding for two-way relaying: Achievable regions and opportunistic scheduling. In *Proc. Annual Allerton Conference on Communication, Control and Computing*, volume 128, 2007.

[175] H. Yao, S. Jaggi, and M. Chen. Passive network tomography for erroneous networks: A network coding approach. *Arxiv preprint arXiv:0908.0711*, 2009.

[176] F. Ye, S. Roy, and H. Wang. Efficient inter-vehicle data dissemination. In *accepted to int'l symposium on Wireless Vehicular Communications*, 2011.

[177] H. Yomo and P. Popovski. Opportunistic scheduling for wireless network coding. In *IEEE Intl. Conf. Communications (ICC'07)*, pages 5610–5615, 2007.

[178] B. Yu, J. Cao, D. Davis, and S. Vander Wiel. Time-varying network tomography: router link data. In *Proc. IEEE International Symposium on Information Theory*, page 79, 2000.

[179] R. Yuster and U. Zwick. Fast sparse matrix multiplication. *ACM Transactions on Algorithms*, 1(1):2–13, 2005.

[180] J. Zhang, K. Tan, S. Xiang, Q. Yin, Q. Luo, Y. He, J. Fang, and Y. Zhang. Experimenting software radio with the sora platform. volume 40, pages 469–470, 2010.

[181] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu. Spatio-temporal compressive sensing and internet traffic matrices. *ACM SIGCOMM Computer Communication Review*, 39(4):267–278, 2009.

[182] J. Zhao, M. Kuhn, A. Wittneben, and G. Bauch. Asymmetric data rate transmission in two-way relaying systems with network coding. In *Proc. IEEE Intl. Conference on Communications*, May 2010.

[183] J. Zhu and S. Roy. MAC for dedicated short range communications in intelligent transport system. *IEEE Communications Magazine*, 41(12):60–67, 2003.

Appendix A

PROOFS OF THEOREMS

A.1 Proofs of Theorems: Section 2.1

Proof of Lemma 1: Let $e_1, e_2, ..., e_K$ be K outgoing links of source s. Recall that $\mathcal{P}_{e_i}$ is defined as set of paths from source to destination that have e_i in common and $\mathcal{P}_{e_1}(d), ..., \mathcal{P}_{e_K}(d)$ is a K-partition of $\mathcal{P}(d)$. Let $P_{e_i}^k$ represent the k-th element of $\mathcal{P}_{e_i}(d)$; i.e. $P_{e_i}^k$ is the k-th path among all paths from s to d passing through e_i. Let $N_i = |\mathcal{P}_{e_i}|$, total number of paths from source to destination starting at e_i. Consider the following definitions:

$$\beta_{k,e_i} = \prod_{j \in P_{e_i}^k} \gamma_j, \tag{A.1}$$

$$\boldsymbol{\beta}_{e_i} = [\beta_{k,e_i}]_{k=1}^{N_i}, \tag{A.2}$$

$$(\boldsymbol{\beta}(G))^T = [\beta_{e_i}]_{i=1}^{K}. \tag{A.3}$$

β_{k,e_i} is the product of linear network coding coefficients of all links on k-th path in $\mathcal{P}_{e_i}$. Vector $\boldsymbol{\beta}(G)$ defined above is a rearranged version of $\boldsymbol{\beta}(G)$ in (11). Let β_{k,e_i} be elements in field $\mathbb{F}_{2^q}$ with the following property for each $i = 1, \ldots, K$:

$$\sum_{k=1}^{N_i} \zeta_k \beta_{k,e_i} = 0, \zeta_k \in \{0,1\} \Leftrightarrow \zeta_k = 0, \ \forall k. \tag{A.4}$$

We shall show that $\boldsymbol{\beta}(G)$ with above property satisfies both conditions in lemma 1. It is worth mentioning that since we are dealing with a logical network, for any two paths there is at least one link that belongs to one but not the other. Hence for any two end-to-end network coding coefficients β_{k,e_i} and β_{k',e_j} $\forall k, k', i, j$, there is a link network coding coefficient γ_l which appears in only one of them. That means from a network designer perspective, β_{k,e_i}s are independent and hence they can be designed separately.

Since in the extended binary field $\mathbb{F}_{2^q}$ addition and subtraction are equivalent, $\mathbf{A}\boldsymbol{\beta}(G) \neq \mathbf{A}\boldsymbol{\beta}(G_l)$ implies $\mathbf{A}(\ \boldsymbol{\beta}(G) + \boldsymbol{\beta}(G_l)\ ) \neq 0$. Define $\tilde{\boldsymbol{\beta}}(G_l) = \boldsymbol{\beta}(G) + \boldsymbol{\beta}(G_l)$. According to (13),

$\tilde{\boldsymbol{\beta}}(G_l)$ can be written as

$$\tilde{\beta}_{k,e_i} = \begin{cases} \beta_{k,e_i} & if \;\; l \in P^k_{e_i}(d) \\ 0 & \text{otherwise} \end{cases}, \tag{A.5}$$

$$\tilde{\boldsymbol{\beta}}_{e_i} = [\tilde{\beta}_{k,e_i}]_{k=1}^{N_i}, \tag{A.6}$$

$$\tilde{\boldsymbol{\beta}}(G_l) = [\tilde{\boldsymbol{\beta}}_{e_i}]_{i=1}^{K}. \tag{A.7}$$

where $\tilde{\boldsymbol{\beta}}(G_l)$ contains network coding coefficient of the paths which go through link l and is zero otherwise. According to broadcast nature of network coding, there is a path from source to destination which goes through link l. Therefore, there is at least one β_{k,e_i} which is not zero, say $\beta_{k^*,e_{i^*}} \neq 0$. On the other hand, $\forall i \in \{1,2,...,K\}$ $\beta_{k,e_i}, k = 1,2,...,N_i$ are selected as in (A.4). Now, define matrix $\mathbf{A} = [a_{i,j}]$ as:

$$a_{i,j} = \begin{cases} 1 & \sum_{k=1}^{j-1} N_k < i \leq \sum_{k=1}^{j} N_k \\ 0 & \text{otherwise} \end{cases}. \tag{A.8}$$

corresponding to the following - in time slot i, source sends symbol $\mathbf{1} \in \mathbb{F}_{2^q}$ over i^{th} outgoing link and zero over others. An example of matrix $\mathbf{A}$ for network in Figure 3 is given below. As it is explained in Example 2, for that network we have $N = 5$, $K = 3$ and $N_1 = 2$, $N_2 = 1$, $N_3 = 2$.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Now let look at the i^* entry of vector $\mathbf{A}\,\tilde{\boldsymbol{\beta}}(G_l)$. Since in time slot i^* source sends $\mathbf{1}$ over e_{i^*} and zero otherwise, the i^* entry of $\mathbf{A}\,\tilde{\boldsymbol{\beta}}(G_l)$ is calculated as follows:

$$\sum_{k=1}^{N_{i^*}} \tilde{\beta}_{k,e_{i^*}}, \tag{A.9}$$

which can be rewritten as

$$\sum_{k=1}^{N_{i^*}} \zeta_k \beta_{k,e_{i^*}}, \tag{A.10}$$

where ζ_k is one if $\tilde{\beta}_{k,e_{i^*}} = \beta_{k,e_{i^*}}$ and is zero otherwise. Since $\tilde{\beta}_{k^*,e_{i^*}} \neq 0$, or equivalently $\tilde{\beta}_{k^*,e_{i^*}} = \beta_{k^*,e_{i^*}}$ (according to Eq. (A.7), $\forall i,k$ $\tilde{\beta}_{k,e_i}$ is either zero or equal β_{k,e_i}), in Eq. (A.10), $\zeta_{k^*} \neq 0$. For that reason, Eq. (A.4) guarantees that Eq. (A.10) is not zero which means i^* entry of $\mathbf{A}\,\tilde{\boldsymbol{\beta}}(G_l)$ is not zero and consequently $\mathbf{A}\,\tilde{\boldsymbol{\beta}}(G_l) \neq 0$

Now we prove matrix $\mathbf{A}$ defined in (A.8) and $\boldsymbol{\beta}$ defined in (A.4) also satisfy the second condition of the theorem. With the same argument, it is sufficient to show that $\mathbf{A}(\boldsymbol{\beta}(G_{l_1}) + \boldsymbol{\beta}(G_{l_2})) \neq 0$. Let $\tilde{\boldsymbol{\beta}}(G_{l_1}, G_{l_2}) = \boldsymbol{\beta}(G_{l_1}) + \boldsymbol{\beta}(G_{l_2})$, using definition of $\boldsymbol{\beta}(G_l)$ in (13) it is easily seen that $\tilde{\boldsymbol{\beta}}(G_{l_1}, G_{l_2})$ has the following structure:

$$\tilde{\beta}_{k,e_i} = \begin{cases} 0 \quad \text{if } (l_1 \notin P^i(d) \text{ and } l_2 \notin P^i(d)) \text{ or} \\ \qquad (l_1 \in P^i(d) \text{ and } l_2 \in P^i(d)) \\ \beta_{k,e_i}(G) \qquad \text{otherwise} \end{cases}, \tag{A.11}$$

$$\tilde{\boldsymbol{\beta}}_{\boldsymbol{e_i}} = [\tilde{\beta}_{k,e_i}]_{k=1}^{N_i}, \tag{A.12}$$

$$\tilde{\boldsymbol{\beta}}(G_{l_1}, G_{l_2}) = [\tilde{\boldsymbol{\beta}}_{\boldsymbol{e_i}}]_{i=1}^{i=K}. \tag{A.13}$$

Basically, $\tilde{\boldsymbol{\beta}}(G_{l_1}, G_{l_2})$ is non-zero over paths which go through one of the links but not both. On the other hand, we have assumed that the network is logical which means there is a path which goes through one of l_1 or l_2 and not both of them. Hence, $\tilde{\beta}(G_{l_1}, G_{l_2})$ has at least one non-zero element, say $\tilde{\beta}_{k^*,e_{i^*}} \neq 0$. By definition of $\tilde{\beta}_{k,e_i}$, if $\tilde{\beta}_{k^*,e_{i^*}}$ is not zero it is equal β_{k,e_i}. By the same argument as before the i^* entry of matrix production $\mathbf{A}\tilde{\boldsymbol{\beta}}(G_{l_1}, G_{l_2})$ is not zero and consequently $\mathbf{A}\tilde{\boldsymbol{\beta}}(G_{l_1}, G_{l_2}) \neq 0$ which implies $\mathbf{A}\boldsymbol{\beta}(G_{l_1}) \neq \mathbf{A}\boldsymbol{\beta}(G_{l_2})$.

■

Proof of Theorems 2 and 3: See the proof of Lemma 1.

■

Proof of Theorem 4:

We prove the theorem by using Schwartz-Zippel Lemma [84], given as follows:

Schwartz-Zippel Lemma: If f, g are two different m-variate polynomial of degree at most d over $\mathbb{F}$, then $P(f = g) \leq \frac{d}{|\mathbb{F}|}$.

There is a status ambiguity between link $l_1 \in E$ and $l_2 \in E$ if received symbols at the destination in case of l_1 congestion be the same as received symbols when l_2 is congested, $\mathbf{y}^{l_1} = \mathbf{y}^{l_2}$. Using above lemma this can happen with the following probability:

$$\begin{aligned} & P(\mathbf{y}^{l_1} = \mathbf{y}^{l_2}) \\ = \;& P(\mathbf{A}\boldsymbol{\beta}(G_{l_1}) = \mathbf{A}\boldsymbol{\beta}(G_{l_2})) \\ = \;& P(\boldsymbol{\alpha}^T[i]\boldsymbol{\beta}(G_{l_1}) = \boldsymbol{\alpha}^T[i]\boldsymbol{\beta}(G_{l_2}), i = 1, 2..., M) \\ \overset{(1)}{=} \;& \prod_{i=1}^{M} P(\boldsymbol{\alpha}^T[i]\boldsymbol{\beta}(G_{l_1}) = \boldsymbol{\alpha}^T[i]\boldsymbol{\beta}(G_{l_2})) \\ \leq \;& \prod_{i=1}^{M} \frac{1}{2^q} = (\frac{1}{2^q})^M, \end{aligned} \tag{A.14}$$

where in equality (1) we are assuming statistical independence of training symbols over time, $\boldsymbol{\alpha}[i]$. In the last inequality we use Schwartz-Zippel lemma where $d = 1$ and $|\mathbb{F}_{2^q}| = 2^q$.

link $l_1 \in E$ is not identifiable if there is a link $l_2 \in E \cup \{\phi\}$ ($\{\phi\}$ stands for no congestion in the network) such that there is status ambiguity between them. The probability that l_1 is not identifiable is calculated as below:

$$\begin{aligned} P(l_1 \text{ is not identifiable}) &= P(\mathbf{y}^{l_1} = \mathbf{y}^{l_2},\, l_2 \in \tilde{E}) \\ &= \sum_{l_2 \in \tilde{E}} P(\mathbf{y}^{l_1} = \mathbf{y}^{l_2}) \\ &\leq (|E| + 1)(\frac{1}{2^q})^M, \end{aligned} \tag{A.15}$$

where $\tilde{E} = E \cup \{\phi\}$.

Graph $G(V, E)$ is identifiable if all links are identifiable. Using above probabilities for identifiability of link $l_1 \in E$, probability of G be identifiable is given as follows:

$$\begin{aligned} P(G \text{ is identifiability}) &= 1 - P(G \text{ is not identifiability}) \\ &= 1 - P(l \text{ is not identifiable}, l \in E) \\ &= 1 - \sum_{l \in E} P(l \text{ is not identifiable}) \\ &\geq 1 - \sum_{l \in E} (|E| + 1)(\frac{1}{2^q})^M \\ &= 1 - |E|(|E| + 1)(\frac{1}{2^q})^M. \end{aligned} \tag{A.16}$$

■

Proof of Theorem 5: Let $H = \{H_1, H_2, ..., H_M\}$ be a M-partition of $Z = \{1, 2, \ldots, K\}$; i.e. $Z = \cup_{i=1}^{M} H_i$ and $H_i \cap H_j = \phi\,, i \neq j$. Suppose for each j, $1 \leq j \leq M$, NC coefficients β_{k,e_i}, $k = 1, ..., N_i$ and $i \in H_j$, have following property:

$$\sum_{i \in H_j} \sum_{k=1}^{N_i} \zeta_{i,k} \beta_{k,e_i} = 0 \Leftrightarrow \zeta_{i,k=0},\ i \in H_j, k = 1, 2, ..., N_i. \tag{A.17}$$

Now, define matrix $\mathbf{A} = [a_{i,j}]$ as the following; In time slot j, source sends symbol $\mathbf{1} \in \mathbb{F}_{2^q}$ over i^{th} outgoing link where $i \in H_j$ and zero over others. By same argument as given in proof of Lemma 1, matrix $\mathbf{A}$ satisfies the following two inequalities:

- $\mathbf{A}\boldsymbol{\beta}(G_l) \neq \mathbf{A}\boldsymbol{\beta}, \forall l_1 \in E$
- $\mathbf{A}\boldsymbol{\beta}(G_{l_1}) \neq \mathbf{A}\boldsymbol{\beta}(G_{l_2}), \forall l_1, l_2 \in E$

which means $G(V, E)$ is identifiable using $\mathbf{A}$ as training sequence.

In finite filed $\mathbb{F}_{2^q}$ there are q numbers φ_k, $k = 1, \ldots, q$ such that [123]:

$$\sum_{k=1}^{q} \zeta_k \varphi_k = 0 \Leftrightarrow \zeta_k = 0, \forall k. \tag{A.18}$$

This property and the condition given in Eq. (A.17) for β_{k,e_i} contribute to the fact that, for a given H, $G(V, E)$ is identifiable if $q \geq \sum_{i \in H_j} N_i$ for all $1 \leq j \leq M$ or equivalently

$q \geq \max_j \sum_{i \in H_j} N_i$. In the other words, for a given partition of Z, $G(V, E)$ is identifiable using M time slots and q bits for NC coefficients if $q \geq \max_j \sum_{i \in H_j} N_i$. Therefore, $G(V, E)$ is identifiable if q is the smallest number amount all partitions of Z; i.e. if q satisfies the following min-max inequality:

$$q \geq \min_{\{H_i, i=1,...,M\} \in \mathcal{Z}_M} \max_i \sum_{j \in H_i} N_j. \tag{A.19}$$

■

Proof of Theorems 8 and 9: See the proof of Theorem 5.

■

A.2 Proofs of Theorems: Section 2.2

Lemma 2:

We first proof the following lemma which characterizes the null space of bi-adjacency matrix of an expander graph and will be used to bound the error in the recovery of $\mathbf{x}$ from its compressed projection $\mathbf{y}$.

lemma 11 *Let $G(V_1, V_2, E)$ be a $(2, d, \epsilon)$-expander with $\epsilon \leq 1/4$ and $\mathbf{A}_{m \times n}$ be its bi-adjacency matrix. Assume $\mathbf{w}$ lies in the null space of $\mathbf{A}$ (i.e., $\mathbf{A}\mathbf{w} = \mathbf{0}$) and let S be any singleton set of coordinates of the $\mathbf{w}$, i.e., $S = \{i\}$, $i \in \{1, ..., n\}$. Then*

$$\| \mathbf{w}_S \|_1 \leq 2\epsilon \| \mathbf{w}_{S^c} \|_1 . \tag{A.20}$$

Proof:

Let $\mathbf{A}'$ be the submatrix of $\mathbf{A}$ containing rows from $N(S)$. Since $|S| = 1$ and graph is left d-regular, $\| \mathbf{A}'\mathbf{w}_S \|_1 = \| \mathbf{A}\mathbf{w}_S \|_1 = d \| \mathbf{w}_S \|_1$. We have

$$\begin{aligned} 0 = \| \mathbf{A}'\mathbf{w} \|_1 &= \| \mathbf{A}'\mathbf{w}_S + \mathbf{A}'\mathbf{w}_{S^c} \|_1 \\ &\geq \| \mathbf{A}'\mathbf{w}_S \|_1 - \| \mathbf{A}'\mathbf{w}_{S^c} \|_1 \\ &= d \| \mathbf{w}_S \|_1 - \| \mathbf{A}'\mathbf{w}_{S^c} \|_1 . \end{aligned} \tag{A.21}$$

Each set of two nodes in the left part has at least $2(1-\epsilon)d$ neighbor nodes on the right side (expansion definition). Since each node at the left has degree d, number of common nodes on the right-hand side [1] is at most $2d - 2(1-\epsilon)d = 2\epsilon d$. That means each column of $\mathbf{A}'$ (except the one corresponding to S) has at most $2\epsilon d$ number of ones, yielding,

$$\| \mathbf{A}'\mathbf{w}_{S^c} \|_1 \leq 2\epsilon d \| \mathbf{w}_{S^c} \|_1 . \tag{A.22}$$

Therefore, we have

$$\| \mathbf{w}_S \|_1 \leq 2\epsilon \| \mathbf{w}_{S^c} \|_1, \tag{A.23}$$

or equivalently

$$\| \mathbf{w}_S \|_1 \leq \frac{2\epsilon}{1+2\epsilon} \| \mathbf{w} \|_1 . \tag{A.24}$$

The above argument is valid if G is a $(2, d, \epsilon)$-expander graph. By the same argument given at the beginning of Section 2.2.4, one can proof that $\epsilon \leq 1/4$.

□

Now, let $\mathbf{y} = \mathbf{x}' - \mathbf{x}$. Clearly $\mathbf{y} \in \mathcal{N}(\mathbf{A})$ and we have:

$$\begin{aligned}
\| \mathbf{x} \|_1 &\geq \| \mathbf{x}' \|_1 \qquad (A.25)\\
&= \| (\mathbf{x}+\mathbf{y})_S \|_1 + \| (\mathbf{x}+\mathbf{y})_{S^c} \|_1 \\
&= \| \mathbf{x}_S + \mathbf{y}_S \|_1 + \| \mathbf{x}_{S^c} + \mathbf{y}_{S^c} \|_1 \\
&\geq \| \mathbf{x}_S \|_1 - \| \mathbf{y}_S \|_1 + \| \mathbf{y}_{S^c} \|_1 - \| \mathbf{x}_{S^c} \|_1 \\
&= \| \mathbf{x} \|_1 - 2 \| \mathbf{x}_{S^c} \|_1 + \| \mathbf{y} \|_1 - 2 \| \mathbf{y}_S \|_1 \\
&\geq \| \mathbf{x} \|_1 - 2 \| \mathbf{x}_{S^c} \|_1 + (1 - \frac{4\epsilon}{1+2\epsilon}) \| \mathbf{y} \|_1,
\end{aligned}$$

where in the last equality, Eq. (A.24) is used. Therefore we have:

$$\| \mathbf{x}' - \mathbf{x} \|_1 = \| \mathbf{y} \|_1 \leq f(\epsilon) \| \mathbf{x}_{S^c} \|_1, \tag{A.26}$$

where $f(\epsilon) = \frac{2(1+2\epsilon)}{1-2\epsilon}$.

■

[1] Nodes on the right which are connected to both nodes on the left-hand side.

Theorem 9:

Let $\mathbf{x}'$ be the solution to the optimization problem in Eq. (19). That means $\mathbf{R}\mathbf{x}' = \mathbf{R}\mathbf{x}$ and $\| \mathbf{x}' \|_1 \leq \| \mathbf{x} \|_1$. On the other hand, G is a $(2, d, \epsilon)$-expander graph with the bi-adjacency matrix $\mathbf{R}$. Consequently, Eq. (2.40) in Lemma 2 holds for $\mathbf{x}$ and $\mathbf{x}'$.

■

Theorem 10:

Theorem 10:

We prove the theorem for the case in which $G(X, Y, H)$ has only two expander subgraphs. The general case can be easily extended by using similar argument. Let $G_1(X_1, Y, H_1)$ with $|X_1| = m$, and $G_2(X_2, Y, H_2)$ with $|X_2| = n - m$, be two d_i-regular ($d_1 \neq d_2$) subgraphs of $G(X, Y, H)$ with bi-adjacency matrices $\mathbf{R}_1$ and $\mathbf{R}_2$, respectively. Without loss of generality, we rename the elements in X such that $\mathbf{R} = [\mathbf{R}_1 \, \mathbf{R}_2]$.

Now, suppose two 1-sparse vectors $\mathbf{u}$ and $\mathbf{v}$ have the same projection for matrix $\mathbf{R}_{r \times n}$, i.e., $\mathbf{R}\mathbf{u} = \mathbf{R}\mathbf{v}$. Without loss of generality, we assume $\| \mathbf{u} \|_1 \geq \| \mathbf{v} \|_1$. Let $\mathbf{w} = \mathbf{v} - \mathbf{u}$. Clearly, $\mathbf{w}$ belongs to the null space of $\mathbf{R}$. Moreover, let $\mathbf{u} = [\mathbf{u}_1^t \mathbf{u}_2^t]^t$, $\mathbf{v} = [\mathbf{v}_1^t \mathbf{v}_2^t]^t$ and $\mathbf{w} = [\mathbf{w}_1^t \mathbf{w}_2^t]^t$. Clearly, the following equalities hold:

$$\begin{aligned} \mathbf{w}_1 &= \mathbf{v}_1 - \mathbf{u}_1, \\ \mathbf{w}_2 &= \mathbf{v}_2 - \mathbf{u}_2. \end{aligned} \tag{A.27}$$

Let $S \subset \{1, 2, \ldots, n\}$ be any singleton set of coordinates ($k = 1$) of $\mathbf{w}$. Further, let $\mathbf{R}'$ be submatrix of $\mathbf{R}$ containing rows from $N(S)$. We consider the following two cases:

Case 1: $S \subset \{1, 2, \ldots, m\}$:

In this case S represents a node in $G_1(X_1, Y, H_1)$ which is, by assumption, a $(2, d_1, \epsilon)$-expander. Similar to proof of Lemma 1, $\| \mathbf{R}'\mathbf{w}_S \|_1 = \| \mathbf{R}\mathbf{w}_S \|_1 = d_1 \| \mathbf{w}_S \|_1$. Thus,

$$\begin{aligned} 0 = \| \mathbf{R}'\mathbf{w} \|_1 &= \| \mathbf{R}'\mathbf{w}_S + \mathbf{R}'\mathbf{w}_{S^c} \|_1 \\ &\geq \| \mathbf{R}'\mathbf{w}_S \|_1 - \| \mathbf{R}'\mathbf{w}_{S^c} \|_1 \\ &= d_1 \| \mathbf{w}_S \|_1 - \| \mathbf{R}'\mathbf{w}_{S^c} \|_1 . \end{aligned} \tag{A.28}$$

Let $\mathbf{r}_i^t$ denote the i-th rows of $\mathbf{R}'$. The bipartite graph G_1 is left d_1-regular and hence matrix $\mathbf{R}'$ has d_1 rows. Therefore, we can put an upper bound on $\| \mathbf{R}'\mathbf{w}_{S^c} \|_1$ as below

$$\begin{aligned} \| \mathbf{R}'\mathbf{w}_{S^c} \|_1 &= \sum_{i=1}^{d_1} |\mathbf{r}_i^t \mathbf{w}_{S^c}| \\ &= \sum_{i=1}^{d_1} \left| \sum_{j=1}^{n} r_{ij} (w_{S^c})_j \right| \\ &\overset{(1)}{\leq} \sum_{i=1}^{d_1} \sum_{j=1}^{n} r_{ij} |(w_{S^c})_j| \\ &= \sum_{j=1}^{n} \sum_{i=1}^{d_1} r_{ij} |(w_{S^c})_j| \\ &= \sum_{j=1}^{n} |(w_{S^c})_j| \sum_{i=1}^{d_1} r_{ij} \\ &= \sum_{j=1}^{m} |(w_{S^c})_j| \sum_{i=1}^{d_1} r_{ij} + \sum_{j=m+1}^{n} |(w_{S^c})_j| \sum_{i=1}^{d_1} r_{ij}, \end{aligned} \tag{A.29}$$

where for inequality (1), we used the triangular inequality and the fact that $r_{ij} \in \{0, 1\}$. Since $G_1(X_1, Y, H_1)$ is an $(2, d_1, \epsilon)$-expander, each two nodes at the left-hand side have at most $2\epsilon d_1$ neighbors at the right in common. That means, each column in $\mathbf{R}'$ has at most $2\epsilon d$ nonzero entries (i.e., $\sum_{i=1}^{d_1} r_{ij} \leq 2\epsilon d_1$) for any $j = \{1, 2, ..., m\}$. On the other hand, since $\mathbf{R}'$ has d_1 rows, $\sum_{i=1}^{d_1} r_{ij} \leq d_1$ for each $j = \{m+1, m+2, ..., n\}$. These facts and Eq. (A.29) result in:

$$\begin{aligned} \| \mathbf{R}'\mathbf{w}_{S^c} \|_1 &\leq \sum_{j=1}^{m} \left(|(w_{S^c})_j| \sum_{i=1}^{d_1} r_{ij} \right) + \sum_{j=m+1}^{n} \left(|(w_{S^c})_j| \sum_{i=1}^{d_1} r_{ij} \right) \\ &\leq 2\epsilon d_1 \sum_{j=1}^{m} |(w_{S^c})_j| + d_1 \sum_{j=m+1}^{n} |(w_{S^c})_j| \\ &= 2\epsilon d_1 \| \mathbf{w}_{1S^c} \|_1 + d_1 \| \mathbf{w}_2 \| . \end{aligned}$$

Substituting above inequality in Eq. (A.28), we have:

$$0 \geq d_1 \| \mathbf{w}_S \|_1 - 2\epsilon d_1 \| \mathbf{w}_{1S^c} \|_1 - d_1 \| \mathbf{w}_2 \|_1 . \tag{A.30}$$

Therefore, we derive the following upper bound

$$\| \mathbf{w}_S \|_1 \leq 2\epsilon \| \mathbf{w}_{1S} \|_1 + \| \mathbf{w}_2 \|_1, \tag{A.31}$$

which can be rewritten as:

$$\| \mathbf{w}_S \|_1 \leq \frac{2\epsilon}{1+2\epsilon} \| \mathbf{w}_1 \|_1 + \frac{1}{1+2\epsilon} \| \mathbf{w}_2 \| . \tag{A.32}$$

By assumption we have $\| \mathbf{u} \|_1 \geq \| \mathbf{v} \|_1$. The above property for $\mathbf{w} = \mathbf{v} - \mathbf{u}$ provides the following lower bound on $\mathbf{u}$:

$$\begin{aligned} \| \mathbf{u} \|_1 &\geq \| \mathbf{v} \|_1 \\ &= \| \mathbf{u}_1 + \mathbf{w}_1 \|_1 + \| \mathbf{u}_2 + \mathbf{w}_2 \|_1 \\ &= \| \mathbf{u}_{1S} + \mathbf{w}_{1S} \|_1 + \| \mathbf{u}_{1S^c} + \mathbf{w}_{1S^c} \|_1 + \\ &\quad \| \mathbf{u}_2 + \mathbf{w}_2 \|_1 \\ &\geq \| \mathbf{u}_{1S} \|_1 - \| \mathbf{w}_{1S} \|_1 + \| \mathbf{w}_{1S^c} \|_1 - \| \mathbf{u}_{1S^c} \|_1 + \\ &\quad \| \mathbf{w}_2 \|_1 - \| \mathbf{u}_2 \|_1 \\ &= \| \mathbf{u}_{1S} \|_1 - (\| \mathbf{u}_{1S^c} \|_1 + \| \mathbf{u}_2 \|_1) + \\ &\quad (\| \mathbf{w}_{1S^c} \|_1 + \| \mathbf{w}_2 \|_1) - \| \mathbf{w}_{1S} \|_1 \end{aligned} \tag{A.33}$$

Since $S \subset \{1, 2, ..., m\}$, we have $\| \mathbf{w}_2 \|_1 + \| \mathbf{w}_{1S^c} \|_1 = \| \mathbf{w}_{S^c} \|_1$ and $\| \mathbf{u}_2 \|_1 + \| \mathbf{u}_{1S^c} \|_1 = \| \mathbf{u}_{S^c} \|_1$. So Eq. (A.33) can be simplified as bellow:

$$\begin{aligned} 2 \| \mathbf{u}_{S^c} \|_1 &\geq \| \mathbf{w}_{S^c} \|_1 - \| \mathbf{w}_{1S} \|_1 \\ &= \| \mathbf{w} \|_1 - 2 \| \mathbf{w}_{1S} \|_1 . \end{aligned} \tag{A.34}$$

By using Eq. (A.32) we have:

$$2 \| \mathbf{u}_{S^c} \|_1 \geq \frac{1-2\epsilon}{1+2\epsilon} \| \mathbf{w_1} \|_1 - \frac{1-2\epsilon}{1+2\epsilon} \| \mathbf{w_2} \|_1 . \tag{A.35}$$

By Eq. (A.27) and triangular inequality we have:

$$\| \mathbf{w}_2 \|_1 \leq \| \mathbf{u}_2 \|_1 + \| \mathbf{v}_2 \|_1 . \tag{A.36}$$

Applying above inequality to (A.35) we have:

$$\frac{1+2\epsilon}{1-2\epsilon} \left[2 \|\mathbf{u}_{1S^c}\|_1 + \frac{1-2\epsilon}{1+2\epsilon} \big(\|\mathbf{u}_2\|_1 + \|\mathbf{v}_2\|_1 \big) \right] \geq \|\mathbf{w}\|_1 . \tag{A.37}$$

Clearly $\| \mathbf{u}_{S^c} \|_1 \geq \| \mathbf{u}_{1S^c} \|_1$, $\| \mathbf{u}_{S^c} \|_1 \geq \| \mathbf{u}_2 \|_1$ and $\| \mathbf{v}_{S^c} \|_1 \geq \| \mathbf{v}_2 \|_1$. Therefore, the following inequalities hold:

$$\frac{3+2\epsilon}{1-2\epsilon} \| \mathbf{u}_{S^c} \|_1 + \| \mathbf{v}_{2S^c} \|_1 \geq \| \mathbf{w} \|_1 . \tag{A.38}$$

Now let $j \in S^c$. There is a path p^* which goes through link j and not the link in S (since it is a logical network). Let $\mathbf{r}_{p^*}$ be the corresponding row for p^* in routing matrix $\mathbf{R}$. Since $\mathbf{Ru} = \mathbf{Rv}$, we have:

$$\mathbf{r}_{p^*}\mathbf{u} = \mathbf{r}_{p^*}\mathbf{v} \geq v_j. \tag{A.39}$$

Since p^* doesn't go through link S, its corresponding entry in $\mathbf{r}_{p^*}$ is zero. Hence we have $\| \mathbf{u}_{S^c} \|_1 \geq \mathbf{r}_{p^*}\mathbf{u}$. Therefore, we can have the following upper bound for every entry of $v_j \ \forall j \in S^c$

$$\| \mathbf{u}_{S^c} \|_1 \geq v_j \ \forall j \in S^c. \tag{A.40}$$

By adding up both side of the inequality for all $j \in S^c$ we have:

$$|S^c| \| \mathbf{u}_{S^c} \|_1 \geq \| \mathbf{v}_{S^c} \|_1 . \tag{A.41}$$

Clearly $n = |X| > |S^c|$. Therefore, the following upper bound is valid for $\mathbf{w} = \mathbf{u} - \mathbf{v}$:

$$(\frac{3+2\epsilon}{1-2\epsilon} + n) \| \mathbf{u}_{S^c} \|_1 \geq \| \mathbf{w} \|_1 . \tag{A.42}$$

Case 2: $S \subset \{m+1, m+2, \ldots, n\}$:

By the same argument as Case 1, we have:

$$\begin{aligned} 0 = \| \mathbf{Rw} \|_1 &= \| \mathbf{R}'\mathbf{w} \|_1 \\ &\geq d_2 \| \mathbf{w}_S \|_1 - \| \mathbf{R}'\mathbf{w}_{S^c} \|_1 . \end{aligned} \tag{A.43}$$

As in case 1, we can put an upper bound on $\| \mathbf{R}'\mathbf{w}_{S^c} \|_1$ as follows

$$\begin{aligned} \| \mathbf{R}'\mathbf{w}_{S^c} \|_1 &\leq \sum_{j=1}^{m} \left(|(w_{S^c})_j| \sum_{i=1}^{d_2} r_{ij} \right) + \sum_{j=m+1}^{n} \left(|(w_{S^c})_j| \sum_{i=1}^{d_2} r_{ij} \right) \\ &\leq d_2 \sum_{j=1}^{m} |(w_{S^c})_j| + 2\epsilon d_2 \sum_{j=m+1}^{n} |(w_{S^c})_j| \\ &\leq d_2 \| \mathbf{w}_1 \|_1 + 2\epsilon d_2 \| \mathbf{w}_{2S^c} \|_1 . \end{aligned}$$

Using above inequality and results from Eq. (A.43), we have the following upper bound for $\| \mathbf{w}_S \|_1$.

$$\| \mathbf{w}_S \|_1 \leq 2\epsilon \| \mathbf{w}_{2S^c} \|_1 + \| \mathbf{w}_1 \|_1, \tag{A.44}$$

Finally, the following property is derived for vector $\mathbf{w} \in \mathcal{N}(\mathbf{R})$:

$$\| \mathbf{w}_S \|_1 \leq \frac{2\epsilon}{1+2\epsilon} \| \mathbf{w}_2 \|_1 + \frac{1}{1+2\epsilon} \| \mathbf{w}_1 \| . \tag{A.45}$$

By the same argument as case 1 we have:

$$(\frac{3+2\epsilon}{1-2\epsilon} + n) \| \mathbf{u}_{S^c} \|_1 \geq \| \mathbf{w} \|_1 . \tag{A.46}$$

Note that set X in $G(X, Y, H)$ is the same as E in network $N(V, E)$. The rest of the proof for LP optimization is the same as Theorem 1.

■

Proof of Corollary 11:

$N(V, E)$ with the routing matrix $\mathbf{R}$ is a 1-identifiable expander. By definition 3, that means bipartite graph $G(X, Y, H)$ with the bi-adjacency matrix $\mathbf{R}$ is a union of left d_i-regular bipartite graphs, $G(X_i, Y, H_i)$ such that:

- $X = \cup X_i$ and $X_i \cap X_j = \emptyset$ for $i \neq j$
- $H = \cup H_i$
- $d_i \neq d_j$ for $i \neq j$
- $G(X_i, Y, H_i)$ is a $(2, d_i, \epsilon)$-expander with $\epsilon \leq \frac{1}{4}$

Now let consider two links l_i and l_j. If $deg(l_i) \neq deg(l_j)$, then clearly one of the first two statements in Corollary 4 would be true. If $deg(l_i) = deg(l_j) = d_i$ they belong to the same subgraph, say, $G(X_i, Y, H_i)$. By the last condition in Definition 3, $G(X_i, Y, H_i)$ is a $(2, d_i, \epsilon)$-expander. Now let $\Phi = \{l_i, l_j\}$. By the definition of the expander graphs in Definition 2, the following holds for Φ:

$$|N(\Phi)| \geq (1-\epsilon)d|\Phi| = 2(1-\epsilon)d. \tag{A.47}$$

By Theorem 2 maximum value possible for ϵ is $\frac{1}{4}$. Therefore minimum value for the right hand side of the above inequality would be achieved if $\epsilon = \frac{1}{4}$ and that is:

$$|N(\Phi)| \geq \frac{3}{2}d. \tag{A.48}$$

$deg(l_i, l_j)$ is defined to be number of nodes connected to both l_i and l_j. We can calculate total number of nodes connected to at least one of l_i and l_j as follows:

$$|N(\Phi)| = deg(l_i) + deg(l_j) - deg(l_i, l_j) = 2d - deg(l_i, l_j). \tag{A.49}$$

Substituting above equality in Eq. (A.48) results in

$$d - 2deg(l_i, l_j) \geq 0, \tag{A.50}$$

which can also be written as follows:

$$2d - 2deg(l_i, l_j) = deg(l_i) + deg(l_j) - 4deg(l_i, l_j) \geq 0. \tag{A.51}$$

■

proof of Theorem 12:

To prove this theorem, we follow the same procedure applied to $k = 1$ (Lemma 2, Theorems 9, and 10). We first assume that $\mathbf{R}$ is a bi-adjacency matrix of a $(2k, d, \epsilon)$-expander graph and prove Lemma 12 which characterizes the null space of $\mathbf{R}$. For a general routing matrix $\mathbf{R}$, where paths are of different lengths (the bipartite graph is not d-regular), the proof is identical to that of Theorem 10 (omitted here for the sake of space saving).

lemma 12 *Assume that* $\mathbf{R}_{r\times n}$ *is a bi-adjacency matrix of a* $(2k, d, \epsilon)$*-expander graph. Let* $\mathbf{w}$ *lie in the null space of* $\mathbf{R}_{r\times n}$ *(i.e.,* $\mathbf{Rw} = \mathbf{0}$*) and let* $\boldsymbol{\mathcal{S}}$ *be any set of coordinates of* $\mathbf{w}$ *with* $|\boldsymbol{\mathcal{S}}| \leq k$, $\boldsymbol{\mathcal{S}} \subset \{1, ..., n\}$. *Then,*

$$\mathbb{E}_{\boldsymbol{\mathcal{S}}}[\| \mathbf{w}_{\boldsymbol{\mathcal{S}}} \|_1] \leq 2\epsilon \mathbb{E}_{\boldsymbol{\mathcal{S}}}[\| \mathbf{w}_{\boldsymbol{\mathcal{S}}^c} \|_1], \tag{A.52}$$

where the expectation is taken over the selection of $\boldsymbol{\mathcal{S}}$.

Proof:

Without loss of generality, we label the n links inside the network $l_1, l_2, \ldots, l_n$ where the delay of link l_j is the jth entry in delay vector $\mathbf{x}$. First, let fix $\boldsymbol{\mathcal{S}} = S$. That is, S is a realization of random variable $\boldsymbol{\mathcal{S}}$. Let $\mathbf{R}'$ be the submatrix of $\mathbf{R}$ that contains rows from $N(S)$. By definition of $\mathbf{w}_S$, we have $\| \mathbf{R}\mathbf{w}_S \|_1 = \| \mathbf{R}'\mathbf{w}_S \|_1$. Thus,

$$\begin{aligned} 0 = \| \mathbf{R}'\mathbf{w} \|_1 &= \| \mathbf{R}'\mathbf{w}_S + \mathbf{R}'\mathbf{w}_{S^c} \|_1 \\ &\geq \| \mathbf{R}'\mathbf{w}_S \|_1 - \| \mathbf{R}'\mathbf{w}_{S^c} \|_1 . \end{aligned} \tag{A.53}$$

First, we impose a lower bound on $\| \mathbf{R}'\mathbf{w}_S \|_1$. By using the result from Lemma 1 in [13], we drive the following inequality:

$$\| \mathbf{R}'\mathbf{w}_S \|_1 \geq d(1-2\epsilon) \| \mathbf{w}_S \|_1 . \tag{A.54}$$

Substituting the above equation into Eq. (A.53) yields:

$$\| \mathbf{R}'\mathbf{w}_{S^c} \|_1 \geq d(1-2\epsilon) \| \mathbf{w}_S \|_1 . \tag{A.55}$$

Equality A.55 is valid for every sample S of random variable $\boldsymbol{\mathcal{S}}$. By taking the expectation over all possible outcomes of $\boldsymbol{\mathcal{S}}$, which is also taken over all possible combinations of congested links, we have

$$\mathbb{E}_{\boldsymbol{\mathcal{S}}}[\| \mathbf{R}'\mathbf{w}_{\boldsymbol{\mathcal{S}}^c} \|_1] \geq d(1-2\epsilon)\mathbb{E}_{\boldsymbol{\mathcal{S}}}[\| \mathbf{w}_{\boldsymbol{\mathcal{S}}} \|_1]. \tag{A.56}$$

Now, we aim to identify an upper bound for $\mathbb{E}_{\boldsymbol{\mathcal{S}}}[\| \mathbf{R}'\mathbf{w}_{\boldsymbol{\mathcal{S}}^c} \|_1]$. Again, let fix $\boldsymbol{\mathcal{S}} = S$. For a path P, define an indicator function $\boldsymbol{\mathcal{I}}_{l \in P}$ as follows:

$$\boldsymbol{\mathcal{I}}_{l \in P} = \begin{cases} 1, & \text{link } l \text{ belongs to path } P; \\ 0, & \text{O.W.} \end{cases} . \tag{A.57}$$

Using the d-regularity assumption of the routing matrix $\mathbf{R}$ means that the link l belongs to exactly d paths. For a link l, therefore, selecting random path P yields a probability $\frac{d}{r}$ for path P passing through l. Equivalently,

$$P(\boldsymbol{\mathcal{I}}_{l \in P} = 1) = \frac{d}{r} = \mathbb{E}[\boldsymbol{\mathcal{I}}_{l \in P}]. \tag{A.58}$$

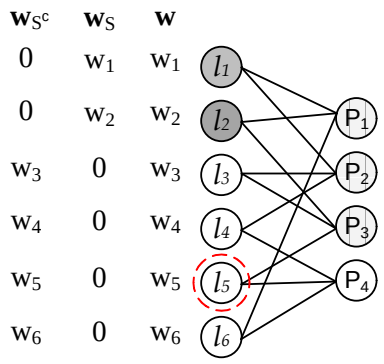


Figure A.1: A realization of $\boldsymbol{\mathcal{S}} = S$. The goal is to find number of paths from the random paths $N(\boldsymbol{\mathcal{S}})$ that pass through link l_j, for a given index j in Eq. (A.60)

Recall that $\mathbf{R}'$ is a submatrix of $\mathbf{R}$ that contains rows from $N(S)$. Thus, each row of $\mathbf{R}'$ is a vector representation of a path. Let $\mathbf{r}_P$ be the row of $\mathbf{R}'$ that corresponds to path P. Therefore,

$$\begin{aligned} \| \mathbf{R}'\mathbf{w}_{S^c} \|_1 &= \sum_{P \in N(S)} |\mathbf{r}_P \cdot \mathbf{w}_{S^c}| \\ &\leq \sum_{P \in N(S)} \sum_{j \in S^c} |w_j| \boldsymbol{\mathcal{I}}_{l_j \in P} \\ &= \sum_{j \in S^c} |w_j| \sum_{P \in N(S)} \boldsymbol{\mathcal{I}}_{l_j \in P} \\ &\leq \sum_{j=1}^{n} |w_j| \sum_{P \in N(S)} \boldsymbol{\mathcal{I}}_{l_j \in P}, \end{aligned} \tag{A.59}$$

where $\cdot$ is the inner products of two vectors.

The above-mentioned inequality is valid for every realization of the S of random variable $\boldsymbol{\mathcal{S}}$. By taking the expectation over all possible outcomes of $\boldsymbol{\mathcal{S}}$, which is also taken over all possible combinations of congested links, we have

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\mathcal{S}}} \left[\| \mathbf{R}'\mathbf{w}_{\boldsymbol{\mathcal{S}}^c} \|_1 \right] &\leq \mathbb{E}_{\boldsymbol{\mathcal{S}}} \left[\sum_{j=1}^{n} |w_j| \sum_{P \in N(\boldsymbol{\mathcal{S}})} \boldsymbol{\mathcal{I}}_{l_j \in P} \right] \\ &= \sum_{j=1}^{n} |w_j| \mathbb{E}_{\boldsymbol{\mathcal{S}}} \left[\sum_{P \in N(\boldsymbol{\mathcal{S}})} \boldsymbol{\mathcal{I}}_{l_j \in P} \right]. \end{aligned} \tag{A.60}$$

Random variable $\boldsymbol{\mathcal{S}}$ presents some random nodes on the right-hand side of the expander graph. These nodes are connected to set of nodes $N(\boldsymbol{\mathcal{S}})$ on the right-hand side. Thus, $N(\boldsymbol{\mathcal{S}})$ is also a random variable[2] that presents random nodes on the right. Equivalently, $N(\boldsymbol{\mathcal{S}})$ shows random paths in the network. That means, the last summation in Eq. (A.60) can be interpreted as follows: for a given index j, how many paths from the random paths $N(\boldsymbol{\mathcal{S}})$ is the link l_j connected to [3]. For any realization of $\boldsymbol{\mathcal{S}}$, we have $|N(\boldsymbol{\mathcal{S}})| \leq kd$. In addition, $\boldsymbol{\mathcal{I}}_{l_j \in P}$ are i.i.d random variables. Thus, for a given index j, link l_j is connected, on average, to $\frac{k\,d^2}{r}$ number of nodes in $N(\boldsymbol{\mathcal{S}})$. Therefore, we can derive the following upper bound:

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\mathcal{S}}}\left[\| \mathbf{R}'\mathbf{w}_{\boldsymbol{\mathcal{S}}^c} \|_1\right] &\leq \sum_{j=1}^{n} |w_j| kd \cdot \frac{d}{r} \\ &= \frac{kd^2}{r} \| \mathbf{w} \|_1 . \end{aligned} \tag{A.61}$$

For $\epsilon > 0$, $k \ll n$, $r \ll n$, in a $(2k, d, \epsilon)$-expander graph with n number of nodes on the left and r number of nodes on the right-hand side, the following inequality is proved [27]:

$$\frac{2kd}{\epsilon} \leq r. \tag{A.62}$$

Finally, the following upper bound for $\mathbb{E}\left[\| \mathbf{R}'\mathbf{w}_{S^c} \|_1\right]$ is derived:

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\mathcal{S}}}\left[\| \mathbf{R}'\mathbf{w}_{\boldsymbol{\mathcal{S}}^c} \|_1\right] &\leq \frac{d\epsilon}{2}\mathbb{E}_{\boldsymbol{\mathcal{S}}}[\| \mathbf{w} \|_1] \\ &= \frac{d\epsilon}{2}\mathbb{E}_{\boldsymbol{\mathcal{S}}}[\| \mathbf{w}_{\boldsymbol{\mathcal{S}}} \|_1] + \mathbb{E}_{\boldsymbol{\mathcal{S}}}[\| \mathbf{w}_{\boldsymbol{\mathcal{S}}^c} \|_1]. \end{aligned} \tag{A.63}$$

Substituting Eq. (A.63) into Eq. (A.56) results in

$$\mathbb{E}_{\boldsymbol{\mathcal{S}}}[\| \mathbf{w}_{\boldsymbol{\mathcal{S}}} \|_1] \leq \frac{\epsilon}{(2-5\epsilon)}\mathbb{E}_{\boldsymbol{\mathcal{S}}}[\| \mathbf{w}_{\boldsymbol{\mathcal{S}}^c} \|]. \tag{A.64}$$

Given that $\epsilon \leq \frac{1}{4}$, we have

$$\frac{\epsilon}{(2-5\epsilon)} \leq 2\epsilon, \tag{A.65}$$

and therefore,

$$\mathbb{E}_{\boldsymbol{\mathcal{S}}}[\| \mathbf{w}_{\boldsymbol{\mathcal{S}}} \|_1] \leq 2\epsilon\mathbb{E}_{\boldsymbol{\mathcal{S}}}[\| \mathbf{w}_{\boldsymbol{\mathcal{S}}^c} \|]. \tag{A.66}$$

---

[2] A function of a random variable is a random variable.

[3] Recall that we label the n links inside the network $l_1, l_2, \ldots, l_n$ where the delay of link l_j is the jth entry in delay vector $\mathbf{x}$.

Inequality (A.66) is similar to Eq. (A.23), except for the expected value; i.e., $\mathbb{E}[\cdot]$. The rest of the proof is similar to those of Theorems 9 and 10, in which $\| \mathbf{x}-\mathbf{x}' \|_1$ is substituted with $\mathbb{E}[\| \mathbf{x}-\mathbf{x}' \|_1]$.

■

A.3 Proofs of Theorems: Section 4.1

Lemma 3: At $t = \mathcal{T}_i$, the probability that any two nodes (e.g. u, v) have the same subspace can be bounded from above as follows:

$$P(S_u(\mathcal{T}_i) \equiv S_v(\mathcal{T}_i)) \leq \sum_{k=1}^{N-1} \min(\frac{i}{k}, \frac{N-i}{N-k-1}) P(dim(S_v(t)) = k+1).$$

Proof: For the sake of simplicity, we will write $S_u(\mathcal{T}_i)$ and $S_v(\mathcal{T}_i)$ as S_u and S_v respectively. By using law of total probability we have:

$$P(S_u \equiv S_v) = \sum_{k=1}^{N-1} P(S_u \equiv S_v | dim(S_v) = k+1) P(dim(S_v) = k+1). \tag{A.67}$$

The reason that in the summation k starts from one instead of zero is that $P(S_u = S_v | dim(S_v) = 1) = 0$. That is because we assume nodes initially contain different information packets. To calculate $P(S_u = S_v | dim(S_v) = k+1)$, let define E_v as the set of all nodes (including v itself) which spans the same space as S_v:

$$E_v = \{u | S_v \equiv S_u\}. \tag{A.68}$$

Since we know $dim(S_v) = k+1$, each member of E_v has dimension $k+1$, which means we have total dimension *increase* of at least $|E_v|k$. At time $t = \mathcal{T}_i$, dimension of whole system is at most by $i(N-1)$ by definition of $\mathcal{T}_i$ in Eq. (4.10). Therefore the following inequality holds:

$$|E_v|k \leq i(N-1), \tag{A.69}$$

which implies

$$|E_v| \leq i(N-1)/k. \tag{A.70}$$

On the other hand, at time $t = \mathcal{T}_i$, nodes need to capture $N(N-1) - i(N-1) = (N-i)(N-1)$ dimension to stop. Node is set E_v, each need $N-k-1$ dimension increase. So the total amount of dimension increase nodes in E_v require is $|E_v|(N-k-1)$ which cannot be greater than total amount of needed dimention at time $\mathcal{T}_i$. Hence the following inequality holds:

$$|E_v| \leq \frac{(N-i)(N-1)}{(N-k-1)} \tag{A.71}$$

Since $|E_v|$ satisfies both inequalities in Eq. (A.70) and Eq. (A.70), it satisfies the following:

$$|E_v| \leq (N-1)\min\left(\frac{N-i}{N-k-1}, \frac{i}{k}\right). \tag{A.72}$$

Now we can calculate $P(S_u \equiv S_v | dim(S_v) = k+1)$. We know that $|E_v|$ nodes have the same dimension as node v. Hence a randomly selected node u (from $N-1$ posibilities) has the same dimension of node v with probability $\frac{|E_v|}{N-1}$. Therefore,

$$P(S_u \equiv S_v | dim(S_v) = k+1) \leq \min\left(\frac{N-i}{N-k-1}, \frac{i}{k}\right). \tag{A.73}$$

By substituting these results in Eq. (A.74), we have the following inequality:

$$P(S_u(\mathcal{T}_i) \equiv S_v(\mathcal{T}_i)) \leq \sum_{k=1}^{N-1} \min\left(\frac{N-i}{N-k-1}, \frac{i}{k}\right) P(dim(S_v) = k+1). \tag{A.74}$$

■

Lemma 4: Probability that node u has dimension k at time $\mathcal{T}_i$ is given by:

$$P(dim(S_u(\mathcal{T}_i)) = k) = \frac{\sum_{j=0}(-1)^j C(N-1, j) C(i(N-1) + N - 2 - jN - k, N-2)}{\sum_{j=0}(-1)^j C(N, j) C(i(N-1) + N - 1 - jN, N-1)}.$$

Proof: The probability of node u having dimension $k+1$ at $\mathcal{T}_i$ is equivalent to the probability of a dimension increase of k for node u at $\mathcal{T}_i$. At time $\mathcal{T}_i$, by definition, the overall dimension increase among all nodes is $i(N-1)$. So, the problem is from the $i(N-1)$ increased dimension, which is the probability that k goes to node u given that the dimension

increase of each node cannot be greater than $N-1$. This is a traditional ball and bins problem.

All possible outcomes of the ball and bins problem must have the same probability of occurrence. Here, all possible outcomes have the same probability because of the following assumptions:

- Nodes are equally lucky in capturing the channels and transmitting in each time slot.
- $P_{uv} = P_{vu}$
- In the initial state (beginning of dissemination), every node has one and only one information packet.

■

Theorem 14: For $0 \leq i \leq N-2$*:*

$$\mathbb{E}[\mathcal{T}_{i+1} - \mathcal{T}_i] \leq \frac{N-1}{\frac{1}{2N}(\sum_{u,v \in V} P_{uv})(1 - P(S_u(\mathcal{T}_{i+1}) \equiv S_v(\mathcal{T}_{i+1})))},$$

and for $i = N-1$

$$\mathbb{E}[\mathcal{T}_N - \mathcal{T}_{N-1}] \leq \frac{N(N-1)}{\frac{1}{2N}(\sum_{u,v \in V} P_{uv})}.$$

Proof: Suppose we are in time slot t where $t \in [\mathcal{T}_i, \mathcal{T}_{i+1})$. By definition, $D(t)$ presents a dimension increase at time t. At time slot t, node u would capture the channel and start transmitting with probability $1/N$. By our assumption, each node transmits one unit of information at each time slot; thus, when node u transmits, the dimension of node v would increase by 1 if it receives the packet transmitted by u (with probability P_{uv}) and if the packet contains at least one dimension that is not in v. This can be written as follows:

$$\mathbb{E}[D(t+1) - D(t)] = \sum_u \frac{1}{N} \sum_{v \neq u} P_{uv} P(\text{packet contains at least one dimension which is not in } v). \tag{A.75}$$

The second probability in above equation can be calculated using the following two events:

1. Set S_u should not fall into set S_v or mathematically speaking $S_u(t) \nsubseteq S_v(t)$.

2. Given above, the NC coefficients must be selected in such a way to cover some dimensions from $S_u(t)$ which are not in $S_v(t)$.

The second event above is calculated in [42] to be:

$$1 - (\frac{1}{q})^{|(S_u(t) \cap S'_v(t))|}, \text{ given } S_u(t) \nsubseteq S_v(t), \tag{A.76}$$

where $S'_v(t)$ is complement of $S_v(t)$ and the finite field is $\mathbb{F}_{2^q}$. Hence given u is transmitting a packet, we have:

$$\begin{aligned} &P(\text{packet contains at least one dimension which is not in } v) \\ &= P(S_u(t) \nsubseteq S_v(t))(1 - (\frac{1}{q})^{|(S_u(t) \cap S'_v(t))|}). \end{aligned} \tag{A.77}$$

Thus the average dimension increasing can be calculated as follows:

$$\mathbb{E}[D(t+1) - D(t)] = \sum_u \frac{1}{N} \sum_{v \neq u} P_{uv} P(S_u(t) \nsubseteq S_v(t))(1 - (\frac{1}{q})^{|(S_u(t) \cap S'_v(t))|}), \tag{A.78}$$

We assume $\frac{1}{q} \approx 0$ which is practical assumption. For example if we assign only 8 bits to NC, $\frac{1}{q} = \frac{1}{256} = .004$. In that case we can rewrite above equation as:

$$\mathbb{E}[D(t+1) - D(t)] = \sum_u \frac{1}{N} \sum_{v \neq u} P_{uv} P(S_u(t) \not\equiv S_v(t)). \tag{A.79}$$

In above summation, for any two nodes $u, v \in V$, we have a summation term $P(S_u(t) \nsubseteq S_v(t)) + P(S_v(t) \nsubseteq S_u(t))$. Clearly, $\{S_u(t) \not\equiv S_v(t)\} \subset \{S_u(t) \nsubseteq S_v(t)\} \cup \{S_v(t) \nsubseteq S_t(t)\}$ which results in the following inequalities:

$$P(S_u(t) \nsubseteq S_v(t)) + P(S_v(t) \nsubseteq S_u(t)) \geq P(S_v(t) \not\equiv S_u(t)) = 1 - P(S_v(t) \equiv S_u(t)) \forall u, v \in V. \tag{A.80}$$

Thus Eq. (A.79) can be rewritten as follows:

$$\begin{aligned}
\mathbb{E}[D(t+1) - D(t)] &= \sum_u \frac{1}{N} \sum_{v \neq u} P_{uv} P(S_u(t) \nsubseteq S_v(t)) \qquad (A.81)\\
&\geq \frac{1}{N} \sum_u \sum_{v, v>u} P_{uv}(1 - P(S_u(t) \equiv S_v(t)))\\
&\overset{(1)}{=} \frac{1}{2N} \Bigg(\sum_u \sum_{v, v>u} P_{uv}(1 - P(S_u(t) \equiv S_v(t)))\\
&\qquad + \sum_v \sum_{u, u>v} P_{vu}(1 - P(S_v(t) \equiv S_u(t))) \Bigg)\\
&= \frac{1}{2N} \sum_{u,v} P_{uv}(1 - P(S_v(t) \equiv S_u(t))),
\end{aligned}$$

where equality (1) is drived by renaming the sum indexes and the last equality is the result of having $P_{uv} = P_{vu}$ and $P(S_v(t) \equiv S_u(t)) = P(S_u(t) \equiv S_v(t))$.

We know that by definition $D(\mathcal{T}_i) = i(N-1)$ so we can add up above statement for $t \in [\mathcal{T}_i, \mathcal{T}_{i+1})$:

$$\sum_{t \in [\mathcal{T}_i, \mathcal{T}_{i+1})} \mathbb{E}[D(t+1) - D(t)] = \frac{1}{2N} \Big(\sum_{u,v \in V} P_{uv} \Big) \sum_{t \in [\mathcal{T}_i, \mathcal{T}_{i+1})} (1 - P(S_u(t) \equiv S_v(t))). \qquad (A.82)$$

In above derivation we also used this fact that $P(S_u(t) = S_v(t))$ does not depend on u and v as given by Lemma 3 and Lemma 4.

By definition of $\mathcal{T}_i$ and $D(t)$ we have:

$$\sum_{t \in [\mathcal{T}_i, \mathcal{T}_{i+1})} \mathbb{E}[D(t+1) - D(t)] = N - 1. \qquad (A.83)$$

Now we consider two possibilities for i.

1) if $0 \leq i \leq N-2$ then Obviously we have:

$$P(S_u(\mathcal{T}_{i+1}) \equiv S_v(\mathcal{T}_{i+1})) \geq P(S_u(t) \equiv S_v(t)) \forall t \in [\mathcal{T}_i, \mathcal{T}_{i+1}). \qquad (A.84)$$

By above inequality and equalities Eq. (A.82) and (A.83) we have:

$$\begin{aligned} N-1 &= \frac{1}{2N}\Big(\sum_{u,v\in V} P_{uv}\Big) \sum_{t\in[\mathcal{T}_i,\mathcal{T}_{i+1})} (1-P(S_u(t)\equiv S_v(t))) \\ &\geq \frac{1}{2N}\Big(\sum_{u,v\in V} P_{uv}\Big) \sum_{t\in[\mathcal{T}_i,\mathcal{T}_{i+1})} (1-P(S_u(\mathcal{T}_{i+1})\equiv S_v(\mathcal{T}_{i+1}))) \\ &= \frac{1}{2N}\Big(\sum_{u,v\in V} P_{uv}\Big)(1-P(S_u(\mathcal{T}_{i+1})\equiv S_v(\mathcal{T}_{i+1})))(\mathbb{E}[\mathcal{T}_{i+1}-\mathcal{T}_i]). \end{aligned} \tag{A.85}$$

and by rearranging the terms we get:

$$\mathbb{E}[\mathcal{T}_{i+1}-\mathcal{T}_i] \leq \frac{N-1}{\frac{1}{2N}(\sum_{u,v\in V} P_{uv})(1-P(S_u(\mathcal{T}_{i+1})\equiv S_v(\mathcal{T}_{i+1})))},\ i=0,...,N-2. \tag{A.86}$$

2) if $i=N-1$, then by definition, for all $u,v\in V$ $S_u(\mathcal{T}_N)\equiv S_v(\mathcal{T}_N)$ which doesn't help us to obtain a bound on $\mathbb{E}[\mathcal{T}_{i+1}-\mathcal{T}_i]=\mathbb{E}[\mathcal{T}_N-\mathcal{T}_{N-1}]$. For that let consider a time slot before $\mathcal{T}_N$; i.e., $t=\mathcal{T}_N-1$. In that time slot the following two statements are valid

- At least one of the nodes does not have the whole space; i.e. $\exists u\in V s.t. |S_u(t)|<N$
- At least one of the nodes does have the whole space; i.e. $\exists v\in V s.t. |S_v(t)|=N$

From the above statements we can conclude:

$$P(S_u(t)\equiv S_v(t)) \leq \frac{N-1}{N} \text{ for } t\in[\mathcal{T}_{N-1},\mathcal{T}_N). \tag{A.87}$$

By substituting above equation in (A.79) and following the same argument as we did for the first case we will have:

$$\mathbb{E}[\mathcal{T}_N-\mathcal{T}_{N-1}] \leq \frac{N(N-1)}{\frac{1}{2N}(\sum_{u,v\in V} P_{uv})}. \tag{A.88}$$

■

Theorem 15: Let $\mathcal{T}$ be stopping time. Then

$$\begin{aligned} \mathbb{E}[\mathcal{T}] &= \mathbb{E}[\mathcal{T}_N] \\ &\leq \frac{N-1}{\frac{1}{2N}\sum_{u,v\in V} P_{uv}}\Big(\sum_{i=1}^{N-1} \frac{1}{1-P\big(S_u(\mathcal{T}_i)\equiv S_v(\mathcal{T}_i)\big)}+N\Big). \end{aligned}$$

Proof:

$$\begin{aligned}\mathbb{E}[\mathcal{T}] &= \mathbb{E}[\mathcal{T}_N] \\ &= \sum_{i=1}^{N}(\mathbb{E}[\mathcal{T}_i - \mathcal{T}_{i-1}]) \\ &= \sum_{i=1}^{N-1}(\mathbb{E}[\mathcal{T}_i - \mathcal{T}_{i-1}]) + \mathbb{E}[\mathcal{T}_N - \mathcal{T}_{N-1}] \\ &\leq \frac{N-1}{\frac{1}{2N}\sum_{u,v\in V} P_{uv}}\Big(\sum_{i=1}^{N-1}\frac{1}{(1-P(S_u(\mathcal{T}_i)\equiv S_v(\mathcal{T}_i)))} + N\Big).\end{aligned} \tag{A.89}$$

■

Corollary 5: In a fully connected wireless network in which every node can receive packet from any other node in the network with a non-zero probability, the average stoping time is of order $O(N)$ when network coding is applied.

Proof: First let prove the following lemma:

lemma 13 *There exist a finite number β such that the following is held:*

$$\sum_{i=1}^{N-1}\frac{1}{(1-P(S_u(\mathcal{T}_i)\equiv S_v(\mathcal{T}_i)))} \leq N\beta \tag{A.90}$$

Proof: By definition of the stopping time $\mathcal{T}$, $P(S_u(t)\equiv S_v(t)) < 1$ for $t < \mathcal{T}$. Let α be the largest number less than 1 such that $P(S_u(t)\equiv S_v(t)) \leq \alpha < 1$. Then

$$\sum_{i=1}^{N-1}\frac{1}{(1-P(S_u(\mathcal{T}_i)\equiv S_v(\mathcal{T}_i)))} \leq \frac{1}{1-\alpha}\sum_{i=1}^{N-1} 1 = \frac{N-1}{1-\alpha}. \tag{A.91}$$

□

We are assuming that nodes can receive packets from each other with probability at leat p. In other words, each node is in transmission rage of all other nodes and p is the smallest reception probability between every two nodes. Note that for $p = 1$, the wireless network acts like a wired network with a complete graph. Mathematically speaking:

$$P_{uv} \geq p \ \forall u, v \in V. \tag{A.92}$$

Using the above equation we have:

$$\sum_{u,v\in V} P_{uv} \geq N^2 \cdot \sum_{u,v\in V} p = N^2 \cdot p \tag{A.93}$$

Using the above equation and Lemma 13, Eq. 4.15 can be bounded as:

$$\begin{aligned} \mathbb{E}[\mathcal{T}] &\leq \frac{N-1}{\frac{1}{2N}\sum_{u,v\in V} P_{uv}} \Big(\sum_{i=1}^{N-1} \frac{1}{(1 - P(S_u(\mathcal{T}_i) \equiv S_v(\mathcal{T}_i)))} + N \Big) \\ &\leq \frac{N-1}{\frac{Np}{2}} (N\beta + N) = \frac{2}{p}(N-1)(1+\beta). \end{aligned} \tag{A.94}$$

Therefore, in this case time is of order $O(N)$.

Corollary 6: In a sparsely connected wireless network the average stoping time is of order $O(N^2)$ when network coding is applied.

In this case each node only receives packets from at most two other nodes. Let assume $p > 0$ be the smallest reception probabilities for all the nodes inside the network. Note that p can not be zero because we are assuming the network is connected which means every node receives data with non-zero probability from at leat one of its neighbors. In other words:

$$\forall u \in V \;\; \exists v \in V \text{ s.t. } P_{vu} \geq p > 0, P_{zu} = 0 \;\; \forall z \in V - \{v\} \tag{A.95}$$

By above formula we have:

$$\sum_{u,v\in V} P_{uv} \geq N\, p \tag{A.96}$$

By substituting above inequality and the result of Lemma 13 in Eq. 4.15 we have:

$$\begin{aligned} \mathbb{E}[\mathcal{T}] &\leq \frac{N-1}{\frac{1}{2N}\sum_{u,v\in V} P_{uv}} \Big(\sum_{i=1}^{N-1} \frac{1}{(1 - P(S_u(\mathcal{T}_i) \equiv S_v(\mathcal{T}_i)))} + N \Big) \\ &\leq \frac{N-1}{\frac{p}{2}} (N\beta + N) = \frac{2}{p}(N-1)N(1+\beta). \end{aligned} \tag{A.97}$$

Therefore, in this case time is of order $O(N^2)$.

A.4 Proofs of Theorems: Section 4.2

Proof of Lemma 7:

If we know $\mathcal{S}_{i-1} = s_i$, it means AP has $M - s_i$ new packets to send to the vehicles. Having that, The denominator equals all possible $m = 1$ objects from $M - s_i$ chosen without replacement. The numerator represents all the possibilities such that there are exactly x common among the m-set at each of the nodes.

■

Proof of lemma 10

For matrix $\mathbf{A}_{m\times n}$, we define $span\{\mathbf{A}_{m\times n}\}$ as the space spanned by rows of matrix $\mathbf{A}_{m\times n}$. By definition of stopping time $\mathcal{T}_{I2V}$ in Eq. (4.33), the event of $\mathcal{T} = t$ is equivalent of having $rank(\mathbf{A}_{2t\times M}) = M$. Now, let $\mathbf{a}_{2t}$ be the last row of matrix $\mathbf{A}_{2t\times M}$ then we have:

$$\begin{aligned} P(\mathcal{T}_{I2V} = t) &= P(\mathbf{A}_{2t\times M} = M) \\ &= P(rank(\mathbf{A}_{2t-1\times M}) = M-1 \ \&\ \mathbf{a}_{2t} \notin span\{\mathbf{A}_{2t-1\times M}\}) \\ &= P(rank(\mathbf{A}_{2t-1\times M}) = M-1) P(\mathbf{a}_{2t} \notin span\{\mathbf{A}_{2t-1\times M}\} | rank(\mathbf{A}_{2t-1\times M}) = M-1). \end{aligned} \tag{A.98}$$

Given the fact that rank of $\mathbf{A}_{2t-1\times M}$ is $M-1$, the probability of adding a row which is not independent of all the other rows can be bounded from below as follows:

$$\begin{aligned} &P(\mathbf{a}_{2t} \in span\{\mathbf{A}_{2t-1\times M}\} | rank(\mathbf{A}_{2t-1\times M}) = M-1) \\ &\geq P(\mathbf{a}_{2t} = 0 | rank(\mathbf{A}_{2t-1\times M}) = M-1) \\ &= \frac{1}{2^q}. \end{aligned}$$

Therefore, we have the following bound for having the last row$\mathbf{A}_{2t\times M}$ being independent of the rest of the rows:

$$\begin{aligned} &P(\mathbf{a}_{2t} \notin span\{\mathbf{A}_{2t-1\times M}\} | rank(\mathbf{A}_{2t-1\times M}) = M-1) \\ &= 1 - P(\mathbf{a}_{2t} \in span\{\mathbf{A}_{2t-1\times M}\} | rank(\mathbf{A}_{2t-1\times M}) = M-1) \\ &\leq \frac{1}{2^q}. \end{aligned} \tag{A.99}$$

On the other hand:

$$\begin{aligned} P(\mathbf{A}_{2t-1\times M} = M-1) &\leq P(\mathbf{A}_{2t-1\times M} \leq M-1) \\ &\leq 1 - P(\mathbf{A}_{2t-1\times M} = M) \\ &= \frac{1}{(2^q)^{2t-n}}. \end{aligned} \tag{A.100}$$

By substituting Eq. (A.99) and Eq. (A.98) in Eq. (A.98) the lemma is proven.

■

proof of Theorem 18:

$$\begin{aligned} E[\mathcal{T}_{NC}] &= \sum_{t=\frac{M}{2}}^{M} t \cdot P(\mathcal{T}_{NC} = t) \\ &\leq \sum_{t=\frac{M}{2}}^{M} t \cdot (1 - \frac{1}{2^q}) \frac{1}{(2^q)^{2t-M}} \\ &= (1 - \frac{1}{2^q})(2^q)^M \sum_{t=\frac{M}{2}}^{M} t(2^{-q})^{2t} \\ &\approx \frac{M}{2} + \frac{1}{2^q - 1}. \end{aligned} \tag{A.101}$$

■